CRYPTOLOGY

and Computer Security

José Ignacio Farrán Martín

jifarran@eii.uva.es

Department of Applied Mathematics

University of Valladolid – Campus of Segovia (Spain)

Escuela Universitaria de Informática

Contents

- Introduction to Cryptology
- Private Key and Public Key
- Practical Applications

Contents

- Introduction to Cryptology
- Private Key and Public Key
- Practical Applications

Introduction to Cryptology

Cryptology

- Goal: protect secret/confidential information against the access by non-authorized persons
- Consists of two areas, apparently opposed each other, but actually complementary:
  - Cryptography: designs methods for enciphering (hide information)
  - Cryptoanalysis: tries to “break” those enciphering methods to obtain the original information (or the secret key)

Basic Scheme (I)

EAVESDROPPER!

$$\downarrow$$

TRANSMITTER $\longrightarrow$ $\boxed{\textbf{PLAIN TEXT}}$ $\longrightarrow$ **RECEIVER**

Basic Scheme (II)

EAVESDROPPER?

$$\downarrow$$

	encipher		decipher	
TRANSMITTER	$\longrightarrow$	$\boxed{\textbf{CIPHER TEXT}}$	$\longrightarrow$	**RECEIVER**

Basic Scheme (II)

$$
\begin{array}{ccccc}
 & & \textbf{EAVESDROPPER?} & & \\
 & & \downarrow & & \\
 & \text{encipher} & & \text{decipher} & \\
\textbf{TRANSMITTER} & \longrightarrow & \boxed{\textbf{CIPHER TEXT}} & \longrightarrow & \textbf{RECEIVER} \\
 & \text{key} & \leftrightarrow & \text{key} &
\end{array}
$$

Key points

- Cryptogram: the result of applying an enciphering process to a plaintext, controlled by a ciphering key
- If the receiver is an authorized person, he knows the deciphering key and can retrieve the original message
- Aim: the “eavesdropper” cannot obtain this information (decipher) without the knowledge of that key, which must be kept secret

Key points

- A good Cryptosystem is that one where
  - the encipher/decipher algorithms are simple and fast if the key is known, but
  - it turns out to be impossible (or at least computationally time-consuming) deciphering without the key

Applications

- Originally, applications were governmental and military

Applications

- Originally, applications were governmental and military
- Currently, there are many other applications:
  - Protect "data banks" (personal, bank data, etc)
  - Control the access to computer networks (passwords, intelligent cards, etc)
  - e-Commerce and secure transactions
  - Digital Signature of electronic documents and communications

Applications

- Originally, applications were governmental and military
- Currently, there are many other applications:
  - Protect “data banks” (personal, bank data, etc)
  - Control the access to computer networks (passwords, intelligent cards, etc)
  - e-Commerce and secure transactions
  - Digital Signature of electronic documents and communications

Applications

- Originally, applications were governmental and military
- Currently, there are many other applications:
  - Protect “data banks” (personal, bank data, etc)
  - Control the access to computer networks (passwords, intelligent cards, etc)
  - e-Commerce and secure transactions
  - Digital Signature of electronic documents and communications

Applications

- Originally, applications were governmental and military
- Currently, there are many other applications:
  - Protect “data banks” (personal, bank data, etc)
  - Control the access to computer networks (passwords, intelligent cards, etc)
  - e-Commerce and secure transactions
  - Digital Signature of electronic documents and communications

Applications

- Originally, applications were governmental and military
- Currently, there are many other applications:
  - Protect “data banks” (personal, bank data, etc)
  - Control the access to computer networks (passwords, intelligent cards, etc)
  - e-Commerce and secure transactions
  - Digital Signature of electronic documents and communications

Historical perspective

* Caesar Cipher (century I b.C.):
  - Take a letter as key (f.e. C)
  - Cipher: add the key to all the letters of the message
    (modulo 21, the number of letters in LATIN)
  - Decipher: subtract the same key (modulo 21) to all the letters of the cryptogram

Historical perspective

* Caesar Cipher (century I b.C.):

- Example:

	V	E	N	I	V	I	D	I	V	I	C	I
+	C	C	C	C	C	C	C	C	C	C	C	C
	A	G	P	L	A	L	F	L	A	L	E	L

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

Historical perspective

* Caesar Cipher (century I b.C.):

  - Once you know the cipher method, deciphering is very fast, even by hand, since we can test all the (non-trivial) 20 keys until we get a message that makes sense

Historical perspective

∗ Vigenère Cipher (1586):

- Take a word of k letters as key (f.e. LOUP)
- Cipher: add the key (repeat, if necessary) to the plaintext (modulo 26, or in general, the number of letters of the used language)
- Decipher: subtract the key (modulo 26) to the cryptogram

