Part IV

Secret-key cryptosystems

- In this chapter we deal with some of the very old or quite old classical (secret-key or symmetric) cryptosystems that were primarily used in the pre-computer era.
- These cryptosystems are too weak nowadays, too easy to break, especially with computers.
- However, these simple cryptosystems give a good illustration of several of the important ideas of the cryptography and cryptanalysis.
- Moreover, most of them can be very useful in combination with more modern cryptosystem - to add a new level of security.

Cryptology (= cryptography + cryptanalysis)
has more than two thousand years of history.

Basic historical observation

- People have always had fascination with keeping information away from others.
- Some people – rulers, diplomats, militaries, businessmen – have always had needs to keep some information away from others.

Importance of cryptography nowadays

- Applications: cryptography is the key tool to make modern information transmission secure, and to create secure information society.
- Foundations: cryptography gave rise to several new key concepts of the foundation of informatics: one-way functions, computationally perfect pseudorandom generators, zero-knowledge proofs, holographic proofs, program self-testing and self-correcting, ...

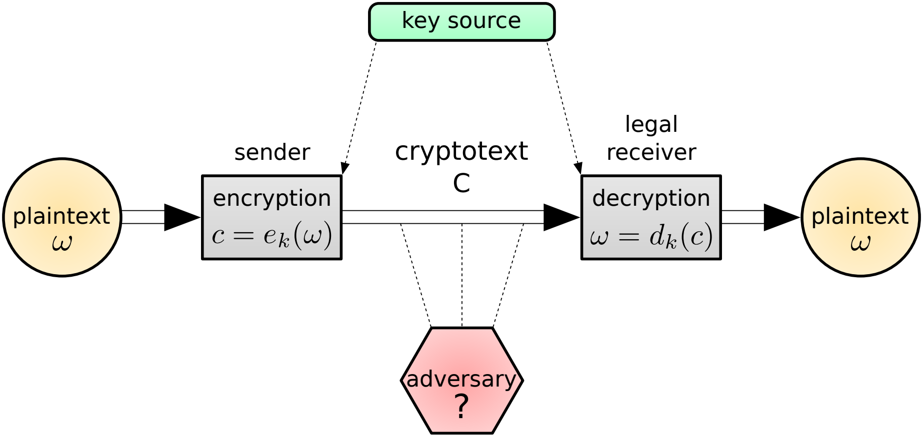
Sound approaches to cryptography

- Shannon's approach based on information theory (enemy has not enough information to break a cryptosystem).
- Current approach based on complexity theory (enemy has not enough computation power to break a cryptosystem).
- Very recent approach based on the laws and limitations of quantum physics (enemy would need to break laws of nature to break a cryptosystem).

Paradoxes of modern cryptography

- Positive results of modern cryptography are based on negative results of complexity theory.
- Computers, that were designed originally for decryption, seem to be now more useful for encryption.

The cryptography deals with problem of sending a message (plaintext, cleartext), through a insecure channel, that may be tapped by an adversary (eavesdropper, cryptanalyst), to a legal receiver.



Components of cryptosystems:

Plaintext-space: P – a set of plaintexts over an alphabet $\sum$

Cryptotext-space: C – a set of cryptotexts (ciphertexts) over alphabet Δ

Key-space: K – a set of keys

Each key k determines an encryption algorithm e_k and an decryption algorithm d_k such that, for any plaintext $w, e_k(w)$ is the corresponding cryptotext and

$$w \in d_k(e_k(w)) \quad \text{or} \quad w = d_k(e_k(w)).$$

Note: As encryption algorithms we can use also randomized algorithms.

CAESAR can be used to encrypt words in any alphabet.

In order to encrypt words in English alphabet we use:

Key-space: $\{0, 1, \ldots, 25\}$

An encryption algorithm e_k substitutes any letter by the letter occurring k positions ahead (cyclically) in the alphabet.

A decryption algorithm d_k substitutes any letter by the one occurring **k** positions backward (cyclically) in the alphabet.

Example e_2(EXAMPLE) $=$ GZCOSNG,
e_2(EXAMPLE) $=$ HADPTOH,
e_1(HAL) $=$ IBM,
e_3(COLD) $=$ FROG

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Example Find the plaintext to the following cryptotext obtained by the encryption with CAESAR with $\mathbf{k} = ?$.

