Introduction to the History of Cryptology

Software Security

Steffen Helke

Chair of Software Engineering

18th November 2018

Brandenburgische
Technische Universität
Cottbus - Senftenberg

Objectives of today's lecture

- ➜ Repetition: How to solve the *refactoring task* from the last exercises?
- ➜ History: Getting to know *first techniques* for transmitting secret messages
- ➜ Understanding the *principle of frequency analyses*, as well as mono- and polyalphabetical substitution
- ➜ Understanding the structure and functionality of *cipher machines* using the Enigma as an example

Repetition: JIF – Java + Information Flow

What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?

JiF Exercises

2 Task: Refactoring *Extract Method*

Objective

- Getting familiar with *begin labels* of methods

Tasks

- Consider for the information flow only the start labels of the methods
- Generate the begin labels in two different ways: First in the most restrictive variant and then in the most general one, such that in the latter case the method could be called in other contexts too

2 Task: Refactoring Extract Method

- In the following program, redundant code is to be extracted into an independent method ➜ Refactoring *Extract Method*
- Which signature (*begin label*) should be used to ensure that the security policy is preserved after restructuring?

```
class Refactoring {
   int {Bob -> Alice,Bob,Steffen} a = 0;
   int {Bob -> Bob} b = 1;
   int {Bob -> Alice,Bob} c;

   public void f {} () {
    if (a == 0) {
         b = 4;
         c = 3;
    }
    if (c == 1) {
         b = 4;
         c = 3;
    }}}
```

What is the meaning of begin and end-labels in JiF?
How is it possible to support JiF-refactorings?

Approaches to solve the refactoring task

➜ There are two alternatives to construct the begin label (bl)

1. Analyze the *context of the method*, i.e. use all *pc labels* at which the method is to be called and afterwards built the *join* over all these pc labels (restrictive approach)

$$bl = sc(a) \sqcup sc(c)$$
$$= \{\text{Bob} -> \text{Alice, Bob, Steffen} \;;\; \text{Bob} -> \text{Alice, Bob}\}$$
$$= \{\text{Bob} -> \text{Alice, Bob}\}$$

2. Analyze the *body of the method*, i.e. use the security labels of the value assignments inside of the method and connect these labels by the *meet* operator (most general approach)

$$bl = sc(b) \sqcap sc(c)$$
$$= \{\text{Bob} -> \text{Bob meet Bob} -> \text{Alice, Bob}\}$$
$$= \{\text{Bob} -> \text{Alice, Bob}\}$$

What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?

Note: In this example we obtain the same result, which is generally not the case

How to calculate join $\sqcup$ and meet $\sqcap$?

Calculating Readers

What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?

$$readers(c \sqcup d) \mathrel{\widehat{=}} readers(c) \cap readers(d)$$
$$readers(c \sqcap d) \mathrel{\widehat{=}} readers(c) \cup readers(d)$$

Calculating Writers

$$writers(c \sqcup d) \mathrel{\widehat{=}} writers(c) \cup writers(d)$$
$$writers(c \sqcap d) \mathrel{\widehat{=}} writers(c) \cap writers(d)$$

Note: $\sqcup$ (*join*) and $\sqcap$ (*meet*) are represented in JIF for two given security labels a and b by $\{a\,;\ b\}$ and $\{a\ meet\ b\}$

Definition Join Operator (Background)

Question for Understanding:

What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?

➜ Why is the join operator for two security labels defined with the meet operator on the sets of read permissions?

$$readers(p, c \sqcup d) \mathrel{\widehat{=}} readers(p, c) \cap readers(p, d)$$

Answer:

- According to Leibnitz, the following characteristic must be fulfilled for a lattice with partial order

$$c \sqsubseteq d \quad \Leftrightarrow \quad c \sqcup d = d \quad \Leftrightarrow \quad c \sqcap d = c$$

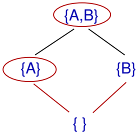
- In contrast to the set, the weakest element of security labels is not described by the empty set, but by the maximum set (*reading is allowed for all*)
- Consequently, the order is reversed and the meaning of join and meet operator is reversed as well

Example: Definition Join Operator

What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?

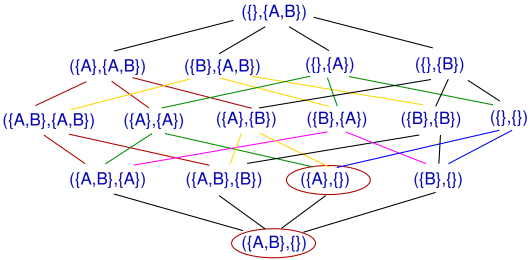
➜ Lattice for traditional sets

$\{A\} \subseteq \{A, B\} \Leftrightarrow \{A\} \cup \{A, B\} = \{A, B\}$



➜ Lattice for for JiF security labels

$(\{A, B\}, \{\}) \sqsubseteq (\{A\}, \{\}) \Leftrightarrow (\{A, B\}, \{\}) \sqcup (\{A\}, \{\}) = (\{A\}, \{\})$



