# COSC130

## Fundamentals of Cybersecurity and Privacy

LECTURE 9: INTRODUCTION TO CRYPTOGRAPHY

# Lecture Overview

1. Introduction to Steganography
2. Ciphers
    1. Transposition
    2. Substitutions
    3. Rotor Machines
    4. DES
    5. AES
3. Public-Key Cryptography
4. Message Authentication

Much of this lecture is based on

**[Stallings, 2020]** William Stallings. *Cryptography and Network Security: Principles and Practice,* Pearson. 8th ed., 2020.

In-text references to this source are typically omitted for readability.

# Cryptography

***Cryptography*** is the art (science, study) of writing in secret letters.

The word ‘Cryptography’ comes from two Ancient Greek words:

1. *kryptós* which means "hidden, secret“
2. *graphein*, which means “to write”

Today, by cryptography we mean more than just ‘encryption’, as we will see in today’s lecture.

There are two types of secret writing:

1. Steganography *(also referred to as “concealment systems”*) hides a secret message in a covering message; thus, steganography attempts to hide the existence of a secret message

2. Cryptography does not conceal the existence of a message, only its meaning.

# Steganography

- Steganography can be seen as an alternative to encryption.

- There are many ways to hide the existence of a secret message:
  - using only a subset of letters/words in a longer message marked in some way;
  - using invisible ink;
  - hiding in LSB (Least Significant Bit) in the graphic image or sound file

- Steganography has some serious drawbacks:
  - high overhead to hide relatively few info bits of information – this is now less of a problem than it used to be for shorter secret messages;
  - once the system is broken, it becomes worthless and can no longer be used

# Example 1

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the summer examination package. All Entry Forms and Fees Forms should be ready for final dispatch to the syndicate by Friday 20th or at the very latest, I’m told, by the 21st. Admin has improved here, though there’s room for improvement still; just give us all two or three more years and we’ll really show you! Please don’t let these wretched 16+ proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours,

# Example 1

Dear George,

Greetings to all at Oxford. Many thanks for your
letter and for the summer examination package.
All Entry Forms and Fees Forms should be ready
For final dispatch to the syndicate by Friday
20th or at the very latest, I’m told, by the 21st.
Admin has improved here, though there’s room
for improvement still; just give us all two or three
more years and we’ll really show you! Please
don’t let these wretched 16+ proposals destroy
your basic O and A pattern. Certainly this
sort of change, if implemented immediately,
would bring chaos.

Sincerely yours,

# Example 2

A German spy transmitted the following message during the WWI:

> *Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on byproducts, ejecting suets and vegetable oils.*

# Example 2

A German spy transmitted the following message during the WWI:

*Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on byproducts, ejecting suets and vegetable oils.*

Pershing sails from NY June 1

# Cryptography - Basic Terminology

**Plaintext** - the original message

**Ciphertext** - the code ("encrypted") message

**Cipher** - algorithm for transforming plaintext to ciphertext

**Key** - information used in cipher known only to sender/receiver

**Enciphering (encrypting)** - converting plaintext to ciphertext

**Deciphering (decrypting)** - recovering plaintext from ciphertext

**Cryptography** – the study of encryption principles/methods

**Cryptanalysis (codebreaking)** - the study of principles/ methods of deciphering ciphertext *without* knowing the key

**Cryptology** = Cryptography + Cryptanalysis

(from ancient Greek words *kryptós* ="hidden, secret" and *logia=* "study")

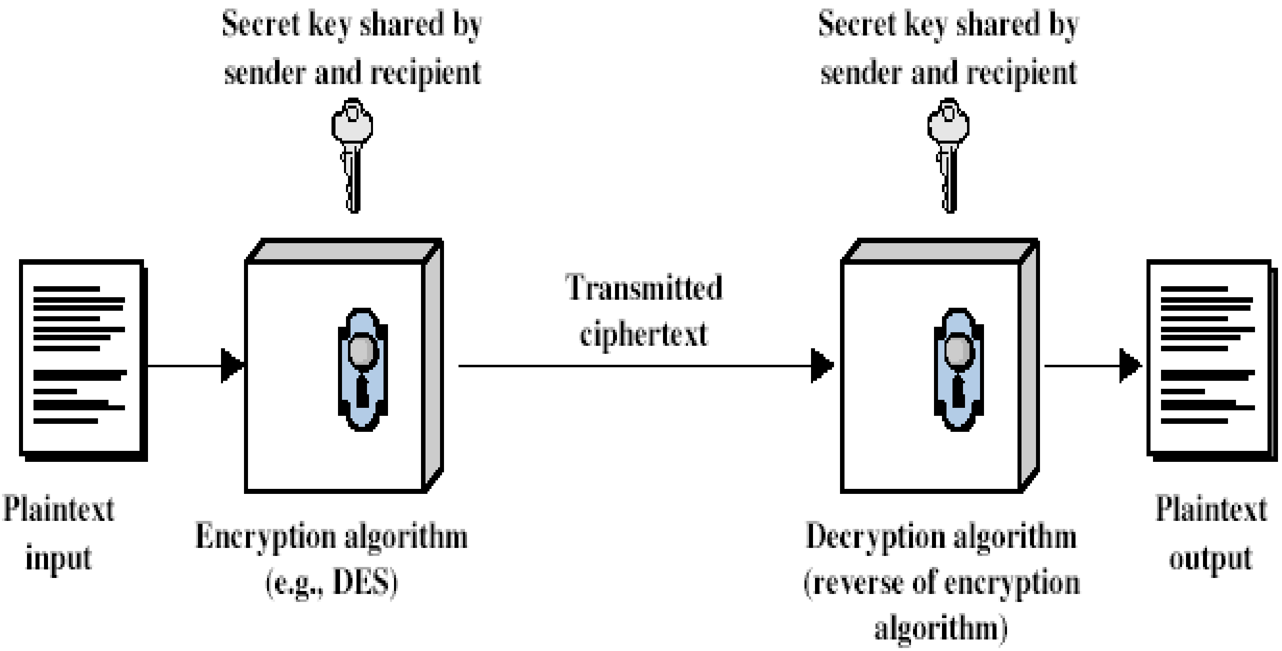
# Encryption

There are two types of encryption:

1. Symmetric encryption, also known as conventional/secret-key/single-key encryption,  where the sender and the recipient share the same key.
2. Public-key (asymmetric) encryption, where the sender’s and the recipient's keys are neither the same nor easily derived from each other

In this lecture we will refer to symmetric encryption unless stated otherwise.

# Symmetric Cipher Model


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Asymmetric, or public-key encryption model is similar, with one important difference: the sender’s and the recipient’s keys are not the same.

Imagine that a sender has a left key (that is, a key that can only turn the lock 90 degrees to the left), and the recipient has a right key. If something is locked with a left key, you need a right key to open it, and vice versa, That is exactly how public-key encryption works.

# Symmetric Encryption Requirements

There are two requirements for secure use of symmetric encryption:

- a strong encryption algorithm
- a secret key known only to sender/receiver

$$Y = \mathrm{E}_K(X)$$
$$X = \mathrm{D}_K(Y)$$

The security of an encryption system should only depend on the secrecy of the key and not the secrecy of the encryption algorithm.

We need a secure channel to distribute keys.

# Symmetric Encryption

In terms of the type  of encryption operations used, we distinguish between

- Transposition ciphers
- Substitution ciphers
- Product ciphers

In terms of the way in which plaintext is processed, we distinguish between

- Block ciphers
- Stream ciphers

# Transposition Ciphers

We said that in order to encrypt the plaintext, we need an enciphering algorithm and an enciphering key. Transposition ciphers rearrange characters according to some scheme often using some geometric figure. The ‘figure’ and the ‘writing-in’ and ‘talking-off’ methods correspond to the enciphering algorithm, while the parameter that determines the figure corresponds to the enciphering key.

