

# HistoCrypt 2020

Proceedings of the 3rd  
International Conference on  
Historical Cryptology

# Proceedings of the 3<sup>rd</sup> International Conference on Historical Cryptology HistoCrypt 2020

Editor  
Beáta Megyesi

Published by:

NEALT Proceedings series 44  
Linköping University Electronic Press, Sweden  
Linköping Electronic Conference Proceedings, No. 171  
ISSN: 1650-3686  
eISSN: 1650-3740  
ISBN: 978-91-7929-827-2  
URL: <http://www.ep.liu.se/ecp/contents.asp?issue=171>





## SPONSORS



UPPSALA  
UNIVERSITET



VETENSKAPSRÅDET  
THE SWEDISH RESEARCH COUNCIL





## Preface

We are very pleased to introduce the proceedings of the 3rd International Conference on Historical Cryptology, HISTOCRYPT 2020. The conference would have taken place in Budapest, Hungary, between June 15 and 17, 2020 but due to the COVID-19 crisis with closed borders, travel restrictions and physical distancing, the actual meeting of HISTOCRYPT had to be canceled.

Just as in previous years, HISTOCRYPT 2020 addresses all aspects of historical cryptology/cryptography including work in closely related disciplines (such as history, history of ideas, computer science, AI, computational linguistics, linguistics, or image processing) with relevance to historical ciphertexts and codes. The subjects of the conference include, but are not limited to the use of cryptography in military, diplomacy, business, and other areas, analysis of historical ciphers with the help of modern computerized methods, unsolved historical cryptograms, the Enigma and other encryption machines, the history of modern (computer-based) cryptography, linguistic aspects of cryptology, the influence of cryptography on the course of history, or teaching and promoting cryptology in schools, universities, and the public.

The scientific program was carefully planned by an international scientific program committee, consisting of researchers in cryptology, history, intelligence and language technology. The program committee welcomed submissions in three distinct tracks: *regular papers* on substantial, original, and unpublished research, including evaluation results, where appropriate; *short papers* on smaller, focused contributions, work in progress, negative results, surveys, or opinion pieces; and *system demos and artifacts* presented as short papers.

The conference received 20 submissions from all over Europe including the Czech republic, France, Germany, Hungary, Italy, Poland, Slovakia, Spain, Sweden, and the UK as well as from Australia, Israel and the United States.

Following the previous events, our primary goal in the program committee was to deliver a high quality program with a wide variety of topics. We applied a double-blind review process and all papers were reviewed by at least three experts in the field. To synchronise recommendations among the reviewers, the senior members of the PC lead the discussion among reviewers on the submissions. The final selection of the papers was made by the senior members of the program committee. We rejected three papers and accepted 85% of the submissions, of which thirteen papers were submitted as long and four were submitted as short papers. All accepted submissions are collected in this volume in alphabetical order after the last name of the first author.

Originally, we also planned for four invited keynote speakers who kindly accepted our invitation: *David Kenyon*, research historian at Bletchley Park and Associate Lecturer in History at Brunel University, and author of the recently published *Bletchley Park and D-Day*; *Liza Mundy*, well-known journalist and author of the *Code Girls: The Untold Story of the American Women Code Breakers of World War II* in the United

States; *Paul Zimmermann*, researcher at Inria, the French National Institute for Research in Digital Science and Technology in Nancy, France, focusing on integer factorization, and *Gerhard F. Strasser*, professor emeritus of German and Comparative Literature at the Pennsylvania State University. After a special invitation from the program committee, we are happy to present Gerhard F. Strasser's work on *Encoded Communication with Ladies in a Turkish Harem, 17th-Century Style* as the first article in the proceedings.

Lastly, the conference program would have included a workshop about the well-known Rohonc Codex, which is located in Budapest. The workshop was planned by Levente Zoltán Király and Gábor Tokai, who are working on the decipherment of this mysterious manuscript from the 15th century. We hope that they will be given the chance to organize the event in connection to another HISTOCRYPT meeting in the near future.

Organizing a conference relies on the goodwill of many colleagues who take their valuable time to contribute to an interesting and fruitful conference. I am very grateful to all senior members of the program committee, Carola Dahlke, Bernhard Esslinger, Benedek Láng, George Lasry, and Dermot Turing for their wise advice and dedication, and the 23 reviewers for their time and effort to give constructive and collegial feedback to help in the selection of papers. All authors without whom these proceedings would not have seen light receive hereby a huge thanks.

Even though we did not get the chance to organize a physical meeting, my greatest debt goes to the local organization, Benedek Láng and Anna Lehofer, whom I always enjoy working with, for carrying the burden of the local organization — it is very sad that we had to cancel the conference when almost everything was in place...

I hope to meet you all at the next HISTOCRYPT in 2021 and I wish you all a joyful time while reading the papers in this volume!

*Beáta Megyesi*

Program Chair for HISTOCRYPT 2020

## **Program Committee**

- Beáta Megyesi (Program Chair), Uppsala University, Sweden
- Carola Dahlke, Deutsches Museum, Germany
- Bernhard Esslinger, University of Siegen, Germany
- Benedek Láng, Budapest University of Technology and Economics, Hungary
- George Lasry, The CrypTool Team, Germany
- Dermot Turing, Kellogg College, Oxford, UK

## **Local Organizing Committee**

- Benedek Láng (Local Chair), Budapest University of Technology and Economics, Hungary
- Anna Lehofer, Budapest University of Technology and Economics, Hungary

## **Steering Committee**

- Arno Wacker, Bundeswehr University Munich, Germany
- Joachim von zur Gathen, Emeritus, Bonn-Aachen International Center for Information Technology, Germany
- Marek Grajek, Poland
- Klaus Schmeh, Private researcher, Germany

## **Extended Program Committee: Reviewers**

- Eugen Antal, Slovak University of Technology in Bratislava, Slovakia
- Paolo Bonavoglia, Mathesis Venezia, Italy
- Nicolas Courtois, University College London, UK
- Camille Desenclos, Université de Haute-Alsace, France
- Joachim von zur Gathen, Emeritus, Bonn-Aachen International Center for Information Technology, Germany
- Pascal Junod, Snap, Switzerland
- Otokar Grošek, Slovak University of Technology in Bratislava, Slovakia



- Bradley Hauer, University of Alberta, Canada
- Julio Hernandez-Castro, School of Computing, University of Kent, UK
- Kevin Knight, DiDi Labs, USA
- Jozef, Kollár, Slovak University of Technology in Bratislava, Slovakia
- Grzegorz Kondrak, University of Alberta, Canada
- Nils Kopal, University of Siegen, Germany
- Karl de Leeuw, University of Amsterdam, Netherlands
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Michal Musilek, University of Hradec Kralove, Czech Republic
- Valerie Nachev, UCL Cergy Paris Université, France
- Diego Navarro, Carlos III University of Madrid, Spain
- Jacques Patarin, Université de Versailles-Saint-Quentin-en-Yvelines, France
- Klaus Schmeh, Cryptovision, Germany
- Serge Vaudenay, Ecole Polytechnique Fédérale de Lausanne, Switzerland
- Arno Wacker, Bundeswehr University of Munich, Germany
- Pavol Zajac, Slovak University of Technology in Bratislava, Slovakia

# Contents

Preface .....	v
Gerhard F. Strasser .....	1
<i>“Encoded” Communication with Ladies in a Turkish Harem, 17th-Century Style</i>	
Eugen Antal and Pavol Zajac .....	18
<i>HCPortal Overview</i>	
Eugen Antal, Pavol Zajac and Otokar Grošek.....	21
<i>Diplomatic Ciphers Used by Slovak Attaché During the WW2</i>	
Richard Bean.....	31
<i>The Use of Project Gutenberg and Hexagram Statistics to Help Solve Famous Unsolved Ciphers</i>	
Paolo Bonavoglia.....	36
<i>A Partenio’s Stegano-Crypto Cipher</i>	
Paolo Bonavoglia.....	46
<i>Trithemius, Bellaso, Vigenère – Origins of the Polyalphabetic Ciphers</i>	
Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés and Beáta Megyesi.....	52
<i>A Web-based Interactive Transcription Tool for Encrypted Manuscripts</i>	
Carola Dahlke.....	60
<i>The Auxiliary Devices of OKW/Chi</i>	
Marek Grajek.....	70
<i>Dawn of Mathematical Cryptology: Probabilists vs Algebraists; Algebraists &amp; Probabilists?</i>	
Nils Kopal.....	77
<i>Of Ciphers and Neurons – Detecting the Type of Ciphers Using Artificial Neural Networks</i>	
Benedek Láng.....	87
<i>Was it a Sudden Shift in Professionalization? Austrian Cryptology and a Description of the Staatskanzlei Key Collection in the Haus-, Hof- und Staatsarchiv of Vienna</i>	
George Lasry.....	96
<i>Solving a Tunny Challenge with Computerized “Testery” Methods</i>	
Beáta Megyesi.....	106
<i>Transcription of Historical Ciphers and Keys</i>	
Štefan Porubský.....	116
<i>A Hungarian Cryptological Manual in Berlin</i>	

Klaus Schmeh.....	126
<i>The Zschweigert Cryptograph – A Remarkable Early Encryption Machine</i>	
George Teşeleanu.....	135
<i>Cracking Matrix Modes of Operation with Goodness-of-Fit Statistics</i>	
Crina Tudor, Beáta Megyesi and Benedek Láng.....	146
<i>Automatic Key Structure Extraction</i>	
Viktor Wase.....	153
<i>The Role of Base 10 in the Beale Papers</i>	



# “Encoded” Communication with Ladies in a Turkish Harem, 17<sup>th</sup>-Century Style

Gerhard F. Strasser

Prof. emeritus, Penn State University, Depts. of German and Comparative Literature

[gfs1@psu.edu](mailto:gfs1@psu.edu)

## Abstract

The Duke August Library in Wolfenbüttel, Germany, preserves a treasured French-Turkish manuscript with an intriguing (translated) title: “Silent Letters, or a Method of Making Love in Turkey without Knowing How to Read or Write.” This unusual piece was prepared in 1679 for Jacobus Colyer, the enterprising 22-year-old son of the Dutch representative to the Sublime Porte in Istanbul. The first and longest of 3 parts consists of an extensive explanatory section in French in which the author details the Turkish system of sending messages (not only to ladies in the Sultan’s *Harem*), so-called *Selams*, “welcome greetings” or “peace wishes” that are remotely similar to the Oriental “language of flowers.” These messages are encoded according to a well-defined system. Without any extant “code books” beyond what the 1679 Wolfenbüttel and scarce later sources yield it becomes clear that the meaning of such encoded *Selam* messages was common knowledge among interested parties—in particular in the Sultan’s *Harem*.

The following analysis will detail this system and also branch out to show how in 1688 this manuscript was adapted in two initially identical publications with totally different endings. Both of them include a reference to the “*Langage muet*”, an early sign language used at the Sultan’s court—*de facto* a second cryptological example associated with the Wolfenbüttel manuscript and an ingenious re-use of the same material for different audiences.

## 1 Preliminaries

In what was to be a presentation of 17<sup>th</sup>-century material at HistoCrypt 2020 I want to analyze a cryptologic manuscript that I unearthed some time ago in the holdings of the Duke August Library in

Wolfenbüttel, Germany, a treasure house of such materials due to the Duke’s own interests in the field of secret communication. Before taking a closer look at this fascinating—and also amusing—manuscript here is an overview of the presentation:

1. Preliminaries – Description of manuscript, its dedicatee, its author
2. Characterization of the “Language of Flowers” and the “Language of Symbols”
3. Overview of the subsequent twenty-one encoded messages in Part I of the manuscript
4. The remainder of this manuscript
5. Confirmation of these *Selams* in later sources
6. The practical application of such non-verbal communication in the two divergent endings of the *Histoire Galante* of 1688 – with an inserted “Excursus” presenting a second cryptographic means of communication, “*Langage muet*” or ‘Silent Language’, an early kind of sign language
7. Closing analysis

Let me now set the stage: The manuscript in question is titled “*Lettres muettes*”—or, to list here its full designation, *Lettres muettes, ou la maniere de faire l’amour en Turquie / Sans Scavoir ny Lire ny Ecrire* (Silent Letters, or the Manner of Making Love in Turkey / Without Knowing how to Read or Write) (Fig. 1).

The manuscript is kept in a folder containing “Cryptographica”, which holds a number of ciphers and nomenclators dating from the latter part of the 17<sup>th</sup> century.<sup>1</sup> The librarian who catalogued this material a long time ago may have included the *Lettres muettes* for three enciphered “sexually allusive” French words in the third part of the manuscript (their Turkish equivalents show the words in 17<sup>th</sup>-century vulgar Turkish usage ...) (Fig. 2).

<sup>1</sup> Herzog August Bibliothek, Cod. Guelf. 389 Nov. 2°. Part (a) contains the various ciphers and nomenclators; (b) is the manuscript in question, the “*Lettres muettes*” (referred to in the text as “*Lm*”, with each of the 3 parts listed before the folio numbers). It consists of 3 parts: Pt. I, 18 pages; Pt. II, 14 pages;

Pt. III (separately listed as (c), 20 pages. Part I is a careful copy, the other two are hastily penned down originals. See Strasser (1988), pages 511-514.

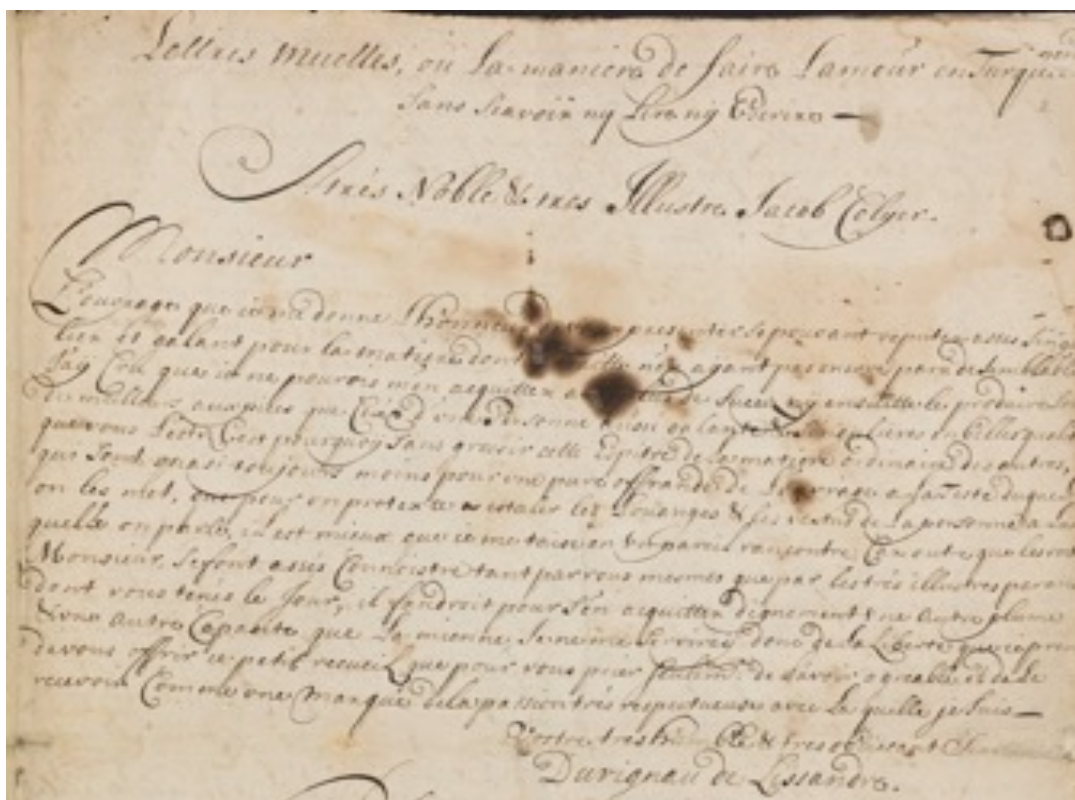


Figure 1: Dedicatory (top) part of *Lettres muettes* manuscript. Courtesy Herzog August Bibliothek, Wolfenbüttel, for all such illustrations.

French	Enciphered	Modern Turkish
Manger	manger on Jer	
Je vous salue	Salam a lekij	
un Chien	bi küpek	
un Boeuf	..... Begir	
un Pain	Eckmeck	
Vilites	Salamlama	
Entrée	Jelbichere	
Porte	Schick	[modern: çük] = penis
QT4X → 02	Sick	[mod. ham = crude, virgin] = vulva
QT4 → 7T5	Ham	
D7X7EL7	Sickme	[çiftleşmek] = copulation
Dormir	Ojir	
Je baigne	Amam	
aller a la Chasse	Alwath Schickma	
Je vous fais mes respects	ben Sama bonu peskies	
un Courteau	bit schack	
une Verre	Cade	
Ecrire	Tafar	
Je prie	ben tafarm	
Je ne pas	lut ma	
il rit	o jilet	

**Duvignau de Lissandre: “Lettres muettes [...]”, Part III: French-Turkish “Pocket Dictionary”**

3 vulgar sexual expressions still in use in the Turkish language today:

[modern: çük] = penis

[mod. ham = crude, virgin] = vulva

[çiftleşmek] = copulation

These are the only three enciphered words in Duvignau’s entire manuscript—a plausible reason for the 19th-century librarian’s including it among Duke August’s other ciphers

Figure 2: The only three enciphered words in the manuscript.

This third section is a homemade French-Turkish “pocket” dictionary, and the librarian may not necessarily have realized that the important first part relies totally on encoded messages. For different reasons all three parts are equally interesting from the point of view of the history of cryptology; of early sign languages; of cultural history in general, and lastly for linguistic matters as the Turkish language used represents a somewhat earlier stage that is not frequently documented.

For the purposes of this discussion the first section—addressed in a beautiful hand to “Trés Noble & tres [sic] Illustre Jacob Colyer” —represents the most intriguing material. The dedicatee was the 22-year-old son of the Dutch representative to the Sublime Porte, Justinus Colyer (1624-1682), who in 1668 was accredited by Sultan Mehmet IV (1642-1693; r. 1648-1687). In 1682, just before his death, Justinus appointed his son Jacobus (1657-1725) to the position of secretary to assure continuity in the Dutch representation. Two years later the States General promoted Jacobus Colyer to ambassador, a function he held until his death. In 1679, when the manuscript was dedicated to him, he had already spent more than a decade in Constantinople and not only mastered Ottoman Turkish, Greek, French and Italian but was apparently also rather knowledgeable in the ways in which contacts with Turkish women could be established in a culture that virtually secluded them from the outside world.

For this very reason an encoded language had developed that may have originated in the “language of flowers” (Cornelissen, 2005; Kakuk 1970; Kakuk and Öztürk, 1986). A nineteenth-century editor of some forty samples of such communication described the situation of Turkish women in his day as follows:

All Turkish women wear a burqa or robe that covers them from head to toe. They cannot be recognized but see everything and everyone. Unfortunately, they have no way of expressing their feelings to whomever their heart would select. They cannot write and are not allowed to speak with strange men. Thanks to their ingenuity they nonetheless created a well-tried means, the “language of flowers” or, to be precise, the language of symbols. In this silent conversation not only various flowers can signify a word but all visible objects that you can carry on you. When a man or a woman hands over an object to his or her beloved the recipient has to pronounce the name of the

object and find a saying that rimes with it and fits the occasion. But how is this possible? When we take into consideration that the Turkish people assign a special meaning to the individual objects and phenomena of this world, that they like to play with rimes and at any given moment are ready to pose or solve a riddle then we will hardly find this matter impossible (Hutter, 1851).

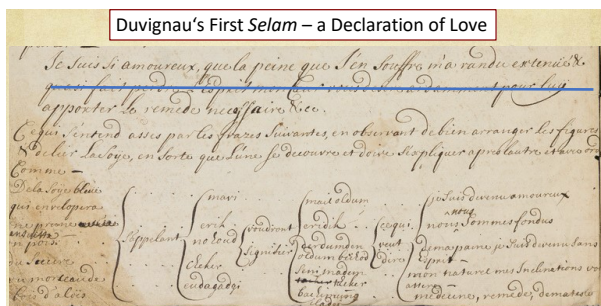
These mid-nineteenth-century observations describe the use of the “language of symbols” (Hutter’s term) in a manner that eliminates the need for “intermediaries” or go-betweens, which is indispensable in contacts initiated according to the Wolfenbüttel manuscript. This document is almost 200 years older and describes the “language of symbols or objects” for the first time in the west. It was prepared by a certain Duvignau de Lissandre, who allegedly was a secretary of the French ambassador for almost a decade, traveled extensively in the Orient and in 1687 wrote insightful, highly critical books on the Ottoman powers (Duvignau, 1687; 1688a). He also—anonously—exploited the material prepared for Colyer in two other publications that incorporated this “language of symbols” in a novellistic fashion, which will be discussed later in a detailed analysis (Duvignau, 1688a; 1688c). It is puzzling that the name “Duvignau de Lissandre” or “Sieur Du Vignau”—a diplomat who prided himself on having been in the service of “one of the ministers of the greatest king on earth”—cannot be verified in the archives of the Quai d’Orsay, the Foreign Ministry. A few years ago, a French researcher finally established the true identity of Duvignau de Lissandre—which turns out to be a pseudonym of Edouard de La Croix (1640/45-1704), who in fact in 1670 became the second secretary of the newly appointed French envoy to the Sublime Porte in Istanbul (Thépaut-Cabasset, 2007). In 1675 he was promoted to first secretary and returned to Paris in 1680; the manuscript in question—written in 1679—would therefore have been produced while he was still in Istanbul. And while there are several critical publications—highly compromising of the Turkish sultan’s court, public policy, and economy purportedly authored by Duvignau since Edouard de La Croix did not want to be identified with these materials—there are the two-volume memoirs published under Edouard de La Croix’s full name (1684), which describe his years of service in Istanbul.



## 2 Characterization of the “Language of Flowers and the “Language of Symbols”

Let us return to the manuscript, which features Duvignau’s presentation to Jacobus Colyer. After a detailed introduction highlighting the history and merits of this encoded communication, called *Selam* in Turkish or “welcome greetings” and/or “peace wishes“, Duvignau begins a listing of the various items needed in such exchanges. It turns out that such communication is based entirely on the sending of a few items that have clearly encoded meanings within each group, a system that certainly falls within the purview of a conference like “HistoCrypt.” This “Dictionary of Love,” as one could call the listings, is always arranged in the same way (Fig. 3): There are four columns where in the very left one the French items required to convey a particular meaning are lined up, followed by their Turkish equivalencies. Next to the Turkish name for each item (column 2) is the “encoded“ Turkish meaning of each of these items, which in column four is followed by an elaboration of this meaning in French. The more effusive “interpretation” of these French translations follows in the later 21 sample letters on the right but is written in this model above the four columns. While not particularly mentioned by Duvignau, the lines in this example and the later 21 letters need to be properly read horizontally, which is at times rather difficult.

Duvignau's First *Selam* – a Declaration of Love




Blue silk that contains a prune, together with	mavi erik	mail oldum eridik	I have fallen in love we have fallen for each other (literally: we have “melted”, blended)
a pea,	nohoud	derdumden oldum	I have lost my mind in my pain
a piece of sugar, and	cheker	seni madem tcheker	my nature, my inclinations attract you
a piece of aloe wood	eudgadgi	bachimung iladgi	medicine, remedy of my head

Figure 3: Duvignau’s first *Selam* with the transcription of the material in the four columns (the French in the first and last ones translated into English).

<sup>2</sup> I have fallen in love so much that the pain that I suffer [from that] has made me look emaciated and has made me lose my mind, so to speak[.] My heart desires you like a

The fifth and last section expresses the concise statements in the fourth in much more elaborate terms and almost reads like a piece taken from *A Thousand and One Nights*:

*Je suis si amoureux que la peine que j’en souffre m’a rendu [!] extenué & quasi fait perdre l’esprit [.] mon Cœur vous desire ardemment pour luiy apporter le remede necessaire.*<sup>2</sup> The 1688 English edition of *The Turkish Secretary* (Du Vignau, Sieur des Joanots, 1688) shows this very same material in a rather close translation (Fig. 4).



A Grape.	Uzum.	Uzi glafzum	My Eyes.
Blew Silk.	Mavi.	Hail oldum	I am fall'n in Love.
A Plum.	Erik.	Eridik.	We dissolve away.
A Pea.	Nohoud.	Derdumden oldum be-houd.	My torment makes me mad.
Sugar.	Cheker.	Seni ma-dem tcheker.	My Bosom longs after you.
Aloe Wood.	Eud Agadgi.	Bachimung iladgi.	Physician, Remedy of my head.
Selam.	Or a thing that is sent.	Whole Signification is	Which in English is literally
	Names.	Significa-tion.	Confirma-tion.

Figure 4: Close English translation of the first four columns of Duvignau’s model exercise.

At this point we need to differentiate the “language of flowers” per se from this system of *Selam* or “welcome greetings” (*selam*, Arabic *salām*, meaning “peace”); it may have existed in ancient China due to early pictograms incorporating floral designs—something that would even hold true for Egyptian hieroglyphs (Goody, 1993; Heilmeyer, 2006; Strasser, 2016) — and began to be known in the west in the 18<sup>th</sup> century.

In the Victorian age, in particular, when verbal communication of sentimental matters was not acceptable in higher circles of society, the significance of such floral greetings became an indispensable means of “silent” exchanges. Not only individual flowers had their encoded, well-known meaning, which to an extent has survived into the 21<sup>st</sup> century (red roses—I love you more than anything or anyone else; an anemone—I want

burning flame so that you can bring to it the necessary remedy. (Transl. G. F. St.)

to be with you forever; but also a dahlia—I am bespoken), and a combination of flowers eventually took on an even more complicated meaning that required written booklets for their decoding. An American custom going back to those days may well be the corsage that young men will present their partners at fancy balls or festive occasions—but even there the way the Victorian lady would pin these flowers on her garment already had an encoded meaning: close to her heart signified mutual feelings while a corsage put in her hairdo was tantamount to a verbal rejection. A German example from 1853 lists an ear of wheat (*Weizenähre*) that was encoded to mean “*Ich bin glücklich, denn du liebst mich wieder*” (I am happy for you love me again)—a meaning that may have survived in the third wedding anniversary in German called *Weizen-Hochzeit* (wheat wedding [anniversary]).

As can be seen in Duvignau’s example, the Turkish *Selams* went one major step beyond the customary language of flowers: The incorporation of a prune, a pea, a lump of sugar and a piece of aloe wood indicates the opening of this non-verbal system to a method in which all sorts of objects were added in, expanding it to a “language of symbols,” if you wish. The expansion seems to be a Turkish invention, and in the latter part of the 17<sup>th</sup> century this system was obviously well known. Nonetheless Duvignau cautions Colyer when he elaborates: “yet while a certain number of figures of this love cipher may be known among interested parties there is a much larger number [of such figures] with which only the experts are familiar, and which can only be learned through long practice in this art or with the help of those who know the most about them” (*Lm*, I, fol. 2v<sup>o</sup>). The author continues with a list of items that could be wrapped in a silken handkerchief (*mendil*), whose color has an encoded meaning to begin with while the size of the piece of silk, often beautifully embroidered, indicated the quality of the compliment (Hammer[-Purgstall], 1834-36; Peirce, 1993; Penzer, 1966; Walther, 1997; Coco, 2002; Roberts, 2007). This silk wrapping could include pieces of wax, iron, bread or any other items from which a word or a phrase may be gleaned that rimes with the respective item in the beginning or end of the word or expression. This is an important mnemonic aid which has to come into play when such an object is presented, whereupon its name can jog the recipient’s memory, as the 19<sup>th</sup>-century quotation spelled out: In his introductory material (see Fig. 3) Duvignau refers to the “blue color of the silk cloth,” which is *mavi*

in Turkish with the meaning *mail oldum* and signifies “I have fallen in love.” Here the rime—sometimes just an alliteration—is in the beginning of the two words, he continues, namely in “*ma*”, which occurs both in “*mail*” and in “*mavi*.” For the opposite riming scheme at the end of words Duvignau lists the example in the fourth line, namely *cheker* (sugar), which rimes with *tcheker* to elicit the metaphorical meaning of the phrase *semi madem tcheker* as “my nature, my inclinations attract you.”

This symbolic language, the author warns Mijnheer Colyer, becomes even more complicated when objects are combined with different other items. His prime example is a piece of string—Turkish *sidgim*—with the extended meaning “your itching is not yet over.” When combined with a slice of onion—*sogan*—this changes for the worse to express “get lost you daughter of a whore,” and even worse when combined with an olive, “that your bier, your dead body be paraded in front of me.” Yet an entirely opposite meaning may also occur: combined with a piece of a brush or a tassel—Turkish *supurghé*—this changes to an imploring “for once have pity with [or on] me.” All this, Duvignau implies, requires an almost total mastery in the encoding of *Selams*—this is where his manuscript becomes indispensable. He also stresses that there is no gender difference in Turkish between French “*ami*” and “*amie*”, between male and female lover.



Figure 5: A Turkish *Harem*, attributed to Franz Hörmann and Hans Gemminger, 1654. Courtesy Pera Museum, Istanbul.

There is yet another, all-important detail that needs to be observed in this encoding process. It is mandatory that the overall sequence of the items in the silken kerchief be strictly observed: These objects, the author spells out before giving his example, have to be properly arranged in the



silken wrapping so that one item can be discovered after another, and in this order (which means that they will be tied together with a silken string to reflect this important order). There remains the overarching question—not addressed in Duvignau’s preface but spelled out in his later *Histoire Galante*—as to how these *Selams* would reach their intended recipients. In this novellistic piece—as in actuality—the delivery of such silken kerchiefs was entrusted to older women—often Jewish—who customarily purchased necessities and trinkets for the ladies in the Sultan’s *Harem* and therefore passed the eunuch gatekeepers without suspicion, “go-betweens” in a literal and metaphorical sense, as will be discussed later. The exclusive attribution of part of a Turkish house to women is highly relevant to the purposes of Duvignau’s manuscript since the *Harem* or *haremluk* meant the “inviolable section of the building where all the female members of a family and their servants were living.” The remainder of the house, the *selamluk*, was reserved to men and was the public part of the building. It follows that the Sultan’s *Harem* (Fig. 5)—by the end of the 17<sup>th</sup> century already located in the *Topkapi Palace*—was not as singular a setting as one might believe; wealthy Turkish families lived in a house set up this way. Nonetheless the Sultan’s was nowhere surpassed in its importance—and in the sheer number of beautiful women within its heavily guarded walls. Leaving the Sultan’s Harem was virtually impossible while it was feasible for women of the lower classes to go to the *hamam* or public baths but only when accompanied by one or two of her servants (Fig. 6). It seems that women’s leaving their homes depended to a great extent on the local observance and interpretation of the Quran: 17<sup>th</sup>-century travelers to Persia report that women there were strictly forbidden to leave their homes while in mid-century an Italian nobleman observed the relative ease with which Turkish ladies could be seen in the bazaar in droves (Olearius, 1671; della Valle, 1674). In view of the obvious intentions for which the Wolfenbüttel manuscript seems to have been prepared this more relaxed religious observance of the Quran in a city like Istanbul is of prime importance.



Figure 6: Francis Smith, A Turkish Lady going to the Bath with her slave, c. 1763. Courtesy Yale Center for British Art, New Haven.

### 3 Overview of the Subsequent Twenty-One Encoded Messages in Part 1 of the Manuscript

What follows on the next six folio-size pages (*Lm*, I, fols. 3v°- 6v°) is Duvignau’s listing of 21 exchanges that were to put his prefatorial account to the test. The arrangement in five columns (Fig. 7) is retained and headed as “*Selam*” – “*Nomenclature* [nomenclature, list of object names in Turkish] – *Mané* [modern Turkish *mana*] ou *Signification* – *Interpretation a la Lettre* [the literal interpretation of the encoded meaning in French] – *Lettre Française premiere* [the first of the 21 letters in French].

Beginning of the Exchange of 21 Letters – Explanation of Columnar Arrangement				
Selam	Nomen- ture	Mané ou signification	Interpretation a la lettre	Lettre Française (premiere)
Selam	Nomen- clature	Mané or meaning	Literal Interpretation	(First) French letter
(List of Objects, object names French)	(List of In Turkish)			
<u>Selam</u>	<u>Mané ou signi-</u>	<u>Interpretation a la</u>	<u>Lettre Française premiere</u>	
	<u>cation –</u>	<u>Selam</u>		
maison	evim	High house	Maison de Selam	Ma maison rend de vous l'honneur que vous lui
gambel	gambel	High house	Maison de Selam	gambel de vous l'honneur que vous lui
flame	al	High house	Maison de Selam	flame de vous l'honneur que vous lui
Dray	ahle	High house	Maison de Selam	Dray de vous l'honneur que vous lui
Gambel	honor	High house	Maison de Selam	Gambel de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	ahle de vous l'honneur que vous lui
ahle	honor	High house	Maison de Selam	

Figure 7: *Lettres muettes*: Beginning of the Exchange of 21 “sample” letters.



Figure 8: The 9<sup>th</sup> Letter—a description of the lover’s sufferings.

The correspondence (and we shall assume here that a man addresses a lady) begins with a declaration of love: a raisin, ginger, something white, a piece of cloth, coal, white silk with alum, something yellow, and aloe are needed to encode a lengthy amatory piece that opens with the glowing admission: “My eyes—as you should know — I have hopelessly fallen in love with you,” the lover says, almost stuttering in this first sentence. He ends his mellifluous lines by asking for a “*billet*” in return that would encourage him to hope for the lady’s embraces. But the first response is a clear rejection—the lady, very much in rage (*très* [!] *en colere*) calls him a liar. He backpedals in his second letter and offers any “reparation” that might please her—he even offers her his life and will be her slave. Yet the lady still is not satisfied; her assembly of ten items—beginning with cabbage and ending with a sugar cane—encodes her determination when she calls him two-faced (*à deux visages*) and a fake from whom she does not want to hear any further protestations until he would give up his long-standing love affairs.

The exchange continues in this vein—he calls her tyrannical, repeats his “protestations” (5) and describes himself as a mere skeleton of himself. In vain (6)—the lady just considers all this frivolous since he has not offered any proof of his feelings. (7) Disappointed that the beloved does not yield and remains utterly cruel the gentleman—in a last-ditch effort, it seems—reminds her that she is still the sovereign of his soul while he has resigned himself to sacrifice her. And—what a surprise—the lady begins to believe in the sincerity of his promises and admits that she cannot defend herself any longer from his desires—indeed, the fire of his love is felt all the way to her heart. And thus, an eternal correspondence is in the offing.

The gentleman stammers in his response (9) (Fig. 8) and begs her to help him in his sufferings—this time a brass thread, hairs, sugar, a violet, a tiny broom, and a nut without its shell suffice to encode this shorter message. (10) And now the technicalities of a first meeting begin to be discussed: She cannot come to see him but welcomes him to her abode in order to offer him the rightful place in her heart. And she will allow him to do with her whatever his heart desires ... . But hold your horses, Duvignau implies—life just is not that easy (11): Unfortunately, the gentleman cannot find her lodging and humbly begs her to come and spend some time at his place, where he will be in an even better position to satisfy her. Out of pity (12)—and now totally infatuated—the lady suggests that she could come to his abode the following day after her stay at the public baths, the *hamam* (Fig. 9), virtually the only occasion for which Turkish women could leave their houses, as we have seen (the *hamams* being reserved for men after dark).



Figure 9: *Hamams* or Turkish Baths.

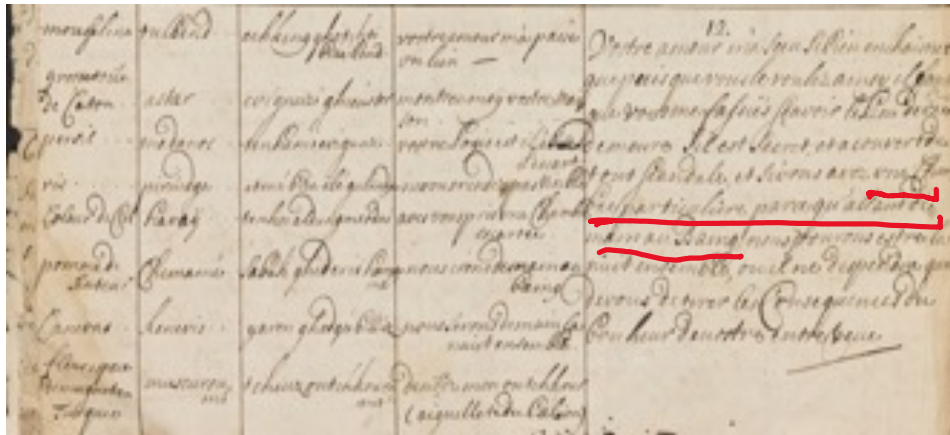


Figure 10: The lady's willingness to spend the night with her lover "at a secret place".

And—lo and behold—the lady is willing to spend the night with him (*nous pourons* [sic] *estre la nuit ensemble*) (Fig. 10)!

With such an encouragement the gentleman now assures his beloved (13) that he would take her to a secret place where all sorts of entertainment—games and dances—would be provided so that she could make him “the happiest of all men.” (The “*verd seladon*” in the *Selam*, a precious piece of celadon ceramics, is the code for an amusement with dancers). Duvignau closes this lengthy letter with a terse statement, “*Correspondence établie*,” an indication that the difficult exchanges at long last led to a physical union of the two lovers.

There follow effusive love letters on the gentleman's part (14): a first, concerned inquiry into the lady's health (15)—the easiest explanation for the lack of contact, which leads him to total martyrdom (*mon martire*) (16). He cannot find solace in anything else, he professes in his next piece (17), having been abandoned by the rest of the world with all his lovesickness, for which there is only one true cure (*la Veritable guairison de / mes Maux*).

Duvignau clearly provides templates for letters for all imaginable circumstances—the four preceding, pleading missives, he obviously imagines, could become handy tools. But the situation changes dramatically with the 18<sup>th</sup> letter: The lady finally responds, and her answer is both an admission of guilt and a list of accusations on her part. There are seven objects needed to encode this communication ranging from pistachios and other nuts to precious velvet and silk, and they convey an ambivalent message: In the letter—ominously titled “*De rupture*” (Fig. 11)—the lady furiously accuses her lover of having stalked her and

surprised her—with another woman (*que vous m'avez surprise*).



Figure 11: The downside of the relationship—Break-up and, finally, Offenses and Insults.

That he ridiculed her does not offend her as much as his own reaction, she cries out: He sought solace in the arms of another woman, the traitor, she retorts in closing, wishing him continued pleasure in this new relationship.

The author has created an intriguing situation—Balzac in his *Comédie humaine* could not have done better almost 200 years later. What is the gentleman to do in such a botched condition? His contrite response (19) is encoded in a singular fashion by means of a string (*sidgim*) that the author had earlier used as an example of the two-sidedness of associations with some of these objects: Here its use bodes ill and introduces an exclusive list of plant-related items, from nuts to vines to leaves of olive trees. And their encoded message is to convey utter contrition—he is not worth the dust on which his lady walks, he professes in Oriental humility (*la poussiere sur laquelle vous marchez*), all the while overlooking the lady's initial breach of trust. Yet the lady prefers not to respond; there is one final piece on his part, truly a last-ditch effort (20). This time there are only two items in the *Selam* to encode a message of contrition, namely a large piece of wool



cloth and a swatch of crude linen, which seemingly anticipate the dismal content of the letter: More self-accusations followed by his fear that he will not be heard.

Duvignau concludes this exchange—which had reached a dead end, it seems—with one last letter (21) to the lady that he titles, “*Derniere Lettre d’imprecations et iniures*” (Fig. 12).

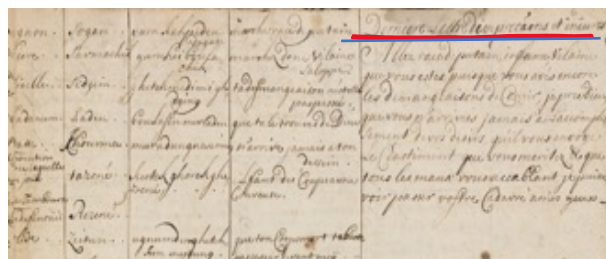


Figure 12: The last of 21 “sample” letters—a list of imprecations hurled at the lady...

By now we should be prepared for the items encoding these verbal assaults: From an onion to the ominous string to an olive we find familiar ingredients to such a dismal message, which indulges in abuses like “brood of whores” (*race de putain*) and culminates in the supreme insult (familiar by now) of wishing to see the corpse (*cadavre*) of his former beloved paraded before his eyes. Like a thunder clap Duvignau comes to a close of what for half of the exchange of letters seemed to be most promising—yet (and he later proved his mettle as an astute author) he did not necessarily believe in Hollywood-style happy endings, as we shall see in one of the print versions of this material.

#### 4 The Remainder of this Manuscript

This exchange of 21 messages is certainly the most intriguing section of the three-part manuscript. In the second half of the first section the set-up changes; Duvignau now lists the Turkish object first, followed by its French equivalent. Just like earlier we then have the Turkish association followed by a rather literal French translation but no more effusive French elaborations. These five folio pages (*Lm*, II, fols. 6r° - 8r°) can be used in

<sup>3</sup> Apparently there was a second manuscript edited by the author one year later (1680). Unfortunately—and I thank Mme Michèle Neveu, Bibliothèques municipales de Chartres for this information—it was kept there but was lost in a fire in 1944. It was titled, “*Lettres muettes ou la manière de faire l’amour en Turquie sans sçavoir lire ni écrire. Ouvrage reveu, augmenté par l’auteur [Du Vignau de Lissandre]. 1680, 68 pages, 16x22,3 cm, quarto size (Omond, 1890).* – The

the decoding of Turkish items contained in a *Selam*, but they are difficult to work with as they are not alphabetized. As if to add weight to the material prepared for Jacobus Colyer the author closes this first section with a number of affidavits (Fig. 13) given by men and three women from Constantinople who certify that the material here presented was indeed in common use and practiced by “the most delicate persons.”



Figure 13: Affidavits of three women on the last, signatory page of the *Lettres muettes* manuscript.

As convincing as these affidavits may appear—especially those of the women in the lower half of the page—the fact that their signatures appear in the same writing as the rest of this first section can either mean that Duvignau “created” these witnesses and their signatures as part of his fiction. It could, however, simply mean—and this would be the kindlier interpretation—that the entire section is a copy from a now-lost original.<sup>3</sup> Since we have no other writing samples of Duvignau’s this question remains unanswered. What also is highly doubtful—and this is a serious concern, of course—is the matter of practicality. While it may have been entirely acceptable to enter into all sorts of communication in this fashion “between the sexes” in order to exchange (non-) verbal declarations of love and more, so to speak, the actuality of a married lady spending the night at another gentleman’s house may have been highly improbable given the strict mores of 18<sup>th</sup>-century Turkish society. If caught, both partners would have faced the death sentence....

existence of a second manuscript is important as far as the authenticity of the Wolfenbüttel manuscript from 1679 is concerned as it would at least confirm the date. As we can see from the various signatures on the last page—all of them in the same hand—the Wolfenbüttel piece cannot be an original.

The second portion of the manuscript<sup>4</sup> (*Lm*, II, fols. 1r° - 11r°) (Fig. 14)—written in a different hand—follows the previous four-part arrangement and can again best be used for the decoding of the extended meaning of the Turkish items assembled in a *Selam*. Together with the third part it contains more explicit expressions (the very first line of Part II, “*nos pieds/jambes entrelassés*”—our feet/legs intertwined—points in this direction).

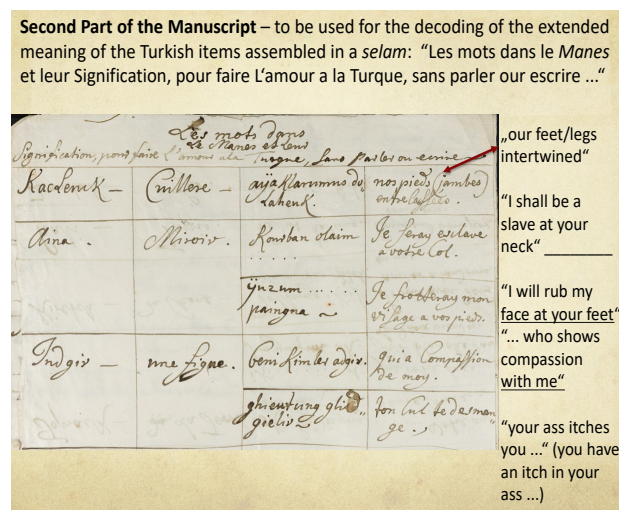


Figure 14: Beginning of Part II of the *Lettres muettes* manuscript.

The third and last part, written hurriedly in the same hand (*Lm*, III, fols. 3v° - 6v°) is a French-Turkish dictionary, bound in a tall, narrow notebook that clearly was intended for use “in the heat of the battle.” There are attempts at grouping the entries, and in three lines (fol. 4 r°) the French words are enciphered (Fig. 15)—which may have caused the Wolfenbüttel librarian who catalogued the manuscript a century ago to title it a “French-Turkish Love Cipher.” The Turkish terms (which the librarian certainly would not guess at) are vulgar sexual expressions for “penis”, “vulva” and “copulation” still in use today....

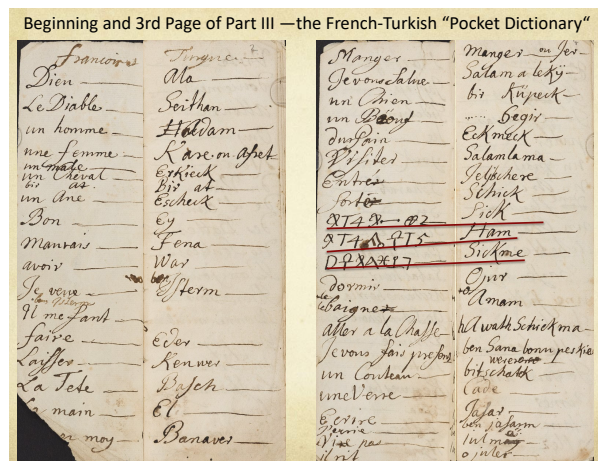


Figure 15: Beginning and third page of the French-Turkish “Pocket Dictionary” (with the three enciphered French words).

## 5 Confirmation of these *Selams* in Later Sources

It is reassuring to find several accounts in somewhat later (western) literary sources well before the 19<sup>th</sup>-century materials cited earlier as they prove the value of this manuscript for the cultural history of the Ottoman empire, for the lives of western diplomats at the Supreme Porte but also for the history of cryptology. Some twenty years after Duvignau’s account another Frenchman, Jean Dumont (1696) mentions in his writings “Monsieur Collier, the Dutch Ambassadeur, whose Reasons made the greater Impression upon [the Grand Visier]”, in other words, the same Jacobus Colyer to whom this manuscript was addressed, and who by 1694 had become the Dutch ambassador upon the death of his father. Dumont describes the method of encoded communication that I have just presented as if he had had access to this manuscript:

When [Turkish women] are in the Humour, and have chosen a promising Play-fellow, they send him a Declaration of Love by some old Confident. But wou’d you not be surpriz’d instead of a Billet-doux to find nothing but Bits of Charcoal, Scarlet Cloth, Saffron, Ashes, and such like Trash, wrapt up in a Piece of Paper. ‘Tis true these are as significant as the most passionate Words; but ‘tis a Mystical Language that cannot be understood without a Turkish Interpreter (Dumont, 1696).

<sup>4</sup> In Cod. Guelf. 389 Nov. 2° this section is listed under (c).



In the French original Dumont more candidly said of this exchange of messages by means of encoded objects, “*mais il faut être Turc pour l’entendre*” (Dumont, 1694) (but you have to be a Turk to understand it—which implied that he himself did not grasp it).

The most extensive—and informative—report, however, occurred in fictional letters written by Lady Mary Wortley Montagu (1689-1762), the wife of the English ambassador to the Sublime Porte (Fig. 16). Today she may be best known for introducing the smallpox inoculation in England seventy years before Edward Jenner developed the safer vaccination. In 1719, upon her return to London, she wrote down her experiences in Turkey in epistolary form. In her “Turkish Embassy Letters” she specifically referred to the custom of “Turkish Love-letters.”



Figure 16: Lady Mary Wortley Montagu and title page of her collection of *Letters*.

In this system, she reports,

there being (I beleive) [sic] a million of verses design’d for this use. There is no colour, no flower, no weed, no fruit, herb, pebble, or feather that has not a verse belonging to it; and you may quarrel, reproach, or send Letters of passion, freindship [sic], or Civillity, or even of news, without ever inking your fingers.

While Lady Mary’s observations date back to 1719, they were only printed in 1763. Much earlier appeared related comments by Aubry de La Mottraye (1727), who had seen “bloody gallantries” by young Turkish men who slit their arms as a token of admiration for their beloved (who witnessed such testimonies from behind a barred window) (Fig. 17). But there seems to be a

much more gentle way of expressing such affection, La Mottraye explains, “*de se faire l’amour, sans se parler ni se voir*”—an almost literal allusion to the title of the *Lettres muettes* manuscript. Last not least—he observes—even the “Odaliques” in the Sultan’s Harem were well versed in various arts of courtly entertainment but could not read or write, which brings him to the conclusion that early on young Turkish women in general learned the art of non-verbal communication as he described it (a remark relevant to the use of such *Selams* in the *Histoire Galante*).



Figure 17: “Turkish Gallantries”—men slitting their arms in front of their beloved as a token of their affection.

## 6 The Practical Application of such Non-Verbal Communication in the Two Totally Different Versions of the *Histoire Galante* of 1688

As has been briefly mentioned the system of *Selam* exchanges, of the sending of such non-verbal messages, is reflected in two different publications that appeared in 1688. A small book authored “Par le Sieur D. L. C.” came out in Holland in 1688; the acronym has been associated with “Duvignau de Lissandre, Chevalier” since Edouard de la Croix did not want to be identified with these imprints. In part its title is almost identical with that of the Wolfenbüttel manuscript: *Le Language* [sic] *müet ou l’Art de faire l’Amour sans parler, sans écrire & sans se voir* (Duvignau, 1688): Here, however, the transmission mode—if I may put it that way—is expanded by stating that making love would not only be possible without talking or writing to the beloved but also without seeing the object of one’s desire.



As it turns out this 100-odd-page booklet in its first part provides a detailed description of what we have called the “language of symbols” as seen through the critical eyes of a foreign observer, material that is very similar to the introductory section of the manuscript: While men in many nations are free to express their feelings in a conventional manner to the women whom they admire, the author posits that Turkish men—who for the most part do not know how to read or write—are nonetheless not “insensible.” To the very contrary, he affirms, they express their passion in totally unconventional ways (*Lm* 1688, fols. § 4 r° and v°) and even slit their arms, just as we have seen in the illustrations taken from early 18<sup>th</sup>-century publications. This first section then begins to describe the “*Amour Müet*”—literally silent love(making)—as illustrated in “*une Histoire Galante et véritable*”, a courteous and truthful story, the author assures us. In order to enable his reader to understand the numerous *Selams* needed for this kind of communication he inserts a “*Dictionnaire [sic] Alphabétique du Language Müet contenant / Le nom, la signification, la valeur & l'Interprétation [sic] des Selams*” (Fig. 18).

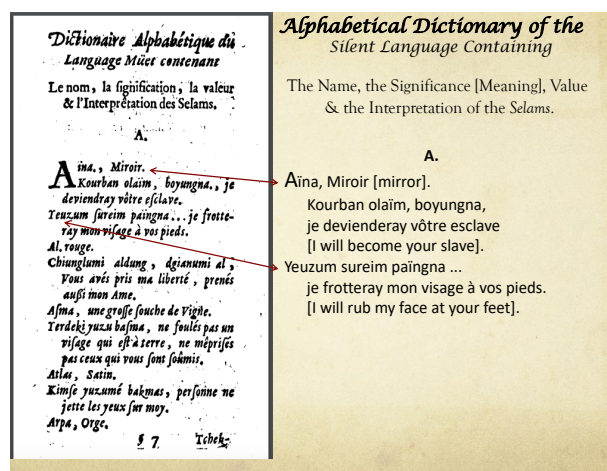


Figure 18: Beginning of the “Alphabetical Dictionary of the Silent Language” in the 1688 imprint titled, *Le Language müet*.

On the next 24 pages we have an alphabetized listing of exactly the names of all the objects that the manuscript contains, beginning with “*aïna*”, its French translation (mirror), the Turkish metaphorical “value” of this object, and again its French equivalent. The first example is particularly

interesting as it not only offers a second Turkish meaning of the same object (*Yeuzum sureim paingna ...*) whose signification (“I will rub my face at your feet”) is compatible with that of the first, “I will become your slave,” it also exemplifies the rarer riming scheme at the end of the entire Turkish expression, which enhances its mnemonic value—*aïna* rimes with *boyungna* and *paingna*. (It is quite obvious that any memory aids such as the riming expressions for the “code words”—to employ this cryptological term—are essential since the users of this system cannot rely on written code lists but have to depend on their mnemonic retention).

After this elaborate dictionary listing the author finally begins an intriguing novelette with the promising title, *Histoire Galante (HG)*. It turns out to be the ideal vehicle for a goodly number of *Selams* which are introduced after the two young protagonists, Issouf and Gulbeas (“White Rose”), both growing up in the same close-knit quarter of Istanbul where Issouf (the son of a wealthy man with his own “*Palais*”) is sitting in on lessons in reading, writing, and musical entertainment given to Gulbeas (the servant of a neighbor) by an old Jewish scholar. The two young people are enjoying each other’s rather restricted company when suddenly Gulbeas is given by her master to the *Sultana Validé*, the mother of the reigning Sultan.

A perfectly normal story of fledgling love, told by Gulbeas and at the end by her intimate friend, Patma, so far is nowhere suggesting the need for encoded communication through *Selam*-messages. Yet with Gulbeas’s sequestered life in the Sultan’s *Harem* the novelette suddenly takes a dramatic turn: While the “White Rose” is preoccupied with her new environment Issouf becomes increasingly desperate and begins to look for ways that could reconnect him to his beloved. At this point a Jewish woman—one of the many who were catering to the needs of the ladies of the *Harem* and bringing rare fruit, toiletries and the like to the hundreds of females inside—offers her services to Issouf. These Jewish women—and research has corroborated this important element in the story<sup>5</sup>—pass through the gates of the *Seraglio* without being checked by the eunuchs, the ruthless gate keepers (Fig. 19). Boullaster, nicely paid for such services, suggests that she could bring *Selam*-messages to Gulbeas. She manages to introduce

<sup>5</sup> See, for instance, the detailed description of these “women-servants of the harem; [...] some of these female servitors lived outside the Imperial Palace and could easily meet foreigners, acting as their contact with the world

outside the harem; they were usually called *kiras*, from the Greek word meaning ‘lady’” (Pedani, 2000).

herself to Gulbeas, shows her precious jewelry which she carried in a box that—lo and behold—also contains a “*billet doux*”, a love letter that Boullaster (knowing that Gulbeas could read it) had written on behalf of Issouf. Hidden deeper in the box Gulbeas also discovers a *Selam* (HG, 13), but assuming right away that the note might be a declaration of love curious Gulbeas proceeds to read it first. Its text, embellished with effusive oriental emotions and covering more than one page of the booklet (HG, 11-12), speaks of overboarding feelings that Issouf had harbored for several years when he was listening to Gulbeas in a corner of the garden next to hers as she sang and played her instruments, an occasion he used to sometimes talk to her.

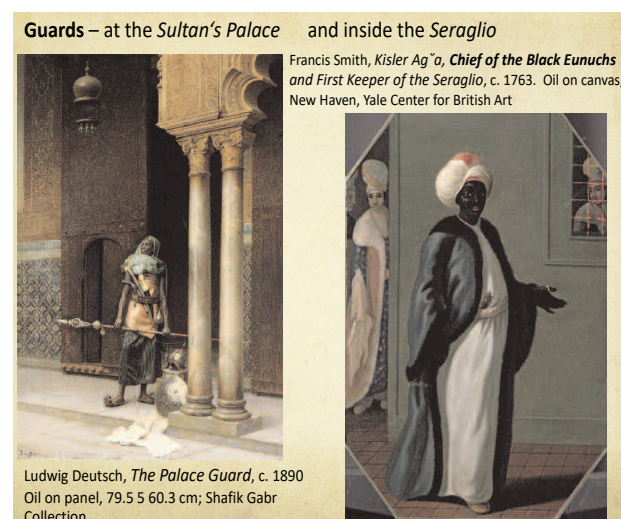


Figure 19: Guards at the Sultan's Palace and the Seraglio.

To recreate such moments Issouf proposes to have Boullaster manage his good fortune while at the same time protect Gulbeas' reputation. With heightened emotions or curiosity Gulbeas then proceeds to “develop the *Selam*” and carefully unwrap it. In this first of five Selam-messages we not only see the French text along with the various items needed to build this *Selam* but also the equivalencies to these items as listed in the preceding *Dictio(n)naire Alphabétique*. It is intriguing to read how the author has worked these five expressions into the embellished prose text whose “interpretation” begins with the translation of the Turkish metaphorical expression for “raisin”, namely “(two) eyes” (Fig. 20).

While pretending not to be satisfied with this “déclaration” that she considers somewhat too explicit Gulbeas listens to her heart that praises Issouf's qualities. And although Boullaster would

just as soon have wanted to introduce him to her apartment Gulbeas—after numerous entreaties—agrees to at least see him in the gardens below from a latticed window. And contrary to what “honor and reason” would have dictated she opts to prepare a *Selam* that is to convey to Issouf that “his passion did not displease” her. Gulbeas does not tell us which objects the *Selam* assembled, but she stresses that they were wrapped in a silken kerchief that she herself had embroidered with gold threads.

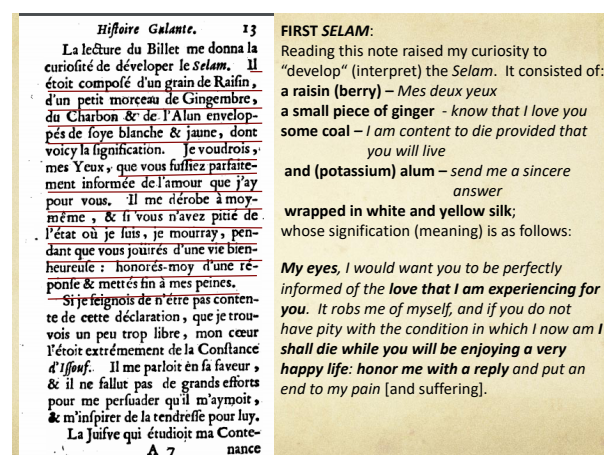


Figure 20: The first *Selam*-message.

Issouf is elated when Boullaster informs him of Gulbeas' feelings for him, drenches the silken kerchief of the *Selam* with his tears of joy and finally opens it: “(precious) silk ‘Isabelle’, a strand of jasmin, a small piece of sponge, mint and myrthe” (HG,16-17). Issouf is overjoyed and eagerly explains it, prodded by Boullaster, who (it seems) does not manage to interpret the *Selam*: “I accept your vows & (please) be convinced of my truthfulness, provided that you yourself are faithful I shall pray to Heaven that He will give you to me, & that our souls be inseparable”.

This now opens an even more problematic chapter in the relationship: How is Issouf to enter the forbidden *Seraglio*? Tormented by these thoughts he finally remembers that Mehemet, a gardener who is indebted to his father, might help him get access to the beautiful terrace below the *Sultana Validé*'s apartments (HG, 18-21). Issouf informs Gulbeas of this ploy in a third, “small” *Selam* (HG, 21), and after carefully assessing the dangers involved in such a “visit” she agrees to wait for him behind a latticed window in a room adjacent to the garden. Appropriately camouflaged as a lowly gardener Issouf appears, and after a long wait the two finally manage to communicate—but (and here to novelette takes another unexpected turn) in view of the proximity to the *Validé*'s



apartments in the *Langage müet* (Fig. 21), which both of them master.

## Excursus: The “Silent Language”

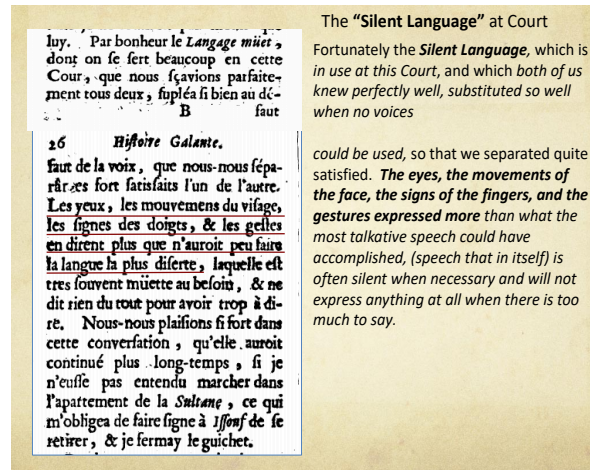


Figure 21: The concise description of the “Silent Language” in use at Court—and between the lovers.

This is the only major departure from the Wolfenbüttel manuscript, for otherwise the *Histoire Galante* uses exactly the material that was prepared for Jacobus Colyer. The excursus in the narrative, while unexpected, is perfectly plausible: This sign language, as it can be called (certainly in one way or another anticipating modern-day sign language used in communication with the hearing impaired), was used and taught at the Sultan’s court where the protocol demanded perfect silence—which means that high-level courtiers, eunuchs, and the Sultan’s favorite dwarfs had to use a non-vocal way of communicating (Fig. 22). Apart from the cryptographic aspect inherent in *Selam* exchanges this *Langage müet* is the second, highly relevant cryptographic example in the novelette.<sup>6</sup> That Gulbeas and, in particular, Issouf would master this complicated sign language is yet another miraculous detail in our romantic novelette (it is well documented that the *Langage müet* was taught by eunuchs inside the *Seraglio*, and we can only speculate how Issouf might have learned it)—yet its use fulfills the very same purpose that the 13<sup>th</sup> letter of the sample exchange in the manuscript summarizes in the taut statement, namely, “Communication established.”

<sup>6</sup> In the second—and initially parallel—narrative that will be discussed on the following pages (see below, pp. 14-15) this communication method is called “le langage par

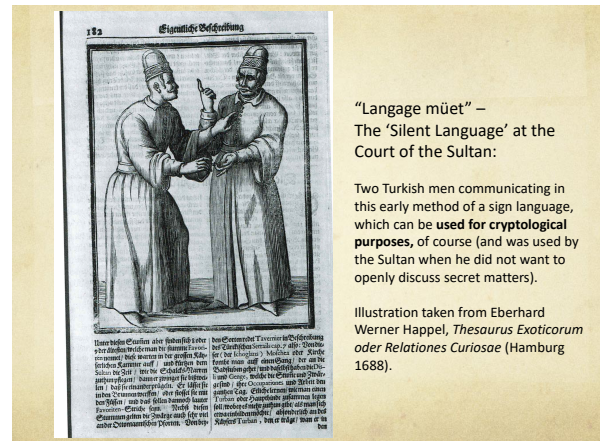


Figure 22: Two eunuchs communicating in the *Langage müet*.

Encouraged by this first, silent meeting, the two lovers contemplate a second get-together—this time, however, inside the *Harem*. Once more well-bribed Boullaster conveys a *Selam* to Gulbeas that lovelorn Issouf has prepared, which—together with Boullaster’s entreaties—results in Gulbeas’ putting together the fifth and last such message in the novelette. And despite the “slippery slope” that “White Rose” was about to take—as Patma, who is narrating this last part of the story, calls the undertaking—Boullaster’s suggestion deemed feasible: Issouf was to enter the *Harem*’s premises disguised as a young girl (fortunately, Patma adds, Issouf did not yet have a beard); the young man could thus pass as Boullaster’s daughter. Little did he know, Patma continues, that he was to meet his own death in this rendez-vous as his beloved Gulbeas had contracted the plague (*HG*, 31-32).

We have reached the moment when the narrative develops in two totally opposite directions. In the *Histoire Galante* Issouf enters Gulbeas’s bedroom only to find her stricken by the deadly disease (*HG*, 34-35)—I shall spare you his heart-rending testimony of love where he suggests that he would gladly die if his beloved were spared. And this is exactly what happens: Upon the difficult return to his own “*Palais*” (*HG*, 37) (guards at the exit of the *Harem* had stopped Boullaster and Issouf when they noticed Issouf’s gait that was by far too clumsy for a young girl) he immediately took to his bed, sent Gulbeas their engagement ring along with a last, heartbreaking note taken down by Boullaster—and died of the disease after three days. In return poor “White Rose,” who had

signes,” a better and more descriptive definition that anticipates modern sign languages.

actually recovered from the plague after their fatal encounter, became increasingly so depressed after having received Issouf's last tokens of love that she pined away and—as Patma reports on the last pages of the *Histoire Galante* (43-44)—showed no signs of ever regaining her health.

In the same year (1688) he published the Colyer manuscript material anew in a totally different, highly informative book. Once again its title—*Le Secretaire Turc, contenant l'art d'exprimer ses pensées sans se voir, sans se parler & sans s'écrire* [...] (Fig. 23)—re-uses part of the Wolfenbüttel manuscript title, but the 340-page quarto-size publication devotes almost half to a detailed description of the life at the Sultan's *Seraglio*. In a long introduction (ST, 1-36) the author explains the *Selam* communication method spelled out in the title and sees its roots in Egyptian hieroglyphs (ST, 10-11) that, he feels, were also precursors of the written word. One charming detail not reported so far is that Turkish *Selam* users often have what one might call a “toolbox” where they keep the most important objects required for their messages. And contrary to the *Langage muet* with the ancillary materials preceding the *Histoire Galante* a “Catalogue” of 179 objects (ST, 158-211) needed to send a *Selam* now follows the “*Histoire de Youssuf-Bey et de Gul-Beyaz*.” Duvignau introduces the piece as “l'*Histoire de la vieille Juifve*” (The Story of the Old Jewess) for her rôle in this narrative that is even more important, as we shall see.

For over one hundred pages (ST, 37-147) the two familiar protagonists, Issouf and Gulbeyaz, go through very much the same painful love relationship. A closer comparison of the two versions would show that in the *Secretaire* Duvignau at times uses textual material verbatim, introduces the same episode where the “*langage par signes*” is the only possible communication method (ST, 109-110), presents some of the same *Selams* but has Boullaster take an even more active rôle as a go-between and organizer. Issouf himself is presented as a very wealthy and well-connected young man who—and here the two versions begin to differ—proposes to make every effort possible to withdraw Gulbeas from the Harem and marry her—as Fatma (formerly Patma, here, however, the narrator throughout) confirms “after both of them had been exposed to the most dangerous proofs of their love” (ST, 112), she wistfully adds.

<sup>7</sup> The extensive title is most descriptive: [...] avec les circonstances d'une Avanture Turque, & une Relation très-

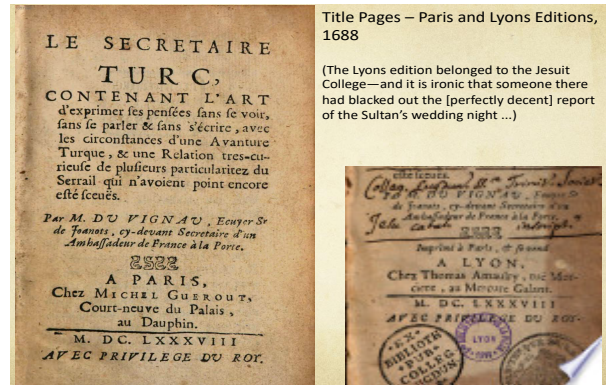


Figure 23: Title pages of the Paris and Lyons editions of *Le Secretaire Turc*.

These dangers—elaborated on on somewhat familiar pages (ST, 120-141)—are once more Gulbeas' plague contamination, Issouf's infection during their fateful rendez-vous at her bedside—but finally his miraculous cure and Gulbeas' similar recovery. Issouf's connections to high nobility—and here Duvignau astutely prepares the ground for his discussion of the Sultan's court in the second part of *Le Secretaire Turc*—will indeed extricate Gulbeas from the Harem and result in an elaborate wedding. Contrary to the *Histoire Galante* with its rather “ungallant”, fatal ending this second version of the manuscript material presents a Hollywood-style “happy ending” that clearly serves one purpose: Duvignau wants to raise the curiosity of his readers to delve further into the latter half of the book, where the author continues his insightful look at Turkish nobility as witnessed in the wedding, and where he presents intimate details of hitherto unknown goings-on in the Sultan's *Seraglio* (SF, 212-340).

## 7 Closing Arguments

This happy ending to materials based on a unique manuscript from the Herzog August Bibliothek may serve as an appropriate way to close a discussion of historical materials that hopefully has offered a glance at two rare 17<sup>th</sup>-century means of communication. While the exchange of *Selam* messages has allowed some insight into this earliest piece of information in the west on a different kind of cryptology, namely an encoding system of numerous objects, the two novelettes also introduced to western readers another and perhaps even more unexpected method used in

curieuse de plusieurs particularitez du Serrail qui n'avoient point encore esté sceuës.

non-verbal, secret exchanges: The *langage müet* mandated at the Sultan's court and practiced by the two lovers in a somewhat unconventional fashion grants at least a glimpse at one more fascinating piece of Turkish and Oriental cultural history, an early sign language that was an important element of secret communication in ruling circles.