2 Solution: Refactoring Extract Method

What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?

```
class Refactoring {
  int {Bob -> Alice,Bob,Steffen} a = 0;
  int {Bob -> Bob} b = 1;
  int {Bob -> Alice,Bob} c;

  public void f {} () {
    if (a == 0) {
      setBC();
    }
    if (c == 1) {
      setBC();
    }
  }
  public void setBC {Bob -> Alice,Bob} () {
    b = 4;
    c = 3;
  }
}
```

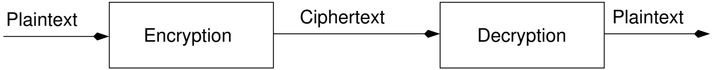
Introduction to the History of Cryptology

Basic Terms

Cryptology ...

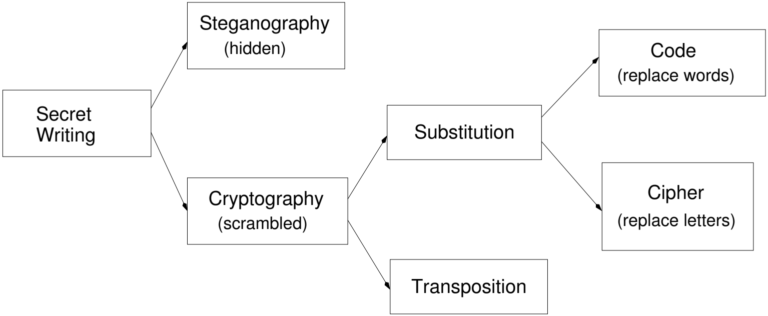
is the wisdom of ciphers

1. **Cryptography ...**
   describes how a secret code works,
   e.g. encryption and decryption algorithm
2. **Cryptanalysis ...**
   defines how to decrypt or analyze a given cipher, e.g. without knowing the key



Secrecy of Messages in History

Main Branches



Source: *Simon Singh, The Code Book, How to make it, break it, hack it, or crack it.* Delacorte Press, 2001.

Examples for Steganography

Historical tradition of Herodotus

1. Shave the head of your messenger
2. Write the message on his scalp
3. Wait for the hair regrow

Ancient China

1. Write your messages on a fine silk
2. Silk is scrunched into a tiny ball and covered in wax
3. Courier has to swallow the wax ball

Examples for Steganography

Italy, 15th century

1. Mixing of alum powder (alum salt) into vinegar
2. Write your message on a hard-boiled egg (with this liquid)
3. Liquid penetrates through the porous eggshell and passes the message to the protein

Invisible Ink, first century AD

1. Write your message on a paper with a transparent organic liquid (e.g. milk of the tithymalus plant or urine)
2. Colouring of the message by heating the paper over a candle

Cryptography using Transposition

Idea

- Rearranging letters results in an *anagram*
- Insecure for short messages

Beipiel 1: Insecure Random Transposition

"cow"

6 variants (cow cwo ocw owc wco woc)

Beipiel 2: Relative Secure Random Transposition

"For example, consider this short setence"

50 000 000 000 000 000 000 000 000 000 000,

➜ i.e. 50 Quintillion (traditional British)

Example for a Systematic Transposition

Idea of the "Rail Fence" Transposition

1. Write your message with alternating letters on separate upper and lower lines
2. Attache the lower row of letters to the upper row of letters in sequential order

Example: Decomposition using two rows

THIS IS A SECRET PIECE OF TEXT THAT CANNOT BE DECODED IMMEDIATELY

↓

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

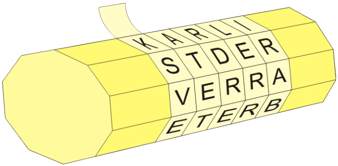
↓

TIIAERTICOTXTACNOBDCDDMEITL HSSSCEPEEFETHTANTEEOEIMDAEY

First Military Cryptographic Device

Scytale (wooden staff), fifth century B.C.

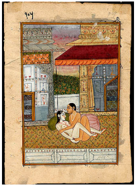
1. Wound a strip of leather or parchment around the scytale
2. Write the message along the length of the scytale
3. Unwind the strip, which appears to carry a list of meaningless letters
4. Use a scytale with the same diameter for decryption



Examples for Monoalphabetic Substitution

Kama Sutra

- First documented use of substitution (4th century BC)



- The book describes a series of self-study arts for modern women
  - Cooking
  - Dressing
  - Massage, love play, etc.
  - But also *encryption techniques for keeping affairs secret*
- Procedure
  1. Random pairing of the letters of the alphabet
  2. Replace the opposing letters in your message

Examples for Monoalphabetic Substitution

Caesar's Substitution

Encryption method for military purposes used by Julius Caesar in the Gallic War

- **Simple Variant**
  Letter substitution of the Roman alphabet by the Greek alphabet

- **Caesar Shift Cipher**
  Replace each letter in the message with the letter that is three places further down the alphabet

Example Caesar Shift Cipher with 3

Coding Rule

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
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

Example: Secret Message

F	A	I	L	U	R	E	R	A	T	E	I	S	T	H	R	E	E
I	D	L	O	X	U	H	U	D	W	H	L	V	W	K	U	H	H

Use of Keywords

Procedure

1. Choose a keyword (e.g. Julius Caesar)
2. Delete all repeating letters
3. Use the keyword only for the beginning and continue with the alphabet after the last letter
4. Be careful not to use repetitive letters in the encoding alphabet

Coding Rule

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

Evaluation of Monoalphabetic Substitution

How to decrypt a ciphertext encrypted by a monoalphabetic substitution using a frequency analysis?