Cryptotext: VHFUHW GH GHXA, VHFUHW GH GLHX,
VHFUHW GH WURLV, VHFUHW GH WRXV.

Numerical version of CAESAR is defined on the set $\{0, 1, 2, \ldots, 25\}$ by the encryption algorithm:

$$e_k(i) = (i + k)(mod\ 26)$$

POLYBIOUS cryptosystem

for encryption of words of the English alphabet without J.

Key-space: Polybious checkerboards 5×5 with 25 English letters and with rows + columns labeled by symbols.

Encryption algorithm: Each symbol is substituted by the pair of symbols denoting the row and the column of the checkerboard in which the symbol is placed.

Example:

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

KONIEC $\rightarrow$

Decryption algorithm: ???

Kerckhoff's Principle

The philosophy of modern cryptanalysis is embodied in the following principle formulated in 1883 by Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof (1835 - 1903).

The security of a cryptosystem must not depend on keeping secret the encryption algorithm. The security should depend only on *keeping secret the key.*

(Sir Francis R. Bacon (1561 - 1626))

1. Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.
2. Given d_k and a cryptotext c, it should be easy to compute $w = d_k(c)$.
3. A cryptotext $e_k(w)$ should not be much longer than the plaintext w.
4. It should be unfeasible to determine w from $e_k(w)$ without knowing d_k.
5. The so called avalanche effect should hold: A small change in the plaintext, or in the key, should lead to a big change in the cryptotext (i.e. a change of one bit of the plaintext should result in a change of all bits of the cryptotext, each with the probability close to 0.5).
6. The cryptosystem should not be closed under composition, i.e. not for every two keys k_1, k_2 there is a key k such that
$$e_k(w) = e_{k_1}(e_{k_2}(w)).$$
7. The set of keys should be very large.

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptoanalytics attack

1. Cryptotexts-only attack. The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \dots, c_n = e_k(w_n)$ and try to infer the key k or as many of the plaintexts $w_1, \dots, w_n$ as possible.
2. Known-plaintexts attack **(given are some pairs plaintext $\rightarrow$ cryptotext)** The cryptanalysts know some pairs $w_i, e_k(w_i), 1 \leq i \leq n$, and try to infer k, or at least w_{n+1} for a new cryptotext $e_k(w_{n+1})$.
3. Chosen-plaintexts attack (given are cryptotext for some chosen plaintexts) The cryptanalysts choose plaintexts $w_1, \dots, w_n$ to get cryptotexts $e_k(w_1), \dots, e_k(w_n)$, and try to infer k or at least w_{n+1} for a new cryptotext $c_{n+1} = e_k(w_{n+1})$. (For example, if they get temporary access to encryption machinery.)

4. Known-encryption-algorithm attack
   The encryption algorithm e_k is given and the cryptanalysts try to get the decryption algorithm d_k.

5. Chosen-cryptotext attack (given are plaintexts for some chosen cryptotexts)
   The cryptanalysts know some pairs

   $$(c_i, d_k(c_i)), \quad 1 \leq i \leq n,$$

   where the cryptotexts c_i have been chosen by the cryptanalysts. The aim is to determine the key. (For example, if cryptanalysts get a temporary access to decryption machinery.)

WHAT CAN a BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

- Eve can read (and try to decrypt) the message.
- Eve can try to get the key that was used and then decrypt all messages encrypted with the same key.
- Eve can change the message sent by Alice into another message, in such a way that Bob will have the feeling, after he gets the changed message, that it was a message from Alice.
- Eve can pretend to be Alice and communicate with Bob, in such a way that Bob thinks he is communicating with Alice.

An eavesdropper can therefore be passive - Eve or active - Mallot.

Basic goals of broadly understood cryptography

Confidentiality: Eve should not be able to decrypt the message Alice sends to Bob.

Data integrity: Bob wants to be sure that Alice's message has not been altered by Eve.

Authentication: Bob wants to be sure that only Alice could have sent the message he has received.

Non-repudiation: Alice should not be able to claim that she did not send messages that she has sent.

Anonymity: Alice does not want that Bob finds who send the message

HILL cryptosystem

The cryptosystem presented in this slide was probably never used. In spite of that this cryptosystem played an important role in the history of modern cryptography.

We describe Hill cryptosystem for a fixed n and the English alphabet.

