# A Brief History of Cryptography

***Dr. Levente Buttyán***
Laboratory of Cryptography and System Security (CrySyS)
Department of Telecommunications
Budapest University of Technology and Economics
buttyan@crysys.hu

***www.crysys.hu***

## Communications security

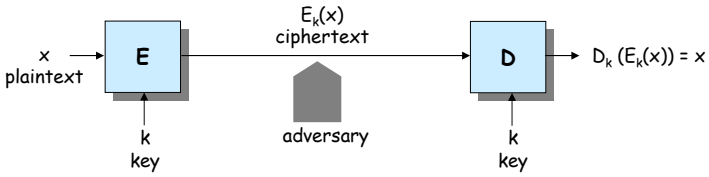
- main security services
  - confidentiality
  - integrity
  - authentication
  - non-repudiation
  - access control
- three pillars of security
  - physical protection
  - procedures (policies, educations, etc.)
  - ***cryptography (algorithmic tools)***

## Evolution of the meaning of cryptography

- until the second half of the 20th century
  - exclusively military (and diplomacy) applications
  - cryptography = confidentiality (encryption)
- from the second half of the 20th century
  - business applications (mainly in the banking sector)
  - cryptography = confidentiality + integrity, authenticity, and non-repudiation
- from the end of the 20th century
  - cryptography is part of everyday life
    - Web security (SSL protocol)
    - GSM security



## Classical model of (symmetric-key) encryption




- goal of the adversary
  - obtain plaintexts in a systematic manner
  - obtain the key
- the Kerkchoff-principle
  - the adversary knows the details of the encoding and decoding algorithms
  - the adversary does not know the key

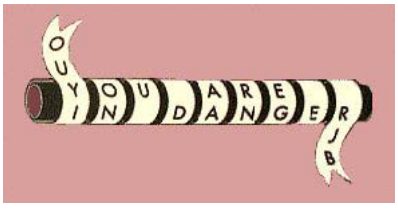
## Historical examples

- „already the ancient Greeks ...” – the skytale of Sparta
- „veni, vidi, vici” – the Caesar cipher
- the „unbreakable” cipher – Vigenère code
- the first cipher machine – Enigma



## Skytale

- used by the Spartans in ~400 B.C.
- transposes the characters of the message
- key = diameter of the baton
  practical size of the key space is small



## The Caesar cipher

- based on character substitution
- encoding rule
  - plain: 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
  - cipher: 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

  RETURN TO ROME → UHWXUA WR URPH
- key = shift of the alphabet (3 in case of Caesar)
  size of the key space = 26-1 = 25



## Mono-alphabetic substitution

- generalization of the Caesar cipher
- code alphabet = random permutation of the alphabet
  - plain: 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
  - cipher: T Q L R B W J G F Y N C P E K S D U V I X M Z A H O
- key = the permutation that is used
  size of the key sapce = 26! ~ $1.56*2^{88}$

## Large numbers

time until next ice age..... $2^{39}$ seconds
time until the sun goes nova..... $2^{55}$ seconds
age of the planet..... $2^{55}$ seconds
age of the Universe..... $2^{59}$ seconds

number of atoms in the planet..... $2^{170}$
number of atoms in the sun..... $2^{190}$
number of atoms in the galaxy..... $2^{223}$
number of atoms in the Universe ..... $2^{265}$
(dark matter excluded)

volume of the universe..... $2^{280}$ $cm^3$

(source: Schneier, Applied Cryptography, 2nd ed., Wiley 1996)



## Breaking the mono-alphabetic substitution cipher

- each language has characteristic letter statistics
  - in an average text, letters do not occur with uniform frequency
  - some letters occur more frequently than others
    e.g., in English:
    e – 12.7%
    t – 9.1%
  - some letters occur less frequently
    e.g., in English:
    z – 0.1%
    j – 0.2%
- in case of mono-alphabetic substitution, the ciphertext preserves the letter frequency of the plaintext (!)
  - e.g., most frequent character must be the encoding of „e" or „t"
  - after identifying a few mappings, the rest of the code alphabet is usually easy to determine