Strategies

1. Only 25 distinct cipher alphabets with normal Caesar Shift
   $\Rightarrow$ Testing of all variations is possible, also by hand
2. 400 000 000 000 000 000 000 000 000 cipher alphabets, if we allow an arbitrary rearrangement of the plaintext alphabet
3. If you use keywords, the number of possible cipher alphabets will be reduced again, but is still quite large

Attacks

- At that time it was not possible to try out all the variants for (**2.**) and (**3.**)
- But *frequency analysis* could help

Cryptanalysis by Frequency Analysis

~~How to decrypt a ciphertext encrypted by a monoalphabetic~~ substitution using a frequency analysis?

Cryptanalysis

- Science of *decryption without knowing the key*
- Technique was first described in the 9th century by an Arab philosopher

Procedure

1. Determine the occurrence frequency for each letter of the plaintext alphabet
2. Determine the occurrence frequency for each letter of the intercepted ciphertext
3. Decode the ciphertext by comparing the two frequency analyses

Frequency of Letters in German Documents

letter	frequency in %	letter	frequency in %
a	6.51	n	9.78
b	1.89	o	2.51
c	3.06	p	0.79
d	5.08	q	0.02
e	17.40	r	7.00
f	1.66	s	7.27
g	3.01	t	6.15
h	4.76	u	4.35
i	7.55	v	0.67
j	0.27	w	1.89
k	1.21	x	0.03
l	3.44	y	0.04
m	2.53	z	1.13

Results come from A. Beutelspacher, Kryptologie, Braunschweig 1993

Frequency Analyses

Drawbacks

- Frequency table only gives average values

> **Example:** From Zanzibar to Zambia and Zaire, ozone zones make zebras run zany zigzags

- 1969: French writer Georges Perec La Disparition writes a 200-page book *without* using the letter *e*
- There is a German translation of Eugen Helmle also without the letter *e* to use

Improvements

- Analyzing the frequencies of bigrams or trigrams
- Bigrams with *e* are *er*, *en* und *ei* (German alphabet)
- Frequency analysis also possible for complete words

Decryption of a Ciphertext (1)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

1. Frequency analysis of letters

   Results:

Decryption of a Ciphertext (1)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

1. Frequency analysis of letters
   Results: $S(68)$

Decryption of a Ciphertext (1)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

1. Frequency analysis of letters
   Results: $S(68)$, $R(32)$

Decryption of a Ciphertext (1)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

1. Frequency analysis of letters
   Results: $S(68)$, $R(32)$, $P(30)$

Decryption of a Ciphertext (1)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

1. Frequency analysis of letters
   Results: $S(68)$, $R(32)$, $P(30)$, $Y(27)$

Decryption of a Ciphertext (1)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

1. Frequency analysis of letters
   Results: $S(68)$, $R(32)$, $P(30)$, $Y(27)$, $Z(24)$
   Conclusion: $S = \phi(e)$, $\forall\, b : \{R, P, Y, Z\} \bullet \phi(b) \in \{n, i, s, r\}$[1]
2. Frequency analysis of bigrams
   Results: SR, SP, SY
   Conclusion: $S = \phi(e)$ seems to be correct

[1] $\phi(x)$ indicates the encryption of the letter x

Decryption of a Ciphertext (2)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

3. Frequency analysis of trigrams
   Results: *SPR* is the most frequent (corresponds to *ein*)
   Conlusion: $P = \phi(i)$ and $R = \phi(n)$
4. Searching for frequent words
   Results: 5 times *GPS* (corresponds to *die*)
   Conlusion: $G = \phi(d)$

Decryption of a Ciphertext (3)

PR ISRSQ YSPUD SYOCREBS GPS NFRZB GSY NCYBVEYCWDPS

SPRS ZVOUDS HVOONVQQSRDSPB, GCZZ GPS NCYBS SPRSY

SPRMPESR WYVHPRM GSR YCFQ SPRSY ECRMSR ZBCGB

SPRRCDQ FRG GPS NCYBS GSZ YSPUDZ GSR SPRSY WYVHPRM.

QPB GSY MSPB ASTYPSGPEBSR GPSZS FSASYQCSZZPE EYVZZSR

NCYBSR RPUDB OCSRESY, FRG QSR SYZBSOOBS SPRS NCYBS

GSZ YSPUDZ, GPS ESRCF GPS EYVSZZS GSZ YSPUDZ DCBBS.