Key-space: matrices M of degree n with elements from the set $\{0, 1, \ldots, 25\}$ such that $M^{-1} mod$ 26 exist.

Plaintext + cryptotext space: English words of length n.

Encoding: For a word w let c_w be the column vector of length n of the integer codes of symbols of w. $(A \rightarrow 0, B \rightarrow 1, C \rightarrow 2, \ldots)$

Encryption: $c_c = Mc_w$ mod 26

Decryption: $c_w = M^{-1}c_c$ mod 26

Example 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

$$M = \begin{bmatrix} 4 & 7 \\ 1 & 1 \end{bmatrix} M^{-1} = \begin{bmatrix} 17 & 11 \\ 9 & 16 \end{bmatrix}$$

Plaintext: $w = $ LONDON

$$C_{LO} = \begin{bmatrix} 11 \\ 14 \end{bmatrix}, C_{ND} = \begin{bmatrix} 13 \\ 3 \end{bmatrix}, C_{ON} = \begin{bmatrix} 14 \\ 13 \end{bmatrix}$$

$$MC_{LO} = \begin{bmatrix} 12 \\ 25 \end{bmatrix}, MC_{ND} = \begin{bmatrix} 21 \\ 16 \end{bmatrix}, MC_{ON} = \begin{bmatrix} 17 \\ 1 \end{bmatrix}$$

Cryptotext: MZVQRB

Theorem

$$\text{If } M = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \text{ then} M^{-1} = \frac{1}{\det\ M} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Proof: Exercise

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: CAESAR, HILL. Another name is symmetric cryptosystem (cryptography).

Two basic types of secret-key cryptosystems

- **substitution** based cryptosystems
- **transposition** based cryptosystems

Two basic types of substitution cryptosystems

- **monoalphabetic cryptosystems** – they use a fixed substitution – CAESAR, POLYBIOUS
- **polyalphabetic cryptosystems** – substitution keeps changing during the encryption

A monoalphabetic cryptosystem with letter-by-letter substitution is uniquely specified by a permutation of letters. (Number of permutations (keys) is 26!)

Example: AFFINE cryptosystem is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14, e_{3,5}(15) = 24 - e_{3,5}(D) = 0, e_{3,5}(P) = Y$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Decryption: $d_{a,b}(y) = a^{-1}(y - b) \bmod 26$

Cryptanalysis

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a frequency count: the number of each letter in the cryptotext is counted. The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext laguage.

The letter with the highest frequency in the cryptotext is likely to be substitute for the letter with highest frequency in the plaintext language . . . . The likehood grows with the length of cryptotext.

Frequency counts in English:

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

and for other languages:

English	%	German	%	Finnish	%	French	%	Italian	%	Spanish	%
E	12.31	E	18.46	A	12.06	E	15.87	E	11.79	E	13.15
T	9.59	N	11.42	I	10.59	A	9.42	A	11.74	A	12.69
A	8.05	I	8.02	T	9.76	I	8.41	I	11.28	O	9.49
O	7.94	R	7.14	N	8.64	S	7.90	O	9.83	S	7.60
N	7.19	S	7.04	E	8.11	T	7.29	N	6.88	N	6.95
I	7.18	A	5.38	S	7.83	N	7.15	L	6.51	R	6.25
S	6.59	T	5.22	L	5.86	R	6.46	R	6.37	I	6.25
R	6.03	U	5.01	O	5.54	U	6.24	T	5.62	L	5.94
H	5.14	D	4.94	K	5.20	L	5.34	S	4.98	D	5.58

The 20 most common digrams are (in decreasing order) TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS. The six most common trigrams: THE, ING, AND, HER, ERE, ENT.

Cryptanalysis of a cryptotext encrypted using the AFINE cryptosystem with an encryption algorithm

$$e_{a,b}(x) = (ax + b) \bmod 26 = (xa + b) \bmod 26$$

where $0 \leq a, b \leq 25, gcd(a, 26) = 1$. (Number of keys: $12 \times 26 = 312$.)

Example: Assume that an English plaintext is divided into blocks of 5 letters and encrypted by an AFINE cryptosystem (ignoring space and interpunctions) as follows:

How to find the plaintext?