## References

- Carla Coco. 2002. *Harem: Sinnbild orientalischer Erotik*. Transl. from the Italian by Claudia Podehl-Fenu. Munich: Orbis, especially "Die Frau im Spiegel der Osmanen", pages 10-21, and "Ein abgeschlossener und exklusiver Raum", pages 22-65.
- Marloes Cornelissen. 2015. *The World of Ambassador Jacobus Colyer: Material Culture of the Dutch 'Nation' in Istanbul During the First Half of the 18<sup>th</sup> Century*. Ph.D. Thesis. Istanbul: Sabancı University, Institute of Social Sciences.
- Edouard de La Croix. 1684. *Mémoires du sieur de La Croix, cy-devant secrétaire de l'Ambassade de Constantinople contenant diverses relations tres-curieuses de l'Empire Othoman*. 2 vols. Paris: Cellier.
- Jean Dumont. 1694. *Nouveau Voyage du Levant* [...]. The Hague: E. Foulque, p. 319.
- Jean Dumont. 1696. *A New Voyage to the Levant*: [...]. 2<sup>nd</sup> ed. London: Gillyflower *et al.*, p. 323.
- Duvignau. 1679. "Lettres muettes, ou la maniere de faire l'amour en Turquie / Sans Scavoir n'y Lire n'y Ecrire." Manuscript, Herzog August Bibliothek, Wolfenbüttel, Germany. Cod. Guelf. 389 Nov. 2°. Part (a) contains various ciphers and nomenclators; (b) is the manuscript in question, the "Lettres muettes". It consists of 3 parts: Pt. I, 18 pages; Pt. II, 14 pages; Pt. III (separately listed as (c), 20 pages. Part I is a careful copy, the other two are hastily penned down originals. See Strasser (1988), pages 511-514.
- Duvignau de Lissandre. 1680. "Lettres muettes ou la manière de faire l'amour en Turquie sans sçavoir lire ni écrire. Ouvrage reveu, augmenté par l'auteur." [Du Vignau de Lissandre]. Manuscript, 68 pages, 16x22,3 cm, quarto size. Listed in Henri Omont *et al.*, eds. *Catalogue général des manuscrits des bibliothèques publiques de France. Départements. Tome XI, Chartres*. Paris: Plon, Ms. 473.—Lost in a fire in 1944.
- Duvignau de Lissandre, Sieur des Joanots (also: Du Vignau; also: Le Sieur D. L. C. [= Duvignau de Lissandre, Chevalier]). 1687. *L'Etat présent de la Puissance Ottomane, Avec les causes de son Accroissement, & celles de sa Décadence* [...]. Paris: Horthemels; The Hague: de Hondt and van Ellinkuysen, 1688.
- . 1688a. *Le Secretaire Turc, contenant l'art d'exprimer ses pensées sans se voir, sans se parler & sans s'écrire* [...]. Lyons: Amaury; Paris: Guérout. Available at URL: <https://books.google.de/books?id=G5HrxQnGhd4C&printsec=frontcover&hl=de#v=onepage&q&f=false> (accessed 02/26/2020).
- . 1688b. *The Turkish Secretary containing the art of expressing ones [sic] thoughts, without seeing, speaking, or writing to one another: with the circumstances of a Turkish adventure* [...]. Transl. John Phillips. London: Hindmarsh and Taylor.
- . [Le Sieur D. L. C. (= Duvignau de Lissandre, Chevalier)]. 1688c. *Le Language [sic] müet ou l'Art de faire l'Amour sans parler, sans écrire & sans se voir*. Middelbourg: Horthemels; Amsterdam [1690?]. The "Histoire de Youssuf-Bey et de Gul-Beyaz" is featured as the opening segment in *Le Language müet*, pages 1-44. Available at URL: [https://books.google.de/books?id=g-tmAAAAcAAJ&printsec=frontcover&hl=de&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.de/books?id=g-tmAAAAcAAJ&printsec=frontcover&hl=de&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false) (accessed 02/26/2020).
- Jack Goody. 1993. *The Language of Flowers*. Cambridge: Cambridge Univ. Press.
- Joseph von Hammer[-Purgstall]. 1834-1836. *Geschichte des Osmanischen Reiches*. 2nd, rev. ed. in 4 vols. Pesth: Hartleben.
- Martina Heilmeyer. 2006. *The Language of Flowers. Symbols and Myths*. Transl. from the German by Stephen Telfer. Rev. ed. Munich *et al.*: Prestel, especially "Introduction: A look at the centuries-old relationship between flowers and their admirers," pages 7-18.
- Josef Hutter. 1851. *Von Orsova bis Kiutahia*. Brunswick, pages 181-185, quoted in Kakuk, "Blumensprache," pages 286-287.
- Suzanne Kakuk. 1970. Über die türkische Blumensprache. In *Acta Orientalia Academiae Scientiarum Hungariae*, volume 23, pages 285-295.
- Zsuzsa Kakuk and Cemil Öztürk. 1986. Lisân-ı Ezhâr. In *Acta Orientalia Academiae Scientiarum Hungariae*, volume 40, pages 3-37.
- Aubry de La Mottraye. 1727. *Voyages [...] en Europe, Asie & Afrique: Ou l'on trouve une grande variété de recherches [...] sur l'Italie, la Grèce, la Turquie* [...]. 2 vols. The Hague: Johnson. Description of the *Selam* communication in I, 290-291; list of objects needed for



the exchange of a message on pages 291-293. Illustration before page 275.

Adam Olearius. 1671. *Außführliche Beschreibung der Kundbaren Reyse nach Muscow und Persien* [...]. Schleswig: Holwein, Book 5, Chapters 21 and 22, pages 602-610; quoted by Walther: *Die Frau im Islam*, pages 65-66.

Maria Pia Pedani. 2000. Safiye's Household and Venetian Diplomacy. In *Turcica*, volume 32, pages 9-32, here pages 11-12.

Leslie P. Peirce. 1993. *The Imperial Harem. Women and Sovereignty in the Ottoman Empire*. New York, Oxford: Oxford Univ. Press, especially pages 113-118, 314-316, 336, 350.

Mary Roberts. 2007. *Intimate Outsiders: The Harem in Ottoman and Orientalist Art and Travel Literature*. Objects/Histories: Critical Perspectives on Art, Material Culture, and Representations. Durham, N.C., London: Duke University Press.

Gerhard F. Strasser. 1988. 'Lettres muettes, ou La maniere de faire L'amour en Turquie Sans Scavoir ny Lire ny Ecrire': Manuskript und Druck einer türkisch-französischen 'Liebes-Chiffre' an der Pforte. In *Chloe. Beihefte zum Daphnis*, volume 10, pages 505-524, here pages 511 ff.

----- . 2016. 'Lettres muettes'. Zu den frühesten Zeugnissen über die orientalische Blumensprache, Selam genannt. In *Floriographie. Die Sprachen der Blumen*. Ed. by Isabel Kranz, Alexander Schwan and

Eike Wittrock. Paderborn: Fink, pages 37-61, especially pages 50-55.

Corinne Thépaut-Cabasset, ed.. 2007. *Le Sérail des empereurs turcs. Relation manuscrite du sieur de La Croix à la fin du règne du sultan Mehmet IV*. Éditions du Comité des travaux historiques et scientifiques, Format 63. Paris: Éditions du Comité des travaux historiques et scientifiques. Biographical information on Edouard de La Croix on pages 16-24.

Pietro della Valle. 1674. *Eines vornehmen Römischen Patritii, Reiß-Beschreibung in unterschiedliche Theile der Welt*. 4 vols. Geneva: Widerhold, I, 19, quoted by Walther: *Die Frau im Islam*, pp. 65-66.

Wiebke Walther. 1997. *Die Frau im Islam*. 3rd, rev. ed. Leipzig: Edition Leipzig, especially pages 65-69.

Lady Mary Wortley Montagu. 1665-1680. *The Complete Letters of Lady Mary Wortley Montagu*. Ed. by Robert Halsband. 3 vols. Revised reprint of vol. 1, Oxford: Oxford Univ. Press, 1980. Introduction pages xiv-xvii; relevant letter pages 387-391, "To Lady – 16 March [1718]."

# HCPortal Overview

**Eugen Antal**

Slovak University of  
Technology in Bratislava  
Slovakia  
eugen.antal@stuba.sk

**Pavol Zajac**

Slovak University of  
Technology in Bratislava  
Slovakia  
pavol.zajac@stuba.sk

## Abstract

HCPortal is a portal consisting of several web pages and tools focusing on historical cryptology. The heart of the project is a comprehensive database of cryptograms accessible for everybody. The front-end of this portal was designed to provide a responsive and modern UI/UX. We used technologies built for the modern web. The major part of the portal's back-end is also available as a public API.

## 1 Introduction

The **Portal of Historical Ciphers** (HCPortal) is a gateway to the world of historical ciphers. You can find a comprehensive database of cryptograms, framework for document analysis, glossary and many more.

This project was created by researchers and students from the Slovak University of Technology in Bratislava in cooperation with other crypto history enthusiasts.

## 2 The Portal

The HCPortal consists of several parts. The portal's home page serves as an entry point, connecting these parts together. While the portal has started only recently, we have already prepared:

- **Home page** - entry point of the portal with navigation and information centre.
- **Database of cryptograms** - database with a public API, also contains visualization (front-end) and advanced search.
- **ManuLab** and **ManuLab online** - software product for statistical analysis, with a public API and example web page.

- **Tools and web pages** - links to external projects.
- **Glossary** - glossary for historical cryptology.

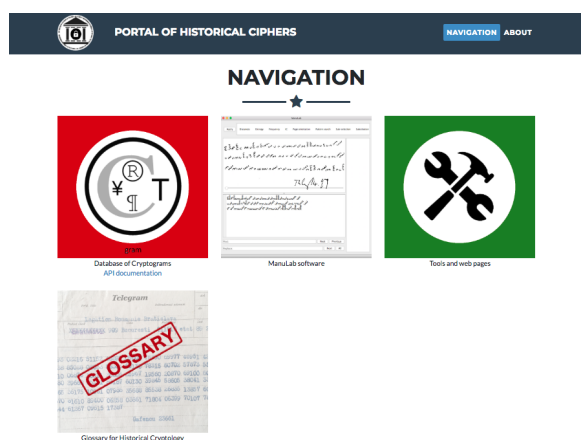


Figure 1: The main navigation screen.

The portal's entry point is available at:  
<https://hcportal.eu/>.

## 3 Database of Cryptograms

We are carefully collecting<sup>1</sup> the most important information about known cryptograms, which are stored in a relational database. Cryptogram descriptions are also available through a web-service (public API). The front-end (web) contains cipher detail visualization and full-text search. We have also implemented an advanced search, where it is possible to find cryptograms based on location, language, sender and other parameters.

<sup>1</sup>The cryptograms are collected (and are planned) mainly from (Klausis Krypto Kolumne, 2019), (Crypto Cellar Research, 2019), (Breaking German Navy Ciphers, 2019), The Slovak National Archive and The Military History Archive of Slovakia, all with permissions.

The API documentation is available at:  
<https://www.cryptograms.hcportal.eu/api/apidoc/index.html>  
 and accessible from:  
<https://cryptograms.hcportal.eu>.

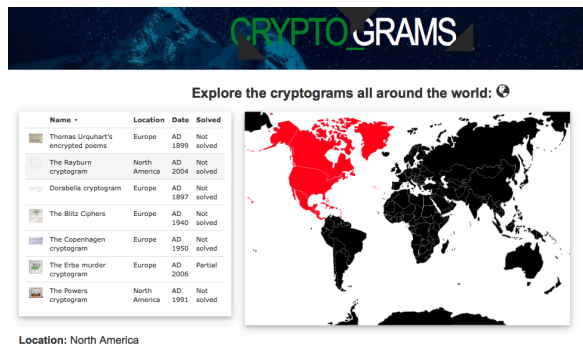


Figure 2: Cryptograms - home screen.

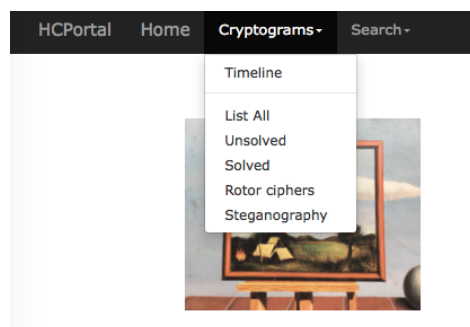


Figure 3: Cryptograms - main menu.



Figure 4: Cryptograms - cryptogram detail.

## 4 ManuLab

**ManuLab** is a software product for statistical analysis of encrypted historical manuscripts. The document analysis is performed via a chain of *filters* (main building elements). A filter represents

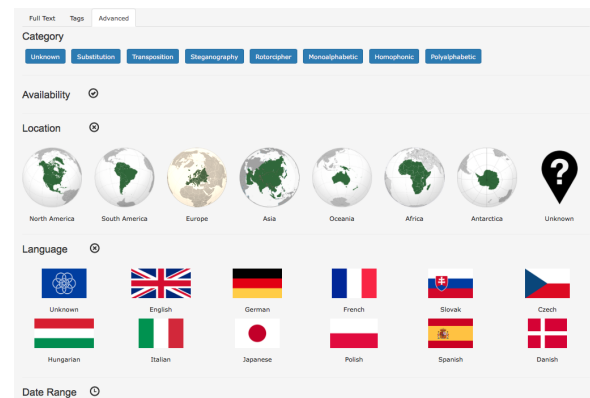


Figure 5: Cryptograms - advanced search options.

any operation realizable on a document transcription divided into a set of pages.

The implemented filters allow to change the reading direction, select sub-pages, or a subsection from the document, and calculate several statistics like the index of coincidence, Shannon's entropy,  $n$ -gram frequency, etc.

Later on, we have decided to create a more general framework independent from the operating systems, and ported the main functionality of the existing application to the web. **ManuLab online** is the online version of the ManuLab application, accessible via PHP scripts.

Furthermore we integrated the existing database of cryptograms directly to the example web page. The users can directly download and analyse text attachment of any cryptogram from the database.

The functionality was extended with cryptanalysis functions like language guess, anagram detection or Sukhotin's vowel detection method.

The source code is available online at the following GIT repository:

<https://bitbucket.org/jugin/manulab.git>.

The API documentation is available at: <https://manulab.hcportal.eu/apidoc> and an example web page (demonstrating the API) is available at: <https://manulab.hcportal.eu/example>.



**Manulab online example**

Statistics
Text operations
Cryptanalysis

Load data from the [HCPortal](#) database

Load text files
Text files

Text
RLS CMW DJP RFP J7O CEP JJN PRG ZTS MCJ  
JEH BLM CRR PLC JCM MEP JNH JDM RBS J7H  
BJP PJP SCB TLC KES REP RCP DTH I7H CRB  
JSB SDG

Guess the used language
Details Off
Guess

Result
The most probable language is: Turkish, with difference of IC: 0.0005.

Figure 6: Manulab online API example.

**Manulab online example**

Statistics
Text operations
Cryptanalysis

Load data from the [HCPortal](#) database

Load text files
Text files

Page 1
5. 3. 27. 38. 32. 14. 21. 8. 66. 8. 70. 39. 5. 9. 12. 18. 2. 3. 56.  
5. 1. 7. 3. 2. 13. 19. 3. 25. 9. 3. 16. 6.  
25. 15. 13. 6. 11. 20. 5. 1. 2. 12. 1. 20. 20. 49. 20. 35. 33.  
4. 6. 8. 35. 5. 33. 5. 5. 18. 10. 3. 11. 32. 42.

X Page 2

X Page 3
25. 11. 39. 4. 4. 10. 3. 54. 50. 19. 1. 18. 1. 5. 9. 58. 15. 1. 4. 17. 1. 42. 32. 77.  
23. 75. 6. 3. 18. 20. 36. 8. 21.  
4. 10. 22. 3. 5. 11. 3. 162. 18. 21. 44. 79. 42. 2. 17. 61. 32. 7. 7. 107. 8. 59. 28.  
54. 31. 113. 42.

Add new page
Clear pages

Frequency analysis
N: 1
Delimiter
Freq. type Relative
Show in table On
Calculate

Figure 7: Manulab online API example - multiple input pages.

## 5 Glossary

This site contains definitions of terms related to historical cryptology, including terminology for codes and nomenclators. Terms related to modern cryptology are not covered. The used terms are mainly from the declassified Friedman's collection - Basic Cryptologic Glossary (REF ID:A64719) and from (Klausis Krypto Kolumne, 2019). We are currently collecting visual examples (pictures) of selected terms to extend this glossary.

GLOSSARY FOR HISTORICAL CRYPTOLOGY	
This site contains definition of terms related to historical cryptology, including terminology for codes and nomenclators. Terms related to modern cryptology are not covered.	
<p>“ Certain terms, through long usage, have become more or less standard and generally acceptable while other terms hold different meanings in different areas. The lack of standardization has resulted, at times, in confusion and misunderstanding.</p> <p>— Ralph J. Canine in Basic Cryptologic Glossary (REF ID:A64719)</p>	
TERM	DESCRIPTION
ADFGVX system	A German high-command cipher system used in World War I. Essentially, a bilateral substitution system employing a 6 x 6 square, to which a columnar transposition was subsequently applied.
Alternate horizontal route transposition	Row transposition in which the route followed is alternately from left to right and from right to left in successive rows. <i>Other terms with the same meaning: boustrophedon.</i>
Alternate vertical route transposition	Columnar transposition in which the route followed is alternately up and down in successive columns. <i>Other terms with the same meaning: boustrophedon.</i>

Figure 8: Glossary.

## Acknowledgments

This work was partially supported by grants VEGA 1/0159/17 and VEGA 2/0072/20.

## References

- Klaus Schmeh. *Klausis Krypto Kolumne* <http://scienceblogs.de/klausis-krypto-kolumne>
- Frode Weierud. *Crypto Cellar Research* <http://cryptocellar.org/>
- Michael Hrenberg. *Breaking German Navy Ciphers* <https://enigma.horenberg.com/>
- Satoshi Tomokiyo. *Cryptiana* <http://cryptiana.web.fc2.com/code/crypto.htm>

# Diplomatic Ciphers Used by Slovak Attaché During the WW2

**Eugen Antal**  
Slovak University of  
Technology in Bratislava  
Slovakia  
eugen.antal@stuba.sk

**Pavol Zajac**  
Slovak University of  
Technology in Bratislava  
Slovakia  
pavol.zajac@stuba.sk

**Otokar Grošek**  
Slovak University of  
Technology in Bratislava  
Slovakia  
otokar.grosek@stuba.sk

## Abstract

Slovakia was an allied (puppet) state of Germany during WW2. In various Slovak and Czech archives, we found previously unknown details about diplomatic ciphers used by the Ministry of Foreign Affairs during WW2 in Slovakia. Here we present cipher systems used by Slovak Attaché and give insight into the encryption problems of the Ministry and embassies.

## 1 Introduction

After the Munich agreement (September 30, 1938), Czechoslovakia was betrayed by her allies and Germany invaded first the Sudeten region, and then Bohemia and Moravia. Former representatives of Czechoslovakia escaped to the UK and organized foreign resistance. In March 1939 a separate Slovak State (Slovakia)<sup>1</sup> was created as a puppet state of the Nazi Germany. The Czech territory was directly absorbed by Germany as a Protectorate.

Cryptology was changing separately in Slovakia and in the Czechoslovakian Government in Exile. We can separate the ciphers used in Slovakia, to *military ciphers*, used by the army<sup>2</sup> and to *diplomatic ciphers* used by the Ministry of Foreign Affairs. In this paper, we focus on the diplomatic ciphers used during the war and their connection to military ciphers. Ciphers, used by the Czechoslovakian Government in Exile, are not covered by this article<sup>3</sup>.

We also introduce a special type of transposition cipher - a *triple columnar transposition*. As far

as we know, there is no information about usage of that kind of transposition in any other country during the WW2.

Presented facts are based on archival documents uncovered in the Military History Archive in Bratislava, Slovak National Archive in Bratislava, Central Military Archives in Prague and in the Security Services Archive in Prague.

## 2 Ciphers Used by the Ministry of Foreign Affairs

The Ministry of Foreign Affairs (Ministerstvo zahraničných vecí - MZV) was completed in 1941 and consisted of four departments. Slovakia had embassies in Berlin, Bern, Budapest, Bucharest, Madrid, Moscow, Rome, Sofia, Warsaw, Vatican, Zagreb and later in Helsinki. Slovak consulates were in Belgrade, Milan, Prague, Stockholm and Vienna.

In diplomatic correspondence, different<sup>4</sup> ciphers were used than those used by the army. The cryptology was a part of the first department and second division of the Ministry (Bielik et al., 1965). The importance of using encrypted telegrams was stressed in a circular letter<sup>5</sup> sent to all foreign representative offices already in 1939.

In the documents we found, the following cipher names were mentioned:

- Hand ciphers: *C*, *XQ*, *R*;
- Cipher machines: Cipher machine<sup>6</sup>, *K*, *Kryha*, *SVERK*.

In the following subsections we briefly introduce the used ciphers (see Figures 6 and 7 for encrypted telegram examples).

<sup>1</sup>The Slovak State name was officially used between March 14 and July 21, 1939. In July 21, 1939 the Slovak State was declared as a republic and renamed to Slovak Republic. The Slovakia acronym was also in use.

<sup>2</sup>Overview of the Slovak military ciphers used during the WW2 can be found in (Antal et al., 2019).

<sup>3</sup>See (Janeček, 1998; Janeček, 2001; Janeček, 2008; Porubský, 2017) for more details.

<sup>4</sup>Except of 10 cipher machines borrowed from the Ministry of National Defence.

<sup>5</sup>Document n. 1882/39 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 42.

<sup>6</sup>Without mentioning a name, borrowed from the Ministry of National Defence.

## 2.1 Cipher C

The most simple cipher in use by the Ministry of Foreign Affairs was a monoalphabetic substitution called *C*. It was used<sup>7</sup> usually alongside with ciphers *K* and *R*. The plain text letters were encrypted in a reversed order (starting from the last letter) and arranged into five letter groups.

A special header was inserted before the encrypted telegram. Firstly a capital letter "C" in a combination with a randomly chosen second letter, then the date, and finally the number of letters of the telegram. Additionally the number of the document was inserted after the telegram.

The weakness of such a primitive cipher was recognized and therefore in the official directive it was only allowed to encrypt less important messages. However, some embassies used only *C* during the first years of the war<sup>8</sup>. In Madrid, the *C* was replaced with a stronger cipher<sup>9</sup> *R* only in 1941. Later on, in April 1942, the Ministry of Foreign Affairs decided to stop distribution of new passwords for *C* due to its weakness<sup>10</sup>.

## 2.2 Ciphers *XQ* and *R*

More powerful hand ciphers were used by the Ministry of Foreign Affairs - a triple columnar transposition called *XQ* and *R*. Based on the available manuals, both names stands for the same cipher<sup>11</sup>. Our opinion is that notation *XQ* means "extended/extra Q", as the Ministry of National Defence used a double transposition cipher called *Q* as a main hand cipher (Antal et al., 2019). The notation *XQ* was later on changed<sup>12</sup> to *R*.

This kind of transposition was used by all embassies and consulates where an encryption service was available<sup>13</sup>.

The triple columnar transposition is an encryption system where three columnar transpositions are applied in a cascade. These ciphers were designed to encrypt messages of length from 50 let-

ters up to 200 for *XQ*, and up to 250 for *R*, respectively. All three transpositions were defined by a specific password (permutation). For *XQ* the password length was limited between 16 and 28, in case of *R* between 16 and 22<sup>14</sup>.

Each password was valid for 24 hours. To avoid encryption with the same password during the day a special alignment technique was used ("usmeriť" and "preskupiť" in original). A simple arrangement could be a rotation of all three permutations until they start with the same number (see Figure 1 - permutations arranged to start with number 7). Another option was to arrange the first permutation to a number  $n$ , the second shifted by one to  $n + 1$ , etc.

Na príklad pre istý deň sú stanovené tieto heslá:

Heslo I:	11, 3, 17, 8, 13, 4, 12, 21, 16, 2, 5, 20, 14, 6, 18, 9, 1, 19, 10, 15, 7.
Heslo II:	9, 5, 10, 4, 17, 3, 11, 2, 15, 8, 12, 1, 18, 13, 7, 16, 6, 14.
Heslo III:	7, 16, 8, 22, 6, 15, 21, 5, 20, 14, 9, 17, 24, 1, 11, 4, 18, 3, 12, 10, 19, 13, 2, 23.

Môže sa na príklad nariadiť, že pre každé odelenie musia heslá začínať stejnou číslou a že sa tieto heslá preskupia pri zachovaní daného poradia čísiel, v smere od ľava do prava. Keď si zvolil šifrujúci v heslách hore uvedených číslou 7 jako začiatok, preskupí ich takto:

Heslo I.:	7, 11, 3, 17, 8, 13, 4, 12, 21, 16, 2, 5, 20, 14, 6, 18, 9, 1, 19, 10, 15.
Heslo II:	7, 16, 6, 14, 9, 5, 10, 4, 17, 3, 11, 2, 15, 8, 12, 1, 18, 13.
Heslo III:	7, 16, 8, 22, 6, 15, 21, 5, 20, 14, 9, 17, 24, 1, 11, 4, 18, 3, 12, 10, 19, 13, 2, 23.

Figure 1: Triple columnar transposition password alignment (from the manual of *XQ*) - in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 580.

This permutation alignment was a part of the message key, and was specially transformed to a five letter group and inserted to the cipher text. Both sides of the communication had to know the position of this group. The arrangement (number) is transformed to a five letter group in two steps.

1. The selected one or two digit number is converted to a five digit number based on the following rules:
  - If the number  $n$  contains one digit only ( $n < 10$ ), create a group consisting of numbers  $\{n, n + 1, \dots, n + 4\}$ , all modulo 10. E.g. 3 is converted to 34567 and 7 is converted to 78901.
  - If the number  $n$  contains two digits ( $n \geq 10$ ), create a group consisting of num-

<sup>14</sup>There is no information why the key space was reduced to the maximal password length 22 for *R*.

<sup>7</sup>Document n. 7865 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 42.

<sup>8</sup>Document n. 28.114, 28.174 and 28.241 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>9</sup>Document n. 28.241 and 28.245 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>10</sup>Document n. 38.009 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

<sup>11</sup>Cipher *R* differs only slightly from *XQ*.

<sup>12</sup>The keys were distributed as *XQ* during years 1939 -1941 and as *R* from 1940/1941. Some documents also contains a dual notation *XQ* with *RR*.

<sup>13</sup>Document n. 28.114 and 28.174 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

bers  $\{n, 0, n+1\}$ . E.g. 11 is converted to 11012.

2. The five digit group is converted to a five letter group using a special "re-encryption" table ("prešifrovacia tabulka" in original). The re-encryption tables were delivered with the daily key. The table consists of 10 columns marked with digits<sup>15</sup> from 1, 2, ..., 9, 0 and contains 26 letters of the English alphabet distributed in three rows in a random order (see Figure 2). For each digit a random letter is selected from the column defined by the digit, so there are two or three options (rows) how to substitute a specific digit with a single letter.

1	2	3	4	5	6	7	8	9	0
X	R	Y	E	B	Q	J	K	H	S
U	N	Z	D	A	C	O	G	V	M
F	W	I	P	L	T				

Figure 2: Triple columnar transposition password alignment re-encryption table (from the manual of XQ) - in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 580.

Before encrypting a message, few rules were defined how to pre-process the input text:

1. Remove accents (e.g.  $\acute{a} \rightarrow A$ ,  $\check{c} \rightarrow C$ , etc.).
2. The end of a sentence can be marked with letter "X".
3. Write numbers with a full name, divided to digits:
  - 1907 as *JEDNA DEVAT NULA SEDEM* (one nine zero seven).
4. Write Roman numbers with a full name after the word "rim":
  - IV as *RIM STYRI*.

<sup>15</sup>We also found variations where the digits starts with 0 and ends with 9.

5. Proper nouns, shortcuts, time, etc. are divided with letter Q.
6. When the text is shorter than 50 letters, insert the "KONIEC" ("STOP") word and a random padding if necessary.

To encrypt<sup>16</sup> a message  $P$  with permutations from the daily key marked  $I, II, III$ , arrangement number  $n$  and position indicator  $ip$ , do the following:

1. Pre-process the input as described above  $P \rightarrow P_1$ .
2. Arrange  $I, II, III \rightarrow I', II', III'$  (based on  $n$ ), and convert  $n$  to a five letter group  $N$ .
3. Apply the first columnar transposition (permutation  $I'$ ) to the input  $P_1 \rightarrow P_2$ .
4. Apply the second columnar transposition (permutation  $II'$ ) to the input  $P_2 \rightarrow P_3$ .
5. Apply the third columnar transposition (permutation  $III'$ ) to the input  $P_3 \rightarrow C'$ .
6. Separate cipher text  $C'$  to five letter groups and insert  $N$  to position  $ip$ , the final cipher text is  $C$ .

It was recommended to use a grid paper for encryption/decryption. See Figure 5 for a step-by-step example of the encryption process ( $n = 7$  and  $ip = 5$ ).

In the investigated cryptologic literature we were unable to find information about any real use of the triple columnar transposition. On the other hand a simpler version - the double columnar transposition - was commonly used as a hand cipher during (and before) WW2. Despite the fact that some special cases (constructions) of this cipher were weak and solvable - it was considered as a secure cipher in general. There are well known materials on how to solve these special constructions. Except of (Kullback, 1934), (Friedman, 1941) and (Barker, 1995) we found also literature about double transposition cryptanalysis in Czech and Slovak language<sup>17</sup>. The most important are:

<sup>16</sup>The decryption is in a reverse order.

<sup>17</sup>Various documents in (Security Services Archive in Prague, 2020), f. Zpravodajská správa Generálního štábu; and (Central Military Archives in Prague, 2020), Security Services Archive, f. MNO HŠ.

- J. Růžek: Encryption systems and manual to solve cryptograms (Šifrovací systémy a návod k luštění kryptogramů), 1926;
- K. Cigán and F. Křepelka: Solving double transpositions (Luštění dvojitéch transpozic)<sup>18</sup>, 1953.

A modern approach to solve the double transposition in general was presented in (Lasry et al., 2014). It is not clear, whether it can be used to solve triple transpositions as well.

### 2.3 Cipher Machines

The Ministry of Foreign Affairs borrowed 10 machines<sup>19</sup> from the Ministry of National Defence on Sep 27, 1939. The machine description is given to consist of

1 box with registration number, 1 crank with screw, 1 auxiliary hook, 1 stand for text, 1 flannel blanket - with each machine, and 1 cipher manual /cipher manual was revised and old one destroyed/.

From comparing the registration numbers<sup>20</sup> it is clear that the borrowed machines used by the diplomacy were the same as used by the army itself. Therefore the machines were available in Czechoslovakia before WW2 (at least from 1938)<sup>21</sup>. In the documents from the Ministry of National Defence, the machine is simply called as "cipher machine" without additional name (Antal et al., 2019). Despite of the missing information, at least the price of the machine is available<sup>22</sup> in Slovak crowns ("120 000 Ks").

Four cipher machines were made available to embassies:

- 1940 - Berlin, Budapest, Moscow;
- 1941 - Rome.

<sup>18</sup>It may be of interest to note that they have broken a double columnar transposition variant used by Yugoslavia.

<sup>19</sup>Document n. 28.050 and 7317 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>20</sup>Document n. 28.114, 1772 and 13.507 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40 and various documents in (Military History Archive in Bratislava, 2020), f. 55.

<sup>21</sup>Document n. 11.654 and 11.331 in (Central Military Archives in Prague, 2020), f. MNO HŠ, boxes n. 283 and 377.

<sup>22</sup>Document n. 75.031 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

In April 1944, only six machines were returned back to the Ministry of National Defence<sup>23</sup>. One machine was burnt in air strike in Berlin (November 1943). The machine from Rome was moved to Venice. It is not clear, whether it reached Slovakia, was destroyed, or captured later in the war. One machine, that was located in Budapest, was presumably faulty, but its final fate is also unknown. Machine from Moscow was left<sup>24</sup> in Sweden embassy basement (without official knowledge of the Swedes) when evacuating Moscow embassy in June 1941. However, the cipher manuals were all destroyed. Further fate of the Moscow cipher machine is also unknown, might it be still in some storage?

In multiple telegrams, the unnamed cipher machines are mentioned along with cipher *K*. According to preserved documents, system *K* was directly connected to the cipher machine borrowed from the Ministry of National Defence. System *K* was used and distributed only in embassies where the cipher machine was sent<sup>25</sup>.

Our early hypothesis was that "K" cipher machine was the same as *Kryha* (see later). There is a circumstantial evidence, that cipher system *K* was a more complex system than *Kryha*. E.g., in July 1940, The Ministry of Foreign Affairs sends<sup>26</sup> a cipher machine to Budapest embassy by courier, along with cipher keys for cipher *K* for the rest of the year. In this telegram, the Ministry urges the embassy not to encrypt messages longer than 200 letters. They also give operation instructions for the machine:

When encrypting with a machine, check each line, by operator marking **the status of the cylinders**, and send the message when you have checked it all out only.

Treat the machine, clean it at least every month and lightly grease. In case of the smallest error that you will not be able to eliminate, do not try to disassemble the machine, but immediately report to

<sup>23</sup>Document n. 28.050 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>24</sup>Document n. 28.304 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>25</sup>Document n. 28.114 and 28.174 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>26</sup>Document n. 6987 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.



the headquarters that the machine is not working, of course also encrypted /R/.

From the description, it is not clear, whether mentioned cylinders ("valce" in original) could denote two cipher rings of *Kryha*. Machine "K" could also be a completely different cipher machine of rotor type ("valce" can also denote drums, gear wheels, or *Enigma* rotors). One potential candidate is the commercial *Enigma K* machine (Hamer et al., 1998) (used also by Switzerland).

In telegram from March 1941, Dr. Šulík<sup>27</sup> sent a wire to Berlin, Budapest, Rome and Moscow embassies<sup>28</sup> that system *K* is recommended for longer messages. However there is a concern with decryption errors caused by transmission errors (by post office). The telegram indicates, that when a single mistake is made in a five letter group, the message group can still be decrypted. However, if two letters are changed (or) swapped, the whole telegram is unreadable and must be resent. The cause of this behaviour is attributed to the "state of the cylinders".

However, situation is more complicated as wrote Dr. Bukovinský<sup>27</sup> in December 1941 (a handwritten note in the original document) :

Because instruction is incorrect, I have burned all originals of the expedition.

From pencilmarks on the telegram, the incorrect part is essentially the description of the behaviour of transmission errors. Thus we cannot properly conclude anything about the cipher system based on this telegram.

Further details reveal that longer messages should be split into groups of at most 300 letters. The telegram starts with *K*, a date (day only), and length of each paragraph. The first starting group of the first paragraph contains six letters of the "individual password" (see Figure 3).

It is not clear how an individual password was used. If the unknown machine was *Kryha*, it could denote setting of clocking pins, or the setting of the alphabet on cipher rings. It could also be a password to a superencryption system. Alternatively, it could be similar to a standard *Enigma*

<sup>27</sup>Secretary of the Ministry of Foreign Affairs working in cipher department. We have no further details available.

<sup>28</sup>Document n. 28.090 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

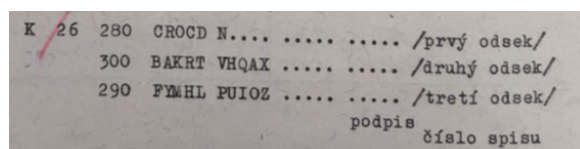


Figure 3: *K* message divided to three paragraphs - Document n. 28.090 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

six-letter indicator, which could support *Enigma K* hypothesis.

Dr. Šulík further mentioned that previous telegrams contained (the same) individual password at the beginning of each group of 300 letters, which practice was forbidden in the telegram. This property can help identify diplomatic telegrams encrypted by the system *K* between July 1940 and March 1941.

There were continuous problems with this cipher machine in Berlin<sup>29</sup>, Budapest<sup>30</sup>, Moscow<sup>31</sup> and Rome<sup>32</sup>. Some embassies also requested a new cipher machine. Probably for this reason the available cipher machines were replaced by a (different/new?) cipher machine openly called *Kryha* in 1943.

In a document<sup>33</sup>, there is an explicit reference to "six complete cipher machines KRYHA-S TAN-DaRD". *Kryha* Standard was a commercial cipher machine released by Alexander (von) *Kryha* in 1924 (Schmeh, 2010). Machine was based on a cipher disk, with 2 rings: outer ring was fixed, and inner ring was rotated by a clockwork machine with irregular stepping. Ring alphabets could be changed by the operator. To encrypt a message, operator pushed the button to rotate the machine, and then replaced plain text letter found on the inner ring by cipher text letter on the outer ring. From cryptological point of view, the cipher is a polyalphabetic substitution with individual alphabets rotated by the amount given by clocking se-

<sup>29</sup>Document n. 75.242 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

<sup>30</sup>Document n. 38.019, 38.024, 38.026 and 38.124 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>31</sup>Document n. 610 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

<sup>32</sup>Document n. 28.272 and 52/dov/Dr.M.-taj in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40 and 41.

<sup>33</sup>Document n. 75.020 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

quence of the machine. Such a system was known to be broken even before WW2 (Marks, 2011).

There were at least eight *Kryha* machines available to the Ministry of Foreign Affairs, some were distributed to embassies in the following years:

- 1943 - Helsinki;
- 1944 - Bucharest, Madrid, Budapest;
- 1945 - Berlin.

Another cipher machine, called *SVERK* (see Figure 4), was sent to Helsinki in 1943. The document<sup>34</sup> also describes some parts of the machine - it contains one encryption wheel and plugs to the wheel. In a different document the machine sent to Helsinki (referring to the same registration number and document number) is called *Kryha*. Therefore we think that *SVERK* is only a cover-name for *Kryha*.

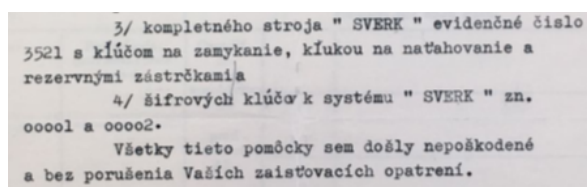


Figure 4: *SVERK* cipher machine - Document n. 12/dov. 1943 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

### 3 Encryption Problems on Embassies and on the Cipher Department

The encryption service on embassies did not work without problems. There were three major types of problems:

1. Telegram corruption - From the documents we found so far, the most frequent problem was that the telegrams could not be decrypted due to corruption. In some cases the post office was responsible<sup>35</sup> for modifying (or dropping) the part of the encrypted text, in other cases, it was a fault in the encryption officers work<sup>36</sup>.

<sup>34</sup>Document n. 75.010 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

<sup>35</sup>Document n. 28.090, 28.272 and 90.000 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40 and 42.

<sup>36</sup>Document n. 38.021 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

2. Not respecting the manuals and encryption directives - E.g. partially encrypted messages<sup>37</sup>; resending previously encrypted messages in plain text<sup>38</sup>; writing about encryption<sup>39</sup> etc.

3. Problems with cipher machines (see section 2.3).

Because of the frequency of operation problems the Ministry of Foreign Affairs informed the embassies several times about cryptographic principles<sup>40</sup>, such as:

- Never use the word "cipher" in documents.
- The content of the message should be reworded.
- All used papers must be burned after encryption/decryption.
- To any encrypted message reply by using encryption only.

In 1941, Dr. Bukovinský created a report<sup>41</sup> about experiences and problems in the cipher department of the Ministry of Foreign Affairs. We briefly summarize his report:

- The department is located in a room, where other personnel (not from the cipher department) is also located, and there are even visits from outside the Ministry.
- There is no curtain on the window, so the cipher machine is visible from the opposite building through the window.
- There are no special blankets used for encryption (only a standard paper).
- The used cipher is marked on telegrams, so the foreign countries can simply sort the encrypted telegrams by the used cipher system.

<sup>37</sup>Document n. 75.219 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

<sup>38</sup>Document n. 760 and 28.061 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

<sup>39</sup>Document n. 38.008 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41 - "chiffraantwort folgt" was used.

<sup>40</sup>Document n. 28.302 and 75.008 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40 and 41.

<sup>41</sup>Document n. 28.300 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

- Cipher machines are not working correctly because of incorrect usage.
- Systems *K* and *R* are used to encrypt non important messages, and the most simple system *C* is used to encrypt important messages.

Later, in 1943, Dr. Bukovinský was asked<sup>42</sup> to check the embassies in Budapest, Zagreb, Rome and Berlin. The goal was to check and correct the lack of encryption:

- The cipher machine in Budapest was set incorrectly.
- Only hand cipher was available in Zagreb. The secretary of the embassy was trained in encryption.
- The cipher machine in Rome was not working.
- In Vatican, there were no ciphers available, and nobody knew encryption.
- In Bern, the head of the office did not know encryption.

We do not know whether these problems and mistakes were exploited by attackers in practice. If cryptanalytic public is interested, we have found some encrypted telegrams in the archives that remain an unsolved challenge.

## Acknowledgments

We are grateful to the Military Intelligence (Ministry of Defence of the Slovak Republic), for the help and resources made available. This work was partially supported by grants VEGA 1/0159/17 and VEGA 2/0072/20.

## References

- Central Military Archives in Prague (Vojenský ústřední archiv v Prahe).
- Military History Archive in Bratislava (Vojenský historický archiv v Bratislave).
- Security Services Archive in Prague (Archív bezpečnostních složek v Prahe).
- Slovak National Archive in Bratislava (Slovenský národný archiv v Bratislave).

<sup>42</sup>Document n. 75.001 in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.

- Eugen Antal, Pavol Zajac and Otokar Grošek. Cryptology in the Slovak State During WWII. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt 2019*, pages 23 - 30.
- František Bielik, Ján Gáll and Klára Kunkelová. Ministerstvo zahraničných vecí 1939 - 1945 Inventár. 1965. Štátný slovenský ústredný archív v Bratislave.
- Wayne G. Barker. Cryptanalysis of the Double Transposition Cipher: Includes Problems and Computer Programs. 1995. Aegean Park Press.
- William F. Friedman. Military Cryptanalysis. 1941. US Government Printing Office.
- David H. Hamer, Geoff Sullivan and Frode Weierud. Enigma variations: An extended family of machines. 1998. *Cryptologia*, 22(3):211-229.
- Jiří Janeček. *Gentleman (ne)čtou cizí dopisy* (in Czech). 1998. Books - bonus A. ISBN:8072420232.
- Jiří Janeček. *Válka šifer* (in Czech). 2001. Votobia. ISBN:8071985058.
- Jiří Janeček. Rozluštěná tajemství (in Czech). 2008. XYZ. ISBN:8086864545.
- Solomon Kullback. General Solution for the Double Transposition Ciphers. 1934. Aegean Park Pr.
- George Lasry, Nils Kopal and Arno Wacker. Solving the Double Transposition Challenge with a Divide-and-Conquer Approach. 2014. *Cryptologia*, 38(3):197-214.
- Klaus Schmeh. Alexander von Kryha and His Encryption Machines. 2010. *Cryptologia*, 34(4):291-300.
- Philip Marks. Operational Use and Cryptanalysis of the Kryha Cipher Machine. 2011. *Cryptologia*, 35(2):114-155.
- Štefan Porubský. Application and Misapplication of the Czechoslovak STP Cipher During WWII. 2017. *Tatra Mountains Mathematical Publications*, 70(1):41-91.



# Appendices

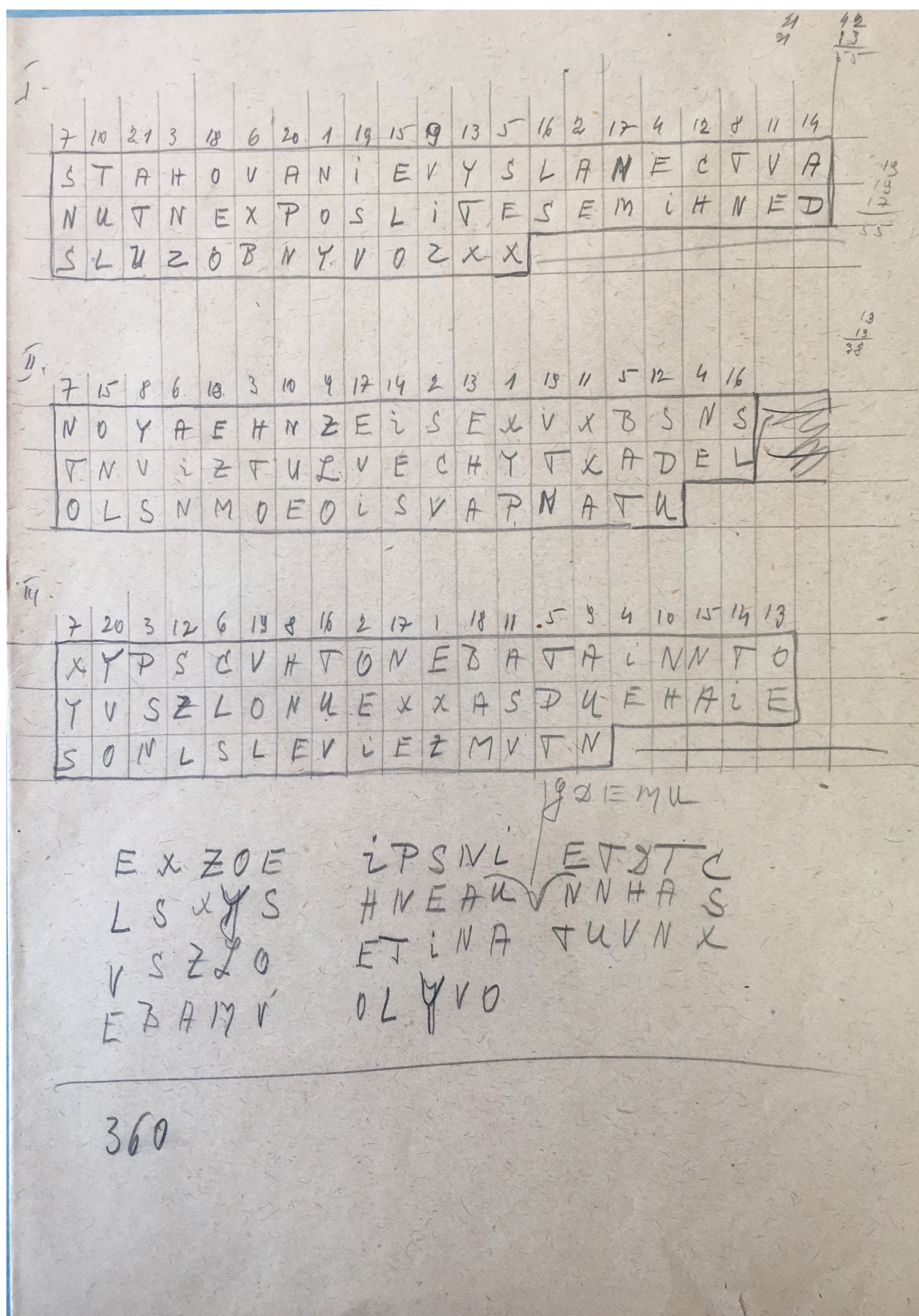


Figure 5: An example of triple columnar transposition encryption - in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 507.



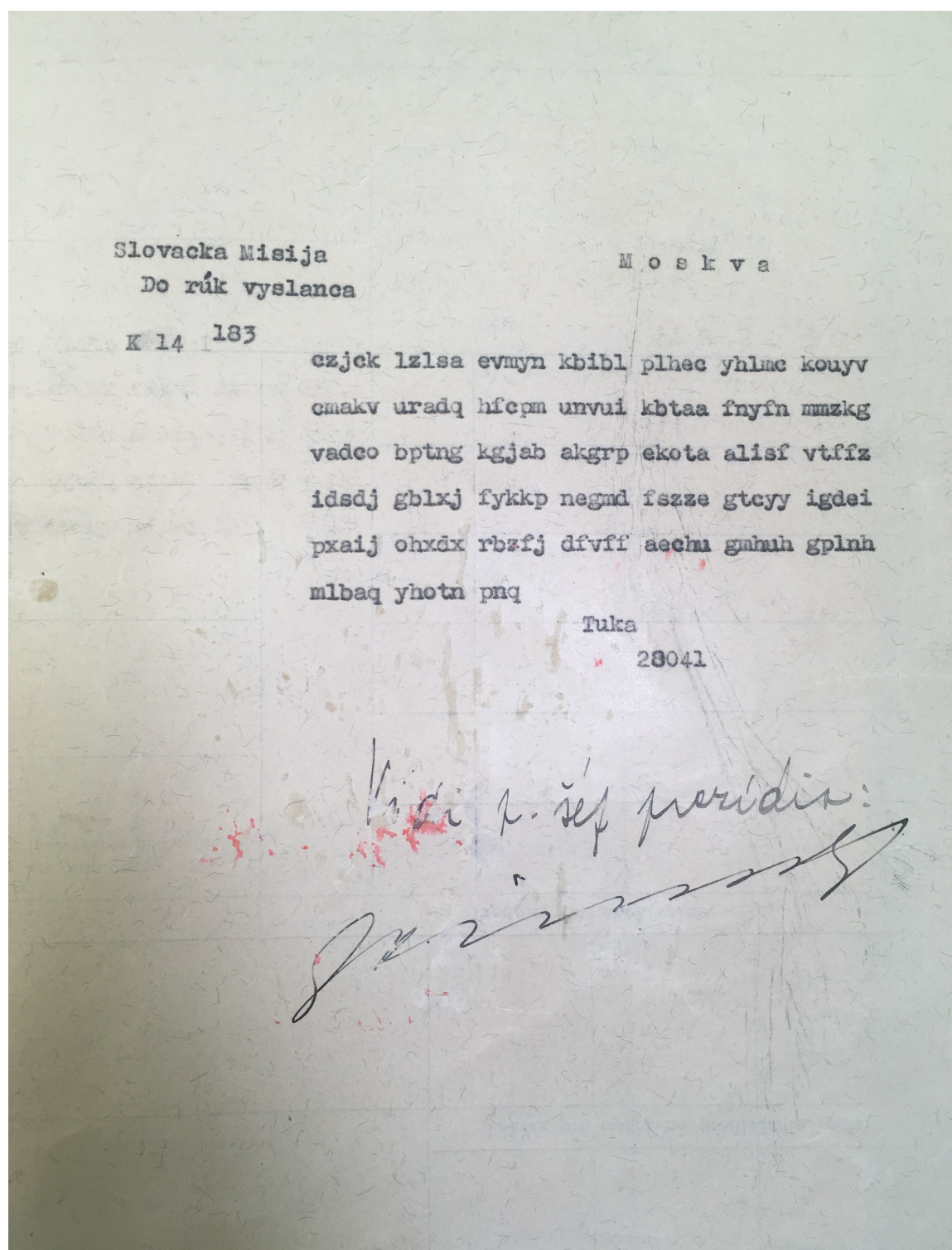


Figure 6: Text encrypted with system K - in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 40.

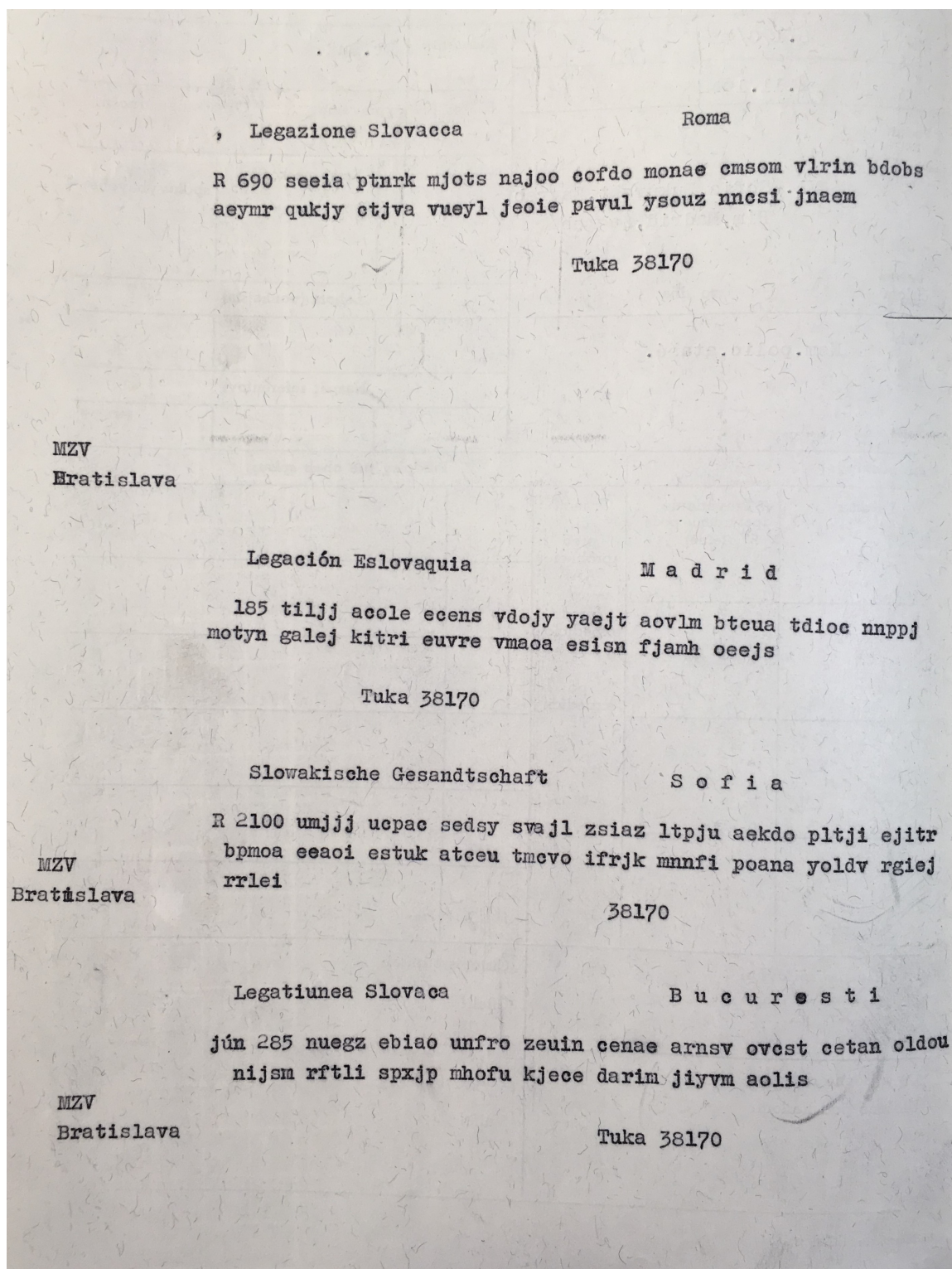


Figure 7: Encrypted telegrams sent to Rome, Madrid, Sofia and Bucharest - in (Slovak National Archive in Bratislava, 2020), f. MZV, box n. 41.



# The Use of Project Gutenberg and Hexagram Statistics to Help Solve Famous Unsolved Ciphers

**Richard Bean**

School of Information Technology and Electrical Engineering  
University of Queensland, Australia 4072  
r.bean1@uq.edu.au

## Abstract

Project Gutenberg, begun by Michael Hart in 1971, is an attempt to make public domain electronic texts available to the public in an easily available and useable form. The number of available texts reached 60,000 by 2019. Classical cryptanalysis methods rely on the development and use of high-quality frequency tables of letter arrangements from a variety of sources. As the amount of text grows, frequency tables of higher orders can be developed and may provide more solving power for classical cryptographic algorithms. As a side-effect of the availability of a wide range of public domain texts, we were able to develop hexagram frequency tables of letters in the English language which were then a crucial factor to solving an unsolved transposition cipher of Mahon and Gillingham (2008). The texts themselves were then used as input to solve a book cipher of Thouless (1948) using the same scoring method.

## 1 Introduction

Project Gutenberg (Hart, 1992) was begun by Michael Hart in 1971. Initially, Hart was given a large amount of computer time on a mainframe computer at the University of Illinois. He used it to type and store the Declaration of Independence as the first “etext” or electronic text of Project Gutenberg. In 1989, the 10th book, the King James Bible, had been posted, and by 1994, the project had digitized 100 books with the release of the Complete Works of Shakespeare. The 1,000 book mark was reached in August 1997, with 10,000 in October 2003, and 60,000 in July 2019.

The use of frequency tables is essential in classical cryptanalysis. For a putative “solution” or

deciphering of a ciphertext, whether by hand or by machine, the cryptanalyst must evaluate how close the solution is to actual text in the target language. In classical cryptanalysis, a small change in the key results in only a small change in the ciphertext. If each solution can be “scored” using frequency table data, the methods of “hill climbing” or “simulated annealing” can be used to improve the score. The idea of the algorithms is to gradually (in the case of hill climbing, monotonically) improve to the highest scoring solution, which may be the correct decipherment. The scoring is generally carried out by the method of log-likelihood; that is, evaluating the likelihood of a text using the product of the probabilities of its component letter frequencies; or more precisely, the sum of the logarithms of the probabilities. A more complete explanation and literature review can be found in (Lasry, 2018).

An  $n$ -gram frequency table will list all possible contiguous sequences of  $n$  letters and their relative frequency as evaluated from some corpus of text available to the creator. In the past, newspapers, the King James Bible, telegrams, and other books have been used as sources for building frequency tables.

For example, in English, the most common 1-gram is the letter “E” while the most common 3-gram is “THE”. A knowledge of the most common English letters (i.e. 1-gram frequencies) allows a cryptanalyst to quickly solve monoalphabetic substitution ciphers, while more complicated ciphers may require the use of bigrams (2-grams), trigrams (3-grams), quadgrams (4-grams), and so on. Books on cryptography published during the 20th century often contained frequency tables for 1, 2, and 3-grams. The computation of  $n$ -gram frequency probabilities over sequences of characters is typically referred to as “character  $n$ -gram language modelling” or simply “language modelling”. (Nuhn et al., 2013; Ravi and Knight, 2008;

Hauer et al., 2014)

Lyons (2012) on his Practical Cryptography website, stated that in his experience, “quadgram frequencies worked slightly better than trigrams, trigrams work slightly better than bigrams, but that going higher than 4 letters does not really add any benefit”.

In this paper we will examine a cipher where hexagram (6-gram) frequency tables enabled the solution of an unsolved cipher. Hexagrams which did not occur in the source text were assumed to have a frequency of one, in order to avoid a “zero probability” in the likelihood evaluation function.

## 2 History of Published Frequency Counts

Gaines (1956) in her classical cryptanalysis textbook based her digram frequency tables on those found in (Pratt, 1942) and (Hitt, 1916). Pratt used 20,000 digrams and trigrams while Hitt used 10,000 letters of semi-military text. A digram chart by O. Phelps Meaker in the book is based on 10,000 letters. Friedman (1923) in his first book also used the counts of Hitt.

Later, Friedman (1952) presented an Appendix of letter frequency counts based on five sets of 10,000 letters from “Governmental plain-text telegrams.” His National Security Agency colleague, Sinkov (1966) in his textbook, based his monogram and digram tables on 80,000 letters of newspaper text. By 1973, Friedman’s co-author Callimahos had published an update “English language statistics based on a count of 2,022,000 letters.” (Callimahos, 1973)

Mahon and Gillogly (2008) described building a frequency table from all the Gutenberg books from 1990 to 2006: 10,607 books, 730 million words, and 4.4 billion letters. Previously Gillogly (1996) had used trigram frequency tables.

A classic highly cited paper on frequency tables was Mayzner (1965) which used 20,000 words. Norvig (2013) updated Mayzner by examining 3,563,505,777,820 letters from the Google Books corpus. Using a count of the number of times each phrase of contiguous words occurred, he developed frequency counts for  $n$ -grams up to  $n=9$ ; although these counts were derived from the Google books  $n$ -gram data, and so they do not reflect statistics based on the raw book data.

## 3 IRA Unsolved Cipher

Mahon and Gillogly (2008) decrypted over 1,000 ciphertexts from the 1920s which were from the estate of Moss Twomey, a former chief of staff of the IRA (Irish Republican Army). Usually, the ciphertexts were incomplete columnar transposition ciphers with a column width, or period, of between 6 and 15, with the most common period being 12. Sometimes, the transposition ciphers contained polyalphabetic ciphers in the middle for extra security.

In his chapter describing the technical aspects of the decryption, Gillogly stated that they eventually produced good decryptions of all but one of the transposition ciphers. This cipher was from 16 November 1926, and was marked as containing 52 letters, although only 51 were present in the ciphertext.

GTHOO RCSNM EOTDE TAEDI NRAHE  
EBFNS INSGD AILLA YTTSE AOITD  
E

Gillogly stated that he tried a number of approaches, including assuming the missing letter was in each of the fifty-two positions, or leaving out a letter in each position, but none of the attacks succeeded.

We tried the same basic approach of Gillogly: a “random restart” or “shotgun” hill-climbing solver, beginning with a random allocation of complete and incomplete columns. The algorithm proceeds sequentially through all possibilities of column pair swaps, and evaluates the score of each result. If a column pair swap is found to increase the score of the result, the swap is carried out and the process is repeated. If no column pair swap increases the score, a different random allocation of columns is chosen and the process restarts.

At first, we used quadgram and 5-gram statistics, but the best scoring results at all periods (6 to 15) were not at all close to English. A comment on a blog of Klaus Schmeh on the cipher suggested the plaintext might be Gaelic; although this seemed unlikely, as all the other solved cryptograms in the book were in English.

A few months later, after noting the success of Lasry in his PhD thesis (Lasry, 2018) with hexagram frequency statistics, we developed the frequency tables based on the Project Gutenberg English language books which were available (about 37,000 books at the time). This amounted to about

10 billion letters.

After this, the scoring function returned a solution with a “local minimum” at period 11; that is, the score of the best solution at period 12 was worse. Thus, we focussed our efforts on period 11. The best solutions all seemed to contain the hexagram “LIGNIT” and in the context of messages about the Irish Republican Army in the 1920s, it seemed logical that the plaintext could contain the word “GELIGNITE”. We inserted the letter “L” between the double “E” in the ciphertext and forced “GELIGNIT” to be present in the plaintext output. The best solution then obtained was as in Table 1.

R	E	G	E	L	I	G	N	I	T		S
C	O	T	L	A	N	D	S	T	A		E
S	T	H	E	Y	R	A	I	D	E		A
N	D	O	B	T	A	I	N	E	D		O
M	E	O	F	T	H	L	S				

Table 1: Plaintext with missing columns.

After we contacted Gillogly, he noted that the obvious “corrected” solution “*Re Gelignite Scotland states they raided and obtained some of this*” would have missing letters E, T, D and S exactly 12 letters apart in the empty column in the table. Thus, the original cipher period was intended to be 12, with plaintext length 56. Gillogly noted the “L” in the “THLS” word was actually an overstrike of “L” and “I”.

After searching back through Project Gutenberg, we discovered the most common transposition key that could lead to the ciphertext column ordering (BCAFIEHGKDLJ) was the 12 letter phrase “CHAMPIONTHUS” from Thomas Mallory’s “Morte d’Arthur” - *endure as his true champion. Thus when Sir Percivale ....*

#### 4 Thouless Unsolved Cipher

In papers published in 1948 and 1949 in the “Proceedings of the Society for Psychical Research”, (Thouless, 1948; Thouless, 1949) Thouless proposed a “test of survival”. Three “passages” with encrypted texts were provided, and the intention was for Thouless to keep the keys for each passage secret in his lifetime, and after his death, attempt to telepathically transmit the keys for each passage via mediums to the living. If he succeeded, the ciphertexts could be deciphered correctly, proving that the keys had been received from beyond the grave. Supposedly, the first passage he proposed was deciphered by a cryptanalyst soon after pub-

lication. The cryptanalyst deciphered Thouless’s Playfair cipher, using the keyword *SURPRISE* resulting in a plaintext from the Shakespeare play *Macbeth: Balm of hurt minds, great nature’s second course....*

The third passage he proposed, intended to replace Passage I, was a doubly enciphered Playfair text, using two keyword based squares. Gillogly and Harnisch (1996) determined that the keywords for Passage III were *BLACK* and *BEAUTY* with plaintext *This is a cipher which will not be read unless I give the key words*. Thus, the only remaining test was Passage II.

This had been enciphered with a book cipher, using modulo 26 arithmetic. The example Thouless gave to demonstrate the cryptographic process used the Shakespearean phrase “To be or not to be...”. Then with T being the 20th letter of the alphabet and O being the 15th,  $20 + 15$  was reduced to 9 modulo 26, represented as the 9th letter of the alphabet I, which was then used as an additive to each letter of the plaintext. Thus the first word of the phrase was used to create an additive for the first letter of the plaintext, and so on.

Passage II’s 74-letter ciphertext was as follows:

INXPH CJKGM JIRPR FBCVY WYWES  
NOECN SCVHE GYRQJ TEBJM TGXAT  
TWPNH CNYBC FNXPFLFXRV QWQL

Gillogly and Harnisch noted that they had tried hundreds of books as the keytext to solve Passage II, including the King James Bible (Gutenberg #10), Shakespeare’s works (Gutenberg #100), and the text of “Black Beauty” by Anna Sewell (#271).

After the stripping and processing of the 37,000 books for the frequency table used above, we decided to see if the Thouless key phrase was contained within the Project Gutenberg texts already scanned. After writing and starting our program, about five days and 31,000 books later, we found that the text of the poem “The Hound of Heaven” by Francis Thompson (#41215) gave a high scoring result.

-5309238 CEVHHZGMKLUCCESS-  
FULEXPERIMENTSOFTNEKKIWTDXDAU-  
GIVESTRVMGEVIDENCEFOROXRVIVAL  
THE HOUND HEAVEN I FLED HIM DOWN  
NIGHTS DAYS ARCHES YEARS ...

This was a huge improvement over the other two best solutions the program had found.

-6137393 HUGFCEWLTGAGJPTJAN-  
NOXPERIMENTSOFTHISKIWTDXDAZVE-

BZTZVRVREPGQJVTUCFLXWBVRRDZ  
VOICE ROUND ME LIKE BURSTING SEA  
ILLUSTRATION ...

-6099427 NOPKOLOKKO-  
HFEIMTENYEUCZWYEWUHMUFD-  
DYSCARDINGREINASWIGGINORN-  
MGBDKHIWDPDIMKZ BUY WELL I WANT  
THEM CAN GO YOUR WAY FAR CON-  
CERNED THERE ONLY ONE THING FOR  
OFFER ...

As the outputs contained “UCCESSFULEXPERIMENTSOFF” and “EVIDENCEFOR” this was evidently the correct plaintext. After some cleaning, this was verified, with plaintext “A number of successful experiments of this kind would give strong evidence for survival”.

The search must have been out of sequence, because Book #1469 “Francis Thompson’s poems” has the poem and was first published in Project Gutenberg in July 1998. This is the 1279th book, if only the English language books are considered in sequence. This indicates that Gillogly and Harnisch would have found the keywords if they had waited two or three more years and examined the English books of Project Gutenberg sequentially.

## 5 Conclusion

The use of large English text corpuses such as Project Gutenberg has enabled the solution of heretofore insoluble ciphers. The IRA challenge cipher was difficult to solve, as it was of a very short length, the preamble contained an incorrectly recorded ciphertext length, while the ciphertext itself had one incorrect letter and four missing letters. However, with some knowledge of the context (likely to refer to “gelignite”) assisted by the hexagram table frequencies, the solving program could be manually guided to the correct solution.

The Thouless cipher could not have remained unsolved forever, as diligent volunteers of Project Gutenberg have been typing in or digitizing public domain books over many years. As Thouless intended to transmit the identity of the key text via medium, it seemed likely that the text would be a well-known one, and it proved to be so. With growing computational speed, networking facilities and storage, the key texts of both remaining passages were discovered relatively soon after Thouless’s death in 1984.

Higher order frequency tables have been used

recently in other cipher challenges. Van Eycke and Helm (from (Schmeh, 2019)) developed an octogram (8-gram) frequency table based on 2 TB of data scraped from around the Internet. This included Project Gutenberg. In 2019, they used this table to solve a bigram challenge of Schmeh, setting a world record of solving a 1,000 and then a 750 letter challenge cipher. Obviously, frequency tables of  $n$ -grams, where  $n$  is even, are particularly amenable to the solution of digraphic cipher challenges, as they can assess the likelihood of several bigrams concatenated together.

## References

- Lambros D Callimahos. 1973. English language statistics based on a count of 2,022,000 letters. Accepted by National Archives of the US, 1978.
- William Frederick Friedman and Lambros D Callimahos. 1952. *Military Cryptanalytics*, volume 1.
- William Frederick Friedman. 1923. *Elements of cryptanalysis*, volume 3.
- Helen F Gaines. 1956. *Cryptanalysis: A Study of Ciphers and Their Solution*, volume 97. Courier Corporation.
- James J Gillogly and Larry Harnisch. 1996. Cryptograms from the crypt. *Cryptologia*, 20(4):325–329.
- Michael Hart. 1992. The history and philosophy of project gutenberg. *Project Gutenberg*, 3:1–11.
- Bradley Hauer, Ryan Hayward, and Grzegorz Kondrak. 2014. Solving substitution ciphers with combined language models. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2314–2325.
- Parker Hitt. 1916. *Manual for the Solution of Military Ciphers*. Press of the Army Service Schools.
- George Lasry. 2018. *A methodology for the cryptanalysis of classical ciphers with search metaheuristics*. Kassel university press GmbH.
- James Lyons. 2012. Quadgram statistics as a fitness measure. <http://practicalcryptography.com/cryptanalysis/text-characterisation/quadgrams/> Visited 27 April 2020.
- Thomas G Mahon and James Gillogly. 2008. *Decoding the IRA*. Mercier Press.
- Mark S Mayzner and Margaret Elizabeth Tresselt. 1965. Tables of single-letter and digram frequency counts for various word-length and letter-position combinations. *Psychonomic monograph supplements*.

- Peter Norvig. 2013. English letter frequency counts: Mayzner revisited. <http://norvig.com/mayzner.html> Visited 24 April 2020.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. Beam search for solving substitution ciphers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1576.
- Fletcher Pratt. 1942. *Secret and urgent: The story of codes and ciphers*. Blue Ribbon Books.
- Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 812–819.
- Klaus Schmeh. 2019. Bigram 750 challenge solved, new world record set. <http://scienceblogs.de/klausis-kryptokolumne/2019/12/19/bigram-750-challenge-solved-new-world-record-set/> Visited 24 April 2020.
- Abraham Sinkov. 1966. *Elementary cryptanalysis*, volume 22. MAA.
- Robert H Thouless. 1948. A test of survival. In *Proceedings of the Society for Psychical Research*, volume 48, pages 253–263. Society for Psychical Research.
- Robert H Thouless. 1949. Additional note on ‘a test of survival’. In *Proceedings of the Society for Psychical Research*, volume 48, page 342. Society for Psychical Research.



# A Partenio's Stegano-Crypto Cipher

Paolo Bonavoglia

Mathesis Venezia c/o Convitto-Liceo Marco Foscarini 4942 Venezia, Italy

paolo.bonavoglia@mathesisvenezia.it

## Abstract

Pietro Partenio's second cipher in the CX<sup>1</sup> book of 1592-93 is an unusual mix of a *cifra sospetta* (suspicious cipher) and a *cifra non sospetta* (non suspicious cipher), that is cryptography and steganography. The cipher has some possible roots in Trithemius's Ave Maria, Vigenère's and Francis Bacon's ciphers.

## 1 Pietro Partenio

Pietro Partenio was one of the most brilliant Venetian cryptologists. He was born in 1538 or possibly in January 1539<sup>2</sup>.

He was a notary whose deeds are stored in the State Archives of Venice, for the 1563-1610 period under his name, and from 1610 to 1628 in association with another name; so he probably died between 1618 and 1628, a very long life for the XVI century.

In his notary deeds the name of Hieronimo di Franceschi, the main CX deputy for ciphers in those years, is often present as a solicitor for other people. So Partenio and Franceschi knew each other, and the first mentions very often Franceschi in his cryptographic papers, comparing the well known *cifra delle caselle*<sup>3</sup> used by Venetian embassies in the 1577-1595 period with his ciphers and boasting the superiority of his ones. Apparently there was a mix of friendship and rivalry between the two.

<sup>1</sup>CX is the acronym for *Consiglio di Dieci* = Council of Ten, the powerful council of the Republic of Venice that had wide powers in matter of security and domestic and foreign policy, and was also in charge for choosing the deputies for ciphers, and approving the ciphers to be used. The ten members were elected by the *Maggior Consiglio*, the House of Lords of Venice; the Doge and his six advisers had the right to join the meetings of the CX, that could thus have up to 17 participants.

<sup>2</sup>The date can be inferred from the letter Partenio wrote to the CX, in January 1606, where he states to have reached the age of 67. *ASVe CX deliberazioni segrete*, f. 28. ASVe is an acronym for *Archivio di Stato di Venezia* (State Archives of Venice).

<sup>3</sup>See (Bonavoglia, 2020).

He started to design ciphers for the CX in the early 1590s when he was in his fifties; between 1592 and 1593 he gave seven ciphers to the CX.

Most of his ciphers are clearly derived from the two main ideas of Franceschi: superencryption of a nomenclator as in the already mentioned *cifra delle caselle*, and fake key ciphers; and Partenio repeatedly criticized Franceschi's ciphers claiming his own were more secure and easier to use.

From the Archives' papers quite a different story comes out; up to now only three diplomatic messages from Paris using one of Partenio's ciphers, in July-August 1595, have been found; apparently the secretaries found the cipher too complicated and cumbersome to use.

In spite of this failure, Partenio's ciphers are fascinating and unusual for those years. One among them is the cipher presented here, a classical nomenclator followed by a super-encryption generating a common language message, that is a kind of steganography.

Before looking at the cipher in detail, a few words about steganography.

## 2 Non Suspicious Ciphers, alias Steganography

Steganography, the art of concealing secret messages inside innocuous texts, is very old, indeed older than cryptography. Invisible inks, dissimulated writing using conventional words and phrases in most cases preceded classical cryptography; so was the case in Venice too, whose first encoded messages used conventional language.<sup>4</sup>

In the Italian cryptographic jargon of those times *cifre sospette* (suspicious ciphers) were normal encryption methods: the resulting cryptograms were easily recognized as encrypted texts, and that's why they were suspicious; *cifre non sospette* (non suspicious ciphers) were methods producing plausible text, apparently innocuous, while hiding a secret message.