AVESZ, HVR GSY ZBYSRES GSY JPZZSRZUDCTB

Procedure

5. <u>Searching for frequent words</u>
   Results: 4 times *GSY* and 3 times *GSZ* (corresponds to *der* and *des*) Conclusion: $Y = \phi(r)$ und $Z = \phi(s)$
6. <u>Inserting the found Letters and Solving the Puzzle</u>
   Results: *reiUD* corresponds to *reich*, *dCss* corresponds to *dass* Conclusion: $U = \phi(c)$, $D = \phi(h)$ und $C = \phi(a)$

Decryption of a Ciphertext (4)

Coding Rule

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
C	-	U	G	S	T	E	D	P	-	-	-	-	-	-	-	-	Y	Z	B	-	-	-	-	-	-

Procedure

7. Searching for the code word
   Conclusion: *C. Auguste Dupin* Master detective in the story by Edgar Allan Poe: *The Murders in the Rue Morgue*

Decryption of a Ciphertext (4)

Coding Rule

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

Procedure

7. Searching for the code word
   Conclusion: *C. Auguste Dupin* Master detective in the story by Edgar Allan Poe: *The Murders in the Rue Morgue*

Decryption of a Ciphertext (5)

In jenem Reich erlangte die Kunst der Kartographie eine solche Vollkommenheit, dass die Karte einer einzigen Provinz den Raum einer ganzen Stadt einnahm und die Karte des Reichs den einer Provinz. Mit der Zeit befriedigten diese uebermaessig grossen Karten nicht laenger, und man erstellte eine Karte des Reichs, die genau die Groesse des Reichs hatte.

(Jorge Luis) Borges, Von der Strenge der Wissenschaft

Improved Monoalphabetic Substitution

Strategies

1. Introduction of additionally letters, like nulls, symbols or other meaningless letters to modify the frequency
2. Injection of misspelling
3. Introduction of code words (substitution at word and sentence level)

Drawbacks

- Dictionaries of code words required for encryption and decryption
- When using only a few code words, meaning can be guessed from context easily

➜ Conlusion: No effective protection against frequency analysis

Polyalphabetic Substitution

Polyalphabetic Substitution using Vigenére Cipher

Do you know an example for a polyalphabetic substitution?

- Goal: Development of a *stronger system than monoalphabetic encryption* (end of the 16th century)
- Preliminary work by Leon Battista Alberti (15th century)
- Main idea: Use of several cipher alphabets
- Advantage: Representation of identical letters of the plaintext by different letters of the ciphertext
  - ➜ Covering the frequency of characters in plaintext
- Vigenére Cipher was named after a French diplomat
  - ➜ *Blaise de Vigenére* (*1523)
- Historically, the development of polyalphabetic substitution was an *important milestone*

Encryption using the Vigenére Square

Klar	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
1	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
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
11	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
12	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
13	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
14	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
15	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
16	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
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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
25	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
26	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

How to apply the Vigenére Cipher?

Remarks

- Vigenére square allows the *encryption with 26 different ciphertext alphabets*
- A code word must be negotiated

Procedure

1. Write the code word several times over the plaintext
2. Find the row of the table where the first entry matches the letter of the code word
3. Determine the encoding letter for the corresponding plaintext letter using this row

Homophonic Substitution Procedures

Motivation

- Vigenére encryption was considered too complicated in practice
- Middle way is homophone substitution procedure

Idea

- Represent each letter of the plaintext with several letters in the ciphertext
- Define the number of possible coding variants for a given plaintext letter according to the frequency of its occurrence

Example: Letter R

- Letter *R* is represented by 7 different cipher letters, because its average occurrence frequency is 7% in plaintext

Comparison and Evaluation

What are the differences between Vigen´ere cipher and Homophonic substitution?

Homophonic Substitution

- Since only a fixed ciphertext alphabet is used, it is a *monoalphabetic substitution*
- A plain text letter is represented by several ciphertext letters (usually numbers), but not the other way round

Vigenére Cipher

- Polyalphabetic substitution
- A ciphertext letter can also represent several plaintext letters

Conclusions

- Vigenere cipher better than Homophone substitution method
- From theoretical point of view, *both procedures are still insecure*!

Weaknesses of Vigenére Cipher

Attacks

- Vigenére cipher was broken by Charles Babbage around 1854
- Unfortunately not published by him!

Procedure

1. Identify recurring patterns in the cipher by analyses
2. Derive the key length from the analysis results
3. Conduct as many frequency analyses as the code word is long

Example: Plaintext constructed from just one letter

L	I	G	H	T	L	I	G	H	T
e	e	e	e	e	e	e	e	e	e
P	M	K	K	X	P	M	K	K	X

Example: How to find the code word length?