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

Frequency analysis of plainext and frequency table for English:

X - 32	J - 11	D - 2
U - 30	O - 6	V - 2
H - 23	R - 6	F - 1
B - 19	G - 5	P - 1
L - 19	M - 4	E - 0
N - 16	Y - 4	I - 0
K - 15	Z - 4	Q - 0
S - 15	C - 3	T - 0
W - 14	A - 2	

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

First guess: $E = X, T = U$

Encodings: $xa + b = y$

$$4a + b = 23 \pmod{26}$$

$$19a + b = 20 \pmod{26}$$

Solutions: $a = 5, b = 3 \rightarrow a^{-1} =$

Translation table

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

```
BHJUH  NBULS  VULRU  SLYXH
ONUUN  BWNUA  XUSNL  UYJSS
WXRLK  GNBON  UUNBW  SWXKX
HKXDH  UZDLK  XBHJU  HBNUO
NUMHU  GSWHU  XMBXR  WXKXL
UXBHJ  UHCXK  XAXKZ  SWKXX
LKOLJ  KCXLC  MXONU  UBVUL
RRWHS  HBHJU  HNBXM  BXRWX
KXNOZ  LJBXX  HBNFU  BHJUH
LUSWX  GLLKZ  LJPHU  ULSYX
BJKXS  WHSSW  XKXNB  HBHJU
HYXWN  UGSWX  GLLK
```

provides from the above cryptotext the plaintext that starts with KGWTG CKTMO OTMIT DMZEG, what does not make sense.

Second guess: $E = X, A = H$

Equations
$$4a + b = 23 \pmod{26}$$
$$b = 7 \pmod{26}$$

Solutions: $a = 4$ or $a = 17$ and therefore $a = 17$

This gives the translation table

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

and the following plaintext from the above cryptotext

SAUNA ISNOT KNOWN TOBEA
FINNI SHINV ENTIO NBUTT
HEWOR DISFI NNISH THERE
AREMA NYMOR ESAUN ASINF
INLAN DTHAN ELSEW HEREO
NESAU NAPER EVERY THREE
ORFOU RPEOP LEFIN NSKNO
WWHAT ASAUN AISEL SEWHE
REIFY OUSEE ASIGN SAUNA
ONTHE DOORY OUCAN NOTBE
SURET HATTH EREIS ASAUN
ABEHI NDTHE DOOR

Example of monoalphabetic cryptosystem

Symbols of the English alphabet will be replaced by squares with or without points and with or without surrounding lines using the following rule:

A:	B:	C:
D:	E:	F:
G:	H:	I:

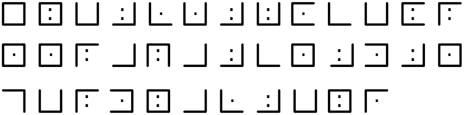
J·	K·	L·
M·	N·	O·
P·	Q·	R·

S	T	U
V	W	X
Y	Z	

For example the plaintext:

WE TALK ABOUT FINNISH SAUNA MANY TIMES LATER

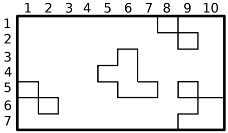
results in the cryptotext:



Garbage in between method: the message (plaintext or cryptotext) is supplemented by "garbage letters".

Richelieu cryptosystem used sheets of card board with holes.

I		L	O	V	E		Y	O	U
I		H	A	V	E		Y	O	U
D	E	E	P		U	N	D	E	R
M	Y		S	K	I	N		M	Y
L	O	V	E		L	A	S	T	S
F	O	R	E	V	E	R		I	N
H	Y	P	E	R	S	P	A	C	E



Playfair cryptosystem

Invented around 1854 by Ch. Wheatstone.

Key – a Playfair square is defined by a word w of length at most 25. In w repeated letters are then removed, remaining letters of alphabets (except j) are then added and resulting word is divided to form an 5 x 5 array (a Playfair square).

Encryption: of a pair of letters x, y

1. If x and y are in the same row (column), then they are replaced by the pair of symbols to the right (bellow) them.
2. If x and y are in different rows and columns they are replaced by symbols in the opposite corners of rectangle created by x and y.

Example: PLAYFAIR is encrypted as LCMNNFCS

Playfair was used in World War I by British army.

Playfair square:

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

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

A 26×26 table is first designed with the first row containing a permutation of all symbols of alphabet and all columns represent CAESAR shifts starting with the symbol of the first row.

Secondly, for a plaintext w a key k is a word of the same length as w.