## Poly-alphabetic substitution – the Vigenère code

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

encoding:

```
      key: RELAT IONSR ELATI ONSRE LATIO NSREL
plaintext: TOBEO RNOTT OBETH ATIST HEQUE STION
ciphertext: KSMEH ZBBLK SMEMP OGAJX SEJCS FLZSY
```

decoding:

```
      key: RELAT IONSR ELATI ONSRE LATIO NSREL
ciphertext: KSMEH ZBBLK SMEMP OGAJX SEJCS FLZSY
 plaintext: TOBEO RNOTT OBETH ATIST HEQUE STION
```

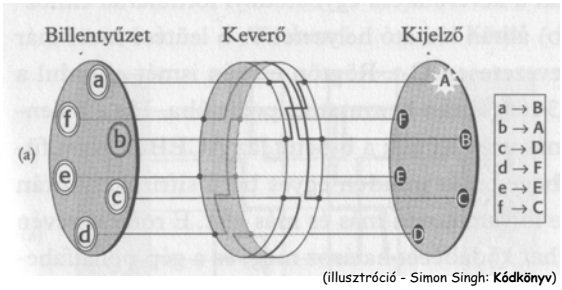


## Enigma

- first electro-mechanical ciphering machine
- patented by Arthur Scherbius in 1918
- introduced in the German Army in 1926

## Operating principle of Enigma

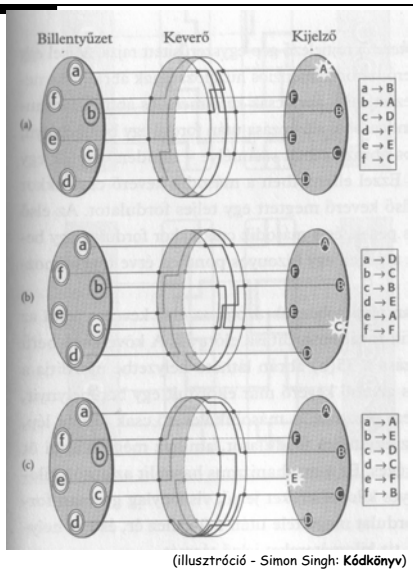
- three main parts:
  - keyboard – for typing in plaintexts and ciphertexts
  - display panel – for displaying plaintexts and ciphertexts
  - mixing unit – to produce ciphertext from plaintext and vice versa
- the soul of Enigma is the rotor




(illusztráció - Simon Singh: **Kódkönyv**)

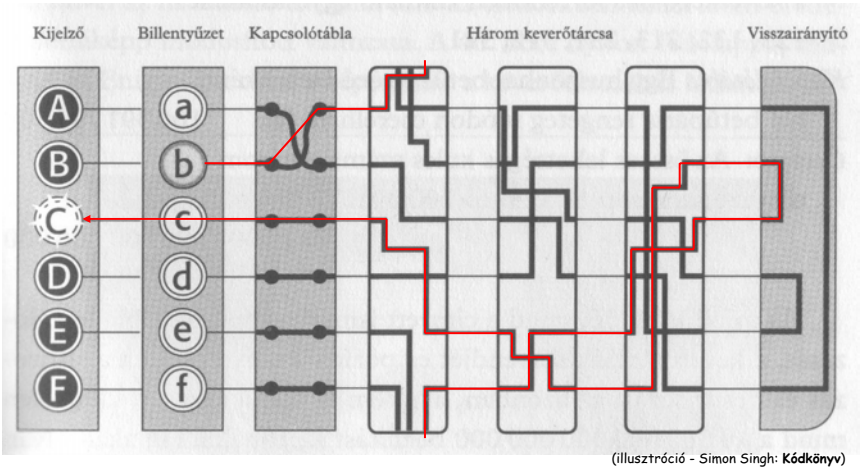


## Operating principle of Enigma




(illusztráció - Simon Singh: **Kódkönyv**)