WUBEFIQLZURMVOFEHMYMWT

IXCGTMPIFKRZUPMVOIRQMM

WOZMPULMBNYVQQQMVMVJLE

YMHFEFNZPSDLPPSDLPEVQM

WCXYMDAVQEEFIQCAYTQOWC

XYMWMSEMEFCFWYEYQETRLI

QYCGMTWCWFBSMYFPLRXTQY

EEXMRULUKSGWFPTLRQAERL

UVPMVYQYCXTWFQLMTELSFJ

PQEHMOZCIWCIWFPZSLMAEZ

IQVLQMZVPPXAWCSMZMORVG

VVQSZETRLQZPBJAZVQIYXE

WWOICCGDWHQMMVOWSGNTJP

FPPAYBIYBJUTWRLQKLLLMD

PYVACDCFQNZPIFPPKSDVPT

IDGXMQQVEBMQALKEZMGCVK

UZKIZBZLIUAMMVZ

Some patterns appear several times:

Example: How to find the code word length?

WUBEFIQLZURMVOFEHMYMWT

IXCGTMPIFKRZUPMVOIRQMM

WOZMPULMBNYVQQQMVMVJLE

YMHFEFNZPSDLPPSDLPEVQM

WCXYMDAVQEEFIQCAYTQOWC

XYMWMSEMEFCFWYEYQETRLI

QYCGMTWCWFBSMYFPLRXTQY

EEXMRULUKSGWFPTLRQAERL

UVPMVYQYCXTWFQLMTELSFJ

PQEHMOZCIWCIWFPZSLMAEZ

IQVLQMZVPPXAWCSMZMORVG

VVQSZETRLQZPBJAZVQIYXE

WWOICCGDWHQMMVOWSGNTJP

FPPAYBIYBJUTWRLQKLLLMD

PYVACDCFQNZPIFPPKSDVPT

IDGXMQQVEBMQALKEZMGCVK

UZKIZBZLIUAMMVZ

Some patterns appear several times:

- E-F-I-Q at a distance of 95 characters

Example: How to find the code word length?

WUBEFIQLZURMVOFEHMYMWT

IXCGTMPIFKRZUPMVOIRQMM

WOZMPULMBNYVQQQMVMVJLE

YMHFEFNZPSDLPPSDLPEVQM

WCXYMDAVQEEFIQCAYTQOWC

XYMWMSEMEFCFWYEYQETRLI

QYCGMTWCWFBSMYFPLRXTQY

EEXMRULUKSGWFPTLRQAERL

UVPMVYQYCXTWFQLMTELSFJ

PQEHMOZCIWCIWFPZSLMAEZ

IQVLQMZVPPXAWCSMZMORVG

VVQSZETRLQZPBJAZVQIYXE

WWOICCGDWHQMMVOWSGNTJP

FPPAYBIYBJUTWRLQKLLLMD

PYVACDCFQNZPIFPPKSDVPT

IDGXMQQVEBMQALKEZMGCVK

UZKIZBZLIUAMMVZ

Some patterns appear several times:

- E-F-I-Q at a distance of 95 characters
- P-S-D-L-P at a distance of 5 characters

Example: How to find the code word length?

WUBEFIQLZURMVOFEHMYMWT

IXCGTMPIFKRZUPMVOIRQMM

WOZMPULMBNYVQQQMVMVJLE

YMHFEFNZPSDLPPSDLPEVQM

WCXYMDAVQEEFIQCAYTQOWC

XYMWMSEMEFCFWYEYQETRLI

QYCGMTWCWFBSMYFPLRXTQY

EEXMRULUKSGWFPTLRQAERL

UVPMVYQYCXTWFQLMTELSFJ

PQEHMOZCIWCIWFPZSLMAEZ

IQVLQMZVPPXAWCSMZMORVG

VVQSZETRLQZPBJAZVQIYXE

WWOICCGDWHQMMVOWSGNTJP

FPPAYBIYBJUTWRLQKLLLMD

PYVACDCFQNZPIFPPKSDVPT

IDGXMQQVEBMQALKEZMGCVK

UZKIZBZLIUAMMVZ

Some patterns appear several times:

- E-F-I-Q at a distance of 95 characters
- P-S-D-L-P at a distance of 5 characters
- W-C-X-Y-M at a distance of 20 characters

Example: How to find the code word length?

WUBEFIQLZURMVOFEHMYMWT

IXCGTMPIFKRZUPMVOIRQMM

WOZMPULMBNYVQQQMVMVJLE

YMHFEFNZPSDLPPSDLPEVQM

WCXYMDAVQEEFIQCAYTQOWC

XYMWMSEMEFCFWYEYQETRLI

QYCGMTWCWFBSMYFPLRXTQY

EEXMRULUKSGWFPTLRQAERL

UVPMVYQYCXTWFQLMTELSFJ

PQEHMOZCIWCIWFPZSLMAEZ

IQVLQMZVPPXAWCSMZMORVG

VVQSZETRLQZPBJAZVQIYXE

WWOICCGDWHQMMVOWSGNTJP

FPPAYBIYBJUTWRLQKLLLMD

PYVACDCFQNZPIFPPKSDVPT

IDGXMQQVEBMQALKEZMGCVK

UZKIZBZLIUAMMVZ

Some patterns appear several times:

- E-F-I-Q at a distance of 95 characters
- P-S-D-L-P at a distance of 5 characters
- W-C-X-Y-M at a distance of 20 characters
- E-T-R-L at a distance of 120 characters

Example: How to find the code word length?