Historical perspective

∗ Vigenère Cipher (1586):

- Example:

```
   PARIS VAUT BIEN UNE MESSE
 + LOUPL OUPL OUPL OUP LOUPL
-----------------------------
   AOLXD JUJE PCTY IHT XSMHP
```

Historical perspective

* Vigenère Cipher (1586):

  • Now the number of keys to try is much higher (26^k if you do not use a dictionary), but the Kasiski method, based on “analysis of frequencies”, allows us determine the length k of the key, and afterwards everything is reduced to solve k simple “Caesar cryptograms” (feasible currently with the aid of a computer)

Historical perspective

∗ Beaufort Cipher (1710):

- Variant of Vigenère Cipher: subtract the plaintext to the key (modulo 26)
- Analysis: similar cryptographic properties to the Vigenère Cipher
- Practical advantage:
  the cipher is an involution
  (use the same key and the same algorithm for cipher and decipher, i.e. ciphering twice you obtain the original message)

Substitution Methods

* The three previous methods are of this type
  - Establish a bijective map between two alphabets of the same cardinality, and substitute each letter by its image in the other alphabet
  - Particular case: generic permutation of an alphabet (number of keys 26!)

• In the cases of Vigenère and Beaufort: divide the message into k parts and apply a different permutation to each part

Analysis of frequencies

- Apparently these substitution systems are secure because of the high number of keys

$$26! = 403291461126605635584000000$$

- Nevertheless, these methods keep the characteristic frequencies and statistics of the used language:
  - Frequencies of letters (by languages, text types, etc)
  - Possible/impossible combinations of letters, and their corresponding frequencies
  - Frequencies for initial and ending letters, dictionaries, etc

Analysis of frequencies

- Therefore, substitution methods are vulnerable to an analysis of frequencies
- This analysis reduces drastically the number of possible keys, and it can be obtain “in a reasonable time” with a computer

The Golden Beetle

(Edgar Alan Poe, 1843)

The Golden Beetle

5377+305))6*;4826)47.)47);806*;48+ 8 ♯
60))85;17(;:7* 8 + 83(88)5*+;46
(;88*96*?;8)*7 (;485);5* + 2:*7(;4956*
2(5* − 4)8 ♯ 8*;4069285);)6 + 8)477
;1(79;48081;8:871;48 + 85;4)485 +
528806*81(79;48;(88;4(7?34
;48)47;161;:188;7?;

Table of frequencies for English

LETTER	n/10000	LETTER	n/10000
A	781	N	728
B	128	O	821
C	293	P	215
D	411	Q	14
E	1305	R	664
F	288	S	646
G	139	T	902
H	585	U	277
I	677	V	100
J	23	W	149
K	42	X	30
L	360	Y	151
M	262	Z	9
VOWELS	3861	N+R+S+T+H	3525

Frequent “digrams” in English

DIGRAM	n/10000	DIGRAM	n/10000
TH	315	EN	120
HE	251	ND	118
AN	172	OR	113
IN	169	TO	111
ER	154	NT	110
RE	148	ED	107
ES	145	IS	106
ON	145	AR	101
EA	131	OU	96
TI	128	TE	94
AT	124	OF	94
ST	121	IT	88

Other statistics from English

- Most frequent "trigrams":
  THE, AND, THA, ENT, ION, TIO, FOR, NDE, HAS, NCE, EDT, TIS, OFT, STH, MEN . . .
- Frequent symmetric pairs:
  ER/RE, ES/SE, AN/NA, TI/IT, ON/NO, IN/NI EN/NE, AT/TA . . .
- Most frequent initial letters:
  T A O S H I W C B P F D M R . . .
- Most frequent ending letters:
  E S T D N R O Y . . .

The Golden Beetle (cont.)

- Count the frequency of appearing each letter in our cryptogram:

SYMBOL:	8	;	4	7	)	*	5	6	(	+	1	0	9	2	:	3	?	#	-	.
TIMES :	33	26	19	16	16	13	12	11	10	8	8	6	5	5	4	4	3	2	1	1

The Golden Beetle

* Conclusions:

  - The symbol 8 should be E
    (the frequent digram 88 corresponds to ‘EE’)

The Golden Beetle

* Conclusions:

  - The symbol 8 should be E
    (the frequent digram 88 corresponds to ‘EE’)
  - The trigram ;48 could be THE
    (coincides with the fact that ‘TH’ is the most frequent digram in English, and we can also see on the cryptogram combinations with H such as ‘;46’, ‘;49’ y ‘;40’)

The Golden Beetle

∗ Conclusions:

- The symbol 8 should be E
  (the frequent digram 88 corresponds to ‘EE’)
- The trigram ;48 could be THE
  (coincides with the fact that ‘TH’ is the most frequent digram in English, and we can also see on the cryptogram combinations with H such as ‘;46’, ‘;49’ y ‘;40’)
- Moreover, with THE we can deduce some beginnings and endings of words . . .