***Example 3.***

*Plaintext: DISCONCERTED COMPOSER*

```
D      O        R         C         O
  I  C   N   E    T    D   O    P   S    R
   S       C        E        M        E
```

*Ciphertext: DORCOICNETDOPSRSCEME*

**The algorithm:** Arrange letters of the plaintext in a rail-like way and read off by rows.

**The key:** the ‘rail’ depth (in this case 3).

# Substitution Ciphers

There are several types of the substition ciphers. In the simplest substitution cipher, known as the *monoalphabetic* substitution cipher replaces each character of the plaintext alphabet with the corresponding character of the ciphertext alphabet. Usually, the ciphertext alphabet is a simple rearrangement of the lexicographic order of the characters in the plaintext alphabet.

*Example 4.*

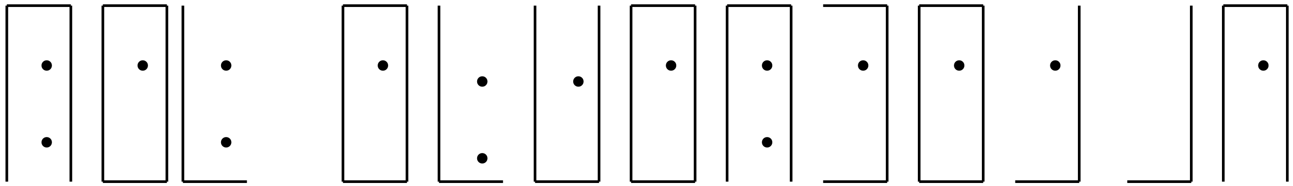
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | Y | D | N | E | O | L | M | P | I | C | G | A | B | F | H | J | K | Q | R | T | U |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

Such a ciphertext alphabet is called a ***keyword mixed alphabet***. In the example above the key of the cipher is *SYDNEY OLYMPIC GAMES*. The repeated letters in the key are dropped and after the key the remaining letters appear in alphabetic order.

The message $M = DOWN\ ELEVATOR$ is encrypted as

$$E_k(M) = NFVB\ EGEUSRFK$$

## *Example 5.* A Churchyard cipher engraved on a tombstone in Trinity Churchyard, New York, 1794:



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

A similar cipher was also engraved on a tombstone in St. Paul's Churchyard, New York, in 1796. The first published solution to this cipher appeared in the New York Herald in 1896 - over 100 years later.

Why did it take so long to break this cipher?

# Breaking Substitution Cipher

Cryptanalysis of a general simple substitution cipher:

1. Brute force attacks: try all 26! decipherments - if 1 decipherment per microsecond, it would take more that $10^3$ years!

2. Instead use a single letter frequency analysis - diagram and trigram distributions are also helpful.

| Character | Percent | |
|---|---|---|
| A | 8.0 | **************** |
| B | 1.5 | *** |
| C | 3 | ****** |
| D | 4.0 | ******** |
| E | 13.0 | ************************** |
| F | 2.0 | **** |
| G | 1.5 | *** |
| H | 6.0 | ************ |
| I | 6.5 | ************* |
| J | 0.5 | * |
| K | 0.5 | * |
| L | 3.5 | ******* |
| M | 3.0 | ******* |
| N | 7.0 | ************** |
| O | 8.0 | **************** |
| P | 2.0 | **** |
| Q | 0.2 | |
| R | 6.5 | ************* |
| S | 6.0 | ************ |
| T | 9.0 | ****************** |
| U | 3.0 | ****** |
| V | 1.0 | ** |
| W | 1.5 | *** |
| X | 0.5 | * |
| Y | 2.0 | **** |
| Z | 0.2 | |

# The Most Frequent English Diagrams

| Diagram | Frequency | Diagram | Frequency |
|---|---|---|---|
| TH | 10.00 | HE | 9.05 |
| IN | 7.17 | ER | 6.65 |
| RE | 5.92 | ON | 5.70 |
| AN | 5.63 | EN | 4.76 |
| AT | 4.72 | ES | 4.24 |
| ED | 4.12 | TE | 4.04 |
| TI | 4.00 | OR | 3.98 |
| ST | 3.81 | AR | 3.54 |
| ND | 3.52 | TO | 3.50 |
| NT | 3.44 | IS | 3.43 |
| OF | 3.38 | IT | 3.26 |
| AL | 3.15 | AS | 3.00 |

# The Most Frequent English Trigrams

ENT

ION

AND

ING

IVE

TIO

FOR

OUR

THI

ONE

# Polyalphabetic Substitution Ciphers and Rotor Machines

***Polyalphabetic substitution ciphers*** conceal the single-letter frequency distribution by using multiple substitutions.

In ***Vigenere cipher*** the key $K$ is a sequence of letters $K = k_1 k_2 \ldots k_d$, where $k_i$ gives the amount of shift in the $i^{th}$ alphabet.

***Example 6:*** Suppose the key is $K = BAND$ (that is, $K = 1\ 0\ 13\ 3$). Then the message $M = RENA\ ISSA\ NCE$ is enciphered as

$C = E_k\,(M) = SEAD\ JSFD\ OCR$

| K | = | B | A | N | D | | B | A | N | D | | B | A | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | = | R | E | N | A | | I | S | S | A | | N | C | E |
| C | = | S | E | A | D | | J | S | F | D | | O | C | R |

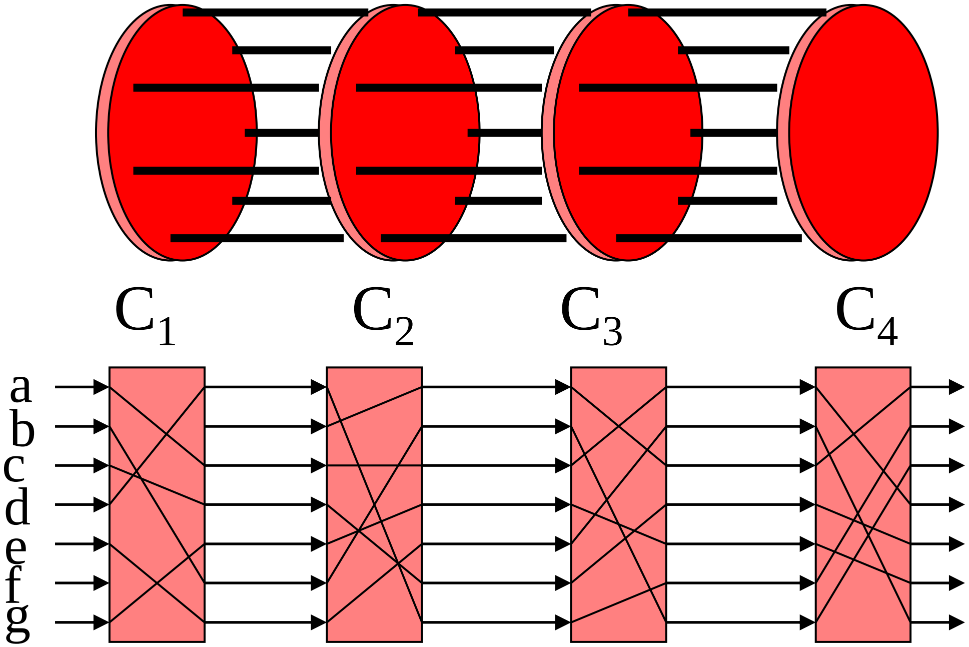
Rotor machines are used to implement polyalphabetic ciphers with a long period.

# Rotor Machines

A Rotor machine consists of a collection of cylinders that can rotate independently of each other. Each cylinder has:

- 26 input pins on its front face, one for each letter in the alphabet
- 26 output pins on its rear face.

Each input pin is wired to a unique output pin. Thus, each cylinder encodes a fixed permutation of the alphabet. After encoding a character in the plaintext, a cylinder is rotated; this changes the relative position of the cylinder and its neighbours.