Procedure

1. Divide the distances of the found patterns by possible key lengths
2. Check by which key lengths the distances of all found patterns are divisible
3. Test all key lengths found in this way for plausibility

Pattern	Distance	Possible key length (distance divisor)																		
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
E-F-I-Q	95				x														x	
P-S-D-L-P	5				x															
W-C-X-Y-M	20	x		x	x					x										x
E-T-R-L	120	x	x	x	x	x		x		x		x			x					x

➜ Key length for this example is 5, i.e. 5 frequency analyses would have to be carried out for decryption

One-Time Pad (also called Vernam Cipher)

How works the encryption using a Vernam cipher? Why is this cipher information-theoretically secure?

General Remarks

- *Information-theoretically secure*
- Developed in 1917 by Major Joseph Mauborgne.
- At the American Army Cryptographic Research Department
- Security based on randomness of the key

Procedure

1. Select a key at random and make sure that it is at least as long as the plaintext
2. Write the key over the plaintext
3. Add (for decrypting subtract) modulo 26, for binary representation use the XOR operation

Development of Cipher Machines

Development of Cipher Machines

… The first cryptographic device was developed in the 15th century by *Leon Alberti*

Design

- Two copper discs of different sizes
- Labeling with alphabet along the edges
- Discs are stacked on top of each other and can be moved together

Basic Concept

- Encryption with Caesar shift for easy application

How was the Enigma designed and what was the key for this electronic cipher machine? How was it possible to decrypt ciphers encrypted by the Enigma?

Cipher Disc

... Encryption disc of the Southern Army in the American Civil War (1861-1865)

Source: Simon Singh, The Code Book, How to make it, break it, hack it, or crack it. Delacorte Press, 2001

Vigenére Encryption using Cipher Machine

Idea

➜ Change the settings of the cipher machine permanently during encryption

Advantage

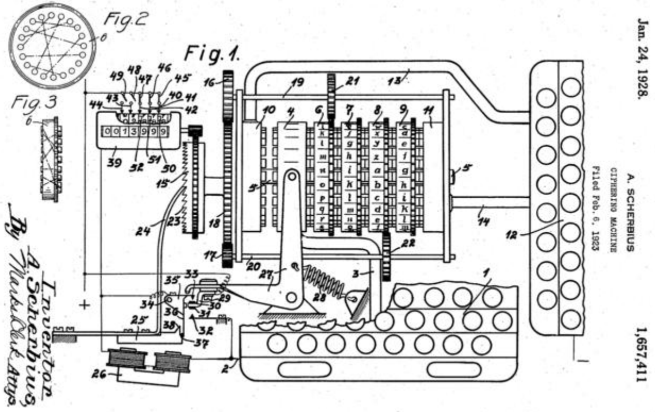
- Polyalphabetic encryption
- Less error-prone than manually encrypting with the Vigenére square

Electronic Cipher Machine: Enigma

- In 1918 a German electronics company was founded
- Company owner *Arthur Scherbius* designed the first Enigma and applied the machine for a patent

- However, there were two parallel developments
  - 1919 *Alexander Koch* (Holland)
  - 1927 *Arvid Damm* (Sweden)

Design of the Enigma by Arthur Scherbius

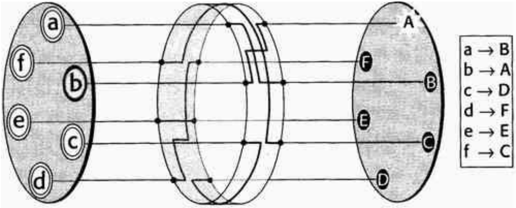


Source: *Patent of the Enigma, published on February 18,1918*

Components of the first Enigma

1. *Keyboard* for entering plaintext letters
2. Encryption unit, a roller riddled with wires (*Rotor*)
3. *Display board* with various lamps for indicating ciphertext letter

Keyboard Rotor Display



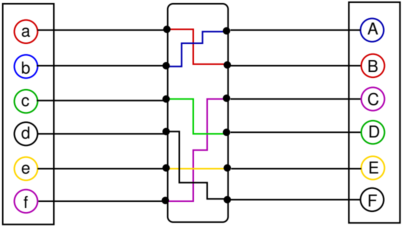
How was the Enigma designed and what was the key for this electronic cipher machine? How was it possible to decrypt ciphers encrypted by the Enigma?

How the first Enigma works?