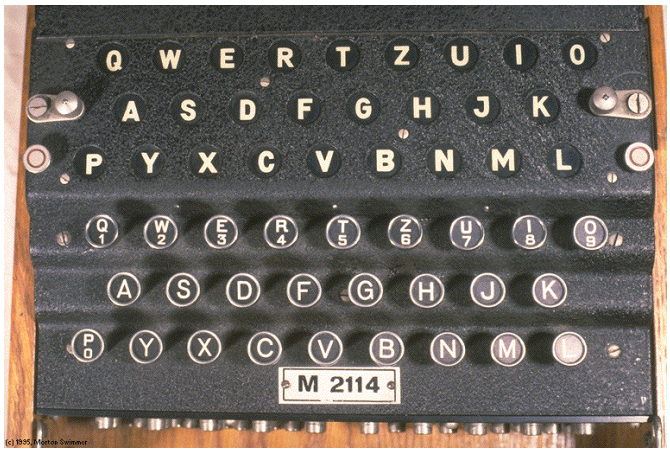
## Operating principle of Enigma



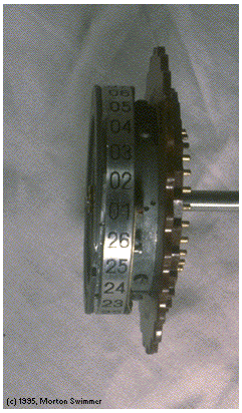

(illusztráció - Simon Singh: **Kódkönyv**)

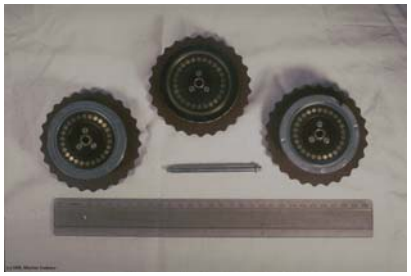


## Enigma – keyboard and display panel



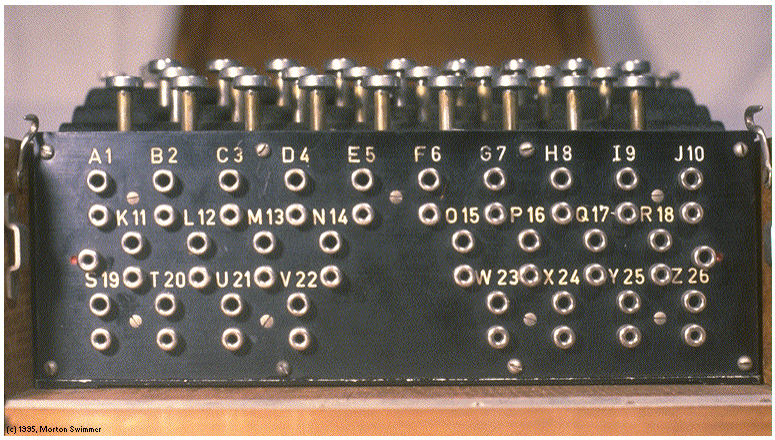
## Enigma – rotors





## Enigma – switching board



## Usage of Enigma

- base setting is determined by the
  - setting of the switching board (pl: A/L – P/R – T/D – B/W – K/F – O/Y)
  - order of the rotors (pl: II – III - I)
  - initial positions of the rotors (pl: Q – C - W)

  (size of the key space = 100391791500 x 6 x $26^3$ ~ $10^{16}$ ~ $2^{53}$)

- plaintext is typed on the keyboard and ciphertext characters are read from the display panel



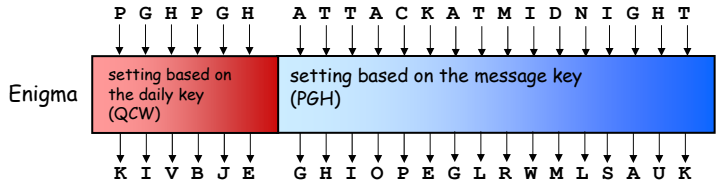
## Breaking the Enigma

Marian Rejewski
Polish mathematician

- Hans-Thilo Schmidt, German spy, sells the manuals of the military Enigma to the French intelligence service [November 8, 1931]
- based on the information obtained from the manuals, the Allies build a copy of the Enigma
- they believe that the Enigma is unbreakable and give the copy to the Polish
- the Polish Biuro Szyfrów hired 20 mathematicians from the University of Poznan, and selected the best three, among them Marian Rejewskit (23)