# Rotor Machines

The rotor machine encryption depends on:

- fixed permutations inside each cylinder
- initial position of each cylinder
- the rule by which the cylinders are rotated.

A rotor machine that consists of $k$ cylinders is capable of providing $26^k$ different encipherments; for example, if there are $4$ cylinders, there are $26^4 = 456{,}976$ different encipherments. In practice, Rotor machines provide a period as long as the plaintext.

# Enigma

A Rotor machine Enigma, used by Germans in World War II, was pretty complex and included a plugboard that permuted the plaintext, and a reflecting rotor that caused each rotor to encrypt each plaintext letter twice. Enigma rotated its cylinders according to the following rule:

- After each plaintext character is enciphered, the first cylinder advances to the next position;
- after the first cylinder has reached a certain position, the second cylinder advances to its next position;
- after the second cylinder has made the complete rotation, the third cylinder advances to its next position, and so on.

Enigma was broken during the World War II by Allies, first by Polish cryptographers. Germans kept modifying Enigma as the war progressed, and the British kept breaking the new versions.

A contributing factor to this successful cryptanalysis was the fact that Germans reused the code-books (keys), and had very stereotyped military messages, often starting with a same phrase.

# Feistel Block Cipher

Feistel block cipher ( 1973) illustrates the underlying principles of many modern block encryption algorithms.

Feistel cipher is a product cipher, which means that it uses a sequence of two or more basic ciphers, so that the final result is cryptographically stronger than any of the components. In particular, Feistel cipher uses a sequence of substitutions and permutations. Such ciphers are also called S-P ciphers.

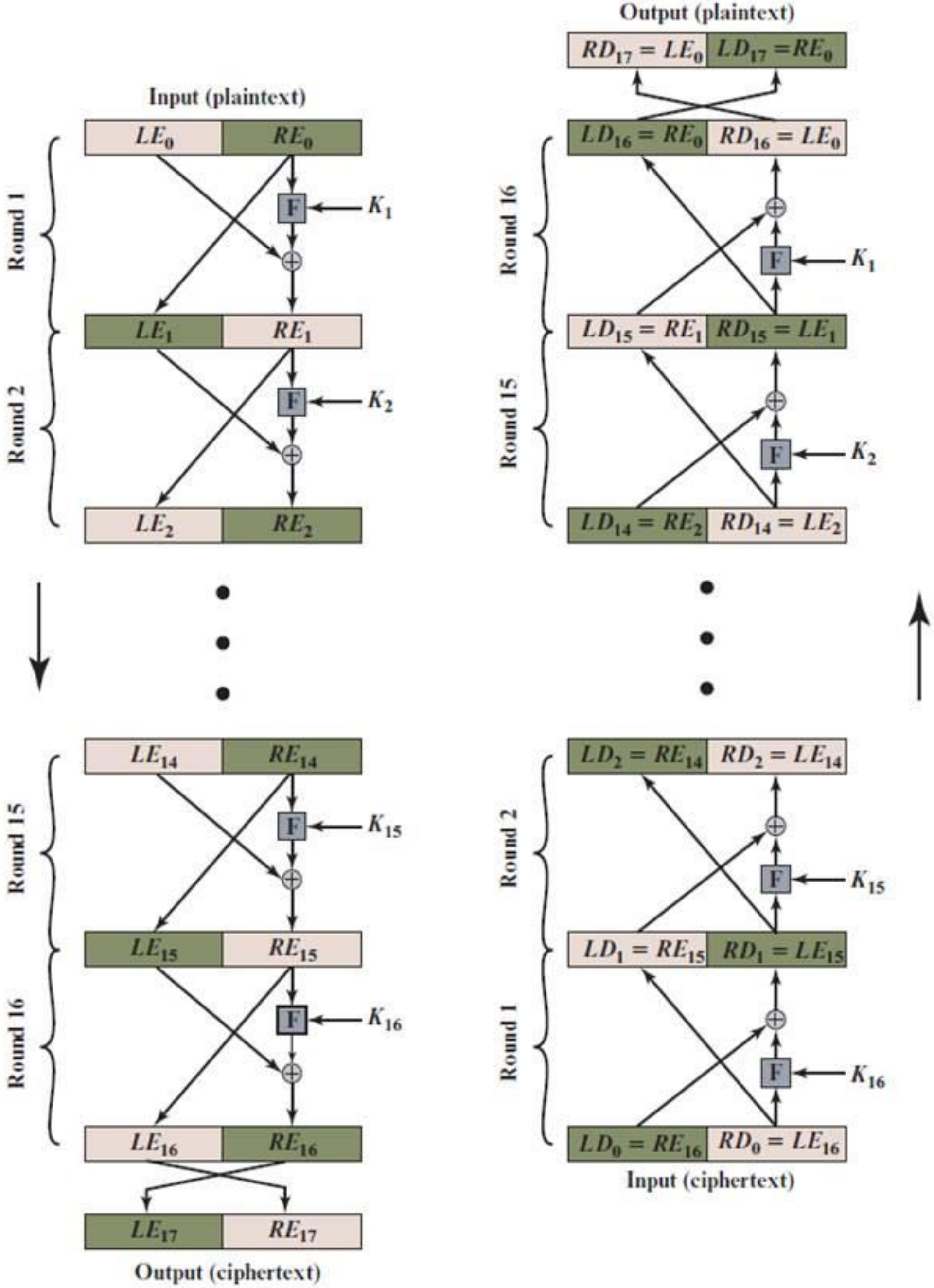
Feistel cipher is based on the work by Shannon, who proposed a development of a product cipher that alternates ***confusion*** and ***diffusion*** functions.

The Feistel cipher takes as input a plaintext block of size $2w$ and a key $K$.

The plaintext block is divided into two halves $L_0$ and $R_0$ which are then passed though $n$ rounds and finally combined together to produce the ciphertext.

Each round has the same structure but uses a different subkey - the subkeys $K_1\ K_2\ K_3\ \ldots\ K_n$ are derived from $K$ and are different from each other.

Each round first applies a round function $F$ to the right half of the data, and takes the $XOR$ of the result and the left half of the data. Then the two halves are interchanged.


Input (plaintext)
LE0
RE0
Round 1
F
K1
LE1
RE1
Round 2
F
K2
LE2
RE2
LE14
RE14
Round 15
F
K15
LE15
RE15
Round 16
F
K16
LE16
RE16
LE17
RE17
Output (ciphertext)
Output (plaintext)
RD17 = LE0
LD17 = RE0
LD16 = RE0
RD16 = LE0
Round 16
F
K1
LD15 = RE1
RD15 = LE1
Round 15
F
K2
LD14 = RE2
RD14 = LE2
LD2 = RE14
RD2 = LE14
Round 2
F
K15
LD1 = RE15
RD1 = LE15
Round 1
F
K16
LD0 = RE16
RD0 = LE16
Input (ciphertext)

# The Data Encryption Standard (DES)

In early 70’s, the National Bureau of Standards (now called the National Institute of Standards and Technology - NIST) issued a request for a proposal for standard encryption algorithm. They had the following design criteria:

- The algorithm must provide a high level of security.
- The algorithm must be completely specified and easy to understand.
- The security of the algorithm must depend solely on the key and not on the secrecy of algorithm.
- The algorithm must be available to all users.
- The algorithm must be adaptable for use in diverse applications.
- The algorithm must be economically implementable.

None of the submissions came close to meeting the requirements, until they received what is now known as DES.

DES was developed by IBM, but with an input from National Security Agency (NSA). It was based on an earlier algorithm by IBM, called Lucifer.

In 1975 the details of DES were published and subjected to criticism from agencies and the general public.

# The Data Encryption Standard (DES)

The two main points of criticism were:

- NSA reduced the key size from the original 128 bits in Lucifer to 56 bits. It was feared the key is too short to withstand the brute-force attack.
- NSA also modified some of DES's S-boxes; although the boxes themselves were public knowledge, the analysis behind them were classified; many feared that NSA has installed a trap-door that would enable them to decrypt a ciphertext without the key.