<sup>4</sup>See (Pasini, 1872).

Steganography was largely neglected by the cipher offices after cryptography became the standard method used by ambassadors and military chiefs to communicate in a secret way. It remained viceversa very popular among amateurs.

In spite of this we find several interesting *cifre non sospette* both before and after Partenio. We will see a few that have something in common with this one.

### 3 Partenio's non Suspicious Ciphers

And now let's go to the Venetian Archives, where a fine handwritten parchment book<sup>5</sup> has a *cifra non sospetta* (non suspicious cipher), a curious mix of cryptography and steganography.

Partenio presented this cipher to the CX during a meeting held in 1592.

Later, in 1606, he wrote a booklet, to be used as a textbook for teaching ciphers and cryptography to a few young pupils. One of them was Ottaviano Medici, a future CX deputy for ciphers. In this booklet he presents again this non suspicious cipher, adding to it a fake key variant.

Let us now see in detail these two ciphers.

### 4 The 1592 Second Cipher

The basic idea of this cipher is to encrypt a message using a 3 digits nomenclator (see figure 1); the resulting cryptogram is super-encrypted substituting orderly every digit with a piece of a sentence to be chosen among ten variants as shown in figure 2.

The pieces of sentence are so conceived to give a plausible message as shown in the following example. Suppose the message to encrypt is:

*È venuto noua che Re di Spagna è risentito con pericolo di uita*<sup>6</sup>,

for a total of 51 letters.

The nomenclator has a cipher, 315, for the statement *È venuto noua che*, a cipher, 678, for *Re di Spagna* and cipher 312 for *È risentito con pericolo di uita*. So the first step gives the cryptogram 314 678 312.

The second encryption requires to encrypt every digit with a different piece of phrase; the first

<sup>5</sup>ASVe CCX Raccordi, Registri 1. CCX is an acronym for *Capi del Consiglio di Dieci*, the three chiefs of the Council of Ten; they were elected monthly and had final court enforcement powers.

<sup>6</sup>English: A news has come that the King of Spain is ill in danger of life.

digit, 3, has to be looked in the first column of the phrasebook, where one finds "*Ser.<sup>mo</sup> Principe*", the second digit 1 has to be looked for in column 2, and you get "*non si marauiglia*", and the job continues until column 9. Finally one gets this fake message:

*Ser.<sup>mo</sup> Principe non si marauiglia se non ha mie lettere che se uolessi dargli raguaglio con lettere sospette sarebbe tutto squarciato con disgusto suo*<sup>7</sup>.

for a total of 125 letters, more than double the ones in the plain text; and the example is somewhat artificial, a best case, being composed of statements present in the nomenclator; if they were not, the message would have to be split into syllables and letters, 30 of which would generate 90 numbers of cipher text, and a thousand letters of fake text; in such case, one would need a much larger phrasebook, or limit himself to very short messages.

Indeed the method is practical only for very short, telegraphic messages. The clumsiness is anyway a problem common to most steganographic methods.

Another problem is that using always the same phrasebook will produce messages very similar, and the enemy intercepting them would be alerted; so the cipher will be no more *non sospetta*. For this reason one should change the key very often, or prepare and exchange a very long strip of hundreds of plausible words or phrases.<sup>8</sup>

Only in this last case the cipher could be considered very difficult to break, without the *scontro* (the phrasebook), even knowing the method.

### 5 The 1606 Remake

As anticipated above, in 1606 Partenio, wrote a book, signed *Pietro Partenio di sua mano*<sup>9</sup> that contains four ciphers, with some new ideas. The third of them is a cipher very similar to the 1592 one, but with an increased phrasebook (15 items instead of 9, see figure 4) allowing for longer messages, a different nomenclator (see figure 2) using more common short messages, and the following interesting variant.

<sup>7</sup>Most Serene Prince, do not be surprised if you do not receive my letters, because if I would give details with suspect letters, you would be torn with disgust.

<sup>8</sup>Something of the like was made by Abbot Trithemius for his *Ave Maria* cipher. See paragraph below.

<sup>9</sup>The manuscript is kept in the ASVe, *CX Cifre, chiavi e scontri di cifra ...*, busta 2.

## 5.1 The *Altro Senso* Variant

This remake has also something really new, the *altro senso* variant. Partenio proposes, as an alternative to the phrasebook, the following complex method to get a plausible text from the nomenclator numbers.

The basic idea is to hide the information in the ligature (binding) between consecutive letters; Partenio defines *unite* the letters united with a continuous writing (without raising the pen from the paper) and *disgiunte* if there is no binding.

The way to get a number in the range 0..9 from these continuities and discontinuities in the handwriting is not so simple and Partenio conceives a complicated set of rules that require a bit of arithmetic. The rules are:

1. Every number begins with two letters *disgiunte* and ends with two letters *unite*. This rule defines the boundaries of the single numbers.
2. A letter *disgiunta* isolated on both sides get a score of 4.
3. A letter *disgiunta* on the left and *unita* on the right gets a score of 4 as well.
4. A letter *unita* with both adjacent letters get a score of 1.
5. A letter *unita* with its left letter, at the end of a number gets a score of 1
6. The resulting number is the sum of all scores from the beginning to the end, as defined above.
7. The first letter of a word inside a number is not computed.

Having this in mind you can use any phrase and write it using continuous or discontinuous writing in such a way as to get the numbers to hide. Using the first example given by Partenio, let's see how to get number 3 out of the word *amor*; one must write it so:

*a m or*

the first two letters **a**, **m** are *disgiunte* and by rule 2 score 4 each, while **o** and **r** are *unite*, but **o** is *disgiunta* on the left and by rule 3 has a score of 4, while the **r** is *unita* and by rule 5 scores 1. As a conclusion we have  $4+4+4+1=13$ . But being 13

out of the range 0..9, you have to subtract 10 and get 3. Here again, like in other ciphers, Partenio uses a modulo 10 arithmetic, to use the modern mathematical language.

But if one writes *il be* this way, at first look equivalent to the previous one:

*il be*

The score is now  $4+4+0+1=9$  because **b** is initial of a word inside the number, while the **o** of *amor* wasn't!

A question arises; can one obtain any digit with these rules?

Partenio addresses this problem, giving the two extreme cases: a) one cannot get 1 with a single letter, which scores 4, so you have to reach at least 11, that is 1 modulo 10. For instance you can get 1 with this sentence:

*il ben fa.*

Indeed this gives  $4+4+0+1+1+0+1=11$  that modulo 10 is 1 (the initial **b** and **fa** not computed, by rule 7).

Partenio at the end shows a complete example of his super-encryption; one has to write the message:

*Le cose sono accomodate.*<sup>10</sup>

Luckily the nomenclator has an entry for this, with cipher 393; now you can use the fake sentence *Illustrissi* to get 393, writing it as follows:

*illustrissi*

Indeed it is:

i l lu	$4+4+4+1=13$	3
s tr	$4+4+1=9$	9
i s si	$4+4+4+1=13$	3

In this case, 11 letters are needed for a 20 letters message, thus the fake message is shorter than the true message; using Bacon's cipher it would require 100 letters. But Partenio's example here is quite artificial, because the message uses a single cipher from the nomenclator, which is the best possible case. If one encrypts it using only letters and syllables, the worst case, he gets  $10 \times 3 = 30$  numbers which would require about 150 letters.

To conclude let's get all 10 digits:

<sup>10</sup>English: Things are settled.

i l ben fa	$4+4+0+1+1+0+1 = 11$	1
i l ben far	$4+4+0+1+1+0+1+1 = 12$	2
e s so	$4+4+4+1 = 13$	3
i de al	$4+4+1+4+1 = 14$	4
i de ale	$4+4+1+4+1+1 = 15$	5
i dei	$4+0+1+1 = 6$	6
i divi	$4+0+1+1+1 = 7$	7
i dieci	$4+0+1+1+1+1 = 8$	8
l ui	$4+4+1 = 9$	9
l oro	$4+4+1+1 = 10$	0

The trick requires great care in writing, to avoid ambiguities while deciphering; at the same time a gap too large may become suspicious to an expert's eye.

## 5.2 Conclusion about the Cipher

This 1606 version of the second cipher of 1592 is an improvement both because the nomenclator has been enlarged with many common phrases, and the phrasebook has been enlarged from nine to fifteen pieces.

The *altro senso* variant is rather puzzling; it is really ingenious in itself, but a bit too demanding, and Partenio seems to be struggling to solve the problem of getting numbers in the 0..9 range. The advantage is that the fake message can be shorter.

The whole cipher looks more a cryptographic *divertissement* than a cipher usable in the real world. No message using this cipher was found up to date, but of course such a *without suspicion* message would be very difficult to find.

## 6 Origins of the Cipher: Trithemius? Vigenère? Bacon?

An interesting problem is to find the sources, if any, of this cipher, and of the calligraphic variant. Were these ideas born from scratch? Or did Partenio stand on the shoulders of the giants who preceded him?

I found a few possible links, the first almost certain, the others more problematic.

Let's start with the first, the cipher known as *Ave Maria* abbot Johannes Trithemius<sup>11</sup>.

<sup>11</sup>Ioannes Trithemius (later spelled Johannes Trithemius, 1462-1516) was a German priest and abbot who wrote about cryptography and steganography but also astrology and occultism; his first book *Steganographia* was placed on the Index of prohibited books by the Catholic Church as heretical, the second *Polygraphia* containing the *Ave Maria* cipher and the *Recta Tabula*, was written in 1506-1508, and published in 1518 after his death.

### 6.1 Trithemius's *Ave Maria* Cipher

In his main cryptographic work *Libri Polygraphiae VI*<sup>12</sup> Trithemius presents two ciphers without suspicion (steganography) followed by four suspicious (cryptography).

Trithemius's best known cipher is the last one, the *Recta Tabula*, but here we are more interested to the cipher described in the first two books, *Liber I* and *Liber II*, best known as the *Ave Maria* cipher<sup>13</sup> cipher<sup>14</sup>.

The basic idea is to encrypt every letter of the plain text with a word taken from a list of 384 alphabets of 24 letters, published from page 107 to 298 of the book, every page having two columns with two alphabets (see the first pages in figure 5). The words of each column are roughly interchangeable, and written in order produce a plausible text; Trithemius in the *explanatio* of Liber 1, gives a simple example<sup>15</sup>: in case a malicious man asks to be recommended to a friend of yours, and you want to alert the friend of the danger, you can give the rascal a message so encrypted:

*Cave tibi ab isto viro, quia fur est, et nequam pessimus.*<sup>16</sup>

Using orderly the list of alphabets you substitute C with *Conditor*, A with *clemens*, V with *discernens*, E with *mundana*, T with *insinuet*, I with *expetentibus* ... and so on. At last you get a very long fake message, so beginning:

*Conditor clemens discernens mundana, insinuet expetentibus amoenitatem seraphicam [...]*

The message has the look of an innocuous religious sermon, and the rascal will bring it, without suspicion of his real content.

The cipher is very bulky, in this example it generates a fake text of ten lines for a single line of plain text, and has the defect that whoever knows the book could easily decipher the fake text, while to write a new fake book is a huge task. Indeed Trithemius was well aware of this and recommended to rewrite the book shuffling the word

<sup>12</sup>Six books of polygraphy (Trithemius, 1508).

<sup>13</sup>I don't know when and why this cipher received the name of *Ave Maria*; Trithemius and Vigenère do not use it. In Liber II there is the sequence of words *Ave Maria gratia plena* ..., maybe it comes from here.

<sup>14</sup>See also (Kahn, 1996), pp. 133-135 and (Schmeh, 2017).

<sup>15</sup>(Trithemius, 1508) p.55

<sup>16</sup>English: Beware of this man, because he is a thief, and the worst criminal

of every column. Not a light task, to rewrite and shuffle 384 pages!

Trithemius himself writes that one can get more comfortable ciphers renouncing the "without suspicion" condition; and the following ciphers do this up to the *Recta Tabula* that again proposes an ordered list of alphabets, this time encrypted with single alphabet letters shifted; *Polygraphia* ends with the simplest polyalphabetic cipher, opening the route to Vigenère's table.

Partenio's superencryption closely resembles this *Ave Maria* cipher of Trithemius. Indeed there are differences: Trithemius uses a 24 letter alphabet, Partenio reduces it to a 10 digits one; this should make things easier when trying to assemble plausible text binding together the single pieces. Trithemius has a 384 alphabets repertory, while Partenio has only 9 or 15, but of course it could be enlarged at will by the user.

Did Partenio know Trithemius's work? Among the papers kept in the Venetian Archives, Trithemius is repeatedly mentioned. Agostino Amadi in his treatise<sup>17</sup> ridicules this cipher writing:

*Il Tritemio abbate che tra sinonimi [...] con tanta fatica, tanto perdimento di tempo, tanto logoramento di carta [...] nascondeua breue et minima cosa.*<sup>18</sup>

Surely Partenio knew Amadi's treatise and maybe his goal was to improve Trithemius's idea, with less effort and less waste of time and paper; besides he was a notary used to write deeds in Latin, so he could read the book without any difficulty. So it is very likely that the first idea came to him from Trithemius.

## 6.2 The Cipher of Francis Bacon

The second possible link is with Bacon's cipher; Francis Bacon is best known as a philosopher and statesman but he gained a place in the history of cryptology also, because of this cipher.

In his book *De dignitate et augmentis scien-*

<sup>17</sup>This 700 handwritten pages treatise (Amadi, 1588) was recovered by the CX after Amadi's death in 1588, and is still kept in the Venetian Archives; the book in ten volumes was his textbook for teaching cryptography and cryptanalysis to the future deputies for ciphers.

<sup>18</sup>English: "Abbot Trithemius among synonymous [...] with so much effort, so much waste of time, so much wear of paper [...] was hiding a short and minimum thing"

*tiarum*<sup>19</sup> he presented this curious cipher<sup>20</sup> producing common language message, a message "without suspicion". He wrote to have conceived the cipher when he was young (*aduluscentuli*) in Paris, during his tour in Europe between 1576 and 1579.

The first step was a MASC cipher where single letters were encrypted with a five letter group using only two letters, **a** and **b**; the 24 letters of the XVII century English alphabet are so encrypted:

A	aaaaa	B	aaaab	C	aaaba	D	aaabb
E	aabaa	F	aabab	G	aabba	H	aabbb
I	abaaa	K	abaab	L	ababa	M	ababb
N	abbaa	O	abbab	P	abbba	Q	abbbb
R	baaaa	S	baaab	T	baaba	V	baabb
W	babaa	X	babab	Y	babba	Z	babbb

Nowadays we can say that using 0 and 1 instead of a and b, these are the binary numbers from 0 to 23. By the way, the binary notation was introduced by Leibniz in 1703.

Once a message is encoded this way you get a sequence of **a** and **b**. Bacon's idea is to print a generic text using two distinguishable fonts, e.g. serif and sans serif, the first for each **a**, the second for each **b**. If the two fonts are not very different in size and look, you get an innocuous message, and one can not guess it hides another secret message.

Of course an expert eye could notice the diverse fonts distributed in such a strange way, and suspect something ... and the cipher is no more without suspicion.

And, again, the message will be much longer than the plain text, here five times longer.

Partenio's *altro senso* variant closely resembles Bacon's cipher; instead of two different fonts, it uses the ligature vs. non ligature difference to encode the message; in either case, it is a font matter. Is it a mere coincidence? Here the relationship is much more unlikely than for the Trithemius's case. Indeed the English version of Bacon's book<sup>21</sup> was published in 1605, but had only a short chapter about ciphers, and no mention of this cipher, which was added to the Latin translation of 1624<sup>22</sup>, 18 years after Partenio's hand-

<sup>19</sup>The book was first published in English in 1605, with the title "Of Proficiency and Advancement of Learning Divine and Human" and later translated into Latin with the cited title; the English text had only a short chapter about ciphers, while in the Latin version he presented this cipher in detail.

<sup>20</sup>See first of all (Bacon, 1624) as the primary source and other books dealing with this cipher:(Fouche, 1939) p. 6, (Kahn, 1996), p. 882 or (Schmeh, 2017), p. 62.

<sup>21</sup>(Bacon, 1605).

<sup>22</sup>(Bacon, 1624).

book; so a link between Partenio and Bacon looks problematic. Maybe there was a common origin.

### 6.3 Vigenère

A possible common root is Vigenère and his treatise. There he proposed a 3 letters substitution cipher, where a letter say A can be substituted by a group of three letters  $a b c^{23}$ , while Bacon used only two letters. A few pages after, Vigenère writes that one can use a single letter in different fonts, without producing non suspicious texts, for example a very suspicious sequence of o and o<sup>24</sup>. Vigenère does not use a second step (super-encryption) here.

Vigenère in his treatise was rather skeptical about Trithemius and similar ciphers, writing<sup>25</sup>:

Mais cela est trop laborieux et bien rarement se peuvent rencontrer des mots, nompas seulement des syllabes bien propres, pour remplir la suite & le contexte de l'oraison, qu'on ne s'appercue de l'artifice [...]<sup>26</sup>

A few lines after, to show that anyway this artifice can be actually used, Vigenère reports that when he was in Venice in 1569, he learned that a similar cipher was proposed to the Venetian Baylo<sup>27</sup> by the physician Lorenzo Ventura to get around the bans by Sultan Selim II to write encrypted messages.

Indeed in the Venetian archives the dispatches of the Baylo in the years from 1566 and 1569 were mostly encrypted with a classical nomenclator, as usual, while one finds several dispatches having parts written using invisible inks<sup>28</sup>. Was this the way to evade Selim's prohibitions, as proposed by Ventura, who wrote a book on medicine and chemistry, not on cryptography? Did Vigenère misunderstand the whole affair? The question remains open, a letter written with steganographic methods is difficult to locate.

<sup>23</sup>See (Vigenere, 1586), ff. 200-201

<sup>24</sup>See (Vigenere, 1586) f. 243r

<sup>25</sup>(Vigenere, 1586), p. 182.

<sup>26</sup>English: But this is too demanding and very rarely can words be found, not only fitting syllables, to fit the text and the context of the prayer, without revealing the artifice.

<sup>27</sup>Baylo or Bailo was the name traditionally given to the Venetian ambassador in Constantinople.

<sup>28</sup>The Baylo, Giacomo Soranzo had a severe reproach from the CX for using lemon juice as an invisible ink, which was a very dangerous practice, since the expedient was also known to the Turks. But more sophisticated invisible inks were used by the Venetians. See (Preto, 1994), p. 281.

More interesting: did Vigenère have contact with Venetian cipher deputies that year? And did Bacon meet Vigenère in Paris during his journey a few years after? Again we are in the realm of conjectures.

## 7 Conclusion about the Origins

This cipher of Partenio is in no way revolutionary, and looks at the same time ingenious and problematic to use. Indeed it is the result of joining a classical nomenclator and a *Ave Maria* like superencryption, while the *altro senso* ligature vs. non ligature method was maybe his own invention with some possible some root in Vigenère's treatise or, much less likely, from Bacon.

What Partenio and Bacon have in common is a two step encryption, producing common language text, the first step being a substitution (cryptography), the second a kind of steganography.

So, we can call this cipher a cryptosteganographic one.

## 8 Can such a Cipher be used Today?

This cipher has many limits: slow and clumsy like other steganographic methods, it would require a much larger phrasebook (well more than 9 or 15 pieces of phrases), and a fastest way to encode the text.

As already stated above, for this reason steganography was largely neglected and left to amateurs. In 1939 Helen Fouché Gaines wrote at the end of her short chapter about steganography:<sup>29</sup>

Concealment cipher has, of course, the unique virtue of being able to convey message under circumstances which make it seem that no communication has passed [...] But we rather suspect that, for the end desired, invisible inks are more convenient and practical.

As we have seen above, invisible inks were used by Venetians, and apparently several messages went unnoticed.

But nowadays in the computer era, the above mentioned problems can be easily overcome. And steganography is again used, in upgraded forms. Secret messages or, worse, secret malicious software can be hidden in a graphic image using a

<sup>29</sup>(Fouche, 1939) p. 6

few pixel, very difficult to spot among millions, or even the Exif data of the jpeg format or other tricks. There are so many bits in an image!

So, why not to implement a Partenio like steganography software producing fake text hiding, without suspicion, secret messages?

Of course this is possible and rather easy to do, as it is the case for many others historical ciphers. Figure 6 and 7 show the output of a software designed for this purpose<sup>30</sup>. Moreover, it is possible to do much better, have a much larger phrasebook, even a Trithemius phrasebook can be stored in a few kilobytes, encrypt and decipher in a matter of seconds what in the past required hours.

Problem number one is to find a safe way to exchange the keys. In this case the nomenclator and the phrasebook are clumsy, huge if you make a Trithemius like phrasebook, but a modern database has room for much larger keys, and modern cryptographic methods like RSA may be used to exchange the key.

Problem number two is more serious; is it possible to implement a software that will produce absolutely plausible, enough long and non suspect texts?

Problem number three: does such a thing make sense, when we have already powerful tools to transmit message in a secret and safe way?

As for the *altro senso* variant, it seems madness, but of course it is possible using fonts making ligature possible, like the *Calligra* used for the above examples. And problem number three remains unchanged.

## 9 Acknowledgments

A special thank goes to Giovanni Caniato and all the other archivists of the *Archivio di Stato di Venezia* for assistance and help in recovering Partenio's papers, and to Antonio Giovanni Colombo for reviewing the English text.

## References

- Agostino Amadi. 1588. *Trattato delle cifre*. Digitized manuscript in ASVe, Inquisitori di Stato, Codice Amadi, Venezia.
- Francis Bacon. 1605-1901. *Of Proficiency and Advancement of Learning Divine and Human*. London.

<sup>30</sup>The software written in PHP/MySQL was useful also to test the cipher. It works fine, within the size limits mentioned above.

- Francis Bacon. 1624. *De dignitate et augmentis scientiarum*. London
- Paolo Bonavoglia. 2020. *The cifra delle caselle a super-encrypted XVI century cipher*. *Cryptologia*, Vol.44-1.
- Paolo Bonavoglia. 2019. *Hieronimo di Franceschi and Pietro Partenio, two unknown Venetian cryptologists*. Proceedings of the HistoCrypt 2019 Linköping University Electronic Press, Sweden.
- Helen Fouché Gaines. 1939, 1956. *Cryptanalysis a study of ciphers and their solution*. American Photographic Publishing Co. Dover, New York.
- David Kahn. 1996. *The Codebreakers*. Scribner, New York.
- Pietro Partenio. 1592, 1593. *Seven cipher offered to the Council of Ten*. Manuscript in ASVe, CX Raccordi 1.
- Pietro Partenio. 1606. *Handwritten booklet*. Manuscript in ASVe, CX chiavi e scontri di cifra, b.2, f.14.
- Luigi Pasini 1872, 2019. *Delle scritture in cifra usate nella Repubblica di Venezia*. Aracne, Roma, 2019.
- Paolo Preto 1994-1999. *I servizi segreti di Venezia*. EST, Milano, 1999.
- Klaus Schmeh. 2017. *Versteckte Botschaften*. Scribner, New York.
- Joannes Trithemius. 1508. *Libri Polygraphiae VI*. Argentorati (Strasbourg), 1613.
- Blaise de Vigenère. 1586. *Traictè des chiffres, ou Secrètes manières d'escrire*. Abel L'Angelier, Paris.



Seconda Cifra di Piero

Partimento di senso corrente.

25

<b>A</b>		160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
----------	--	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

Se ne fa l'infiammazione della prima cifra

[illegible]

Figure 2: Partenio’s 1606 three digits nomenclator. *ASVe CX Cifre, chiavi e scontri di cifra con studi successivi, busta 2 fasc. 14.*



1 Ser. <sup>mo</sup> Principe	1 No si marauiglia v. ser. <sup>mo</sup>	1 se non gli seruiro	1 Perche son certo	1 che quando
2 Ser. <sup>mo</sup> P. sig. col. <sup>mo</sup>	2 Non si dia marauiglia	2 se no la raguaglia	2 perche son certo	2 che quando io
3 Ser. <sup>mo</sup> P. sig. mio col. <sup>mo</sup>	3 Non resti marauigliata	3 se no ha auiso da me	3 perche tengo certo	3 che ogni fiata d'
4 Principe ser. <sup>mo</sup>	4 No restari marauigliata	4 se no gli do raguaglio	4 perche tengo p. certo	4 d' ogni uolta che
5 Principe ser. <sup>mo</sup> sig. col. <sup>mo</sup>	5 Non si altera	5 se non ha mie lettere	5 perche credo	5 che ogni fiata che io
6 Ser. <sup>mo</sup> P. et sig. col. <sup>mo</sup>	6 No babbia ammirazione	6 se no ha lettere mie	6 perche credo certo	6 d' ogni uolta che io
7 Principe ser. <sup>mo</sup> et sig. col. <sup>mo</sup>	7 No prenda ammirazione	7 se no riceue mie lre	7 perche e cosa certa	7 che se
8 P. ser. <sup>mo</sup> et sig. mio col. <sup>mo</sup>	8 Non prenda marauiglia	8 se no e auisata da me	8 perche tengo p. fermo	8 che se io
9 Ser. <sup>mo</sup> P. et mio sig. col. <sup>mo</sup>	9 No babbia alc. ammirat.	9 se no riceue lre mie	9 perche son sicuriss.	9 che mentre
10 Principe sig. mio col. <sup>mo</sup>	10 No babbia alc. amara.	10 se no e da me auisata	10 perche tengo p. cosa certa	10 che mentre io
1 gli seruiressi	1 con lre in cifra	1 sarebbe tutto squarciato	1 con dispiacer suo	
2 La raguagliassi	2 con lre in cifra	2 sarebbe tutto abingiato	2 con dispiacer suo	
3 La auissassi	3 con lre sospette	3 sarebbe tutto malmenato	3 con mala satisfat. sua	
4 gli dessi raguaglio	4 co lre scritte in cifra	4 sarebbe tutto dissipato	4 con assai dispiacer suo	
5 Volessi seruirogl.	5 co caratte. sospetto	5 andrebbe tutto a male	5 co poca satisfat. sua	
6 uollessi auissarla	6 co caratte. di cifra	6 capiterebbe tutto male	6 co non poco dispiacer suo	
7 uollessi raguagliarla	7 in caratte. di cifra	7 tutto sarebbe squarciato	7 con assai dispiacer suo	
8 uollessi darli ragu.	8 in caratte. no inteso	8 tutto sarebbe abingiato	8 co molto dispiacer suo	
9 uollessi darli auiso	9 in caratte. sospetto	9 tutto sarebbe malmenato	9 con alterazione sua	
10 gli seruiressi cose ali.	10 co lre non intese	10 tutto sarebbe dissipato	10 co qualto alterato sua.	

Figure 3: The phrasebook of the 1592 CCX cipher, ASVe CCX Raccordi 1

1. Il sig.	1. Ringrazio	1. Sua diuina Maesta	1. spero	1. E queste cose mie	1. Saranno in
2. Ecc. sig.	2. Ringraziando	2. Sua Maesta di uero	2. Credo	2. E queste mie cose	2. Tendarono in
3. Il sig. ecc. sig.	3. Rendendo grazie	3. La diuina Maesta	3. Son sicuro	3. E queste inuenture	3. Rinfurcorno in
4. Sig. ecc. sig.	4. Infirmita che rende	4. La diuina Maesta	4. Son certo	4. E queste finche	4. Rinfurcorno in
5. Sig. ecc. sig.	5. Rendendo grazie	5. La diuina Maesta	5. Tengo per certo	5. E queste mie finche	5. Rinfurcorno in
6. Sig. ecc. sig.	6. Io ringrazio	6. La diuina Maesta	6. Tengo per fermo	6. E queste finche mie	6. Rinfurcorno in
7. Sig. ecc. sig.	7. Io ringrazio molto	7. La diuina Maesta	7. Son certo	7. E questi sudori	7. Saranno adprate in
8. Sig. ecc. sig.	8. Rendendo grazie	8. La diuina Maesta	8. Sono sicuro	8. E questi miei sudori	8. Saranno adprate in
9. Sig. ecc. sig.	9. Io rendo grazie	9. La diuina Maesta	9. Sono certo	9. E questi miei sudori	9. Saranno adprate in
10. Sig. ecc. sig.	10. Io rendo grazie	10. La diuina Maesta	10. Sono certo	10. E questi miei sudori	10. Saranno adprate in
1. E si ha degnato	1. E si ha degnato	1. mi concedano liberta	1. Sua benedictio	1. a confusione	1. De suoi nemici
2. E si ha degnato	2. E si ha degnato	2. mi concedano liberta	2. Sua Maesta	2. a destrukcio	2. De suoi contrari
3. E si ha degnato	3. E si ha degnato	3. mi concedano liberta	3. Sua Maesta	3. con dolore	3. De suoi auersari
4. E si ha degnato	4. E si ha degnato	4. mi concedano liberta	4. Sua Maesta	4. con romore	4. De suoi emuli
5. E si ha degnato	5. E si ha degnato	5. mi concedano liberta	5. Sua Maesta	5. con dispiacer	5. De suoi nemici
6. E si ha degnato	6. E si ha degnato	6. mi concedano liberta	6. Sua Maesta	6. con dispiacer	6. De contrari suoi
7. E si ha degnato	7. E si ha degnato	7. mi concedano liberta	7. Sua Maesta	7. con dolore	7. Di ogni suo nemico
8. E si ha degnato	8. E si ha degnato	8. mi concedano liberta	8. Sua Maesta	8. al dispiacer	8. Di ogni suo contrari
9. E si ha degnato	9. E si ha degnato	9. mi concedano liberta	9. Sua Maesta	9. con gran dolore	9. Di auersari suoi
10. E si ha degnato	10. E si ha degnato	10. mi concedano liberta	10. Sua Maesta	10. con gran spaur	10. Di auersari suoi
1. Di amasione	1. Di amasione	1. Di amasione	1. Di amasione	1. Di amasione	1. Di amasione
2. Di amasione	2. Di amasione	2. Di amasione	2. Di amasione	2. Di amasione	2. Di amasione
3. Di amasione	3. Di amasione	3. Di amasione	3. Di amasione	3. Di amasione	3. Di amasione
4. Di amasione	4. Di amasione	4. Di amasione	4. Di amasione	4. Di amasione	4. Di amasione
5. Di amasione	5. Di amasione	5. Di amasione	5. Di amasione	5. Di amasione	5. Di amasione
6. Di amasione	6. Di amasione	6. Di amasione	6. Di amasione	6. Di amasione	6. Di amasione
7. Di amasione	7. Di amasione	7. Di amasione	7. Di amasione	7. Di amasione	7. Di amasione
8. Di amasione	8. Di amasione	8. Di amasione	8. Di amasione	8. Di amasione	8. Di amasione
9. Di amasione	9. Di amasione	9. Di amasione	9. Di amasione	9. Di amasione	9. Di amasione
10. Di amasione	10. Di amasione	10. Di amasione	10. Di amasione	10. Di amasione	10. Di amasione

Figure 4: The phrasebook of the 1606 booklet cipher. ASVe CX Cifre, chiavi e scontri di cifra con studi successivi, busta 2 fasc. 14.

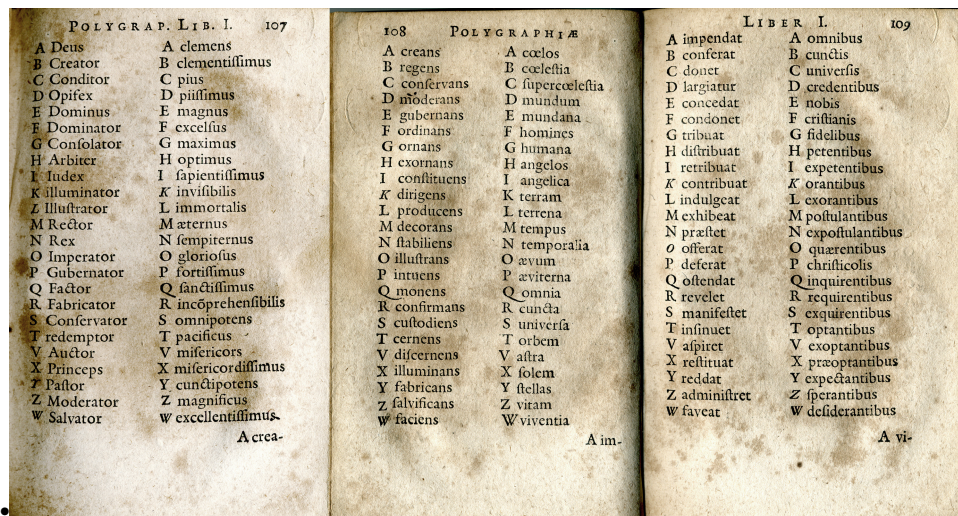


Figure 5: The first three pages of Trithemius's *Ave Maria* cipher.

<p>è uenuta noua che 315</p> <p>Ser.mo P. S.r mio col.mo non si marauiglia se non ha mie lettere</p>	<p>Re di Spagna 678</p> <p>perché credo certo che se gli dessi raguaglio</p>	<p>è risentito con pericolo di uita 312</p> <p>con lettere sospette sarebbe tutto squarciato con disgusto suo</p>
--	--	---

Figure 6: Partenio's example, encrypted by a software

Ser.mo P. S.r mio col.mo 3	non si marauiglia 1	se non ha mie lettere 5	perché credo certo 6	che se 7	gli dessi raguaglio 8	con lettere sospette 3	sarebbe tutto squarciato 1	con disgusto suo 2
315 è uenuta noua che			678 Re di Spagna			312 è risentito con pericolo di uita		

Figure 7: The same example deciphered by software

# Trithemius, Bellaso, Vigenère Origins of the Polyalphabetic Ciphers

Paolo Bonavoglia

Mathesis Venezia c/o Convitto-Liceo Marco Foscarini 4942 Venezia, Italy  
paolo.bonavoglia@mathesisvenezia.it

## Abstract

The purpose of this paper is to show how polyalphabetic ciphers developed, using primary sources, from Trithemius and Bellaso to Vigenère, including the recent discovery of the Bellaso 1552 zero cipher.

## 1 Primary Sources

Doing research using only primary sources is of course impossible, except for a few limited cases. A great number of mistakes, small or big, arise from using secondary sources; errors of transcription, translation, interpretation accumulate, migrate from book to book, even of the most authoritative authors, and are very hard to die.

I will try to use this method about the origin of poly-alphabetic ciphers. Nowadays Google Books, great libraries and others publish more and more digitized original books, making possible the use of primary sources without the burden of visiting remote libraries.

The first polyalphabetic cipher published in print (1518) is the one of abbot Trithemius, the *Recta Tabula* present in the *Libri Polygraphiae VI*.<sup>1</sup>

The second well known polyalphabetic cipher is the one of G.B. Bellaso published in Venice in 1553, which for the first time introduces what today is called a password or pass-phrase as the key. Bellaso writes in the preface this cipher was a remake of a 1552 cipher printed on leaflets; and it was one of these

leaflets the one I found in November 2018 in the State Archives of Venice.<sup>2</sup> See figure 1.

The best known polyalphabetic cipher remains the one of Blaise de Vigenère, published in 1586. Vigenère in his work mentions both Trithemius and Bellaso, and merges their ideas into his square table.

These ciphers are all basically square tables, as shown in the figure at the end of this paper (7).

## 2 Johannes Trithemius

Johannes Trithemius<sup>3</sup> in his book *Libri Polygraphiae VI*<sup>4</sup> introduced the *Recta Tabula*, Latin for square table, shown in figure 2. It uses a 24 letters alphabet, the ancient Latin alphabet extended with the three Greek letters **K**, **Y**, **Z** and the new letter **W**.<sup>5</sup>

One should use the first alphabet to encrypt the first letter, the second alphabet to encrypt the second letter and so on. So the same plaintext letter may be encrypted using different ciphertext letters, thus confusing frequency analysis.

<sup>1</sup>As a matter of fact Leon Battista Alberti had written a treatise on ciphers before 1470, proposing an encrypting disk and a few ways to use it, but the book was kept secret for about a century and published in Venice only in 1568. This is a common problem with many ciphers, kept secret for years or even centuries.

<sup>2</sup>See (Bonavoglia-2018)

<sup>3</sup>Johann Heidenberg or Johannes Zeller (1462-1516) was born in Tritenheim, a village that gave him the surname Trithemius.

<sup>4</sup>(Trithemius, 1518) The book was written between 1506 and 1508 and published in 1518, after his death.

<sup>5</sup>It may be appropriate to remark that the letter **W**, as a consonant variant of the Latin vowel **V**, (lowercase **u**) was introduced before the splitting of **V** in the vowel **U** and the consonant **V**. In English it is still known as *double u*.





Figure 1: Bellaso's cipher zero of 1552, discovered in December 2018 in Venice. No instructions were found. *Archivio di Stato di Venezia, Cifre, chiavi e scontri di cifra ... busta 3*. Any commercial use of this image forbidden.



Recta transpositionistabula.

a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z
b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a
c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b
d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c
e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d
f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e
g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f
h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g
i	k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h
k	l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i
l	m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k
m	n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l
n	o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m
o	p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n
p	q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o
q	r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p
r	s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q
s	t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r
t	u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s
u	x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t
x	y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u
y	z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x
z	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	x	y	z

In hac tabula literarum canonica siue recta tot ex uno & usuali nostro  
 latinarum literarum ipsarum permutationem seu transpositionem habes  
 alphabeti, quoniam ea per totum sunt monogrammata, uidelicet quater  
 & uigies quatuor & uiginti, quae faciunt in numero D. lxxvi. ac per to-  
 tidē multiplicata, paulo efficiunt minus q̄ quatuordecē milia.

o n̄

Hosted by Google

Figure 2: Trithemius *Recta Tabula*.

### 3 Giovan Battista Bellaso, 1552

This cipher had been printed in 1552, and was given to friends and other people. See figure 1. Basically his table uses 22 reciprocal alphabets, one for each letter, listed in a "vowels first" order. If one removes the superfluous first line of each list, a 22x22 square table, like the ones of Trithemius or Vigenère, remains, (see figure 7 at the end of this paper).

There were no instructions for using it, as confirmed by Bellaso himself in the preface to his 1553 paper, see next section.

### 4 Bellaso's Cipher of 1553

Indeed in the preface of his 1553 booklet Bellaso wrote<sup>6</sup>:

La onde à prieghi et consigli di  
 molti, & per mio minor fastidio,  
 mi sono risoluto di farla ristampare

<sup>6</sup>English: Therefore [answering] to prayers and advice of many people, and for my minor trouble, I resolved to have it reprinted for common satisfaction, and to the service of Christian Princes. And in addition to this I reduced it to the fourth part of what it was before, and to such brevity and ease, that a single glance includes it all, and they could also be memorized in the shortest period of time, [...]

AB	a	b	c	d	e	f	g	h	i	l	m
	n	o	p	q	r	f	t	u	x	y	z
CD	a	b	c	d	e	f	g	h	i	l	m
	t	u	x	y	z	n	o	p	q	r	f
EF	a	b	c	d	e	f	g	h	i	l	m
	z	n	o	p	q	r	f	t	u	x	y
GH	a	b	c	d	e	f	g	h	i	l	m
	f	t	u	x	y	z	n	o	p	q	r
IL	a	b	c	d	e	f	g	h	i	l	m
	y	z	n	o	p	q	r	f	t	u	x
MN	a	b	c	d	e	f	g	h	i	l	m
	r	f	t	u	x	y	z	n	o	p	q
OP	a	b	c	d	e	f	g	h	i	l	m
	x	y	z	n	o	p	q	r	f	t	u
QR	a	b	c	d	e	f	g	h	i	l	m
	q	r	f	t	u	x	y	z	n	o	p
ST	a	b	c	d	e	f	g	h	i	l	m
	p	q	r	f	t	u	x	y	z	n	o
VX	a	b	c	d	e	f	g	h	i	l	m
	u	x	y	z	n	o	p	q	r	f	t
YZ	a	b	c	d	e	f	g	h	i	l	m
	o	p	q	r	f	t	u	x	y	z	n

Digitized by Google

Figure 3: Bellaso 1553 cipher.

per commune sodisfattione, & serui-  
 gio de Principi Cristiani. & holla  
 oltre à cio ridotta alla quarta parte  
 di quello che era prima, & à tanta  
 breuità & ageuolezza, che una sola ri-  
 uolta d'occhio la comprende tutta, &  
 potrebbesi ancora in breuissimo spa-  
 tio di tempo imparare à mente, [...]

This cipher is clearly a remake of the 1552 cipher with letters coupled and ordered in the normal alphabetical order. This cipher has been known for centuries as *Porta's table*, a typical example of the mistakes arising from the use of secondary sources.<sup>7</sup>

### 5 Vigenère

Blaise de Vigenère in his famous 1586 *Traicté des chiffres* presented, at page 46r, a table using only the original 20 Latin alphabet (without the Greek **Y** and **Z**), honestly mentioning "*un certain Belasio*" as the inventor of this cipher. Really it is Bellaso's 1553 table, reduced to the 20 letters classical Latin alphabet, excluding **Y** and **Z**, Greek letters added at the end of the Latin alphabet. See figure 4.

After a few examples of use he writes<sup>8</sup>:

<sup>7</sup>See (Buonafalce, 2006)

<sup>8</sup>English: But all this can be done as well, even better, by the following table, in a way in which everything is reduced to one, taking the traverse capital letters which are at the front up, for the meaning we

A	a	b	c	d	e	f	g	h	i	l
B	m	n	o	p	q	r	s	t	u	x
C	a	b	c	d	e	f	g	h	i	l
D	x	m	n	o	p	q	r	s	t	u
E	a	b	c	d	e	f	g	h	i	l
F	u	x	m	n	o	p	q	r	s	t
G	a	b	c	d	e	f	g	h	i	l
H	t	u	x	m	n	o	p	q	r	s
I	a	b	c	d	e	f	g	h	i	l
L	s	t	u	x	m	n	o	p	q	r
M	a	b	c	d	e	f	g	h	i	l
N	r	s	t	u	x	m	n	o	p	q
O	a	b	c	d	e	f	g	h	i	l
P	q	r	s	t	u	x	m	n	o	p
Q	a	b	c	d	e	f	g	h	i	l
R	p	q	r	s	t	u	x	m	n	o
S	a	b	c	d	e	f	g	h	i	l
T	o	p	q	r	s	t	u	x	m	n
V	a	b	c	d	e	f	g	h	i	l
X	n	o	p	q	r	s	t	u	x	m

Digitized by Google

Figure 4: Bellaso's table adapted by Vigenère. *Traicté des chiffres*, p. 46r.

Mais tout cecy se peut practiquer aussi bien, voire trop mieux, par la table encore suiivante, combien que tout reuienne presque à vn, prenant les capitales trauersantes qui sont au front d'enhaut, pour le sens qu'on veut exprimer: & les perpendiculaires au costé gauche descendant en bas, au lieu de clefs. l'en ay mis icy deux renees: l'une de noir, l'autre de rouge, pour monstrier que les alphabets tant de l'escritur, que des clefs, se peuuent transposer & changer en tante de sortes qu'on voudra [...]<sup>9</sup>.

So Vigenère converts Bellaso's cipher into a Trithemius like square, using a key word; it is simpler to use than Bellaso's and safer than Trithemius's. You look for the letter of the plaintext ( $p$ ) among the column labels and the letter of the key ( $k$ ) among the row labels or viceversa, the operation is commutative. The cipher ( $c$ ) is anyway at the crossing of column and row.

want to express: and the perpendiculars to the left side descending downward, for the keys. I have put here two rows: one in black, the other in red, to show that the alphabets of the text, as well as those of the keys, can be shifted and changed in as many sorts as one wants [...]

<sup>9</sup>(Vigenere, 1587) p. 49v.

		O	P	Q	R	S	T	V	X	A	B	C	D	E	F	G	H	I	L	M	N
		E	F	G	H	I	L	M	N	O	P	Q	R	S	T	V	X	A	B	C	D
O	E	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	x
P	F	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	x	a
Q	G	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	x	a	b
R	H	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	x	a	b	c
S	I	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	x	a	b	c	d
T	L	f	g	h	i	l	m	n	o	p	q	r	s	t	u	x	a	b	c	d	e
V	M	g	h	i	l	m	n	o	p	q	r	s	t	u	x	a	b	c	d	e	f
X	N	h	i	l	m	n	o	p	q	r	s	t	u	x	a	b	c	d	e	f	g
A	O	i	l	m	n	o	p	q	r	s	t	u	x	a	b	c	d	e	f	g	h
B	P	l	m	n	o	p	q	r	s	t	u	x	a	b	c	d	e	f	g	h	i
C	Q	m	n	o	p	q	r	s	t	u	x	a	b	c	d	e	f	g	h	i	l
D	R	n	o	p	q	r	s	t	u	x	a	b	c	d	e	f	g	h	i	l	m
E	S	o	p	q	r	s	t	u	x	a	b	c	d	e	f	g	h	i	l	m	n
F	T	p	q	r	s	t	u	x	a	b	c	d	e	f	g	h	i	l	m	n	o
G	V	q	r	s	t	u	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p
H	X	r	s	t	u	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q
I	A	s	t	u	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r
L	B	t	u	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s
M	C	u	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t
N	D	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u

Digitized by Google

Figure 5: Vigenère's original table from his treatise. *Traicté des chiffres*, p. 51v.

The table has red headings and black headings. One can shift the alphabets of  $s$  steps; starting with the letter **E** the shift is of 4 steps.

Indeed it is a simplification, without the shifting, of the table in figure 6, the one that became popular as Vigenère's table:

Mathematically, assigning to every letter his ordinal number in the alphabet, stating from 0, the encoding procedure is a simple arithmetic addition modulo 20 (for a 20 letters alphabet).

$$c = p + k + s \mod 20$$

The introduction of the shifting improved the security only a bit, mathematically it just removes the constant  $s$  from the addition:

$$c = p + k \mod 20$$

Security depends mainly on the length of the key. The longer the key, the safer the cipher.

## 6 Conclusion

Figure 7 is the best summary of this paper, showing at a glance the evolution of these ciphers, here written in square table form for better comparison. The classic Vigenère table is a Trithemius like cipher, using a Bellaso's like keyword.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 6: Vigenère’s table as it is known today, adapted to the modern 26 letters alphabet.

## 7 Acknowledgments

A special thank to Giovanni Caniato and the other archivists of the *Archivio di Stato di Venezia* for assistance and help, to Augusto Buonafalce for advice and to Antonio Giovanni Colombo for reviewing the English text.

## References

- Paolo Bonavoglia. 2019. *Cryptologia Bellaso’s 1552 Cipher recovered in Venice*. Philadelphia, PA. DOI 10.1080/01611194.2019.1596181
- Augusto Buonafalce. 2006. *Cryptologia Bellaso’s Reciprocal Ciphers*. Philadelphia, PA. DOI 10.1080/01611190500383581
- G. B. Bellaso. 1553. *La cifra del sig. Giouan Battista Bellaso .... Venezia*.
- Joannes Trithemius. 1508. *Libri Polygraphiae VI*. Ioannis Haselbergi de Aia, 1518. Argentorati (Strasbourg), 1613.
- Blaise de Vigenère. 1587. *Traicté des chiffres, ou Secrètes manières d’escrire*. Abel L’Angelier, Paris 1586 1587



# A Web-Based Interactive Transcription Tool for Encrypted Manuscripts

Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés

Computer Vision Center  
Computer Science Department  
Universitat Autònoma de Barcelona  
{jchen,msouibgui,afornes}@cvc.uab.es

Beáta Megyesi

Dept. of Linguistics and Philology  
Uppsala University, Sweden  
beata.megyesi@lingfil.uu.se

## Abstract

Manual transcription of handwritten text is a time consuming task. In the case of encrypted manuscripts, the recognition is even more complex due to the huge variety of alphabets and symbol sets. To speed up and ease this process, we present a web-based tool aimed to (semi)-automatically transcribe the encrypted sources. The user uploads one or several images of the desired encrypted document(s) as input, and the system returns the transcription(s). This process is carried out in an interactive fashion with the user to obtain more accurate results. For discovering and testing, the developed web tool is freely available <sup>1</sup>.

## 1 Introduction

Nowadays, artificial intelligence and pattern recognition are playing an important role in historical manuscript processing and recognition. Some research projects with focus on digital paleography, including the transcription of historical manuscripts are, for example, HIMANIS (Stutzmann et al., 2017), Transkribus (Kahle et al., 2017), and *From Quill to Bytes* (q2b, 2013). For the case of encrypted historical manuscripts analysis, which constitute the main subject of this paper, the project DECRYPT (Megyesi et al., 2020) is joining the expertise in computer vision, computational linguistics, philology, cryptanalysis and history for the aim of making advances in historical cryptology.

The first step toward decrypting a handwritten ciphertext is transcription. Intuitively speaking, the transcription could be done manually

but it turns out to be a time-consuming, error-prone, and expensive task (Piotrowski, 2012). During the last decade, several handwritten text recognition (HTR) methods have been developed and applied successfully to historical handwritten sources, allowing (semi-)automatic transcription (Kahle et al., 2017; Romero et al., 2017). Alternative approaches use word spotting (Santoro et al., 2017), speech recognition (Granell et al., 2018) or even gamification (Chen et al., 2018) for speeding up the manual transcription. However, all these tools have been developed to only deal with known scripts (e.g. Roman alphabet). Indeed, the transcription of encrypted sources is more complicated as they often include symbols that are taken from a wide range of alphabets and symbol sets. For a more generic and flexible transcription within and across ciphers, the use of generic annotation tools such as Alethea (Clausner et al., 2011) or Pixlabeler (Saund et al., 2009) could be preferable. But, the annotation process through these tools is fully manual, leading to a huge cost in term of time especially for encrypted manuscripts with unknown symbol sets. Therefore, semi-automatic image processing tools would be the suitable solution to this kind of applications.

In this paper, we present a tool for transcription of encrypted sources consisting of various symbols sets. The tool processes document images (e.g. scanned images of manuscripts) and outputs the corresponding transcription. The system interacts with the user at certain steps for a more accurate transcription (in a semi-automatic fashion). Users could be paleographers, cryptologists, archive workers, etc. We start by briefly describing previous efforts on (semi-)automatic transcription of ciphers, and then present our interactive tool.

<sup>1</sup><https://cl.lingfil.uu.se/decode/transcription/>



## 2 Automatic Transcription of Encrypted Sources

The main challenge in HTR is to locate and segment the actual text parts into paragraphs, lines, and individual symbols (glyphs). In addition, the system shall identify the various allographs (variants) of each symbol type (graphem). The system shall also be able to determine the various elements of a graphem, such as dots and commas, and leave out unintentional ink spots, bleed-through, or marks from a damaged paper or parchment. In a fully automatic system, computers handle the entire process in one step, while in a semi-automatic system the user can interact with the system to improve the result during the transcription or as a post-processing step to correct the output of an automatic process.

Experiments on automatic transcription by image processing have been performed on numeric cipher sequences (Fornés et al., 2017) and a wide range of glyphs belonging to alchemic and Zodiac signs, digits, and Roman and Greek letters (Baró et al., 2019). Preliminary results show that image processing can be used as base for transcription followed by a post-processing step with user validation and correction. Even though image processing techniques need to be trained on individual hand-writings to reach high(er) accuracy, unsupervised techniques (i.e. no labelled data is required to train) can also be used for speeding up the transcription. In addition, they might be of great help to identify the symbol set represented in the manuscript and to make clear distinctions between symbols, hence can be used as a support tool for the transcriber.

## 3 Interactive Transcription Tool

Our interactive transcription tool is generic in the sense that it should be applicable to any symbol sets, and it does not need any labelled data to train the image processing algorithms. The tool consists of three main steps, as illustrated in Figure 1. First, the input cipher images are segmented into lines and symbols. Then these symbols are clustered (grouped) according to their shape similarity. Finally, the transcription is performed, obtaining the final transcribed cipher-

text. Executing these stages in an automatic way leads to the transcription of a given cipher image. But, since the efficacy of each step highly depends on the correctness of the previous step output, it is preferable to use the tool in a semi-automatic way. In other words, if the user intervenes in each stage to validate or correct the intermediate results, then more accurate transcription can be obtained. In what follows, a detailed description of those steps is provided.

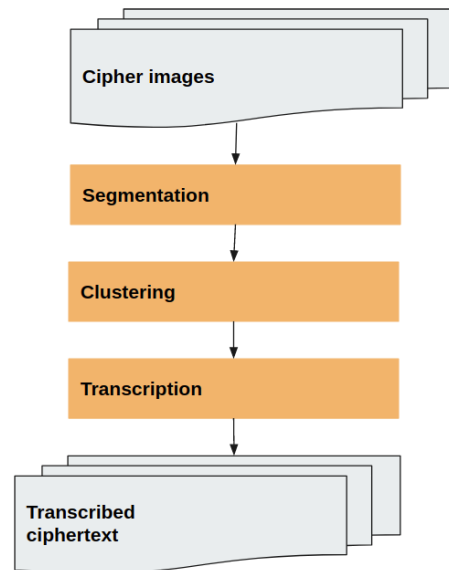


Figure 1: The architecture of the Interactive Transcription Tool.

### 3.1 Image Upload

First, the user uploads the image(s) into the tool. The system accepts PNG, JPEG or TIFF image file formats. Since the transcription accuracy depends on the images quality, we recommend to use colored images of high resolution (e.g. 300-600 dpi) as stated in (van Dormolen, 2019). This is recommended as well in ISO/TS 19264-1:2017 technical specification for cultural heritage imaging, even though the tool accepts low resolution images as well. It is to note that the image processing algorithms are based on the analysis of the symbols shapes. Thus, the document images should be selected from the same manuscript with the same symbol set and handwriting style to obtain a more reliable transcription. In this stage, the system creates a first JSON-file, it will be used to store all the intermediate results that will be obtained during the

different stages. This file will be sent to the user after each subsequent step of the transcription process.

### 3.2 Segmentation

The first step of our unsupervised transcription pipeline consists of segmenting the document image(s) into isolated symbols by creating bounding boxes for each symbol to be transcribed. Although the user can manually segment all symbols using our tool, it is a time consuming task. Hence, the optimal choice is to request an automatic segmentation and manually validate the results. The segmentation method consists of applying horizontal projections to detect the text lines, connected components to segment the symbols, and grouping to obtain the final bounding boxes of each symbol. An example of the automatic segmentation obtained can be seen in Figure 2.

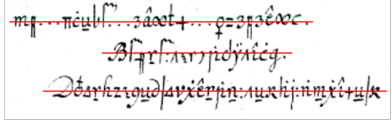
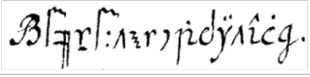
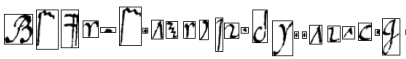
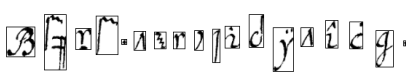
Horizontal Projection	
Segmented Line	
Segmented Symbols	
Symbol Grouping	

Figure 2: The stages to segment a cipher document into isolated symbols by the tool.

Although the segmentation algorithm can run using the default options, our interface provides some advanced options as illustrated in Figure 3, which are very useful for trained and experienced users when applying the automatic segmentation. These advanced options are:

- **Symbol size:** Big/Small. This value is used to inform on the size of the symbols with respect to the page. For example, the Copiale cipher (Knight et al., 2011) contains small symbols regarding the pages, whereas the Borg cipher (Aldarrab et al., 2017) contains big symbols in the pages.

- **Binarize image:** Yes/No. The user can chose whether to binarize the image or not. Because our current method works only on binary images, the user will receive an error if it is set "No". This option is added to guarantee scalability, since we are planning to add other segmentation methods to work on colored images as well.
- **Minimum line distance:** A number (in pixels) indicates the minimum distance between lines. Example: In the Copiale cipher, most lines have 120 pixels of separation.
- **Lines threshold:** it is a decimal/float number between 0 and 1. This value is used to state that only those lines with an amplitude higher than this threshold will be detected (this acts as a line filter).
- **Max. distance symbols:** This number (in pixels) indicates the maximum distance between symbols. This parameter is useful when grouping symbols that contain diacritics, super- och subscripts (e.g. dots or accents like á or ÿ). When the segmentation is based on connected components, these small elements are separated. For this reason, the system tends to group nearby symbols, i.e. symbols that are closer to the given threshold distance.
- **Min. symbol size:** This number (in pixels) indicates the minimum symbol size that could be found in the manuscript. This is used to filter components that are smaller than this size, which usually corresponds to background noise in historical manuscripts.

When the segmentation process ends, the user will receive (in their indicated email) a JSON file containing the results of the segmentation step. To visualize these results, the user should upload the JSON file and the cipher image to the web tool. Figure 4 shows an example of the output from the segmentation part.

Although the user can apply the segmentation algorithm using different setups (i.e. different values in the advanced options interface), it is difficult to obtain a perfect segmentation with an unsupervised segmentation method. The

Figure 3: The interface for the segmentation request, showing the advanced options.

main reason is that the segmentation algorithm is generic, so it has no information on the type of symbol set used in the encrypted source. Moreover, most encrypted manuscripts use a cursive writing, so touching and overlapping symbols are frequent, which make the segmentation even harder. In this stage, the user interaction is highly recommended, so that the clustering stage can be more efficient and less error-prone. Therefore, the tool allows the user to verify and manually correct any segmentation errors. Figure 5 shows and example of correcting a wrong segmentation. It is to note that the users cannot only delete or modify the bounding boxes, but they can also create new ones for any symbol missed by the automatic segmentation.

### 3.3 Clustering

Once the user obtain the set of isolated symbols (assumed to be correctly segmented), they can proceed to the clustering. Clustering means grouping visually similar symbols in different sets, called clusters. Our tool applies the hierarchical K-Means algorithm for clustering (Arai and Barakbah, 2007). As advanced setting, the

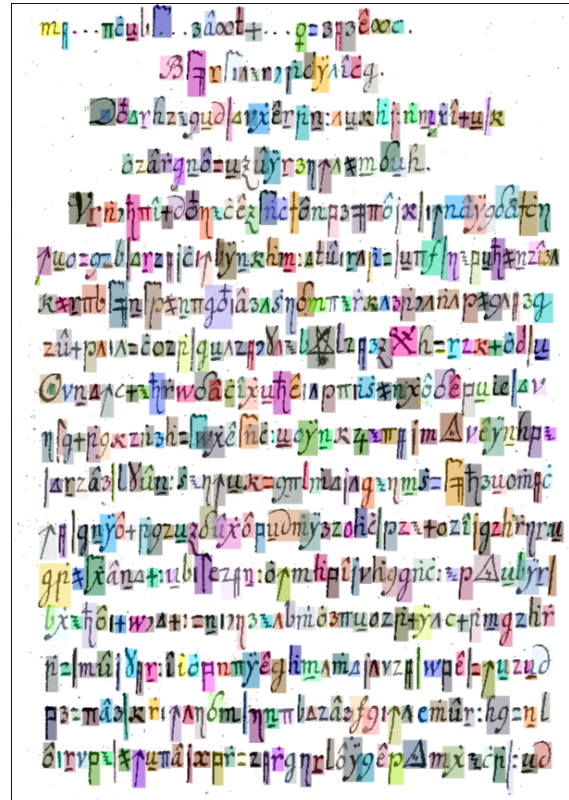


Figure 4: Visualization of the bounding boxes after the segmentation step.

user can define the minimum number of symbols that could be assigned to one cluster, called the Min. cluster images. The K-means algorithm starts by assuming that all the symbols are belonging to a single cluster, then, splitting it recursively until the clusters are no more divisible or when reaching the minimum amount of images per cluster. Figure 6 shows the clustering request interface.

Similar to the segmentation step, the user will receive the results of the clustering via e-mail. The user can visualize the clusters by uploading the received JSON file as shown in Figure 7. The tool bar on the right hand side called "Clusters" shows all the clusters provided by the K-means. The user can press the 'eye' icon to visualize the symbols belonging to each cluster. Figure 8 illustrates the symbols (instances) within a specific cluster.

In the ideal case, each cluster should contain instances from the same symbol. However, there is a high degree of visual similarity between the different symbols in many encrypted sources. As a result, some clusters can contain instances

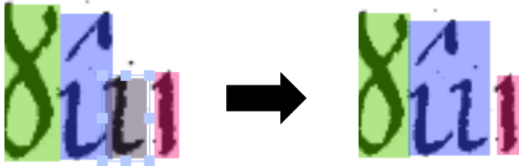


Figure 5: An example of correcting an over-segmented symbol. The grey bounding box must be merged to the previous symbol marked in blue.

Figure 6: Clustering request, showing the advanced options.

from different, although similar symbols. Thus, our tool allows the user to correct errors in the clusters. The user can clean a cluster by removing those symbols that do not belong to that cluster. An illustrative example can be seen in Figure 9.

After cleaning the clusters, the removed symbols remain unlabelled, i.e. not assigned to any cluster. The tool also allows the user to create new clusters, assign symbols to clusters, and

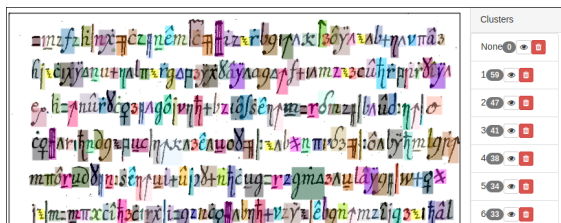


Figure 7: On the right, the system shows the clusters (i.e. group of symbols) obtained by the K-Means algorithm.

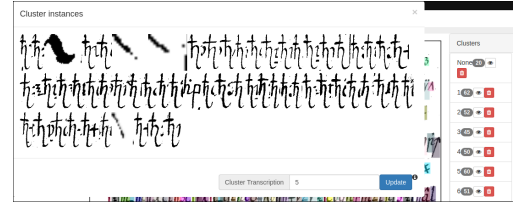


Figure 8: Example of one cluster after the label propagation step.

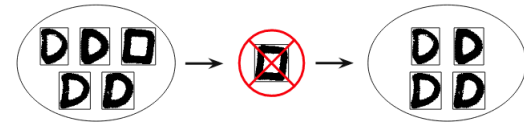


Figure 9: An example of cleaning a cluster: the user removes the symbol that does not belong to this cluster.

change the obtained clusters for the symbols. Cleaning the clusters facilitates the subsequent label propagation step, where symbols will be assigned to the most similar cluster.

### 3.4 Transcription

After the clustering step, the user can request the actual transcription where a label is assigned to each symbol according to the label of cluster the symbol belongs to. We call this process label propagation. The objective is to propagate the label of the clusters to the unlabeled symbols. The setup of the label propagation request has two options as illustrated in Figure 10:

- **Seeds number:** The number of the most populated clusters that will be used as seeds to propagate labels. This number should be at least equal to the alphabet size (if it is known). After setting the seeds number, the user can visualize the selected clusters in the cluster bar tool. The default value of seed numbers is 10 due to many ciphertext containing digits only (0-9).
- **Change class threshold:** A value between 0 and 1 determines how easy is to propagate a label through the instances. If the value is close to 0, the propagation will be more stable (less changeability), but it can lead to poor results when the user is transcribing few pages. Contrary, if the value is close to 1, it will make the propagation

Figure 10: Label propagation request, showing the advanced options.

unstable (high changeability) which leads sometimes to propagation of wrong labels.

The label propagation determines the final clusters and assigns the labels. The output is the sets of instances in each cluster, as shown in Figure 8.

At this moment, the only user intervention consists in assigning the desired transcription label to each cluster as shown in Figure 11. All the symbols in the cluster will be transcribed with the label assigned to the particular cluster. Note, however, that each symbol has a value between 0 and 1, representing the degree of belonging to this specific cluster. This means that if a symbol has a low value, the system is not confident in labelling the correct transcription. Therefore, the recommendation is to manually transcribe symbols with a low value to increase transcription correctness.

There is a trade-off between transcription correctness (precision) and transcription completeness (recall). As illustrated in Figure 12, a low transcription confidence threshold leads to more complete transcriptions. On the other hand, this leads to a higher possibility of errors. Contrary, a high confidence threshold means that only symbols with a high confidence value will be transcribed, whereas the rest will lack correct transcription. These non-transcribed symbols ap-

pear as "NONE" (or '\*'') in the transcription file, and the user shall dedicate more time to manually transcribe those symbols. In order to make a fewer intervention with higher accuracy, we tried to balance this by choosing the threshold confidence to be 0,5. As the final step, the user can download the obtained transcription using the download request with various types of output formats (e.g. text, XML, JSON), see Figure 13.

Figure 13: The downloading interface, where the user can select different kind of output files.

## 4 Conclusion

We presented a tool serving as an aid for faster and more accurate transcription of encrypted sources with various cipher text alphabets. The transcription system segments the lines and then suggests the segmentation of each individual symbol, which could be corrected by the user. Then, the segmented symbols are clustered into groups on the basis of similarity measures and the symbols in the same cluster receive the same transcription. The user can edit the suggestions given by the system in each step, correct the output, and upload a new, improved versions for further processing.

To the best of our knowledge, there is no similar tool that allows for the (semi)-automatic transcription of manuscripts with various alphabets and scripts. We hope that the ITT tool will be useful for the transcription of the historical and encrypted sources. The tool is under development and we plan to add more image processing techniques in the different transcription steps to



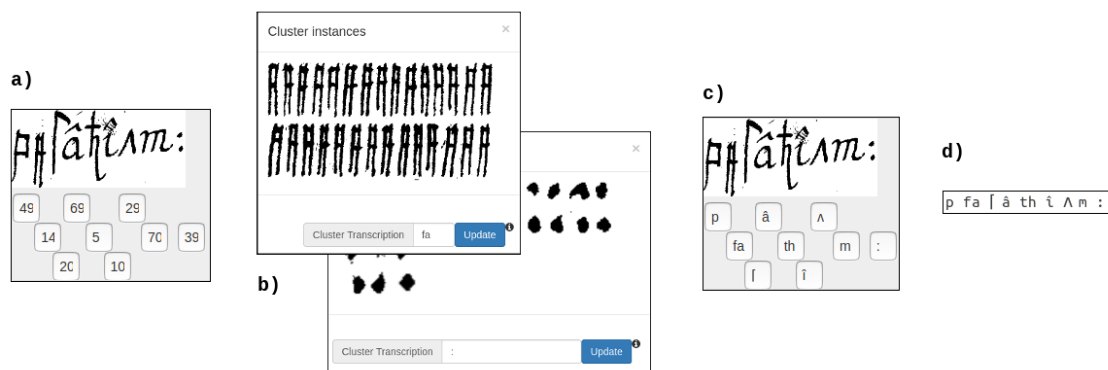


Figure 11: Transcription step. a) Line transcription using default cluster labels (numbers). b) The user changes the cluster labels to the desired transcription. c) Line transcription using the desired transcription. d) A text file with the line transcription.

Reconstructed line		q	w	1	w	v	x	d	1
Ground-truth		q	w	1	w	v	x	d	1
Our method Assigned code	Thr 0.4	q	w	1	w	*	x	d	1
	Thr 0.6	q	w	*	w	*	x	d	1
	Thr 0.8	q	w	*	w	*	*	*	1

Figure 12: In the transcription phase, by changing the transcription threshold, the symbols with lower confidence than the given threshold will be transcribed as '\*'. The symbols with lower confidence are marked with '\*' in the table.

enhance the accuracy and reduce the user intervention.

## Acknowledgments

This work has been partially supported by the Swedish Research Council, grant 2018-06074: *DECRYPT - Decryption of historical manuscripts*, the Spanish project RTI2018-095645-B-C21, the Ramon y Cajal Fellowship RYC-2014-16831 and the CERCA Program / Generalitat de Catalunya.

## References

Nada Aldarrab, Kevin Knight, and Beáta Megyesi. 2017. The Borg Cipher. <https://cl.lingfil.uu.se/ bea/borg>. Accessed: 2020-01-31.

Kohei Arai and Ali Ridho Barakbah. 2007. Hierarchical K-means: An Algorithm for Centroids Initialization for K-means. *Reports of the Faculty of*

*Science and Engineering, Saga University*, 36:25–31.

Arnau Baró, Jialuo Chen, Alicia Fornés, and Beáta Megyesi. 2019. Towards a Generic Unsupervised Method for Transcription of Encoded Manuscripts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage (DATECH)*, pages 73–78.

Jialuo Chen, Pau Riba, Alicia Fornés, Joan Mas, Josep Lladós, and Joana Maria Pujadas-Mora. 2018. Word-Hunter: A Gamesourcing Experience to Validate the Transcription of Historical Manuscripts. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 528–533. IEEE.

Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. 2011. Aletheia – An Advanced Document Layout and Text Ground-Truthing System for Production Environments. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 48–52. IEEE.

Alicia Fornés, Beáta Megyesi, and Joan Mas. 2017. Transcription of Encoded Manuscripts with Image Processing Techniques. In *Digital Humanities*.

Emilio Granell, Verónica Romero, and Carlos D. Martínez-Hinarejos. 2018. Multimodality, Interactivity, and Crowdsourcing for Document Transcription. *Computational Intelligence*.

Philip Kahle, Sebastian Colutto, Gunter Hackl, and Gunter Muhlberger. 2017. Transkribus – A Service Platform for Transcription, Recognition and Retrieval of Historical Documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 19–24.

Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. The Copiale Cipher. In *Invited talk at*

*ACL Workshop on Building and Using Comparable Corpora (BUCC)*. Association for Computational Linguistics.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of Historical Manuscripts: The DECRYPT Project. *Cryptologia*, 0(0):1–15.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Morgan Claypool Publishers.

q2b. 2013. q2b – From Quill to Bytes. <https://www.it.uu.se/research/project/q2b?lang=sv>. Accessed: 2020-04-21.

Verónica Romero, Vicente Bosch, Celio Hernández-Tornero, Enrique Vidal, and Joan Andreu Sánchez. 2017. A Historical Document Handwriting Transcription End-to-end System. In *8th Iberian Conference on Pattern Recognition and Image Analysis*, pages 149–157. Springer International Publishing.

Adolfo Santoro, Claudio De Stefano, and Angelo Marcelli. 2017. Assisted Transcription of Historical Documents by Keyword Spotting: A Performance Model. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 971–976.

Eric Saund, Jing Lin, and Prateek Sarkar. 2009. Pixlabeler: User Interface for Pixel-Level Labeling of Elements in Document Images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 646–650. IEEE.

Dominique Stutzmann, Jean-François Moufflet, and Sbastien Hamel. 2017. La Recherche en Texte dans les Sources Manuscrites Mdiévales : Enjeux et Perspectives du Projet HIMANIS pour L’édition Électronique. *Médiévales*, 73:67–96.

Hans van Dormolen. 2019. Metamorfoze Preservation Imaging Guidelines, version 2.0. In *Archiving Conference*, pages 9–11.

# The Auxiliary Devices of OKW/Chi

Carola Dahlke

Deutsches Museum, Germany

c.dahlke@deutsches-museum.de

## Abstract

Between 1942 and 1945, section IVb of OKW/Chi designed mechanical and electro-mechanical devices to statistically evaluate intercepted encrypted messages. By the help of TICOM protocols and a never published dissertation draft written by Willi Jensen in 1955, an overview of the cryptanalytic devices is given. Most probably all the equipment was destroyed at the end of the war. Only dredged-up remnants of one type of equipment could be recovered by German divers.

## 1 Hindered Research on German Signal Intelligence

Thanks to the recent declassification and publication of many TICOM files (Target Intelligence Committee interrogation protocols and summaries), it has become possible to take a closer look at the German side of cryptanalysis during the Second World War. As Weierud & Zabell (2019) already listed in detail, there are three main difficulties that complicate research on German signal intelligence: First of all, Germany lost the war, and destroyed almost all relevant documents and equipment. Even after the 1970s, when the allied nations finally began to talk about their signal intelligence achievements, German cryptanalysts kept their experiences in the Second World War secret until their death. Only a handful of papers exist which, although not published, have nevertheless been written by German cryptanalysts, and were given to archives, libraries or universities for safekeeping (e.g. Hüttenhain 1970, Jensen 1955).