## Breaking the Enigma

- the Germans used a two level key hierarchy
  - daily key – valid for one day, used to encrypt message keys
  - message key – used to encrypt a single message, changed for every message
    - order of the rotors and setting of the switching board is the same as for the daily key, initial positions of the rotors are supposed to be randomly chosen
- for reliability reasons, the message key was encrypted twice with the daily key
- example:






## Breaking the Enigma

- Rejewski realized that weak point is the repetition of the message key
- he developed a method to break the Enigma based on this observation
- he automated the method
  - built a machine using several copies of the Enigma
  - due to the ticking noise generated by the machine, it was called *bomb*
- thanks to Rejewski, the Plosish intelligence service could routinely break the German communications from 1933

## Breaking the Enigma

- in 1938, the Germans strengthened the Enigma
  - 2 new rotors (number of possible ordering increased from 6 to 60)
  - number character swappings on the switching board was increased from 6 to 10
  - size of the key space increased to $1.59*10^{20}$
- the Germans prepare to invade Poland
- the Poles decide to reveal their knowledge to the Allies [July, 1939]
- the documentation of the bombs is transported to London [August 16, 1939]
- Germany invades Poland [September 1, 1939]



## Breaking the Enigma

Bletchley Park
[August 1939]

## Breaking the enigma

- the British develop the bombs further, and invent new breaking techniques
  - cilly
    - the German Enigma operators often used very simple message keys, such as those consisting of neighboring characters on the keyboard (e.g., QWE, BNM)
    - an Enigma operator always used the initials of his girl friend (C.I.L.)
    - hence, such a weak key was called *cilly* (~silly)
  - requirements on the order of the rotors
    - the Germans changed the order of the rotors every day (daily key)
    - a given rotor was not allowed to stay in the same position for two consecutive days
    - e.g., after I-II-V, the order III-II-IV was not allowed
    - in fact, this requirement decreased the number of possible orders to be tested by the Allies
    - similarly, neighboring characters were not allowed to be swapped on the switching board



## Breaking the Enigma

Alan Turing
British mathematician

- Alan Turing joined Bletchley Park in September 1939
  - he was 27, but already known from his Turing machine
- his task was to find a new method to break the Enigma, which does not take advantage of the repetition of the message key at the beginning of the message
- he solved the problem, the new method exploited the fact that some messages contained known words at known positions
  - German messages were very well structured
  - every evening at 6pm, they sent a weather forecast which contained the word “wetter” in a known position
- based on Turing’s work, the British built new *bombs* (machines) called Victory and Agnus Dei [March-August 1940]
- the Germans changed their key exchange method [May 1940]

## Breaking the Enigma

- Bletchley Park played a very important role in the victory of the Allies
- according to some historians estimates, World War II could have lasted until 1948 without breaking the Enigma
- after the war, the bombs were disassembled, and all related documents were destroyed
- the cryptographers of Bletchley Park returned to their normal civil life
- Alan Turing committed suicide on June 7, 1954.



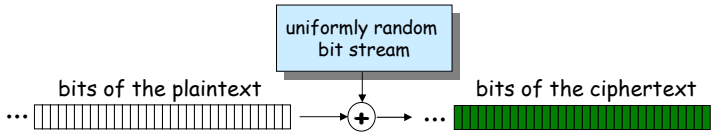
## A truly unbreakable cipher: the One-Time Pad