In 1977, DES was adopted by National Bureau of Standards as a national encryption standard.

In 1998, the Electronic Frontier Foundation spent less than $250,000 to build a DES cracking machine that can get a key in three days.

Not surprisingly, DES is not considered secure anymore; Triple-DES) may still be used, especially for legacy systems.

Despite its main weakness (short key), DES is cryptographically strong; it is resistant to some attacks described in the open scientific literature in 90's.

# Advanced Encryption Standard

Clearly, a replacement for DES was needed due to brute force attack.

Triple-DES could be used instead as it uses the algorithm that has been exposed to more scrutiny than any other algorithm.
If only security was considered, 3DES would have been an appropriate choice.

3DES has the following drawbacks:

- DES itself was designed for mid 70s hardware implementations and does not produce efficient software code;
- 3DES has three times as many rounds as DES
- it uses 64 bit blocks - larger block size is needed

# Advanced Encryption Standard

US NIST issued call for ciphers in 1997.

Out of 21 submissions 15 candidates accepted in Jun 98:

- CAST-256 (Entrust Technologies)
- CRYPTON (Future Systems)
- DEAL (Richard Outerbridge, Lars Knudsen)
- DFC (National Centre for Scientific Research, France)
- E2 (NTT)
- FROG (TecApro Internacional)
- HPC (Rich Schroeppel)
- **LOKI97 (Lawrie Brown, Josef Pieprzyk, Jenniffer Seberry)**
- MAGENTA (Deutche Telekom)
- Mars (IBM)
- RC6 (RSA)
- **Rijndael (Joan Daemon, Vincent Rijmen)**
- Safer+ (Cylink)
- Serpant (Ross Anderson, Eli Biham, Lars Knudsen)
- Twofish (Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson).

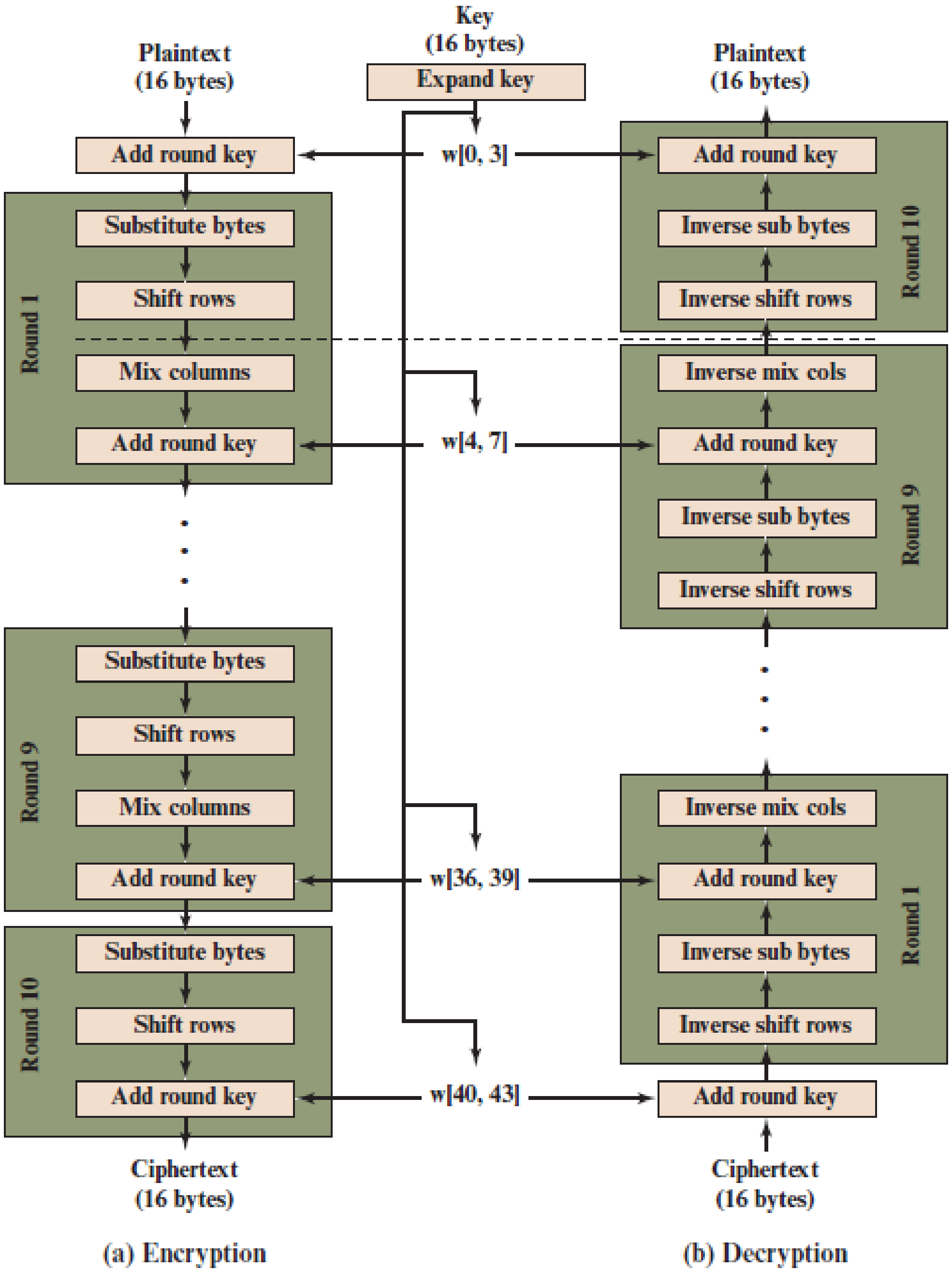
# Advanced Encryption Standard

Five submissions  were shortlisted in August 1999.

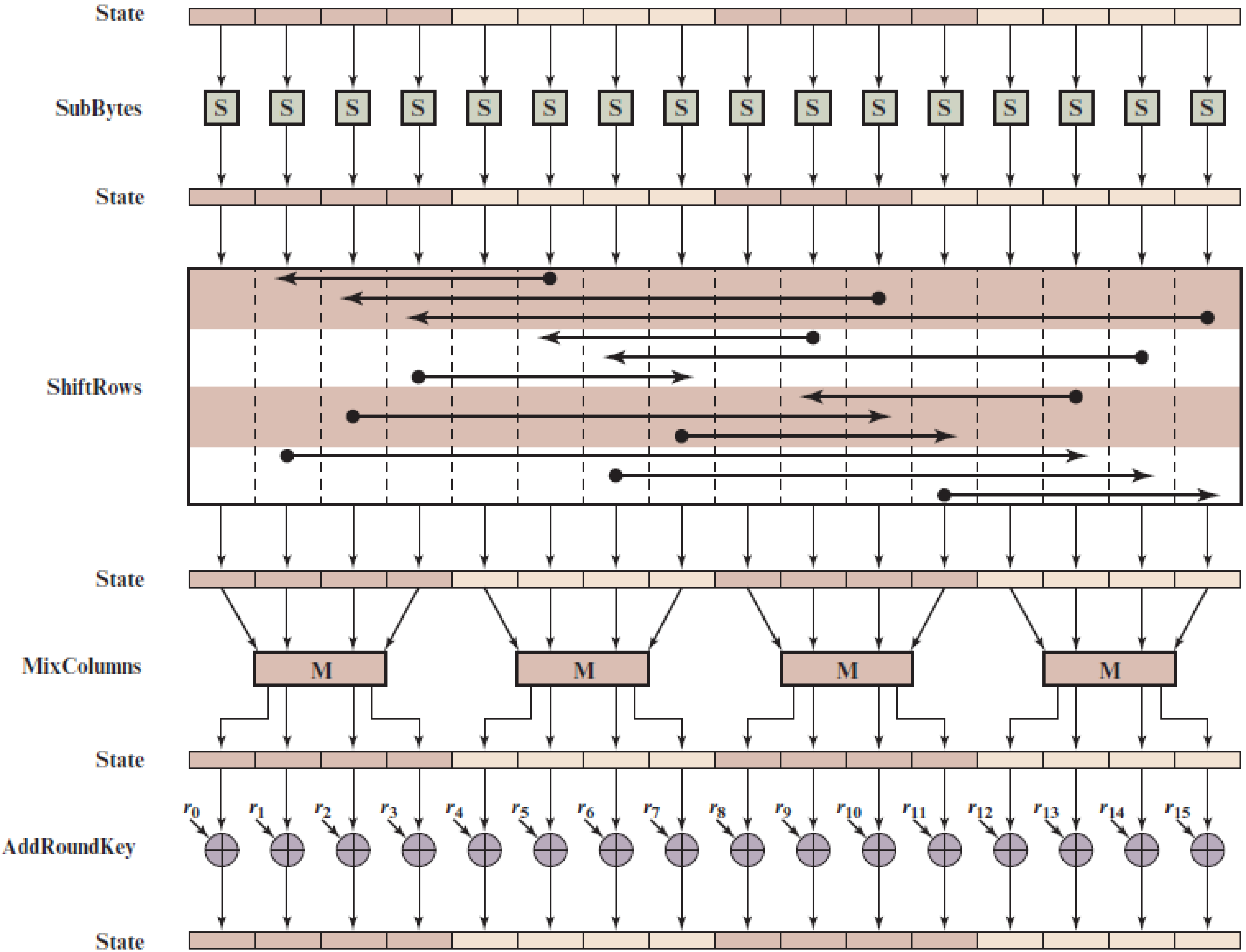
- MARS
- RC6
- Rijndael
- Serpent
- Twofish