Apart from the lack of sources and remaining artefacts, historic research is hampered by the fact that Germany had not only one but eight

different intelligence sections<sup>1</sup> during the Second World War, some of which worked completely independently of each other:

- **OKW/Chi:** Cipher department of the High Command of the Armed Forces (“Chiffrierstelle des Oberkommandos der Wehrmacht”).
- **In 7/IV, In 7/VI:** Inspectorate 7 group 4 and 6 (“Inspektion 7 Gruppe 4 und 6”), the cipher department of the army; in 1944 reorganized and combined as **OKH/GdNA:** Signal intelligence agency of the High Command of the army (“General der Nachrichtenaufklärung des Oberkommandos des Heeres”).
- **OKM/B-Dienst:** Intelligence service of the Naval High Command (Beobachtungsdienst der deutschen Kriegsmarine).
- **O.b.L/Chi:** Signal intelligence agency of the German Air Force (“Chiffrierstelle, Chi-Stelle“ des Oberbefehlshabers der Luftwaffe“); in 1944, reorganized and renamed in **OKL/LN Abt 350:** Aerial news division 350 of the Airforce High Command (“Luftnachrichten Abteilung 350 des Oberkommandos der Luftwaffe“).
- **RLM/FA:** Research office of the State Ministry of Aviation (“Forschungsamt des Reichsluftfahrtministeriums”), i.e. the cryptological service of the Nazi party.
- **AA/Pers Z S & Pers Z Chi:** Cipher department of the Foreign Office (Chiffrierstelle des Auswärtigen Amts).

<sup>1</sup> For comprehensive descriptions, see e.g. Mowry (1989); Weierud & Zabell (2019); EASI Vol 2-7.

- **Abwehr:** Secret service of the military; part of the High Command of the Armed Forces OKW until 1944, then reorganized and integrated into the espionage section of the SS.
- **RSHA/Amt IV E:** Secret service of the Reich Security Administration, i.e. of the SS (“Abwehr des Reichssicherheits-hauptamts”) until 1944; then reorganized and combined with the Abwehr of OKW into **RSHA/Amt VI**.

This polycratic appearing coexistence of competing institutions with similar competences was typical for the regime of National Socialism. Attempts had been made to create a central intelligence office, but were not realized (see e.g. TICOM DF-187, p. 14<sup>2</sup>; Bauer 1997, p. 31). Instead, the consisting signal intelligence offices were partly reorganized, e.g. as a result of the coup attempt on the 20th of July 1944. This means that the few historical sources that are available can often only be assigned to one of the different departments, or to a person who may have changed affiliation or departments several times during the war.

This study will focus on the cipher department of the Oberkommando der Wehrmacht (OKW/Chi), and on the deciphering devices that have been designed and used there.

## 2 OKW/Chi

The OKW/Chi had originally been the cipher office of the Reich War Ministry. It was renamed the cipher office of the OKW in 1938 with about 30 staff members initially, but grew up to 250 in 1942, and sank to only 120 persons by the end of war (TICOM I-206, p. 9). A description of the organization of OKW/Chi can be found e.g. in EASI Vol 3, in TICOM I-39, in Rezabek (2013), and very detailed information on the mathematical staff is summarized by Weierud & Zabell (2019).

2 According to TICOM interrogation protocols, attempts were made by Wilhelm Fenner, Franz Thiele (who was hanged after the coup attempt on Adolf Hitler on the 20th of July 1944) and brigade commander Schlieberg, to set up a joint cryptanalysis agency. The plan was to take the best analysts from all the agencies that had existed so far and put them under Fenner's care.

OKW/Chi was two-fold: One part of the organization was mainly concerned with monitoring the broadcast or news of enemy and neutral states. The other part dealt with signal intelligence. The cipher telegrams of about 30 countries were watched by OKW/Chi, and the task was to decipher only important diplomatic letters, i.e. telegraphic communications of diplomats, military attachés, government and economic authorities etc. (EASI, Vol 3, p. 15). According to the interrogation papers of Wilhelm Fenner, who was in charge of the OKW/Chi's cryptanalysis sections IV and V, the successful years of OKW/Chi were between autumn 1939 and autumn 1943. His team deciphered about 100 messages per day, sometimes several pages long (TICOM DF-187A, p. 16), although never attaining its full potential due to bombing attacks, broken furniture, dirt, cold and chronic undernourishment of the staff (TICOM I-206, p. 9).

In general, it can be said that OKW/Chi did not achieve great successes, but at least constantly managed many minor decipherments (EASI Vol 3, p. 55). The OKW/Chi's cryptanalytic successes are e.g. mentioned in TICOM I-31, pp. 5ff, and are summarized in EASI Vol 3, chapter IV.

In 1944, the OKW/Chi (apart from its archive<sup>3</sup>) was transferred from Berlin to Halle/Saale, where it continued its work until April 1945. Dr. Buggisch stated (TICOM I-176, p. 12) that all OKW/Chi machinery was taken to Halle, too.

On the 13<sup>th</sup> of April 1945, the remaining staff<sup>4</sup> of OKW/Chi took a train from Halle to Werfen/

3 The archive of the OKW/Chi went to the intercept station at Lauf, and remained there until spring 1945. On the 10<sup>th</sup> of April 1945, the Lauf station moved south to the lake Schliersee, where the staff dumped about nine-tenth of its equipment and the complete OKW/Chi archive into the lake (EASI Vol 3, p. 34). The boxes with the archives were recovered shortly after by the TICOM Team 5 (TICOM Team 5, Rezabek 2013), kept classified until 2013 and is now available at the Politisches Archiv in Berlin.

4 In the end of the war, parts of the OKW/Chi leadership, namely Mettig, Kettler, Dr. Hüttenhain and Fricke, travelled to the north of Germany (EASI Vol 3, p. 34+35). Please note: neither the dates of the disintegration of OKW/Chi nor the accounts about the changes in the organization of OKW/Chi were consistent in the

Salzach in Austria, to join with the “General der Nachrichtenaufklärung Süd”, i.e. the Southern cipher department of the Oberkommando of the Heer (OKH/GdNA) – one of the other pendants of competing intelligence offices mentioned before. OKW/Chi was disbanded that day. Fenner stated that since the American invasion was expected, all material was set on fire or thrown into the river Salzach (TICOM DF-187, p. 14).

## 2.1 Sub-section IVb

Under the head of Dr. Hüttenhain a special OKW/Chi subsection IVb was installed in 1942 to develop cryptanalytic machinery. IVb consisted of 28 staff members, i.e. two graduate engineers, three working engineers and 25 mechanics. The main idea of the mechanical devices was “to replace the speed of fingers in statistical operations” (Fenner, TICOM DF-187A, p. 20).

In general, Hollerith machines were used whenever possible, but so-called “Hilfsgeräte” (auxiliary devices; in the TICOM protocols they are entitled as rapid analytical machinery) were developed for special cryptanalytic purposes.

Fenner stated that these devices were mainly experimental models, and the technical possibilities could not be exhausted (TICOM DF-187, p. 15). Nevertheless, the machines that were developed in this section were mentioned as an outstanding achievement of OKW/Chi in the TICOM reports (EASI Vol 3, p. 72). As well, TICOM documents refer to the visit of an Italian cryptanalyst Augusto Bigi, who saw the OKW/Chi machines in 1942 and was impressed (see EASI Vol 3, p. 73 & TICOM IF-1517, pp. 14-15).

## 2.2 A Dissertation Never Published

Willi Jensen, a freshly graduated engineer, born in Kiel, was among the twenty-eight members of subsection IVb, under the command of the telecommunication engineer Mr. Rotscheidt (formerly with Siemens).

Ten years after the end of the war, Willi Jensen submitted a dissertation at the Technical

University of Munich with the title “Hilfsgeräte der Kryptografie” (auxiliary devices of cryptography, Jensen 1955). According to Bauer (2009, p. 388), the professor in question did not feel responsible, and the work remained unevaluated. We cannot be sure, but presumably, the professor in question was the mathematician Professor Robert Sauer, in whose estate at the Technical University of Munich a copy of the work was found (see the TUM university library, section mathematics and computer science, in Garching, signature 0109/I 305+306).

Apparently, Jensen did not submit this work anywhere else either<sup>5</sup>. He apparently never received the doctorate. So far, the author of this article knows nothing about Jensen’s life after 1955. Since he submitted the draft of his dissertation with the German title “Postrat” (i.e. a councillor of a post office), he was most probably spending some time of his life in the postal service as a telecommunication engineer.

In his dissertation manuscript Jensen describes fourteen auxiliary devices that OKW/Chi apparently developed and constructed under his supervision. It is of course not surprising and due to the post-war period that he does not mention any other people who worked with him on the equipment. In TICOM interrogation protocol I-37, p. 8, Dr. Hüttenhain states that both graduate engineers Rotscheidt and Jensen were responsible for developing the rapid machinery according to the specifications of the cryptanalysts of OKW/Chi.

Jensen’s manuscript is divided into seven sections. First, the basics of cryptography and second, the basic problems of deciphering are explained. This is followed by a third chapter on the cryptographic elements of the auxiliary devices. A fourth short chapter notes some technical matters on the subject of reading punched tapes. The fifth chapter explains the modular components from which the auxiliary devices were built. Chapter six explains the design and function of the devices. In addition, chapter seven is an elaborate second volume with

interrogation protocols of TICOM; more details can be found in EASI Vol 3, pp. 33-35.

5 Today a second copy is in the possession of the Bayerische Staatsbibliothek Munich which was probably in the ownership of Dr. Hüttenhain before. (Manuscript section, BSB signature Cgm 9303)



technical drawings of all equipment that complements the manuscript.

While Jensen speaks of fourteen devices, the TICOM protocols list only eight, and TICOM also describes that some devices were only in the planning stage and had not yet been fully constructed at the end of the war. The attempt to match the devices from Jensen's manuscript with those from the TICOM protocols was complicated by the fact that the German terms and names of the devices were not always compatible to the TICOM interrogation. This is probably primarily due to the fact that neither Jensen nor Rotschidt, who were mainly responsible for the development of the auxiliary devices, were ever interviewed by TICOM. As well it should be mentioned that Jensen's approach to describe the machinery was primarily technically, and less cryptanalytically, driven.

### 3 The Auxiliary Devices of OKW/Chi

As mentioned before, OKW/Chi used Hollerith machines<sup>6</sup> whenever possible. Dr. Hüttenhain stated that IBM machines could be used for sorting processes in the first place (I-37, p. 2). But for all other applications, section IVb developed special apparatus from 1942 on.

In general, a kind of modular system was created, so that the OKW/Chi's cryptanalysts could reassemble the devices according to their needs (see e.g. I-37, p. 9). This modular system consisted of three major components:

With the so-called reading apparatus "**Abtastwerke**" (see Jensen 1955, pp. 48-55) punched tape was scanned for the criterion: hole or no hole. The result was converted into electrical impulses. Initially OKW/Chi used already available mechanical sensing levers of the punched tape transmitters from Siemens and Lorenz. But soon it became clear that this was

too slow. As a result, photoelectric scanning units were developed (see fig. 1).

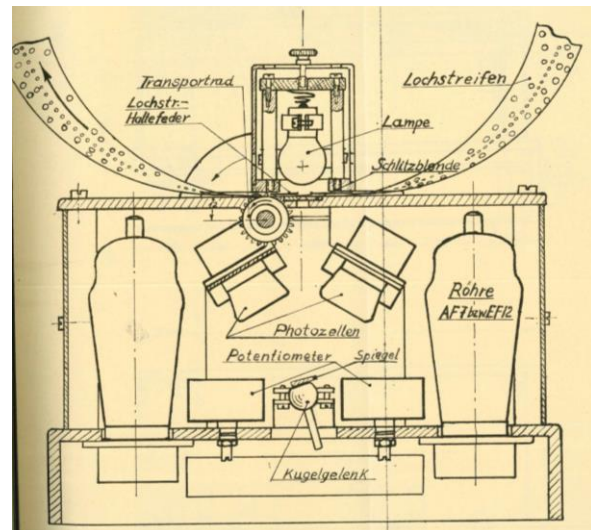


Fig. 1: Photoelectric reading apparatus (drawn by Jensen 1955, Annex 7)

After reading the punched tapes, "**Auswertewerke**" evaluated the information, i.e. the information from the punched tape went in and came out again as groups of letters or numbers. Telegraph relays and telephone relays were mainly used here as so-called cascade converters.

Logical operations, like e.g. XOR, were carried out for statistical calculation of the punched tape information. Jensen called these labyrinths (e.g. character comparison labyrinth, or superimposition labyrinth). These labyrinths were not permanently soldered but pluggable, to remain programmable. Extra calculation cascades performed addition, or subtraction modulo 10 automatically. Jensen described simple cascades versus cascades with a storage function.

Last, so-called "**Registrierwerke**" recorders were used to either display the output of the evaluation on counters or to transfer it to paper or punched tape, by the help of available tape punchers or via automatic typewriter, i.e. a modified Mercedes-Elektra typewriter equipped with electromagnets.

As counters, post office counters were used, but these proved to be impractical, as only five counts per second were possible. Furthermore, they could not be reset to 00000. For this reason, an overrun counter was developed on the basis of a voice coil, or plunger coil.

<sup>6</sup> TICOM protocols report that in general, cryptanalytic machinery was first introduced in Germany with the adoption of Hollerith machines (I.B.M. machinery) by the army in 1941 (see EASI Vol 3 p. 72, & TICOM I-93, p. 5). According to these reports, OKW/Chi did not own any Hollerith machines, but – most probably – used the IBM equipment of the army's cryptanalytic agency (OKH/In 7/VI), since they were housed in adjacent buildings in Berlin.

Jensen divided the auxiliary devices of OKW/Chi into five major categories according to their application:

- Recognition of secret messages
- Deciphering of recognized ciphers
- Decryption of solved ciphers
- Security scrutiny of own ciphers
- Production of secret keys

Analogue to this classification, the equipment will now be described, taking all information into account that could be found in Jensen (1955) and in the relevant TICOM literature.

Almost each of these following devices would be worth going into more detail with an own study. For reasons of space only a superficial description can be given here as a basis for further follow-up studies.

### 3.1 Recognition of secret messages

The simple counting apparatus (“**einfaches Zählgerät**”) determined the frequency distribution of up to 100 different elements. By means of a lever it was possible to switch between the five-digit telegraph alphabet and the 100-digit numbers from 0-99. It was composed of a mechanical scanning unit, a cascade converter with digit-bigram cascade and 100 post office counters to display the results. Due to the relatively low operating speed of the post counters, the device worked about five times faster than one would have needed by hand. (Jensen 1955, p. 35 & 77-81; TICOM I-37, p. 7)

The statistics’ recording apparatus (“**Auswahlzählgerät**”), improved the simple counting apparatus: It determined the frequency distribution of up to 1024 different elements position, as well as feature-related frequencies, vowel spacing, and word lengths. For this purpose, photoelectric scanning units, a bigram cascade of 68 relays and a recorder system consisting of 1036 tracking counters (“**Nachlaufzähler**”), which were particularly developed from plunger coils, were used.

In TICOM I-37 (p. 8) Dr. Hüttenhain mentioned it as a device that was planned or under construction, to be ready in four months. This statement does not correspond to Jensen’s description. According to him, the device

performed the work of 14 working hours in two minutes (Jensen, 1955, p.35 & 82-86).

The “**Sawyer’s Jack**” phase-search apparatus & “**Tower clock**” statistical depth increaser (“**Perioden- und Phasensuchgerät, Sägebock & Turmuhr**”) automatically calculated coincidences of single letters, bigrams, trigrams etc. (i.e. index of coincidence<sup>7</sup>) within one or two cipher texts. As well, the apparatus was able to statistically find out if cipher text passages had been encrypted with the same key, i.e. were in depth. This device was composed of two photoelectric scanning units, a character comparison labyrinth with a large storage bank of telegraph relays, and a special recorder system.

TICOM assumed that OKW/Chi wrote the statistics by hand, because the idea of using such a large and unnecessary bank of relays for this purpose seemed absurd to the interrogators (EASI Vol 2 p. 57). But in fact following Jensen’s description, a huge storage bank of relays had been used here.

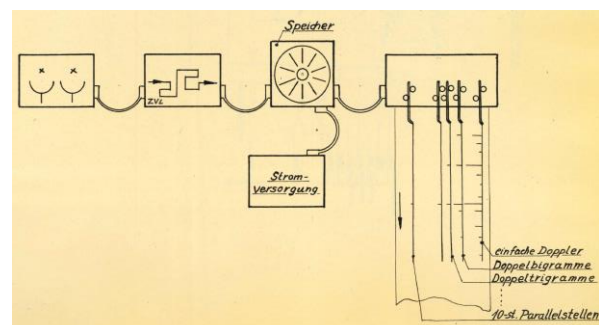


Fig 2. Sägebock & Turmuhr, drawn by Jensen 1955, Annex 43

Interestingly, TICOM documents treat this machine as two machines (see e.g. EASI Vol 2, chart no. 2-3), but Jensen describes it as one apparatus. Bauer (1997, p.303) cites this device from OKW/Chi as well, and repeats Jensen’s statement that it was hundred times faster than manual statistic would have been. In addition, it delivered the results in a very concise way (Jensen, 1955, p. 36-37 & 87-92; TICOM I-31, p. 4; TICOM DF-187A, p. 23).

The repeat finder (“**Parallelstellen-suchgerät**”) was designed to scan text passages for repeats at ultra-high-speed, i.e. approximately

7 Kappa test, Friedman-test, introduced in 1920

10 000 comparisons per second. This should have limited large amounts of text to a few with a higher than average frequency of repeats, allowing them to be examined more intensively with the Sawyer's jack and Tower clock device. It was also intended to test whether scanning at such high speed would still produce accurate results. To be fast, the cipher text passages being compared were not punched on punched tape, but on normal film. For this purpose, a special 2-out-of-10 alphabet was used. Along with photoelectric scanners, the apparatus consisted of a device which, when a repeat passage occurred, produced a spark that burned a hole in an aluminum foil covered with thin paper.

However, Jensen reports that the apparatus had to be destroyed shortly before completion. It seems to have been of particular interest to Jensen, as he uses many pages to describe this device. This device is regrettably mentioned in the TICOM documents that it was unfortunate that there were hardly any technical details about it available (Jensen, 1955, p. 37 & 93-101; EASI Vol 2, p. 64-65; Bauer, 1997, p. 311).

### 3.2 Deciphering of recognized ciphers

The periodic substitution cipher tester ("**Spaltencäsaren-Textgerät**") decided whether a cipher text piece had been encrypted with a known periodic substitution or not. For this purpose, the frequency analyses per cipher text alphabet of the known periodic substitution cipher had to be calculated and stored in the device beforehand. It consisted of a two-headed scanning unit, a cascade converter with the stored frequencies per alphabet and an electromagnetically controlled recorder with rack and writing pen.

This device does not appear at all in the TICOM documents. It is not known to the author if and how successful it has been. Jensen states a working speed of 40 times faster than manual evaluation (Jensen, 1955, p. 38 & p. 102-103).

The bigram weight recorder ("**Bigramm-bewertungsgerät**") was a device for making frequency evaluations of digraphs. It consisted of two tape readers, a bank with five relays<sup>8</sup>, a

plugboard to weight the bigrams according to their usual frequencies in plain language, and a recording pen and drum. It most probably represents the only device of which contemporary photos exist (see fig. 3).

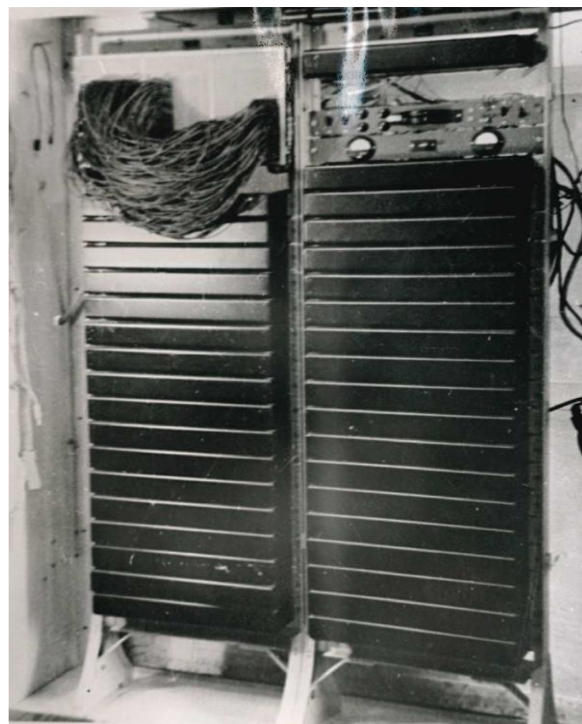


Fig.3: The cascade converter of the bigram weight recorder (Jensen, 1955, p. 106)

According to Dr. Hüttenhain (TICOM I-31, p. 4), it was used to solve the Japanese two-letter transposed code J-19, or Fuji. Solutions could be found in less than 2 hours, doing the work of 20 people (Jensen, 1955, p. 39 & 104-107; Bauer, 1997, p. 399).

The differencing device with storage ("**Differenzenrechnergerät mit Speicher**") is used to automatically form all differences with modulo 10 from a group of cipher text passages that are (most probably) in depth. If two cipher text messages encrypted with the same key are subtracted from each other, the key is removed from both cipher text passages – according to Kahn (1996, p. 440) this was one of the most typical procedures of cryptanalysis during the Second World War. It delivered the base to solve super-enciphered code problems, i.e. after stripping off the key, known codes of suspected words could be added to decrypt the cipher text passages.

The device consisted of a two-headed scanning mechanism, a calculation cascade with storage and an automatic typewriter with digits. This meant that it was possible to work four

<sup>8</sup> Dr. Hüttenhain speaks of 700 telegraph relays (see TICOM I-37, p. 6), and Fenner as well mentions 26<sup>2</sup> relays according to the numbers of bigrams normally possible (TICOM DF-187A, p. 23).



times as fast as by hand, with the result being immediately available in an orderly and clear form, and could be run through without interruption even at night (Jensen, 1955, p. 39-40 & 108-110; Bauer, 1997, p. 339; EASI Vol 2, p. 60-61; TICOM DF-187A, p. 22).

If the text material to be examined was not extensive enough, the difference forming device (**“Differenzenbildungsgerät für Handbetrieb“**) could be used for manual operation. It was also known as the roller apparatus. A maximum of 30 text passages in depth could be subtracted from each other, and codes of suspected words could be added experimentally. The device functioned purely mechanically with the help of five thin metal rods on which 30 small metal rollers were arranged. Each roller contained the numbers from 0 - 9, and according to Jensen, the device was made in two different versions, one for reading with a hanger to place the device comfortably in front of oneself on the table (see fig. 4), and one as a printing device with numbers in mirror writing. The rows of numbers could thus be painted with paint in every intermediate position. A rubber roller was used to make an imprint of the entire constellation of numbers on a sheet of paper laid over it, and a statistician was given the opportunity to examine it. In this way, 10 statisticians could work continuously with only one device.

The printing variant was already the subject of a study by Gallehawk et al (2017). In EASI Vol 2, p. 57ff, this device is described being equivalent to the National Cash Register differencing calculator from the American rapid analytic machinery.

Although the above mentioned eight different cipher departments worked more or less independently, there were nevertheless exchanges from time to time. All of the machines developed by OKW/Chi were shown to the three military Services and the Foreign Office; some were constructed for the other Services, particularly the Roller apparatus (see TICOM I-31, p.5). This could mean that considerably more pieces were made of this device than of others. (Jensen 1955, p. 40 & 111-112; TICOM I-37, p. 2-3; EASI Vol 2, p. 57-60; TICOM DF-187A, p. 21-22).

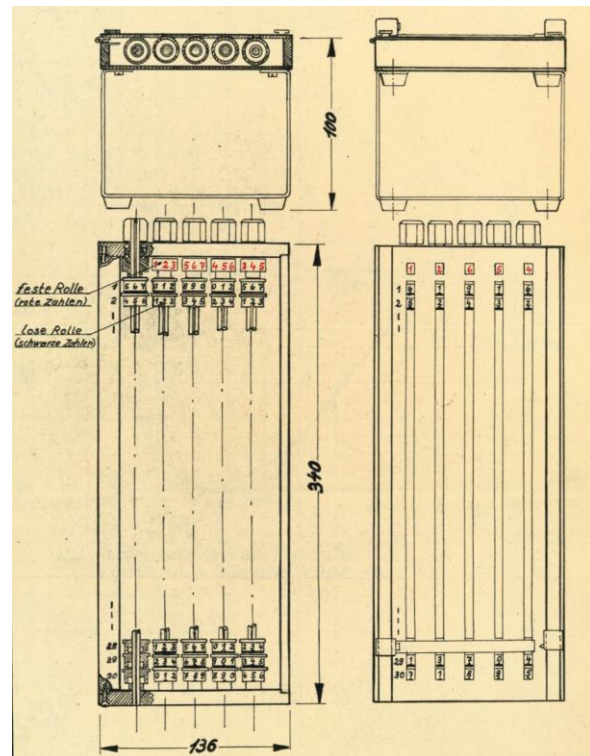


Fig. 4: Roller apparatus (drawn by Jensen, 1955, Annex 62)

If code groups in depth had already been cleared from the key by difference calculation, this manual device called likely-additive selector (**“Reduktionsgerät, Witzkiste”**) could help to check the code groups for the most frequently used codes. It was designed especially for the decipherment of four-digit-codes, and it worked with the superposition of probabilities in a photographically way on 4x4 lattices: Most frequent codes as well as the code groups to be examined were engraved as bright coordinates in two blackened glass plates. When the overlapping plates were illuminated, patterns were created on film material that represented the most probable reduction number. A sketch drawn by Jensen can be seen in fig. 5.

The name “Witzkiste” (i.e. brainbox; “Witz” can mean joke or brain in German) referred to its inventor Prof. Dr. Witt, who worked at OKW/Chi (Jensen, 1955, p. 40-41, 113-120, EASI Vol 2, p. 61-63; TICOM I-31, p. 21; TICOM I-37, p. 8; Weierud & Zabell, 2019, p. 4-5).

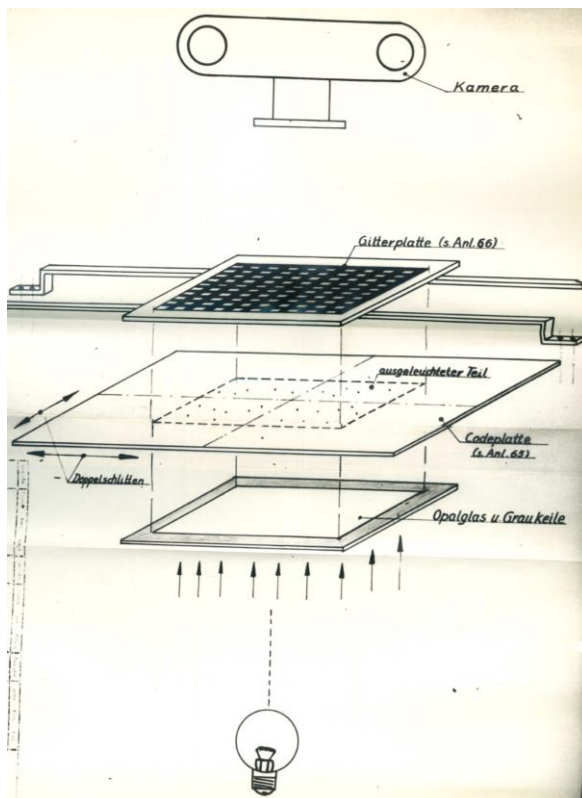


Fig. 5: "Witzkiste" (drawn by Jensen, 1955, Annex 67)

### 3.3 Decryption of solved ciphers

The differencing calculator ("**Differenzen-rechengerät ohne Speicher**") was used to subtract an already recognized encryption number with modulo 10 from a secret code. It could also perform the steps of differentiating between ciphertext passages in depth, but was not as convenient. It was composed of a two-headed photoelectric scanning, a simple computing cascade and an automatic typewriter for digits. It could also be used for encryption (Jensen, 1955, p. 42 & 121-122, TICOM I-37, p. 4).

The converter ("**Tauschumsetzer**") was used to quickly convert text passages encrypted with an already deciphered cipher text alphabet into plain text. For this purpose, an automatic typewriter was extended with an extra panel to plug in the exchange letters. It could also be operated fully automatically with punched tape (Jensen 1955, p. 42 & p. 123; TICOM DF-187A, p. 22).

### 3.4 Security Scrutiny of own ciphers

The mechanical grille columnar transposition device ("**Rasterwürfelgerät**") was a tool to assess the security of the cipher "columnar transposition encoded with grille". Neither the

columnar transposition nor the grille were considered secure. In combination a satisfactory level of security was assumed. The grille created gaps which could be fixed with this device. It was also ideally suited for solving simple columnar transpositions. The structure reminded of a system of co-ordinates made of metal, on which grid fields could be moved and labelled. It was not mentioned in any TICOM document (Jensen, 1955, p. 43 & p. 124).

The superimposing device ("**Überlagerungs-gerät**") was used for the security check of cipher machines with regular rotation of the drums. Jensen did not say this explicitly, but he must have meant the Enigma variants. In order to check sub-periods in different phase positions, the impulse superposition was tested on two punched strips: two scanning units, or two Lorenz transmitters, an overlay labyrinth with 10 telegraph relays and a receiver tape-puncher. The speed of the device was slow because of the puncher. It is not mentioned in TICOM documents as a device (Jensen, 1955, p. 43 & p. 125-126).

### 3.5 Production of Secret Keys

In the lack of a true random generator, one-time-tapes were created using a Siemens T-52c secret writer: the key of the secret writer was over-encrypted with itself and printed on punched tape (Jensen, 1955, p. 44 & p. 127-130).

## 4 Lost & Found

The interesting question to be posed now is: What happened to the rapid analytical machinery? In Jensen's manuscript it can already be read in the introduction that all the devices were destroyed at the end of the war (Jensen, 1955, p. 2). Fenner (TICOM DF\_187, p. 14) reports a mass destruction of just this machinery at the Salzach River near Werfen/ Austria.

It is possible that TICOM employees took devices with them to the USA or to UK. The author has therefore made a request to the depots of the NSA museum and the depot of the GHQC. Unfortunately, the employees of these institutions have not yet been able to find any relics of these devices.

However, between 2005 and 2007 divers succeeded in recovering one type of device several times from a depth of 40m of fresh water: the Roller Apparatus. Unfortunately, the



community of divers and treasure hunters does not allow finding out more about the location of these artefacts. In the beginning there was talk about a lake in Austria and later on about Schliersee. The finding place Schliersee would at least fit to the fact that the whole OKW/Chi-archive including equipment was dumped into the lake<sup>9</sup>. But unfortunately there is no direct contact to the divers to ask for more details.

The devices found so far were resold by a collector in East Germany. According to the author's knowledge, a handful of these devices should exist. Three artifacts are directly known. One of them is located in England and led to the already mentioned paper of Gallehawk et al. (2017).

On a second unit, owned by a private collector in Germany, at least the nameplate with the serial number "SW19" is clearly visible (fig. 6). So now we know that these machines were manufactured by the manufacturer F. Zimmermann & Co. in Berlin. Unfortunately, this company was dissolved in 2004 for financial reasons after 86 years of existence. Whether a company archive still exists, could not yet be found out.



Fig 6: Roller Apparatus, freshly recovered from a lake; by courtesy of Klaus Kopacz, 2019

The third device is owned by the Museumsstiftung Post und Telekommunikation MSTP depot of the Communication Museum in Frankfurt Heusenstamm (fig. 7).

Unfortunately, so far nothing more is known about remaining OKW/Chi auxiliary devices. The author would be very grateful for hints and further knowledge.

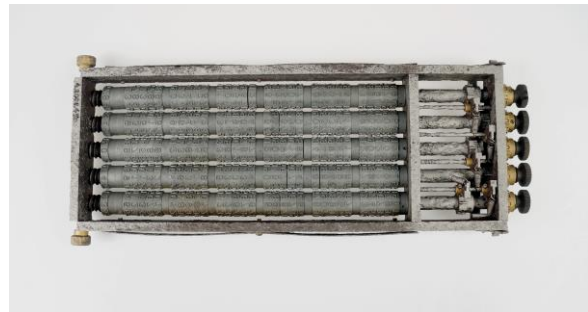


Fig 7: Roller Apparatus, recovered from a lake and cleaned, by courtesy of the MSPT Heusenstamm, Inv. No. 4.2008.450

## Acknowledgments

The author would like to thank Frank Gnegel and the Depot Heusenstamm, Frankfurt as well as Klaus Kopacz for their friendly support and the provision of photos. Furthermore, many thanks go to Katja Rasch, Tina Kubot and Dermot Turing as well as to three anonymous reviewers for their valuable comments.

## References

- John Alexander, John Gallehawk, John Jackson, Allen Pearce & Edward Simpson. 2017. *A German machine for differencing and testing additives*. Cryptologia, 41:3, 269-280, DOI: 10.1080/01611194.2017.1289718
- Friedrich L. Bauer. 1997. *Decrypted Secrets. Methods and Maxims of Cryptology*. Springer-Verlag Berlin Heidelberg
- Friedrich L. Bauer. 2009. *Historische Notizen zur Informatik*. Springer-Verlag Berlin Heidelberg
- EASI European Axis Signal Intelligence Volume 2: *Notes on German High Level Cryptography and Cryptanalysis*. 1945. TICOM DOCID: 3560816. source: <https://archive.org/stream/ticom/>
- EASI European Axis Signal Intelligence Volume 3: *The Signal Intelligence Agency of the Supreme Command, Armed Forces*. TICOM DOCID: 3560827. Source: <https://archive.org/stream/ticom/>
- Erich Hüttenhain. 1970. *Einzeldarstellungen aus dem Gebiet der Kryptologie*. Manuscripts collection, Bayerische Staatsbibliothek, Signature Cgm 9304a, München
- Willi Jensen. 1955. *Hilfsgeräte der Kryptographie, Hauptband + Anlagenband*. Universitätsbibliothek TUM, Signatures 0109/I 305+306, Garching
- David Kahn. 1996. *The Codebreakers*. Scribner, New York

<sup>9</sup> The TICOM Team 5 mentioned, that in August 1945 the northern shore of the Schliersee was still littered with radios and teleprinters (see TICOM Team 5, p. 5)

- David Mowry. 1989. *German Clandestine Activities in South America in World War II*. TICOM DOCID: 3525901. Source: <https://archive.org/stream/ticom/>
- Randy Rezabek. 2013. *TICOM and the Search for OKW/Chi*. *Cryptologia*, 37:2, 139-153, DOI: 10.1080/01611194.2012.687430
- Frode Weierud & Sandy Zabell. 2019. *German mathematicians and cryptology in WWII*. *Cryptologia*, DOI: 10.1080/01611194.2019.1600076
- TICOM DF-187, 1949. *The Career of Wilhelm Fenner with special regard to his activity in the field of cryptography and cryptanalysis*. Source: <https://archive.org/stream/ticom/>
- TICOM DF-187A, 1949. *Organization of the cryptologic agency of the armed forces high command, with names, activities, and number of emplyes together with a descriptioipn of the devices used*. Source: <https://archive.org/stream/ticom/>
- TICOM I-31. 1945. *Detailed Interrogations of Dr. Huettenhain, Formerly Head of Research Section of OKW/Chi*. DOCID: 6587332. Source: <http://chris-intel-corner.blogspot.com/2017/>
- TICOM I-37. 1945. *Translation of Paper Written by Reg. Rat. Dr. Huettenhain of OKW/Chi on Special Apparatus Used as Aids to Cryptanalysis*. Source: <https://archive.org/stream/ticom/>
- TICOM I-176. 1945. *Homework by 'Wachtmeister Dr. Otto Buggisch of OKH/Chi and OKW/Chi*. Source: <https://archive.org/stream/ticom/>
- TICOM IF-1517. 1944. *First Detailed Interrogation of Bigi, Augusto*. TICOM REF ID: A65381. Source: <https://archive.org/stream/ticom/>
- TICOM IF-167. 1945. *Final Report of the Visit of TICOM Team 5 to the Schliersee Area 3rd August 1945 to 7th October 1945*. DOCID: 3745507. Source: <https://archive.org/stream/ticom/>
- TICOM I-206. 1947. *Extracts from Homework written by Min. Rat Wilhelm Fenner of OKW/Chi*. Source: <https://archive.org/stream/ticom/>

# Dawn of Mathematical Cryptology: Probabilists vs Algebraists or Algebraists & Probabilists?

**Marek Grajek**

freelance consultant and historian, Poland

mjg@interia.eu

## 1 Introduction

Traditional cryptology, before the advent of the ciphering machines, relied mostly on the linguistical methods, and the role of mathematics in the codebreaking was limited to counting the frequency of letters, their pairs and triplets. Machine cryptology changed everything; only mathematicians were able to interpret even the bare numbers of combinations resulting from the use of the ciphering machine. The first successful application of advanced mathematics in cryptology, Marian Rejewski's success with Enigma, marked a change of paradigm; his attack was based on the algebra and the group theory. However, soon after the outbreak of WW2 the Germans had changed the Enigma operational procedures, rendering most Polish methods of attack ineffective. British mathematicians, who took over from the Poles, had to revert to the old and proven methods based on probability and statistics, which dominated their work during, and well after WW2. It was only 30 years after the end of this conflict that the role of the algebraic methods was restored.

This paper presents the early period of the development of the mathematical cryptology, focusing on the clash of two approaches to the codebreaking; that based on statistics and probability on the one side, and algebraic methods on the other.

## 2 Historical context

Although traditional, historical codebreaking has always been based on linguistics rather than mathematics, at least since eight century a component of simple application of math was present therein. Al-Kindi, an Arab polymath living in Baghdad in ninth century, described in his "Manuscript on Deciphering Cryptographic Messages" an attack on the monoalphabetic substitution ciphers based on the natural frequency of letters in the language of the clear text. As far as we know his work was based on the earlier (and presently lost) writings of Al-

Khalil<sup>1</sup> (also known as Ahmed al-Farahidi), living in Basra in eight century. Their works linked early cryptography with the equally early methods of mathematical statistics. It should be noted that Al-Kindi's interest in statistics was not limited to the secret writings. He proposed also a statistical approach to the medical treatment evaluation.

During the mediaeval and early modern periods attacks based on the letter frequency were still popular due to the popularity of the nomenclators; monoalphabetic substitution representing a part of the nomenclator made it vulnerable to statistical attack. Later on, when the codes and nomenclators started to lose their popularity in favour of simpler and more practical ciphers, statistical attacks have gained in importance; codebreakers started analysing not only the frequency of the single letters, but also their pairs, triplets and entire, popular words.

Invention and fast adoption of the telegraph has changed this picture for a moment. Early telegraph required not only hiding the message content, but also its compacting. Codes provided an easy and practical answer; second half of the nineteenth century was heavily dominated by the use of codes, which, from the codebreaker's perspective, required the application of the linguistical rather than statistical methods of attack. Use of the radio during the Great War has brought another game changer. Both sides used radio on a mass scale. Ease of interception of the radio messages forced the application of cryptography at the equally mass scale, and the traditional codes were getting more and more impractical; during World War One ciphers replaced the codes as the mainstream of cryptography, and, consequently, statistics replaced the linguistics as the mainstream of cryptanalysis.

However, statistical methods used in cryptanalysis represented rather elementary applications of mathematics, which could be dealt with by amateurs. Immediately after the end of World War One agencies of major countries dealing with cryptology did not realize the need

to employ or train mathematicians. If some mathematicians happened to be employed in the crypto world, it was only due to their general intellectual discipline, and not to the particular skills resulting from their scientific discipline. Werner Kunze was employed at the German foreign ministry cipher office in 1919, but it was only in 1936 that he became the head of its newly created mathematical section. William Friedman published in 1930 an offer to employ three “government mathematicians” at some obscure agency of the US Army. From the memories of Solomon Kullback, Frank Rowlett and Abraham Sinkov, whom he selected from among the candidates, we learn that for the first several years nature of their occupations was rather distant from mathematics.

It seems that the first agency dealing with cryptology that consciously and purposefully decided to employ and train mathematicians was the Polish Cipher Bureau in 1928. Effects of that decision are well known among the historians of cryptology; after the half year training in cryptology in Poznań, in 1929, and three years long period of apprenticeship in the codebreaking, Marian Rejewski was asked to take a look at the real objective of this effort – the Enigma cipher. It took him less than three months to break the cipher and, simultaneously, to change the nature of cryptology forever.

### 3 Probabilists vs Algebraists

When in October 1932 Maksymilian Ciężki had asked Marian Rejewski to take a look at the materials that Polish Cipher Bureau was able to gather about Enigma (Rejewski, 1967), Rejewski, in spite of his over three years long training in cryptology, was still a mathematician rather than the codebreaker. One might say, luckily for the civilized world; had he been the cryptologist, he would have probably tried to apply the traditional codebreaking methods, completely ineffective versus Enigma cipher. Rejewski started his work identifying some purely mathematical features of the cipher and continued transforming his entire knowledge about the machine and its cipher into a system of equations. He was unable to solve these equations outright, as the variables they contained represented unknown permutations rather than the numbers. Theory permitting to solve such type of equations was missing and Rejewski had to provide it himself, which he did,

and in the last days of 1932 he managed to solve his equations, reengineering thus Enigma machine in a purely mathematical way.

Terms used in the description above do not leave a shadow of doubt that Rejewski was using an absolutely pioneering approach. System of equations represents a term functioning in the purely algebraic context. Permutations are used in the context of the theory of groups. Neither reminds ideas or notions used in the probability or statistics, dominating codebreaking up to that moment. It is somewhat surprising that Rejewski had not started his attack from the statistical approach, considering his earlier professional plans. Just after having completed his studies in mathematics at the Poznań University, he decided to continue education in the actuarial statistics at Göttingen. One of his relatives was among the founders of the first life insurance company in Wielkopolska, and Marian Rejewski obviously planned to start a career in the insurance business.

Algebraic and group theoretic approach, used by Rejewski in his breakthrough attack at the Enigma cipher, had numerous advantages over the statistical attacks used against the earlier hand ciphers. Its crucial advantage was that it worked. Codebreaking agencies of the major countries initially declared helplessness when confronted with the Enigma cipher. William Clarke, one of the veterans of British Room 40, remarked in a memorandum written in 1937 that “only one cloud obscures the horizon – possibility of general application of the ciphering machines. One can argue that it would mean the complete end of the codebreaking”. Frank Birch noted an opinion of one of G.C.&C.S. heads of section stating that “(a)ll the German ciphers are unbreakable. (...) putting pundits on them represents a waste of time”.

Rejewski’s approach was unique among the traditional codebreaking methods, as its success was deterministic rather than probabilistic. Most codebreaking methods used up to his breakthrough were offering only a promise of success, without granting it. Success depended on many factors beyond the codebreaker’s control: errors committed by the cipher clerk, external evidence permitting to guess the content of the message, inspired guess of the probable plaintext. Algebraic approach invented by Rejewski virtually granted the success, provided that the codebreaker was able to accumulate some 80-100 messages during a single day.



Finally, with the proper tooling Rejewski's method was extremely efficient and fast. In 1935 Polish team decided to construct a simple electromechanical device named cyclometer. Cyclometer was used to simplify the preparation of the catalogue of so called cyclic characteristics. Ready catalogue of the cyclic characteristics permitted to break over 70% of the intercepted German messages within just few hours after interception. In many cases the deciphered messages reached the eyes of the Polish intelligence officers before they landed on the desk of their rightful German receiver.

Success reached using the algebraic approach did not make Polish mathematicians blind to the possibilities offered by the traditional statistics. In fact, the team seems to have been divided between the adepts of algebra and group theory and those of the statistics. Jerzy Różycki, the youngest member of the team, from the very beginning was focused on the statistics, usually with great success. Still during their apprenticeship the team was asked to break the training code of the German Navy. Różycki started his work with the observation that in any language number of words starting with any particular letter of alphabet represents characteristic feature of the language. He divided the intercepted codewords into the groups of various frequency and started thus successful recovery of the codebook (Rejewski, 1967). A little later Różycki invented the ingenious method permitting determination of right-hand Enigma rotor, called the "clock method" (Rejewski, 1967). His method relied on the idea of the index of coincidence, originally described by William Friedman in 1922. There are some indications that Różycki might have discovered the index of coincidence independently of Friedman's original work (Grajek, 2019).

Algebraic approach served the Polish team well until 1938, then the situation started to get complicated. During the Munich crisis German army has modified Enigma's ciphering procedure, making cyclometer and the catalogue of cyclic characteristics obsolete. In the increasingly confusing political situation the codebreakers had to find a new way to break the cipher, and to find it fast. Rejewski (1967) responded with a concept of the "bomba" – an electromechanical device running through the entire key space within less than two hours and stopping whenever potential solution was found.

He developed the new idea within a month and it took the AVA company working for Polish intelligence service another month to deliver six prototypes, but nobody was proud of this achievement. First – because in December the Germans increased the number of rotors to five, increasing tenfold the number of bombas necessary to break the cipher. Second – because Rejewski seemed to consider necessity to reach for the machinery as the failure of his beloved mathematics. And third – because the bomba did not implement the attack based on the algebraic, but only statistical approach.

Most Enigma historians assume that bomba was designed to look for so called females, i.e. one letter long cycles in the Enigma cipher, transforming some letter of the cleartext into the same letter of ciphertext twice in the distance of exactly three characters. The very idea of females was valid only in the context of another method of attack, being developed in parallel to the bombas by Henryk Zygalski, and therefore referred to as Zygalski sheets. The females in the Zygalski sheets represented the cyclic property of the Enigma cipher and their existence and nature resulted directly from the algebraic considerations regarding the cipher.

This was not the case with the pairs of letters sought for by the bomba. In his description of his construction Rejewski (1967) referred to the object of its search using the term "spectacles" rather than females, stressing the difference between both concepts.

Spectacles represented a purely probabilistic property of the cipher and therefore the bomba did function only in the probabilistic and not deterministic fashion; under certain circumstances it could find the key to the cipher, but the solution was not granted. Rejewski never openly demonstrated his disappointment with his own idea. However, an emphatic reader may easily spot the difference in the tone of his description of bomba and purely mathematical methods of attack. Describing the bomba Rejewski pretends to have forgotten the details of its construction and functioning and attempts to diminish its role, revealing involuntarily his emotional attitude towards his own creation. His remarks provide a strong contrast with his comments regarding the Zygalski sheets which, although they do not represent his own idea, belong to the mainstream of his algebraic thinking about the cipher. It is a pity that even writing his memoirs in the late 1960s, he remained ignorant that his bomba represented the

foundation for a family of machines constituting the basis of the Allied cryptologic effort during the war.

It was true, however, that the algebraic approach preferred by Rejewski and his colleagues has reached its apex sometime in 1937/1938, and was doomed to decline over the next few years: events they were able to keep under control since 1932 started to slip out of their hands. Everything started from the changes introduced by their adversary around the Munich crisis. Members of the Polish team used to comment mistakes made by the German crypto service saying “they’d better do it in this or that way...”. And surprisingly, in just few months their adversary was changing his systems strictly following their own opinions. Poles started to suspect the existence of a mole within their closest circle (which, according to our present state of knowledge, was not true).

One of the conclusions of the Pyry meeting in July 1939 divided the efforts between the cooperating parties; British codebreakers were responsible for the construction of the necessary equipment, French for using their agent in Berlin to get more information about Enigma, and the Poles for the studies in the theory of Enigma ciphers. That division soon fell victim of the wartime reality. Polish team was able to find a refuge in France and to reorganize in P.C. Bruno only to discover, that the Poles represented virtually entire cryptology of the French army. Concentrating their efforts on the daily, operational codebreaking they were unable to continue their studies in the theory – initiative passed into the hands of the more resourceful British codebreakers (Grajek, 2019).

One might think that the British would be naturally inclined to continue their work more or less along the lines drawn by their Polish predecessors. Most of the young mathematicians entering the gates of Bletchley Park were educated in the intellectual tradition best epitomized by opinions by Godfrey Hardy, stressing the importance of pure vs applied mathematics (including well known “[r]eal mathematics has no effects on war. No one has yet discovered any warlike purpose to be served by the theory of numbers or relativity, and it seems very unlikely that anyone will do so for many years”) (Hardy, 1940). In the reality of 1939/1940 attempts to continue algebraic attacks at the Enigma cipher would almost certainly lead

to nowhere. So it was very fortunate that one of the first mathematicians to cross the gates of Bletchley Park was Alan Turing, who was never particularly concerned with the opinions of his professional circle and was usually following his own ways. This permitted him to create an interesting synthesis of the original, Polish algebraic approach with a new one, based on probability rather than algebra.

He took Rejewski’s earliest discovery, that of Enigma cipher’s cyclic property, as the foundation of his design; his machine was supposed to traverse the key space searching for the closed cycles (Turing, 1940). Contrary to Rejewski’s original design he was not planning to search for these cycles within the message headers, but rather in the message contents. We might easily recognize Dilly Knox behind that decision. Immediately after his return from Pyry Knox expressed opinion that all Polish successes were based on a factor which might be removed by the adversary any moment: double encipherment of the message key. Dilly was right; that was precisely what happened on May 1<sup>st</sup>, 1940. At that time Turing bombe was already in the production process, and it did not rely on the analysis of the message indicator, so the change did not affect its construction.

There was, however, a price to be paid. Turing had designed his bombe so that it could search for the cycles within the fragment of the probable text (a crib) assumed by the codebreaker to be present in the coded message. His bombe was able to provide a solution only, and exclusively only, when this guess was right. Bombe’s functioning was algebraic and deterministic with regard to the cycle search, and purely probabilistic with regard to the choice and position of the probable text. Later on Gordon Welchman strengthened the deterministic part of its job, adding the diagonal table, but overall the efficiency of the bombe was determined by the probabilistic component. As long as the codebreakers were able to provide a good and stable crib, they were able to break the key; otherwise the cipher remained invulnerable.

Bombe provided a practical solution for the networks of the German land and air army. Navy was using Enigma in much more ingenious way, resisting British attacks until late spring 1941. It was in the context of the struggle with the naval Enigma, that the British codebreakers switched entirely to the probabilistic attacks. Alan Turing and his colleagues proposed at least three

different methods of attack at the naval Enigma, all of them based purely on the statistical properties of the cipher. E-rack represented most elementary of them, using the well-known fact that letter “e” represents most frequent letter in the German language, appearing in the written texts with outstanding frequency of nearly 17%. E-rack was based on the slightly paradoxical assumption that entire text being analysed consists of letter “e” only (Alexander, 1945). After the initial breakthrough with the naval Enigma E-rack assured several successes with the “Offizier” variant of the cipher.

Another method invented in the process was called “EINSing” (Alexander, 1945). Analysing decoded texts of German military messages the codebreakers have noticed that the most frequent single word encountered therein was EINS. They have designed a simple device enciphering EINS at each and every position of Enigma rotors and registering the result on the perforated cards. Then it was enough to register intercepted messages on the perforated cards and compare them, using the electromechanical sorter, with cards containing the EINS catalogue.

Third and most advanced method of attack on the naval Enigma was banburismus (Alexander, 1945). Its goal was to identify the right-hand Enigma rotor and, consequently, to narrow the number of rotor combinations being checked by the bombe. Banburismus represented the extension of the pre-war method proposed by Jerzy Różycki and referred to as the “clock method”. Różycki used to analyse pairs of messages, whose indicators differed only in the last position; banburismus extended his approach for the pairs of messages differing in two, and under certain circumstances even three positions. Codebreakers were registering incoming messages on the special sheets (manufactured in Banbury, hence the name of the method) and sliding pairs of sheets vs each other, looking for repeats. Every repeat one, two or three letters long was weighted using specially designed tables, measuring the probability of the coincidence. Sum of the partial results determined the probability that both messages were enciphered at the same or similar Enigma settings. It is worth noting that for the sake of banburismus Alan Turing invented the concept of “ban” – a measure of information equivalent to bit proposed by Claude Shannon. Banburismus was further extended to the “tetra catalogue” – repeats four or more letters long, processed using sorters and tabulators in the section called (from

the name of its head) “Freebornery” (Alexander, 1945).

Neither of the described methods of attacks permitted breaking of the cipher directly. All of them were interdependent; efficient application of one depended on the earlier success of the other. Alan Turing and his colleagues had to wait until April/May 1941, when the documents captured onboard of some German ships permitted to overcome the crisis, and to start more or less regular operation of breaking the naval Enigma.

Their brief description above illustrates their nature sufficiently to recognize their purely probabilistic character. Under the pressure of the war necessity British codebreakers have given up algebraic approach, switching almost entirely to the well-established probabilistic and statistical methods. This tendency was further strengthened later on, during the attacks on the German teletype ciphers. Functioning of both devices constructed by the British codebreakers for this purpose, Heath Robinson and Colossus, relied on counting the measure of coincidence between the intercepted text and the pattern enciphered at every setting of the ciphering machine.

Two factors regarding British preference for probabilistic and statistical methods deserve additional comment. When Alan Turing was looking for a base for his banburismus, he decided to choose the less popular branch of statistics, the Bayesian inference, taking thus the position in the old debate between the *a priori* and *a posteriori* statisticians. Interestingly, using an *a priori* approach assured the German mathematicians about the security of Enigma ciphers (Ratcliff, 2003). Turing did not agree with a very principle of using an *a priori* approach; he argued that the ciphertext itself reveals some information about the cipher and the codebreaker should take this information into account. It was thus natural to reach for an *a posteriori* inference, and the Bayesian statistics provided a natural tool.

Second factor was of purely human nature. Most of the mathematicians recruited to Blechley Park belonged simultaneously to the top ranking group of chess players, at least in Britain, and some of them (among them C.H’O.D. Alexander) represented the top world level. Among the various circles, groups and clubs organized at the BP to provide recreation, chess club belonged to most numerous and most active.

So far no one was able to formulate an algebraic theory of the chess game; chess player naturally formulates his thinking about the game in terms of probability. It was thus natural to extend this model of thinking in the new game that the chess playing mathematicians were participating in.

As far as we know after the end of hostilities the codebreaking has for many years remained heavily dominated by the probabilistic and statistical methods. The landscape started to change only in late 1960s and early 1970s, when algebraic approach started to regain its citizenship rights in cryptology, being bravely accompanied by the number theory.

#### 4 Algebraists & Probabilists

Although some simple statistical methods have been traditionally used in the codebreaking for over ten centuries, Polish success with Enigma in 1932 marked the real birth of the mathematical cryptology.

Interestingly, it was based on the oldest, perhaps next to geometry, field of mathematics, algebra. Rejewski's breakthrough was by no means accidental. Polish Cipher Bureau was the first cryptology agency in the world, which not only decided to employ mathematicians, but also expected, encouraged and trained them to apply their mathematical workshop in the codebreaking. Other codebreaking services followed its suit only after learning, directly or indirectly, about Polish success.

Methods of Enigma breaking invented by Polish team were somewhat exceptional. Their algebraic character made them deterministic: they granted breaking the cipher whenever Cipher Bureau was able to accumulate sufficient number of messages, without additional conditions regarding their contents. In that aspect they represented almost an antithesis of the then mainstream of traditional cryptanalysis, relying entirely on statistics and probability. Moreover, they were invented and used just in time to demonstrate their power. A few years later German crypto services started restructuring their operations, recruiting more mathematicians and permitting them to look at the codes, ciphers and machines from the perspective of their discipline. This new approach permitted to eliminate some mistakes and idiosyncrasies in the design of German ciphers, among them those, which made Polish approach so effective.

It was fortunate for the Allied cause that right at that moment the initiative in the attacks at Enigma ciphers passed into the British hands. The situation was developing in a somewhat paradoxical way. Marian Rejewski's studies in actuarial statistics indicate his interest in the applied mathematics. In spite of that he developed a theory of attack at the cipher in the best style of pure math. Most British codebreakers were educated in the respect for the pure math, and in spite of that decided to change the paradigm and switch to the probabilistic and statistical methods, the only ones practical considering the necessities of war and the only ones offering the prospects of success.

The history of attacks at Enigma ciphers is almost synonymous with the earliest period of the development of mathematical cryptology. It is fascinating to note that during that very early period Allied codebreakers developed and successfully applied methods based on two mutually complementary areas of modern cryptology; algebra on one part, and probability and statistics on the other. Present cryptanalysis relies on the mixture of both approaches. Its first stage usually involves the exploitation of cipher's algebraic properties to limit the search space. Then the probability and statistics take suit to find the solution within that limited space. It is interesting to note that precisely this approach provided the base for the construction of the Turing bombe.

#### References

- Alexander C.H.O'D., *Cryptographic History of Work on German Naval Enigma*, 1945, NA HW 25/1
- Al-Farahidi Ahmed, *Book of Cryptographic Messages*
- Friedman, W.F., 1922, *The index of coincidence and its applications in cryptology*. Department of Ciphers. Publ 22. Geneva, Illinois, USA: Riverbank Laboratories
- Grajek Marek, 2019, *Sztafeta Enigmy. Odnaleziony raport polskich kryptologów*, Agencja Bezpieczeństwa Wewnętrznego, Emów 2019
- Hardy, G. H., 1940, *A Mathematician's Apology*, Cambridge University Press 1940
- Al-Kindi, *Manuscript on Deciphering Cryptographic Messages*
- Ratcliff R.A., 2003, *How Statistics Let the Germans to Believe Enigma Secure and Why They Were Wrong: Neglecting the Practical Mathematics of Cipher Machines*, *Cryptologia* 2003/2
- Rejewski Marian, 1967, *Memories of my work at the Cipher Bureau of the General Staff Second*

*Department 1930-1945*, Adam Mickiewicz  
University Press, Poznań 2013

Turing Alan, 1940, *Prof's Book*, NA HW 25/3



# Of Ciphers and Neurons

## Detecting the Type of Ciphers Using Artificial Neural Networks

**Nils Kopal**

University of Siegen, Germany  
nils.kopal@uni-siegen.de

### Abstract

There are many (historical) unsolved ciphertexts from which we don't know the type of cipher which was used to encrypt these. A first step each cryptanalyst does is to try to identify their cipher types using different (statistical) methods. This can be difficult, since a multitude of cipher types exist. To help cryptanalysts, we developed a first version of an artificial neural network that is right now able to differentiate between five classical ciphers: simple monoalphabetic substitution, Vigenère, Playfair, Hill, and transposition. The network is based on Google's TensorFlow library as well as Keras. This paper presents the current progress in the research of using such networks for detecting the cipher type. We tried to classify all ciphers of a new MysteryTwister C3 challenge called "Cipher ID" created by Stamp in 2019. The network is able to classify about 90% of the ciphertexts of the challenge correctly. Furthermore, the paper presents the current state-of-the-art of cipher type detection. Finally, we present a method which shows that one can save about 54% computation time for classification of cipher types when using our artificial neural network instead of trying different solvers for all ciphertext messages of Stamp's challenge.

## 1 Introduction

Artificial neural networks (ANNs) experienced a renaissance over the past years. Supported by the development of easy-to-use software libraries, e.g. TensorFlow and Keras, as well as the wide range of new powerful hardware (especially graphic card processors and application-specific integrated cir-

cuits). ANNs found usages in a broad set of different applications and research fields. Their main purpose is fast filtering, classifying, and processing of (mostly) non-linear data, e.g. image processing, speech recognition, and language translation. Besides that, scientists were also able to "teach" ANNs to play games or to create paintings in the style of famous artists.

Inspired by the vast growth of ANNs, also cryptologists started to use them for different cryptographic and cryptanalytic problems. Examples are the learning of complex cryptographic algorithms, e.g. the Enigma machine, or the detection of the type of cipher used for encrypting a specific ciphertext.

In late 2019 Stamp published a challenge on the MysteryTwister C3 (MTC3) website called "Cipher ID". The goal of the challenge is to assign the type of cipher to each ciphertext out of a set of 500 ciphertexts, while 5 different types of ciphers were used to encrypt these ciphertexts using random keys. Each cipher type was used exactly 100 times and the different ciphertexts were shuffled then. While the intention of the author was to motivate people to start research in the field of machine learning and cipher type detection, all previous solvers solved the challenge by breaking the ciphertexts using solvers for the 5 different cipher types. Thus, after revealing the plaintext of each cipher, the participants knew which type of encryption algorithm was used.

We started to work on the cipher type detection problem in 2019 with the intention to detect the ciphers' types solely using ANNs. TensorFlow (Abadi et al., 2016) and Keras (Chollet, 2015) were used. TensorFlow is a free and open-source data flow and math library developed by Google written in Python, C++, and CUDA, and was publicly released in 2015. Keras is a free and open-source library for developing ANNs developed by Chollet and also written in

Python. In 2017 Google’s TensorFlow team decided to support Keras in the TensorFlow core library. While working on the cipher type detection problem, Stamp’s challenge was published. We then adapted our code and tools to the requirements of the challenge. Therefore, in this paper, we present our current progress of implementing a cipher type detection ANN with the help of the aforementioned libraries especially for the MTC3 challenge. At the time of writing this paper, we are able to classify the type of ciphers of the aforementioned challenge at a success rate of about 90%. Despite this relatively good detection rate it is still not good enough to solve the challenge on its own. Therefore, we also propose a first idea of a detection (and solving) method for ciphertexts with unknown cipher types.

The contributions and goals of this paper are:

1. First public ANN classifier for classical ciphers developed with TensorFlow and Keras.
2. Presentation of the basics of ANNs to the audience of HistoCrypt, who are from different research areas, e.g. history and linguistics (but mostly no computers scientists).
3. Example Python code which can be used to directly implement our methods in TensorFlow and Keras.
4. Overview of the existing work in the field of ANNs and cryptanalysis of classical/historical ciphers and cipher type detection.
5. Presentation of a first idea of a method which does both, cipher type detection and solving of classical ciphers.

The rest of this paper is structured as follows: Section 2 presents the related work in the field of machine learning and cryptanalysis with a focus on ANNs. Section 3 shows the foundation on which we created our methods. Here, firstly we discuss ANNs in general. Secondly, we briefly present TensorFlow as well as Keras. After that, Section 4 presents our cipher type detection approach based on the aforementioned libraries. Then, Section 5 discusses our first ideas for a cipher type detection and solving method. Finally, Section 6 briefly concludes the paper and gives an overview of planned future work with regards to ANNs and cryptology.

## 2 Related Work

In this section, we present different papers and articles, which deal with ANNs and cryptology. The usage of ANNs in the paper ranges between the emulation of ciphers, the detection of the cipher type, and the recovering of cryptographic keys. Also, there are papers where the authors worked with other techniques to detect the cipher type.

1. Ibrahim (Khalel Ibrahim Al-Ubaidy, 2004) presents two ideas: First, to determine the key from a given plaintext-ciphertext pair. He calls this the “cryptanalysis approach”. Second, the emulation of an unknown cipher. He calls this the “emulation approach”. He used an ANN with two hidden layers in his approach. For training his model he used Levenberg-Marquardt (LM). He successfully trains Vigenère cipher as well as two different stream ciphers (GEFFE and THRESHOLD, which are both linear feedback shift registers).
2. Chandra (Chandra et al., 2007) present their method of cipher type identification. They created different ANNs which are able to distinguish between different modern ciphers, e.g. RC6 and Serpent. Their ANN architecture is comparable small, consisting only of 2 hidden layers, where each layer has at most 25 neurons. They used different techniques to map from the ciphertext to 400 “input patterns”, which they fed to their network.
3. Sivagurunathan (Sivagurunathan et al., 2010) created an ANN with one hidden layer to distinguish between Vigenère cipher, Hill cipher, and Playfair cipher. While their network was able to detect Playfair ciphers with an accuracy of 100%, the detection rate of Vigenère and Hill was between 69% and 85%, depending on their test scenarios.
4. The BION classifiers from BION’s gadget website<sup>1</sup> are browser-based classifiers, integrated in two well working cipher type detection methods built in JavaScript. The first one works with random decision forests and the second one is based on a multitude of ANNs. The basic idea with the second classifier (ANN-based) is, that the different networks (different layers, activation functions,

<sup>1</sup>see <https://bionsgadgets.appspot.com/>

etc.) each have a “vote” for the cipher type. In the end, the votes are shown, and the correct cipher type probably has the most votes. The classifiers are able to detect the cipher types defined by the American cryptogram association (ACM).

5. Nuhn and Knight’s (Nuhn and Knight, 2014) extensive work on cipher type detection used a support vector machine based on the lib-SVM toolkit (Chang and Lin, 2011). In their work, they used 58 different features to successfully classify 50 different cipher types out of 56 cipher types specified by the American cryptogram association (ACA).
6. Greydanus (Greydanus, 2017) used recurrent neural networks (RNN) to learn the Enigma. An RNN has connections going from successive hidden layer neurons to neurons in preceding layers. He showed that an RNN with a 3000-unit long short-term memory cell can learn the decryption function of an Enigma machine with three rotors, of a Vigenère cipher, and of a Vigenère Autokey cipher. Furthermore, he created an RNN network which was able to recover keys (length one to six) of Vigenère and Vigenère Autokey.
7. Focardi and Luccio (Focardi and Luccio, 2018) present their method of breaking Caesar and Vigenère ciphers with the help of neural networks. They used fairly simple neural networks having only one hidden layer. They were able to recover substitution keys with a success rate of about 93%, where at most 2 mappings in the keys were wrong.
8. Abd (Abd and Al-Janabi, 2019) developed three different classifiers based on neural networks. Their work is the closest related to our work. Their idea is to create three classifiers, each a single ANN, with different levels (1, 2, and 3), where each level increases the detection accuracy. The first level differentiates between natural language, substitution ciphers, transposition ciphers, and combined ciphers. Then, their second level differentiates between monoalphabetic, polyalphabetic, and polygraphic. Their last level differentiates between Playfair and different Hill ciphers. They state that their success rate is about 99.6%.

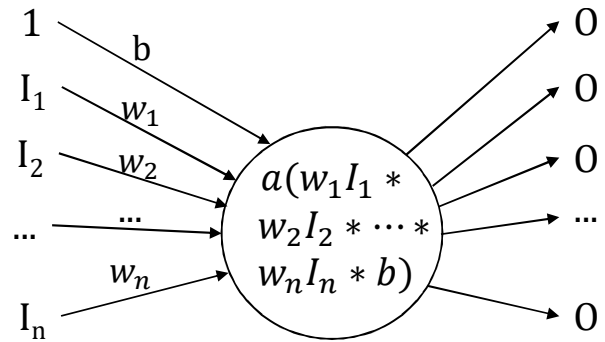


Figure 1: A single neuron of an ANN with inputs, outputs, bias, and activation function

### 3 Foundation

In this section, we describe the foundation used for our detection method. First, we discuss the ANN in general. Then, we give an introduction to TensorFlow and Keras and show some example Python code building an ANN.

#### 3.1 Artificial Neural Network

Artificial neural networks (ANNs) are computing models (organized as graphs) that are in principle inspired by the human brain. The book “Make your own neural network” from (Rashid, 2016) gives a good introduction into ANNs. Different **neurons** are connected via input and output connections, providing **signals**, having different **weights** assigned to them. A neuron itself contains an **activation function**  $a$ , which fires the neuron’s outputs based on the neuron’s input values. For example, all the values of the input connections are combined with their respective weight values. Then, all resulting values are combined and a bias value  $b$  is also added to the result. After that, an activation function is computed using the result of the combined values. Figure 1 depicts an example of one neuron with different input connections, a bias input connection, an activation function  $a$ , and output connections. Usually, the value of the bias input connection is set to 1.

A common practice in ANNs is to organize neurons in so-called **layers**. The input data is given to an **input layer** consisting of  $n$  different neurons. The input layer is then connected to one or more **hidden layers**. Finally, the last hidden layer is connected to an **output layer**. Each neuron of the previous layer is connected to each neuron of the following layer. Figure 2 depicts an example of an ANN with only a single hidden layer. In general,

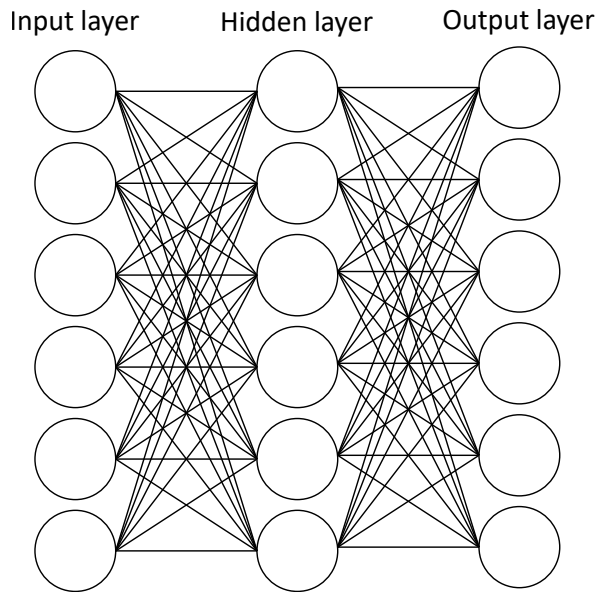


Figure 2: An ANN with input, hidden, and output layers

when working with ANNs having several hidden layers, researchers refer to the term of **deep learning** (Wartala, 2018).

The learning, in general, is performed by adapting the weights of the connections between the neurons. There exist different methods for learning, e.g. supervised and unsupervised learning. Here, we focus on supervised learning, which is suited well for classification tasks. The input data is given as a so called **feature vector**  $x$  from the input space  $X$  and the output is a **label**  $y$  from the output space  $Y$ . A label, in general, clusters a set of similar input values, i.e. each of the input values of the same **cluster** is mapped to the same label. The goal is to find a function  $f : X \rightarrow Y$  that maps each element of the input space correctly to the labels of the output space.

As a basic idea, the ANN's connection weights are initialized with random values. Then, a set of data (inputs and desired labels) is feeded to the network. While doing so, the actual output labels as well as the desired labels are compared using a **loss function**. Using **back propagation** the error is propagated in the reverse order through the network and the weight values are changed for each neuron of each layer accordingly.

Different parameters and attributes of the ANN and the learning process influence the success rate of the learning: e.g. the quality and quantity of the input data and labels, the number of hidden layers of the ANN, the number of neurons of each

layer, the types of used activation functions of the neurons, the used loss function, and the number of times the input data is feeded to the network.

Usually, the input data and their respective labels are divided into two different sets: **training data** and **test data**. For the actual learning, the training data is used. Then, to measure the quality of the ANN the test data is used. In the best case, after training the ANN is able to classify the test data correctly. In the worst case, the ANN only learned the training data (perfectly), but fails in classifying the test data. In this case, researcher refer to the term **overfitting**.

### 3.2 TensorFlow and Keras

TensorFlow (Abadi et al., 2016) is a software library developed by Google and firstly released in 2015. Its name is based on the term "tensor", which describes a mathematical function that maps a specific number of input vectors to output vectors, and on the term "flow", the idea of different tensors flowing as data streams through a dataflow graph. Keras (Chollet, 2015) is an open-source deep learning Python library and since 2017 also included in TensorFlow.

Working with TensorFlow and Keras (with ANNs), in general, consists of the following five steps:

1. Loading and preparing training and test data
2. Creating a model
3. Training the model
4. Testing and optimizing the model
5. Persisting the model

In the following, we describe the above steps involved in the creation, training, and usage of a Keras model. TensorFlow models work on multi-dimensional Python numpy arrays.