- mod 2 addition $\oplus$: $a \oplus b = (a + b) \bmod 2$

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

- properties of mod 2 addition:
  1. $x \oplus x = 0$
  2. $x \oplus 0 = x$

## Operation of the One-Time Pad




- encoding
  - $y_i = x_i \oplus k_i$
  - where $x_i$ is the i-th bit of the plaintext, and $y_i$ is the i-th bit of the ciphertext
  - $k_i$ is the i-th bit of the uniformly random bit stream
- decoding
  - $x_i = y_i \oplus k_i = x_i \oplus k_i \oplus k_i = x_i$



## Perfectness of the One-Time Pad

- assume that the adversary observes ciphertext Y
- as all possible keys are equally likely to be the key that has been used to produce Y, all possible plaintext are equally likely to be the message
- this intuition was formalized by Claude Shannon [1949]

$$I(X; Y) = H(X) - H(X|Y) = 0$$

- Shannon also gave necessary conditions for a cipher to be perfect:

$$H(K) \geq H(X)$$

practically this means that the key must be as large as the compressed message

## Modern cryptography

- the One-Time Pad provides unconditional security, but it is impractical
- in practice, we are happy with a cipher that provides conditional security
  - the cipher cannot be broken with less than a given amount of resources (computing power)
- practical ciphers are not even proven to be conditionally secure
- well-known examples:
  - DES (Data Encryption Standard)
  - RSA (Rivest-Shamir-Adleman)



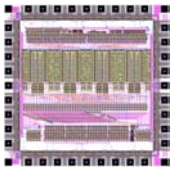
## Shannon's approach

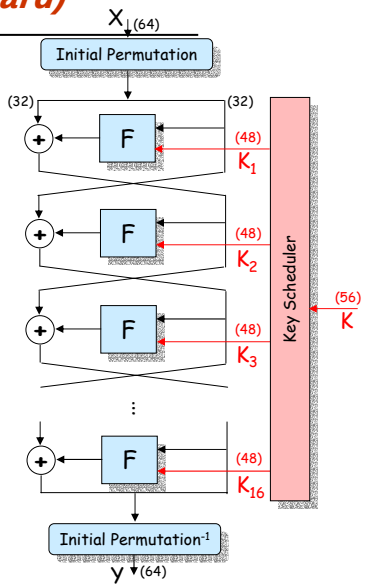
Claude E. Shannon

- build a complex cipher by repeatedly using many individually weak transformations
  - small substitutions
  - bit permutations
  - simple logical and arithmetic operations
  - ...
- none of these simple transformations would be sufficient alone, but together they may provide strong security

## DES (Data Encryption Standard)

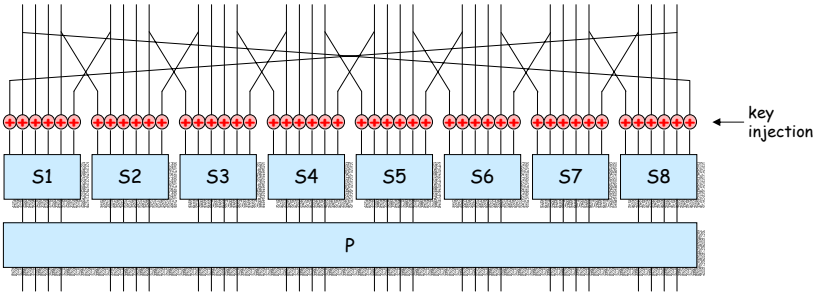
- designed by IBM in the 70's (originally called Lucifer)
- symmetric-key block cipher
- features:
  - Feistel structure (encoding and decoding scheme is the same)
  - number of rounds: 16
  - input size: 64 bits
  - output size: 64 bits
  - key size: 56 bits








## DES round function




- Si – substitution box (S-box)
- P – permutation box (P-box)

## Linear and differential cryptanalysis of DES

- linear cryptanalysis (LC)
  - linear cryptanalysis is the most powerful attack against DES to date
  - requires an enormous number (~$2^{43}$) known plaintext-ciphertext pairs → infeasible in practical environments
  - could work in a ciphertext only model if plaintexts are redundant (e.g., contain parity bits)
- differential cryptanalysis (DC)
  - most general cryptanalytic tool to date against iterated block ciphers (including DES, FEAL, IDEA)
  - primarily a chosen-plaintext attack
  - in case of DES, it requires ~$2^{47}$ chosen plaintext-ciphertext pairs → infeasible in practical environments
- DES was optimized against DC when it was designed
- it can, however, be improved with respect to LC (apparently the designers of DES was not aware of this attack at that time)