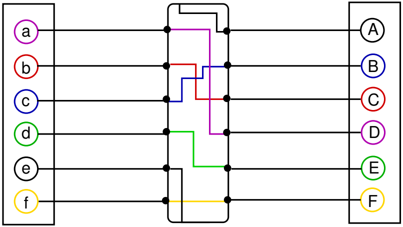
Idea

- When encrypting, the rotor should rotate step by step

Run: Step 1



Run: Step 2



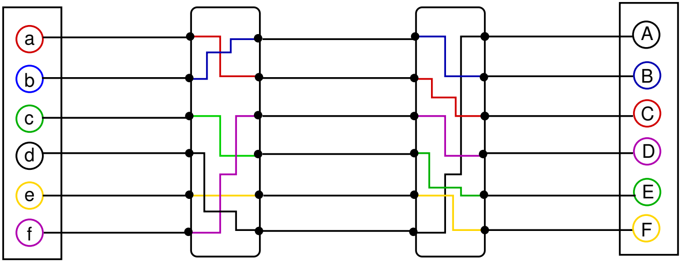
Problem

- Patterns appear in the ciphertext after just one roll rotation

Use of Two Encryption Rotors

- Concept similar to a kilometre counter
- i.e. only after complete rotation of the first rotor, the second one is moved once
- This results in 36 different rotor settings, if you assume only 6 letters on a rotor for simplification

Operation for two rotors



Complete Enigma

Source: *http://de.wikipedia.org/wiki/Enigma_%28Maschine%29*

First Enigma designed by Scherbius

- Additional degree of complexity by installing a third rotor

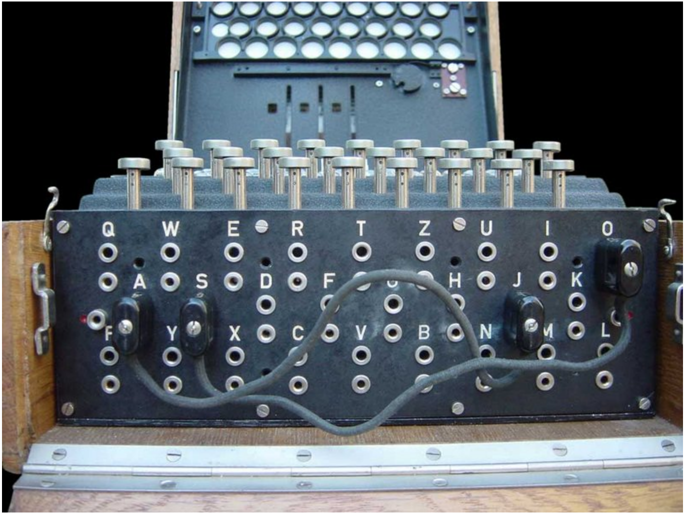
  $26 * 26 * 26 =$ 17576 different rotor settings

- In addition, installation of a **reflector**
  - **Effect:**
    Reflector returns a received signal (other way)
  - **Advantage:**
    Encryption and decryption become mirror-inverted processes, i.e. the same Enigma could be used for encryption and decryption with the same basic configuration

How was the Enigma used in practice?

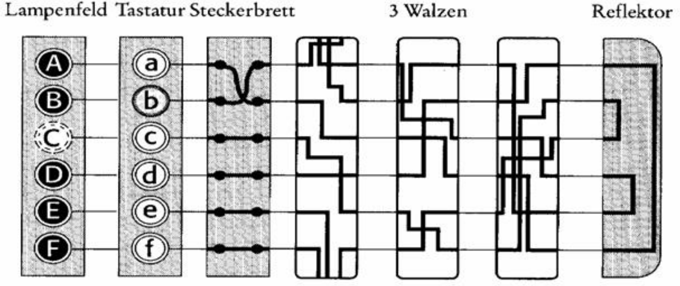
- **Key**
  - Basic setting of the Enigma
- **Requirements** for decryption
  - An identical Enigma
  - Key book
- **Complexity** was further increased by the following activities
  - The position of the rotors could be varied
  - A so-called *plugboard* were also installed, which allowed the swapping of six pairs of letters
- **Classification of the keys** to be used
  - Day keys
  - Message keys

Plugboard of an Enigma



Source: *http://de.wikipedia.org/wiki/Enigma_%28Maschine%29*

How to use the Plugboard?



Complexity of the Enigma

... Three factors have an impact on complexity

1. Rotor settings
   $26 * 26 * 26 = 17.576$
2. Position of the rotors
   6 permutations
3. Plugboard
   6 letter pairs of 26 gives 100.391.791.500 variants

Overall complexity: 10.000.000.000.000.000

Attacks to decrypt the Enigma

- First successful attack by the Polish scientist *Marian Rejewski*
- in collaboration with colleagues

- Poles received documents about the Enigma from the German *Hans Thilo Schmidt*
  1. Instruction manual for an Enigma
  2. Instructions for using the keys

Idea of the Attack

- Analyzing patterns of the ciphertext that may occur as a result of repeated input
- Good circumstance:
  The three-digit *message key was always sent twice* by the Germans at the beginning of a communication

Examples for intercepted keys

1.	L	O	K	R	G	M
2.	M	V	T	X	Z	E
3.	J	K	T	M	P	E
4.	D	V	Y	P	Z	X

How to find pattern?