Step 1) First, the data has to be loaded and then split into a test and a training data set. In the following example, we split a data set of 5000 test data and their according labels (each label corresponds to one output class) into two disjunct sets of training and test data and labels:

---

```
# data is a set of data
# labels is a set of labels
# here, we split both into
# two different sets
```

```
train_data = data[0:4500]
train_labels = labels[0:4500]
test_data = data[4500:5000]
test_labels = labels[4500:5000]
```

---

Step 2) The second step is the creation of a Keras model. TensorFlow and Keras offer different methods of creating a model. The easiest method is to use the sequential model, which creates a multi-layered ANN. An example call of creating a simple ANN with an input layer, a single hidden layer, and an output layer is the following:

```
# create model:
m = keras.Sequential()
# create and add input layer:
m.add(Flatten(input_shape=(100,)))
# create and add hidden layer:
m.add(Dense(100,
    activation='relu',
    use_bias=True))
# create and add output layer:
m.add(Dense(5,
    activation='softmax'))
m.compile(optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
```

---

The first call creates a sequential Keras model. With the add-function, layers are added to the model. We add an input layer with 100 neurons (or features), a hidden layer with 100 neurons, and an output layer with 5 neurons. Each neuron of the next layer is automatically connected to each neuron of the previous layer, as shown in Figure 2. In this example, we classify some data with 100 features into 5 different output classes. Some remarks on the parameters: the activation function of the hidden layer is set to rectified linear unit ('relu'), which is defined as  $y = \max(0, x)$ . The activation function of the output layer is set to 'softmax', which is also known as a normalized exponential function. It maps an input vector to a probability distribution consisting, in our case, of 5 different probabilities. Each probability corresponds to one of five classes, in which we classify the input vectors. The last call is the actual creation of the model using the compile-function. Different loss-functions, optimizers, and metrics can be used. In our example we use the 'sparse\_categorical\_crossentropy' loss function, and as a metric the accuracy. The Adam

optimizer is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. (For details on Adam, see (Kingma and Ba, 2014)).

Step 3) The next step is to train the newly created model using the prepared test data and labels:

```
m.fit(train_data, train_labels,
    epochs=20,
    batch_size=32)
```

---

Calling the fit-function starts the training. In our case we use the train\_data and train\_labels to train the model. Epochs define how many times the model should be trained using the data set. The data is always given in a different ordering to the model. The batch\_size is the amount of samples which are feeded to the ANN in a single training step.

Step 4) After training, the test data is used for testing the accuracy of the model:

```
# predict the test data
prediction = m.predict(test_data)
# we count the correct predictions
correct = 0.0
# do the counting
for i in range(0, len(prediction)):
    if test_labels[i] ==
        np.argmax(prediction[i]):
        correct = correct + 1
print('Correct:', 100.0 * correct /
    len(prediction))
```

---

First, we call the predict function on the model to predict labels of the test\_data. After that, to check how accurate the prediction with the trained model is, we count how many times the prediction equals the correct label and calculate the correctness as percentage value. In the end, we output the value to the console.

Step 5) In the last step, we persist the model by storing it in the hierarchical data format (.h5).

```
# save the model to the hard drive
m.save("mymodel.h5")
# delete the model
del m
# load model from hard drive
m = load_model("mymodel.h5")
```

---

After persisting the model, it can be deleted from memory and later be loaded from the hard drive using the 'load\_model' function.



## 4 Our Cipher Type Detection Approach

In this section, we present our cipher type detection approach. First, we give a short overview of the MysteryTwister C3 challenge created by Stamp. Then, we discuss the cipher ID problem as a classification problem. After that, we present our cipher detection ANN in detail (input/hidden/output-layers, features, training and test data).

### 4.1 The MTC3 Cipher ID Challenge

MysteryTwister C3 (MTC3) is an online platform for publishing cryptographic riddles (= challenges). In 2019, Stamp published a cipher type detection challenge<sup>2</sup> on MTC3, named “Cipher ID – Part 1”. The detection of the cipher type of an unsolved ciphertext is a difficult problem, since a multitude of different (classical as well as modern) ciphers exist. E.g. in the DECODE database (Megyesi et al., 2019), there is a huge collection of (historical) ciphertexts of which we don’t know the (exact) type of cipher. Without knowing the type, breaking of such texts is impossible. Thus, a first cryptanalysis step is always to determine the cipher type. Different metrics, like text frequency analysis and the index of coincidence are helpful tools and indicators for the type of the cipher.

The MTC3 challenge is based on the aforementioned problem of often not knowing the type of ciphers of historic encrypted texts. The term “Cipher ID” refers to the type of used algorithm, or its “identifier”. In the challenge the participants have to identify different ciphers that were used for encryption of a given dataset of 500 ciphertexts, where each is 100 characters long. The goal is to determine the type of cipher used to encrypt each message. The following ciphers were used exactly 100 times each: simple monoalphabetic substitution cipher, Vigenère cipher, columnar transposition cipher, Playfair cipher, and the Hill cipher. The English plaintexts are randomly taken from the Brown University Standard Corpus<sup>3</sup>. The set of provided ciphertexts is shuffled.

### 4.2 Cipher ID as a Classification Problem

The general idea is to treat the detection of the cipher type as a classification problem. Each type of

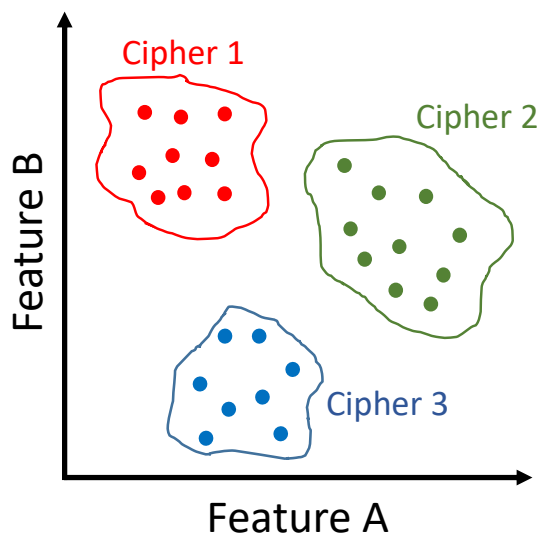


Figure 3: Ciphertexts (dots) in a multidimensional feature space. Classified into three cipher classes (red, green, blue)

cipher is regarded as a disjunct class, hence, there is a monoalphabetic substitution class, a Vigenère class, etc. Figure 3 depicts the general idea. In the figure, two feature dimensions (A and B) are shown. Based on the cipher’s characteristics, features have stronger or weaker influence on the output. Examples for features are the frequency of the letter ‘A’ or the index of coincidence. The colored dots (red, green, and blue) represent different ciphertexts. The dots are surrounded by a line showing the classes (or ciphers) each ciphertext belongs to.

With Stamp’s challenge, we have 5 different classes, one for each cipher type. The ciphertexts’ features are given as input vectors to an ANN which then classifies the text into one of the aforementioned classes. As output, the ANN then returns the ID of the detected cipher.

### 4.3 A Cipher ID Detection ANN

In the following we discuss the development of a cipher ID detection ANN based on the steps introduced in Section 3.2. Since it is a trivial step, we omit the persisting step (Step 5):

#### Step 1: Loading/preparing training/test data

To train an ANN a sufficient amount of training and test data is needed. In the case of the cipher ID detection ANN, ciphertexts of the types which should be detected are needed. Therefore, we first implemented all 5 ciphers in Python. We also created a Python script which extracts random texts

<sup>2</sup><https://www.mysterytwisterc3.org/en/challenges/level-2/cipher-id-part-1>

<sup>3</sup>Brown University Standard Corpus of Present-Day American English, available for download at <http://www.cs.toronto.edu/~gpenn/csc401/aires.html>

from a local copy of the Gutenberg library. Using this script, we can create an arbitrary amount of different (English) plaintexts of a specific length. After extracting a sufficient amount of plaintexts of length 100 each, we encrypted these with the ciphers – always using randomly generated keys. We created different sets of ciphertext files with different amounts of ciphertexts for each cipher (1000, 5000, 50000, 100000, and 250000). Thus, the total amount of ciphertexts provided to the ANN is a multiple of 5 of those numbers.

Since the ANN is not able to work on text directly, the data has to be transformed into a numerical representation. Our first idea was to directly give each letter as a number to the network, thus, having a feature vector of 100 float values. As this lead to a poor performance of our network we began experimenting with different other features, i.e. statistical values of the ciphertext. The next step shows our features and the overall ANN.

**Step 2: Creating a model** We experimented with different features as input values as well as with different amounts of hidden layers, widths of hidden layers, activation functions, optimizers, etc. We here now present the final ANN setup which performed best in our tests.

We use the following features:

- 1 neuron: index of coincidence (unigrams)
- 1 neuron: index of coincidence (bigrams)
- 26 neurons: text frequency distribution of unigrams
- 676 neurons: text frequency distribution of bigrams

Thus, the ANN has an input layer consisting of a total of 704 input neurons. After that, we create 5 hidden layers, where each layer has a total of

$$\left\lfloor \frac{2}{3} \cdot \text{inputSize} + \text{outputSize} \right\rfloor = \left\lfloor \frac{2}{3} * 704 + 5 \right\rfloor = 474 \quad (1)$$

neurons. Since we have 5 classes of cipher types, the output layer consists of five output neurons, each one for a specific cipher type. In Python, we created the network with the following code:

---

```
# sizes of layers
inputSize = 704
outputSize = 5
hiddenSize = 2 * (inputSize / 3) +
```

```
outputSize
# create ANN model with Keras
model = keras.Sequential()
# create input layer
model.add(keras.layers.Flatten(
    input_shape=(inputSize,)))
# create five hidden layers
for i in range(0, 5) :
    model.add(keras.layers.Dense(
        (int(hiddenSize)),
        activation="relu",
        use_bias=True))
# create output layer
model.add(keras.layers.Dense(
    outputSize,
    activation='softmax'))
```

---

The type of the hidden layer's activation function is 'relu' and the output layer's activation function is 'softmax' (see Section 3.1).

**Step 3: Training the model** We trained different configurations of our model with different amounts of ciphertexts. We used different sizes of training data sets and obtained the following results (output of our test program) with our best model:

```
Training data: 4,500 ciphertexts
Test data: 500 ciphertexts
– Simple Substitution: 87%
– Vigenere: 75%
– Columnar Transposition: 100%
– Playfair: 80%
– Hill: 32%
Total correct: 74%
```

```
Training data: 24,500 ciphertexts
Test data: 500 ciphertexts
– Simple Substitution: 88%
– Vigenere: 54%
– Columnar Transposition: 100%
– Playfair: 93%
– Hill: 64%
Total correct: 79%
```

```
Training data: 249,500 ciphertexts:
Test data: 500 ciphertexts
– Simple Substitution: 97%
– Vigenere: 63%
– Columnar Transposition: 100%
– Playfair: 99%
– Hill: 70%
```

Total correct: 86%

Training data: 499,500 ciphertexts:

Test data: 500 ciphertexts

- Simple Substitution: 99%
- Vigenere: 63%
- Columnar Transposition: 100%
- Playfair: 97%
- Hill: 67%

Total correct: 87%

Train. data: 1,249,500 ciphertexts:

Test data: 500 ciphertexts

- Simple Substitution: 100%
- Vigenere: 69%
- Columnar Transposition: 100%
- Playfair: 99%
- Hill: 78%

Total correct: 90%

The first two training runs were done in a few minutes. The third test already took about an hour on an AMD FX8350 with 8 cores. The last two tests took several hours to run. Since there is a problem with the CUDA support of TensorFlow with the newest Nvidia driver in Microsoft Windows, we could only work with the CPU and not with the GPU, making the test runs quite slow.

During our tests, we saw that with increasing the size of our training data, we could also increase the quality of our detection ANN. Nevertheless, the detection rate of the Vigenère cipher and the Hill cipher is too low (between 60% and 80%). In our first experiment, ciphertexts encrypted with the Hill cipher were only correctly detected by 32% and Vigenère was only 75%. We assume, that there is a problem for our ANN to differentiate between those two ciphers, since their statistical values (text frequencies, index of coincidence) are similar.

**Step 4: Testing and optimizing the model** For optimizing our model (with respect to detection performance), we tested other additional features provided to the ANN. Those features are:

- Text frequency distribution of trigrams
- Contains double letters
- Contains letter J
- Chi square
- Pattern repetitions

- Entropy
- Auto correlation

The text frequencies of trigrams had no noticeable influence on the detection rate, but made the training phase much slower, since  $26^3 = 17576$  additional input neurons were needed. Also, an equivalent number of neurons in the hidden layers were needed. Thus, we removed the trigrams from our experiment.

The "Contains double letters" feature did also have no influence. We additionally realized that the double letters are also detected by the bigram frequencies. Thus, we also removed this feature. Same applies to the "Contains letter J" feature. The idea here was, that the Playfair cipher has  $I = J$ , thus, there is no  $J$  in the ciphertext.

The chi square feature also had no influence on the detection rate.

With pattern repetitions, we aimed at giving the network an "idea" of the repetitive character of Vigenère ciphertexts. Unfortunately it did not help to increase the detection rate.

Entropy and auto correlation of the ciphertext were also given as features. Also no influence on the detection rate was realized.

Finally, we kept only index of coincidence on unigrams and bigrams as well as letter frequencies of unigrams and bigrams.

## 5 Cipher Type Detection and Solving Method for Stamp's Challenge

To actually solve Stamp's challenge this method brings together the following parts:

- **Cipher type detection ANN**
- Monoalphabetic substitution solver
- Vigenère solver
- Transposition solver
- Playfair solver

The method consists of the cipher type detection ANN and of solvers for each cipher despite the Hill cipher. The basic idea is the following: First, the set of ciphers is classified by the cipher detection ANN. After that, each cipher has been assigned a cipher ID. Since we know that only about 90% of the cipher types is classified correctly, we have to check each cipher type for correctness, in

order to reach a overall classification correctness of 100%. Thus, each ciphertext is then tested in a first run using its corresponding solver, despite the ciphertexts marked as Hill cipher. Hill cipher, especially in the case of a 4x4-matrix and ciphertext-only is a hard to solve cipher.

After that, all ciphertexts that could be successfully solved using the solvers are marked as “correctly classified”. The remaining ciphertexts, that could not be solved using the assigned cipher type, are then tested using the three other solvers. In the end, there should only be a set of 100 ciphertexts (in the case of the Stamp challenge), which cannot be solved with the four solvers. In that case, these 100 remaining ciphertexts must be encrypted by the Hill cipher. Since there is no good solver available for Hill ciphers, which performs much better than brute-force in the ciphertext-only case, this is very time consuming or nearly impractical for the Hill cipher.

**Execution time for classification with additional help of solvers** Let  $S$  be the time a single solver needs to test a given ciphertext, and this time is the same for all solvers. After  $S$  time is elapsed, the solver either produced a correct result or we stop it, since we assume that the solver is the wrong one for the specific ciphertext. In the case that we do not use the cipher detection ANN, we would need an overall of  $4 \cdot 500 \cdot S = 2000 \cdot S$  amount of time to test each ciphertext with 4 different solvers. If after executing all solvers exactly 100 unsolved ciphertexts remain, these are most probably texts encrypted using the Hill cipher. In that case, we solved Stamp’s challenge.

Now, let’s assume that testing a ciphertext using the ANN takes only a fraction of  $S$ , i.e. the classification time for a single ciphertext is  $T$  where  $T \ll S$ . In the real world, this is true since testing the 500 ciphertexts using our ANN only takes less than a second to be done. Generally, applying (testing) an ANN is much faster than training it. Since we know that the classification is only correct by about 90%, we have to test each ciphertext using the classified cipher type despite those classified as Hill cipher-encrypted. Let’s assume that about 100 texts are classified as hill cipher, thus about 400 ciphertexts remain to be analyzed. Since we know that 90% of those 400 texts are already classified correctly, 10% of those texts remain unsolved. These 10% plus the 100 hill-cipher classified texts have now to be analyzed

using all 4 solvers (this can be further optimized by only testing the remaining 10% with the three unused solvers). This leads to the following total amount of time needed for classification:

$$500 \cdot T + 400 \cdot S + 40 \cdot 3 \cdot S + 100 \cdot 4 \cdot S$$

which is  $920 \cdot S$  is so small that it can be left out of the calculation since  $T \ll S$ . Thus, we have a total execution time saving of about  $100\% - 100\% \cdot \frac{920 \cdot S}{2,000 \cdot S} = 54\%$  for the classification of the ciphertexts of Stamp’s challenge.

If we assume that a solver needs about one minute to successfully solve a ciphertext, using all solvers for testing would take about 2,000 minutes (about 33h). Using the ANN to reduce the amount of needed solvers, this time would now be 920 minutes (about 15h). Clearly, in the case of the ANN the time for training the network has also to be considered, which can also take several hours. Nevertheless, this time is only needed once, since the resulting ANN can be reused for classification tasks. The solvers could be executed in parallel, which further reduces the overall elapsed time.

## 6 Conclusion

This paper shows the current progress of our work in the area of artificial neural networks (ANN) used to detect the cipher types of ciphertexts encrypted with five different classical ciphers: simple monoalphabetic substitution, columnar transposition, Vigenère, Hill, and Playfair. For creation and training of an ANN consisting of five hidden layers, we used Google’s TensorFlow library and Keras. The goal of our initial research was to solve Stamp’s challenge (see Section 4.1), which required to determine the cipher type of 500 encrypted using the aforementioned five classical ciphers. The network was able to detect about 90% of the ciphers correctly. Detection rates for Playfair and Hill were too low to solve the challenge completely. Besides the creation of the ANN we also proposed a method (see Section 4) for solving the challenge using the ANN as well as different solvers, e.g. from CrypTool 2 (Kopal et al., 2014). Examples, how the solvers of CrypTool 2 can be used are shown in (Kopal, 2018). With the method, described in Section 5, about 54% execution time could be saved for solving Stamp’s challenge. Another part of this paper is a survey of the related work with respect to ANN and cryptanalysis of classic ciphers (see Section 2) and an intro-

duction into the topic for the HistoCrypt audience (see Section 3).

In future work, we want to extend our network (e.g. by using different ANN architectures) and method (e.g. by finding better features) in order to detect more different and difficult cipher types. We also want to use the methods in the DECRYPT research project (Megyesi et al., 2020) to further identify unknown types of several ciphers currently stored in the DECODE database.

## Acknowledgments

This work has been supported by the Swedish Research Council, grant 2018-06074, DECRYPT – Decryption of historical manuscripts.

## References

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.
- Ahmed J Abd and Sufyan Al-Janabi. 2019. Classification and Identification of Classical Cipher Type Using Artificial Neural Networks. *Journal of Engineering and Applied Sciences*, 14(11):3549–3556.
- B Chandra, P Paul Varghese, Pramod K Saxena, and Shri Kant. 2007. Neural Networks for Identification of Crypto Systems. In *IICAI*, pages 402–411.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIB-SVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.
- François Chollet. 2015. Keras: Deep learning library for theano and tensorflow. URL: <https://keras.io/>, 7(8):T1.
- Riccardo Focardi and Flaminia L Luccio. 2018. Neural Cryptanalysis of Classical Ciphers. In *ICTCS*, pages 104–115.
- Sam Greydanus. 2017. Learning the Enigma with Recurrent Neural Networks. *arXiv preprint arXiv:1708.07576*.
- Mahmood Khalel Ibrahim Al-Ubaidy. 2004. Black-box attack using neuro-identifier. *Cryptologia*, 28(4):358–372.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.
- Nils Kopal, Olga Kieselmann, Arno Wacker, and Bernhard Esslinger. 2014. CrypTool 2.0. *Datenschutz und Datensicherheit-DuD*, 38(10):701–708.
- Nils Kopal. 2018. Solving Classical Ciphers with CrypTool 2. In *Proceedings of the 1st International Conference on Historical Cryptology HistoCrypt 2018*, number 149, pages 29–38. Linköping University Electronic Press.
- Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Historical Ciphers and Keys. In *The 2nd International Conference on Historical Cryptology, HistoCrypt 2019, June 23-26 2019, Mons, Belgium*, pages 69–78.
- Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldspühl. 2020. Decryption of historical manuscripts: the decrypt project. *Cryptologia*, pages 1–15.
- Malte Nuhn and Kevin Knight. 2014. Cipher Type Detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1769–1773.
- Tariq Rashid. 2016. *Make your own neural network*. CreateSpace Independent Publishing Platform.
- G Sivagurunathan, V Rajendran, and T Purusothaman. 2010. Classification of Substitution Ciphers using Neural Networks. *International Journal of computer science and network Security*, 10(3):274–279.
- Ramon Wartala. 2018. *Praxiseinstieg Deep Learning: Mit Python, Caffé, TensorFlow und Spark eigene Deep-Learning-Anwendungen erstellen*. O’Reilly.



# Was it a Sudden Shift in Professionalization?

## Austrian Cryptology and a Description of the Staatskanzlei Key Collection in the Haus-, Hof- und Staatsarchiv of Vienna

**Benedek Láng**

Budapest University of Technology and  
Economics

1111, Budapest, Egry József u. 1

[benedeklang@gmail.com](mailto:benedeklang@gmail.com)

### Abstract

Cipher keys and code tables in the archives are easy to recognize, but hard to locate. The Staatskanzlei materials preserved in the Haus-, Hof- und Staatsarchiv in Vienna include half a thousand cipher keys and code tables in large cardboard boxes. This exceptionally rich and concentrated cryptologic collection sketches beautifully the four-hundred years of Habsburg diplomacy, as it was precisely the State Chancellery (and its predecessor organizations) that controlled Austrian foreign policy. The paper provides the first detailed description of this collection which is fairly exceptional not only for its historical significance, but also because historians rarely find such a large collection of keys in one single place.

### 1 Introduction

Encrypted Austrian despatches did not constitute a challenge to foreign deciphering cabinets (particularly to the English Deciphering Branch) in the first half of the eighteenth century. Following the mid-century, however, the situation changed dramatically, and Viennese messages started resisting adverse codebreaking efforts efficiently. Meanwhile, Austrians became famous for being able to decrypt French codes

(Ellis 1958, 73). This change had to do with the reorganization of the Austrian black chamber (the *Geheime Kabinets-Kanzlei*) under its newly appointed head, Baron Ignaz von Koch, and under the State Chancellor, Wenzel Anton Kaunitz, who initiated a complete turn in foreign policy, the so-called “diplomatic revolution.” (Andrew, 2018, 277-279). The quick growth in professionalization did not remain unnoticed in international diplomacy, Baron von Koch famously complained: “Unfortunately we have the reputation of being too skillful in this art and as a result, the courts which fear that we could be in possession of their correspondence change their [cipher] keys and each time adopt ones which are more difficult and troublesome to decipher” (Kahn, 1967, 163-165).

This was the era when large and professional codebreaking units were already in function all over Europe: the so called black chambers, which were larger and already more organized than a small group of talented mathematicians and their fellow clerks, that used to constitute the typical codebreaker units of the 16<sup>th</sup>-17<sup>th</sup> centuries (Leeuw, 2015). The Austrian black chamber was one famous actor in this chapter of crypto-history (Auer, 2015; Pecho, 2015; Walter, 2018).

Though the successful emergence of the Austrian black chamber might have many – both organizational and technical – aspects, this paper addresses one specific question: how far is this change reflected in the encrypting methods applied?

A practical way to answer this question is to review systematically the cipher keys and code tables of the Austrian empire. Fortunately, an exhaustive collection of them survived in the archives of the State Chancellery of Vienna (within the Haus-, Hof- und Staatsarchiv). The Staatskanzlei sources contain nearly half a thousand keys in rough temporal and alphabetical order classified in nine large boxes (ÖStA HHStA Staatskanzlei Interiora Chiffrenschlüssel Kt. 13–21.) out of which the first six – including 480 cipher keys and code tables – form the basis of this investigation.

This exceptionally rich collection sketches beautifully the four-hundred years of Habsburg diplomacy (Láng, 2018). The State Chancellery controlled foreign policy from the mid-eighteenth century until 1848. Its documents – kept in the House, Court and State Archives – ultimately incorporated the key collections produced by its pre- and parallel organizations, the Hofkanzlei (1527-1558), the Reichshofkanzlei (1558-1806) and the Österreichische Hofkanzlei (1620-1848), hence the researcher is privileged to find a complete documentation of Austrian crypto-history in one place (Auer, 2015; Fazekas, 1998).

A complete list and detailed description of the archival items discussed in this article are available in the Decode database (Megyesi et al., 2019).<sup>1</sup>

## 2 Boxes no. 13 and 14

The first part of the first large cardboard box (Kt. 13. Fasc 19) has not much cryptologic significance, it contains ceremonial documents. It is in the following fascicle (Kt. 13. Fasc 20, fols. 1-257 Benannte Schlüssel), where the real story begins. These nearly 500 pages primarily – but not exclusively – contain 16<sup>th</sup> and 17<sup>th</sup> century cipher keys. These keys originate most probably from the collections of the predecessor institutions of the State Chancellery. Apparently, no security measures aimed at the destruction of the keys, which did not seem necessary in the center of the Holy Roman Empire, just the opposite, they scrupulously preserved and classified them. Not surprisingly, the most typical structure appears to be a one-page system, consisting of a homophonic method with three or

four alphabets and a nomenclator table with approximately one hundred code words. Cipher alphabets are usually numbers, sometimes letter-groups. Inventive graphic signs are not missing either, particularly from the beginning of the period covered, but occasionally even from the 17<sup>th</sup> century.

On fols. 9-15 (and once again on fols. 15-19) one can see a rather rare homophonic system, where two letters correspond to each letter of the plain alphabet, while bigrams composed of letters appear also in the nomenclator table (this time the first letter is often capitalized). The list of nomenclators extend beyond four hundred. More than one hundred nulls – “errantes” as they are named in the system – are listed, and they have the same appearance as the other cipher letters, which make the system resistant. This table was used in 1568 by the Austrian ambassador next to the Pope: as usual in other collections as well, those systems seem to be the most advanced, that were used in communication with Italian political centers.

A beautiful example of a special subtype of nomenclators appears on fol. 28 used in relation to the Polish delegate (of which subtype one can see many more examples in the following boxes), where meaningful codewords, metaphors correspond to the name of political actors in the table. Dux is primus, Princeps is secundus, Pontifex maximus is bonus in a somewhat recognizable way, but when Imperator is gravis, Imperatrix is mens, and Palatinus Cracoviensis is species, the codebreaker quickly loses thread.

An exciting example of differentiating between encoding (chiffre chiffant) and decrypting (chiffre déchiffant) tables can be seen on fols. 32-33 and 34-36. The chiffre chiffant is arranged in alphabetic order, while the chiffre déchiffant is arranged according to the numbers of the cipher alphabet. This table is unfortunately undated, probably it survived from the 18<sup>th</sup> century, and it was used in relation to Berlin.

The following tables (that of Ogier Ghislain de Busbecq, the Habsburg delegate (of Dutch origin) to Suleiman between 1554 and 1562, and those Castaldo and Caraffa, all from the mid-16<sup>th</sup> century) share a preference towards sophisticated graphic signs in the alphabet, and towards no less sophisticated metaphors in the nomenclator table (Papa: Andromedes, Cardinalis: Antistes, Petrus Aldombrandinus: Amorius, Imperator: Benignus, Rex: Bruno, etc).

<sup>1</sup> <https://cl.lingfil.uu.se/decode/database/search>

A large double table system of Prince Eugene of Savoy from the years 1690 also appears to be in this collection: on fols. 90–103, one reads a well-structured encoding system composed of 2–4 numbers, while on fols. 104–123 the déchiffrant of the same system arranged according to the numbers – up to 2400.

As for the initial question of this article, the table of delegate Hoffman in London has special significance (fols. 152–157, Figure 1). It is a large system composed of numbers, clearly dated from the pre-Saatskanzlei period (1721, i.e. when Austrian codes were easy to break by the English codebreaking department). However, the system is so wide, composed of one thousand items and assigning three trigram homophones of trigrams to each syllables, that it is hard to imagine it was indeed vulnerable.

If one last example can be highlighted from the collection of this box, the choice would certainly fall on the 1583 table of Archduke Karl (fols. 243–244, Figure 2). This is a particularly beautiful system copied on parchment (as opposed to most of the others copied on paper): a three-page system with the usual preference towards beautiful graphic signs, combined with a few numbers.

Interestingly, the same fascicle is continued in the following box (Kt. 14. Fasc 20, fols. 259–429). The content, approximately 120 keys from the 16<sup>th</sup>–17<sup>th</sup>, and rarely from the 18<sup>th</sup> centuries, is not different either.

Besides the dominance of the usual one or two-page homophonic systems (named and dated in a larger proportion than in the previous box), fols. 132–135 should be highlighted, because these contain pre-printed lists comprising of an alphabet and a large list of nomenclators. The user, that is, the inventor of the cipher system, has no other duty than to fill in the sheet with randomly assigned numbers, giving birth to a new system. On these folios, one finds four different ways of filling in the table (i.e. four different cipher systems). One of these was used in relation to France, but it is not dated. On fols. 136–141, the same pre-printed tables remained empty. Such an automatized preparation of inventing new ciphers definitely marks an important moment in professionalization.

The 1570 system of Carolus Rym (fols. 291–302, Figure 3) is worth mentioning because of its use of nullities. As it was mentioned above, in the Austrian cipher systems there was a tendency to include all those *types* of symbols among nullities, which were otherwise used in the cipher,

in order to avoid that the codebreaker can easily distinguish between nulls, symbols standing for letters and nomenclators on mere visual grounds. In Rym's system, however, a new type of null is introduced: typical conjunctions in Latin language (quapropter, deinde, simulatque, quoniam) as well as a few average words (mandavimus, dedimus, renunciatum). Usually, such words may be left as cleartext in encrypted letters. Using them as nulls, is a clever improvement.

On fols. 311–313, one reads again a nomenclator table with metaphors, which allow mapping up a whole power and alliance system of Europe: Papa: pater, Imperator Carolus: dominus, Rex Francorum: patronus, Rex Angliae: theologus, Rex Poloniae: amicus, Eques: vacca, etc. The editor of the system did not lack sense of humor.

In the last part of the box, there is some numbering confusion. In an un-numbered fascicle (or again, numbered as 20<sup>th</sup>?), we find again ciphers up to the early 18<sup>th</sup> century on 43 folios, and this is followed by a last fascicle with ciphers and instructions on 16 folios, quite mixed in date and nature.

### 3 Boxes no 15 and 16

In the following unit of the collection, the landscape changes perceptibly. While one or few page homophonic tables dominated the previous boxes, and multi-page code-tables (where alphabets play only a minor role) played secondary role, here the typical 18<sup>th</sup> century genre of cryptology, the code-table booklets dominate the collection. The genre of cipher keys becomes more uniform as cryptology enters into a new phase of professionalization. Another change is that most often than not, a new text type is attached to the tables: the “Instructions.” While previous cipher keys were also often complemented with a few sentences that explained how they are supposed to be used, from the 18<sup>th</sup> century, separate two-page long instructions aim to help the user systematically.

The 15<sup>th</sup> box (fasciculus 21) starts with a large codebook containing an extensive four-digit system (fols. 3–14). However, the alphabetical order of the words and the sequence of the numbers grow parallel, which renders the otherwise strong, nearly 10.000-unit system vulnerable.

A smaller cipher table from 1750 follows (fol. 19), which was used in French relation. It shares the same strength and weakness as the previous one: it is a large, one-page table of nearly 2000 units, in which only odd numbers appear (even numbers are systematically nullities – as the separate French instructions explain on fols. 20-21), but again, the number sequence and the alphabetical order of the nomenclators coincide. Even though it is from the post-Staatskanzlei period (i.e. that is supposed to be very advanced cryptologically), this table rather demonstrates the usual law in history of science: evolution of the methods is not uniform.

Much more resistant is a French speaking system from Milano, that dates from as late as 1824 (fols. 38-47 and 48-53): it has two parts: chiffant and déchiffant: large format, multi-page booklets of four-digit numbers, with extensive instructions (*Remarques pour l'usage de ce chiffre*). Not only words, but usual word combinations are also encoded (such as “avec vous”; “à ces”) months, numbers, nations, cities, rivers and person names separately, the alphabetical order not following the numerical one.

This fascicule (the 21<sup>st</sup>) as well as the following (fasc. 22) contain a lot of similar tables, most of them from the second half of the 18<sup>th</sup> century, and most of them named after the ambassador who used it. Usually, their measures exceed one thousand items, but do not go above 10 000. Many of them are written in French, a feature somewhat surprising in the center of the Austrian empire. This analysis will skip them now, as they are not structurally different from those discussed above.

Box no 16 goes back in time: its first part contains undated cipher keys from the 16<sup>th</sup> and the first part of the 17<sup>th</sup> centuries. Leafing through these 16 folios with the well-known, mostly one page homophonic tables, the reader quickly gets to the time of Emperor Charles IV (starting from fol. 17): 1711-1740.

On fols. 19-20, for example, one can see a system, in which the chiffant and the déchiffant parts are already separated, but these are not yet codebooks, rather large homophonic sheets, incarnating the typical cipher key of the period directly preceding the professionalization turn, that arrive with the formation of the Staatskanzlei in the mid-18<sup>th</sup> century. These sheets (as those on fols. 22-23, fols. 24-26) together with Leopold I's ciphers (separate fascicule within fasc 23, fols. 1-29), and even

many from the time of Maria Theresa (1740-1780) (alt fasc 18/a: fols. 1-84) are typical for this transition period, easily distinguishable from the full-fledged codebooks contained by the previous box and discussed above. Contradicting our intuition, some of these keys belonging to the pre-codebook period are dated from 1752, and even from 1759 (Maria Theresa, fols. 19-22), which is a challenge to explain.

Fortunately, the box finishes with proper codebooks (fols. 61-84) from 1770.

#### 4 Boxes no 17 and 18

The shift in professionalization becomes complete in box no. 17.

Fascicule no 24 is the second part of Maria Theresa's cryptology (1740-1780). Cipher keys are always composed of three parts: the 2-4 page long instructions (such as on fols. 3-4), the chiffre chiffant using 4 four-digits in alphabetic order (fols. 16-29) and the chiffre déchiffant arranged according to the numbers (fols. 5-16). These large tables try to be inclusive as far as encrypted words, names and notions are concerned, they contain approximately 10 000 items, that is, ten times as large as the previously detailed one sheet homophonic tables (such as on fol. 60, which is clearly an exception in this box, true, it is undated). Instructions have a tendency to define nullities (*errantes*) in increasing sophistication. Most of them are in French in these times.

The next fascicule contains anonym keys from the time of Joseph II and Leopold II. Fols. 87-100 is a huge code-book from 1789, fols. 101-104, and 105-112 is another one from 1790, fols. 113-115, and 116-119 is a third one from 1792, all of them in French.

This is followed by a parcel containing the ciphers of Francis II (1792-1835). A 1803 key used in relation to St. Petersburg assigns characters to the comma, question mark, parentheses, and numbers that serve as special markers and they are meant to delete the previous or the following character (fols. 120-124 and fols. 126-131). This key is not entirely French anymore, it contains Latin and German words as well, probably with the intention of being as practical for letter writing (which often happened in a somewhat mixed terminology) as possible. The whole fascicule is composed of such booklets, sometimes even bound in

beautiful paper binding (fols. 194-8, from 1812, Déchiffrant pour la correspondance militaire).

And finally, box no. 18 (fasc. 25) is a collection of ten claves (fols. 1-144). The first is an un-named, relatively small (one sheet with three digit numbers), probably early system (fols. 1-4).

This is followed by several multi-page booklets (separating the encoding and decrypting parts), with four digit numbers and with instructions – these times in German (fol. 38-40; 58-60; 78-80, 102-104, 115, 124-126, and 132-135). Alphabets are not separated anymore from the table, letters appear among the codewords, double letters and other characters. The very last one, the tenth cipher has instructions (fol. 140), but the tables are half empty. It is prepared with the words on a few pages, but the cipher characters are not assigned, the system remained unfinished (fols. 139-142).

## 5 Conclusions

What kind of answer can be given on the basis of this methodical analysis to the initial question? As for the dramatic change taking place in around in 1742-4 under the leadership of Baron von Koch, as a result of which Austrian ciphers started to resist the codebreaking attempts of English cryptanalysts, the results are ambiguous.

On the one hand, one can plausibly argue that important changes took place in these years, this is when the Staatskanzlei was formed, which took over the tasks of its predecessors. Only a few keys and code tables survived in the collection, that was used in relation to London, and many of these are not even dated. In general, however, comparing the complexity of the pre-1742 keys with the keys of the Chancellery dating from the second half of the 18<sup>th</sup> century, one can say that there was really a change. The majority of the former keys are composed of 1000 items, usually numbers from 1 to 999, and these are complex homophonic tables with nomenclatures. The majority of the post 1742 keys, however, are code books, several page long leaflets, usually composed of 10 000 items (four digit numbers) complemented with professional instructions.

On the other hand, there are too many exceptions from this rule. There are several huge Austrian codebooks already from the pre-1742 period (including for example one from 1721, London), which make the impression of being

very hard to decrypt, and there are also many one-page homophonic tables from the post 1742 period, which seem to be easy to solve (including one for example from 1750, France). Having reviewed a large number of materials (almost 500 keys and codebooks), one can only claim with reservations, that a dramatic technical improvement was introduced in those years.

It is logical to suspect, nevertheless, that something else was really improved with the professionalization of the Chancellery. It is perhaps not – or not only – the cipher systems on a technical level, but rather their application: more care was paid when using the given cipher systems. It is not an over-interpretation of the archival material to suppose that scribes were following the “instructions,” the descriptions attached to the keys, more carefully, and thus gave birth to better encrypted messages. Besides the systematic introduction of these “instructions”, a consequent differentiation between chiffant and déchiffrant tables was also introduced (chiffant being alphabetically arranged, while the déchiffrant numerically arranged), which allowed a more practical use of the ciphers, and gave less temptation to arrange cipher keys horizontally or vertically in a way that made them vulnerable. But again, all this happened gradually in the previous and following decades, and not exactly in 1742.

## Acknowledgments

This work has been supported by the Swedish Research Council, grant 2018-06074, DECRYPT Decryption of historical manuscripts.

I thank the Österreichisches Staatsarchiv, Haus-, Hof- und Staatsarchiv for granting the permission to reproduce the manuscript copies.

I thank István Fazekas and Géza Pálffy for calling my attention to the particularly rich collection of cipher keys and code tables discussed in the paper, Karl de Leeuw for many advices and lively discussions on the significance of the conclusions of the paper, Anna Lehofer for the meticulous work of preparing and uploading the copies of the keys into the database, and finally Beata Megyesi for making this research possible.

## References

Christopher Andrew. 2018. *The Secret World: A History of Intelligence*. Yale University Press, New Haven.



- Leopold Auer. 2015. Die Verwendung von Chiffren in der diplomatischen Korrespondenz des Kaiserhofes im 17. und 18. Jahrhundert. In Anne-Simone Rous and Martin Mulsow eds. *Geheime Post Kryptologie und Steganographie der diplomatischen Korrespondenz europäischer Höfe während der Frühen Neuzeit*. 153-170. Duncker & Humblot, Berlin.
- Chiffrenschlüssel, Österreichisches Staatsarchiv, Haus-, Hof- und Staatsarchiv, Staatskanzlei Interiora, Kt. 13–18. For a list and description of the cipher keys, see: the Decode database: <https://cl.lingfil.uu.se/decode/database/search>
- Kenneth Ellis. 1958. *Post Office in the Eighteenth Century: A Study in Administrative History*. Oxford University Press, Oxford.
- István Fazekas. 1998. Az Osztrák Állami Levéltár nyomtatásban megjelent segédletei. (The printed assistances of the Austrian State Archives) *Levéltári Közlemények* 69: 195–219.
- David Kahn. 1967. *The Codebreakers – The Story of Secret Writing*. Macmillan, New York; revised and updated edition: 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. New York: Scribner.
- Benedek Láng. 2018. *Real Life Cryptology: Ciphers and Secrets in Early Modern Hungary*. Amsterdam University Press, Amsterdam.
- Karl de Leeuw. 2015. Books, Science, and the Rise of the Black Chambers in Early Modern Europe. In Rous and Mulsow eds. *Geheime Post*. 87-102.
- Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Ciphers and Keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt19*, Mons, Belgium. 2015.
- Carolyn Pecho. 2015. Der Habsburger-Code. Chiffrierte Briefe von Erzherzog Ferdinand an Erzherzog Leopold während des Erbfolgekrieges um Jülich-Kleve als Versuche der Gemeinschaftsstiftung (1609–1610). In Rous and Mulsow eds. *Geheime Post*. 137-152.
- Maren Walter. 2018. Ein Maulwurf in Wien? Informationssicherheit, geheimdiplomatische Maßnahmen und Wissensgenerierung während der Vorverhandlungen des Westfälischen Friedenskongresses 1643–1644. In Guido Braun, ed. *Diplomatische Wissenskulturen der Frühen Neuzeit. Erfahrungsräume und Orte der Wissensproduktion*. 161–176. De Gruyter, Berlin.

Sum chifri = 1000															
1. 2. 3.				1. 2. 3.				1. 2. 3.				1. 2. 3.			
<b>A</b>				<b>C</b>				<b>E</b>							
ab	151	41	40	ca	58	142	42	eb	68	208	45				
ac	122	44	41	cc	72	146	43	ec	209	219	46				
ad	196	48	42	ce	65	151	44	ed	221	227	118				
ae	127	53	42	ch	84	174	59	ef	113	183	207				
af	208	58	43	cha	140	250	203	eg	155	771	211				
ag	145	62	45	che	319	226	124	eh	225	210	119				
ah	223	67	42	chi	381	258	127	ei	215	156	62				
ai	224	71	46	cho	405	270	206	ek	194	56	63				
aj	169	75	46	chu	397	61	210	el	490	59	177				
ak	178	78	42	ci	193	57	482	em	502	137	478				
al	296	196	204	ck	179	66	483	en	480	213	72				
am	60	201	205	cla	417	488	73	eo	472	223	74				
an	53	206	209	cle	504	495	75	ep	407	147	481				
ap	179	211	480	cli	233	505	489	eq	418	151	484				
aq	406	218	76	clo	142	505	496	er	773	128	485				
ar	416	224	77	clu	222	411	492	es	607	129	487				
as	491	229	486	co	197	415	78	et	69	423	217				
at	518	234	488	cra	224	419	215	eu	73	459	218				
au	232	141	490	cre	146	449	216	ew	595	495	491				
ax	546	145	497	cri	232	424	498	ex	610	487	225				
ay	503	150	506	cro	771	539	231	ey	680	504	493				
az	587	154	228	cru	755	497	144	ez	601	553	499				
<b>B</b>				cs	701	486	157	<b>F</b>							
ba	597	54	229	ca	62	480	155	fa	70	490	593				
be	524	72	507	cu	121	427	148	fb	56	420	143				
bf	49	76	145	cq	133	417	143	fc	50	428	152				
bh	120	197	156	<b>D</b>				fd	118	418	148				
bi	59	200	142	da	47	414	144	fe	113	418	148				
bla	63	207	147	de	61	415	152	fi	48	485	146				
ble	586	212	672	dh	57	436	671	fla	755	482	670				
bli	608	217	753	di	139	475	755	fle	677	547	151				
blo	742	225	754	do	77	477	673	fli	611	472	674				
blu	770	134	636	dra	64	410	762	flo	621	475	675				
bo	619	198	687	dre	620	478	761	fo	537	529	685				
bra	678	313	688	dri	609	409	764	fra	515	531	670				
bre	734	314	690	dru	694	420	693	fre	525	476	763				
bri	693	315	691	ds	772	422	774	fri	404	402	676				
bro	702	317	766	dt	780	595	775	fro	427	725	689				
bru	770	318	769	du	679	596	780	fru	419	591	698				
bs	771	321	697					fs	495	594	699				
bt	772	322	777					ft	505	599	712				
bu	773	323	778					fu	426	743	716				
bz	779	326	779												

Figure 1. The first page of the cipher key of delegate Hoffman in London, 1721. ÖStA HHStA Kt. 13. fol. 152.



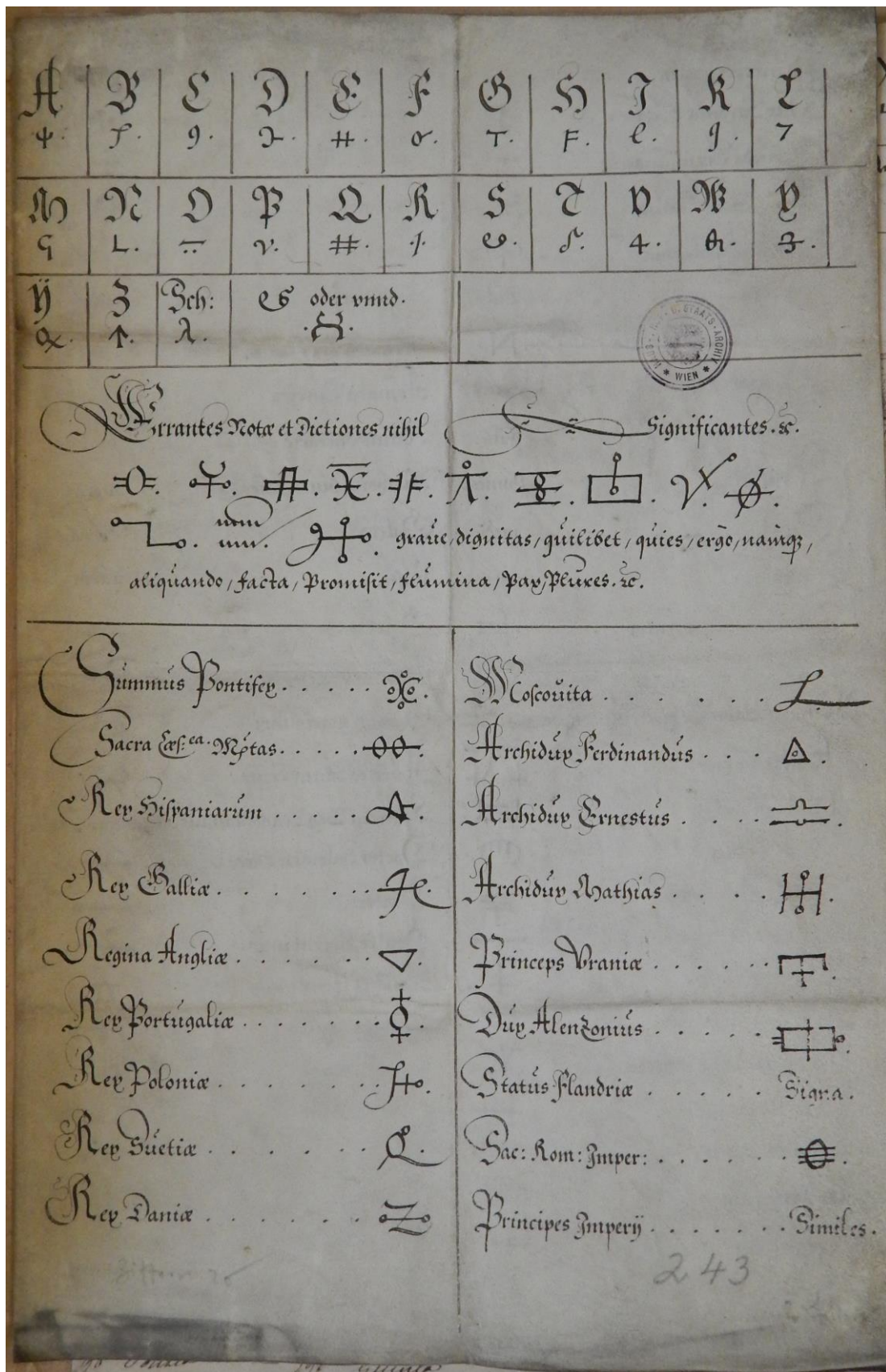


Figure 2. The table of Archduke Karl, 1583. ÖStA HHStA Kt. 13. fol. 243.





# Solving a Tunny Challenge with Computerized “Testery” Methods

George Lasry

The DECRYPT Project

george.lasry@gmail.com

## Abstract

The Lorenz SZ42, codenamed Tunny, was a teleprinter encryption device used by Germany during WW2 for strategic communications. Its successful cryptanalysis at Bletchley Park (BP) provided the Allies with high-grade intelligence about several fronts, as well as for the preparations for the D-Day landings. The story of Tunny’s code-breaking and Colossus is well known, following the declassification of the General Report on Tunny in 2000 (Good et al., 1945), and the publication of several books (Reeds et al., 2015; Gannon, 2014; Copeland, 2010; Roberts, 2017; Mayo-Smith, 2014). The work on Colossus and other machines was carried out in the Newmanry, under the leadership of the mathematician Max Newman.

The work of the Testery, the other Tunny section at BP, is less known. Named after his commander, Major Ralph Tester, the Testery was responsible for the development and application of hand methods, that complemented the work of machines like Colossus. For some reason, the report on the Testery was not declassified until 2018. Following its recent release, it is possible to fully assess the achievements of the Testery cryptanalysts and their key contribution to BP’s success against Tunny (Testery, 1945).

The work described in this article is an attempt to determine whether the Testery manual methods can be mechanized with modern computing. The author was able to automate some of the techniques and partially automate some others. With these techniques, the author also succeeded in recovering the key settings and the plaintext of two Tunny challenge messages.

This article is structured as follows: In Section 1, a functional description of the Lorenz SZ42 is given. In Section 2, the contents of the Testery report are surveyed, highlighting the parts that reveal new information. In Section 3, the primary techniques for the cryptanalysis of Tunny are described. In Section 4, a Tunny cipher challenge is introduced. In Section 5, new automated or partially automated versions of the Testery manual methods are described, as well as how they were used to solve the Tunny cipher challenge. In Section 6, the main results of this study are summarized.

## 1 The Lorenz SZ42 (Tunny)

The history of the Lorenz SZ42 and the details of its design and functioning are documented in the references (Reeds et al., 2015; Gannon, 2014; Copeland, 2010). In this section, only a brief functional description is given.

The Lorenz SZ42 is a teleprinter encryption device. It encodes Baudot teleprinter symbols that consist of five impulses. Each impulse can have one of two states. It can be active, denoted as *cross* according to BP terminology, or **x**. Or it can be inactive, denoted as *dot* or **•**. The Baudot alphabet, as well as BP’s notation for the Baudot symbols, is given in Table 1.

The Lorenz SZ42 functions as a Vernam device. It applies an XOR addition (denoted as  $\oplus$ ) to encrypt plaintext Baudot symbols. The effect of the XOR operation on a pair of impulses *a* and *b* is described in Table 2. The XOR operation can also be applied to a pair of Baudot symbols with five impulses each. In that case, it is applied sequentially one impulse at a time. An example is given in Table 3. It should be noted that adding (using an XOR addition) a symbol to itself, results in the symbol **•••••** which has only dots, as illustrated in Table 4.

The Lorenz SZ42 generates a keystream *K* of pseudo-random symbols and performs an XOR ad-



Symbol	BP Notation	Meaning in Letter Shift	Meaning in Figure Shift
•••••	/	null	
••••x	E	E	3
•••x•	4	carriage return	
•••xx	A	A	-
••x••	9	space	
••x•x	S	S	,
••xx•	I	I	8
••xxx	U	U	7
•x•••	3	line feed	
•x••x	D	D	Who are you?
•x•x•	R	R	4
•x•xx	J	J	BELL
•xx••	N	N	,
•xx•x	F	F	%
•xxx•	C	C	:
•xxxx	K	K	(
x••••	T	T	5
x•••x	Z	Z	+
x••x•	L	L	)
x••xx	W	W	2
x•x••	H	H	£
x•x•x	Y	Y	6
x•xxx	P	P	0
x•xxx	Q	Q	1
xx•••	O	O	9
xx••x	B	B	?
xx•x•	G	G	&
xx•xx	5 or +	figure shift	
xxx••	M	M	.
xxx•x	X	X	/
xxxx•	V	V	;
xxxxx	8 or -	letter shift	

Table 1: The Baudot Teleprinter Alphabet

$a$	$b$	$a \oplus b$
•	•	•
•	x	x
x	•	x
x	x	•

Table 2: The XOR ( $\oplus$ ) Operation

K	•xxxx
G	xx•x•
$K \oplus G$	x•x•x

Table 3: XOR ( $\oplus$ ) on the Symbols K and G

G	xx•x•
G	xx•x•
$G \oplus G$	•••••

Table 4: XOR ( $\oplus$ ) on the Same Symbol

dition on a stream of plaintext  $P$ , producing the ciphertext  $Z$ , as described in Equation 1, the encryption formula.

$$Z = P \oplus K \quad (1)$$

Encryption and decryption are implemented identically. This is possible since adding (XOR) the keystream  $K$  to the ciphertext  $Z$  cancels out the effect of the keystream  $K$  originally added during encryption, as shown in Equation 2, the decryption formula.

$$Z \oplus K = (P \oplus K) \oplus K = P \oplus (K \oplus K) = P \quad (2)$$

As a result, two machines using identical settings can communicate properly, one side encrypting plaintext and transmitting ciphertext, the other receiving and decrypting the ciphertext.

The functioning of the Lorenz SZ42 is illustrated in Figure 3 in the Appendix. The keystream  $K$  is generated by a set of twelve wheels, divided into three functional groups:

- **Five  $\chi$  wheels,  $\chi_1$  to  $\chi_5$ :** Those wheels have 41, 31, 29, 26, and 23 pins, respectively. Each pin can be set to either an active (cross) or an inactive (dot) state. The  $\chi$  wheels regularly step after the encryption (or decryption) of each symbol. The stream of Baudot symbols generated by the five  $\chi$  wheels is denoted as the  $\chi$  stream.
- **Five  $\psi$  wheels,  $\psi_1$  to  $\psi_5$ :** Those wheels have 43, 47, 51, 53, and 59 pins, respectively. Each pin can be set to either an active or an inactive state. Their stepping is governed by the motor wheels. Either all five  $\psi$  wheels step or none of them steps. The actual stream of symbols generated by the  $\psi$  wheels is denoted as the  $\psi'$  stream. It differs from a theoretical  $\psi$  stream, that would have been generated if the  $\psi$  wheels always stepped. The  $\psi'$  stream is an extended version of the  $\psi$  stream, with symbols duplicated at positions where the  $\psi$  wheels did not step.
- **Two motor or  $\mu$  wheels,  $\mu_1$  and  $\mu_2$ :** Wheel  $\mu_1$  has 61 pins, which govern the stepping of wheel  $\mu_2$ . If the current pin of wheel  $\mu_1$  is active (cross), wheel  $\mu_2$  steps. Wheel  $\mu_2$  has 37 pins, and if its current pin is active, all five  $\psi$  wheels step. The single-impulse stream

generated by wheel  $\mu_2$  is denoted as the *base motor stream*. In later models of the Lorenz SZ42, various *motor limitations* were introduced to reduce the number of *motor stops*, that is, positions where the  $\psi$  wheels are not stepping.<sup>1</sup>

The keystream  $K$  consists of the (XOR) addition of two streams,  $\chi$ , and  $\psi'$ :

$$K = \chi \oplus \psi' \quad (3)$$

Therefore:

$$Z = P \oplus K = P \oplus \chi \oplus \psi' \quad (4)$$

We define  $D$ , also known as the *dechi stream* (or simply, the *dechi*), as:

$$D = Z \oplus \chi \quad (5)$$

The term *dechi* originates from the fact that we are removing  $\chi$  from the ciphertext  $Z$ , by adding it so that the original contribution of  $\chi$  cancels out:

$$D = Z \oplus \chi = P \oplus \chi \oplus \psi' \oplus \chi = P \oplus \psi'. \quad (6)$$

If we add  $\psi'$  to both sides of  $D = P \oplus \psi'$ , it also follows that  $P = D \oplus \psi'$ .

## 2 The Testery Report

Each of the two main Tunny sections at BP – the Newmanry and the Testery – wrote a report. The General Report on Tunny with Emphasis on Statistical Methods (GRT) was written in 1945 by I.J. Good, D. Mitchie, and G. Timms from the Newmanry (Good et al., 1945; Reeds et al., 2015). It was declassified in 2000. It describes in detail the work on codebreaking machines such as the Heath Robinson and Colossus in the Newmanry, as well as their mathematical and statistical foundations. While it provides a wealth of technical information, the GRT is not easy to read, and its structure does not always follow a clear logical flow.

<sup>1</sup>A motor limitation forces the  $\psi$  wheels to move at positions where the base motor stream is a dot, and the  $\psi$  wheels would otherwise not step. Motor limitations are governed by a combination of one or more impulses from the  $P$ ,  $\chi$ , and  $\psi'$  streams, at previous positions. The combined effect of the  $\mu$  wheels (the base motor stream) and of the motor limitations is denoted as the *total motor stream*. A description of the various types of motor limitations may be found in (Reeds et al., 2015, Chapter 11B, p. 13). As described in Section 3, most attacks against Tunny take advantage of skewed statistics at motor stop positions. Motor limitations are intended to reduce the number of motor stops, making cryptanalysis more challenging.

REPORT ON TUNNY (MAJOR TESTER'S SECTION).	
CONTENTS.	
Chapter I	Introduction
Chapter II	Prehistory of Tunny
Chapter III	The Tunny Era
Chapter IV	The Q&P Era
Chapter V	Cribbs
Chapter VI	Discovery and Treatment of Depths
Chapter VII	Keybreaking
Chapter VIII	De-X breaking
Chapter IX	The work of Room 40
Chapter X	Decoding and Issuing
Chapter XI	Mathematical Techniques and Theories involved in Testery Cryptography.
Chapter XII	Fish Organisation
Appendix 1	Coalescence
Appendix 2	Follow on Messages
Glossary	

Figure 1: Testery Report – Table of Contents

In some places, it lacks some details or examples necessary to understand some of the key points.

The GRT only briefly mentions the work and methods of the Testery. Those hand methods are also described (in even less detail) in testimonies and books written by Testery veterans (Roberts, 2017; Mayo-Smith, 2014).

The Testery report was not declassified together with the GRT back in 2000. A possible reason is that the Testery report may have contained sensitive information about methods still in use after WW2. This contrasts with a statement by D. Mitchie, one of the GRT authors, who was allowed to review the Testery report, and wrote that “*a good deal of [the Testery report’s] content is directly inferable from other sources, including General Report on Tunny. The full Testery report amplifies this knowledge.*” (Copeland, 2010, P. 246)

In 2017, at the NSA Symposium on Cryptologic History, the author met the GCHQ historian, Dr. Tony Comer, and inquired about the possible release of the Testery report. In July 2018, the author was pleasantly surprised to receive the following email from Dr. Comer:

“*The mills have been grinding slowly since my return from the Symposium, but I am delighted to say that we have transferred HW 25/28 to TNA.*”<sup>2</sup>

The author soon after traveled to Kew and made of copy of the report at TNA. The report is named

<sup>2</sup>The National Archives, Kew, UK.

*Solution of German Teleprinter Ciphers ("Testery") Linguistic Methods* (on its cover) and also *Report on Tunny (Major Tester's Section)* in the table of contents page (Testery, 1945). It contains 229 pages. It has twelve chapters, two appendices, and a glossary. Figure 1 shows the table of contents.

From a study of the report, it indeed emerges that most of the contents of the Testery report generally appears in the GRT, but often with significant differences. In contrast with the GRT, the Testery report follows a clearer presentation flow. The cryptanalytic methods are better explained, with useful examples, which were missing from the GRT. For example, a detailed example is given in the Testery report to illustrate the indicator method (Testery, 1945, Chapter II, section 10), and *Turingery*, Turing's method for extracting the  $\chi$  wheel patterns from a keystream, is described in detail (Testery, 1945, Chapter III, section 2). As a result, the text of the Testery report is more readable. To quote Jim Reeds, one of the authors of the modern edition of the GRT: "*The Testery report was written by grown-ups.*"<sup>3</sup>

More importantly, the Testery report contains new material or material that was only briefly mentioned in the GRT. A major example is a description of the operational process for finding cribs to help with cryptanalysis in (Testery, 1945, Chapter V). This work was carried out by Sixta, BP's traffic analysis section. The Sixta History report, like the Testery report, was declassified only in 2018, long after the release of the GRT (Sixta, 1945). It is possible that both the Testery and the Sixta reports were kept classified for a longer period in order not to expose GCHQ's traffic analysis techniques and the role of traffic analysis in assisting cryptanalysis.

From the cryptanalytic perspective, the primary addition of the Testery report, compared to the GRT, consists of more detailed material about the Testery hand methods (Testery, 1945, Chapter VIII), mainly:

- **$\psi$ -Setting:** Finding the  $\psi$  wheel settings (i.e., the  $\psi$  wheel starting positions) from a dechi stream, when the wheel pin patterns are known.
- **$\psi$ -Breaking:** Finding the  $\psi$  wheel pin patterns from a dechi stream, when the patterns are unknown.

While both topics are covered in the GRT (Chapters 28B and 28C), Chapter VIII of the Testery

report methodically lays out the rationale for the manual methods, and the various techniques involved. Those techniques take advantage of some features of the German teleprinter language, which may vary according to the traffic on the specific link. For example, some Tunny links may use a different sequence of Baudot symbols to mark a full stop or a comma (e.g., by adding extra spaces or duplicating special symbols such as Figure Shift or Letter Shift). Other techniques rely on German operator habits and mistakes, such as sending messages in depth (encrypted with the same key settings) or "go-backs" – repeating the last 100 symbols of a message at the beginning of the next one (Testery, 1945, Chapter VIII).

The work of the Newmanry on the Colossus, and the role of Colossus in the history of modern computing, have taken center stage in the story of Tunny codebreaking at BP, leaving the achievements of the Testery in the shadow. The Testery report provides a more balanced view, highlighting the critical role played by the Testery in the daily recovery of keys and settings. Repeatedly, when the Germans introduced new security measures, such as motor limitations, the Testery was able to diagnose the modifications and find ways to circumvent them. In other cases, the Testery was often able to find and correct errors in the dechis, the output of the Newmanry's machines. As an illustration of the operational success of the Testery, the following figures are given for April 1945 : Out of 806 dechis provided by the Newmanry, 88% (707) were broken by the Testery. (Reeds et al., 2015, p. 243) (Good et al., 1945, p. 261)

### 3 Tunny Codebreaking Overview

A complete decryption of the machine and of the hand methods for the cryptanalysis of Tunny, as well as of the multitude of codebreaking scenarios the methods cover, is outside the scope of this paper and may be found in the Testery report and the GRT (Testery, 1945; Good et al., 1945). This section focuses on the main cryptanalytic scenarios.

The most challenging scenario is *breaking*, when the wheel patterns are unknown, there are no messages in depth (encrypted with the same key settings), and no crib is available. Historically, codebreaking for such a scenario included the following steps:

- The recovery by the Newmanry of the  $\chi$  wheel patterns, using the *rectangling* method devel-

<sup>3</sup>Private conversation with the author, 2019.

oped by Bill Tutte, and later performed with the help of Colossus (Reeds et al., 2015, p. 110-112). After the  $\chi$  wheel patterns had been recovered, the dechi stream  $D = Z \oplus \chi$  was produced by the Newmanry.

- The recovery by the Testery of the  $\psi'$  stream from the dechi stream  $D$ , using hand methods. From  $\psi'$ , the  $\psi$  wheel patterns could be recovered.
- The recovery of the motor wheel patterns, also by the Testery, from the  $\psi'$  stream.
- The decoding of the ciphertext (by the Testery).

For *setting*, when the wheel patterns are known, but the wheel starting positions are unknown for a specific ciphertext, the process was simpler. Historically,  $\chi$ -setting was done by the Newmanry, and the settings for the  $\psi$  and motor wheels were recovered by the Testery.<sup>4</sup>

In case two or more messages in depth were available, their plaintexts could be recovered using linguistic methods, and using segments of plaintext, the keystream  $K (= Z \oplus P)$  could also be extracted. From  $K$ , the wheel patterns were then recovered by the Testery.<sup>5</sup> A similar process was possible with the help of a long-enough crib.

But unless a crib is available, or plaintext can be extracted from depths, all attacks – for setting and breaking – rely on a major weakness of Tunny, which is described here.

We first introduce the notation  $\Delta$ , or *differenced* stream. A differenced stream consists of adding (using XOR addition) to each element of an original (undifferenced) stream the value of the element right after it. Differencing can be applied to a single impulse, or to a stream of Baudot symbols, impulse by impulse. An important characteristic of a differenced stream is that if two consecutive symbols are identical, their differenced value is the symbol  $\bullet\bullet\bullet\bullet$  (all impulses inactive).

In Section 1, Equation 6, it was shown that the dechi stream  $D = Z \oplus \chi = P \oplus \psi'$ .

We analyze here the frequency distribution of the symbols in the dechi stream  $D$ . The  $\psi$  wheels may or may not step after each encryption (or decryption), but if they step, they all step together.

<sup>4</sup>For some motor limitations (or if no motor limitation was used), the setting of the  $\psi$  and motor wheels could also be performed using the more advanced models of Colossus.

<sup>5</sup>*Turingery*, a method for extracting the  $\chi$  patterns from  $K$ , was developed by Alan Turing.

When the wheels do not step (i.e., a motor stop), the corresponding symbol of  $\psi'$  is duplicated, and as a result, the corresponding  $\Delta\psi'$  symbol has only dots ( $\bullet\bullet\bullet\bullet$ ). This means that at positions where there is a motor stop,  $\Delta D = \Delta P$ . Therefore  $\Delta D$  at motor stops has the same frequency distribution as for  $\Delta P$ .<sup>6</sup> Even though the symbols of  $\Delta D$  are (roughly) randomly distributed at positions the  $\psi$  wheels step, overall, the frequency distribution of  $\Delta D$  symbols is skewed toward the frequency distribution of  $\Delta P$  symbols.

This important characteristic can be exploited for setting the  $\chi$  wheels. While the plaintext for a given ciphertext is unknown, it is possible to compute the distribution of the *expected* differenced plaintext  $\Delta P$ , using a corpus of the language (e.g., from prior decryptions). To set the  $\chi$  wheels, we search for the  $\chi$  wheel positions that result in the symbol distribution of  $\Delta D = \Delta Z \oplus \Delta\chi$  being as close as possible to the expected frequency distribution of  $\Delta P$  in the reference corpus. A similar methodology can be applied for  $\chi$  breaking, to find the optimal  $\chi$  patterns, so that the resulting  $\Delta D$  best matches the expected distribution of  $\Delta P$  in the reference corpus.

Due to the limits of WW2 technology, those techniques could only be applied to a pair of impulses at a time, e.g., impulses 1 and 2 (the so-called  $\Delta_{1+2}$  method), rather than to all five impulses at the same time (Reeds et al., 2015, p. 110-112).

The same characteristic of  $\Delta D$  can be used to recover  $\psi'$  from dechi, as described in Section 5.

## 4 A Tunny Challenge

In 2015, while working on the computerized cryptanalysis of Tunny, the author was able to find several original ciphertexts on the website of the late Tony Sale ([www.codesandciphers.org.uk](http://www.codesandciphers.org.uk)), as well as the relevant wheel patterns and settings. Those included settings and patterns used during WW2 in Tunny links like the one between Berlin and Rome, codenamed *Bream*. To further validate his new computerized methods, the author needed additional ciphertexts for which the patterns and settings were unknown. Frode Weierud, an expert on the history of cipher machines, provided the author with two ciphertexts of unknown origin, together with a set of wheel patterns that might have

<sup>6</sup>During cryptanalysis, the positions where there is a motor stop and  $\psi$  wheels do not step are unknown.

been used to encrypt the messages. Each ciphertext consists of approximately 5,500 symbols.

The author made several attempts to set the messages using the provided patterns without any success. Next, the author tried to set the messages using patterns found in Tony Sale's website, using a new method which he developed.<sup>7</sup> Setting was successful for the  $\chi$  wheels, using the Bream link  $\chi$  patterns from Tony Sale's website (the Bream patterns are given in an appendix at the end of this article).

However, all attempts to set the remaining wheels failed, using the Bream patterns and also trying various motor limitations. To make further progress, there was no choice other than to try and recover the motor and  $\psi$  wheel patterns, i.e., to perform motor and  $\psi$ -breaking instead of just setting. While the author had also developed new methods for motor and  $\psi$  breaking<sup>8</sup>, those require at least 10,000–15,000 symbols, many more than the 5,500 symbols in the challenge messages. No further progress could be made on solving the challenges until 2019.

## 5 Mechanizing the Testery and Solving the Challenge

The main Testery methods are based on the characteristic of  $\Delta D$ , as described in Section 3. Due to the  $\psi$  wheels often not stepping, there are numerous repetitions of consecutive symbols in  $\psi'$ , and as a result a high frequency of  $\bullet\bullet\bullet\bullet$  symbols (all impulses inactive) in  $\Delta\psi'$ .

Due to security measures introduced by the Germans (Reeds et al., 2015, p. 306), there is also a high frequency of  $\text{xxxxx}$  symbols (all five impulses active) in  $\Delta\psi'$ , at positions where the  $\psi$  wheels step.<sup>9</sup> Furthermore, the frequency of  $\Delta\psi'$  symbols with a majority of crosses (e.g.,  $\bullet\text{xxxx}$  or  $\bullet\bullet\text{xxx}$ ) is significantly higher than the frequency of symbols with only one or two crosses (e.g.,  $\bullet\text{xx}\bullet$  or  $\bullet\bullet\text{x}\bullet$ ). In addition, the probability for a  $\bullet\bullet\bullet\bullet$  symbol at positions where the  $\psi$  wheels are stepping is very low.

Historically, the work of the Testery started after receiving the dechi  $D$ , extracted from ciphertext by the Newmanry using mechanized methods. The Testery cryptanalysts tried various possible cribs  $P$  at different positions, examining the

resulting (putative)  $\Delta\psi' = \Delta D \oplus \Delta P$ . A putative  $\Delta\psi'$  mostly consisting of  $\bullet\bullet\bullet\bullet$  or  $\text{xxxxx}$  symbols, and the remaining symbols with a majority of crosses, was likely to indicate a correct crib guess. Still, there was always some probability for a wrong guess, especially if the crib was short. This process was labor-intensive and required extensive trial-and-error by the cryptanalysts, who had to memorize the full XOR addition table ( $32 \cdot 32 = 1024$  elements) to mentally perform XOR additions (Roberts, 2017; Mayo-Smith, 2014).

For  $\psi$  setting, a machine named *Dragon* was developed to “drag” a crib over the whole dechi stream (Reeds et al., 2015, p. 346). For  $\psi$  breaking, there was no other choice but to test cribs manually.

After positioning a likely crib, the cryptanalyst would then try to extend it by testing additional symbols inserted before and after the crib, and checking the resulting new putative  $\Delta\psi'$ . With a long enough-crib and from the resulting  $\psi'$  segment ( $\psi' = D \oplus P$ ), it was possible to recover the  $\psi$  patterns.

With modern computing, a more efficient process can be implemented. As part of this study, the author has developed a series of new algorithms, which partially automate the Testery manual processes, described in the following sections.

Crib P:	89MANNERN89UND89FRAUEN5
Dechi D:	RCPDIIJ/IYZLBMZQTSEUSX
$\psi'$ :	YRRRRRRRRYYYYYGGIIDI
$\Delta\psi'$ :	8////////8////K/M//KK
$\Delta\psi$ crosses:	500000005000004030044

Figure 2: Example of Crib Hit

### 5.1 Dictionary Search and Ranking

This new algorithm processes cribs taken from a large dictionary. A space is added before and after the crib, which is tested at all positions of the ciphertext. The results (the crib and their possible positions) are ranked using the resulting putative  $\Delta\psi'$ , taking into account the number of “good” symbols in  $\Delta\psi'$  such as  $\bullet\bullet\bullet\bullet$  or  $\text{xxxxx}$  symbols, and penalizing symbols with a small (non-zero) number of crosses. The ranked results are manually inspected, and the more likely ones entered into a database of crib hits. Figure 2 shows an example of a particularly good crib hit. In this example, the elements of  $\Delta\psi'$  have either no crosses, only crosses, or a majority of crosses (three or

<sup>7</sup>To be described in a separate paper.

<sup>8</sup>To be also described in a separate paper.