## The birth of asymmetric-key cryptography

- Whitfield Diffie and Martin Hellman:
  New Directions in Cryptography
  IEEE Transactions on Information Theory, 1976

Raplh Merkle, Martin Hellman, és Whitfield Diffie

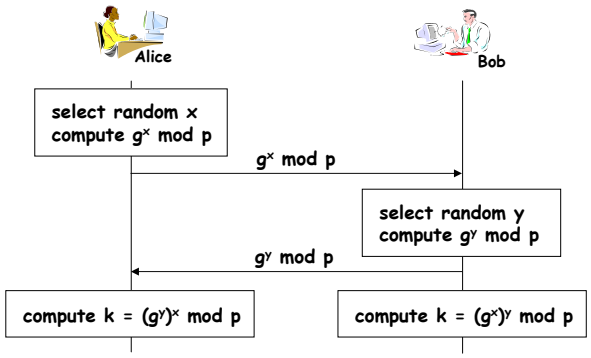
## Discrete logarithm as a one-way function

- let p be a prime
- $Z_p^* = \{1, 2, ..., p-1\}$
- let g be a generator of $Z_p^*$
  - $Z_p^* = \{g^0 \bmod p, g^1 \bmod p, ..., g^{p-2} \bmod p\}$
- $f(x) = g^x \bmod p$
  - given x, it is easy to compute f(x)
  - given y = f(x), it is difficult to compute x (discrete log problem)
- example:
  - $p = 7$, $Z_7^* = \{1, 2, 3, 4, 5, 6\}$, $g = 3$
  - $3^0 = 1$, $3^1 = 3$, $3^2 = 9 = 2$, $3^3 = 27 = 6$, $3^4 = 81 = 4$, $3^5 = 243 = 5$
- now, try to compute x from $453^x \bmod 21\,997 = 5789$

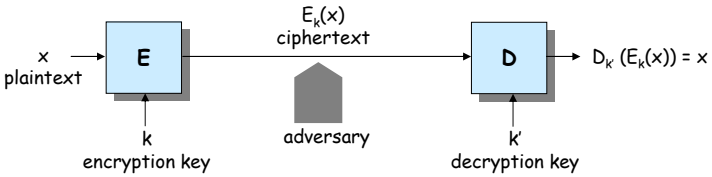


## The Diffie-Hellman key agreement protocol

**assumption:** prime p and generator g of $Z_p^* = \{1, 2, ..., p-1\}$ are publicly known



## Model of asymmetric-key cryptography




- asymmetric key means
  - computing k' from k is hard (NP)
  - k can be made public (hence the name public key cryptography)
- authenticity and integrity of the public key must still be ensured !
  - widely used approach is based on public key certificates and the infrastructure for managing them (PKI)



## The RSA cryptosystem

- Ronald Rivest, Adi Shamir, és Leonard Adleman,
  A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, 1978


Ronald Rivest


Adi Shamir


Leonard Adleman

## Operation of the RSA cryptosystem

- key generation
  - select p, q large primes (about 500 bits each)
  - $n = pq$, $\phi(n) = (p-1)(q-1)$
  - select e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$
  - compute d such that $ed \bmod \phi(n) = 1$ (this is easy if $\phi(n)$ is known)
  - the public key is (e, n)
  - the private key is d

- encryption
  - represent the message as an integer m in [0, n-1]
  - compute $c = m^e \bmod n$

- decryption
  - compute $m = c^d \bmod n$



## A toy example

- let p = 73, q = 151
- n = 73*151 = 11023
- $\phi(n) = 72*150 = 10800$
- let e = 11 [ lnko(11, 10800) = 1, as $10800 = 2^4*3*5^2*9$ ]
- d can be computed with the Euclidean algorithm: d = 5891