The Golden Beetle

- For example, the last combination ‘;48’ in our cryptogram follows like this:

The Golden Beetle

- For example, the last combination ‘;48’ in our cryptogram follows like this:

  ;48;(88;4($\ldots =$ THE T(EETH($\ldots$

The Golden Beetle

- For example, the last combination ‘;48’ in our cryptogram follows like this:

  ;48;(88;4($\ldots =$ THE T(EETH($\ldots$

- Looking at the dictionary we deduce:

$$(= \mathrm{R}$$

The Golden Beetle

∗ Thus we have:

- 8=E
- ;=T
- 4=H
- (=R

and substitute in the cryptogram . . .

The Golden Beetle

5377+305))6∗;4826)47.)47);806∗;48+ 8 ♯
60))85;17(;:7∗ 8 + 83(88)5∗+;46
(;88∗96∗?;8)∗7 (;485);5∗ + 2:∗7(;4956∗
2(5∗ − 4)8 ♯ 8∗;4069285);)6 + 8)477 ;1(79
;48081;8:871;48 + 85;4)485 +
528806∗81(79;48;(88;4(7?34
;48)47;161;:188;7?;

The Golden Beetle

$\mathrm{5377{+}305))6{*}THE26)H7.)H7)TE06{*}THE{+}}$
$\mathrm{E \sharp 60))E5T17RT{:}7{*} E{+}E3REE)5{*}{+}TH6}$
$\mathrm{RTEE{*}96{*}?TE){*}7}$
$\mathrm{RTHE5)T5{*} + 2{:}{*}7RTH956{*} 2R5{*} - H)E \sharp}$
$\mathrm{E{*}TH0692E5)T)6{+}E)H77 T1R79}$
$\mathrm{THE0E1TE{:}E71THE{+}E5TH)HE5 +}$
$\mathrm{52EE06{*}E1R79THETREETHR7?3H THE}$
$\mathrm{)H7T161T{:}1EET7?T}$

The Golden Beetle

$5377+305))6*$THE$26)$H$7.)$H$7)$TE$06*$THE$+$
E $\sharp$ 60))E5T17RT:7$*$ E$+$E3REE)5$*+$TH6
RTEE$*$96$*$?TE)$*$7
RTHE5)T5$* + 2$:$*$7RTH956$*$ 2R5$* -$ H)E $\sharp$
E$*$TH0692E5)T)6$+$E)H77 T1R79
THE0E1TE:E71THE$+$E5TH)HE5 $+$
52EE06$*$E1R79THETREETHR7?3H THE
)H7T161T:1EET7?T

The Golden Beetle

5377+305))6∗THE26)H7.)H7)TE06∗THE+
E ♯ 60))E5T17RT:7∗ E+E3REE)5∗+TH6
RTEE∗96∗?TE)∗7
RTHE5)T5∗ + 2:∗7RTH956∗ 2R5∗ − H)E ♯
E∗TH0692E5)T)6+E)H77 T1R79
THE0E1TE:E71THE+E5TH)HE5 +
52EE06∗E1R79THETREETHR7?3H THE
)H7T161T:1EET7?T

THROUGH

The Golden Beetle

5GOO$+$G05))6$\ast$THE26)HO.)HO)TE06$\ast$THE$+$
E $\sharp$ 60))E5T1ORT:O$\ast$
E$+EGREE)5 \ast +$TH6 RTEE$\ast$96$\ast$UTE)$\ast$
ORTHE5)T5$\ast + 2$:$\ast$ORTH956$\ast$
2R5$\ast - \mathrm{H}$)E $\sharp$ E$\ast$TH0692E5)T)6$+$E)HOO
T1RO9 THE0E1TE:EO1THE$+$E5TH)HE5
$+$ 52EE06$\ast$E1RO9 THE TREE
THROUGH THE) HOT161T:1EETOUT

The Golden Beetle

5GOO+G05))6∗THE26)HO.)HO)TE06∗THE+
E ♯ 60))E5T1ORT:O∗
E+*EGREE*)5 ∗ +TH6 RTEE∗96∗UTE)∗
ORTHE5)T5∗ + 2:∗ORTH956∗
2R5∗ − H)E ♯ E∗TH0692E5)T)6+E)HOO
T1RO9 THE0E1TE:EO1THE+E5TH)HE5
+ 52EE06∗E1RO9 THE TREE
THROUGH THE) HOT161T:1EETOUT