<sup>9</sup>To create a seemingly more random output  $Z$  as well as  $\Delta Z$ , each pin on a given  $\psi$  wheel was more likely to be followed by a pin in the opposite state.



four).<sup>10</sup> In a more typical case, there will be less "good" symbols, and the  $\psi$  wheels are likely to step more often.

The reason the results must be manually inspected is that the algorithm produces a large number of false crib hits, which must be filtered out manually based on the expected traffic contents, or adjacent crib hits. Also, there might be conflicting crib hits at the same position or overlapping.

## 5.2 Extending Matching Cribbs

A manual attempt is then made to extend the most promising cribbs, by guessing additional symbols at their beginning and at their end, so that the (longer) putative  $\Delta\psi'$  still has good characteristics. With a solid knowledge of the language and of the traffic contents, it is possible to extend the crib further so that a long stretch of  $\psi'$  can be obtained. Then, by removing repeated consecutive symbols from  $\psi'$ , it is possible to obtain the (unextended)  $\psi$  stream and from it to extract the  $\psi$  wheel patterns. Historically, the Testery cryptanalysts would first recover the  $\psi$  patterns as described here, and finally, the motor wheel patterns.

With the current Tunny challenge, due to the author's limited knowledge of the language and the lack of prior information about the traffic contents, he was unable to extend the cribbs enough so that the  $\psi$  patterns may be recovered.

## 5.3 Recovering the Motor Wheel Patterns

Instead of first recovering the  $\psi$  patterns, as it was done historically, the author had to develop an algorithm to recover the motor wheel patterns based on the crib hits in the database. This new method uses hillclimbing, and it searches for  $\mu_1$  and  $\mu_2$  patterns that generate an optimal motor stream. Such an optimal motor stream should maximize the number of motor stops at positions with  $\Delta\psi'$  being  $\bullet\bullet\bullet\bullet$ , and minimize the occurrences of motor stops at other positions (where the  $\Delta\psi'$  symbol has at least one cross). The  $\Delta\psi'$  symbols are obviously examined only at those positions covered by a crib that appears in the database of crib hits.

Using this new algorithm, combined with extensive trial-and-error to rule out some crib hypotheses and to test new ones, the author was able to recover the complete  $\mu_1$  and  $\mu_2$  patterns for the challenge. The  $\mu_1$  and  $\mu_2$  patterns turned out to be

minor variations of the  $\mu_1$  and  $\mu_2$  patterns for the Bream link. More importantly, it turned out that no motor limitation was used. A motor limitation would have made the recovery process more challenging.

## 5.4 Recovering the $\psi$ Wheel Patterns

Knowing the  $\mu_1$  and  $\mu_2$  patterns, and therefore all the positions where the  $\psi$  wheels step (or stop), allows for a more accurate assessment of potential cribbs, by applying stricter criteria for valid cribbs. Instead of relying on counting the proportion of "good"  $\Delta\psi'$  symbols as described above (such as  $\bullet\bullet\bullet\bullet$  or  $\text{xxxx}$ ), a valid crib should always result in a  $\bullet\bullet\bullet\bullet$  symbol in  $\Delta\psi'$  at motor stops, and in other symbols (with a very high probability) at positions where the  $\psi$  wheels step.

In addition to just ranking possible crib hits, it is now possible to rule out most of the wrong hits. This allows for better crib hits to be processed, in order to extend the crib, ultimately increasing the amount of recovered  $\psi'$  material. More importantly, since the  $\psi$  wheel positions are now known, it is possible to combine disjoint  $\psi'$  segments that have been recovered.

The author wrote a program to extract the  $\psi$  patterns automatically from such  $\psi'$  segments, also checking for possible conflicts. As a result of detecting some conflicts, minor corrections needed to be made to some matching cribbs (for example, a particular crib word turned out to be followed by a comma instead of by a wrongly guessed full stop period). The  $\psi$  wheel patterns for the challenge were successfully recovered, and surprisingly, they turned out to be the same as those of the Bream link.

## 5.5 Deciphering the Challenge Messages

Finally, with all the wheel patterns recovered, the ciphertext could be deciphered, and the first challenge message read. After formatting the plaintext, the deciphered message starts as follows:

KRIEGSMASCHINE AUF HOHER SEE DIE U.S.S. LINCOLN ANDRIAN KREYE ANFLUG AUF DEN FLUGZEUGTRAGER U.S.S. ABRAHAM LINCOLN. FUNFUNDNEUNZIGTAUSEND TONNEN ATOMGETRIEBENEN STAHL, DIE GROSSTE KRIEGSMASCHINE IN DER GESCHICHTE DER MENSCHHEIT.

It ends with the following text, which includes the crib *MANNERN UND FRAUEN*:

<sup>10</sup>As shown in Figure 1, in BP notation / represents the  $\bullet\bullet\bullet\bullet$  symbol, 8 represents  $\text{xxxx}$ , K represents  $\bullet\text{xxxx}$ , and M represents  $\text{xxx}\bullet\bullet$ .

JEDER QUADRATZENTIMETER AUF DEM SCHIFF  
HAT SEINE FUNKTION, JEDER FALSCHER SCHRITT  
KANN DAS WOHL DURCHDACHTE ZUSAMMENSPIEL  
VON SECHSTAUSEND *MANNERN UND FRAUEN*, SIEBZIG  
FLUGZEUGEN UND TECHNISCHES GERÄT FÜR  
MEHRERE MILLIARDEN DOLLAR AUS DEM TAKT  
BRINGEN.

The second ciphertext was successfully set<sup>11</sup> using the same wheel patterns (but different settings, i.e., different wheel starting positions). The plaintext was identified as an email in English sent encrypted from Frode Weierud to David Hamer. Its ciphertext is given in an appendix and its decipherment is left as an exercise to the reader, who is invited to send the solution to the author. The Bream patterns are also provided for reference.

## 6 Conclusion

The release of the Testery report has shed new light on the outstanding achievements of the Testery, using hand methods. The work on the mechanization of the Testery techniques and on solving the challenge has enabled the author to fully appreciate the ingenuity and creativity demonstrated by the Testery cryptanalysts. In addition, it is possible to assess the importance of the close cooperation between the Testery and the Newmanry, which was critical to making sure that BP's resources would be fully utilized, and large scale production of strategic intelligence from Tunny traffic could be achieved. Moreover, it is clear that the familiarity of the Testery cryptanalysts with the traffic they processed manually ensured that when the Germans introduced new changes and security measures, those could be promptly diagnosed by the Testery, and the cryptanalytic methods adapted to cope with those changes.

Another conclusion from this study is that the security of the system was greatly reduced by the fact that all five  $\psi$  wheels of the Lorenz SZ42 either step or stop together. If the  $\psi$  wheel motion had been implemented differently, the vast majority of the mechanized and manual methods developed at BP would have been rendered useless.

<sup>11</sup>The starting positions of the wheels were recovered, allowing for the message to be deciphered.

## Acknowledgments

This work has been supported by the Swedish Research Council, Grant 2018-06074, DECRYPT – Decryption of historical manuscripts. In addition, the author would like to thank Dr. Tony Comer for facilitating the release of the Testery Report, and Frode Weierud for providing the challenge messages and reviewing an earlier version of this paper.

## References

- Jack Copeland. 2010. *Colossus: The Secrets of Bletchley Park's Code-breaking Computers*. OUP Oxford.
- Paul Gannon. 2014. *Colossus: Bletchley Park's Last Secret*. Atlantic Books Ltd.
- Jack Good, Donald Michie, and Geoffrey Timms. 1945. *General Report on Tunny: With Emphasis on Statistical Methods*. Bletchley Park Report HW 25/4. Kew, London: U.K. National Archives.
- Ian Mayo-Smith. 2014. *Eavesdropping on Adolph Hitler: Deciphering the Daily Messages in the Tunny cipher*. Four Pillars Media Group, Connecticut, USA.
- James Reeds, Whitfield Diffie, and J.V. Field. 2015. *Breaking Teleprinter Ciphers at Bletchley Park: An Edition of I.J. Good, D. Michie and G. Timms: General Report on Tunny with Emphasis on Statistical methods (1945)*. John Wiley & Sons.
- Jerry Roberts. 2017. *Lorenz: Breaking Hitler's Top Secret Code at Bletchley Park*. The History Press, Stroud, Gloucestershire UK.
- Sixta. 1945. *The Sixta History*. Bletchley Park Report HW 43/82. Kew, London: U.K. National Archives.
- Testery. 1945. *Solution of German Teleprinter Cyphers (Testery) Linguistic Methods*. Bletchley Park Report HW 25/28. Kew, London: U.K. National Archives.

## Appendix – Functional Diagram

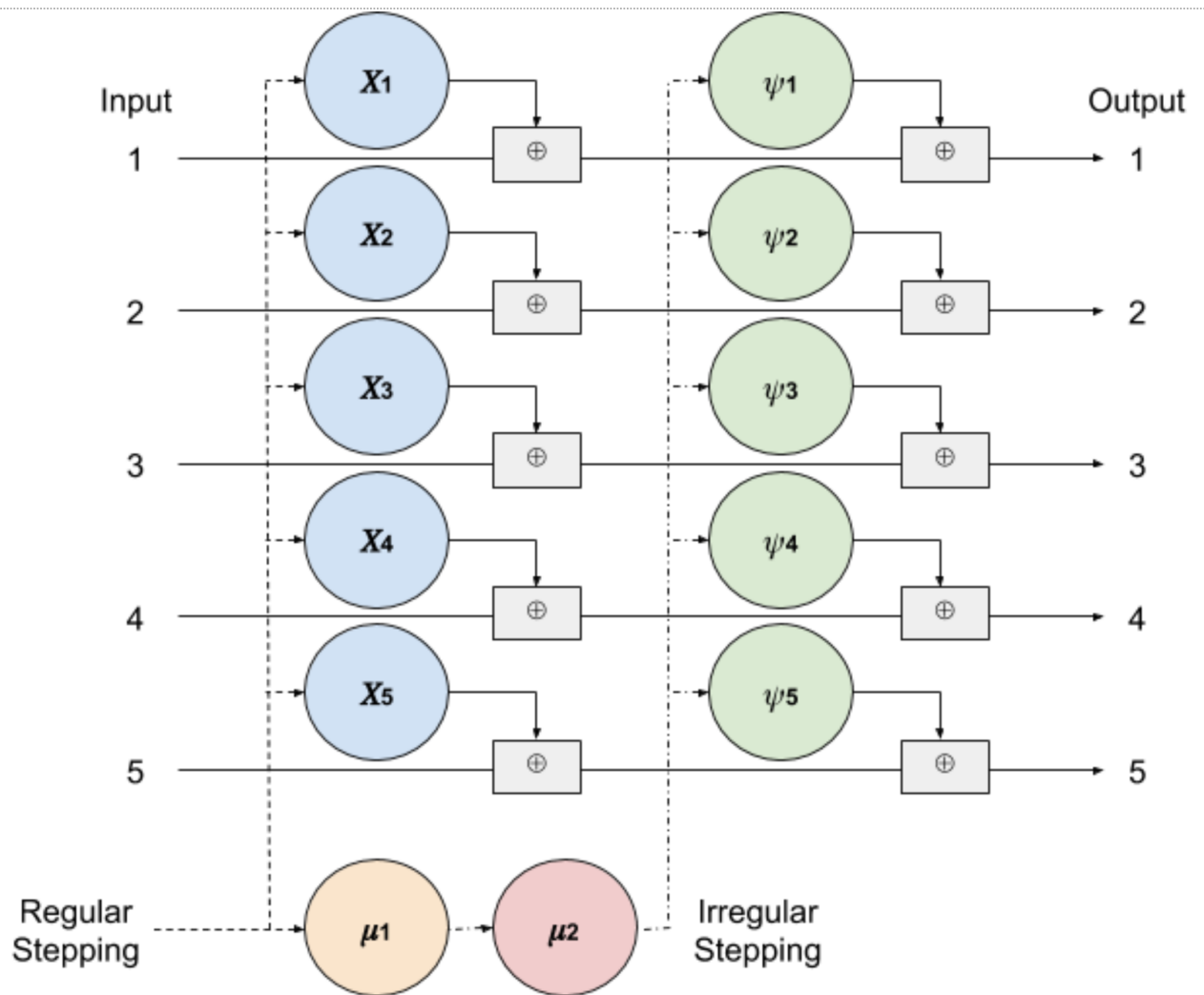


Figure 3: Tunny Lorenz SZ42 – Functional Diagram (Source: The author)

## 7 Appendix – Second Challenge Ciphertext

WGXXAQKQ9AT4RYAFD/I4SE8KNDaidNAYN8APH/MDHAOBM8Z9WRHHYBOZNOMPSVPSVSVJOMYKW+OJ4HJWFA9SRBU4FNUOMFLWFRM3JGG3JVKHTNZNF4BZWZT4ISZE3/9BMO+  
DYSYEGM/UGFWFEKSI8CGODGIGENS4HREOIJJDKIVJIMT4SB3BAW3+PQJBUC+OVLBET8HKFLKQV+SZD9BI9R4YDBO4OQVEE9WJXCXCGMV3VMB4SUOMY83TLTX8SNQK4PL8HJ  
J8A3JVAI98S8SOKCMI/HWV/UGYWC4PJVIEKGF+SWEXFBZHA9KWGI8BZKN3CCL/ASZSPXVNETXPIOF88P48DBMZBXUANHKGONLCU3V/F+SEORH+ORTATRU4T4WLQOEACGCY  
8LIA4+OFVUFUEB/E8+V+NERZVENEH/OKPUUEZJ/UJ+3I4EFTRK8G3ACATNVZNW3FZRS8FNRCOY3QACY9WPFPRKSILOIGMMUY+YESI/KFJ+UR/FOBL/G/JUGG3TCDJGJVEVLJI  
IUNXCTQLOTAIDIAIPNN4GX4HC/UCPMBWLXTUHGXYCK++XZDRNV19KRIAWAND+ML/XFPABBLWFBJCX4BZKMS3EOQEAQDO/F4OBR9V8EJFHFRQPETVKE8VF/4SWB93C/8CMIYI  
4C+EW88OXOOAUZXLFPQFM9LCM++TEEMAYBNJH3YIRB9Z4S+OVATHNQSF4WB9UAS94SBMTLZO/FSGHDBFN+DAYYXFVQ4AU+NWSJXBQKNHZ+8/KC8ZPHO+UNDPBSJG/8DRHS  
ZY+/DE/VPJWVPCW8G9U/+9VUI+IIWO9AEBX/TZTRFCN3QW3XV8KLFUWAI4N/OACEUPTALHTTL4JFYDQVQVMTL8EOEVIYWAS+3APEXPNUBGWO3NFJYVYGAHOL/ELMY43S3PI  
K3XYVXXHH+943JV3N+RHLZRRHRESBLYJA/KKLYCQX4TVGCVHFNHYHB+UPZ/KB/+M/SE/URJIRIECHTGJDI98R9HIVXZZSZ4MOKADBH/Z8/DDRQ4DENYU8PPJZ3IA3OAWIN  
D8OS/8+ZCJZWJOGEL3XMBLZRHBM/YQKYBU8ZL+38L4JA4/O+RBHD4T8MYS9RW3WKE4/C8/GTL84PAB4DFHOUNVHSC8KZSS8AUYXXZT4RN8SFMFJMJLFTAJOKW4YINSKUX4W  
KIOOMQSMGSC9GI+JGLETHABDRERXCKQEUNWUEG/HOJPBZ9P4/GZDMYQ9PVWT//MLRF08AISMPDTRPRM9F8TPGEUL4UZ4AVYBPLQL/4PBI18PRCYTJ+ZMEEGBNVPGDPPER  
RBFL/GM+KKVSTLIPYEWL/NQAPFT4ABZBZLQ4WY9KF+4X4MVZCBI8MBZFFCNDGC/94/ZH9QG+CLSPJDEMBEAWKVG+AZGN4OC3++LBMDIU/CUDIA+YP8YTDU43VUHC8C/3GLCI  
BNU8U4GWROJB/HEVZBLYX9W8NP+YVK9X9TJAWV4HUT/E4UJISKG8SOZUKNRDYBAM/+PLH3ANS4LF4P3UEFHGJ4U+K3XTLFEYAAA4XNHK4ISPSYEE4LDU38VHF3LVPFODVJ  
+LBXP/GOP+R3F8H9++4LNLITJ4NXH+LM+DM3OE34/DLM//T3S++AZR44AOGLYRA3TWCWAX+OPZUIJ+VLKPWR/+EF+/EH8TJDC+TIIKX38JSJ+HIDTUILOPNEXIG+CRGKJGT  
YJIA++CIDPYU3MEY3/8JYGTXCCT/RNEJGT/S+UHJSGVXFDP/IPQUUHNCAZDKO9MLXXCKV/8L3ROOKTEDNHBL4ZPQPS3EVHJW9YJUDP9CS/HP/4/GKAIMT9KKHBB3KNRQOFTY  
RZ9S3RGMFFDE84SHYIWE9PC993SYJ4C4XWHCLJ/+D3E4JVC3YEDG+ELGJHRD3PQZB/IRGOWA3DTIQSK3NDJXAGRAFCH/P4E+983MLLRORW9+FPGCVRUVHGKBB/RFY+9GVI  
TA+4FNRDI/XDFT44IYWLN/LU9H+KBMJ9098L8RYBI9U+/TTZZXKNOETXAW9YUQL4GAEBYBC4NK4FIWEIWK93FPPHJ94U+WZFUW8Z2QPBYI/49+IPZSQKLWB8QTUM/QX+RUU  
PRXVMD+VRB/ZNWJVSWE9GHKIDV8OQAZFWOW+SV+P+BCA9RIMEVGIXPP+VHBK4M3NBJBA3USVC+/ZHXPKKIHH3CVBQXKKIRBQNYQBPJSPJ4EOJK3OOAHV4J3+N9LVECCNG  
TXVHY+RXRQ3XB/J3CY9/IR4/M/AYV/YQBHQPUXGW4BI9XCEOHKSJYVBEDDNNXNBK3L84TITEMRZPHMX93RYXPJZYDTHVVJTP8XSVS9A8LWAKLUAAKXOJ/FXTOLZ/UDNVH9SG  
BWHH+R8LENIABBBQPKBC9WUE4IT4LPX3E9/PXU3QY+FDCEXOT+VTESNPF+T8P3NH3AG8+B9M4ZRSRYRHXGJJEV4MMWCMKNV9ACRFBQ89QJAPSVX8+EECQ3DYWGMI/G+SG4  
XXTQOI+TKCUIYYTPA+J4JOPM99CAH4TNZHZ/GAYYGLLT/V4DWC4MWYBHCJWMDT2CZ8GJV88TMCXHTDZSZP3OIR+U8UUFJNPPZCBLIROG/8NLZ/JPORMUDDT44U9VZGV/HJBP  
MBOHL+3B48IA9/PUJQIAUTOYOURQD9R/LFQ+APUQJYS+OQ3WTD8ZDPHFITYIWPV4I/FRWXE8LMU9KUDJ9ZYMQMGNOXYCZWFLSJ49WDBHBJVJT3YFNSMJTRJDMKSOUXPCVK  
+P+3XS89GWN4UPGQ4HCCQW4YKQL9AZLT+YON/QABQUTK+ZH3/GBQUOTSSKVHGX8JPDWB9BIDBLOBGMKJN8K9ZUUNNSHRWZ/3TCXKBLZLEK9I/AUOJ8JBTBHQWR3OQLDV+PCJ  
9HMQIVPWXO+JR3V3EF8GTXTP8MR/AAI4ZPFCAGLZ4YZP8NL+9PSSP+TS+P/+QHUEFJR8V3N8AX49E84QK43JYQT88U8UIC+UIOWSHW89AICN+FYGMS9GDFUYWM4G//TILINC  
CCYWXCRDX9EBLE/8ER/IJGLDDOJNESZTJJDPMWXZ9LLLM+XMTBV8BMXNCWZF9LF+DW9WHRNFATBW+M+UWPHDUWUZPMMZHDA3OXAYBV8FJ8++4RPWUTMQRUREREZUAK4ZGV  
P+JDZEHL3XNATMVNEHZPRJFU9MGGKPUFYC9N9PY9VFSB8+3PLUYR/M+TKMR/JX9HMKUL8PZINB+H8TS/W/Q33YJB+GUCFTXHPX+E/9DGMTLCS48NFKGB34CIX4VEUGAMAQL  
K4M3D+UBLVG8TU3W+TY9XP4JC3MLG9MX+KPKF9/QEDPS9NYHHZVL4ECYZ+Y43ASND++GQQJXLVGGOEYZF+XK8XMK9TN+YAMYE9LQNNY48SSW9HX4SSIOQXM9XBSIFW4FMQNL  
SZ3GB8P/JDN3TP+XX/8SJGKDLU3H9KQTELCWUX4RBI9NO3DRXMXISPG9DQJ4IKEBDAYWTBI43/4DENES8VOQAHQXWDQHHECD49ZPLPMBQ4+WOW9NW98VQHRKDV/OZHCSI  
YKNDXGXMY3BWHHJ4BFLKL+GZDYBAUVPXGNMHSVQHDVEU/ZULZABCTRR9Q9PPK+SMCBH+TISZZJ/F4/ARA4QL+FOBYU/S+J9WBOCY3YX/NFIMB9IFXP/D99CSWOA9BOOPDZDV  
L8IUSNKL+STSGFV/KPXGLOGRJRJ3XCS8HJHEJOPRXZRS9XZL+ORVOJAP4P/4GM4BW4Y8L/JVEQZYU/R4+P4NRX4GWHOGS3JBAAGVRRMI4YEEPVJZHYZR9JZ/8ZRNC/L/  
LOKWL8OW/M3VAUZXI4VJKLBPMW90DLAAXCCZVGMXQ3I+JNJLRDD/3HUMMGSEIBFUTEUGP9X/BXJ/3+VZL3/MR4MGP43IRRRITLPLKFKRM/DC4ZNERP+DIOYB+B3N9+NK8M8W  
RBB3SSQDCIJQN+NUOE9E9Z+O/SSCEM4FDKEMCJZ4TCTIO/WRYELWIYJNW8BSVFMNVFBGTR8Q9JS3HWCXNRORWL3M+BFBAZTR3X3OHNDV+LSQUANASYOFKXBPVZCCOCL+XB  
LWKT+NMNU44TVBCFEURH4LNGWQYHMQTHSNSTGZFNFGNF4ESNLHFCBTRATPTK/WK3BWVTJ4+AGQZRA4C/TK39Y/HIE3MU8UNRURUG9DKMPHNVPFCOQRMJSA93H4HZN8MO/AY  
PPTX8KN3C4YLCPW8M9LSH9YAPMCP/9MULFWG/4SPHLKUSEIP+VX8IXY38HSHJ4M8VRB4JLME4S3A9SPBQMGTCW+EU+BRX8JJ39LFAAGNSQGFEDH8OUHU9QGEPESLHXHJ+JGH4  
+CPT38JVEDQXLAKKKRLZ448OBRR+NMW3GV/ZWN9XXFRLSYKZY+JVJTOQZUUNYX/RGSB9DFEPB4NAVKNP3NAFMQB/GWW4R83PDZS3HUI89MFRZ+QMDAITAQPMTKMEKPBQNJ  
4VOAHQNA/CHTXRQKY49XKTFULWUD84VTHWJFSV/KL8I3S3Z4R8YLLN8+ATWCLPECZNRDGLXGK+APCHGYJGLENGIEYYLBR4UT+RTAKNEKUXHUICMUNNXN9AVDUNEXGUDN//  
PPWAZR8KNP3UXQQ93FYQW3IHKVPUGFNFT9BIDE9KPP/9HGT+L3FG//YCLXYDYXP83LWU98UTOLVBJPJGQMOIGVZMRPLJ4SL4XMOQMNBNI48PZO/F/LEQVCVF+CXRUERGOGXN  
/YG+ISV8INOQMFTNLGG9VYX8TZN4P+DDJOLNTAX/ZNYBFV8PVWUVQBR9/JGSYI/YFCXKFYU/48YP8MMUIZHZZEMVMMNZ3OS3BRCAXFNNHMKU4OVD9SXT3C9N94GFISDWHJ+  
IWHKU4HHSIQI4PGY3SPCXW4Z4IPMNNFTGTCI4HVXPWJIIJY4WN3KQIMDIZLZMTCBBSQSCBMT/AHY8NCDVCM4XQ4BX+9S3BUYDDZAYJZX8W4+8IVH3LSFS4QWB3ZNMJUY4/  
MQCQDQFSPTA9QFK4FKSL9K/ELBCRFOWNPGWVT9T9TOHNSD4FQAMAMDFSD9CTCR3W3BQGJL8C+FTESHKJ08E+UJ3ZVZDOSIG938N9MF/PTPSKXZICSN/EEDC84++TM4XGWZCF  
Z9AGF8PMUNNTMYCFT8PQSKZG378XHPFJDZISVEHZ9J9MIREP8K+CC+8/8Y3W1MLUQ+/GQQSZEJ9JUZ9KDW9B9JWNTKHLX9GIGQ/BAJJRZGLTZN8ELB8JFLLMF

Bream wheel patterns:<sup>12</sup>

$\chi_1$  ●●●XXXX●●●XX●●●X●XX●X●XX●X●XXXX●  
 $\chi_2$  XX●●XXX●XX●●●X●X●XX●●●X●●●XXXX●  
 $\chi_3$  ●●XXX●XX●●X●●●●XXX●●XX●XX●●XX  
 $\chi_4$  ●●XX●●X●XX●●X●XX●●X●●XXXX  
 $\chi_5$  ●X●●●X●XX●●●●●●XXX●XXXX●  
 $\psi_1$  ●●X●X●X●X●X●X●X●X●XX●XX●X●X●XX●XXX●●XXXX●●  
 $\psi_2$  ●●X●XX●X●X●X●X●X●XX●XX●X●X●XXXX●●●●●XXXX●X●XX  
 $\psi_3$  X●X●X●X●X●X●X●X●XX●XX●X●X●XXXX●●●XXXX●●XXXX●XX●●X●X  
 $\psi_4$  X●X●XXX●X●X●X●X●X●XX●X●●●XXXX●XX●XX●XXXXX●X●X●●●●X●  
 $\psi_5$  ●X●X●X●X●XX●●●X●X●●XXX●XXXX●XX●X●●●●X●●●X●●X●●XX●XX●XX●X●X●X  
 $\mu_1$  XXX●X●XX●●XX●●XX●●●XXXX●X●XX●XX●●●XX●●●●XXXX●XX●XX●●●XX●●●●X  
 $\mu_2$  X●XXX●X●X●X●X●X●XXX●X●X●X●X●X●X●X●

<sup>12</sup>While the Bream patterns for the  $\chi$  and  $\psi$  wheels were used to encrypt the challenge original plaintexts, the  $\mu$  wheel patterns used for encryption are slightly different from the  $\mu$  wheel patterns given here.

# Transcription of Historical Ciphers and Keys

Beáta Megyesi

Department of Linguistics and Philology

Uppsala University, Sweden

beata.megyesi@lingfil.uu.se

## Abstract

Historical ciphertexts and keys contain a wide range of symbols from digits and letters from known alphabets to various types of graphic signs. To be able to study ciphertexts and keys empirically in large(r) scale, consistent representation of the symbol systems used in ciphers is inevitable. In this paper, we present guidelines for transcription of ciphertexts, keys and cipher-related cleartext documents. We hope that the guidelines contribute not only to the systematic and consistent text representation across ciphertexts and keys, but also help in more accurate and reliable transcriptions.

## 1 Introduction

Usually, the first necessary, albeit time-consuming and probably least fun step in attacking a hand-written cipher is the conversion of the cipher image into a machine-readable format. The goal is to represent the ciphertext image as a text file, allowing various types of analyses. The process of converting the ciphertext image into a text document is called transcription. And the first, often cumbersome, albeit fun step in this process is the identification of the symbols, also called glyphs, in the ciphertext. During transcription, we need to identify and uniquely represent each symbol type by investigating the glyphs and their context. For this purpose, we usually create a transcription scheme, where each symbol type has its own and unique text representation. Then, we transcribe each glyph in the ciphertext according to our transcription scheme. We type in all glyphs, symbol by symbol, as they appear in the ciphertext in the text file.

The ciphertext alphabet might contain a wide range of symbols, such as letters, dig-

its, punctuation marks, or other graphic signs. The identification of the symbol set is often unproblematic if the ciphertext is built up of some standard symbol set(s), such as digits (0-9), the Roman alphabet (a-z, A-Z), or a combination of the two. These symbols can be typed in easily and fast on a keyboard, and saved as a text file using some character encoding, such as a Unicode (UTF-8) format. However, ciphertexts often include a palette of symbols from various alphabets (Roman and Greek), graphic signs (Zodiac symbols or alchemical signs), diacritics, and punctuation marks (dots, commas). Nice examples of ciphertexts with mixed symbol sets is the Borg<sup>1</sup> (Aldarrab, 2017) and the Copiale<sup>2</sup> (Knight et al., 2011) ciphers with available transcriptions stored in the DECODE database (Megyesi et al., 2019).

The identification of the cipher alphabet is far from easy as symbols might look similar to each other although they represent different plaintext entities. Symbols can have diacritics, dots or other marks attached to them, or these can be unintentional ink spots or dirt that should not be part of the transcription. While the encoded sequences in ciphertexts are usually meticulously written and often segmented glyph by glyph to avoid any kind of ambiguity for the receiver to be able to decode the content, sequences of connected symbols or sloppy handwriting are also frequent. In addition, the ciphertext might be embedded in cleartext, i.e. texts written in a known natural language.

Presumably, the transcriber strives for a simple and fast transcription process and chooses a mnemonic, easy to remember transcription scheme. He/she makes decisions about how to represent each symbol type, and

---

<sup>1</sup><https://cl.lingfil.uu.se/~bea/borg/>

<sup>2</sup><https://cl.lingfil.uu.se/~bea/copiale/>



how to transcribe each glyph, space, punctuation mark, along with margin notes, catchwords, and cleartext sequences. While the transcriber freely designs his/her transcription principles, we get a large variety of transcriptions which makes it hard to comparatively study these historical sources.

The aim of this paper is to present transcription guidelines to represent ciphertexts and keys with a great variation of symbol system in a text format. First, we give an overview of the basic principles for transcription, then we describe the guidelines for the transcription of ciphertext images and keys, followed by cleartext images representing the original plaintext or a text related to the ciphertext, for example in a letter correspondence. Lastly, we conclude the paper.

## 2 Transcription of Ciphers

Transcription is the systematic representation of language in written form, an effort "to report—insofar as typography allows—precisely what the textual inscription of a manuscript consists of" (Meulen and Tanselle, 1999). In what follows, we apply the terminology concerning writing systems as defined by Sproat (2006).

Not surprisingly, there is no standard convention for the transcription of manuscripts due to the great variety and heterogeneous nature of historical written sources (Meulen and Tanselle, 1999). Transcription is always based on the transcriber's interpretation, and can be said to be non-neutral given that the transcriber needs to decide upon how detailed or close the transcription should be to the original image (Rosenberg, 2006). Various considerations can be taken to decide which reading is the most likely to the original, and how detailed the transcription shall be. Such details can include the distinction of letters (e.g. *i* with or without a dot), capitalization and graphic emphasis such as section titles, abbreviations in original and their expansions, gaps and damages, as well as the scribe's self-corrections, in particular insertions, replacements and changes (Cipolla, 2018).

The level of the detail required depends on the aim (Koester, 2010). Even in a single manuscript written by one scribe, the shape

of the letters can vary greatly, and deletions, additions, notes, marks can occur in many different ways which influence our interpretation (Driscoll and Pierazzo, 2016). Knowledge of the historical context, the culture and society in which the manuscript was produced is also relevant. A high level of granularity in the transcription provides insight into the practice of copying and its procedural character (Burnard et al., 2006) which might be needed for editorial work for philologists and historians.

Our main purpose of transcription is to replicate the text content of the manuscripts to create a machine-readable text file for (crypt)analysis. In the case of ciphers, being it ciphertexts, keys, plaintexts or cleartexts, the most important task is to map the symbols in the ciphertext onto symbol representation as a written language. Transcription is rather straightforward if the symbol set of the cipher belongs to a known script, a writing system of a particular language. However, transcription is challenging when it comes to ciphers — while written language is an idealization, made up of a limited set of clearly distinct and discrete symbols (Piotrowski, 2012), ciphertexts are made up of symbols of a potentially unlimited number taken from various alphabets (e.g. Latin or Greek) and arbitrary symbol sets (e.g. Zodiac or alchemical signs).

The transcription conventions we apply need to be easy-to-use (Kline and Perdue, 2020) and to put into practice, albeit precise to be useful for decryption purposes. The transcription shall be i) computer-readable, ii) stored as plaintext files, iii) in a uniform encoding allowing to represent various scripts and symbols. All symbols that are part of the cipher shall be present in the transcription and represented so that all necessary information that might have impact on the interpretation and decryption of the manuscript is present. The transcription shall reflect the intention of the encoder and remain as faithful to the original manuscript as possible, which includes retaining the original line length, capitalization, punctuation or lack thereof, spelling and misspellings, additions, and marks.

In addition, information about the transcription shall be provided in terms of meta-

data containing information about the original image(s) of the encrypted manuscript (Desenclos, 2016), and the transcription process with possibility to leave comments. Metadata should follow the TEI guidelines (TEI Consortium, 2020) and as for the format, XML is recommended but the transcription process might become slow and time-consuming. We leave to the transcriber to decide upon his/her own metadata and in the following, we give only a minimal set to serve as suggestion, as an example. For our current purposes in the DECRYPT project, we store metadata about the encrypted source directly in the DECODE database, and we do not need a repeated set of metadata in the transcription files. Here, we store information about the type of the encrypted source (ciphertext, key, cleartext), the name of the folder and the image where the original is located, and the name or ID of the transcriber. We also store information about the transcription, the date when the transcription was created, and the approximate time it took to transcribe the image along with the transcription method so we can compare various methods. Examples are manual transcription by typing or dictating, or semi- or fully automatic methods using hand-written text recognition. The transcriber can also leave comments about difficulties and problems.

The transcription guidelines presented in this paper constitute a summary of a detailed set of guidelines for encrypted sources, presented in (Megyesi, 2020) with many illustrations and examples. The guidelines have been applied to the transcription of several hundred of encrypted manuscripts and stored in the DECODE database (Megyesi et al., 2019). The transcriptions we create serve for the decryption and analysis of ciphers, including ciphertexts, keys, and cipher-related cleartext documents. The guidelines are continuously developed as we stumble on new types of encrypted sources. In the following, we describe the typical problems and cases and describe how we deal with them.

### 3 Transcription of Ciphertext

Ciphertexts contain symbol sequences, letters from existing alphabets, digits, other graphic signs, or a mixture of these. Ciphertexts might

contain spaces, or the symbols follow each other one by one without any space or other marks between words, so called scriptura continua, used to hide word boundaries. Similar to historical text, punctuation marks are not frequent, sentence boundaries are typically not marked, and capitalized initial characters in the beginning of the sentence are usually missing, but they might appear. On the other hand, dots, commas or other marks might be used to indicate special codes or code groups. We can also find nulls in ciphertexts, i.e. symbols without any corresponding plaintext characters to confuse the cryptanalyst to make decryption even harder.

#### 3.1 Metadata

Each transcript file of a particular cipher (which may consist of multiple images) starts with metadata with information about the file. Each line is initiated by '#' followed by a transcription attribute and its value as illustrated in Figure 1.

```
#CIPHERTEXT
#CATALOG NAME: your own index, i.e. file location, e.g. Segr. di Stato Francia 3/1/
#IMAGE NAME: the name of the image(s) representing the cipher, e.g. 117r.jpg-117v.jpg
#TRANSCRIBER NAME: full name or initials of the transcriber, e.g. TimB
#DATE OF TRANSCRIPTION: the date the transcription was created, e.g. February 3, 2016
#TRANSCRIPTION TIME: the time it took to transcribe all images of a cipher in hours and
minutes without counting breaks and quality checks, e.g. 30+30+60 mins=120 minutes
#TRANSCRIPTION METHOD: speech recognition (Google Docs).
#COMMENTS: description of e.g. difficulties, problems
```

Figure 1: Metadata of the ciphertext.

#### 3.2 Content

Next, the content of the page is transcribed. Each new image in a cipher starts with a new comment line with information about the name of the image followed by a possible comment line, see Figure 2. Then, the actual content of the ciphertext is transcribed.

```
#IMAGE NAME: the name of the image, e.g. 234v.jpg
#COMMENTS: any comments, e.g. difficult to read line 3, bleed-through
```

Figure 2: Metadata of one page ciphertext.

The transcription is carried out symbol by symbol and row by row. This means that numbers are transcribed as numerals in ASCII, as typed in on the keyboard. The same applies to the letters in the Latin alphabet including capitalized letters, as well as punctuation marks. For other symbols, we use the Unicode name

representation where the name of the symbol is given following the Unicode standard.

Handwriting varies greatly not only between individuals but also for the same writer, which is why transcription of ciphertexts containing special symbols is especially challenging.

The transcription shall represent the original ciphertext shown in the image, keeping line breaks, spaces, punctuation marks, dots, underlined symbols, and cleartext words, phrases, sentences, paragraphs, as shown in the original image.

### 3.2.1 Line breaks, Spaces, Punctuation and Diacritical Marks

Line breaks are kept so that when a new line starts, a new line is added in the transcription.

Space ( ' ') is represented as <SPACE> if it is clear from the ciphertext that space might indicate word boundaries, i.e. appear on regular basis in every line in a systematic way, as illustrated in Figure 3.

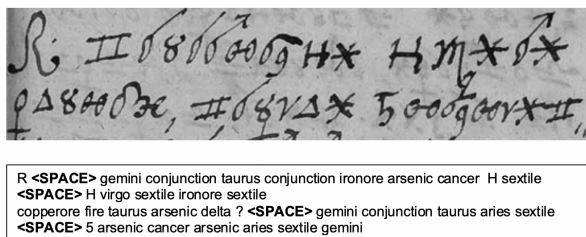


Figure 3: Transcription of a cipher with graphic signs represented as Unicode names and word boundaries marked as <SPACE> in the ciphertext.

If space occurs, but apparently not in a systematic way, just happen to be there, the space can be transcribed with two or more space characters written in ASCII ' ' in the transcription, as illustrated in Figure 4. The reason for allowing several space characters is that a larger space in the original might mark word boundaries which the encryptor unintentionally left there when encrypting the manuscript, which can be helpful in the decryption process as they might denote word boundaries.

Punctuation marks such as periods, commas, and question marks are transcribed as such. Sometimes, punctuation marks (e.g.

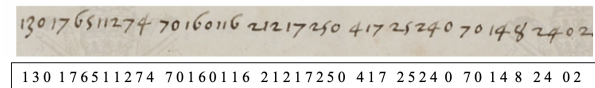


Figure 4: Transcription of a cipher with digits represented as ASCII characters and space marked as ' '.

dots, commas, accents, underscores) appear above or under specific symbols. It could be ink splash, but if they appear in a systematic way, they are transcribed as well. If the mark appears above the symbol, the sequence is transcribed as the symbol, followed by '^' and the specific mark (e.g. dot or comma). If the mark appears under the symbol, it is marked by an '\_' placed between the symbol and the mark ' ' (e.g. \_). Similarly, underlined symbols are marked with '\_' (double underscore) immediately following the symbol, except when the whole ciphertext is underlined. Sub- and/or superscripts shall be indicated on all individual symbols in a sequence of symbols.

Example of some special symbols and their transcription is given in Figure 5. To avoid ambiguous cases for symbols with sub- and/or superscript, we mark the sub- and the superscript in brackets in the form *SYMBOL*{*superscript*}{*subscript*}.

	Glyph	Transcribed as
Dot on top	3̇	3^.
Accent on top	3́	3^'
Dot on bottom	3̣	3_.
Dot on top and bottom	3̣̇	3{^}{_}

Figure 5: Transcription of symbols with diacritical marks.

### 3.2.2 Symbols

Symbols from other alphabets, such as Greek letters ( $\alpha$ ,  $\beta$ ) Roman numerals (I, II), or graphic signs, such as the alchemical or Zodiac signs are also common in ciphertexts. To transcribe those, we use their Unicode representation transcribed by its Unicode name

which then can be automatically converted to Unicode code to visualize the symbol in some font. Figure 6 illustrates the Zodiac signs, each with its Unicode name and code, followed by the glyph.

If the symbol cannot be covered by the symbols from some common alphabet (Latin and Greek) or digit (Arabic or Roman), the transcriber should look at the Zodiac signs first, followed by the alchemical signs as those symbols occur often in (European) encrypted manuscripts. If it is not possible to find any similar symbol among them, a symbol that reminds the most of the original can be searched for in the large Unicode table of symbols. What is important to keep in mind, that the symbol is transcribed with a unique name to make it distinguishable from the other symbol types in the cipher.

Name	Code	Glyph
aries	2648	♈
taurus	2649	♉
gemini	264A	♊
cancer	264B	♋
leo	264C	♌
virgo	264D	♍
libra	264E	♎
scorpio	264F	♏
sagittarius	2650	♐
capricorn	2651	♑
aquarius	2652	♒
pisces	2653	♓

Figure 6: Zodiac signs.

An example of the transcription of a ciphertext with alphabetical characters (Roman and Greek) and graphic signs consisting of Zodiac and alchemical signs is shown in Figure 7 along with the transcription indicated by the Unicode symbol name, its automatic conversion to Unicode codes, and lastly the final visualization of the transcription.

Uncertain symbols are transcribed with added question mark '?' immediately following the uncertain symbol. Possible interpretations of a symbol can be transcribed using the delimiter '/'. For example, if it is not clear if

a symbol represents a 0 or 6, it is transcribed as '0/6?'. It is highly desirable that all symbols are transcribed somehow, and no symbols are left out in the transcription for reliable decryption. The question mark ensures that all symbols have some representation in the transcription.

### 3.2.3 Catchwords

Historical manuscripts might contain catchwords placed at the foot of the page to mark page order (instead of digits), as illustrated in Figure 8. Catchwords are a sequence of symbols anticipated as the first symbol(s) of the following page. In ciphers, catchwords might denote an actual word, unintentionally, and transcribed as *<CATCHWORD Symbol\_Sequence>*, as exemplified in Figure 8.

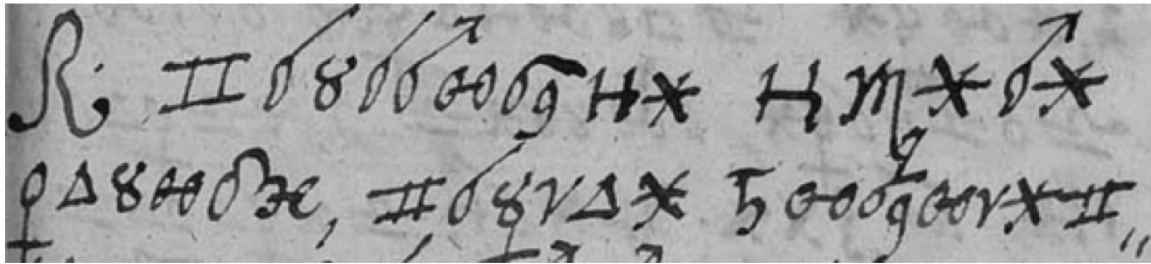
### 3.2.4 Notes in Margins

Sometimes ciphertexts are also included in the margins. This happens basically for two reasons: for corrections indicated in the ciphertext with a mark and the item is written in the margin, or the ciphertext continues in the margin to save space.

Transcription shall always reflect the intention of the encoder, i.e. the corrected segments as visualized in the original are transcribed. For example, if numbers are crossed-off in the original, these are not transcribed. If such cases occur, the transcriber leaves a comment about it in the comment line of the metadata. Similarly, insertions of corrections between symbols are transcribed, as they intended to appear. Ciphertext/cleartext written in the margin is added into the specific place as indicated by the given mark in the original. In Figure 9, the '+' written by the encoder intended to insert the cipher sequence written in the margin marked in red, and the transcription mirrors the intention of the encoder by directly adding the cipher sequence in the margin to the ciphertext.

Notes in the margin that are not corrections are transcribed after the transcription of the ciphertext, initially marked by a comment line with a short description that the upcoming sequence is a note in the left or right margin.





*Transcription by Unicode name:*

R gemini conjunction taurus conjunction ironore arsenic cancer H sextile <SPACE> H virgo sextile ironore sextile  
copperore fire taurus arsenic delta ? <SPACE> gemini conjunction taurus aries sextile SPACE 5  
arsenic cancer arsenic aries sextile gemini

*Reproduced by Unicode codes \*automatically\*:*

R <SPACE> 264A 260C 2649 260C 2642 29df 264b H 26b9 <SPACE> H 264d 26b9 2642 26b9  
2640 25b3 2649 29df 03b4 ? <SPACE> 264a 260C 2649 2648 25b3 26b9 SPACE 5 29df 264b  
29df 2648 26b9 264a

*Final representation for visualization:*

R <SPACE> ♊♈♉♊♋♌♍♎♏♐♑♒♓♔♕♖♗♘♙♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩♪♫♬♭♮♯♰♱♲♳♴♵♶♷♸♹♺♻♼♽♾♿♿ H \* <SPACE> H ♍♎♏♐♑♒♓♔♕♖♗♘♙♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩♪♫♬♭♮♯♰♱♲♳♴♵♶♷♸♹♺♻♼♽♾♿ \*  
♀♈♉♊♋♌♍♎♏♐♑♒♓♔♕♖♗♘♙♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩♪♫♬♭♮♯♰♱♲♳♴♵♶♷♸♹♺♻♼♽♾♿ <SPACE> ♊♋♌♍♎♏♐♑♒♓♔♕♖♗♘♙♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩♪♫♬♭♮♯♰♱♲♳♴♵♶♷♸♹♺♻♼♽♾♿ Δ \* <SPACE> 5 ♋♌♍♎♏♐♑♒♓♔♕♖♗♘♙♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩♪♫♬♭♮♯♰♱♲♳♴♵♶♷♸♹♺♻♼♽♾♿ \* ♊

Figure 7: Transcription of cipher with graphic signs and alphabetical characters, Zodiac signs marked in purple.

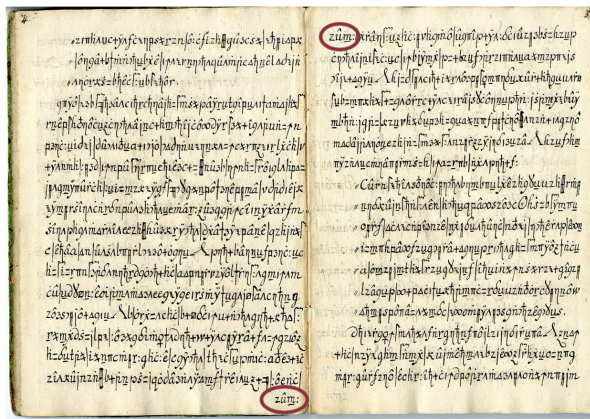


Figure 8: A cipher with catchword.

### 3.2.5 Ciphertext, Cleartext and Plaintext

The cipher sequences might be embedded in cleartext, i.e. non-encrypted text written in a natural language, or cleartext might be embedded in ciphertext. Cleartext embedded in ciphertext is illustrated in Figure 10 where the

Spanish word sequence 'comè la mi comanda' is embedded in the surrounding ciphertext.

To be able to distinguish between ciphertext and cleartext sequences, the latter is clearly marked in brackets as <CLEARTEXT LANG Letter/Word\_sequence> where the tag <CLEARTEXT... > denotes where the cleartext starts and ends as illustrated in the transcription in Figure 10. If the manuscript contains several lines of cleartext, each new line is represented by a new <CLEARTEXT... > tag. LANG represents the language the cleartext is written in, marked by a language ID as defined by ISO 639-1 two-letter codes<sup>3</sup> for languages (e.g. ES for Spanish, FR for French).

If there is some doubt about the cleartext/plaintext language, the language ID shall be defined as UN, indicating an unidentified language. For those cases where the cleartext does not necessarily constitute a certain language, such as dates (17.02.1725), years (1872)

<sup>3</sup>[https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)



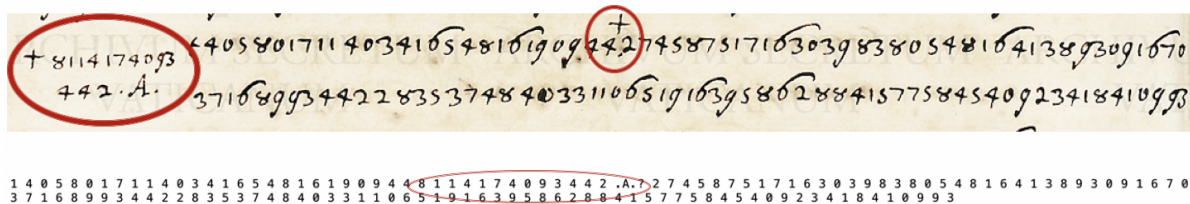


Figure 9: A ciphertext with corrections on the margin and its transcription.

130 176511274 70160116 21217250 41725240701482402101362701227  
220245845627670122721025024176 25 621224 0502484252617  
1301222 2 <CLARETEXT ES comè la mi coma^~da> 2225024701248474417 25242  
50727121601442464723847252 560244722202951224625212

Figure 10: Transcription of a cleartext embedded in ciphertext.

or paragraph markers (P.25), the language tag N/A (not applicable) is applied, as shown in <CLARETEXT N/A 1872>.

The cipher image might contain not only embedded (non-encrypted) cleartext, but also decrypted plaintext. We find decrypted plaintext written over the ciphertext sequences by the receiver, as illustrated in Figure 11. Similar to cleartext, plaintext is transcribed as <PLAINTEXT LANG Letter/Word\_sequence> in a separate line.

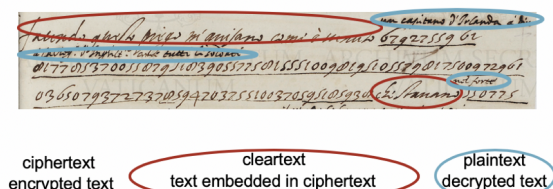


Figure 11: Cleartext and plaintext embedded in ciphertext.

### 3.2.6 Abbreviations

Sometimes we find abbreviations in the plaintext or cleartext sequences. Original text shall be transcribed as such, and in cases where abbreviations occur, the expansion of the abbreviated segment given as <ABBR expanded-abbreviation>. For example, *sre* in 'Del sre Bianco' is the abbreviation of *signore* and transcribed as in 'Del sre <ABBR signore> Bianco'.

## 4 Transcription of Keys

A key defines how each entity in the original plaintext shall be encrypted. Keys might contain substitution of not only characters in the plaintext alphabet, but also space to hide word boundaries, or nomenclatures where bigrams, trigrams, syllables, morphemes, common words, and/or named entities, typically referring to persons, geographic areas, or dates, are substituted with certain symbol(s). Punctuation marks or capital letters might occur in keys. A key might also contain nulls, i.e. symbols without any corresponding plaintext characters to confuse the cryptanalyst to make decryption even harder, explained in cleartext, or given as cipher symbol (Megyesi et al., 2019). Codes might also be present without any plaintext, serving as placeholders (Tudor et al., 2020).

The codes in a key might be of variable length. Each type of entity to be encrypted can be encoded by one symbol only, two symbols, three symbols, and so on. For example, the plaintext alphabet characters might be encrypted with codes using two-digit numbers, the nomenclatures with three-digit numbers, space with one-digit numbers, and the nulls with two-digit numbers, etc. Figure 12 illustrates a key based on homophonic substitution with nomenclature from the second half of the 17th century. Each sign in the alphabet is represented by at least one ciphertext symbol (e.g. A->18, m; B->20; C->19). The vowels and double consonants are assigned an additional ciphertext sign. The key also con-

tains encoded syllables with two-digit numbers or bigram characters (e.g. ba->65; be->66), followed by a nomenclature in the form of a list of Spanish words encoded with three-digit numbers or symbols (e.g. apustamiento->106). Keys might also include cleartext with explanation to (some parts of) the key. Similar to ciphertexts, metadata of the key is defined first, followed by the transcription and possible cleartext appearing in the original key.

#### 4.1 Metadata

Before the actual transcription, original keys are described by a set of metadata, related to the transcription and the description of the key, each initialized by a hashtag (#) as defined in Figure 13.

#### 4.2 Codes

After the metadata, the actual transcription of the content of the keys follows. The transcription guidelines for keys are partly based on the master thesis of Tudor (2019) and the transcription guidelines for ciphers (Megyesi, 2020). For keys, the same principles apply as for ciphertexts, when it comes to symbols described in Section 3.2.2, and cleartext sequences, which often contains explanations about the cipher key and explained in Section 3.2.5.

Since keys can be structured in many different ways, often as tables with or without explanations in cleartext, the graphical structure of the keys cannot be represented in any simple way in the transcription. Here, we make an interpretation of the content of the coding scheme instead. We list the key items as `<CODE-PLAINTEXT>` pairs where each unique pair is written in a line, first the code followed by the separator `'-'`, then the plaintext unit, being it a character in the alphabet, syllable, word, null, or punctuation mark. Nulls are transcribed as `<NULL>` and missing plaintext of a code is transcribed as `<EMPTY>` (Tudor et al., 2020).

To illustrate the key representation, as shown in the key in Figure 12, the first three letters *A*, *B*, and *C* with their first code, are represented in the transcription as follows:

```
18 - A
20 - B
19 - C
```

A plaintext unit can be coded by several ciphertext symbols, such as in homophonic ciphers. In those cases, the possible codes are transcribed sequentially separated by a bar `'|'` followed by `'-'` and the plaintext unit. For example, in our example in Figure 12, *A* can be coded not one but two possible ways, with the number *18* and the letter *m*. The alternative codes are transcribed in one line even when these are written in two lines in the original key, as illustrated below:

```
18 | m - A
20 - B
19 - C
```

Similarly, in case of polyphonic cipher keys where a ciphertext symbol can be mapped to several plaintext units, each plaintext symbol is listed with the code, separated by a bar `'|'` in one line, no matter if they appear on separate lines in the original. For example, if the code *0* in the key might encode two plaintext letters, e.g. *a* and *t*, we would transcribe it as:

```
0 - a|t.
```

Please note that the separator bar `'|'` aimed for separating code or plaintext alternatives in keys is written in ASCII. However, if the ciphertext symbol represents the glyph `'|'` in the code itself, it is transcribed with its Unicode name `'verticalline'`.

### 5 Transcription of Cleartext

Cleartexts are defined as non-encrypted plaintexts. These could be letters without any ciphertext that appear in the context of a cipher, e.g. in a letter correspondence, or it could appear embedded in ciphertext, as described in Section 3.2.5.

#### 5.1 Metadata

The metadata for cleartext documents contains the information shown in Figure 14.

#### 5.2 Cleartext Content

Next, the content of the image is transcribed. Each new image starts with a new comment line with information about the name of the image followed by a possible comment line, similar to Figure 2.

The transcription shall represent the original text shown in the image, keeping line

Figure 12: A key from the second half of the 17th century.

Figure 13: Metadata of a key.

Figure 14: Metadata of a cleartext document.

Figure 14: Metadata of a cleartext document.

breaks, spaces, punctuation marks, dots, underlined symbols, and cleartext words, phrases, sentences, and paragraphs, as shown in the original image. More specifically:

- Line breaks are kept so that when a new line starts, a new line is added in the trans-

scription.

- Space is represented as space. Punctuation marks, such as periods, commas, and question marks are transcribed as such.
- Uncertain words or characters are transcribed with added question mark '?' immediately following the uncertain sequence. Possible interpretations of a symbol can be transcribed using the delimiter '/'. For example, if it is not clear if the word should be transcribed as *and* or *und*, all interpretations shall be transcribed with a question mark, as in 'and/und?'.

- Unidentified letters or words shall be marked with an asterisk (\*).
- Abbreviations. Original text shall be transcribed as such, and in cases where abbreviations occur, the expansion of the abbreviation can be inserted after the abbreviated word given as <ABBR expanded-abbreviation>.

## 6 Conclusion

We presented guidelines for a systematic and consistent transcription of historical encrypted sources: ciphertexts, keys, and cleartexts. Consistent transcription across ciphers provides the possibility to study and compare historical sources systematically in large scale. The guidelines might be also a useful resource in case we employ several transcribers of the same document for more accurate transcription. Our hope is that the guidelines will serve in getting a more accurate, unambiguous and consistent transcription within and across ciphertexts and keys, a first step taken to a standardized transcription of historical encrypted sources. Lastly, and most importantly, consistent transcription across symbols sets and scripts can also support (semi-)automatic transcription allowing sophisticated hand-written text recognition models — with or without human intervention — to take care of the tedious transcription process of historical manuscripts.

## Acknowledgements

I would like to thank my colleagues in the DECRYPT project, in particular Michelle Waldispühl, George Lasry, and Nils Kopal for their valuable feedback on the guidelines. Crina Tudor deserves special thanks for her work on the transcription of historical cipher keys as part of her master thesis. Lastly, transcription of hundreds of ciphers would not have been possible without all students who take the time and effort to transcribe these fascinating historical sources. This work was supported by the Swedish Research Council, grant 2018-06074.

## References

- Nada Aldarrab. 2017. Decipherment of historical manuscripts. Master’s thesis, University of Southern California. Master thesis in Computer Science.
- Lou Burnard, Katherine O’Brien O’Keefe, and John Unsworth. 2006. *Electronic Textual Editing*. The Modern Language Association of America, New York.
- Adele Cipolla. 2018. *Digital Philology: New Thoughts on Old Questions*. Libreria Universitaria Edizioni, Padova, Italy.
- Camille Desenclos. 2016. Early Modern Correspondence: A New Challenge for Digital Editions. In *Digital Scholarly Editing: Theories and Practices*, UK. Open Book Publishers.
- Matthew James Driscoll and Elena Pierazzo. 2016. *Digital Scholarly Editing: Theories and Practices*. Open Book Publishers, UK.
- Mary-Jo Kline and Susan Holbrook Perdue. 2020. *A Guide to Documentary Editing*. The Association for Documentary Editing, Online edition, 3 edition.
- Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. The Copiale Cipher. In *Invited talk at ACL Workshop on Building and Using Comparable Corpora (BUCC)*. Association for Computational Linguistics.
- Almut Koester. 2010. Building Small Specialized Corpora. In *The Routledge Handbook of Corpus Linguistics*, pages 66–79.
- Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Ciphers and Keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt19*, Mons, Belgium, June.
- Beáta Megyesi. 2020. Transcription of Historical Ciphers and Keys: Guidelines. <https://cl.lingfil.uu.se/~bea/publ/transcription-guidelines200221.pdf>. Version: February 10, 2020.
- David L. Vander Meulen and G. Thomas Tanselle. 1999. A System of Manuscript Transcription. *Studies in Bibliography*, 52:201–212.
- Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Morgan Claypool Publishers.
- Robert Rosenberg. 2006. Documentary Editing. In *Electronic Textual Editing*, pages 92–104, New York. The Modern Language Association of America.
- Richard Sproat. 2006. *A Computational Theory of Writing Systems*. Studies in Natural Language Processing. Cambridge University Press.
- TEI P5 TEI Consortium. 2020. Guidelines for Electronic Text Encoding and Interchange (version 4.0.0). Accessed: 2020-04-27.
- Crina Tudor, Beáta Megyesi, and Benedek Láng. 2020. Automatic Key Structure Extraction. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt20*, Budapest, Hungary, June.
- Crina Tudor. 2019. Studies of Cipher Keys from the 16th Century: Transcription, Systematisation and Analysis. Master thesis in Language Technology, Uppsala University, Sweden.



# A Hungarian Cryptological Manual in Berlin

Štefan Porubský

Institute of Computer Science of the Czech Academy of Sciences

Pod Vodárenskou věží 271/2, 182 07 Praha 8

Czech Republic

sporubsky@hotmail.com

## Abstract

This is a report on some activities of the Hungarian SIGINT department and a Hungarian cryptographic manual written by the head of its Department X István Petrikovits as found in the Archive of the German Federal Foreign Office in Berlin.

## 1 Introduction

Archive file TICOM Box No. 3843 in the Archive of the German Federal Foreign Office (Politisches Archiv des Auswärtigen Amtes) in Berlin contains a cryptographic typewritten manual entitled *Rejtjel – Segédlet* (A Cipher Aid) written by the head of the Hungarian military cryptological center István Petrikovits. The (slightly damaged) characterization of the file by a TICOM officer says: “*??y general notes on code and cypher, and ??tography, in Hungarian, undated. Includes letter frequencies counts to depth of 100,000 letters of following languages: Hungarian, German, Roumanien, Russian, Serbian, Croatian, Slovak, Czech. From the Hungarian Crypt. Unit, Eggenfelden.*”<sup>1</sup> The document is not dated, but from the given author’s military rank “General Major” we can deduce that it was written after May 20, 1943, the date when Petrikovits was “exceptionally and of mercy” awarded this honorary rank (vezérőrnagy in Hungarian). There are no details at disposal on the prehistory of the manuscript or about the way how it got to the TICOM Archive. One possible indication is the fact that between 2 May 1945 and 28 July 1946, Petrikovits was a prisoner of war, detained by the USA (Szakály, 2016).<sup>2</sup>

<sup>1</sup>Notice that according to the last pre-Trianon 1910 census there lived numerous ethnic minorities within the borders of the “Hungary-proper”, i.e. excluding Croatia-Slavonia: 16.1% Romanians, 10.5% Slovaks, 10.4% Germans, 2.5% Ruthenians, 2.5% Serbs and 8% others.

<sup>2</sup>See also (Jakus, 2013) where however the author names him as Viktor Petrikovits instead.

## 2 Stephanus Petrikovics vs. István von Petrikovits

István Petrikovits was born on September 24, 1888 in the town of Hlohovec (Galgóc or Galgócz in Hungarian or Freistadt an der Waag in German or in its Slovak colloquial variant Frašták, at that time) which today lies in Slovakia. In that time it also was a predominantly Slovak town. In the church register of the local Roman Catholic church written in Latin we can read that he was baptized on September 30 as Stephanus Robertus Matheus Petrikovics. Here Petrikovics is a more usual Hungarian transcription of the Slavic surname Petrikovič. His god-father was certain Robertus Petrikovics an engineer (geometra) from Párdány, a village today lying in Serbia. It is predominately a Serbian village nowadays, but in that time it had originally two parts: Serb Pardanj and Slovak Pardanj. In the middle of the 18th century, Germans and Hungarians settled here, mainly in Slovak Pardanj and so its name changes in accordance with the structure of the population.<sup>3</sup>



Figure 1. Borders of Slovakia 1939-1945.

Stephanus’ father registered as Mathias Ignatius Franciscus Petrikovics<sup>4</sup> (born April 4, 1852) came

<sup>3</sup>Two villages (the former Serb Pardanj and the former Slovak/German/Hungarian Pardanj) united into a single village in 1907 today called Meda. According to the 1910 census there lived 3,213 inhabitants in both settlements with the following ethnicity: German - 1,874, Serbian - 1,052, Hungarian - 243.

<sup>4</sup>His given names are written in this form in the local



from the village of Bory (Bori in Hungarian) in Slovakia which was predominantly Hungarian with a strong Slovak minority. His mother Natalia Maria Stephana Juliana Biróczy (born October 8, 1866) came from the village of Dedinka (Fajkürt in Hungarian) which at that time was a small predominantly Hungarian village with a small Slovak minority.

Mathias' father Eduardus had six children and his surname in their local church registers is written in three different ways: once Petrikovich, four times Petrikovics and once Petrikovits. When Stephanus (István in Hungarian) decided to write his surname employing the older Hungarian orthographic possibility<sup>5</sup> with *-ts* instead of *-cs* at the end of his name to stress his noble descent<sup>6</sup> is not known to the author.<sup>7</sup> On the list of the officers of the 15th Honved Infantry Regiment (Honved-Infanterieregiment Nr. 15 / Trencsényi 15. honvéd gyalogezered)<sup>8</sup> which was intended for the front in Galicia against Russia and existed till the end of WWI, we can find this form of his surname on the list of 31 regiment Captains. Surprisingly,<sup>9</sup> his direct superior István Ujszászy<sup>10</sup> also writes his surname in the form Petrikovics or even as Petnikovics (Ujszászy, 2007). Petrikovits died on April 16, 1947 in Budapest several months after his return from the PoW camp. He is buried in Farkasréti cemetery in Budapest.

### 3 Hungarian Military Sigint

After the collapse of the Austria-Hungary Monarchy the new Hungarian military structure rose up

church register contrary to the surname which is not written in the form Petrikovič. For his photo cf. (Sziklay and Borovzsky, 1898, p. 441).

<sup>5</sup>Older Hungarian texts are heterogenous due to the absence of the generally accepted spelling norms. For the phonem [tS] non existing in Latin (in the south Slavic denoted as *ć* or in the west Slavic as *č*) there were used the digraphs *ts*, *cs* or *ch*.

<sup>6</sup>To stress this fact he also used to write István von Petrikovits.

<sup>7</sup>In the second half of 19th century and later many inhabitants of Hungarian part of Austria-Hungary Magyarized their names.

<sup>8</sup>The nationality structure of the regiment was 85% Slovaks and 15% other nationalities and its recruitment was district of Trenčín (Trencsén in Hungarian and Trentschin in German) in north-west Slovakia. District of Trenčín was predominantly Slovak.

<sup>9</sup>Laxity or merely an indication of a not close service interrelationship between both of them?

<sup>10</sup>István Ujszászy served as the head of the Hungarian General Staff's counter-espionage department VKF-2 from 1939 to 1942.

on the ruins of the old Empire one. The structure of the later corresponded to that of the political framework of the country. The Empire army had three branches: the joint one, called the Imperial and Royal and recruited from the whole Empire, and then two branches recruited from each part separately, the Imperial-Royal Landwehr for the Austrian part and the Royal Hungarian Landwehr (Honvéd) for the Hungarian one. From our point of view, the directorate of military intelligence – the *k.u.k. Evidenzbureau* headquartered in Vienna, was a whole Empire unit. Thus the independent Hungarian national military intelligence and counter-intelligence services have been built out of its own and based mainly on the Hungarian staff from various military intelligence units of the Monarchy army. The basic structure of the Hungarian new unit undergone numerous structural changes since its establishment in 1918. At the beginning, after the Aster Revolution already on November 1, 1918 to build up such a unit was entrusted Dimitrije (Demeter) Stojaković (or Sztojakovics)<sup>11</sup> who in period 1917-1918 was the head of the Balkan section of the Evidenzbureau in Baden near Vienna. The primary aim of the newly established unit was the intelligence service against the antagonistic neighbor states Czechoslovakia,<sup>12</sup> Romania and Serbia<sup>13</sup>.

In 1919 the first Minister of Defense (MoD), the Hungarian Social Democrat Vilmos Böhm founded an intelligence department headed by Sztojakovics at MoD. Under the short-lived<sup>14</sup> communist Hungarian Soviet Republic (Hungarian: Magyarországi Tanácsköztársaság or Magyarországi Szocialista Szövetséges Tanácsköztársaság) the department changed only slightly and served as a subordinate unit called

<sup>11</sup>Born 1883 into a Serb family. He Magyarized his name to Sztójay Döme on November 4, 1935. In his birthplace Versec (Serbian: Vršac, German: Werschetz) a half of inhabitants were Germans and one third the Serbians in 1910. Sztójay, an avid supporter of the National Socialists, served as a military attaché in Berlin from 1925 to 1933. From 1933 to 1935 he served in the Ministry of Defence and from 1935 to 1944 as the Hungarian ambassador to Germany. Between March and August 1944 he was appointed the Prime Minister and Minister of Foreign Affairs of a pro-German government. After the war he was found guilty of war crimes and crimes against the Hungarian people, sentenced to death, and executed in 1946.

<sup>12</sup>Czecho-Slovakia, later Czechoslovakia, split into Protectorate of Bohemia and Moravia and the Slovak Republic in March 1939, a division which lasted till the end of WWII.

<sup>13</sup>More precisely, The Kingdom of the Serbs, Croats, and Slovenes from 1918 to 1929, and after October 3, 1929 The Kingdom of Yugoslavia.

<sup>14</sup>March 21, 1919 till August 1, 1919



Figure 2. Austria-Hungary and its neighbours borders history ((a) state borders of Austria-Hungary around 1900; (b) states borders around 2000). (<https://i.pinimg.com/originals/d2/be/e3/d2bee3efc7fac13e655bd305788d3c4d.jpg>)



Figure 3. Hungary borders history 1900-1945. (<https://www.globalsecurity.org/military/world/europe/images/hu-map-1921-2.jpg>)

Department II (or VK II group) of the General Staff (Hungarian: VK II. csoportja, VK for Vezér Kar).

After the fall of the Hungarian Soviet republic, it was The Treaty of Trianon, the peace agreement of 1920, which regulated the status of the new independent Hungarian state. Conditions of the Treaty copied often those imposed on Germany by the Treaty of Versailles. The army was to be restricted, there was to be no conscription, heavy artillery, tanks and air force were prohibited, etc.<sup>15</sup> Since the army high command was also prohibited, the General Staff was established under the cover of the MoD on 1 July 1921 as the Main Directorate VI of the Ministry of Defence. Within this Main Directorate VI the 2nd Department was charged with intelligence and counter-intelligence. Its official name was VI-2 Department of the Ministry of Defence (VI/2. osztály). This Bureau of the Second Division operated mainly on the rules taken over from the time of the Evidenzbureau and essentially functioned in this form until 2nd March 1938, when the General Staff and the 2nd Department was officially established and Gyula Gömbös, the former head of Department VI was appointed as the main commander of the Hungarian royal army. From that point Department VI-2 was called “VKF-2” (General Staff 2nd Department, Hungarian: vezérkari főnökség 2. osztálya). The internal organizational structure of VKF – with unsubstantial modifications and extensions – principally remained in the form as it was designed by Colonel Döme Sztójay (Hajma, 2013):

- Register subdivision (Nyilvántartó alosztály, “Nyil”): military, political and defense data processing
- Central offensive subdivision (Központi offenzív alosztály “Koffa”): intelligence assessment, organization and control
- Defensive subdivision (Defenzív alosztály “Def”): anti-spy and cooperation with military police
- Directly subordinated groups (Közvetlenek); “X” Department, etc.

The X Department was charged with SIGINT and both cryptography and cryptanalysis. The “central figure” of the X Department was General Hermann Pokorny. Pokorny was born on April 7,

<sup>15</sup>For instance, due to the strategic importance of the railway, no railway would be built with more than one track!!

1882 into a German family in Kroměříž, (German: Kremsier), a Moravian town in a historical region in the east of the Czech Republic.<sup>16</sup> At that time the town was bilingual with 13% of German speaking minority to which belonged also the Pokorny’s family. Actually, the word ‘pokorný’ in Czech or Slovak means ‘the humble one’.



Figure 4. H. Pokorny in the radio interception station on the East front in 1915 (Pokorny, 2000).

Major Pokorny<sup>17</sup> was one of the best “language expert” in the Evidenzbureau, who spoke German, French, Russian, Polish, Serbian, Czech, Slovak, Bulgarian etc. Immediately after the begin of WWI, as a member of the Austria-Hungary SIGINT group on the East front, he proved his brilliant cryptologic abilities by cracking Russian ciphers<sup>18</sup> during the Battle of Tannenberg, the Siege of Premysl<sup>19</sup> or at the seizure of Brest-Litovsk. He was the head of Russian subsection of the Austro-

<sup>16</sup>Kroměříž is one of the most beautiful cities in Moravia region called the “Athens of Moravia”. In 1885, Emperor Franz Joseph and Tsar Alexander III met in Kroměříž to political talks.

<sup>17</sup>He joined the k.u.k. Austro-Hungarian Army in August 1900 as the cadet and by 1918, when the Austro-Hungarian Empire collapsed, his rank was Lieutenant Colonel. He was promoted to Colonel in 1925 and retired in 1935 in the rank of Major General (since 1928) and in October 1945 he was promoted to General.