Encryption: the *i-th* letter of the plaintext - w_i is replaced by the letter in the w_i-row and k_i-column of the table.
VIGENERE cryptosystem: a short keyword p is chosen and

$$k = Prefix_{|w|} p^{oo}$$

VIGENERE is actually a cyclic version of the CAESAR cryptosystem.

AUTOCLAVE cryptosystem: $k = Prefix_{|w|} pw$

VIGENERE and AUTOCLAVE cryptosystems

Example:

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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

Keyword:	H A M B U R G
Plaintext:	I N J E D E M M E N S C H E N G E S I C H T E S T E H T S E I N E G
Vigenere-key:	H A M B U R G H A M B U R G H A M B U R G H A M B U R G H A M B U R
Autoclave-key:	H A M B U R G I N J E D E M M E N S C H E N G E S I C H T E S T E H
Vigerere-cryp.:	P N V F X V S T E Z T W Y K U G Q T C T N A E E V Y Y Z Z E U O Y X
Autoclave-cryp.:	P N V F X V S U R W W F L Q Z K R K K J L G K W L M J A L I A G I N

1. Task 1 – to find the length of the key

Kasiski method (1852) - invented also by Charles Babbage (1853).

Basic observation If a subword of a plaintext is repeated at a distance that is a multiple of the length of the key, then the corresponding subwords of the cryptotext are the same.

Example, cryptotext:

CHRGQPWOEIRULYANDOSHCHRIZKEBUSNOFKYWROPDCHRKGAXBNRHROAKERBKSCHRIWK

Substring “CHR” occurs in positions 1, 21, 41, 66: expected keyword length is therefore 5.

Method. Determine the greatest common divisor of the distances between identical subwords (of length 3 or more) of the cryptotext.

Friedman method Let n_i be the number of occurrences of the **i**-*th* letter in the cryptotext.

Let **l** be the length of the keyword.

Let **n** be the length of the cryptotext.

Then it holds $l = \frac{0.027n}{(n-2)I - 0.038n + 0.065}$, $I = \sum_{i=1}^{26} \frac{n_i(n_i-1)}{n(n-1)}$

Once the length of the keyword is found it is easy to determine the key using the statistical (frequency analysis) method of analyzing monoalphabetic cryptosystems.

1. Let n_i be the number of occurrences of i-th alphabet symbol in a text of length n. The probability that if one selects a pair of symbols from the text, then they are the same is

$$I = \frac{\sum_{i=1}^{26} n_i(n_i-1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{n_i}{2}}{\binom{n}{2}}$$

and it is called the index of coincidence.

2. Let p_i be the probability that a randomly chosen symbol is the i-th symbol of the alphabet. The probability that two randomly chosen symbols are the same is

$$\sum_{i=1}^{26} p_i^2$$

For English text one has

$$\sum_{i=1}^{26} p_i^2 = 0.065$$

For randomly chosen text:

$$\sum_{i=1}^{26} p_i^2 = \sum_{i=1}^{26} \frac{1}{26^2} = 0.038$$

Approximately

$$I = \sum_{i=1}^{26} p_i^2$$

Derivation of the Friedman method

Assume that a cryptotext is organized into l columns headed by the letters of the keyword

letters S_l	S_1	S_2	S_3	$\dots$	S_l
	x_1	x_2	x_3	$\dots$	X_l
	x_{l+1}	x_{l+2}	x_{l+3}		X_{2l}
	x_{2l+1}	x_{2l+2}	x_{2l+3}	$\dots$	x_{3l}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem. Probability that two randomly chosen letters are the same in

- the same column is 0.065.
- different columns is 0.038.

The number of pairs of letters in the same column: $\frac{l}{2} \cdot \frac{n}{l}(\frac{n}{l} - 1) = \frac{n(n-l)}{2l}$

The number of pairs of letters in different columns: $\frac{l(l-1)}{2} \cdot \frac{n^2}{l^2} = \frac{n^2(n-l)}{2l}$

The expected number A of pairs of equals letters is $A = \frac{n(n-l)}{2l} \cdot 0.065 + \frac{n^2(l-1)}{2l} \cdot 0.038$

Since $l = \frac{A}{\frac{n(n-1)}{2}} = \frac{1}{l(n-1)}[0.027 + l(0.038n - 0.065)]$

one gets the formula for l from the previous slide.