DEGREE

The Golden Beetle

5GOODG05))6∗THE26)HO.)HO)TE06∗THE
DE ♯ 60))E5T1ORT:O∗ E DEGREE)5∗D
TH6 RTEE∗96∗UTE)∗
ORTHE5)T5∗D2:∗ORTH956∗ 2R5∗ — H)E
♯ E∗TH0692E5)T)6DE)HOO T1RO9
THE0E1TE:EO1THE DE5TH)HE5 D
52EE06∗E1RO9 THE TREE THROUGH
THE) HOT161T:1EETOUT

The Golden Beetle

5GOOD
G05))6$*$THE26)HO.)HO)TE06$*$THE DE $\sharp$
60))E5T1ORT:O$*$ E DEGREE)5$*$D TH6
RTEE$*$96$*$UTE)$*$
ORTHE5)T5$*$D2:$*$ORTH956$*$ 2R5$*$ $-$ H)E
$\sharp$ E$*$TH0692E5)T)6DE)HOO T1RO9
THE0E1TE:EO1THE DE5TH)HE5 D
52EE06$*$E1RO9 THE TREE THROUGH
THE) HOT161T:1EETOUT

$5 =$ ‘A’

The Golden Beetle

A GOOD
G0A))6∗THE26)HO.)HO)TE06∗THE DE ♯
60))EAT1ORT:O∗ E DEGREE)A∗D TH6
RTEE∗96∗UTE)∗
ORTHEA)TA∗D2:∗ORTH9A6∗ 2RA∗ − H)E
♯ E∗TH0692EA)T)6DE)HOO T1RO9
THE0E1TE:EO1THE DEATH)HEAD
A2EE06∗E1RO9 THE TREE THROUGH
THE) HOT161T:1EETOUT

The Golden Beetle

A GOOD
G0A))6$*$THE26)HO.)HO)TE06$*$THE DE $\sharp$
60))EAT1ORT:O$*$ E DEGREE)A$*$D TH6
RTEE$*$96$*$UTE)$*$
ORTHEA)TA$*$D2:$*$ORTH9A6$*$ 2RA$*$ $-$ H)E
$\sharp$ E$*$TH0692EA)T)6DE)HOO T1RO9
THE0E1TE:EO1THE DEATH)HEAD
A2EE06$*$E1RO9 THE TREE THROUGH
THE) HOT161T:1EETOUT

$$) = \text{`S'}$$

The Golden Beetle

A GOOD
$G0ASS$6$*$THE26SHO.SHOSTE06$*$THE
DE $\sharp$ 60 SSEAT1ORT:O$*$ E
DEGREESA$*$D TH6 RTEE$*$96$*$UTES$*$
ORTHEASTA$*$D2:$*$ORTH9A6$*$
2RA$* - \mathbb{H}$SE $\sharp$ E$*$TH0692EASTS
6DESHOO T1RO9 THE0E1TE:EO1THE
DEATH'S HEAD A2EE06$*$E1RO9 THE
TREE THROUGH THE
SHOT161T:1EETOUT

The Golden Beetle

A GOOD $G0ASS$6$*$
THE26SHO.SHOSTE0 6$*$THE DE $\sharp$ 60
SSEAT1ORT:O$*$ E DEGREESA$*$D TH6
RTEE$*$96$*$UTES$*$
ORTHEASTA$*$D2:$*$ORTH9A6$*$
2RA$* -$ HSE $\sharp$ E$*$TH0692EASTS
6DESHOO T1RO9 THE0E1TE:EO1THE
DEATH'S HEAD A2EE06$*$E1RO9 THE
TREE THROUGH THE
SHOT161T:1EETOUT

$0 =$ 'L' $6 =$ 'I' $6* =$ 'IF' or 'IN'

The Golden Beetle

A GOOD GLASS IN
THE2ISHO.SHOSTEL IN THE DE ♯ IL'S
SEAT1ORT:ONE DEGREESAND
THIRTEEN9INUTES
NORTHEASTAND2:NORTH9AIN
2RAN–HSE ♯ ENTHLI92EASTS
IDESHOO T1RO9 THELE1TE:EO1THE
DEATH'S HEAD A2EELINE1RO9 THE
TREE THROUGH THE
SHOT1I1T:1EETOUT

...