<sup>18</sup>In the first 20 months of the war he solved several thousands!! intercepted Russian radiograms. For instance, he recognized that the Russians used a system in which they reduced the 35-letter Russian alphabet to 24 letters, while replacing the 11 missing letters with some of the used 24 ones. The results of his activity of tapping and decrypting Russian radio telegrams he described in his book (Pokorny, 2000). His effort to publish it in Germany in 1939 did not find a support there. In January 1945, however, the Russian General Staff took over a copy of this book, with all 18 of original Russian keys decryptions and approx. 12,000 deciphered radiograms. The book was translated into Russian and used later as a secret aid to their staff.

<sup>19</sup>Today a town in southeastern Poland, in that time in Galizia in Cisleithanian Austria-Hungary; German: Premissel, Czech: Přemysl, Ukrainian: Peremyshl (Перемисьль).

Hungarian Deschiffrierdienst. Thought being a German-language native speaker, Pokorny did not request neither Czechoslovak citizenship because of his German origin nor the Austrian one because he had been born in the Czech part of the Monarchy and feared that as a person born outside Austria, he would be considered a second-class citizen. Therefore he decided for the Hungarian citizenship after the WWI and moved to Budapest.

In 1919 Pokorny was charged to set up the Hungarian cryptological bureau on the bases in Vienna operating the so-called *S*-group. The new group was named, as mentioned above, the *X*-Department. Why *X* in its name, is not known. After Pokorny build up this cryptologic section in 1920 he acted as its head until the end of April 1925.<sup>20</sup> He was replaced by his deputy, Colonel Vilmos Kabina<sup>21</sup>. Kabina, as Pokorny, also served during WWI as a cipher officer. Kabina retired on March 1, 1927, but held the position as the head of the *X*-group until January 31, 1935 when he definitely left military service.<sup>22</sup> The next day the head of *X*-group became Colonel István Petrikovits who lead the unit until May 2, 1945, despite his retirement on 1 November 1942.

The main sections of the *X*-department were (Ritter, 2010):

- two radiocommunication intelligence battalions
- deployed interception stations
- central decryption section

The central decryption section was divided into subsections, each having 4-5 cryptologists, and covering specified regions or state groups. Their number changed according to the political situation and military importance. Around 1944 the sections had the following "territorial competences":

**Turkish section:** Turkey,

**English section:** British Commonwealth, Egypt, USA,

<sup>20</sup>In 1935 already mentioned Gömbös, now premier minister of Hungary, forced Pokorny to leave his active military service arguing with his non-Hungarian origin.

<sup>21</sup>Born as József Vilmos János Zsigmond Kabina on May 4, 1876 in the town of Levice (Hungarian: Léva, German: Lewenz; the Old Slavic name of the town was *Leva*, which means "the Left One", since the town lies on the left bank of the Hron river), now in Slovakia. In that time it was predominantly a Hungarian town. On June 17, 1951 the communist regime forced him and his wife, both barely able to walk, to leave their apartment in Budapest and move to a small town Kunszentmárton in the central Hungary.

<sup>22</sup>According to some source, e.g. (Ritter, 2010), H. Pokorny temporarily headed the *X*-Department for a half year period during 1935/36.

**French section:** Vichy France and its colonies, Swiss, Belgian, Holland and Greece emigrant governments,

**Russian section:** USSR, Bulgaria, Czechoslovak and Yugoslavian emigrant governments, Independent Croatia,

**Romanian section:** Romania,

**Swedish section:** Sweden, Danish and Norwegian emigrant governments,

**Italian section:** Italy and Vatican,

**Spain section:** Spain and Portugal,

**Japan section:** Japan and China.

István Petrikovits' language "expertise" were Slovak and Bulgarian. He participated on the work of the Russian section. Further members of the 'Russian half' of the section were lieutenant Pál Krisztinkovics and major Elemér Lajtos. The substantive part of work of the Russian section was oriented to follow radio communication and to gather intelligence information from and in the direction of the Soviet Union and Yugoslavia. The section was successful in cracking some ciphers used by the Soviets. They cracked at least one important cipher used by the Soviet Foreign Ministry.

In 1940 Petrikovits reciprocated the visit of representatives of the Finnish SIGINT group in Hungary and reported about the Finnish achievements in the deciphering of Soviet military ciphers.<sup>23</sup> Thus for instance, the Finns were able to gather the airdropped Soviet military material thanks to the information intercepted and deciphered from the Soviet radiograms. As a result of a close collaboration not only between both SIGINT groups, several Hungarian officers were awarded Finnish orders. One of them was I. Ujszászy and also I. Petrikovits who was awarded the Finnish Order of the Cross of Liberty with swords of the 2nd Class (Sallay, 2014). This order was founded 1918 upon the initiative of General C.G.E. Mannerheim.

Another interesting collaboration was that with Japan. In July 1938 the Japan military attache moved his headquarter from Vienna to Budapest and an intensive military collaboration between Hungarian and Japan on the field of intelligence and decipherment started (Sallay, 2007). One aspect of this collaboration was an intensive ex-

<sup>23</sup>The good relations between Hungarians and Finns goes back to the Finno-Ugric linguistic affinity cultivated since the end of the 19th century. Hungarian volunteers fought on the side of Finland during the Winter War (1939–1940) against the Soviet Union. Even Albert Szent-Györgyi offered all of his Nobel prize money which he received in 1937 to Finland in 1940.

change of distinctions. The Order of the Rising Sun awarded in nine classes was established in 1875 as Japan's first order. The third through sixth classes were conferred upon individuals who have made significant contributions to Japan. István Ujszászy was awarded this order twice: in 1940 it was its 4th Class and in 1942 the 3rd Class. In the 1942 "wave of honours exchange" István Petrikovits was awarded the Order of the Sacred Treasure of the 3rd Class, an imperial order established in 1888.

In November 1944, before the advancing Red Army, the X-department escaped to Und (German Undten: Croatian Unda), a mostly Croatian municipality in the Sopron-Fertőd region in western Hungary close to the border with Austria. Gradually moving to the west, the group gave up on May 2, 1945 to the Americans next to Eggenfelden, a small town in the Lower Bavaria. All transported and historically important material became a part of the TICOM archive.

The activities of VKF or of the army general-ity in general were not always completely consistent with the visible official pro-German politics of Hungary (cf. e.g. (Szakály, 1987)). On one side, Wilhelm Höttl (1915-1999), the young Austrian Nazi Party member serving in SD-Ausland (Sicherheitsdienst = Security Servis), and by 1944 acting as a head of the R.S.H.A.<sup>24</sup> branch for Central and South East Europe conveyed an impressive tribute to the work of the Hungarian secret intelligence during WWII (cf. (Kahn, 1996, p. 453)). According to Höttl, Hitler, who ensnared Admirál Miklós Horthy<sup>25</sup> into Axis alignment by restoring some of Hungary lost territories, deeply distrusted Horthy. The augury of a bleak outcome of the war forced Horthy's cabinet<sup>26</sup>

<sup>24</sup>Reichssicherheitshauptamt = Reich Central Security Office

<sup>25</sup>Miklós Horthy de Nagybánya or German Nikolaus Horthy Ritter von Nagybánya (1868 - 1957) was a Hungarian admiral and statesman, who served as the Regent of the Kingdom of Hungary from 1 March 1920 to 15 October 1944.

<sup>26</sup>Already 1943 the Prime Minister Miklós Kállay (1942-1944) sent envoys to Istanbul. In 1944 one of them was the Nobel Prize Winner Albert Szent-Györgyi. Till the end 1944 also the Defence Minister L.Csatay, Chief of Staff General F.Szombathelyi and VKF's Department 2 all sent their representatives to Istanbul.

One of the key figures behind the scene in Istanbul was in Russia born and during WWI volunteer of the Russia army, Colonel Harold Gibson, SIS station head in Turkey. Gibson as a "visa clerk" of the British embassy in Prague played also a crucial role in the reorientation swap of the Czechoslovak military secret service from the French to the British secret service and in the organisation of a spectacular flight of

to secret negotiations with Western Allies where an important role was played by Major General István Ujszászy.<sup>27</sup> Contact with Western Allies led to the *Mission Sparrow* when OSS airdropped a three men group under Colonel Florimond Duke in Hungary. Three days later, on March 19, the Germans in *Operation Margarethe* invaded Hungary and captured all three members.<sup>28</sup> When on August 23 a cup replaced pro-Nazi Romanian government by a Soviet-aligned one, Horthy plotted with Ujszászy and the commandant of Budapest to seize Budapest and to start secret negotiations with Moscow. But Germans were again ahead mainly due to intelligence activities of Höttl's SD-Ausland which penetrated Hungarian Secret Service.<sup>29</sup>

Finally, it would be perhaps interesting to the reader to note that a cousin of Hermann Pokorny, Major Franciszek Pokorny born June 15, 1891 in the village of Mosty<sup>30</sup> was a Polish Army offi-

Colonel Moravec, chief of the Czechoslovak secret service with 10 of his close collaborators, from Prague to London on the eve before the German invasion of Czechoslovakia on March 15, 1939, cf. (Porubský, 2017a; Porubský, 2017b). After his retirement 1958 Gibson was found shot dead under unexplained circumstances in his flat in Rome in 1960. Possible collaboration with the Soviet secret service as a reason for a suicide is not excluded.

<sup>27</sup>István Ujszászy was the head of the internal security apparatus subordinate to the Interior Minister known as the State Protection Center (Hungarian: Államvédelmi Központ) from 1942 to 1944 and he was one of the key figures in the preparation for the so-called "bail out" (Hungarian: kiugrás). After the German occupation of Hungary, he was arrested by the SS Security Service SD. Then since February 1945 by the NKVD. After interrogations in Moscow he was allegedly transferred back to Hungary to a detention camp of the infamous Hungarian secret police State Protection Authority (Hungarian: Államvédelmi Hatóság or ÁVH) in the summer of 1948. His final fate disappears in the fog.

Ujszászy's handwritten protocols written for the ÁVH kept in the archives of the Ministry of the Interior for decades are published in (Ujszászy, 2007).

In the interwar period 1930-1938 Ujszászy's served also as the military attache in Paris, Warsaw and Prague. For the comments on his stay in Prague see (Moravec, 1975). However his name is misspelled as Ujzazy here.

<sup>28</sup>Hungary's German occupation was justified via argument of the "unresolved Jewish question" and the "unfaithfulness" of the Hungarian political leadership. On the other hand, recent archive discoveries indicate (Peterecz, 2012) that the Allies played a two-faced game and that the aim of the air-drop was also to provoke the Germans to sent military forces to Hungary and thus to weaken the German military position in the West before the invasion of Normandy disregarding possible Jewish casualties in the Hungarian population.

<sup>29</sup>Höttl was recruited by the United States Army Counter Intelligence Corps (CIC) after the war.

<sup>30</sup>Mosty u Jablunkova (Polish: Mosty koło Jabłonkowa, German: Mosty bei Jablunkau or Mosty in den Beskiden), lies today in the Moravian-Silesian Region of the Czech



cer who, after World War I, from 1925 till 1929 headed the Polish General Staff's Cipher Bureau (Referat Radio i Szyfrów Oddziału II Sztabu Generalnego (Głównego)) the predecessor of the famous Biuro Szyfrów.

## 4 The Manual

The cryptological manual authored by István Petrikovits has 91 on one side typewritten pages with the following contents:<sup>31</sup>

The aim of this cipher aid .....	1
Significance of cryptography .....	2
History of the cryptography .....	3
General rules of cryptography .....	4
Methods of cryptography .....	5
Basic terms .....	6
Cryptographic systems .....	7
Language structure .....	8
Analytics of the Hungarian language .....	9
Analytics of the German language .....	14
Analytics of the Romanian language .....	19
Analytics of the Russian language .....	23
Analytics of the Serbian language .....	26
Analytics of the Croatian language .....	29
Analytics of the Slovak language .....	32
Analytics of the Czech language .....	36
Details of cipher systems .....	40
Simple substitution systems .....	41
Keys possibilities for substitutions .....	44
Composite substitutions .....	48
Cipher tables .....	50
Decipherment of composite substitutions .....	54
Keyword reconstruction .....	58
Reconstruction of the key tableaux .....	61
The autokey cipher .....	62
Encryption with one letter password .....	65
Transpositions .....	68
Simple transpositions .....	68
Encryption with miscellaneous tables .....	73
Grilles .....	77
Composite transpositions .....	80
Double transpositions .....	81
On cipher codes .....	82
Hints for cipher texts exploration .....	86
Epilogue .....	87
Appendices. Solutions of problems .....	88

Republic. At that time, during the Austria-Hungary Empire, with a predominant majority of population being native Polish-speakers.

<sup>31</sup>This is contents given by Petrikovits at the end of manual. Actual headings are partly different.

Contents .....	91
----------------	----

In the introductory part<sup>32</sup> of the manual Petrikovics settles the basic terminology. He recognizes three main branches of cryptography:

- real (apparent) cryptography, mostly based on mathematical ideas
- covered (hidden) cryptography, for instance to use passphrases to initialize previously prearranged actions
- invisible writing using chemical processes (invisible ink, etc.)

What concerns (in his conception called *real*) cryptological systems he distinguishes between permutations and substitutions.<sup>33</sup>

Almost one half of the manual is devoted to a thorough description of the frequency analysis of the basic structural elements of written documents, as the frequencies of the letters, bigrams or words of the aforesaid languages. The selection of languages indicates that the manual arose from the needs of Department X since apart of Hungarian and German, they are languages used in the enemy states. On the other side, the briefness of the description of cryptological techniques suggests that it was not intended to be used as a textbook, rather as a succinct introductory guide, maybe for personal use or as a basis for a future project. Throughout the text scattered problems, with solution given at the end, indicate that the manual was not written as a report during the captivity.

This analysis of the written form of languages constitutes the 2nd Chapter called *Language analytics* (Nyelv-analytika). Petrikovits gives relatively detailed 'anatomy' of the Hungarian, German, Romanian, Russian, Serbian, Croatian, Slovak and Czech language. The corresponding reports follow the same structure for each of these languages. The given characteristics are based on the analysis of sample texts having approximately 100,000 characters in total (Russian as an exemption uses only 50,000 characters). Certain speciality is that these 100,000 characters stem from a collection of (not closely identified) independent sub-texts each having approximately from 3,000

<sup>32</sup>Though not denoted as Chapter 1 it is so meant, as follows from the rest of the manual.

<sup>33</sup>For some concepts he also gives their German translation, thus substitutions are in Hungarian *helyettesítő* or in German *Ersatzverfahren* and permutations are *keverőrendszerek* or *Versatzverfahren*, respectively.

to 6,000 characters. The reason for considering such sub-divided collections of texts is a bit unusual. They are base for several tables of the letter frequencies. Besides the standard tables of letter frequencies based on the whole collections of 100,000 characters, interesting min-max tables of frequencies are given. These tables show the minimal and maximal letter frequencies in these sub-texts. For instance, for the computation of the characteristics of the Hungarian they have about 3,000 letters each. As an example, the letter with the maximal frequency in the Hungarian is *e*. It appeared in the whole sample 10,656 times, i.e. with frequency 10.66%, while in sub-texts it appeared with frequency lying between 8.26% and 14.70%. The six-columns Table a occupying one page and giving total absolute and relative, minimal or maximal frequencies is followed by an analogical Table b showing analogical recalculated frequencies of letters of the telegraphic alphabets (that is without diacritic accents). Petrikovits explains the significant differences of the frequencies in comparison with the previous global Table a arguing that some parts were written in dialects, or that some of them are written by uneducated persons, etc. For instance, letter *e* appears in the Hungarian alphabet as *e* or *é*. The general frequency of *e* is 14.14% while in min-max table the given frequencies are 10.80% and 19.90%. The next part contains comments on distribution of vowels and consonants. Actually there are no numerical characteristics here, only comments on their pattern alternations. The following section contains notes to the mixed patterns of vowels and consonants. The last part comments the words frequencies in the sample of 100,000 characters. Lists of the absolute frequencies of the most frequent one-, two-, three- and four-letters words are also given.

Then frequency characterizations of characters and words of German, Romanian, Russian (based on sample of  $6 \times 8,333$  characters texts), Serbian ( $16 \times 6,250$  characters), Croatian ( $16 \times 6,250$  characters), Slovak ( $16 \times 5,900 + 16 \times 5,600$  characters) and Czech ( $17 \times 5,900$  characters) are given following the same pattern.

The third chapter of the manual is devoted to a very short description of the basic cryptographic techniques. It starts with the monoalphabetic cipher. The idea how to solve a monoalphabetic cipher is briefly demonstrated using an atbash like cipher. He points out its weakness when the char-

acters are substituted by simple letters, or by couples of digits stressing the fact that we have 26 characters but only 10 digits when replacing letters by pairs of digits. To defuse this defect he shows two substitutions employing couples of digits or letters with more or less equidistributed components. He also mentions the usage of nulls.

In the part devoted to the polyalphabetic substitutions (called composite (in Hungarian *bonyolult*) substitutions in the contents) Petrikovits works with a periodic Vigenère's cipher. He shows Trithem's and variants of Vigenère's tables with numeric or alphabetic heads<sup>34</sup> and show how to solve this type of a cipher. The solution is based on Kasiski's test without to mentioning Kasiski's name. To apply it he counts distances between repeating bigramms and trigramms. After finding the key pattern he shows how to recover the used keyword (and indirectly also the message language) and cipher alphabet taking into account also the frequency tables of different languages.

The next section is devoted to the autokey cipher. Petrikovits handles its text-autokey type where he uses either the message text or its enciphering to determine the next element in the keystream. He shows how to solve the Vigenère autoclav of the above first type based on the tabula recta when the key is a single letter.

The part dealing with transpositions starts with a short critics of Cardinal Richelieu's simple transposition cipher, and continues with the columnar transposition (with nulls) and the standardly given hints for its solution. Then Petrikovits presents the initiatory, and rarely given, ideas how to solve a 90° turning, in his case a  $6 \times 6$ , grid cipher. Provided we know that a turning grill was used, the solution idea is based on the observation of symmetries of bigramms in the 1st and 3rd or 2nd and 4th turns.

This part of the manual ends with short comments on what he calls composite transpositions. These are either transpositions of previously by a substitution encrypted texts or double columnar transpositions. An example of the second type is given with a comment that a double columnar transposition should be consider to be unsolvable from the cryptanalytical point of view.

Then follows a section devoted to general description of the use of nomenclators in cipher

<sup>34</sup>He names them Vigenère's, Gronfeld's (not Gronsfield?) ciphers.

texts. Petrikovits describes several possibilities for the form of the inserted codewords. For instance, their numerical or literal form, their most used length or possibilities to use tables or dictionaries. The description is a bit lengthy and too general, and was incorporated probably due to their general use in the intelligence and diplomatic correspondence.

The final ‘scholarly’ section is devoted to some general instructions for examining cryptographic materials.

The concluding part containing the solutions of the problems given in the text lists their solutions without any comment.

## 5 Appendix

In the TICOM collection found by the author in the Archive of the German Federal Foreign Office there was another file registered as TICOM report No. 3870. Its characterisation says: *Bried notes in Hungarian on types of Bulgarian, Czech and Yugoslavian keys used 1921-35. From the Hungarian Crypt. Unit, Eggenfelden.*

The file contains two reports both covering period February 1, 1921 through August 1, 1936. Surprisingly, though they contain principally identical information about the cryptological activities, they are not identical.

Both reports are typewritten and each is 1 and half side long. They are classified as *strictly confidential* and are written in Hungarian. The (of this paper author’s) translation of the substance of their contents is as follows:

### Report

on the cipher keys of foreign countries which were deciphered by lieutenant-colonel István Petrikovits in the period February 1, 1921 – August 1, 1936.

#### Czechia<sup>35</sup>

- it was effective 1921/11/1 through 1922/7/30: small diplomatic cipher key.
- it was effective 1922/8/1 through 1923/8/1: big diplomatic cipher key.
- it was effective 1929/8/1 through 1934/10/1: cipher key of an army division (katonai csapat)

<sup>35</sup> Meant is Czechoslovakia. It was a custom in Hungary in the interwar time to use the name Czechia or Czech Republic (Csehszág) instead of Czechoslovakia and all its citizens to call simply as Czechs. For some aspects of the relations between Czechoslovakia and Hungary (including some Ujszászy’s activities) cf. (Miklós, 2017).

- it was effective 1930/8/1 through 1931/7/31: cipher key of an army division
- it was effective 1931/8/1 through 1932/7/31: cipher key of an army division
- it was effective 1932/8/15 through 1933/10/1: cipher key of an army division
- it was effective 1933/8/15 through 1934/10/1: cipher key of an army division

*Remark in the Report:* Every army cipher key changed on daily basis 5-5 within the cipher system and thus within every cipher system deciphering of 75-75 new recipherings were realized.

#### Yugoslavia

- it was effective 1923/6/1 through 1926/1/1: big diplomatic cipher key.
- it was effective 1926/1/1 through 1927/6/1: big diplomatic cipher key.
- it was effective 1927/6/1 through 1929/12/31: big diplomatic cipher key.
- it was effective 1930/1/1 through 1932/12/31: big diplomatic cipher key.
- it was effective 1933/1/1 through 1935/4/1: big diplomatic cipher key.
- since 1927/8/1 diplomatic cipher keys changed cipher tables every month. Altogether more than 100 reciphering tables.
- it was effective 1926/1/1 through 1934/12/31: consular cipher key.
- it was effective 1930/1/1 till today valid royal court cipher key

#### Bulgaria

- it was effective 1926/8/1 through 1933/3/1: diplomatic cipher key 975
- it was effective 1930/8/1 through 1933/1/1: diplomatic cipher key 03210
- since 1935/1/1 till today: diplomatic cipher key 00062
- since 1935/1/1 till today: diplomatic cipher key 67676
- it was effective 1930/1/1 through 1935/1/1: royal court cipher key.

*Remark in the Report:* Here listed Czech, Yugoslav and Bulgarian cipher keys were deciphered by lieutenant-colonel István Petrikovits who deciphered thousands of telegrams.

Date: Budapest August 10, 1936 and signed by Pokorný (followed by an unreadable sign part)

As mentioned the second report is not a copy of the first one. Its head reads: Report on the cipher keys of foreign countries on which decipherment there cooperated lieutenant-colonel István Petrikovits in the period February 1, 1921 – August 1, 1936. Further, the first Yugoslavian item report has the following footnote: deciphering a code requires 6-12 months. Otherwise the contents (but not the form of the lists) are identical. Finally, the closing remark says: Here listed Czech, Yugoslavian and Bulgarian cipher keys also deciphered in a co-operation of lieutenant-colonel István Petrikovits who deciphered thousands of telegrams. This second report is again signed by Pokorny but in contrast to the first one on February 10, 1937.

## Acknowledgments

The author thanks the unknown referees for their valuable comments. He was supported by the strategic development financing RVO:67985807.

## References

- Deborah S. Cornelius. 2011. *Hungary in World War II: Caught in the Cauldron*. Series: World War II: The Global, Human, and Ethical Dimension Fordham University Press, New York Project MUSE muse.jhu.edu/book/14532.
- Lajos Hajma. 2001. *A katonai felderítés és hírszerzés története. (History of military reconnaissance and intelligence)*. Zrínyi Miklós Nemzetvédelmi Egyetem, Felderítő Tanszék (Miklós Zrínyi National Defense University).
- János Jakus. 2013. A magyar rádió- és rádióelektronikai felderítés szervezeti változásai 1990-ig. (Organizational changes of the Hungarian radio- and radioelectronics reconnaissance until 1990). *Rendvédelem-történeti Füzetek (Acta historiae praesidii ordinis)*, 23 (27-30): 101–128.
- David Kahn. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, New York.
- Dániel Miklós. 2017. The Picture of the Czechs through the Eyes of Hungarian Politicians. In *Az első világháború irodalmi és történelmi aspektusai a kelet-európai régióban*. Tanulmánykötet, Trefort-Kert Alapítvány - ELTE Doktorandusz Önkormányzat, Budapest, pages 47–62.
- František Moravec. 1975. *Master of spies: the memoirs of General František Moravec*. Doubleday, Garden City, N.Y.
- Zoltan Peterecz. 2012. Sparrow Mission: A US Intelligence Failure during World War II. *Intelligence and National Security*, 27(2):241–260.
- Hermann Pokorny. 2000. *Emlékeim: a láthatatlan hírszerző. (My memories. The invisible intelligence officer)*. Petit Real, Budapest.
- Štefan Porubský. 2017. STP cipher of the Czechoslovak in-exile Ministry of defence in London during WWII. In *Proceedings of EuroHCC 2017 (3rd European Historical Cipher Colloquim, Smolenice, Slovakia, May 18-19,2017*, pages 47–66.
- Štefan Porubský. 2017. Application and misapplication of the Czechoslovak STP cipher during WWII (report on an unpublished manuscript). *Tatra Mt. Math. Publ.* 70:41–91.
- László Ritter. 2010. A Magyar rádiófelderítés a második világháborúban. (The Hungarian radio reconnaissance in the second world war. *Felderítő Szemle (Intelligence Review)*, IX (2):149–168.
- Gergely Pál Sallay. 2007. Japán-magyar katonadiplomáciai kapcsolatok 1938-1944. (Japanese-Hungarian military diplomatic relations 1938-1944). *Hadtörténelmi Közlemények (Military History Bulletin)* 120:183–202.
- Gergely Pál Sallay. 2014. Finn kitüntetések a Magyar Nemzeti Múzeum gyűjteményében. (Finnish honors in the collection of the Hungarian National Museum). *Folia Historica* 30:155–176.
- Sándor Szakály. 1987. Wojskowy ruch oporu na Węgrzech v latach drugiej wojny światowej (Polish) [The Hungarian military resistance movement in World War II]. In: *Razem w walce*, Węgierski instytut kultury & Wojskowy instytut historyczny, Warszawa, pages 65–91.
- Sándor Szakály. 2016. From the Evidenzbureau to the Establishment of the Independent Hungarian Military Intelligence. *National Security Review (Military National Security Service, Budapest)*, 2:13–28.
- János Sziklay and Samu Borovszky. 1898. Magyarország vármegyéi és városai. (Counties and cities of Hungary), volume 4: Nyitra vármegye. Apollo irodalmi társaság, Budapest.
- István Ujszászy. 2007. *Vallomások a holtak házából Ujszászy István vezérőrnagynak a 2. vkf. osztály és az Államvédelmi Központ vezetőjének az ÁVH fogságában írott feljegyzései [Testimonies from the House of the Dead (Protocols written by Major-General István Ujszászy the head of the VKF 2nd Department and of the AVK during his captivity by ÁVH)]*. G.Haraszti and Z.A.Kovács and Sz.Szita eds. Corvina, Budapest.

# The Zschweigert Cryptograph – A Remarkable Early Encryption Machine

**Klaus Schmeh**  
Private Scholar  
www.schmeh.org  
klaus@schmeh.org

## Abstract

The *Zschweigert Cryptograph* is one of the many cipher machine designs developed in the years following the First World War (1914-1918). It was invented by textile engineer Rudolf Zschweigert, who had designed programmable stitching machines before and apparently transferred his computing expertise to cryptology. Unlike the Enigma and as good as all other crypto devices of the time, the *Zschweigert Cryptograph* implements a transposition cipher, not a substitution cipher. To the author's knowledge, it was the first encryption machine that worked with keys provided on punched cards. The goal of this paper is to introduce the *Zschweigert Cryptograph* and its history, to provide a mathematical specification of its encryption algorithm, and to explore how it can be cryptanalyzed. It will be shown that the *Zschweigert Cryptograph*, which was probably never used in practice, was insecure even by the standards of the 1920s and not convenient enough to compete with other encryption machines of the time.

## 1 Introduction

It is a well-known fact that the failure of almost all important (manual) encryption systems used in the First World War led to the invention of numerous encryption machines in the years after. Among the best-known crypto devices of this era are the Enigma, the Hebern rotor designs, the Kryha encryption machines and Arvid Damm's cipher devices – just to name a few.

A lesser known encryption machine from the post-WW1 years is the *Maschine zum Herstellen*

*chiffrierter Schriftstücke* (“Machine for producing enciphered documents”) by German engineer Rudolf Zschweigert. We will refer to this machine as *Zschweigert Cryptograph*.

To the author's knowledge, the *Zschweigert Cryptograph* was never built (perhaps with the exception of prototypes that are now lost), let alone used in practice. The only known source describing this machine is a patent filed by Rudolf Zschweigert in 1919 and granted one year later (Zschweigert, 1920).

Though it was never used in practice, the *Zschweigert Cryptograph* is noteworthy for several reasons:

- Contrary to virtually all other mechanical and electric cipher machine designs, the *Zschweigert Cryptograph* implements a transposition cipher (not a substitution cipher). This property is the reason why this machine is mentioned in (LANAKI, 1996) and (Nichols, 1998). However, both sources give no description of the *Zschweigert Cryptograph*. As far as the author knows, nothing detailed has ever been published about this device, except the patent. The *Zschweigert Cryptograph* should not be confused with the transposition cipher tool (it's not really a machine) invented by Luigi Nicoletti in 1918, which is mentioned in (Kahn, 1996).
- The *Zschweigert Cryptograph* was invented by a textile entrepreneur. As is well known, the textile industry adapted computing hardware long before encryption technology did. As will be shown, the *Zschweigert Cryptograph* represents a design that transferred computing expertise from the textile industry to cryptology.
- The *Zschweigert Cryptograph* is the earliest cipher machine the author is aware of that ap-



plies a punched card as key.

## 2 Rudolf Zschweigert

Rudolf Zschweigert (1873-1947) was a German engineer, who lived in the cities of Chemnitz, Plauen, and Hof, Germany. In the 1930s, he was a member of the city council of Hof. He was married to Gertrud (1891-1982). Zschweigert is best remembered for having built up a major mineral and meteorite collection, which is today preserved in the *Museum Reich der Kristalle* in Munich, Germany (Wilson, 2019).

Rudolf Zschweigert's professional dedication was that of a textile manufacturer and factory owner. The *Weberei Zschweigert* ("Weaving Mill Zschweigert") existed from 1921 to the 1960s. Between 1909 and 1934, Zschweigert was granted at least 15 patents in Germany, Austria, Switzerland and the USA. 14 of these patents concerned textile technology, especially looms and stitching machines. Zschweigert's only patent not related to textiles is the one relating to the encryption machine discussed in this paper.

Rudolf Zschweigert was not the only cipher machine inventor with a background in the textile industry. A second and much more prominent person of this kind was Swedish engineer Arvid Damm (1869-1927), who cooperated with his country man Boris Hagelin in the 1920s and laid the foundation of what was to become Crypto AG, a company that still exists today (Hagelin, 1994).

## 3 Specification of the Encryption Algorithm

In the following, we provide a formal specification of the encryption algorithm implemented by the *Zschweigert Cryptograph*. It is based on the informal description in the patent.

The *Zschweigert Cryptograph* uses a  $9 \times n$  binary matrix  $K$  as key, with  $n$  being a positive integer. Every row of  $K$  has a Hamming weight of one, which means that there is exactly one one per row, while the eight other values are set to zero. Here's an example (with  $n = 5$ ) we denote as  $K_{exmpl}$ :

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

In the following, we will denote the position of the one in row  $i$  as  $k_i$ . In other words:

$$k_i = j : \Leftrightarrow K_{i,j} = 1$$

The key space of the *Zschweigert Cryptograph* is, of course, dependent on  $n$ , the number of rows of the matrix. As there are nine possibilities for each row, the number of keys is  $9^n$ . This means that with a 40-rows matrix, exhaustive key search is about as laborious as with a 128-bit key.

The alphabet used by the *Zschweigert Cryptograph* is not specified in the patent. Instead, it is assumed that every character provided by the typewriter in use can be encrypted. To keep things simple, we assume that only upper-case letters from A to Z are encrypted, which makes an alphabet of 26 characters. It seems likely that such an alphabet would also have been used in practice.

We denote the plaintext as  $P = p_i$  with  $i = 0, 1, \dots, l-1$  and  $l$  being the number of letters in the plaintext. As an example, we take  $P_{example} := "HISTOCRYPTTWENTY"$ , which means that  $p_0 = "H", p_1 = "I", p_2 = "S", \dots, p_{15} = "Y"$  and  $l = 16$ .

The ciphertext is represented by another matrix,  $C$ .  $C$  has nine columns. The elements of  $C$  are from the set  $\{A, \dots, Z, -\}$  with  $-$  representing a null character. At the beginning, all elements of  $C$  are set to  $-$ . When we write  $C$ , we omit all lines containing only the null character.

### 3.1 Encryption

To define the encryption algorithm, we need the following function:

```
Write-to-Matrix  ( $C, column \in \{1 \dots 9\}, p \in \{A, \dots, Z\}$ )
 $i = 0$ 
while  $C_{i,column} \neq "-" : i = i + 1$ 
   $C_{i,column} := p$ 
return  $C$ 
```

The encryption algorithm is specified as follows:

### **Encrypt** ( $P, K$ )

$n :=$  number of rows of  $K$

For  $i = 0$  to  $l - 1$ :

$C := \text{Write-to-Matrix}(C, k_{i \bmod n}, p_i)$   
return  $C$

This means that the first letter of the plaintext takes the column of the one in the first line of the key matrix. The second character takes the column of the one in the second line and so on. Each letter is written into the highest line of the plaintext matrix that is still empty.

With  $P_{\text{exmpl}}$  and  $K_{\text{exmpl}}$ , we get the following ciphertext (denoted as  $C_{\text{exmpl}}$ , see also figure 1):

$$\begin{pmatrix} T & - & - & H & - & S & I & - & - \\ P & - & - & C & - & O & R & - & - \\ N & - & - & T & - & Y & W & - & - \\ - & - & - & Y & - & T & - & - & - \\ - & - & - & - & - & E & - & - & - \\ - & - & - & - & - & T & - & - & - \end{pmatrix}$$

Noting the ciphertext this way is unpractical if it is, for instance, sent by telegram. The patent therefore suggests the use of separators, but details are not given. A possible way to write down the ciphertext is: TPN - - HCTY - SOYTET IRW - -.

## **3.2 Decryption**

To define the decryption algorithm, we need the following function:

**Read-from-Matrix** ( $C, \text{column} \in \{1 \dots 9\}$ )

$i = 0$

while  $C_{i, \text{column}} = "-" : i = i + 1$

$p := C_{i, \text{column}}$

$C_{i, \text{column}} := "-"$

return  $p$

The decryption algorithm now can be specified as follows:

### **Decrypt** ( $C, K$ )

$n :=$  number of rows of  $K$

For  $i = 0$  to  $l - 1$ :

$p_i := \text{Read-from-Matrix}(C, k_{i \bmod n}, p_i)$   
return  $P$

## **4 Construction of the Machine**

While the patent provides only short coverage of the encryption method (not to mention a theoretical foundation), the construction of the machine is described in great detail. This is probably because Rudolf Zschweigert was familiar with mechanical engineering, but not with cryptography.

As can be seen in figure 2, the *Zschweigert Cryptograph* is based on a mechanical typewriter. Instead of printing on a piece of paper, this typewriter prints on nine separate paper rolls. The roll used for a certain letter is controlled by a unit that works with a punched card. This punched card corresponds with the matrix introduced in the previous chapter.

The punched card has nine columns and an arbitrary number of rows. In each row, there is exactly one hole. The mechanics of the machine always move the type used to the paper roll that corresponds with the column of the current punched card row and types a letter.

After a letter has been typed, the respective roll turns up by one unit and the next row of the punched card is read. When the end of the punched card is reached, the control unit starts with the first row again.

At the end, the user takes the nine paper rolls and reads the letter sequences on them. According to the patent, this can be done in a key-dependent order. However, from a cryptographic point of view, changing the order of the rolls doesn't make much sense, as this is equivalent with changing the order of the columns on the punched card, which can be done while the card is produced (unless, the card is reused and a different order of the rolls is applied each time – a case we don't cover in this paper).

If the encrypted message is transmitted by radio, the sender can read the ciphertext directly from the nine paper rolls and transmit them. If sent by letter, it is necessary to copy the ciphertext from the rolls (unless, of course, one doesn't mind sending nine paper strips by mail).

Decrypting works very similar as encrypting. Of course, an identical punched (key) card is necessary. No stylus is needed. The operator presses the space key repeatedly. The control unit will always move the paper roll to the center, where the next plaintext letter can be read. The receiver needs to copy each letter and thus receives the plaintext.

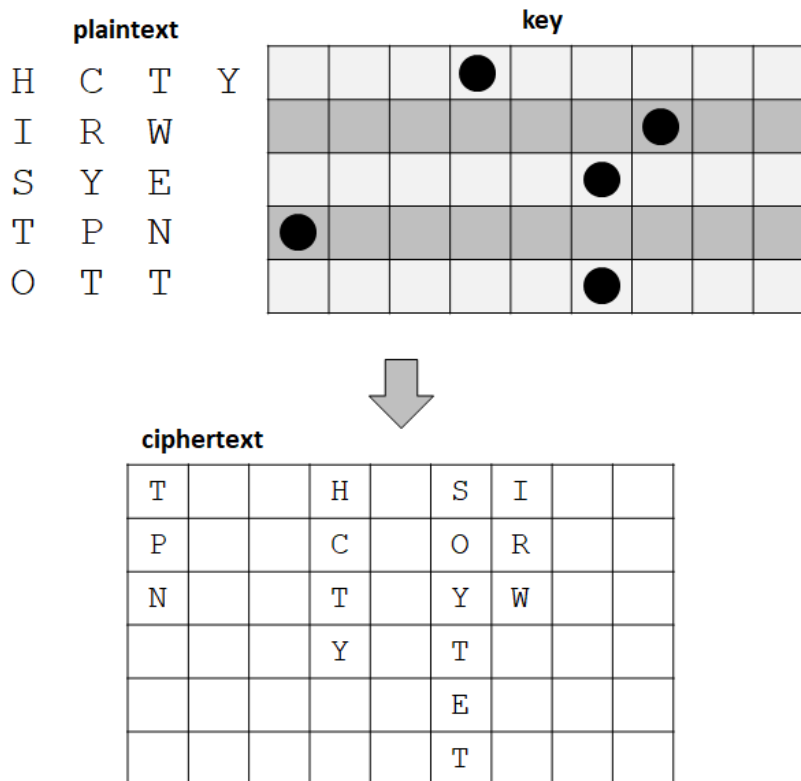


Figure 1: Using a matrix (represented by a punched card) as key, the plaintext HISTOCRYPT TWENTY is encrypted to a ciphertext that can be written as: TPN - - HCTY - SOYTET IRW - -.

It should be clear that encrypting a message with the *Zschweigert Cryptograph* is not especially convenient. The sender needs to copy the output in order to bring it to a format that can be sent by telegram or teletype. The receiver needs to copy every decrypted letter from the machine. This means that although the *Zschweigert Cryptograph* includes a typewriter, manual writing is necessary.

## 5 Historical Background

As is well-known, the textile industry played an important role in the history of information technology. In 1804, Joseph Marie Jacquard introduced the Jacquard machine, a loom controlled by punched cards (Jacquard, 2019). The Jacquard machine (figure 5) is generally regarded as the first programmable hardware in history. The concept of programming a machine with a punched card became widely accepted in the 20th century, first in Hollerith machines, later in computers.

It is an interesting question whether the crypto machine designs of aforementioned Swedish engineer Arvid Damm were influenced by computing technology he encountered in the textile industry.

To our knowledge, this question has never been researched.

In the case of Rudolf Zschweigert, we have found a source that might link the computing technology of the textile industry with cryptology. In 1908, Zschweigert was awarded two patents for a stitching machine that is controlled by a punched card. The one patent concerns the machine itself (Zschweigert, 1908a), the other one a device for punching the holes into the card (Zschweigert, 1908b).

It seems likely that this stitching machine laid the foundation for the *Zschweigert Cryptograph* that was invented a decade later. While the punched card in the stitching machine controlled the production of a pattern on a piece of cloth, the punched card in the cipher machine controlled an encryption process on a typewriter.

To the author's knowledge, the *Zschweigert Cryptograph* is the earliest cipher machine that used punched-card keys. Many others were to follow, including the HC-9 (Reuvers, 2019), the Fialka (Reuvers, 2019), the KW-7 (Reuvers, 2019), and the T-310 (Schmeh, 2006).

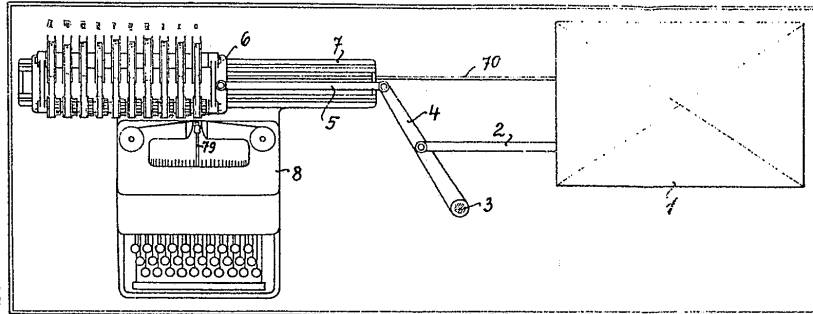


Figure 2: The *Zschweigert Cryptograph* is based on a mechanical typewriter. It uses nine movable co-axial paper roles (left) that are controled by a unit (right), the details of which are not depicted in this diagram. *Source: Patent*

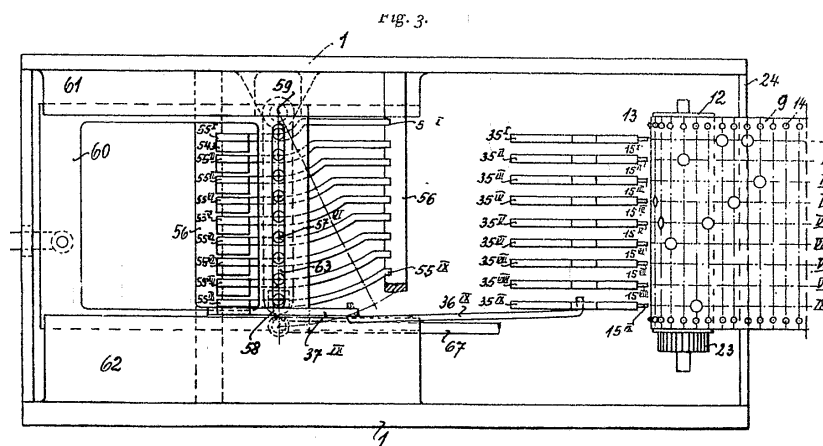


Figure 3: The key of the *Zschweigert Cryptograph* is provided on a punched card with nine columns (right). *Source: Patent*



Figure 4: The Jacquard machine is a 19th century loom controlled by a punched card. It is considered the first programmable device in history. Rudolf Zschweigert, a textile engineer, might have been influenced by the Jacquard machine when he designed his punched-card controlled cryptograph. *Source: Wikimedia Commons / 29263a,b / Dmm2va7*



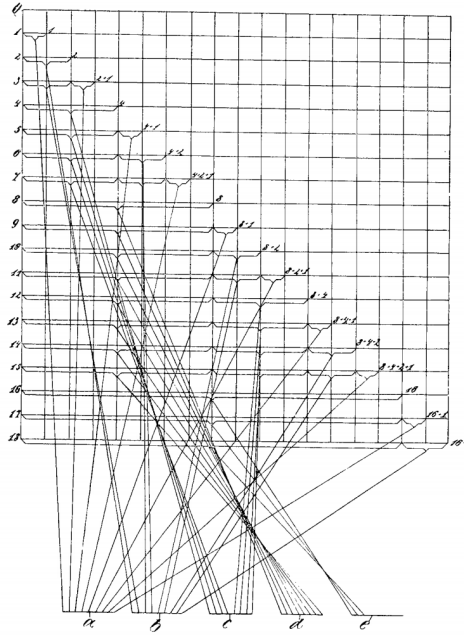


Figure 5: Rudolf Zschweigert invented a stitching machine that is controlled by a punched card. It seems likely that this device machine laid the foundation for the *Zschweigert Cryptograph*. Source: Patent

## 6 Cryptanalysis Considerations

When it comes to cryptanalyzing the *Zschweigert Cryptograph*, two steps need to be distinguished. In the first one, the codebreaker tries to find out how many rows the key matrix has; in the second step, the position of the ones in the matrix is determined. When the matrix is completely reconstructed, the ciphertext can be easily decrypted.

### 6.1 Determining the Number of Matrix Rows

The most obvious method for determining the number of rows in the key matrix is brute force. If we look at the example ciphertext  $C_{\text{exmpl}}$ , we see that it consists of 16 letters. With a computer program it is not very difficult to check every matrix length between, say, 4 and 16. We need to apply the second step (locating the ones in the matrix) on each of these candidates.

While brute force (with a computer program) is certainly an appropriate approach today, the cryptanalysts of the 1920s needed an attack that could be carried out manually. In fact, such a method is available. If we look at our example ciphertext  $C_{\text{exmpl}} = \text{TPN} - - \text{HCTY} - \text{SOYTET IRW} - -$ , we see that the number of letters in the nine columns is 3, 0, 0, 4, 0, 6, 3, 0, and 0. With the exception of 4, each of these numbers is divisible by three. When 4 is divided by 3, the remainder is 1. Taking into account that we are dealing with a 16-letter mes-

sage, this can best be explained with a five-row matrix, the first row of which is used four times, while rows 2-4 are used three times each. This means that the key matrix has five rows.

Of course, it is also possible that the number of rows is 16, which would mean that the matrix is as long as the plaintext. However, following Occam's Razor, which states that the simplest explanation should be taken first, a cryptanalyst will usually start with examining the five-rows hypothesis.

Things might not always be this easy, especially when the plaintext is longer than in our example and the matrix has more rows. However, we assume that guessing the number of rows in the key matrix will usually be possible. To find out more, further research is necessary.

### 6.2 Locating the Ones in the Matrix

We now assume that the number of matrix rows is known or that a guess has been made (for instance, in the course of a brute-force attack). In the next step, we need to determine the location of the ones. The task of the cryptanalyst becomes easier if the number of rows is considerably smaller than the message length, i.e., if each row is used to encrypt several letters. The case where the number of matrix rows exceeds the plaintext length is not relevant, as we can always ignore the rows not used.

In the example shown in figure 1 five matrix rows encrypt a plaintext consisting of 16 letters.

One weakness of the *Zschweigert Cryptograph* is obvious: Just by looking at the ciphertext we can easily derive the number of ones of each column (i.e., the Hamming weight). If we look at the example ciphertext  $C_{\text{exmpl}} = \text{TPN} - - \text{HCTY} - \text{SOYTET IRW} - -$ , we immediately see that the second, the third, the fifth, the eighth, and the ninth column of the matrix must be empty, because there are no letters in the corresponding positions of the ciphertext.

Considering that there are three letters in both the first and in the seventh column of the ciphertext, we can conclude that each of the corresponding matrix columns contains exactly one one. The six letters in the sixth ciphertext column lead to the conclusion that the sixth matrix column contains two ones. The four letters in the fourth ciphertext column are especially helpful, as they not only tell us that there is one one in the fourth matrix column but also that this one is located in the top matrix row.

We have now reconstructed the first matrix row, and we know that columns 2, 3, 5, 8, and 9 are empty. This leaves us with  $4! = 24$  possibilities for the positions of the ones in rows 2 to 5. We can even reduce this number to its half because we know that there are two equal rows, which are interchangeable. So, in the end, there are only 12 combinations to try. With a computer program, this can easily be achieved by brute force.

If no computer is available, as it was the case when the *Zschweigert Cryptograph* was invented, the technique of multiple anagramming, as described by Helen Fouché Gaines in her book *Elementary Cryptanalysis*, can be used (Fouché Gaines, 1939). The details are not within the scope of this paper.

Things become a little more complicated, of course, if we use a key matrix with more rows. This is especially the case if the matrix is as long as the plaintext. Multiple anagramming still seems possible, even if it is much more laborious than in the simple example we provided. We assume that the computer-based technique of hill climbing (Schmeh, 2017), which has proven extremely powerful in the breaking of historical ciphers, is the best means to attack a cryptogram of this kind and we believe that this approach would work well against the *Zschweigert Cryptograph*. Again, the details are out of scope in this paper.

Overall, we can conclude that breaking a mes-

sage encrypted with the *Zschweigert Cryptograph* is feasible, even with the means of a 1920 cryptanalyst. The machine can be made more secure by using matrices with more columns and by forbidding the use of matrices that are shorter than the plaintext. Nevertheless, the author's impression is that the concept of the *Zschweigert Cryptograph* is not suitable for a reasonably secure encryption machine. Future research might go into more detail about this question.

## 7 Future Work

As far as the author of this work knows, this paper is the first publication about the *Zschweigert Cryptograph*, except the patent. It is therefore obvious that additional research work is necessary in order to understand this machine and its background. Especially, the following items should be researched:

- The biography of Rudolf Zschweigert appears to be not especially well documented. While there is some information available online, the author of this paper is not aware of a comprehensive overview, let alone a detailed account of Zschweigert's life. The author assumes that one needs to research the archives in Zschweigert's home places Chemnitz, Plauen, and Hof in order to learn more.
- It is not known how Zschweigert came to the idea to construct an encryption machine and how much he was influenced by the textile technology of the time and his own inventions in this area. Perhaps, things become clearer when more about Zschweigert's biography is known.
- In this paper, the author provided a few approaches to cryptanalyze the *Zschweigert Cryptograph*. Further research might examine this topic in more detail. Especially, it will be interesting to explore additional methods for determining the number of rows of the key matrix. In addition, the use of hill climbing or a similar technique for locating the positions of the ones in the matrix deserves further investigation.
- As mentioned, the *Zschweigert Cryptograph* is one of the first (or even the first) encryption machines working with a key provided on a

punched card. Many others were to follow. A comprehensive treatise of punched cards in cryptology would be an interesting research project.

- A software implementation of the algorithm of the *Zschweigert Cryptograph* or even a simulator of the machine could be created. Such a program could be integrated into CrypTool or a similar software.

## 8 Conclusion

The 1920s were a special time in the history of mechanical encryption technology. On the one hand, the necessity for automated encryption had become evident, which led to the first generation of encryption machines being developed. On the other hand, the topic was not especially well understood yet. This resulted in numerous cipher machine designs that were not suited for practical use. For instance, the first prototypes of the Enigma (with up to seven rotors and a typewriter functionality) proved too complex and too expensive. Alexander von Kryha's encryption machines had an impressing visual design and were marketed very well, but were completely insecure. The same is true for devices such as Cryptocode and the Beyrer Cryptograph. Arvid Damm's original designs were not very successful, either.

The *Zschweigert Cryptograph* fits perfectly well with the aforementioned crypto devices. Though it implements a few promising concepts – especially the punched card used as key –, it must be considered an experimental machine that was not suited to be used in practice.

The transposition cipher the *Zschweigert Cryptograph* realizes turned out to be an evolutionary dead end. No machine of this kind ever played a major role when machine encryption became popular in later years.

## Acknowledgments

The author would like to thank Christiane Angermayr.

## References

- Fouché Gaines, Helen. 1939. *Cryptanalysis*. Dover Publications, New York, USA.
- Boris Hagelin. 1994. *The Story of Hagelin Cryptos*. Cryptologia Volume 18, 1994 (3):204-242
- David Kahn. 1996. *The Codebreakers*. Scribner, New York City, NY:764.
- LANAKI. 1996. *Classical Cryptography Course*. [www.ahazu.com/papers/lanaki:Lesson 21](http://www.ahazu.com/papers/lanaki:Lesson%20).
- Randall K. Nichols. 1996. *ICSA Guide to Cryptography*. McGraw-Hill, New York City, NY:153.
- Paul Reuvers, Marc Simons. 2019. *The HC-9*. [www.cryptomuseum.com](http://www.cryptomuseum.com).
- Paul Reuvers, Marc Simons. 2019. *The Fialka*. [www.cryptomuseum.com](http://www.cryptomuseum.com).
- Paul Reuvers, Marc Simons. 2019. *The KW-7*. [www.cryptomuseum.com](http://www.cryptomuseum.com).
- Klaus Schmeh. 2006. *The East German Encryption Machine T-310 and the Algorithm It Used*. Cryptologia Volume 30, 2006 (3):251-257
- Klaus Schmeh. 2017. *A mird in the hand is worth two in the mush: Solving ciphers with Hill Climbing*. <http://scienceblogs.de/klausis-krypto-kolumne/2017/03/26/a-mird-in-the-hand-is-worth-two-in-the-mush-solving-ciphers-with-hill-climbing/>
- Wikipedia entry "Jacquard machine". retrieved 2019-12-15.
- Wendell E. Wilson. 2019. *Mineralogical Record*. Biographical Archive, at [www.mineralogicalrecord.com](http://www.mineralogicalrecord.com).
- Rudolf Zschweigert. 1908. *Einrichtung zum Verstellen des Strickrahmens für automatische Strickmaschinen*. Patentschrift 45620 des Eidgenössischen Amts für geistiges Eigentum vom 31. August 1908.
- Rudolf Zschweigert. 1908. *Kartenschlagmaschine zum Lochen von Karten für Vorrichtungen zum automatischen Bewegen von Strickrahmen*. Patentschrift 46570 des Eidgenössischen Amts für geistiges Eigentum vom 31. August 1908.
- Rudolf Zschweigert. 1920. *Maschine zum Herstellen chiffrierter Schriftstücke*. Patentschrift 329067 des Reichspatentamts ausgegeben am 12. November 1920.

# Cracking Matrix Modes of Operation with Goodness-of-Fit Statistics

George Teşeleanu 

Advanced Technologies Institute  
10 Dinu Vintilă, Bucharest, Romania  
tgeorge@dcti.ro

and Simion Stoilow Institute of Mathematics of the Romanian Academy  
21 Calea Grivitei, Bucharest, Romania

## Abstract

The Hill cipher is a classical poly-alphabetical cipher based on matrices. Although known plaintext attacks for the Hill cipher have been known for almost a century, feasible ciphertext only attacks have been developed only about ten years ago and for small matrix dimensions. In this paper, we extend the ciphertext only attacks against the Hill cipher in two ways. First, we describe an attack against the affine version of the Hill cipher. Secondly, we show how to extend the (affine) Hill attack to several modes of operations. We also provide the reader with several experimental results and show how the message's language can influence the presented attacks.

## 1 Introduction

Two classical ciphers based on linear algebra are the Hill cipher (Hill, 1929) and its affine version (Hill, 1931). Both use invertible matrices over integers modulo  $a$  to encipher messages, where  $a$  is the size of the language alphabet  $\mathcal{A}$ . The first step of the encryption process is the encoding of each plaintext letter into a numerical equivalent. The simplest encoding is "a" = 0, "b" = 1 and so on. After encoding, the plaintext is divided into blocks of size  $\lambda$  and, then, each block is multiplied with an invertible matrix of size  $\lambda$ . In the affine case, a second matrix is added to the result. After each block is transformed, the result is converted back into letters. To decipher messages, one must perform the above steps in reverse.

Although both ciphers are vulnerable to known plaintext attacks<sup>1</sup>, efficient ciphertext only attacks

have been developed only a decade ago (Bauer and Millward, 2007) and only for the Hill cipher<sup>2</sup> with small  $\lambda$ s. Note that as  $\lambda$  increases simple brute force attacks fail. For example, in the case of the Hill cipher with  $a = 26$ , we have around  $2^{17}$  keys for  $\lambda = 2$ ,  $2^{40}$  keys for  $\lambda = 3$  and  $2^{73}$  keys for  $\lambda = 4$  (Bauer and Millward, 2007). According to (Overbey et al., 2005; Bauer, 2002), given  $a$  and  $\lambda$  the exact number of invertible matrices can be computed. Note that in the case of the affine Hill cipher the computational effort made to brute force the Hill cipher is multiplied with  $a^\lambda$ .

In 2007, Bauer and Millward (Bauer and Millward, 2007) introduced a ciphertext only attack for the Hill cipher<sup>3</sup>, that was later improved in (Yum and Lee, 2009; Leap et al., 2016; McDevitt et al., 2018). The attack was independently published by Khazaei and Ahmadi (Khazaei and Ahmadi, 2017). The main idea of these attacks is to do a brute force attack on the key rows, instead of the whole matrix, and then recover the decryption matrix.

In (Kiele, 1990), Kiele suggests the usage of block-chaining procedures to complicate the algebraic cryptanalytic techniques developed for the Hill cipher. We will show in this paper how to adapt the attacks described in (Bauer and Millward, 2007; Yum and Lee, 2009; Khazaei and Ahmadi, 2017) to different modes of operation (not only the block-chaining one) for both the Hill cipher and its affine version. Note that some modes do not require the key to be invertible, thus the attack presented in (Leap et al., 2016) does not work for all Hill based modes. For uniformity, we will only extend Yum and Lee's attack and leave as future work the extension of (Leap et al., 2016) to modes requiring invertible matrices. We stress that

<sup>1</sup>*i.e.* after a number of known messages are encrypted, one can easily recover the encryption key(s) if he has access to the corresponding ciphertexts.

<sup>2</sup>To the authors' knowledge no attack against the affine Hill cipher has been published.

<sup>3</sup>Bauer and Millward's attack for  $\lambda = 3$  was previously and independently described online by Wutka (Wutka, ).

out of the three attacks (Bauer and Millward, 2007; Yum and Lee, 2009; Khazaei and Ahmadi, 2017) Yum and Lee’s attack has the best performance to message recovery ratio.

Another paper that motivated this study is (Bauer et al., 2016). The authors of (Bauer et al., 2016) conjecture that the fourth cryptogram of the Kryptos sculpture (kry, 2020) is either encrypted using the affine Hill cipher or some other sort of cipher mode of operation. We provide the reader with a preliminary study of these conjectures. To prove or disprove these conjectures, one has to find a way to adapt all the presented ciphertext attacks to the secret encoding versions of the (affine) Hill cipher and their corresponding modes of operation. Various partial answers for the secret encoding Hill cipher are provided in (Yum and Lee, 2009).

**Structure of the paper.** Notations and definitions are presented in Section 2. The core of the paper consists of two parts, Sections 3 and 4, that contain several key ranking functions and ciphertext only attacks. Experimental results are provided in Section 5. We conclude in Section 6. The letter frequencies use in our attacks are given in Appendix A.

## 2 Preliminaries

**Notations.** Throughout the paper,  $\lambda$  will denote a security parameter. We use the notation  $x \xleftarrow{\$} X$  when selecting a random element  $x$  from a sample space  $X$ . We also denote by  $x \leftarrow y$  the assignment of value  $y$  to variable  $x$ . The subset  $\{0, \dots, q-1\} \in \mathbb{N}$  will be referred to as  $[0, q]$ . The set of matrices with  $\alpha$  rows,  $\beta$  columns and entries from  $\mathbb{G}$  is denoted by  $M(\alpha, \beta, \mathbb{G})$ , the set of invertible matrices by  $GL(\alpha, \mathbb{G})$  and the transpose of matrix  $A$  by  $A^T$ . The number of letters in a string  $m$  is represented by  $|m|$  and the set of all strings by  $\mathcal{A}^\times$ .

In this paper we use some C++ language operators (i.e.  $==$  for equality testing,  $+=$ ,  $*=$  as compound assignment operators,  $++$  for incrementing a variable and  $\&$  as reference to a variable) as well as some native function (i.e.  $size()$  for returning the size of the object,  $substring(pos, npos)$  for returning a substring starting from  $pos$  and containing  $npos$  characters,  $push_back(val)$  to add  $val$  at the end of a vector and  $sort$  to sort a vector in descending order). For initializing all the entries

of a vector  $vec$  with a value  $val$  we use the notation  $vec \leftarrow \{val\}$ . When presenting algorithms, we consider only lower case messages represented by ASCII codes (i.e. "c" – "a" = 99 – 97 = 2).

**Conventions.** To minimize repetitions, we employ the following system. When reading the attacks against the Hill based modes of operation we invite the reader to ignore red colored text, while in the case of the affine Hill based modes, the blue text. Also, when describing algorithms, we prefer using verbose names for variables, while, for mathematical descriptions, we prefer notations. The last convention used is to store constants in look-up tables when their size is small (e.g. letter frequencies) and in maps, otherwise (e.g. quadgraph frequencies).

### 2.1 Ciphers

A cipher consists of three probabilistic polynomial-time algorithms: *Setup*, *Encrypt* and *Decrypt*. The first one takes as input a security parameter and outputs the secret key. The secret key together with the *Encrypt* algorithm are used to encrypt a message  $m$ . The last algorithm decrypts any message encrypted using the known secret key.

**Hill cipher.** The Hill cipher is a poly-alphabetical cipher based on linear algebra introduced by Lester S. Hill in (Hill, 1929). We briefly provide the algorithms for the Hill cipher.

*Setup*( $\lambda$ ): Choose  $K_1 \xleftarrow{\$} GL(\lambda, \mathbb{Z}_a)$ . Also, choose a public one-to-one function  $convert : \mathcal{A}^\times \rightarrow \mathbb{Z}_a^\times$  and compute its inverse  $unconvert : \mathbb{Z}_a^\times \rightarrow \mathcal{A}^\times$ . Output the secret key is  $sk = K_1$ . Publish the *convert* and *unconvert* functions.

*Encrypt*( $sk, m$ ): Pad message  $m$  until  $|m| \equiv 0 \pmod{\lambda^4}$ . Convert and divide  $m$  into blocks  $convert(m) = m_1 \parallel \dots \parallel m_\ell$ , where  $|m_i| = \lambda$ . Compute  $c_i^T \leftarrow K_1 \cdot m_i^T$ . Output the ciphertext  $c = unconvert(c_1 \parallel \dots \parallel c_\ell)$ .

*Decrypt*( $sk, c$ ): Divide  $convert(c)$  into  $\ell$  blocks and compute  $m_i^T \leftarrow K_1^{-1} \cdot c_i^T$ . Recover  $m$  by applying *unconvert* and removing the padding.

**Affine Hill cipher.** An affine variation of the Hill cipher was introduced in (Hill, 1931). We shortly provide the algorithms for the affine Hill cipher.

<sup>4</sup>Usually an uncommon letter, such as "x", is appended to  $m$  until we get the desired length.



*Setup*( $\lambda$ ): Choose  $K_1 \xleftarrow{\$} GL(\lambda, \mathbb{Z}_a)$  and  $K_2 \xleftarrow{\$} M(\lambda, 1, \mathbb{Z}_a)$ . Also, choose a public one-to-one function  $convert : \mathcal{A}^\times \rightarrow \mathbb{Z}_a^\times$  and compute its inverse  $unconvert : \mathbb{Z}_a^\times \rightarrow \mathcal{A}^\times$ . Output the secret key is  $sk = (K_1, K_2)$ . Publish the  $convert$  and  $unconvert$  functions.

*Encrypt*( $sk, m$ ): Pad message  $m$  until  $|m| \equiv 0 \pmod{\lambda}$ . Convert and divide  $m$  into blocks  $convert(m) = m_1 \parallel \dots \parallel m_\ell$ , where  $|m_i| = \lambda$ . Compute  $c_i^T \leftarrow K_1 \cdot m_i^T + K_2$ . Output the ciphertext  $c = unconvert(c_1 \parallel \dots \parallel c_\ell)$ .

*Decrypt*( $sk, c$ ): Divide  $convert(c)$  into  $\ell$  blocks and compute  $m_i^T \leftarrow K_1^{-1} \cdot (c_i^T - K_2)$ . Recover  $m$  by applying  $unconvert$  and removing the padding.

**Affine variations.** In Table 1 we present all the possible affine variations of the Hill cipher. Note that  $K_3 \xleftarrow{\$} M(\lambda, 1, \mathbb{Z}_a)$ . After performing some computations, we can see that all variations can be decrypted using the function  $f(c_i) = K'_1 \cdot c_i^T + K'_2$ . Since we are interested only in recovering the encrypted messages and not the initial secret keys, all the presented attacks try to recover  $K'_1$  and  $K'_2$ . Thus, for the affine Hill cipher we consider  $f$  as the decryption function.

<i>Encrypt</i>	<i>Decrypt</i>
$c_i^T \leftarrow K_1 \cdot m_i^T + K_2$	$m_i^T \leftarrow K_1^{-1} \cdot (c_i^T - K_2)$
$c_i^T \leftarrow K_1 \cdot (m_i^T + K_2)$	$m_i^T \leftarrow K_1^{-1} \cdot c_i^T - K_2$
$c_i^T \leftarrow K_1 \cdot (m_i^T + K_2) + K_3$	$m_i^T \leftarrow K_1^{-1} \cdot (c_i^T - K_3) - K_2$
$K'_1$	$K'_2$
$K_1^{-1}$	$-K_1^{-1} K_2$
$K_1^{-1}$	$-K_2$
$K_1^{-1}$	$-K_1^{-1} K_3 - K_2$

Table 1: Affine variations of the Hill cipher.

## 2.2 Cipher Modes of Operation

When we encrypt messages block by block<sup>5</sup>, equal blocks are mapped into equal ciphertexts. Thus, block patterns are preserved. In some cases, this leakage can lead to security concerns. To address this issue several cipher modes of operation were introduced (Dworkin, 2001): CBC, CTR, CFB and OFB.

In (Alagic and Russell, 2017), the authors introduce a generalization of the CBC-MAC construction<sup>6</sup>. Based on Alagic et al.'s generalization, we

<sup>5</sup>ECB mode of operation

<sup>6</sup>the XOR operation is replaced with a generic group operation

present a possible adaptation of the CBC, CTR and CFB modes of operation to the (affine) Hill cipher. Note that the CFB and CTR modes do not require  $K_1$  to be invertible.

Let  $E_k, D_k : M(\lambda, \lambda, \mathbb{Z}_a) \rightarrow M(\lambda, \lambda, \mathbb{Z}_a)$  be the matrix transformations of the (affine) Hill cipher's encryption and decryption. We further describe the encryption and decryption algorithms for CBC and CFB.

*Encrypt*( $sk, m$ ): Choose  $iv \xleftarrow{\$} M(1, \lambda, \mathbb{Z}_a)$  and pad message  $m$  until  $|m| \equiv 0 \pmod{\lambda}$ . Convert and divide  $m$  into blocks  $convert(m) = m_1 \parallel \dots \parallel m_\ell$ , where  $|m_i| = \lambda$ . Let  $m_0 \leftarrow IV$ . For CBC compute  $c_i \leftarrow E_k(c_{i-1} + m_i)$ , while for CFB compute  $c_i \leftarrow E_k(c_{i-1}) + m_i$ . Let  $c = unconvert(c_1 \parallel \dots \parallel c_\ell)$ . The output is ciphertext  $(iv, c)$ .

*Decrypt*( $sk, iv, c$ ): Convert and divide  $c$  into  $\ell$  blocks. For CBC compute  $m_i \leftarrow D_k(c_i) - c_{i-1}$  and for CFB compute  $m_i \leftarrow c_i - E_k(c_{i-1})$ . Recover  $m$  by applying  $unconvert$  and removing the padding.

In the case of CTR, the sender and the receiver each keep a state  $ctr \xleftarrow{\$} M(1, \lambda, \mathbb{Z}_a)$  that is updated before each encryption.

*Update*( $ctr$ ): Let  $ctr^T = (\alpha_0, \dots, \alpha_{\lambda-1})$  and  $i \leftarrow \lambda - 1$ . Compute the following

1.  $\alpha_i \leftarrow (\alpha_i + 1) \pmod{a}$ ,
2. If  $\alpha_i == 0$ , then  $i \leftarrow (i - 1) \pmod{\lambda}$  and go to step 1.

*Encrypt*( $sk, m$ ): Pad message  $m$  until  $|m| \equiv 0 \pmod{\lambda}$ . Convert and divide  $m$  into blocks  $convert(m) = m_1 \parallel \dots \parallel m_\ell$ , where  $|m_i| = \lambda$ . Compute  $ctr \leftarrow Update(ctr)$  and  $c_i \leftarrow E_k(ctr) + m_i$ . The output is ciphertext  $c = unconvert(c_1 \parallel \dots \parallel c_\ell)$ .

*Decrypt*( $sk, iv, c$ ): Convert and divide  $c$  into  $\ell$  blocks. Compute  $ctr \leftarrow Update(ctr)$  and  $m_i \leftarrow c_i - E_k(ctr)$ . Recover  $m$  by applying  $unconvert$  and removing the padding.

A generalization of the OFB mode can also be derived. Unfortunately, our attacks do not apply to it. Thus, we omit OFB's description.

## 2.3 Statistical Models

In order to rank<sup>7</sup> all possible rows for the decryption key, Yum and Lee (Yum and Lee, 2009) introduce a goodness-of-fit score function. Compared to the score functions presented in (Bauer and Millward, 2007; Khazaei and Ahmadi, 2017), Yum and Lee’s function describes the exact probability of the recovered plaintext. We briefly describe the goodness-of-fit score function in Algorithm 1.

Let  $E_K$  and  $D_K$  be the encryption and, respectively, decryption function of a cipher. Also, let  $c \leftarrow E_K(m)$  be the given cryptogram and  $K'$  the key we want to rank. The goodness-of-fit function takes as input the letter frequency table  $letter\_freq$  associated with the language  $m$  is written in (see Appendix A for some examples) and the letter frequency table  $occ$  observed in  $D_{K'}(c)$ .

**Algorithm 1.** The goodness-of-fit score function.

---

**Input:** A vector of letter occurrences  $occ$ .  
**Output:**  $occ$ ’s goodness-of-fit score  $scr$ .

```

1 Function  $gof(letter\_freq, occ)$ :
2    $scr \leftarrow 1$ ;
3   for  $i \in [0, alph\_sz]$  do
4      $scr *= letter\_freq[i]^{occ[i]} / occ[i]!$ 
5   return  $scr$ ;
```

---

To automatically separate meaningful messages from random texts, we use an approach similar with the ones described in (Hasinoff, ; Lyons, 2012). When testing a list of strings for meaning, we first score each of them using Algorithm 2 and, then, output the highest scoring message.

The first and second inputs of the score function are a string  $in$  and the block frequency map (in our case either a digraph  $di\_freq$  or a quadgraph  $quad\_freq$  frequency map) associated with the language we are interested in. The fourth variable  $nb\_letters$  controls if we are observing digraphs (i.e.  $nb\_letters = 2$ ) or quadgraph (i.e.  $nb\_letters = 4$ ). When computing block frequency maps, some blocks may be missing entirely from the training corpus. To avoid assigning a likelihood of zero to these blocks, we use the *ad hoc* method found in (Lyons, 2012)<sup>8</sup>.

To ease description, all frequency tables/maps will be implicit when presenting algorithms, un-

<sup>7</sup>according to their relevance to a given cryptogram

<sup>8</sup>i.e.  $block\_def \leftarrow \log_{10}(0.01/nb\_blocks)$ , where the total number of blocks found in the training corpus is denoted by  $nb\_blocks$

**Algorithm 2.** The score function.

**Input:** A string  $in$ , the bound  $nb\_rows$ .

**Output:** The string’s score  $scr$ .

---

```

1 Function  $scr\_fct(in, block\_freq,$ 
    $block\_def, nb\_letters)$ :
2    $scr \leftarrow 0$ ;
3   for  $i \in [0, in.size() - nb\_letters]$  do
4      $temp \leftarrow in.substr(i, nb\_letters)$ ;
5     if  $temp \in block\_freq$  then
6        $scr += block\_freq[temp]$ ;
7     else
8        $scr += block\_def$ ;
9   return  $scr$ ;
```

---

less otherwise specified.

## 3 Ranking Functions

The first step in attacking the (affine) Hill cipher and the associated modes of operation is to rank all possible rows according to their relevance to a given cryptogram. In this section we describe the ranking functions latter used in the attacks presented in Section 4.

### 3.1 (Affine) ECB

In (Yum and Lee, 2009), the authors describe a ranking algorithm for the Hill cipher. We chose to present it in this section (Algorithm 3, red text) because it is tightly linked with the affine version we introduce (Algorithm 3, blue text).