ONE-TIME PAD cryptosystem – Vernam's cipher

$$\text{Binary case:}\quad \left.\begin{array}{ll} \text{plaintext} & w \\ \text{key} & k \\ \text{cryptotext} & c \end{array}\right\} \text{ are binary words of the same length}$$

Encryption: $c = w \oplus k$
Decryption: $w = c \oplus k$
Example:

$$\begin{aligned} w &= 101101011 \\ k &= 011011010 \\ c &= 110110001 \end{aligned}$$

What happens if the same key is used twice or 3 times for encryption?

$$c_1 = w_1 \oplus k, c_2 = w_2 \oplus k, c_3 = w_3 \oplus k$$

$$\begin{aligned} c_1 \oplus c_2 &= w_1 \oplus w_2 \\ c_1 \oplus c_3 &= w_1 \oplus w_3 \\ c_2 \oplus c_3 &= w_2 \oplus w_3 \end{aligned}$$

Perfect secret cryptosystems

By Shannon, a cryptosystem is perfect if the knowledge of the cryptotext provides no information whatsoever about its plaintext (with the exception of its length).

It follows from Shannon's results that perfect secrecy is possible if the key-space is as large as the plaintext-space. In addition, a key has to be as long as plaintext and the same key should not be used twice.

An example of a perfect cryptosystem ONE-TIME PAD cryptosystem (Gilbert S. Vernam (1917) - AT&T + Major Joseph Mauborgne).

If used with the English alphabet, it is simply a polyalphabetic substitution cryptosystem of VIGENERE with the key being a randomly chosen English word of the same length as the plaintext.

Proof of perfect secrecy: by the proper choice of the key any plaintext of the same length could provide the given cryptotext.

Did we gain something? The problem of secure communication of the plaintext got transformed to the problem of secure communication of the key of the same length.

Yes:

1. ONE-TIME PAD cryptosystem is used in critical applications
2. It suggests an idea how to construct practically secure cryptosystems.

The basic idea is very simple: permutate the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose n, write plaintext into rows, with n symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

Newton wrote to Leibnitz

$$a^7 c^2 d^2 e^{14} f^2 i^7 l^3 m^1 n^8 o^4 q^3 r^2 s^4 t^8 v^{12} x^1$$

what stands for: "data aequatione quodcumque fluentes quantitates involvente, fluxiones invenire et vice versa"

Example $\quad a^2 cdef^3 g^2 i^2 jkmn^8 o^5 prs^2 t^2 u^3 z$

Solution:

Choose an integer $0 < k < 25$ and a string, called keyword, of length at most 25 with all letters different.

The keyword is then written bellow the English alphabet letters, beginning with the k-symbol, and the remaining letters are written in the alphabetic order and cyclicly after the keyword.

Example: keyword: HOW MANY ELKS, $k = 8$

0								8																	
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
P	Q	R	T	U	V	X	Z	H	O	W	M	A	N	Y	E	L	K	S	B	C	D	F	G	I	J

Exercise Decrypt the following cryptotext encrypted using the KEYWORD CAESAR and determine the keyword and k

T IVD ZCRTIC FQNIQ TU TF
Q XAVFCZ FEQXC PCQUCZ WK
Q FUVBC FNRRTXTCIUAK WTY
DTUP MCFECXU UV UPC BVANHC
VR UPC FEQXC UPC FUVBC
XVIUQTIF FUVICF NFNQAAK
VI UPC UVE UV UQGC Q FQNIQ
WQUP TU TF QAFV ICXCFFQMK
UPQU UPC FUVBC TF EMVECMAK
PCQUCZ QIZ UPQU KVN PQBC
UPC RQXTATUK VR UPMVDTIY
DQUCM VI UPC FUVICF

KEYWORD CAESAR cryptosystem

Step 1. Make the frequency counts:

	Number		Number		Number
U	32	X	8	W	3
C	31	K	7	Y	2
Q	23	N	7	G	1
F	22	E	6	H	1
V	20	M	6	J	0
P	15	R	6	L	0
T	15	B	5	O	0
I	14	Z	5	S	0
A	8	D	4		
	180=74.69%		54=22.41%		7=2.90%

Step 2. Cryptotext contains two one-letter words T and Q. They must be A and I. Since T occurs once and Q three times it is likely that T is I and Q is A.

The three letter word UPC occurs 7 times and all other 3-letter words occur only once. Hence

UPC is likely to be THE.