The Golden Beetle

A GOOD GLASS
IN THE BISHOP'S HOSTEL
IN THE DEVIL'S SEAT
FORTY ONE DEGREES AND
THIRTEEN MINUTES NORTH EAST
AND BY NORTH
MAIN BRANCH SEVENTH LIMB
EAST SIDE
SHOOT FROM THE LEFT EYE OF THE
DEATH'S HEAD
A BEE LINE FROM THE TREE
THROUGH THE SHOT FIFTY FEET OUT

The Golden Beetle

THE END

Entropy

- Each language is an information source, with symbols and their corresponding probabilities (frequencies), from which we can compute its Entropy

Entropy

- Each language is an information source, with symbols and their corresponding probabilities (frequencies), from which we can compute its Entropy
- Several levels of Entropy: considering isolated symbols, digrams, trigrams . . .

Entropy

- Each language is an information source, with symbols and their corresponding probabilities (frequencies), from which we can compute its Entropy
- Several levels of Entropy: considering isolated symbols, digrams, trigrams . . .
- Eavesdropper: calculate the Entropies of the cryptogram and check if they are close to the suspected language; then we may assume that a substitution method has been used, and proceed with an analysis of frequencies . . .

Transposition Methods

- Consist of “shuffle” (mix) the symbols of the plaintext
- Keep the language frequencies, but destroy the morphology and grammar structures . . .
- Origin: the Scitala (Esparta, old Greek, century V b.C.)
  - Two identical sticks (the key, i.e. the same thickness)
  - One rolls a strip on the stick and writes
  - When one unrolls the strip nothing is readable . . .
  - . . . until it is enrolled again on the twin stick

Transposition Methods

$*$ From a mathematical point of view . . .

- Divide the message into blocks of fixed length k (key)
  $\rightarrow$ this is the width of the 'stick'
- Permute the symbols of each block according to a fixed permutation (key)
  $\rightarrow$ this is 'unroll' the strip
- Invert the (secret) permutation to read the message
  $\rightarrow$ this is 'roll' the strip again

Linear Cipher

$*$ Destroys the frequencies, and thus is more secure than transposition methods, but it is vulnerable to more sophisticated attacks

- Divide the message into blocks of a fixed length k
- Multiply each block by a fixed invertible matrix A
- Keep k and A secret
- In order to decipher you need the inverse A^{-1}

…

Vernam Cipher (1917)

- Consists of add bit-wise to the message (XOR) a random key with the same length as the message
- Shannon proved in 1949 that this system is “completely secure” provided these three conditions are satisfied:

1. The key must be a really random sequence of bits
2. The key must have the same size as the message
3. The key must be used only once (one–time pad)

Vernam Cipher (1917)

- The key destroys every internal structure of the message (random key), so that the cryptograph gives no information about the original message
- If M is the original message and C is the cryptogram, Shannon proved

$$I(M|C) = 0$$

- That is, from C all the possible messages M are equiprobable . . .

Vernam Cipher (1917)

- Because of the consequences of this result, Shannon named the assumptions of his theorem as perfect secret conditions
- Additional (implicit) assumption:

  “The cryptoanalyst only has access to the ciphertext”
- Thus, it is “intrinsecally” impossible decipher, not even with infinite power of computing capabilities (quantum computers included)

Vernam Cipher (1917)

- Because of the consequences of this result, Shannon named the assumptions of his theorem as perfect secret conditions
- Additional (implicit) assumption:

  “The cryptoanalyst only has access to the ciphertext”
- Thus, it is “intrinsecally” impossible decipher, not even with infinite power of computing capabilities (quantum computers included)
- Is the problem over?

Vernam Cipher (1917)

⋆ Limitations:

- Do random sequences really exist?

  Quantum Computing

  In practice: just pseudorandom sequences

- The size of the key is a problem (generate, store and share)

  Quantum Cryptography

- It only works for private key systems, and not for public key system (networks of users)

One–time Pad

- If we use the same key more than once, the security drops drastically

One–time Pad

- If we use the same key more than once, the security drops drastically

Idea of the proof:

$$
\begin{aligned}
&(C = M \oplus K) \;\wedge\; (C' = M' \oplus K) \\
&\Rightarrow C \oplus M = C' \oplus M' \\
&\Rightarrow C \oplus C' = M \oplus M'
\end{aligned}
$$

$\ast$ That is: the sum of the cryptograms is vulnerable to an analysis of frequencies . . .