Rijndael was selected as the AES in October 2000 and issued as FIPS PUB 197 standard in November 2001.

# Rijndael (AES)



# Rijndael (AES)


State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
$r_0$ $r_1$ $r_2$ $r_3$ $r_4$ $r_5$ $r_6$ $r_7$ $r_8$ $r_9$ $r_{10}$ $r_{11}$ $r_{12}$ $r_{13}$ $r_{14}$ $r_{15}$
State

# Public-Key Cryptography

In 1976, Diffie and Hellman proposed ***public-key cryptography.*** Encryption key and decryption key are *not* the same.

Each user $A$ has a public encryption procedure $E_A$ which may be placed in a public directory, and a private decryption procedure $D_A$ which they keep secret.

Public-key cryptosystem has following properties:

1. $D\big(E(M)\big) = M$
2. Both $E$ and $D$ are easy to compute.
3. It is not easy to compute $D$ from $E$.
4. $E(D(M)) = M$

Like conventional cryptosystem, public-key cryptosystem can provide both **confidentiality** and **authenticity**. Unlike conventional cryptosystem, public-key cryptosystem can also provide a method of implementing **digital signatures**, and it does not need an exchange of secret key prior to private communication.

In 1978, Rivest, Shamir and Adleman published the first method of realizing public-key cryptography the famous RSA system, which can be used to provide both encryption and digital signature.

# Confidentiality

1. Bob ($B$) wants to send a private message to Alice ($A$).
2. First Bob retrieves $E_A$ from the public directory.
3. Then Bob enciphers $M$ obtaining $E_A(M)$ and he sends it to Alice.
4. Alice deciphers $E_A(M)$ by computing $D_A(E_A(M)) = M$

*Confidentiality* is provided by step 3: Alice is the only one who can decipher $E_A(M)$.

Advantages:

1. Encryption key (public key) can be sent as a plaintext or be placed in the public directory.
2. There is no need for distribution of secret decryption key.

# Signatures

Authenticity can be provided by the means of digital signature.

1. Bob receives a message $M$ signed by Alice.
2. Bob must be able to validate Alice's signature on $M$.
3. Nobody can forge Alice's signature.
4. A judge or third party can check whether it is Alice's signature or not.

Signature must be both message and signer dependent.

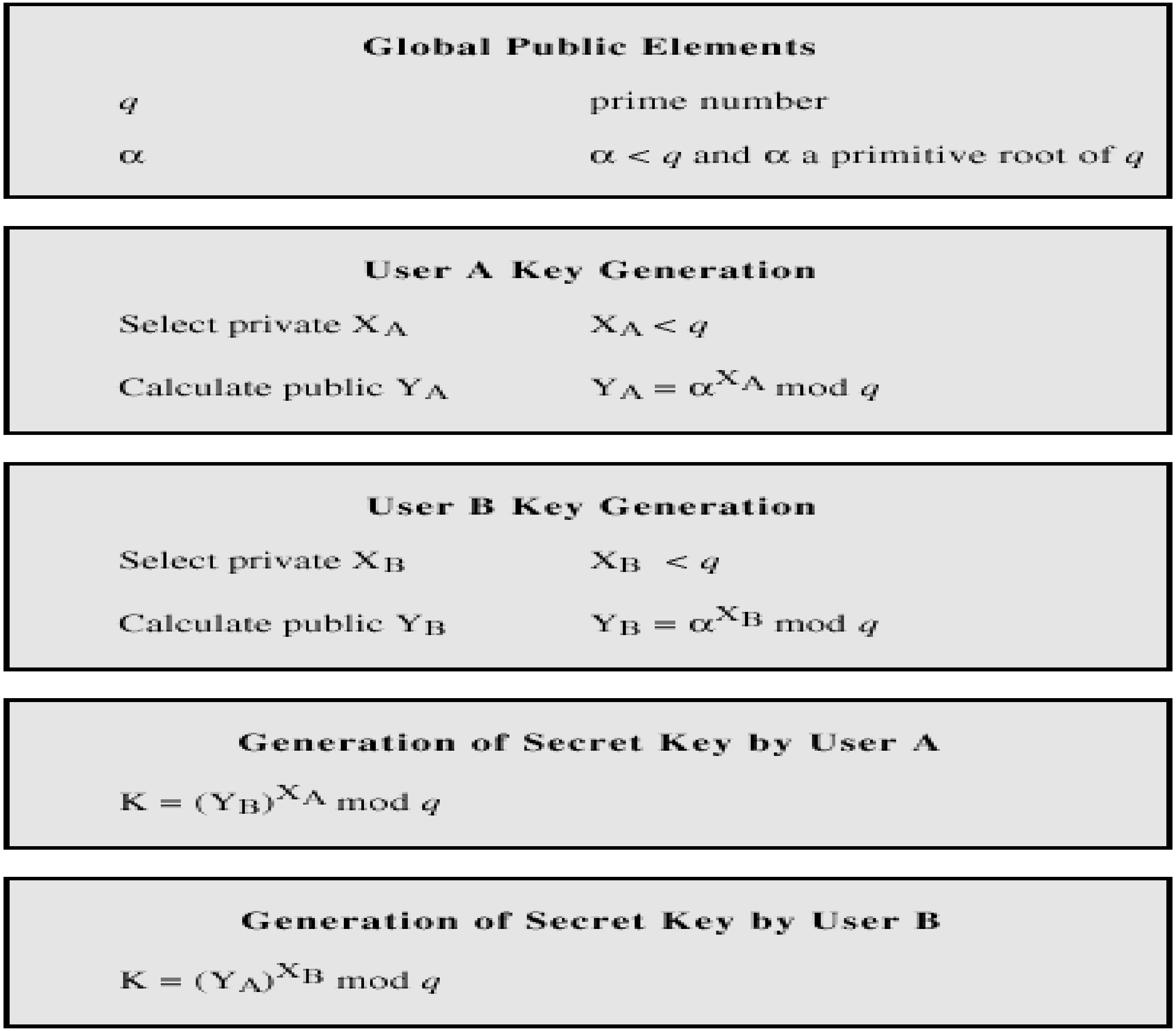
How does Alice send a signed message $M$ to Bob?