Let  $mat\_sz = \lambda = 2$  and let  $enc = c$  be a Hill cipher cryptogram. We illustrate the influence of a given row on the decrypted plaintext  $p$  in Figure 1. We observe that if the first and second rows are equal we obtain the same letter  $p^i$  after decryption. Thus, is enough to decrypt the ciphertext using only the first row (*hill\_line\_dec*). Since we do not have duplicates, the resulting text  $msg$  is  $\lambda$  times shorter than  $c$ . After decryption we compute the letter frequency observed in  $msg$  and use the *gof* function to obtain the row’s score. After all the rows have been ranked, we sort them in descending order according to their score. In the case of the affine Hill cipher the ranking algorithm is similar. The main difference is that instead of having to brute force  $k_0$  and  $k_1$ , we also have to do an exhaustive search on  $k_2$  (Figure 2). The algorithm for the generic case is given in Algorithm 3.

In some cases storing a vector of size  $a^{\lambda 9}$  might be troublesome. Thus, we further consider that  $fit.size() = B$ , where  $B$  is dependent on the available memory. Note that in this case  $fit$  must be

<sup>9</sup> $a^{\lambda+1}$  for the affine version

sorted and when an element is inserted we first check if its score is higher than the lowest score from *fit* and if it is, the element replaces the lowest scoring element from *fit*.

We usually work with small values of *alph\_sz* and *msg.size()* and thus we consider the complexity of the *gof* and of multiplication as  $O(1)$ . Hence, the Hill version of Algorithm 5 performs  $O(a^\lambda)$  *hill\_line\_decs* and sorts a vector of size  $B$ . So, it has a complexity of  $O(\lambda a^\lambda + B \log B)$ . In the case of the affine Hill cipher, the only change is that we perform  $O(a^{\lambda+1})$  *aff\_hill\_line\_decs*. So, the complexity becomes  $O(\lambda a^{\lambda+1} + B \log B)$ .

$$\begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} = \begin{array}{|c|} \hline p^i \\ \hline \end{array}$$

(a) Line 1.

$$\begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} = \begin{array}{|c|} \hline p^i \\ \hline p^i \\ \hline \end{array}$$

(b) Line 2.

Figure 1: Line propagation in ECB.

$$\begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} + \begin{array}{|c|} \hline k_2 \\ \hline k_2 \\ \hline \end{array} = \begin{array}{|c|} \hline p^i \\ \hline \end{array}$$

(a) Line 1.

$$\begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} + \begin{array}{|c|} \hline k_2 \\ \hline k_2 \\ \hline \end{array} = \begin{array}{|c|} \hline p^i \\ \hline p^i \\ \hline \end{array}$$

(b) Line 2.

Figure 2: Line propagation in affine ECB.

### 3.2 (Affine) CBC, CTR, CFB

Again, let *mat\_sz* = 2 and let *enc* be a Hill cipher cryptogram. The effect of a given row on the decrypted plaintext is shown in Figure 3 for CBC, in Figure 4 for CTR and in Figure 5 for CFB. Compared to ECB, we can easily see that if the first and second row are identical the resulting letters are different. Thus, we need the full decryption of the Hill cipher to rank rows. After decryption, we break the resulting *msg* in two parts *msg<sub>0</sub>* and

**Algorithm 3.** The algorithm for ranking all possible rows for (affine) ECB.

**Input:** The ciphertext *enc*.

**Output:** A vector *fit* containing all possible rows sorted by the goodness-of-fit score.

```

1 Function aff_hill_line_dec(conv, k1, k2):
2   msg_int[enc.size() / mat_sz] ← {0};
3   for i ∈ [0, conv.size() / mat_sz] do
4     for j ∈ [0, mat_sz] do
5       idx ← i · mat_sz + j;
6       msg_int[i] ← (msg_int[i] +
7         k1[j] · conv[idx]) mod alph_sz;
8     msg_int[i] ← (msg_int[i] +
9       k2[i mod mat_sz]) mod alph_sz;
10    return msg_int;
11 Function aff_ecb_rank(enc):
12 for
13   k1[0], ..., k1[mat_sz - 1] ∈ [0, alph_sz]
14 do
15   for k2 ∈ [0, alph_sz] do
16     occ[alph_sz] ← {0};
17     conv ← convert(enc);
18     msg_int ←
19       hill_line_dec(enc, k1);
20     msg_int ←
21       aff_hill_line_dec(enc, k1, k2);
22     msg ← unconvert(msg_int)
23     for i ∈ [0, msg.size()] do
24       | occ[msg[i] - "a"]++;
25     scr ← gof(letter_freq, occ);
26     fit.push_back((k1, scr));
27     fit.push_back((k1, k2, scr));
28   fit.sort();
29   return fit;

```

*msg<sub>1</sub>*. The first part contains the letters in even positions and the second one the letters in odd positions. After we score each part, we store them in *fit*[0] and, respectively, *fit*[1]. The last step is to sort the two vectors in descending order by score. The case of the affine Hill cipher is similar.

For the Hill modes attack, we perform  $O(a^\lambda)$  decryptions, while for the affine version the number of decryptions is  $O(a^{\lambda+1})$ . Both algorithms sort  $\lambda$  vectors of size  $B$ . Thus, the complexities are  $O(\lambda^2 a^\lambda + \lambda B \log B)$  and  $O(\lambda^2 a^{\lambda+1} + \lambda B \log B)$  for the Hill attack and, respectively, for the affine attack.

## 4 Message Recovering Attacks

After the ranking step is over, we can proceed to the recovering step. When searching for the original message a lot of random text is produced. To filter random messages from ones with meaning we use the *scr\_fct* to score each message and we always output the highest scoring one.

$$\begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^{i-1} \\ \hline c_1^{i-1} \\ \hline \end{array} - \begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} = \begin{array}{|c|} \hline p_0^i \\ \hline \\ \hline \end{array}$$

(a) Line 1.

$$\begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^{i-1} \\ \hline c_1^{i-1} \\ \hline \end{array} - \begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} = \begin{array}{|c|} \hline p_0^i \\ \hline p_1^i \\ \hline \end{array}$$

(b) Line 2.

Figure 3: Line propagation in CBC.

$$\begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} - \begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline n_0 \\ \hline n_1 \\ \hline \end{array} = \begin{array}{|c|} \hline p_0^i \\ \hline p_1^i \\ \hline \end{array}$$

(a) Line 1.

$$\begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} - \begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline n_0 \\ \hline n_1 \\ \hline \end{array} = \begin{array}{|c|} \hline p_0^i \\ \hline p_1^i \\ \hline \end{array}$$

(b) Line 2.

Figure 4: Line propagation in CTR.

$$\begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} - \begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^{i-1} \\ \hline c_1^{i-1} \\ \hline \end{array} = \begin{array}{|c|} \hline p_0^i \\ \hline p_1^i \\ \hline \end{array}$$

(a) Line 1.

$$\begin{array}{|c|} \hline c_0^i \\ \hline c_1^i \\ \hline \end{array} - \begin{array}{|c|c|} \hline k_0 & k_1 \\ \hline k_0 & k_1 \\ \hline \end{array} \times \begin{array}{|c|} \hline c_0^{i-1} \\ \hline c_1^{i-1} \\ \hline \end{array} = \begin{array}{|c|} \hline p_0^i \\ \hline p_1^i \\ \hline \end{array}$$

(b) Line 2.

Figure 5: Line propagation in CFB.

#### 4.1 (Affine) ECB

The authors of (Bauer and Millward, 2007; Yum and Lee, 2009) describe the message recovering algorithm for the Hill cipher, but they do not provide an automatic detection method for the original message. On the other hand, the authors of (Khazaei and Ahmadi, 2017) trade-off success probability for a unique output. The gap is filled in (Leap et al., 2016). We present the algorithm in this section (Algorithm 5, red text), instead of Sec-

**Algorithm 4.** The algorithm for ranking all possible rows for (affine) CBC, CTR, CFB.

**Input:** The ciphertext *enc* and the initialization vector *iv*.

**Output:** A family of vectors *fit* containing all possible rows sorted by the goodness-of-fit score.

```

1 Function aff_mode_rank(enc, iv):
2   for a[0], ..., a[mat_sz - 1] ∈ [0, alph_sz]
3     do
4       for b ∈ [0, alph_sz] do
5         occ[mat_sz][alph_sz] ← {0};
6         for i ∈ [0, mat_sz] do
7           for j ∈ [0, mat_sz] do
8             k1[i][j] ← a[j];
9             k2[i] ← b;
10        conv ← convert(enc);
11        msg_int ← mode_dec(enc, iv, k1);
12        msg_int ← aff_mode_dec(enc, iv, k1, k2);
13        msg ← unconvert(msg_int)
14        for i ∈ [0, msg.size() / mat_sz]
15          do
16            for j ∈ [0, mat_sz] do
17              occ[j][msg[i · mat_sz + j] - "a"]++;
18            scr ← gof(letter_freq, occ[i]);
19            fit[i].push_back((a, scr));
20            fit[i].push_back((a, b, scr));
21        for i ∈ [0, mat_sz] do
22          fit[i].sort();
23    return fit;

```

tion 2, because of its link to the affine version we introduce (Algorithm 5, blue text). Due to better results in practice, in Algorithm 5 we use a different scoring function<sup>10</sup> than the one from (Leap et al., 2016)<sup>11</sup>. Also, compared to (Leap et al., 2016), we only output the highest scoring message without lowering the success probability.

After ranking all possible rows, we need to find the decryption key's rows (*ck\_vars*) and their order (*ck\_var*). Thus, Algorithm 5 checks all possible row combinations with index less than *nb\_rows* = *B*. Note that the success probability is dependent on *nb\_rows*<sup>12</sup>. After selecting  $\lambda$  rows from *fit*, we test all possible row permutations<sup>13</sup>, decrypt *enc* and rank the result. If one of the decrypted texts has a higher score than the stored message *glb\_msg*, we overwrite *glb\_msg* and up-

<sup>10</sup>based on quadgraphs

<sup>11</sup>based on the index of coincidence

<sup>12</sup>see Section 5 for the experimental results

<sup>13</sup> $\sigma_i$  denotes the *i*th permutation of length *mat\_size*

date  $glb\_scr$ . The main differences between the Hill cipher attack and the affine Hill cipher attack are: the call to the affine ranking algorithm, the creation of  $k_2$  and the call to the affine decryption algorithm.

---

**Algorithm 5.** The algorithm for breaking (affine) ECB.

---

**Input:** The ciphertext  $enc$ , the bound  $nb\_rows$ .

**Output:** The best possible message  $glb\_msg$  and its associated score  $glb\_scr$ .

```

1 Function  $ck\_var(enc, rows, \&glb\_scr,$ 
   $\&glb\_msg)$ :
2    $best\_scr \leftarrow -\infty$ ;
3   for  $i \in [0, mat\_sz]$  do
4     for  $s \in [0, mat\_sz]$  do
5       for  $t \in [0, mat\_sz]$  do
6          $k_1[s][t] \leftarrow rows[\sigma_i[s]].k_1[t]$ ;
7          $k_2[s] \leftarrow rows[\sigma_i[s]].k_2$ ;
8          $try\_msg \leftarrow hill\_dec(enc, k_1)$ ;
9          $try\_msg \leftarrow aff\_hill\_dec(enc,$ 
           $k_1, k_2)$ ;
10         $try\_scr \leftarrow scr\_fct(try\_msg,$ 
           $quad\_freq, quad\_freq, 4)$ ;
11        if  $try\_scr > best\_scr$  then
12           $best\_scr \leftarrow try\_scr$ ;
13           $best\_msg \leftarrow try\_msg$ ;
14        if  $best\_scr > glb\_scr$  then
15           $glb\_scr \leftarrow best\_scr$ ;
16           $glb\_msg \leftarrow best\_msg$ ;
17 Function  $ck\_vars(enc, fit, nb\_rows)$ :
18    $glb\_scr \leftarrow -\infty$ ;
19    $glb\_msg \leftarrow ""$ ;
20   for  $i_0 \in [0, nb\_rows]$  do
21     for  $i_1 \in [i_0 + 1, nb\_rows]$  do
22       ...
23       for  $i_{mat\_sz-1} \in$ 
           $[i_{mat\_sz-2} + 1, nb\_rows]$  do
24          $try\_rows \leftarrow \emptyset$ ;
25         for  $j \in [0, mat\_sz]$  do
26            $try\_rows.push\_back(fit[i_j])$ ;
27          $ck\_var(enc, try\_rows,$ 
           $glb\_scr, glb\_msg)$ ;
28   return  $(glb\_scr, glb\_msg)$ ;
29 Function  $aff\_ecb\_attack(enc, nb\_rows)$ :
30    $fit \leftarrow aff\_ecb\_rank(enc)$ ;
31   return  $ck\_var(enc, fit, nb\_rows)$ ;

```

---

For the same reasons as in Section 3.1, we further consider the complexity of the  $scr\_fct$  as  $O(1)$ . After the row ranking step, both message recovering algorithms perform  $O(B!/(B-\lambda)!)$  decryptions. Thus, the complexities for the Hill attack and for the affine attack are  $O(\lambda a^\lambda + B \log B + \lambda^2 B!/(B-\lambda)!)$  and, respectively,  $O(\lambda^2 a^{\lambda+1} + B \log B + \lambda^2 B!/(B-\lambda)!)$ .

## 4.2 (Affine) CBC, CTR, CFB

The main difference between ECB and the other modes is that after the ranking step is over, in the former case we know the exact position of the key rows. Thus, in Algorithm 6 we iterate over all rows ( $ck\_vars\_mode$ ), decrypt the cryptogram and then score the result ( $ck\_var\_mode$ ).

The  $ck\_vars\_mode$  function performs  $O(B^\lambda)$  decryptions. Thus, Algorithm 6's complexity for the Hill based modes attack and for the affine versions is  $O(\lambda^2 a^\lambda + \lambda B \log B + \lambda^2 B^\lambda)$  and, respectively,  $O(\lambda^2 a^{\lambda+1} + \lambda B \log B + \lambda^2 B^\lambda)$ .

---

**Algorithm 6.** The algorithm for breaking (affine) CBC, CTR, CFB.

---

**Input:** The ciphertext  $enc$ , the initialization vector  $iv$ , the bound  $nb\_rows$ .

**Output:** The best possible message  $glb\_msg$  and its associated score  $glb\_scr$ .

```

1 Function  $ck\_var\_mode(enc, iv, rows,$ 
   $\&glb\_scr, \&glb\_msg)$ :
2   for  $s \in [0, mat\_sz]$  do
3     for  $t \in [0, mat\_sz]$  do
4        $k_1[s][t] \leftarrow rows[s].a[t]$ ;
5        $k_2[s] \leftarrow rows[s].b$ ;
6        $try\_msg \leftarrow mode\_dec(enc, iv, k_1)$ ;
7        $try\_msg \leftarrow aff\_mode\_dec(enc, iv,$ 
           $k_1, k_2)$ ;
8        $try\_scr \leftarrow scr\_fct(try\_msg,$ 
           $quad\_freq, quad\_freq, 4)$ ;
9       if  $try\_scr > glb\_scr$  then
10         $glb\_scr \leftarrow try\_scr$ ;
11         $glb\_msg \leftarrow try\_msg$ ;
12 Function  $ck\_vars\_mode(enc, fit,$ 
   $nb\_rows)$ :
13    $glb\_scr \leftarrow -\infty$ ;
14    $glb\_msg \leftarrow ""$ ;
15   for  $i_0 \in [0, nb\_rows]$  do
16     for  $i_1 \in [0, nb\_rows]$  do
17       ...
18       for  $i_{mat\_sz-1} \in [0, nb\_rows]$  do
19          $try\_rows \leftarrow \emptyset$ ;
20         for  $j \in [0, mat\_sz]$  do
21            $try\_rows.push\_back(fit[j][i_j])$ ;
22          $ck\_var\_mode(enc, iv, try\_rows,$ 
           $glb\_scr, glb\_msg)$ ;
23   return  $(glb\_scr, glb\_msg)$ ;
24 Function  $aff\_mode\_attack(enc,$ 
   $nb\_rows)$ :
25    $fit \leftarrow aff\_mode\_rank(enc, iv)$ ;
26   return  $ck\_vars\_mode(enc, iv, fit,$ 
           $nb\_rows)$ ;

```

---

## 5 Experimental Results

We implemented Algorithms 5 and 6 in order to see the relation between  $B$  and the algorithms' suc-



cess probability. The results are presented in Tables 3 to 8. To see the influence of the message’s native language on the attack algorithms’ recovery rate, we tested this type of relation for eight languages: Danish (DN), English (EN), Finnish (FN), French (FR), German (GE), Polish (PL), Spanish (SP) and Swedish (SW). We also computed the running time of Algorithms 5 and 6 for English and  $\lambda = 2$  (Section 5.2).

In our implementations, frequency tables have  $a = 26$  values and are derived from the frequencies provided in (Lyons, 2012). For completeness, we describe the tables in Appendix A. The quadgrams for the English language are downloaded from (Lyons, 2012), while the digraph<sup>14</sup> frequencies are computed from the quadgraph map.

For computing the success probability we used 100 texts with 100 letters (without diacritical marks) for each language. Each text was encrypted with a different key(s)/initialization vector/counter. The texts are taken from news items found in the Leipzig Corpora Collection (Goldhahn et al., 2012). The keys, initialization vectors and counters are generated using the default generator found in the GMP library (gmp, ). When invertible keys were needed, we computed the inverse using the Armadillo library (Sanderson and Curtin, 2016) and tested if the determinant is coprime with 26.

## 5.1 Unicity Distance of a Cipher

When analyzing the experimental results, the reader will observe different message recovery rates for different languages. These differences arise from distinct unicity distances<sup>15</sup> for different languages. The exact formula for the unicity distance when  $a = 26$  is  $\log_2 26^\lambda / (\log_2 26 - H)$ , where  $H$  is the language’s entropy. Note that in our case the unicity distance is computed for one key row and we estimated the entropy from the frequency tables provided in Appendix A. The results for the unicity distance are provided in Table 2. We can see that in the case of the Polish language we need more letters per row than for the Finnish language. This gap will be more pronounced when determining the message recovery rates.

<sup>14</sup>If  $abcd$  is a quadgraph, we consider  $ac$  as a digraph.

<sup>15</sup>The minimum ciphertext length required to determine the secret key almost uniquely.

Language	$\lambda = 2$	$\lambda = 3$	$\lambda = 4$
Danish	15.4323	23.1485	30.8647
English	18.2180	27.3270	36.4359
Finnish	12.0307	18.0460	24.0614
French	13.3713	20.0569	26.7425
German	15.6257	23.4386	31.2515
Polish	22.3918	33.5878	44.7837
Spanish	13.7891	20.6836	27.5781
Swedish	16.4837	24.7256	32.9674

Table 2: Unicity distance.

	$B$	DN	EN	FN	FR	GE	PL	SP	SW
ECB	2	94	93	100	96	95	84	96	95
	4	99	100	100	98	100	91	100	100
CBC	1	95	95	100	99	97	84	99	99
	2	99	99	100	100	100	90	100	100
CTR	1	96	93	100	96	98	87	100	98
	2	99	98	100	99	100	90	100	100
CFB	1	97	92	99	96	95	87	98	98
	2	100	99	100	100	99	91	100	100

Table 3: Number of recovered messages for the Hill modes of operation when  $\lambda = 2$ .

	$B$	DN	EN	FN	FR	GE	PL	SP	SW
ECB	8	88	59	97	90	71	22	87	80
	16	95	77	100	95	86	45	96	94
	32	97	87	100	98	94	68	99	99
CBC	4	86	57	99	92	71	18	91	78
	8	93	68	99	96	80	34	96	86
	16	96	80	100	96	89	55	97	96
CTR	4	64	40	84	65	46	11	68	45
	8	80	59	94	87	67	19	83	66
	16	91	75	97	93	80	48	92	77
CFB	4	85	53	99	90	73	12	89	78
	8	93	66	99	94	81	36	94	87
	16	96	79	100	97	91	52	96	96

Table 4: Number of recovered messages for the Hill modes of operation when  $\lambda = 3$ .

	$B$	DN	EN	FN	FR	GE	PL	SP	SW
ECB	512	78	48	97	89	72	10	85	74
	1024	88	65	98	91	89	19	94	86
	2048	95	80	99	95	94	39	95	93
CBC	32	78	50	97	89	69	13	88	72
	64	87	67	99	91	86	21	93	84
	128	93	78	99	95	94	45	95	93
CTR	32	71	37	91	77	55	6	80	64
	64	87	58	97	90	79	21	90	83
	128	93	75	100	95	94	40	99	88
CFB	32	78	48	97	88	69	14	86	73
	64	87	65	98	91	85	18	92	85
	128	93	75	99	95	95	45	94	95

Table 5: Number of recovered messages for the Hill modes of operation when  $\lambda = 4$ .

## 5.2 Running time

In this section we provide some benchmarks for Algorithms 5 and 6. The algorithms were run

	$B$	DN	EN	FN	FR	GE	PL	SP	SW
ECB	2	89	80	100	90	88	54	93	92
	4	97	94	100	98	99	79	98	99
	8	99	99	100	99	99	87	99	100
CBC	1	93	85	100	99	85	57	96	93
	2	97	88	100	99	93	68	98	100
	4	99	95	100	99	99	78	100	100
CTR	1	92	72	100	93	90	48	96	95
	2	97	88	100	96	98	68	99	99
	4	98	97	100	99	99	78	100	100
CFB	1	89	80	100	95	91	54	98	93
	2	97	92	100	98	97	69	100	99
	4	99	97	100	99	99	83	100	100

Table 6: Number of recovered messages for the affine Hill modes of operation when  $\lambda = 2$ .

	$B$	DN	EN	FN	FR	GE	PL	SP	SW
ECB	32	70	43	97	86	49	3	85	63
	64	84	50	99	91	62	11	87	75
	128	93	65	99	93	79	21	94	88
CBC	32	71	40	98	86	47	5	83	61
	64	82	50	99	93	65	11	90	74
	128	90	65	99	93	78	25	95	97
CTR	32	35	13	56	40	19	3	37	18
	64	58	28	85	63	36	6	60	45
	128	81	49	98	82	59	13	83	77
CFB	32	70	38	97	87	50	3	83	74
	64	84	49	99	93	64	8	89	86
	128	91	63	99	93	77	23	94	96

Table 7: Number of recovered messages for the affine Hill modes of operation when  $\lambda = 3$ .

	$B$	DN	EN	FN	FR	GE	PL	SP	SW
ECB	16384	82	53	98	90	79	14	89	79
	32768	92	69	99	93	93	26	94	88
	65536	96	83	100	95	95	54	96	94
CBC	16384	80	53	98	89	76	14	88	78
	32768	89	69	99	93	92	27	94	87
	65536	96	80	100	95	95	61	96	93
CTR	16384	77	46	95	86	63	11	86	74
	32768	87	66	98	92	89	26	92	85
	65536	95	79	100	97	95	53	96	92
CFB	16384	81	53	98	89	76	15	88	77
	32768	90	68	99	93	92	27	94	87
	65536	96	81	100	95	95	59	96	93

Table 8: Number of recovered messages for the affine Hill modes of operation when  $\lambda = 4$ .

on a CPU Intel i7-4790 4.00 GHz and compiled with GCC with the O3 flag activated and the `omp_get_wtime()` function (`omp,` ) was used to compute the running times. Due to resource constraints, we stopped the experiments at  $\lambda = 3$  for the Hill attacks and at  $\lambda = 2$  for the affine attacks. To obtain a fair comparison, when computing the running times, we used higher  $B$  values than the one presented in Tables 3 to 8. We present the ex-

Mode	Hill ( $\lambda = 2$ )	Afine Hill ( $\lambda = 2$ )	Hill ( $\lambda = 3$ )
ECB	4 (100%)	8 (99%)	128 (97%)
CBC	2 (99%)	4 (95%)	128 (95%)
CTR	2 (98%)	4 (97%)	128 (96%)
CFB	2 (99%)	4 (97%)	128 (96%)

Table 9: The threshold  $B$  and the corresponding success probability for the English language.

act margins in Table 9.

In Table 10, the second and third columns contain the total time necessary to recover 100 independent texts, while the fourth column contains the total time necessary to recover 8 texts.

Mode	Hill ( $\lambda = 2$ )	Afine Hill ( $\lambda = 2$ )	Hill ( $\lambda = 3$ )
ECB	0.94057	23.1658	1415.60
CBC	1.75324	45.4769	1502.20
CTR	1.75827	45.9883	1423.39
CFB	1.75271	48.5864	1509.62

Table 10: Running times of Algorithms 5 and 6.

Let  $\lambda = 2$ . To see if the chosen bounds have the same success rate for other texts, we encrypted 1000 independent texts<sup>16</sup> and then we ran Algorithms 5 and 6. The number of plaintexts recovered is presented in Table 11. We can see that for the Hill based modes the success probabilities are almost the same, while for the affine versions the probabilities are a little lower than the initial estimates.

Cipher	ECB	CBC	CTR	CFB
Hill	995	987	982	982
Afine Hill	970	956	945	953

Table 11: Success rates for Algorithms 5 and 6 when  $\lambda = 2$ .

## 6 Conclusions

In this paper we adapted Yum and Lee’s attack to the affine Hill cipher. Also, we introduced new ranking and message recovery algorithms for the CBC, CTR and CFB modes of operation. We also conducted a series of experiments to determine and test the success rates of these algorithms.

**Future Work.** The row ranking algorithms perform the same instructions for disjoint rows. Thus,

<sup>16</sup>different from the 100 texts used for computing the bounds

an interesting implementation direction is to parallelize Algorithms 3 and 4. The recovering algorithms also perform the same instructions, but for independent keys. Hence, Algorithms 5 and 6 can also be parallelized.

Another possible speed-up is to parallelize the algorithm presented (Leap et al., 2016) for the Hill cipher. Note that this speed-up can also be applied to the Hill CBC mode. From a theoretical point of view, it would be interesting to see if the Leap et al.’s algorithm can be tweaked to work for the affine Hill cipher. If it can be tweaked we might obtain faster decryption times for the affine Hill and the corresponding CBC mode.

A time-memory trade-off attack for the Hill cipher is presented in (McDevitt et al., 2018). Thus, it might be interesting to see if this attack can be adapted to the affine version and to the (affine) modes of operation versions. From an implementation point of view, it might worth seeing if McDevitt et al.’s attack can be parallelized.

In (Yum and Lee, 2009), the authors provide a ranking algorithm when the *convert* and the *unconvert* functions are unknown, but they do not describe a message recovery algorithm. This cipher can be seen as a composition of a substitution cipher, a Hill cipher and a second substitution cipher. Note that the two substitution ciphers do not necessarily have the same key. A generic version of the secret coding cipher can be obtained by combining a generic Vigenère cipher<sup>17</sup>, a Hill cipher and a second generic Vigenère cipher. Note that in this case Yum and Lee’s ranking algorithm still works. Hence, another possible research direction is to find message recovery algorithms<sup>18</sup> for this generic cipher.

In (Hill, 1931), Hill introduces a variation of the affine Hill cipher in which the elements of the key matrix are matrices. Thus, an interesting problem is to study the impact of the message recovering algorithms on the version presented in (Hill, 1931).

## References

- [Alagic and Russell2017] Gorjan Alagic and Alexander Russell. 2017. Quantum-Secure Symmetric-Key Cryptography Based on Hidden Shifts. In *EUROCRYPT 2018*, volume 10212 of *Lecture Notes in Computer Science*, pages 65–93. Springer.
- [Bauer and Millward2007] Craig Bauer and Katherine Millward. 2007. Cracking Matrix Encryption Row by Row. *Cryptologia*, 31(1):76–83.
- [Bauer et al.2016] Craig Bauer, Gregory Link, and Dante Molle. 2016. James Sanborns Kryptos and the Matrix Encryption Conjecture. *Cryptologia*, 40(6):541–552.
- [Bauer2002] Friedrich Ludwig Bauer. 2002. *Decrypted Secrets: Methods and Maxims of Cryptology*. Springer.
- [Dworkin2001] Morris Dworkin. 2001. Recommendation for Block Cipher Modes of Operation. Methods and Techniques. Technical report, NIST.
- [gmp] The GNU Multiple Precision Arithmetic Library. <https://gmplib.org/>.
- [Goldhahn et al.2012] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *LREC 2012*, volume 29, pages 31–43. European Language Resources Association (ELRA).
- [Hasinoff] Sam Hasinoff. Solving Substitution Ciphers. <https://people.csail.mit.edu/hasinoff/pubs/hasinoff-quipster-2003.pdf>.
- [Hill1929] Lester S Hill. 1929. Cryptography in an Algebraic Alphabet. *The American Mathematical Monthly*, 36(6):306–312.
- [Hill1931] Lester S Hill. 1931. Concerning Certain Linear Transformation Apparatus of Cryptography. *The American Mathematical Monthly*, 38(3):135–154.
- [Khazaei and Ahmadi2017] Shahram Khazaei and Siavash Ahmadi. 2017. Ciphertext-Only Attack on  $d \times d$  Hill in  $O(d13^d)$ . *Information Processing Letters*, 118:25–29.
- [Kiele1990] William A Kiele. 1990. A Tensor-Theoretic Enhancement to the Hill Cipher System. *Cryptologia*, 14(3):225–233.
- [kry2020] 2020. Kryptos. <https://en.wikipedia.org/wiki/Kryptos>.
- [Leap et al.2016] Tom Leap, Tim McDevitt, Kayla Novak, and Nicolette Siermine. 2016. Further Improvements to the Bauer-Millward Attack on the Hill Cipher. *Cryptologia*, 40(5):452–468.
- [Lyons2012] James Lyons. 2012. Practical Cryptography, <http://practicalcryptography.com/>.
- [McDevitt et al.2018] Tim McDevitt, Jessica Lehr, and Ting Gu. 2018. A Parallel Time-memory Tradeoff Attack on the Hill Cipher. *Cryptologia*, 42(5):1–19.
- [omp] OpenMP. <https://www.openmp.org/>.

<sup>17</sup>By a generic Vigenère cipher we understand a Vigenère cipher with random alphabets.

<sup>18</sup>that might use Yum and Lee’s ranking algorithm

[Overbey et al.2005] Jeffrey Overbey, William Traves, and Jerzy Wojdyllo. 2005. On the Keyspace of the Hill Cipher. *Cryptologia*, 29(1):59–72.

[Sanderson and Curtin2016] Conrad Sanderson and Ryan Curtin. 2016. Armadillo: A Template-Based C++ Library for Linear Algebra. *Journal of Open Source Software*, 1(2):26.

[Wutka] Mark Wutka. The Crypto Forum, <http://s13.zetaboards.com/Crypto/topic/123721/1/>.

[Yum and Lee2009] Dae Hyun Yum and Pil Joong Lee. 2009. Cracking Hill Ciphers with Goodness-of-Fit Statistics. *Cryptologia*, 33(4):335–342.

## Appendix A Letter Frequencies

To have uniform letter frequency tables, we added the probability of letters with diacritical marks to the probability of their base letter. For example, in Danish, the letter O has a 0.0464 occurrence probability and the letter Ø one of 0.0094. We added the two and we recorded O's probability as 0.0558. Note that the frequency tables we used for computing our tables are from (Lyons, 2012).

A, Å, Æ	0.0809	J	0.0073	S	0.0581
B	0.0200	K	0.0339	T	0.0686
C	0.0056	L	0.0523	U	0.0198
D	0.0586	M	0.0324	V	0.0233
E	0.1545	N	0.0724	W	0.0007
F	0.0241	O, Ø	0.0558	X	0.0003
G	0.0408	P	0.0176	Y	0.0070
H	0.0162	Q	0.0001	Z	0.0003
I	0.0600	R	0.0896		

Table 12: Relative frequencies of Danish letters.

A	0.0855	J	0.0022	S	0.0673
B	0.0160	K	0.0081	T	0.0894
C	0.0316	L	0.0421	U	0.0268
D	0.0387	M	0.0253	V	0.0106
E	0.1210	N	0.0717	W	0.0183
F	0.0218	O	0.0747	X	0.0019
G	0.0209	P	0.0207	Y	0.0172
H	0.0496	Q	0.0010	Z	0.0011
I	0.0733	R	0.0633		

Table 13: Relative frequencies of English letters.

A, Ä	0.1580	J	0.0204	S	0.0786
B	0.0028	K	0.0497	T	0.0875
C	0.0028	L	0.0576	U	0.0501
D	0.0104	M	0.0320	V	0.0225
E	0.0797	N	0.0883	W	0.0009
F	0.0019	O, Ö	0.0605	X	0.0003
G	0.0039	P	0.0184	Y	0.0174
H	0.0185	Q	0.0001	Z	0.0005
I	0.1082	R	0.0287		

Table 14: Relative frequencies of Finnish letters.

A, Ä, Å	0.0808	J	0.0030	S	0.0798
B	0.0096	K	0.0016	T	0.0711
C, Ç	0.0344	L	0.0586	U, Û	0.0559
D	0.0408	M	0.0278	Û, Ü	
E, È, É, Ê	0.1745	N	0.0732	V	0.0129
F	0.0112	O, Ô, Æ	0.0546	W	0.0008
G	0.0118	P	0.0298	X	0.0043
H	0.0093	Q	0.0085	Y	0.0034
I, Î, Ï	0.0726	R	0.0686	Z	0.0010

Table 15: Relative frequencies of French letters.

A, Ä	0.0688	J	0.0027	S, ß	0.0656
B	0.0221	K	0.0150	T	0.0643
C	0.0271	L	0.0372	U, Ü	0.0376
D	0.0492	M	0.0275	V	0.0094
E	0.1599	N	0.0959	W	0.0140
F	0.0180	O, Ö	0.0299	X	0.0007
G	0.0302	P	0.0106	Y	0.0013
H	0.0411	Q	0.0004	Z	0.0122
I	0.0760	R	0.0771		

Table 16: Relative frequencies of German letters.

A, Ą	0.0997	J	0.0226	S, Ś	0.0504
B	0.0139	K	0.0354	T	0.0394
C, Ć	0.0422	L, Ł	0.0418	U	0.0259
D	0.0323	M	0.0273	V	0.0000
E, Ę	0.0849	N, Ń	0.0602	W	0.0478
F	0.0041	O, Ó	0.0879	X	0.0000
G	0.0154	P	0.0292	Y	0.0370
H	0.0125	Q	0.0000	Z, Ż, Ź	0.0590
I	0.0809	R	0.0506		

Table 17: Relative frequencies of Polish letters.

A	0.1250	J	0.0045	S	0.0744
B	0.0127	K	0.0008	T	0.0442
C	0.0443	L	0.0584	U	0.0400
D	0.0514	M	0.0261	V	0.0098
E	0.1324	N, Ñ	0.0731	W	0.0003
F	0.0079	O	0.0898	X	0.0019
G	0.0117	P	0.0275	Y	0.0079
H	0.0081	Q	0.0083	Z	0.0042
I	0.0691	R	0.0662		

Table 18: Relative frequencies of Spanish letters.

A, Ä, Å	0.1252	J	0.0061	S	0.0659
B	0.0154	K	0.0314	T	0.0769
C	0.0149	L	0.0528	U	0.0192
D	0.0470	M	0.0347	V	0.0242
E	0.1015	N	0.0854	W	0.0014
F	0.0203	O, Ö	0.0579	X	0.0016
G	0.0286	P	0.0184	Y	0.0071
H	0.0209	Q	0.0002	Z	0.0007
I	0.0582	R	0.0843		

Table 19: Relative frequencies of Swedish letters.

# Automatic Key Structure Extraction

**Crina Tudor, Beáta Megyesi**

Dept. of Linguistics and Philology  
Uppsala University  
Sweden  
first.last@lingfil.uu.se

**Benedek Láng**

Dept. of Philosophy and History of Science  
Budapest University of Technology  
and Economics, Hungary  
lang@filozofia.bme.hu

## Abstract

Studying original cipher keys constructed throughout history gives important insights into encryption methods and cipher systems. We can study the type of encryption used, the code structure and their corresponding plaintext entities, be it letters, morphemes, words, or named entities. The insights can lead us to better decryption methods, and the understanding of the development of historical ciphers. In this paper, we present a tool for automatic key structure extraction that describes the symbol system and the code structure along with the encoded plaintext features and the mapping between the two. The tool is aimed at the empirical study of historical keys given transcribed keys.

## 1 Introduction

In historical cryptology, the key (also called *clavis*), according to the definition of Kahn (1996), “specifies the arrangement of letters within a cipher alphabet”. Keys exist — as Kahn goes on — “within a general system and control that system’s variable elements.” Of course, a key is defined differently in transposition ciphers where it is a pattern of shuffling and in cipher machines where it is a disk alignment. In this article, however, we will concentrate on early modern monoalphabetic, homophonic and polyphonic ciphers, so the above shortened approach will be followed. In such ciphers, the cipher alphabet is often complemented with a nomenclature table, a list of code-words, where ciphertext symbols stand for words, common notions, names, geographical unities, bigrams, etc. Nulls, i.e. cipher characters without meaning, are also often added to the system (Láng, 2018). We will call “key” those — usually one page — tables that comprise the cipher

alphabet, the code words and the nulls. It should be added, that in classical cryptology, the sender and the receiver must use the same key while in the post WWII era, this requirement has been changed.

Being able to automatically identify and describe key structure can prove to be useful if we want to conduct an extensive study on the structure of keys, for example from a chronological perspective. An automatic method would be much more effective if we are looking into the structure of keys over a longer period of time or if we are targeting a certain era. This way, we could investigate changes in the structure of keys throughout time, and see in which way the means of encryption have evolved.

In this study, we present a tool that automatically extracts the inner structure of historical original keys, taking into account the symbol system used for encryption, the code structure and the encoded plaintext entities as defined in the key.

Despite the vast advances when it comes to computational decipherment techniques, there currently seems to be a lack of large scale systematic studies that focus on keys. We believe that it is vital to start exploring historical cipher keys by means of computational methods, as well as to develop a way to classify keys. Other than Kahn (1996) there are not many other comprehensive cryptological studies, and even his is mostly targeting ciphers rather than keys. Given the fact that historical cryptology is a relatively young discipline, there can still be differences in notation from one study to another. It is for this reason that we will use Kahn’s work as a basis for the terminology we employ in this paper.

## 2 Key Structure Extraction

First, we need to transform the key image into a text file by transcribing the key, then extract the key structure given the transcribed text file. Be-



low we give a brief overview of the transcription of keys, followed by a description of the key structure extraction.

## 2.1 Transcription

In order to be able to use computational methods for this purpose, we must first establish a transcription standard. This way, we ensure a stable and uniform basis to provide a reliable comparison across keys.

Our proposed method makes use of plain text files (“.txt”) containing the transcription of the original key document. The transcription replicates the original document as closely as possible, both in terms of its structure as well as its content. In large terms, we follow the same guidelines (Megyesi, 2020) as those used in the DECODE database (Megyesi et al., 2019), and expand on them in order to adapt to the specific key structure.

The header of our transcription file consists of metadata which we extract from the DECODE database (Megyesi et al., 2019). The metadata is preceded by a number sign (“#”), followed by the name of the field (e.g. catalog name, language etc.) and the corresponding information: “#CATALOG NAME: BAV\_Barb.lat.6960-17”. Each new type of metadata is transcribed on a new line.

We then proceed to transcribe the content of the original key, following its layout, generally top to bottom, left to right. The transcription of key entries consists of ciphertext – plaintext pairs, where ciphertext represents the symbols used to encode the plaintext message. For those cases where the method of encryption is either homophonic, i.e. a plaintext entity can be encoded by several codes as depicted in Figure 1, or polyphonic substitution, i.e. ciphers in which one ciphertext symbol is used to encode several plaintext elements, as shown in Figure 2, we use the logical operator “|” (“or”) as a way to separate between several ciphertext or plaintext entries, such as illustrated in the following example:

- 72|37 - a → the letter “a” is encoded by either “72” or “37”
- 24 - a|m → the number “24” can either encode the letter “a” or “m”

For the transcription of those symbols that are not part of the extended Latin character set, we follow the DECODE standard of transcribing graphic

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	W	X	Y	Z
27	35	41	11	26	20	1	18	40	47	23	25	13	4	30	38	21	5	16	25	24	5	10	
9	28	38	42	12	27	39	2	19	34	24	36	14	43	32	22	4	15	26	5	17	2	11	
50	70	24	45	13	60	15	17	64	29	24	27	29	4	31	31	25	30	20	43	14			

Figure 1: Example of plaintext alphabet encoded by means of homophonic substitution, extracted from a key from the UK (TNA-SP106, 2020c). Plaintext units on the top row, codes on the bottom row.

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	W	X	Y	Z
As	tn	mo	im	lu	ce	pd	bz	gg															
3	6	5	8	9	7	0	02	04															

Figure 2: Example of plaintext alphabet encoded by means of polyphonic substitution, inspired by a key from the 16th century (ASV-ARM-XLIV-7, 2016). Plaintext units on the top row, codes on the bottom row.

signs according to their name in the Unicode database (Unicode, 2019).

More advanced keys can also have codes that do not map to any kind of lexical unit, and which are commonly referred to as “nulls” (Kahn, 1996). For these cases, we use the tag “<NULL>” as a placeholder for plaintext (e.g. “73 - <NULL>”). Sometimes keys can be incomplete where we find codes without any plaintext elements attached. These empty plaintext elements are not indicated as nulls, they are just missing. To be able to distinguish those from the intended nulls, we transcribe the empty entity as “<EMPTY>”.

Some keys also contain portions of text that are not part of the encoding scheme, such as section headers or notes from the original writer of the document. We refer to such information as “clear-

Fioiti	-	-	-	102
Indulua	-	-	-	103
Gdulua	-	-	-	104
-	-	-	-	105
-	-	-	-	106
-	-	-	-	107

Figure 3: Example of codes mapping to empty plaintext inspired a key from 1692 (ÖstA HHStA Staatzkanzlei Interiora, Kt. 13. Fasc. 20. f. 22., 2020).

text” (Megyesi et al., 2019) and we mark it with its own respective tag, followed by a language ID and the body of text, namely <CLEARTEXT LANGID TEXT>. If the transcriber is able to identify the language of the cleartext, they can use a two-letter ID to mark it in the cleartext tag according to the ISO 639-1 nomenclature (Byrum, 1999). Otherwise, they can replace the language ID with the letters “UN” (i.e. “unknown”).

The transcription file can also contain comments from the person who is transcribing the original document. These can be remarks about the quality of the document, such as bleed-through, ink stains or torn paper, or simply general remarks about the transcription process. An example comment can look as follows: “#COMMENT: torn paper, some symbols were lost”.

Given the transcription file, we can proceed to investigate the components and the structure of the key.

## 2.2 Automatic Extraction

In order to be able to automatically extract statistical information from the transcription file, we write a Python script that analyses the text file and returns a detailed analysis of its content. The script can be run in a terminal window, with the file name as its argument, and then it prints out the results in four main parts.

Other than statistical information, the script also returns the metadata present in the file. This generally includes the original file’s catalog name, the plaintext language (if recognizable), whether or not the transcription is complete or partial, and the transcription time. Optionally, the metadata can also include information about the type of entities in the nomenclature, should they be provided by the transcriber (e.g. names, towns, common words, morphemes etc.). The transcription file can also include additional comments from the transcriber, regarding the transcription process, the state of the original document or its layout, for example. If the script encounters such comments while processing the file, it will not print them as with the metadata, but it will inform the end user of their existence so that they can check the original file in case the comments might be relevant to them.

### 2.2.1 Symbol Set

The first major section of our output focuses on the analysis of ciphertext symbols, beginning with

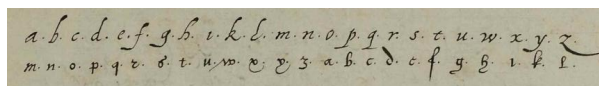


Figure 4: Example of plaintext alphabet encoded by means of simple substitution, extracted from a key from 1596 (TNA-SP106, 2020b). Plaintext units on the top row, codes on the bottom row.

the type of symbols used for encryption. Here we differentiate between 3 major types, namely Latin alphabet, digits, and graphic signs. By “Latin alphabet” we refer to those cases where the individual letters are used for encryption of plaintext, as shown in Figure 4.

In this context, we use “graphic signs” as an umbrella term that refers to any kind of symbol representation that is not part of the Latin alphabet (a-z, A-Z) or digits (0-9). These symbols can be Roman numerals (I-X), zodiac symbols, alchemical signs or any other symbols. When detecting such symbols, the script can further categorize and identify specific sets, as grouped in the Unicode standard (Unicode, 2019). Any other miscellaneous symbols will simply be referred to as “graphic signs” in the output.

#### 2.2.2 Code Structure

The next section of the output looks more in-depth into the internal structure of the ciphertext symbols, which we will refer to as unigraphs, bigraphs, trigraphs and 4+graphs. What counts as unigraphs are usually digits, isolated letters or graphic signs. Digraphs, trigraphs and 4+grams are usually clusters of 2, 3, or 4+ symbols of any of the aforementioned kinds, respectively.

In the cases where digits are included in the ciphertext representation for a key entry, the script will calculate how many of the total ciphertext items are digits. With respect to Figure 5, the output for this section would look as follows:

*unigraphs: 6*

*out of which digits: 4*

Moreover, the encountered ciphertext symbols are mapped to an existing database of symbols so that the user can also get information on how many items have been matched against those already recorded, and how many are new. If any new symbols are found, they will be printed on screen, along with their count.

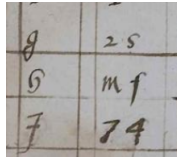


Figure 5: Example of plaintext alphabet encoded by both letters and numbers, extracted from a key from 1569 (TNA-SP106, 2020a). Plaintext units on the left, codes on right.

### 2.2.3 Plaintext Analysis

We then move on to investigate plaintext units. Similarly to ciphertext, these are separated in unigrams, bigrams, trigrams, and 4+grams. We do add, however, a 5<sup>th</sup> category of plaintext, which is that of nulls.

For the most part, the type of plaintext unigrams that we find in keys are either letters or digits, even punctuation in some cases. Bigrams and trigrams are commonly either non-lexical units (e.g. double letters that occur frequently in the language of encryption, such as “ll” or “ee” in English, syllables, morphemes etc.), or short function words (“at”, “for”, “to”, “and” etc.). Under 4+grams we include those units that consist of 4 or more elements, such as longer function words or nomenclature entries, which can consist of names, places, common words. Nomenclatures can also include words that are specific to the lingo used in the topic the key was designed for - army terms in military correspondence, for instance.

Nulls are important to keep track of even though they do not always carry any lexical significance. Nulls can be vital in the process of decipherment, as they might be markers for whitespace or word delimitation, as well as fillers in null ciphers (Kahn, 1996), where only every  $n^{th}$  element carries significance.

In cases where only the code is written without any attached plaintext (<EMPTY>), the program indicates the number of occurrences of such empty placeholders.

### 2.2.4 Code Distribution

Once we described the code and plaintext structure, we can analyze the distribution of ciphertext symbols to plaintext elements from several different perspectives.

#### Cipher type

By analyzing the ratio of plaintext elements to

ciphertext units, the script can differentiate between three different types of encoding, namely simple substitution, homophonic substitution or polyphonic substitution. If the encryption method is consistent throughout the key, we print one of these three possible outputs. There are cases, however, where more than one method is used within the same key. A common case is that the alphabet will be encoded by means of homophonic substitution, while the nomenclature will only use simple substitution. For these instances, we will print all types that are used in the key.

#### Code type

Here we look into the ciphertext symbols only, and determine whether the same type of ngraph was used throughout the whole key or not. Assuming that we are dealing exclusively with bigraphs we can say that the code distribution is fixed. Else, if the key mixes bigraphs and trigraphs, for example, we say that the length is variable.

Establishing the difference between fixed and variable lengths of code is meaningful because, while ciphers with fixed distributions are easier to crack, those who use different ngraph levels can make it more difficult to isolate each individual code from the body of the cipher, and therefore the decryption process becomes more challenging.

#### Codes encoding plaintext

Next, we look into the number of codes that are used in order to encode a certain level of plaintext. If we have “unigrams: 30” as an example output, this would mean that the key uses 30 codes to encipher plaintext unigrams. A potential way to interpret this, assuming that the only unigrams we are dealing with are alphabet letters, would be that the majority of the alphabet entries map to only one code, but that there are some letters that map to several ciphertext symbols. Oftentimes, these letters would be the most frequent ones, which in turn makes the frequency distribution more uniform.

Moreover, the information about the number of unique codes encoding plaintext entities also indicates the complexity of the cipher — the more codes, the harder the decryption of the cipher would be.

#### Ciphertext:plaintext distribution

For the last feature type, we look at the ratio of ciphertext to plaintext units, for four levels of plaintext, namely alphabet, nomenclature, and empty

plaintext elements, being it nulls or placeholders. For each section, we display 4 different possible distributions, which we illustrate below:

- *Alphabet*  
1:1 16  
2:1 5  
3:1 0  
4+:1 3

Keeping in mind that the pairs represent the “ciphertext:plaintext” distribution, in this exact order, this example output tells us that there are 16 instances where one ciphertext symbol maps to one plaintext unigram, 5 instances where a plaintext unit has 2 ciphertext representations, and 3 instances where one plaintext element maps to 4 or more ciphertext symbols.

For polyphonic ciphers, the distribution scheme could be reversed (i.e. 1:1, 1:2, 1:3, 1:4+), to show exactly how many plaintext elements map to one single ciphertext unit.

If the distribution is uniformly 1:1 for a certain category, be it alphabet, nulls, or nomenclature, the script will simply print a message stating this fact, since printing the whole distribution scheme for such cases would be superfluous.

### 2.3 Error Analysis

In order to ensure that the key structure analysis is as accurate as possible, we took the additional step of implementing an error analysis part in our code, that aims to check that the input text follows the same format that we describe in Section 2.1. This way, not only do we make sure that the analysis is accurate, but also that the transcriptions we analyse are all uniform so that we can reliably compare keys among each other.

We differentiate between three major types of user errors that the script can identify, locate, and provide suggestions on how to address them. The most common types of errors that we can encounter are either related to metadata that is not mark correctly or to the fact that the ciphertext and the plaintext are not separated properly. The other type of formatting error that we can detect, and which can be difficult for the user to identify on their own, is the accidental use of tab spacing instead of regular spacing. Even though this might not happen as often, it would negatively impact the way the key statistics are calculated.

### 3 Conclusion & Future Work

In this paper, we presented a tool for the automatic analysis of original historical keys including a common transcription scheme applied to various types of keys. The extracted features include a description of the symbol system used to encode various types of plaintext entities, the code structure, the type of encoded plaintext entities, and the mapping and relation between the code and plaintext entities.

In the future, we are considering expanding to another output format in addition to the already existing plain text one, presented in the Appendix. A viable alternative would be to create a corresponding JSON file, which would in turn allow for easier data manipulation and processing. We plan to add automatic language identification of the plaintext to be included in the automatic structure extraction of keys. Lastly, we would also like to test the tool on a large number of keys of various types, and expand it to allow comparisons across several keys simultaneously.

### Acknowledgments

This work has been supported by the Swedish Research Council, grant 2018-06074: DECRYPT - Decryption of historical manuscripts.

### References

- ASV-ARM-XLIV-7. 2016. Segr.di.Stato-ASV-ARM-XLIV-7-1 Archivio Apostolico Vaticano. DECODE link: <https://cl.lingfil.uu.se/decode/database/record/205>.
- John D. Byrum. 1999. Iso 639-1 and iso 639-2: International standards for language codes. iso 15924: International standard for names of scripts.
- David Kahn. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner.
- Benedek Láng. 2018. *Real Life Cryptology*. Amsterdam University Press.
- Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Ciphers and Keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt19*, Mons, Belgium.
- Beáta Megyesi. 2020. Transcription of historical ciphers and keys. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt20*.

TNA-SP106. 2020a. Reproduced image based on TNA-SP106/1-ElizabethI-f53(0056) National Archives in Kew, UK. DECODE link: <https://cl.lingfil.uu.se/decode/database/record/331>.

TNA-SP106. 2020b. Reproduced image based on TNA-SP106/2-ElizabethI-f58(0069) National Archives in Kew, UK. DECODE link: <https://cl.lingfil.uu.se/decode/database/record/345>.

TNA-SP106. 2020c. Reproduced image based on TNA-SP106/6-CharlesII-(0030-0031) National Archives in Kew, UK. DECODE link: <https://cl.lingfil.uu.se/decode/database/record/428>.

Unicode. 2019. The unicode® standard version 12.0 – core specification.

ÖstA HHStA Staatskanzlei Interiora, Kt. 13. Fasc. 20. f. 22. 2020. Reproduced image from Österreichisches Staatsarchiv, Haus-, Hof- und Staatsarchiv, Staatskanzlei Interiora, Chiffrenschlüssel, Kt. 13. Fasc. 20. 22. DECODE link: <https://cl.lingfil.uu.se/decode/database/record/1199>. The picture has been reproduced by the permission of the Österreichisches Staatsarchiv, Haus-, Hof- und Staatsarchiv, all rights reserved.

## A Appendix - Example Output

An example output is illustrated for a reproduced key from the Österreichisches Staatsarchiv, Haus-, Hof- und Staatsarchiv (ÖstA HHStA Staatskanzlei Interiora, Kt. 13. Fasc. 20. f. 22., 2020), shown in Figure 6.

### Metadata

```
#CATALOG NAME: ÖStA_HHStA_Stk_Int_
Chiffrenschlüssel_fasc_20_22
#LANGUAGE: IT
#STATUS:complete
#TRANSCRIPTION TIME: 4h 10min
```

Cipher symbols:digits

Total number of unique ciphertext symbols:337

```
unigrams:9
  out of which digits: 9
digraphs:90
  out of which digits: 90
trigraphs:238
  out of which digits: 238
4+graphs: 0
```

Total number of ciphertext symbols matched: 337

No new ciphertext symbols were found.

```
Total number of unique plaintext units: 249
  out of which unigrams: 20
  out of which bigrams: 2
  out of which trigrams: 3
  out of which 4+grams: 223
  out of which nulls: 1
  out of which empty: 1
```

Code distribution

Cipher type:mixed  
(homophonic substitution, simple substitution)

Code type:variable length

```
Number of codes encoding plaintext
unigrams:44
bigrams: 2
trigrams: 3
4+grams:223
nulls: 24
empty: 41
```

Distribution according to plaintext type  
(ciphertext:plaintext)

```
1. Alphabet
  1:1      1
  2:1     14
  3:1      5
  4+:1     0
2. Nomenclature
  The nomenclature has a uniform 1:1
  distribution.
3. Nulls
  4+:1      1
4. Empty
  4+:1      1
```

The transcription file contains comments from the transcriber and/or transcriptions of cleartext from the original document which are not included in the statistics above. Please check the transcription file for more details.



A, B, C, D, E, F, G, H, I, L, M, N, O, P, Q, R, S, T, V, W, X, Y, Z																				Nulle				
30	47	45	43	41	38	36	34	32	29	27	25	23	20	18	17	15	13	11		8	del 6 all' 80.			
44	40	44	42	40	37	35	33	31	28	26	24	22	19							7				
45			39					30					21											
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106
108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132
134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184
186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262
264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288
290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314
316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366
368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392
394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418
420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444
446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470
472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496
498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522
524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548
550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626
628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652
654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678
680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704
706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730
732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756
758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782
784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808
810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834
836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860
862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886
888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912
914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938
940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964
966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990
992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016
1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042
1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068
1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094
1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120
1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146
1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172
1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198
1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224
1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250
1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276
1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302
1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328
1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354
1356	1357	1358	1359	1360	1361	1362	1363	1364	1365															

# The Role of Base 10 in the Beale Papers

Viktor Wase

Stockholm, Sweden

viktorwase@gmail.com

## Abstract

The *Beale Papers* is an 1885 pamphlet claiming to contain the location of a huge hidden treasure. The only snag is that the message is encrypted and, as of writing this, unsolved. This study investigates the authenticity of the ciphers by comparing the distribution of the numbers in the ciphers to each other, in different bases. Humans are generally ill-suited to the task of generating random numbers. As such, one might suspect that the behaviour of the distributions in base 10 would be widely different from the other bases if the ciphers were faked. The results of this study strongly indicate that this is the case.

## 1 Background

The *Beale Papers* is a pamphlet from 1885 in which the unnamed writer claims to have come into possession of three ciphers, which are contained in the publication (Ward, 1885). The ciphers are said to be written by a man called Thomas J. Beale - hence the name. The writer goes on to explain how he manages to break cipher number 2. It was a book cipher that could only be read using the correct book as key. The correct key in this case being nothing other than the *Declaration of Independence*. The cracked cipher claims that cipher 1 describes the location of the most valuable treasure of precious metals and gems ever to be found, and cipher 3 lists the name and place of all next-of-kin that have a rightful claim to the riches. The pamphlet costs 50 cents, which is roughly equivalent to 13 dollars today.

One could, at this point, disregard this as a simple ploy to sell a few pamphlets of false hope to the more gullible parts of the population. However before the idea is dismissed out of hand, there are some things that need be told. One of the characters in the pamphlet is named Robert Morriss.

He is an in-keeper who knew Beale from before, and promised to keep these ciphers until Beale returned. If Beale was unable to return in ten years, then an unnamed friend of his would send a letter allowing Morriss to read the ciphers. This letter would arrive no earlier than June 1832. But no letter ever arrived, and neither did Beale. The interesting thing is that the local newspaper *the St. Louis Beacon* has a section where they listed mail couldn't be delivered and was being held at the post office. One of the newspapers from August of 1832, which would fit the time line of the pamphlet's story, such a letter is held for Robert Morriss (Chan, 2008)! It should be noted how unusual the spelling of his last name is, as there were no Morriss listed in the 1840 census of St. Louis, at all.

Another piece of information supporting the credibility of the Beale Papers appeared on Boxing Day 2001. A classified study for an NSA conference in 1970 was released to the public under the Freedom of Information act (Hammer, 1970). It was written by Dr. Carl Hammer, director of computer science at Univac Federal Government Marketing. His doctorate was in mathematical statistics and probability theory. The study investigated some *signatures* of the ciphers as well as of other similar ciphers, and found that they were both data and process dependent. Meaning that one can gain insights about the plain-text as well as the process of enciphering by studying these signatures. This line of inquiry gave such strong evidence for the credibility of the Beale Papers that the last line of the abstract is: "[the signatures] indicate also very strongly that Mr. Beale's cyphers are for real and that it is merely a matter of time before someone finds the correct source document and locates the right vault in the Commonwealth of Virginia." One could easily become rather conspiratorial by thinking too long and hard about why the US government kept this report secret for

over three decades.

It should be noted that there is plenty of evidence against their authenticity as well. The most concise comes from 1927 when an author named Kendell F. Crossen asked how the next of kin can possibly be listed in cipher 3 (Kruh, 1982). In the pamphlet it says that there were 30 men in Beale's company, and the cipher is only 618 signs long. That does not leave many letters per person.

Another common argument against it is that it would actually be surprisingly hard to decipher the second text with the Declaration of Independence. First of all, there are several versions of the declaration and unless you have the correct one it will be hard to decipher it. However, even if you do have the correct version it will be hard to decipher it for there are a few peculiarities in the cipher (Mateer, 2013):

- a word was miscounted around position 630, as well as 670.
- 480 was used to represent two different words.
- "self-evident" was counted as one word, instead of two.
- For some reason ten words were skipped around position 480.

The first three might have been mistakes you could realise and fix as you were deciphering, but the last one is harder to explain away.

## 2 Related Work

The Beale Papers have been studied from a statistical perspective before. Partly by Hammer in 1970 (Hammer, 1970), as mentioned above, who concluded that the ciphers might just be real.

A contrary position was taken by L. Kruh, who considered the possibility that author of the pamphlet also wrote the letters from Beale that are included in the pamphlet (Kruh, 1988). The language use of these texts were compared. Statistics such as the distribution of words per sentence, the distribution of verb tenses, as well as the distribution of word classes were considered. These comparisons show that the texts are very similar, and might be written by the same author.

## 3 The Importance of Base Ten

Humans are really rather terrible at being random. In a meta-study on the subject from 1972 only one

out of the 15 studies were positive towards human's ability to create random sequences (Wagenaar, 1972). In that study the participants were asked to create a sequence of O's and X's using two stamps. Usually, humans will switch symbol more often than what would happen in a true 50/50 random sequence. The author of the meta-study argued that it might in fact be the boredom and laziness of the participants that made them use the same stamp repeatedly - they simply wanted it done with.

On a not entirely different note: if a person was to write a long list of random numbers, then they would probably do so in base 10. This might not seem relevant, but number might be perceived to have certain properties depending on which base they are written in. For example, the numbers  $101010_2$  and 42 are not obviously identical, and one of them might even be perceived as having a *symmetry* the other does not. Or how about 12345 and 17836<sub>9</sub> - the first one does seem less random and more ordered than the second one. Which is utter nonsense, of course - they are the same number. But it goes to show some of the limitations of the human perception of numbers.

This would mean that three ciphers, all encoded with the same method, should give similar results under statistical investigation. If two of the ciphers were faked, on the other hand, then such statistical investigations might give similar results in base 10, where humans have intuition, but probably not in other bases.

### 3.1 The Last Digit

Let's start at the end, with the last digit in each number. The numbers used in the Beale ciphers span a rather large range: from 1 to 2906. With such a large range, one can imagine that the last number is evenly distributed over all digits - that it is simply noise compared to the larger magnitudes.

This hypothesis can be tested using a discrete Kolmogorov-Smirnov (KS) test. This test is used to estimate the likelihood of two sets of samples being drawn from different distribution. The KS statistic is defined as the maximum difference of the cumulative distribution functions of the sets of samples. That is

$$\max_x |f_1(x) - f_2(x)|$$

where  $f_1$  and  $f_2$  are the cumulative distribution functions, and  $x$  is a real number. The test is applied to the given distribution (uniform) and the

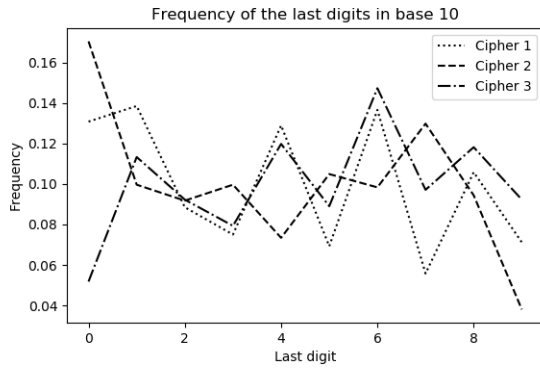


Figure 1: Frequencies of the the last digit of each number, for all ciphers.

given cipher. One starts by generating  $n$  random points from the distribution, where  $n$  is the number of data points in the cipher. If the KS statistic of the new points and the uniform distribution is larger than the original statistic, then one makes a note of it. This step is repeated multiple times, and the  $p$ -value is the portion of iterations with a larger KS difference than the original KS difference.

Using 1,000,000 iterations per cipher one can conclude that the hypothesis is wrong. The distribution of the last numbers are not uniform, for any of the ciphers. The  $p$ -values for the three ciphers are: 0.4%, 0.02% and 0.4%. The distributions can be seen in figure 1. The first and third ciphers oscillate a bit, in a way which means that the numbers are more likely to be even than odd.

This is where things are starting to get interesting. The same experiment is repeated but in different bases. Only bases that are relatively prime to 10 are considered, to avoid that the effects of base 10 spill over. For example, the even-odd disparity will show up in all even-numbered bases. See figure 2 for the result.

The  $p$ -values of cipher 2 are consistently small, meaning that the numbers are not uniformly distributed and there is nothing inherently specific about base 10. This cannot be said for ciphers 1 and 3. Their  $p$ -values are really rather large for all bases except 10, and its neighbours. This strongly indicates that the digits are uniformly distributed, with the exception of when they are given in base 10.

### 3.2 Benford's Law

Another hypothesis one might consider is that perhaps the leading digit follows Benford's Law (Benford, 1938). It's not a *law* in any sense of

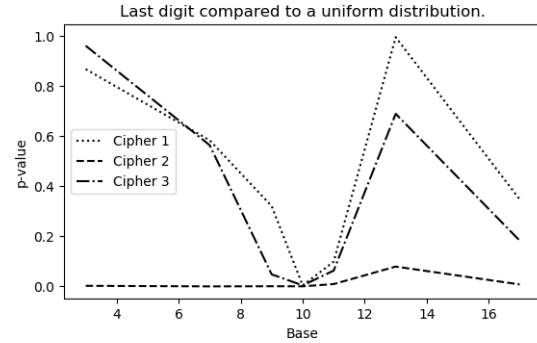


Figure 2: Results of the discrete Kolmogorov-Smirnov test, comparing the distribution of the last digit of all numbers to a uniform distribution.

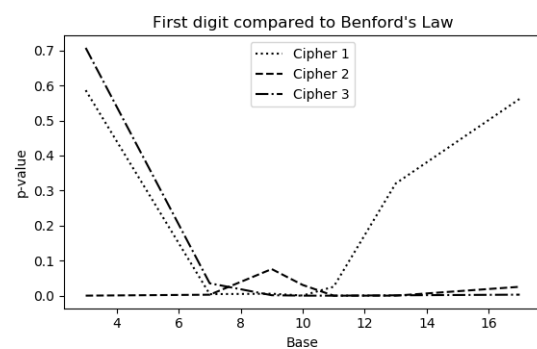


Figure 3: Results of the discrete Kolmogorov-Smirnov test, comparing the distribution of the first digit of all numbers to Benford's Law.

the word, nor was it created by Benford. Nomenclature issues aside, the law states that if a distribution spans several multitudes, then the distribution of the first digit might follow the distribution:  $\log 1 + \frac{1}{d}$  where  $d$  is the leading digit (1-9). It is often used for fraud detection (Nigrini, 1999).

This hypothesis was tested using the KS test described above, with 100,000 iterations. However, it turned out to be false this time too. The  $p$ -values are 0.043%, 3.1% and 0.004% for the three ciphers. But, once again, what holds true in base 10 might not hold true in other bases. A KS test was performed, same as last, but looking at the leading digit as compared with Benford's Law. The results can be seen in figure 3.

Cipher 1 takes a clear dive at base 7, and doesn't start growing again until after base 10. Cipher 3 never starts growing again. The valley around 10 is wider than when looking at the final digit, but that is to be expected since the last digit is more sensitive to a small base change than the first digit. As of why the third cipher never grows, I

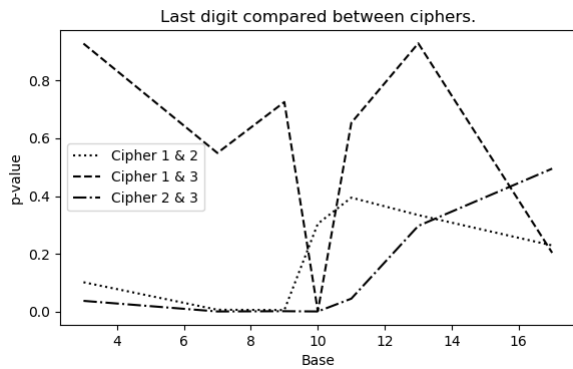


Figure 4: Results of the discrete two-sample Kolmogorov-Smirnov test, based on the last digit in each number.

find the most likely explanation to be that it spans the range [1, 975] and cipher 1 spans the range [1, 2906] and Benford's Law works better when several orders of magnitudes are spanned.

### 3.3 Non-assumptive Comparisons

Instead of comparing the ciphers against analytical distribution one can compare the ciphers against each other. Using a discrete two-sample KS test each pair of ciphers was tested to see if they were drawn from different distributions - for the first and the last digits, separately. The general idea of the test is simple, one calculates the KS difference between the ciphers. Then the samples are randomly shuffled between the ciphers, and a new KS difference is calculated using the shuffled samples. The shuffling is repeated 100,000 times, and each time the shuffled KS value is larger than the original value, one makes a note. The idea is that if the ciphers are different, then shuffling the samples between them should decrease the difference, but if they are created from the same distribution then the difference between them shouldn't change by shuffling the samples.

The last digits are compared in figure 4. Base 10 shines like a beacon, but not in the expected way. The comparison of cipher 1 and 3 has consistently large  $p$ -values, with the exception of a large valley at base 10. The values of comparison of 2 and 3 are low for all bases at 10 or lower. However, contrary to the expectations, the comparison of 1 and 2 does not show a valley around 10. Figure 5 shows the comparison of the first digits, and it shows a distinct valley at base 10, although not nearly as pronounced as in the earlier experiments.

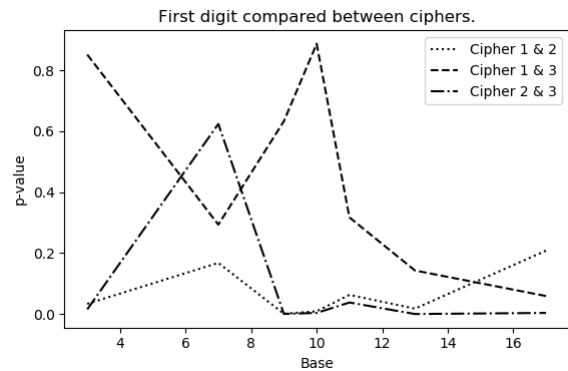


Figure 5: Results of the discrete two-sample Kolmogorov-Smirnov test, based on the first digit in each number.

## 4 Conclusion

One can, with a greater certainty than before, declare the Beale Papers to be frauds. The statistical tests strongly indicate that cipher 1 and 3 are inherently different when viewed through the lens of base 10 as compared with the other bases. I would consider this, in and of itself, to be damning evidence. But the fact that we have access to a real cipher, supposedly written by the same author, which does not show the same behaviour in the slightest, makes the argument even stronger.

There might possibly be different explanations for this behaviour. Perhaps the key to the book cipher has ten words on every line, or a multiple of ten lines per page. Perhaps the creator of the ciphers predicted this line of attack and deliberately made sure that we would get this result to throw us off their scent. Or perhaps they were just human with all our faults and limitations and learned to count, write and talk numbers in base 10.

## 5 Acknowledgements

I would like to express my very great appreciation to Johannes Wennberg for many and long fruitful discussions, as well as novel and interesting ideas. Thank you!

## References

- Frank Benford. 1938. The law of anomalous number. *Proceedings of the American Philosophical Society*, 78(4):551–572.
- Wayne S. Chan. 2008. Key enclosed: Examining the evidence for the missing key letter of the beale cipher. *Cryptologia*, 32(1):33–36.



- Carl Hammer. 1970. Signature simulation and certain cryptographic codes. *Third Annual Simulation Symposium*.
- Louis Kruh. 1982. A basic probe of the beale cipher as a bamboozlement. *Cryptologia*, 6(4):378–382.
- Louis Kruh. 1988. The beale cipher as a bamboozlement - part ii. *Cryptologia*, 12(4):241–246.
- Todd D. Mateer. 2013. Cryptanalysis of beale cipher number two. *Cryptologia*, 37(3):215–232.
- Mark J. Nigrini. 1999. I’ve got your number. *Journal of Accountancy*, 187(5):79 – 84.
- W. A. Wagenaar. 1972. Generation of random sequences by human subject: A critical survey of literature. *Psychological Bulletin*, 77(1):65–72.
- James B. Ward. 1885. *The Beale Papers Containing Authentic Statements Regarding the Treasure Buried in 1819 and 1821 near Bufords in Bedford County, Virginia*.

Published by

NEALT Proceedings Series 44

Linköping University Electronic Press, Sweden

Linköping Electronic Conference Proceedings, No. 171

ISSN: 1650-3686

eISSN: 1650-3740

ISBN: 978-91-7929-827-2

URL: <http://www.ep.liu.se/ecp/contents.asp?issue=171>