Types of Security

- Unconditional Security: the system is safe against an attacker with unlimited time and computational resources (Vernam Cipher)

Types of Security

- Unconditional Security: the system is safe against an attacker with unlimited time and computational resources (Vernam Cipher)
- Computational Security: the system is safe against an attacker with limited time and computational resources (RSA, based on prime numbers)

Types of Security

- Unconditional Security: the system is safe against an attacker with unlimited time and computational resources (Vernam Cipher)
- Computational Security: the system is safe against an attacker with limited time and computational resources (RSA, based on prime numbers)
- Probable Security: nobody has proven it is secure, but in practical purposes it works (DES, based on certain “black boxes”)

Types of Security

- Unconditional Security: the system is safe against an attacker with unlimited time and computational resources (Vernam Cipher)
- Computational Security: the system is safe against an attacker with limited time and computational resources
(RSA, based on prime numbers)
- Probable Security: nobody has proven it is secure, but in practical purposes it works (DES, based on certain “black boxes”)
- Conditional Security: the system is safe under further assumptions about the limitations of the attacker

Types of Attacks

- Active
- Passive

Types of Attacks

- Active (The man in the middle)
  - Impersonation
    (pretend to be someone else)
  - Substitute the intercepted message by another one
  - Intentionally produce errors in the cryptogram (force a new transmission)
- Passive

Types of Attacks

- Active
- Passive
  - “Brute Force”:
    try with all possible keys . . .
  - Known ciphertext
    (Variant: obtain several cryptograms of the same M with different keys)
  - Known plaintext
    (Linear cipher: vulnerable to this attack)
  - Chosen plaintext
  - Other variants . . .

Attacks and Security

∗ Precautions:

- Do not change the key when repeating a transmission
- Do not cipher public information
- Chose random enough keys (dictionaries)
- Change frequently the key

Types of Cipher

- Stream Cipher: the message is ciphered and transmitted at the same time, symbol by symbol
  (interesting in communications; f.e. Vernam Cipher)
- Block Cipher: the message is divided into blocks and these blocks are ciphered and sent separately
  (file protection in a PC; f.e. DES, RSA, etc)

Contents

- Introduction to Cryptology
- Private Key and Public Key
- Practical Applications

Private Key and Public Key

Private Key and Public Key

- Private Key cryptosystems: there is only a transmitter and a receiver (that can also be the same person), who share keys to cipher and decipher (normally both keys are the same), which must be kept secret, (this is the case, for example, in military communications or when encrypting files in a PC)
- Public Key cryptosystems: there is a set of users connected by a network, each of one has a public key (published and known) that can be used by the other users to send him a message, and a private key (kept secret) to read the messages addressed to him

Private Key systems

- Vernam Cipher (random sequences)
- DES (Data Encryption Standard, patented by IBM)
  Complicated combinación os substitutions, transpositions and some non-linear “black boxes”
  - Probable security (currently not advisable in practice, because of the current power of computers)
  - It is an involution (the same algorithm encrypts and decrypts)
- IDEA (International Data Encryption Algorithm)
  Based on the mixture of incompatible arithmetic operations, in different algebraic groups . . .
  - Accepts several “rounds” (double cipher, triple, etc)
  - Security: immune to Differential Cryptoanalysis
    (from 4 rounds; in practice 8 rounds are used)
  - It is not an involution

ssh / crypt

- `ssh`: is a program to establish secure remote connections
- `crypt`: it was an on-line command in old Unix systems to encrypt files
  - Sintax:

    ```
    crypt key <file> encrypted_file
    ```

  - It is an involution if the same key is used (based on the DES)
  - Compatible with the option '–x' of the editor `vi`

    ```
    vi -x file
    ```

Public Key systems

- They are asymmetric:
  - Encrypt must be done fast by anyone
  - Decrypt must be done (fast) only by the right user, and it should be impossible (computationally time-consuming) for any other user (computational security)
- They are based on one–way functions (no return) and trapdoor functions

One–way functions

- They are invertible functions such that:
  - The direct image can be efficiently computed, but . . .
  - . . . whose inverse image is time-consuming to compute
- They lie on hard mathematical problems (NP problems, with exponential complexity)

Trapdoor functions

- They are one–way functions whose inverse image can be efficiently computed provided one knows a suitable datum called certificate (usually related to the private key)
- The basic idea is to use the direct function to encrypt and the inverse to decrypt, and the right user can skip the NP problem with the aid of his own private key

Diffie–Hellman conditions

1. Computing and distributing keys (public and private) must be efficient
2. Ciphering must must be efficient provided one has the public key
3. Deciphering must also be efficient provided one has the private key
4. Computing the private key from either the public key or the cryptogram must be computationally time-consuming (at least in average, for well-chosen keys)
5. Finally, obtaining the plaintext from the ciphertext and the public key must also be computationally time-consuming

NP problems for Cryptography (I)

- Factorization of integers:

  For a given $n = p \cdot q$, factor n to find the primes p and q

  - One–way function: multiply p and q to get n
  - The inverse function: factor a given n to find p and q
  - Trapdoor function: keep one factor p as certificate (then $q = n/p$)

- The cryptosystem RSA (Rivest, Shamir and Adleman, 1978) is based on this NP–problem

NP problems for Cryptography (II)

- Discrete Logarithm (modular):
  For a, n and m given, find x such that $a^x \equiv m \pmod{n}$ (if possible)
  - One–way function: modular exponentiation
    `m:=a**x mod n`
  - Inverse function: the discrete logarithm of m in the basis a, modulo n (the exponent x)
  - Trapdoor function: do not delete x once computed m ...
- There exist several cryptosystems based on the discrete logarithm ...