1. Alice first 'signs' a message $M$ by computing $D_A(M) = S$ for authenticity.
2. Then Alice encrypts $S$ by computing $E_B(S)$ for confidentiality.
3. Bob first compute $S = D_B(E_B(S))$.
4. Then Bob obtains $M$ by computing $E_A(S) = E_A(D_A(M)) = M$

Alice can not later deny having sent Bob this message, since no one else could have created $S = D_A(M)$.

Bob can not forge Alice's signature since he does not know $D_A$.

# Diffie-Hellman Key Exchange



**Global Public Elements**

| | |
|---|---|
| $q$ | prime number |
| $\alpha$ | $\alpha < q$ and $\alpha$ a primitive root of $q$ |

**User A Key Generation**

| | |
|---|---|
| Select private $X_A$ | $X_A < q$ |
| Calculate public $Y_A$ | $Y_A = \alpha^{X_A} \bmod q$ |

**User B Key Generation**

| | |
|---|---|
| Select private $X_B$ | $X_B < q$ |
| Calculate public $Y_B$ | $Y_B = \alpha^{X_B} \bmod q$ |

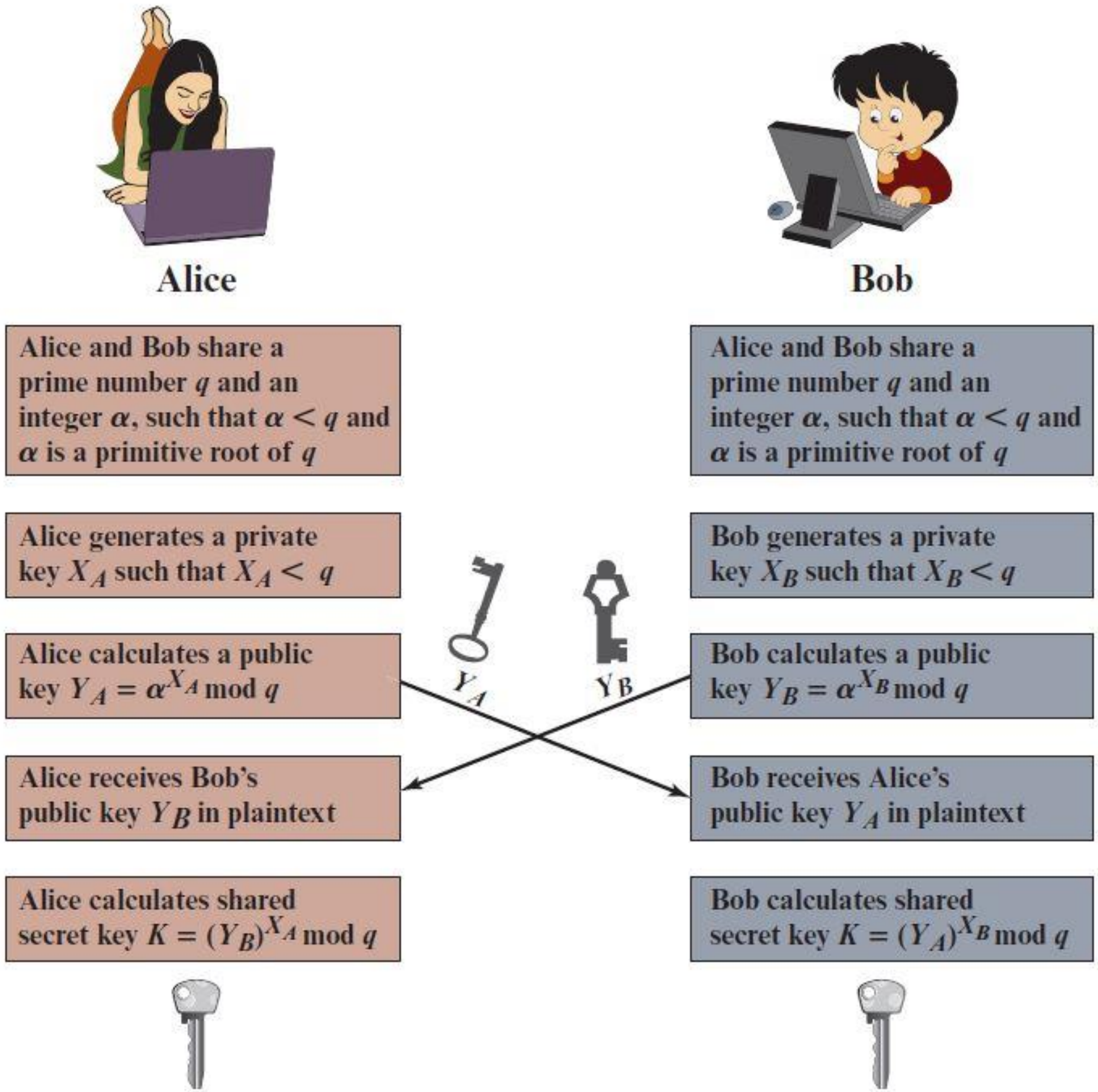
**Generation of Secret Key by User A**

$K = (Y_B)^{X_A} \bmod q$

**Generation of Secret Key by User B**

$K = (Y_A)^{X_B} \bmod q$

# Diffie-Hellman Key Exchange



Alice and Bob share a prime number $q$ and an integer $\alpha$, such that $\alpha < q$ and $\alpha$ is a primitive root of $q$

Alice generates a private key $X_A$ such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$

Alice receives Bob's public key $Y_B$ in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$

Alice and Bob share a prime number $q$ and an integer $\alpha$, such that $\alpha < q$ and $\alpha$ is a primitive root of $q$

Bob generates a private key $X_B$ such that $X_B < q$

Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$

Bob receives Alice's public key $Y_A$ in plaintext

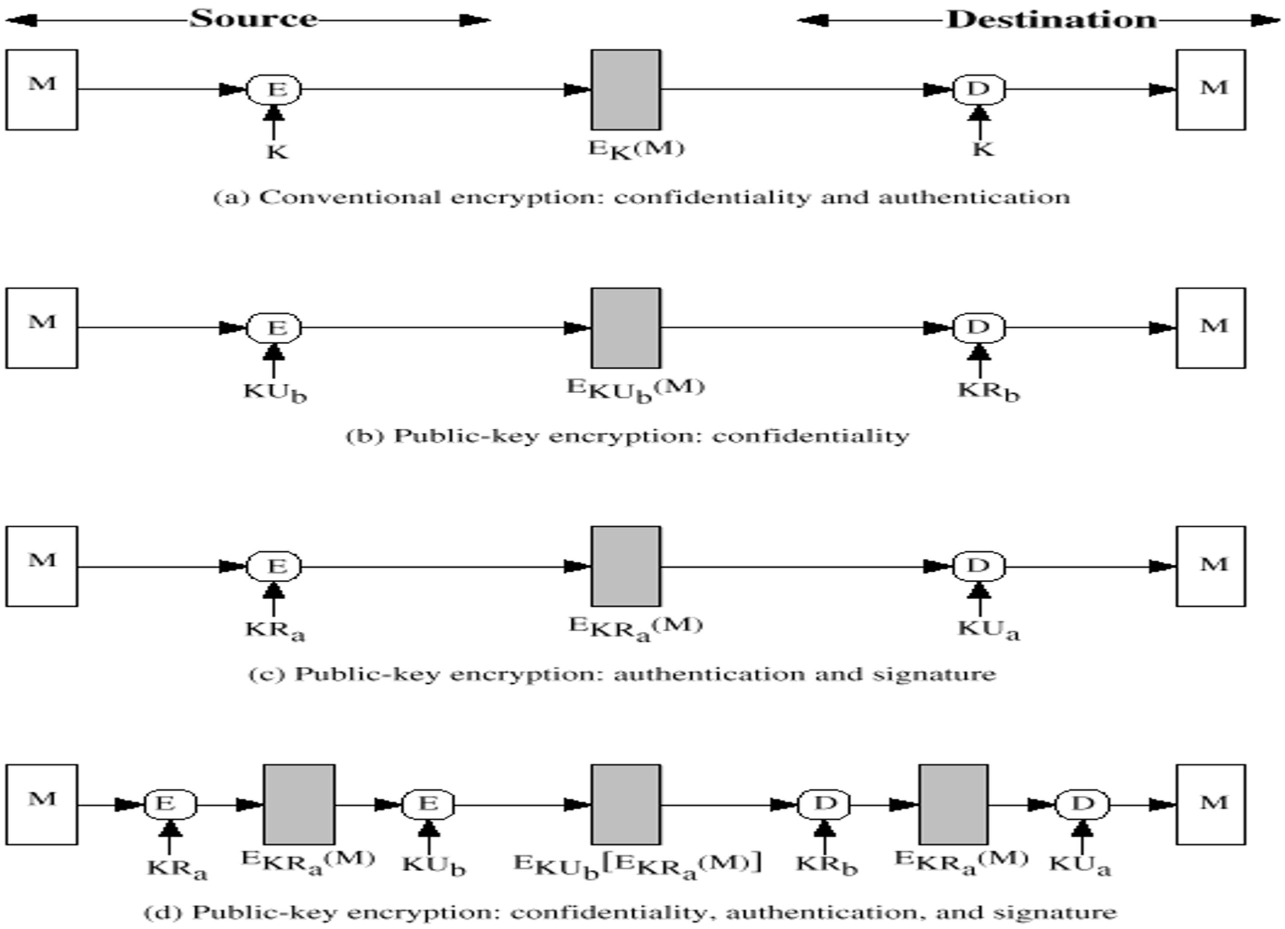
Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$