- assume that m = 17
- $c = 17^{11} \bmod 11023 = 1782$
- $m = 1782^{5891} \bmod 11023$
- computing is done using the “square and multiply” approach:

$5891 = 2^0 + 2^1 + 2^8 + 2^9 + 2^{10} + 2^{12}$

$1782^{5891} = 1782^{2^{12}} * 1782^{2^{10}} * 1782^{2^9} * 1782^{2^8} * 1782^{2^2} * 1782 =$

$= (\ldots ((1782^2)^2)^2 \ldots)^2 * (\ldots ((1782^2)^2)^2 \ldots)^2 * \ldots$

$= 17 \pmod{11023}$

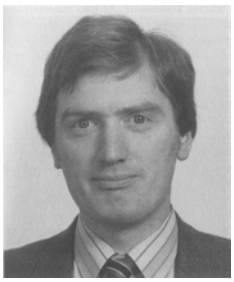
## Security of RSA

- the problem of computing d from (e, n) is computationally equivalent to the problem of factoring n
  - if one can factor n, then he can easily compute d
  - if one can compute d, then he can efficiently factor n
- the problem of computing m from c and (e, n) (RSA problem) is believed to be computationally equivalent to factoring
  - if one can factor n, then he can easily compute m from c and (e, n)
  - there's no formal proof for the other direction
- given the latest progress in developing algorithms for factoring, the size of the modulus should at least be 1024 bits



## The secret story of public key cryptography

James Ellis



Clifford Cocks

Malcolm Williamson

## The secret story of public key cryptography

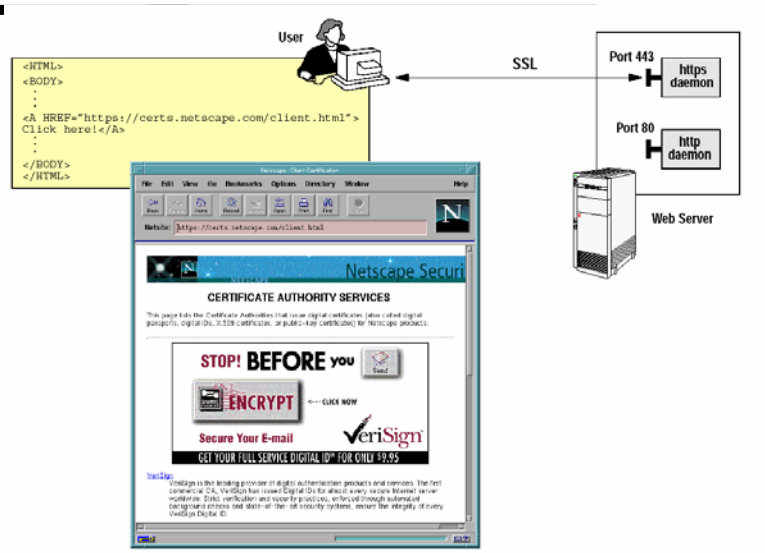
- Ellis, Cocks, and Williamson worked for the British intelligence service
- in 1969, Ellis developed the idea of public key cryptography
  - he had the model, and
  - he knew that some kind of a one-way function is needed that can be inverted only if some secret information is available
- in 1973, Cocks invented the RSA algorithm
  - Cocks was exposed to the problem, and solved it in half an hour
  - he was a mathematician working on number theory, so he immediately though of factoring as a hard problem that can serve as a basis for the one-way function needed
- in 1974, Williamson (a friend of Cocks) discovered the Diffie-Hellman key agreement protocol
- by 1975, Ellis, Cocks, és Williamson developed all the fundamental elements of public key cryptography, but they had to keep them secret (until 1997)



## The SSL protocol

- developed by Netscape in the mid 90's
- makes it possible to establish a secured TCP connection between two remote computers
- typical application is securing web transactions (communications between a browser and a web server)
- widely used in practice (SSL 3.0)
- standardized by the IETF under the name of TLS (Transport Layer Security)

## Operation of SSL






## Operation of SSL (simplified)