ElGamal cryptosystem

Based on the Discrete Logarithm:

- Let G be a finite cyclic group with n elements, generated by an element g, i.e.

$$G = \{1, g, g^2, \ldots, g^{n-1}\}$$

- Discrete Logarithm problem:

  For a given $h \in G$, find k such that $g^k = h$

- The best known computation times to solve this problem are of “subexponential” type

ElGamal cryptosystem

Choice of the group:

- Let $p >> 0$ be a (large enough) prime, and consider the multiplicative group (modulo p)

$$G = Z_p^* := \{1, 2, \ldots, p-1\}$$

  with cardinality $n = p - 1$
- The possible messages are the elements of G
- Fix a generator g of the group G

ElGamal cryptosystem

Choice of the keys:

- Each user A (resp. B) chooses and element a (resp. b) in G as private key, which is kept secret ...
- ... and computes the public key $k_a = g^a \ (mod \ p)$ (resp. $k_b = g^b \ (mod \ p)$)

ElGamal cryptosystem

Encryption – If A wants to send the message m to B the procedure is as follows:

1. A chooses $r \in G$ at random and computes $g^r \ (mod \ p)$
2. Using the public key k_b of B, A computes

$$m \cdot (k_b)^r = mg^{br} \ (mod \ p)$$

3. A sends to B the couple (g^r, mg^{br})

ElGamal cryptosystem

Decryption – In order to read B his received message, he proceeds as follows:

- Using his own private key b, B computes

$$(g^r)^b = g^{br} \ (mod \ p)$$

- Finally, B obtains the message by computing

$$m = \frac{mg^{br}}{g^{br}} \ (mod \ p)$$

$\ast$ If any other user C wished to obtain m, he should calculate b from $k_b = g^b$ (solve a discrete logarithm!)

ElGamal cryptosystem

Other applications:

- Digital Signature
- Authentication of messages
- Distribution os symmetric keys

RSA

- Each user i has to choose a couple of primes $p_i, q_i >> 0$
- Calculate $n_i = p_i q_i$
- Compute the Euler Phi function

$$\varphi(n_i) := (p_i - 1) \cdot (q_i - 1)$$

- Choose at random $0 < e_i < \varphi(n_i)$ such that $\gcd(e_i, \varphi(n_i)) = 1$
- Compute the modular inverse $d_i \equiv e_i^{-1} \ (mod \ \ \varphi(n_i))$, that is $e_i \cdot d_i \mod \varphi(n_i) = 1$

RSA

Keys:

- Public: (n_i, e_i)
- Private: d_i

$\star$ Trapdoor: for computing d_i one needs $\varphi(n_i)$, and hence one has to factorize n_i ...

RSA

- Encryption:

$$M \mapsto C \equiv M^{e_i} \ (mod \ \ n_i)$$

- Decryption:

$$C \mapsto C^{d_i} \equiv M^{e_i d_i} \ (mod \ \ n_i)$$

Theorem:

$$M^{e_i d_i} \equiv M \ (mod \ \ n_i)$$

RSA

Computational issues:

- Testing if an integer is prime or not (at least probabilistically) is efficient . . .
- . . . but NOT factorizing !
- The typical operations from (modular) arithmetic (gcd, modular exponentiation and inverses, etc) are efficient

RSA

Precautions:

- The primes p_i and q_i must not be close to $\sqrt{n_i}$, since you can easily factorize otherwise (Fermat method)
- The numbers $p_i - 1$ and $q_i - 1$ must not have all the prime factors "small" (Pollard $p - 1$ method)
- Similarly, the numbers $p_i + 1$ and $q_i + 1$ must not have all the prime factors "small" ($p + 1$ method)

RSA

Risks:

- After ciphering you may get $C = M$
  This happens the following number of times:

$$(1 + gcd(e_i - 1, p_i - 1)) \cdot (1 + gcd(e_i - 1, q_i - 1))$$

- Thus, you should get both gcd's "small"
  (minimize the risk of no ciphering)
- Apart from extreme case
  (to be detected and avoided)
  the probability is not sensible