# Authentication Functions

Authentication functions fall into the following classes:

1. **message encryption** – the authenticator is the ciphertext of the entire message
2. **message authentication code (MAC)** - the authenticator is  a fixed-length value produced from the message and a key by a public function
3. **hash function** - the authenticator is  a fixed-length hash value produced from a message of any length by a public function

# Message Encryption



(a) Conventional encryption: confidentiality and authentication

(b) Public-key encryption: confidentiality

(c) Public-key encryption: authentication and signature

(d) Public-key encryption: confidentiality, authentication, and signature

If public-key encryption is used, encryption provides no confidentiality since anyone can learn the public-key.

However, if

- sender **signs** message using their private-key
- then encrypts with recipient's public key,

we have both confidentiality and authentication (and signature!), but at cost of two public-key uses on message

# Message Authentication Code

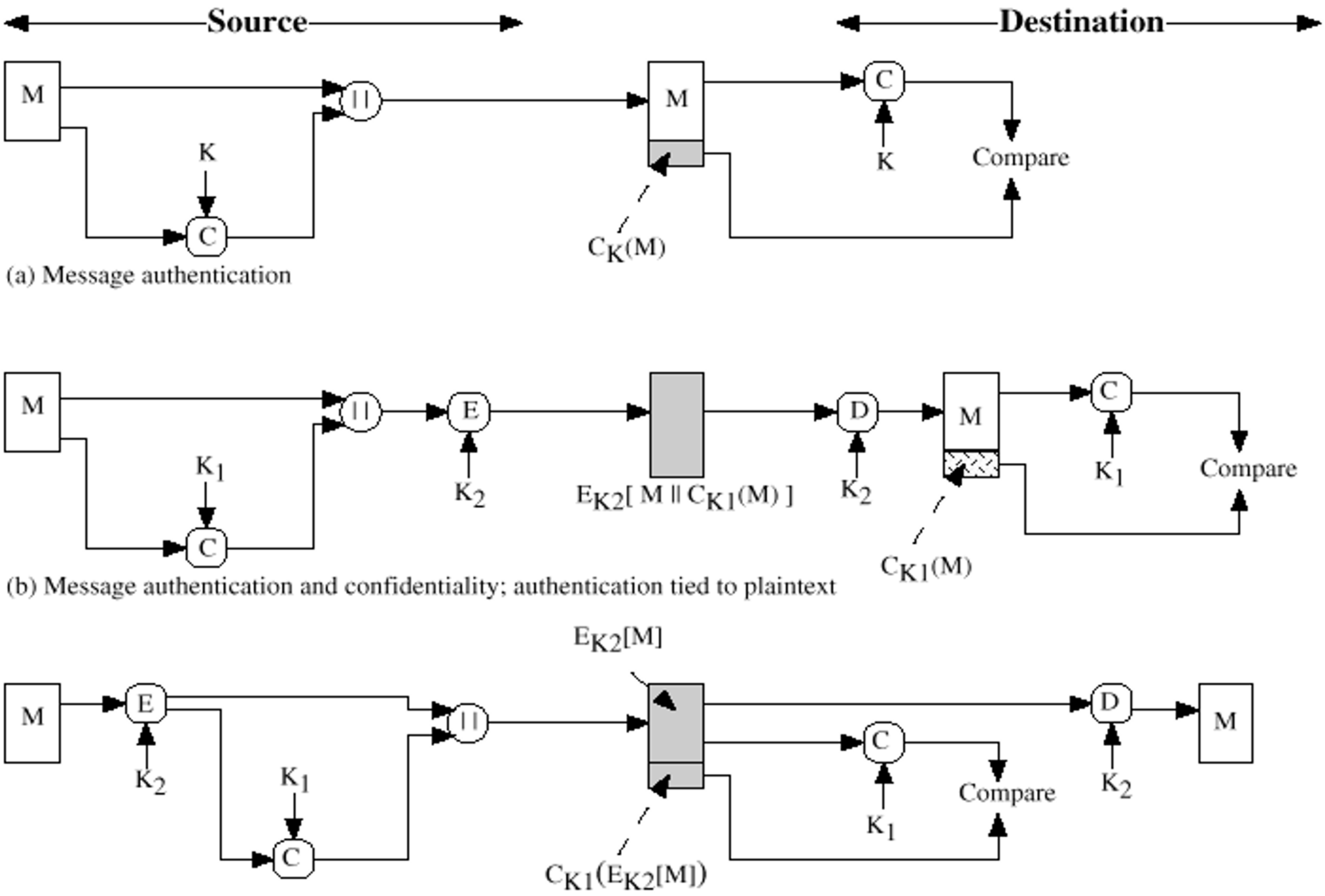
A secret key, known only to the sender and the receiver, is used to generate a small fixed-sized block of data (called cryptographic checksum or MAC), which is appended to the message. The receiver uses the same key to obtain the MAC and compare it to the received MAC.

The receiver is assured that:

1. the message has not been altered
2. the message comes from the sender
3. if the message includes a sequence number, the receiver also knows that the ordering is correct.

We can use encryption to add confidentiality, as illustrated in the next figure.

# Message Authentication Code



(a) Message authentication

(b) Message authentication and confidentiality; authentication tied to plaintext

(c) Message authentication and confidentiality; authentication tied to ciphertext

# Message Authentication Code

A MAC function is similar to encryption, but MAC algorithm is not necessarily reversible, and the checksum is of the fixed length.