1. Construct an alphabet step-by-step

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

2. Search for chains (pattern)

$$
\begin{array}{lllllllllllllllllllll}
A & \Rightarrow & F & \Rightarrow & W & \Rightarrow & A \\
B & \Rightarrow & Q & \Rightarrow & Z & \Rightarrow & K & \Rightarrow & V & \Rightarrow & E & \Rightarrow & L & \Rightarrow & R & \Rightarrow & I & \Rightarrow & B \\
C & \Rightarrow & H & \Rightarrow & G & \Rightarrow & O & \Rightarrow & Y & \Rightarrow & D & \Rightarrow & P & \Rightarrow & C
\end{array}
$$

3. Derive the rotor configuration from the pattern

How to derive a rotor configuration from a pattern?

Observation

- Chain length (pattern) is *independent of the plugboard*

A catalogue could be created

- Determining the chain pattern for 105.456 $(6 * (26 * 26 * 26))$ different rotor configurations
- This has been the work of several employees *all year round*

Conclusion

➜ The chain lengths were *fingerprints* leading to a specific rotor configuration!

How the plugboard configuration was determined?

Simple Procedure

1. Initialize your Enigma using a found rotor configuration
2. Disconnect all connectors of the plugboard
3. Decrypt intercepted message with this setting
4. Search for recognizable word formations
5. Determine the connectors of the plugboard step by step

Improvements

- The manually created catalogue for determining the rotor configuration was later no longer necessary
- Calculation machines for decryption were built, called *Bomba*

Enigma becomes even more complex (1938)

Upgrades

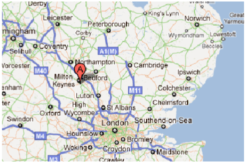
1. Use of 5 rotors and thus 60 different rotor positions
2. The number of possible plug connections has been increased from 6 to 10.

Problem

- Poland lacked the necessary technical means
- So they passed on their knowledge to the English

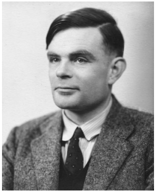
Decryption Attempts by England

- Englishmen built a huge decryption center with up to 7000 employees (*Bletchley Park*)



- At first, attacks were still possible because the message keys were transferred twice at the beginning of a communication

- But the analytical method has also been conceptually expanded by *Alan Turing*



Improved attacks on the Enigma

Problem

What happens if the message key is not sent twice in a row?

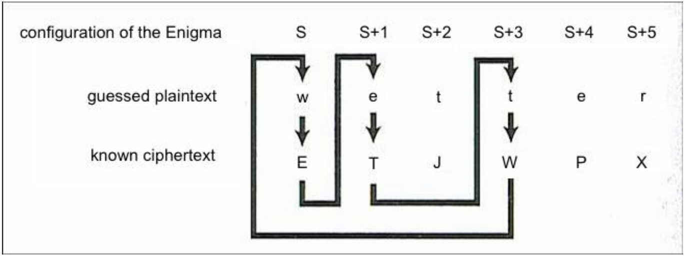
Idea

- Analyze the large library of intercepted radio communications messages
- Search for *clues*, e.g. at 6 o' clock the weather report was always sent, i.e. search for the word *weather*!

Conceptual Implementation

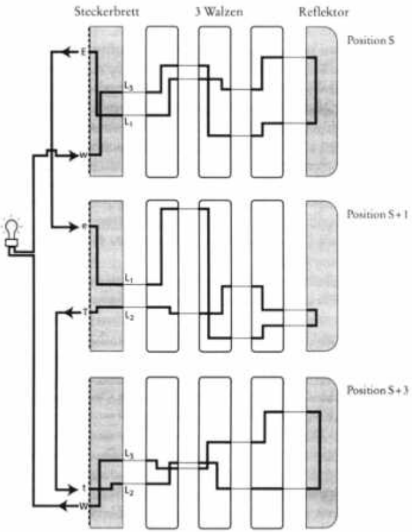
- Search loops in patterns found
- Procedure was very similar to the Polish chain search
- However, much more complex decoding machines have been built for automation

Search Loops



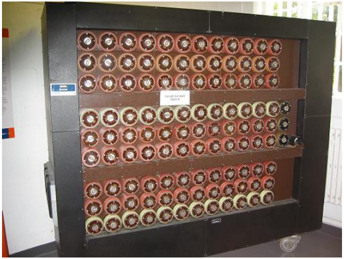
How could the complexity of the plugboard connectors be eliminated?

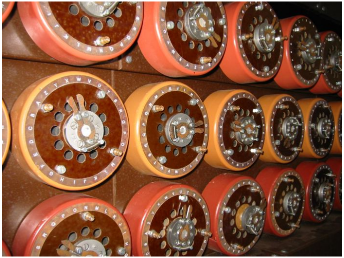
- It was decrypted on the basis of three connected machines
- The plug connections have eliminated each other
- A lamp lit up whenever the search for a loop was successful



Decryption Machine

- The decryption was based on those of Alan Turing developed machines *also called bombs*
- The first prototype was created in 1940
- 49 machines were put into operation in 1942





Source: *http://de.wikipedia.org/wiki/Enigma_%28Maschine%29*