Contents

- Introduction to Cryptology
- Private Key and Public Key
- Practical Applications

Practical Applications

Cryptographical Protocols

$*$ Private Key:

- Authentication
- Digital Signature
- Identification

Cryptographic Protocols

∗ Private Key:

- Authentication

  Goal: the receiver can check if the received message has been modified or not by a third part

  - f.e.: send both the plaintext and the ciphertext (loss of confidentiality, and risk of attack)
  - You detect if a spy has manipulated the plaintext
  - There are better methods . . . by using public key systems

- Digital Signature
- Identification

Cryptographic Protocols

∗ Private Key:

- Authentication
- Digital Signature

  It is an engagement for the signer to maintain his word, and prevents from modifications of the content by the receiver

  1. Implicit: is part of the message itself
  2. Explicit: is added to the message as a separate mark
  3. Private: only the receiver can identify the signer
  4. Public: anyone can identify the signer
  5. Revocable: the signer can deny *a posteriori* that that is his signature
  6. Irrevocable: the receiver can prove that the transmitter has signed the message

- Identification

Cryptographic Protocols

$*$ Private Key:

- Authentication
- Digital Signature
- Identification

  Aim: the receiver wants to check if the transmitter is really who he claims to be

  - The usual magnetic cards are subject to fraude by duplication, alteration or falsification
  - Because of that, they tend to be substituted by the so-called intelligent cards, having a chip with memory

Cryptographic Protocols

∗ Public Key:

- Authentication / Identification / Digital Signature

  The public key increases the confidentiality,
  at the expense of speed in the protocols

- Secret Sharing / Exchange / Sale
- Proof of zero-knowledge

  Prove that one has a secret without revealing its content

- Signing a contract
- Electoral Scheme

  Counting votes of authorized individuals exactly once,
  so that the content of each vote remains secret

- Mail with Acknowledgement of Receipt

Digital Signature (ElGamal)

The user A wants to sign a message m

1. Generate h at random such that $gcd(h, \varphi(n)) = 1$
2. Compute $r \equiv g^h \ (mod \ n)$
3. Solve the congruence

$$m \equiv a \cdot r + h \cdot s \ (mod \ \varphi(n))$$

4. The digital signature of m is the couple (r, s)

Digital Signature (ElGamal)

The user B wants now to check the signature of A

1. Compute $r^s \equiv g^{hs} \ (mod \ n)$
2. Compute $g^{ar} \ (mod \ n)$
3. Compute $x \equiv g^{hs} \cdot g^{ar} \ (mod \ n)$
4. Check whether $x \equiv g^m \ (mod \ n)$ or not

Practical Applications

PRIVATE KEY

- **Hardware**:
  - Firmware cards for PC (protection of software)
  - Ciphers for transmission on-line (communications)
  - Intelligent Cards and Cryptographic Cards
  - PIN keyboards (cash dispensers)
- **Software**:
  - Watermark (copyright protection)
  - Programs to encrypt files
  - Programs for network access (login in terminals)
  - Integral security packages

Practical Applications

PUBLIC KEY

- Communications Networks:
  - Security systems for phone networks
  - Broadcasting (digital TV with pay-per-view)
  - Military security and espionage
  - Electronic Voting
- Information Systems:
  - Security systems for computer networks
  - e-Mail PGP
  - Security in Databases
  - Secure electronic transactions on-line https

Network/Computer Security

∗ Cryptography is only a part . . .

- Secure connections (POP, cookies, etc)
- Management of networks, sessions, keys, etc
- Physical security against “hackers” (firewalls, opened ports and services)
- Policy of access and permissions as `root`
  Viruses: one (more) weakness of You-Know-What-OS
- . . .

Integral Plan for Network/Computer Security

- Redundance and BackUp
- Separate responsibilities
- Access restrictions
- Analysis of Software programs
- Cryptography
  - Encrypt files
  - Cipher communications with session keys (one–time pad)
  - User access with I–Card + PIN
  - PIN's must be stored encrypted!
- Management/Policy of (random) keys
- Security must be independent on the type of terminal
- Cipher/decipher speed higher than the transmission speed

Bibliography

- D. Welsh: "Codes and Cryptography", *Ed. Clarendon Press* (1988).
- K. Jamsa, L. Klander: "Hacker Proof: The ultimate guide to network security", *Ed. Thomson Delmar Learning* (2002).
- N. Koblitz: "A course in Number Theory and Cryptography", *Springer–Verlag* (1994).
- G.J. Simmons et al.: "Contemporary Cryptology", *IEEE Press* (1992).
- B. Schneier: "Applied Cryptography", *Ed. John Wiley & Sons, Inc.* (1994).

Questions?