Let us now decrypt the remaining letters in the high frequency group: F,V,I

$$\text{From the words TU, TF} \Rightarrow \text{F=S}$$
$$\text{From UV} \Rightarrow \text{V=O}$$
$$\text{From VI} \Rightarrow \text{I=N}$$

The result after the remaining guesses

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
L	V	E	W	P	S	K	M	N	?	Y	?	R	U	?	H	E	F	?	I	T	O	B	C	G	D

UNICITY DISTANCE of CRYPTOSYSTEMS

Redundancy of natural languages is of the key importance for cryptanalysis.

Would all letters of a 26-symbol alphabet have the same probability, a character would carry $\lg 26 = 4.7$ bits of Information.

The estimated average amount of information carried per letter in a meaningful English text is 1.5 bits.

The unicity distance of a cryptosystem is the minimum number of cryptotext (number of letters) required to a computationally unlimited adversary to recover the unique encryption key.

Empirical evidence indicates that if any simple cryptosystem is applied to a meaningful English message, then about 25 cryptotext characters is enough for an experienced cryptanalyst to recover the plaintext.

ANAGRAMS – EXAMPLES

German:

IRI BRÄTER, GENF	Briefträgerin
FRANK PEKL, REGEN	...
PEER ASSSTIL, MELK	...
INGO DILMR, PEINE	...
EMIL REST, GERA	...
KARL SORDORT, PEINE	...

English:

algorithms	logarithms
antagonist	stagnation
compressed	decompress
coordinate	decoration
creativity	reactivity
deductions	discounted
descriptor	predictors
impression	permission
introduces	reductions
procedures	reproduces

APPENDIX

STREAM CRYPTOSYSTEMS

Two basic types of cryptosystems are:

- Block cryptosystems (Hill cryptosystem,. . .) – they are used to encrypt simultaneously blocks of plaintext.
- Stream cryptosystems (CAESAR, ONE-TIME PAD,. . .) – they encrypt plaintext letter by letter, or block by block, using an encryption that may vary during the encryption process.

Stream cryptosystems are more appropriate in some applications (telecommunication), usually are simpler to implement (also in hardware), usually are faster and usually have no error propagation (what is of importance when transmission errors are highly probable).

Two basic types of stream cryptosystems: secret key cryptosystems (ONE-TIME PAD) and public-key cryptosystems (Blum-Goldwasser)

Block versus stream cryptosystems

In block cryptosystems the same key is used to encrypt arbitrarily long plaintext – block by block - (after dividing each long plaintext w into a sequence of subplaintexts (blocks) $w_1 w_2 w_3$).

In stream cryptosystems each block is encryptyd using a different key

- The fixed key k is used to encrypt all blocks. In such a case the resulting cryptotext has the form

$$c = c_1 c_2 c_3 \ldots = e_k(w_1) e_k(w_2) e_k(w_3) \ldots$$

- A stream of keys is used to encrypt subplaintexts. The basic idea is to generate a key-stream $K = k_1, k_2, k_3, \ldots$ and then to compute the cryptotext as follows

$$c = c_1 c_2 c_3 \ldots = e_{k1}(w_1) e_{k2}(w_2) e_{k3}(w_3).$$

CRYPTOSYSTEMS WITH STREAMS OF KEYS

Various techniques are used to compute a sequence of keys. For example, given a key k

$$k_i = f_i(k, k_1, k_2, \ldots, k_{i-1})$$

In such a case encryption and decryption processes generate the following sequences:

Encryption: To encrypt the plaintext $w_1 w_2 w_3 \ldots$ the sequence

$$k_1, c_1, k_2, c_2, k_3, c_3, \ldots$$

of keys and sub-cryptotexts is computed.

Decryption: To decrypt the cryptotext $c_1 c_2 c_3 \ldots$ the sequence

$$k_1, w_1, k_2, w_2, k_3, w_3, \ldots$$

of keys and subplaintexts is computed.

EXAMPLES

A keystream is called synchronous if it is independent of the plaintext.

KEYWORD VIGENERE cryptosystem can be seen as an example of a synchronous keystream cryptosystem.

Another type of the binary keystream cryptosystem is specified by an initial sequence of keys $k_1, k_2, k_3 \dots k_m$

and a initial sequence of binary constants $b_1, b_2, b_3 \dots b_{m-1}$

and the remaining keys are computed using the rule

$$k_{i+m} = \sum_{j=0}^{m-1} b_j k_{i+j} \bmod 2$$

A keystream is called periodic with period p if $k_{i+p} = k_i$ for all i.