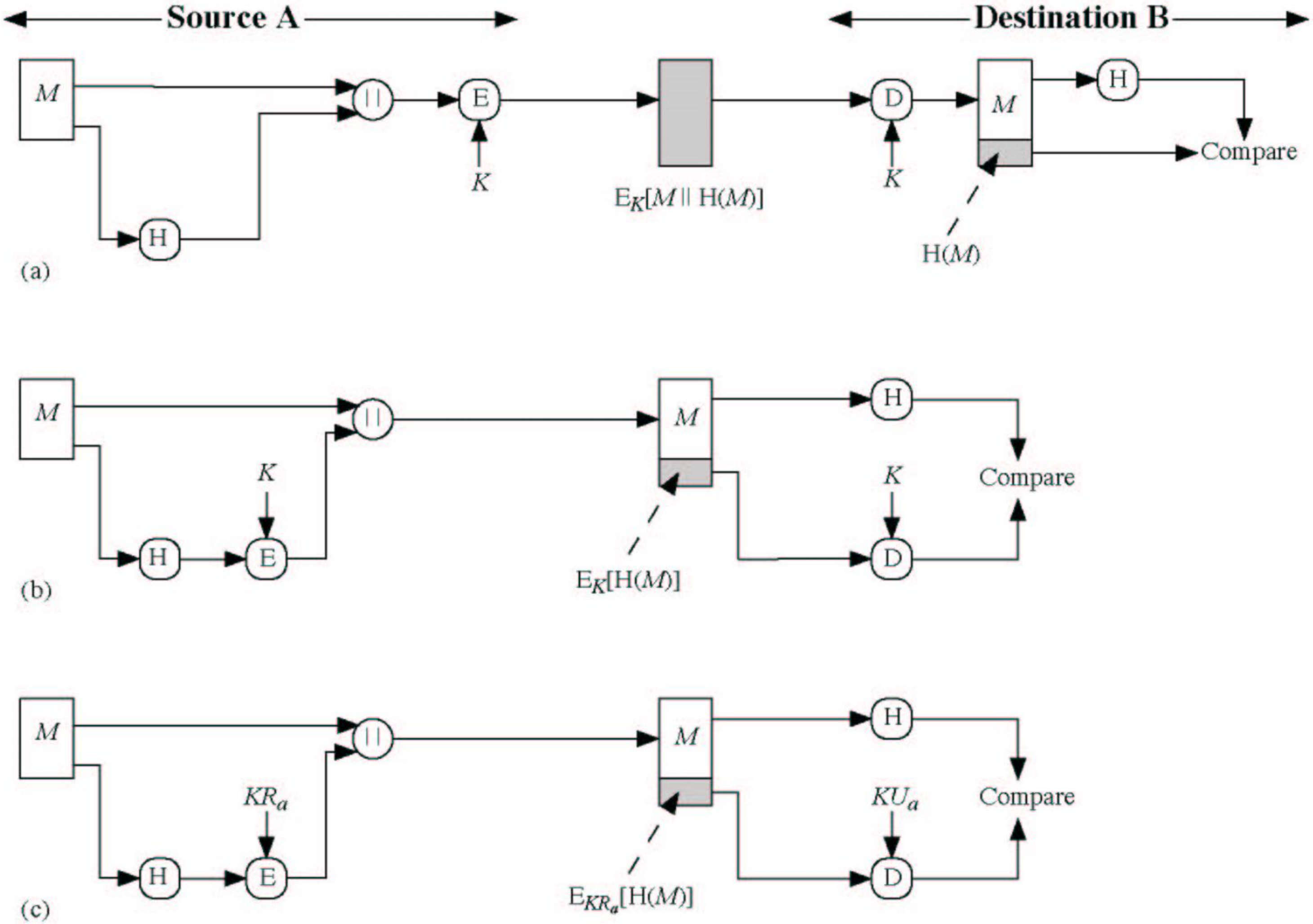
Advantages of using MAC rather than encryption:

- Authentication and confidentiality are separated - can be implemented on different levels
- It is faster - only MAC is encrypted
- Authentication can be checked on need basis
- Message is protected on the target machine and not only in transit

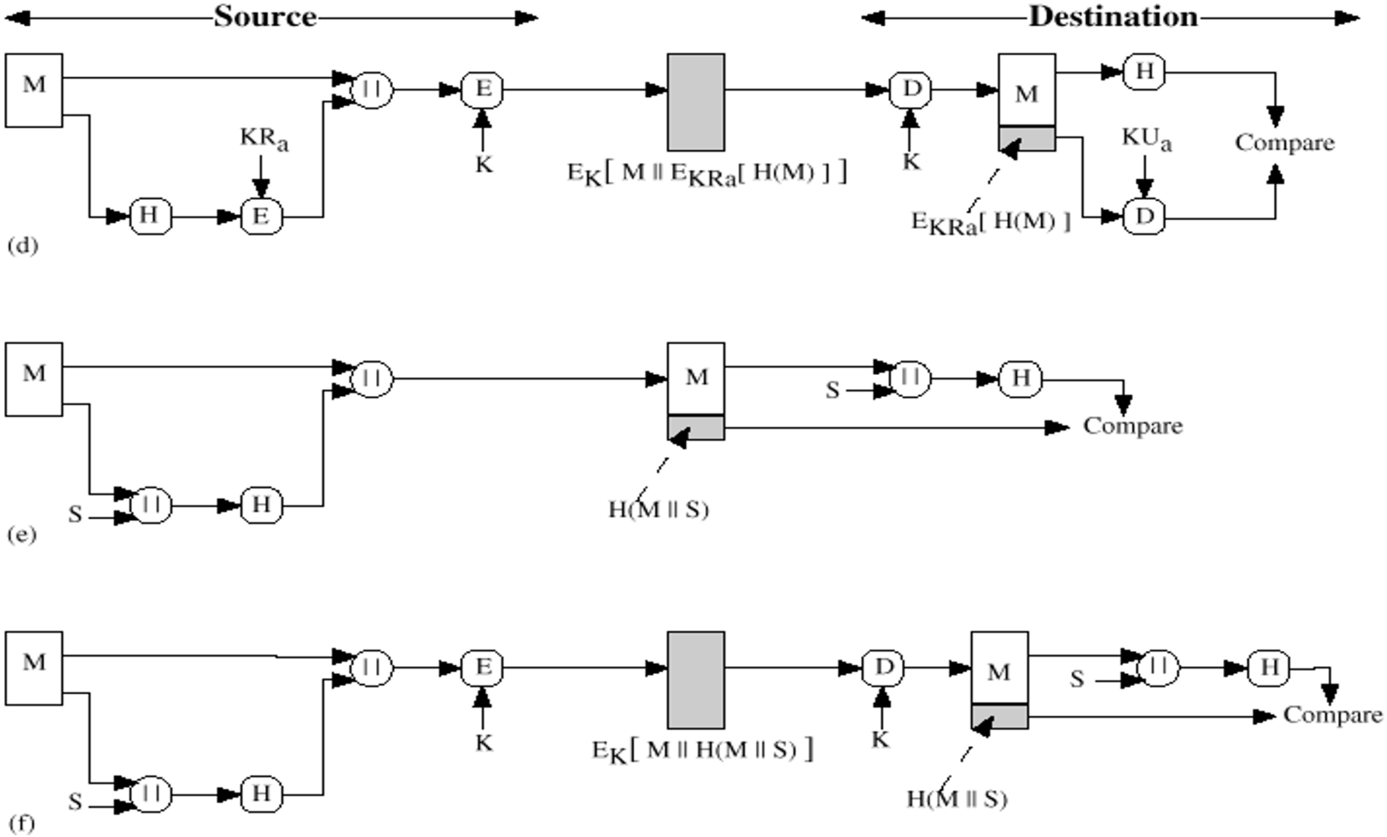
# Hash Function

A hash function takes as input a variable-size message $M$ and produces as output a fixed-size hash code $H(M)$ (also called a message digest). A hash function is one-way function - it is NOT reversible.

Different ways in which a hash code can provide authentication are illustrated in the next figure.



# Hash Function



## Reasons for avoiding encryption:

- Encryption is slow
- Hardware cost
- Hardware is optimized towards larger messages
- In the past some encryption algorithms were patented and were subjects to export controls

# References

[**Stallings, 2020**] William Stallings. *Cryptography and Network Security: Principles and Practice,* Pearson. 8th ed., 2020.