Example Let the keystream be generated by the rule

$$k_{i+4} = k_i \oplus k_{i+1}$$

If the initial sequence of keys is (1,0,0,0), then we get the following keystream:

1,0,0,0,1,0,0,1,1,0,1,0 1,1,1, ...

of period 15.

Let **P, K** and **C** be sets of plaintexts, keys and cryptotexts.
Let $p_K(k)$ be the probability that the key k is chosen from **K** and let a priori probability that plaintext w is chosen be $p_p(w)$.
If for a key $k \in K, C(k) = \{e_k(w) | w \in P\}$, then for the probability $P_C(y)$ that c is the cryptotext that is transmitted it holds

$$p_c(c) = \sum_{\{k | c \in C(k)\}} p_K(k) p_P(d_k(c)).$$

For the conditional probability $p_c(c|w)$ that c is the cryptotext if w is the plaintext it holds

$$p_c(c|w) = \sum_{\{k | w = d_k(c)\}} p_K(k).$$

Using Bayes' conditional probability formula $p(y)p(x|y) = p(x)p(y|x)$ we get for probability $p_P(w|c)$ that w is the plaintext if c is the cryptotext the expression

$$p_P = \frac{P_P(w) \sum_{\{k | w = d_k(c)\}} p_K(k)}{\sum_{\{k | c \in C(K)\}} p_K(k) p_P(d_K(c))}.$$

Definition A cryptosystem has perfect secrecy if

$$p_P(w|c) = p_P(w) \text{ for all } w \in P \text{ and } c \in C.$$

(That is, the a posteriori probability that the plaintext is w,given that the cryptotext is c is obtained, is the same as a priori probability that the plaintext is w.)

Example CAESAR cryptosystem has perfect secrecy if any of the 26 keys is used with the same probability to encode any symbol of the plaintext.

Proof Exercise.

An analysis of perfect secrecy: The condition $p_P(w|c) = p_P(w)$ is for all $w \in P$ and $c \in C$ equivalent to the condition $p_C(c|w) = p_C(c)$.

Let us now assume that $p_C(c) > 0$ for all $c \in C$.

Fix $w \in P$. For each $c \in C$ we have $p_C(c|w) = p_C(c) > 0$. Hence, for each $c \in C$ there must exist at least one key k such that $e_k(w) = c$. Consequently, $|K| \geq |C| \geq |P|$.

In a special case $|K| = |C| = |P|$, the following nice characterization of the perfect secrecy can be obtained:

Theorem A cryptosystem in which $|P| = |K| = |C|$ provides perfect secrecy if and only if every key is used with the same probability and for every $w \in P$ and every $c \in C$ there is a unique key k such that $e_k(w) = c$.

Proof Exercise.

PRODUCT CRYPTOSYSTEMS

A cryptosystem $S = (P, K, C, e, d)$ with the sets of plaintexts P, keys K and cryptotexts C and encryption (decryption) algorithms $e(d)$ is called **endomorphic** if $P = C$.

If $S_1 = (P, K_1, P, e^{(1)}, d^{(1)})andS_2 = (P, K_2, P, e^{(2)}, d^{(2)})$ are endomorphic cryptosystems, then the **product cryptosystem** is

$$S_1 \otimes S_2 = (P, K_1 \otimes K_2, P, e, d),$$

where encryption is performed by the procedure

$$e_{(k1,k2)}(w) = e_{k2}(e_{k1}(w))$$

and decryption by the procedure

$$d_{(k1,k2)}(c) = d_{k1}(d_{k2}(c)).$$

Example (Multiplicative cryptosystem):

Encryption: $e_a(w) = aw \bmod p$; decryption: $d_a(c) = a^{-1}c \bmod 26$.

If M denote the multiplicative cryptosystem, then clearly CAESAR $\times$ M is actually the AFFINE cryptosystem.

Exercise Show that also M $\otimes$ CAESAR is actually the AFFINE cryptosystem.

Two cryptosystems S_1 and S_2 are called **commutative** if $S_1 \otimes S_2 = S_2 \otimes S_1$.

A cryptosystem S is called **idempotent** if $S \otimes S = S$.