Security Aspects for Maritime Networks

CSP004_S_M

PRESENTATION BY:
DR. STEFAN SCHAUER
DR. ABDELKADER SHAABAN
AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

Acknowledgement

- *Co-Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.*
- *Project Agreement no. 101083594*

Security Aspects for Maritime Networks

Overview

- Topic-1: Secure Network Architecture and Design
- Topic-2: Cryptographic Techniques for Ensuring Secure Data Transmission
- Topic-3: Security mechanisms, services, and attacks in OSI reference model

Agenda

01. Introduction

02. Overview

03. Symmetric Encryption

04. Asymmetric Encryption

05. Digital Signature

Introduction

Significance of Cryptography: The Korean Register (KR) underscores the critical role of cryptography in the safekeeping of data.

Objectives

The Korean Register outlines the **goals of type approval** for **maritime** cybersecurity.

Security Requirements

Details the specific **security requirements** needed and their respective **levels**.

Cryptography

Highlights the importance of **cryptography** in **securely** storing crucial **data**.

https://safety4sea.com/kr-explains-the-importance-of-cryptography-in-safekeeping-data/

Introduction

- **Maritime Cyber Security Type Approval**
  - This type of approval certifies manufacturers for **equipment intended** for use
  - The approval is granted based on the results of **examinations**, **tests**, and **inspections** as specified in the guidance.
  - Equipment must **meet** these before **installation** on **board** is **approved** by the society.

- KR notes that the use of cryptography is a **common requirement across security levels 1 to 4.**

- **Manufacturers** seeking **cybersecurity type approval** from the Korean Register must **demonstrate** that the encryption algorithms used in their systems/equipment are **secure** and **not vulnerable**.

•Use of cryptography: "If **cryptography** is required, the component should use **cryptographic security** mechanisms according to **internationally recognized** and **proven** security **practices** and **recommendations**" … Korean Register reports

Section 1 General	Section 5 Data Confidentiality	Section 9 Software Application Requirements
Sections 2 Identification and Authentication	Section 6 Restricted Data Flow	Section 10 Embedded Device Requirements
Section 3 Use Control	Section 7 Timely Response to Events	Section 11 Host Device Requirements
Section 4 System Integrity	Section 8 Resource Availability	Section 12 Network Device Requirements

https://safety4sea.com/kr-explains-the-importance-of-cryptography-in-safekeeping-data/
http://krs.westus.cloudapp.azure.com/Files/KRRules/KRRules2020/data/data_other/ENGLISH/gc31e000.pdf

Cryptography

- **Cryptography** is the field of applying **mathematical principles** to **encrypt** and **decrypt** data.
- It allows for the secure **storage** and **transmission** of sensitive **information**, **protecting** it from **unauthorized** access, **particularly** when traversing insecure **networks** such as the **Internet**.
- While cryptography **focuses** on **data security**, **cryptanalysis concentrates** on **analyzing** and **breaking** secure communication.
- Classical **cryptanalysis** involves a **combination** of **analytical reasoning**, **mathematical tools**, pattern **recognition**, patience, **determination**, and **sometimes** luck.
- Those **engaged** in cryptanalysis are often referred to as **attackers**.

Cryptography Academy, https://cryptographyacademy.com/introduction/, accessed in February 2024

Cryptography

- Cryptographic strength is assessed based on the **time** and **resources** needed to **decrypt** the plaintext.
- **Strong cryptography** produces **ciphertext** that is exceptionally **challenging** to **decipher** without employing the appropriate **decryption tool** or **method**.
- Even with the huge **computational power of modern computers**, including a billion computers performing a billion checks per **second**, **deciphering strong cryptography** would take longer than the **lifespan** of the universe.

Cryptography Academy, https://cryptographyacademy.com/introduction/, accessed in February 2024

History of Cryptography

- The origin of **cryptography** is linked to the **age** when **humans** started **writing**.
- As **civilizations** evolved, **societies** were organized into **tribes**, **groups**, and **kingdoms**.
- This resulted in the rise of concepts like **power**, **battles**, **supremacy**, and **politics**.
- These increased the need for **secret communication** among **individuals**.
- Cryptography evolved **continuously** to **meet** this demand for **covert communication**.
- The **roots** of **cryptography** are found in **Roman** and **Egyptian civilizations**.

History of Cryptography

- Scholars later transitioned to employing simple **mono-alphabetic substitution** ciphers around **500** to **600** BC.
- This method entailed **replacing letters** in a **message** with other **letters** according to a **secret rule**.
- The rule served as a key to **deciphering** the **message** from the **scrambled** text.
- The **ancient Roman cryptographic** technique, commonly referred to as the **Caesar Shift Cipher**, involves **shifting** the **letters** of a message by a **predetermined number** (often three).
- The **recipient** of the message would then **reverse** the **shift** by the **same number** to **retrieve** the original message.

History of Cryptography

Hieroglyph – The Oldest Cryptographic Technique

- The first known evidence of **cryptography** can be traced to the use of '**hieroglyph**'.
- Around **4000** years ago, the **Egyptians** communicated through **messages** written in **hieroglyphs**, a code kept **confidential** by **scribes** entrusted with **transmitting** messages for the **kings**.
- An example of such a **hieroglyph**.

https://www.tutorialspoint.com/cryptography/origin_of_cryptography.htm, accessed in February 2024

Features of Cryptography

Confidentiality

Data is **exclusively accessible** to its **intended recipient**, **preventing** access by any **unauthorized** individuals

Integrity

Data remains **unaltered** during **storage** or **transmission** between the **sender** and intended **recipient**, with any **modifications** being readily **detectable**

Non-repudiation

The **creator** or **sender** of data **cannot** disclaim their **intent** to **transmit** the **information** at a later stage

Authentication

The identities of **both** the **sender** and **receiver** are **verified**, along with the **confirmation** of the information's **source** and **destination**

https://www.tutorialspoint.com/cryptography/origin_of_cryptography.htm, accessed in February 2024

Types Of Cryptography

Symmetric Key Cryptography: In this encryption system, both the **sender** and **receiver** share a **single key** for **encrypting** and **decrypting** messages. While **Symmetric** Key Systems **offer speed** and **simplicity**, **securely** exchanging the **key** between the **sender** and **receiver** poses a challenge. Common **examples** of symmetric key cryptography systems include the **Data Encryption Standard** (**DES**) and the **Advanced Encryption Standard** (**AES**).

Hash Functions: This algorithm operates **without** any **key**. It computes a **fixed-length hash** value based on the **input plain text**, rendering it **impossible** to **retrieve** the original content from the hash. Hash **functions** are commonly employed in various **operating systems** for **password encryption** purposes.

Asymmetric Key Cryptography: In this system, a **pair of keys** is employed to **encrypt** and **decrypt** data. The **receiver's public** key is utilized for **encryption**, while their **private** key is employed for **decryption**. The **public** and **private** keys are **distinct**. Even if the **public** key is **widely known**, only the intended **recipient**, who **possesses** the **private key**, can **decode** the **message**. The **RSA** algorithm is **one** of the most **well-known asymmetric** key **cryptography algorithms**.

Cryptography and its Types – GeeksforGeeks, accessed in February 2024

Symmetric Encryption

Symmetric Encryption

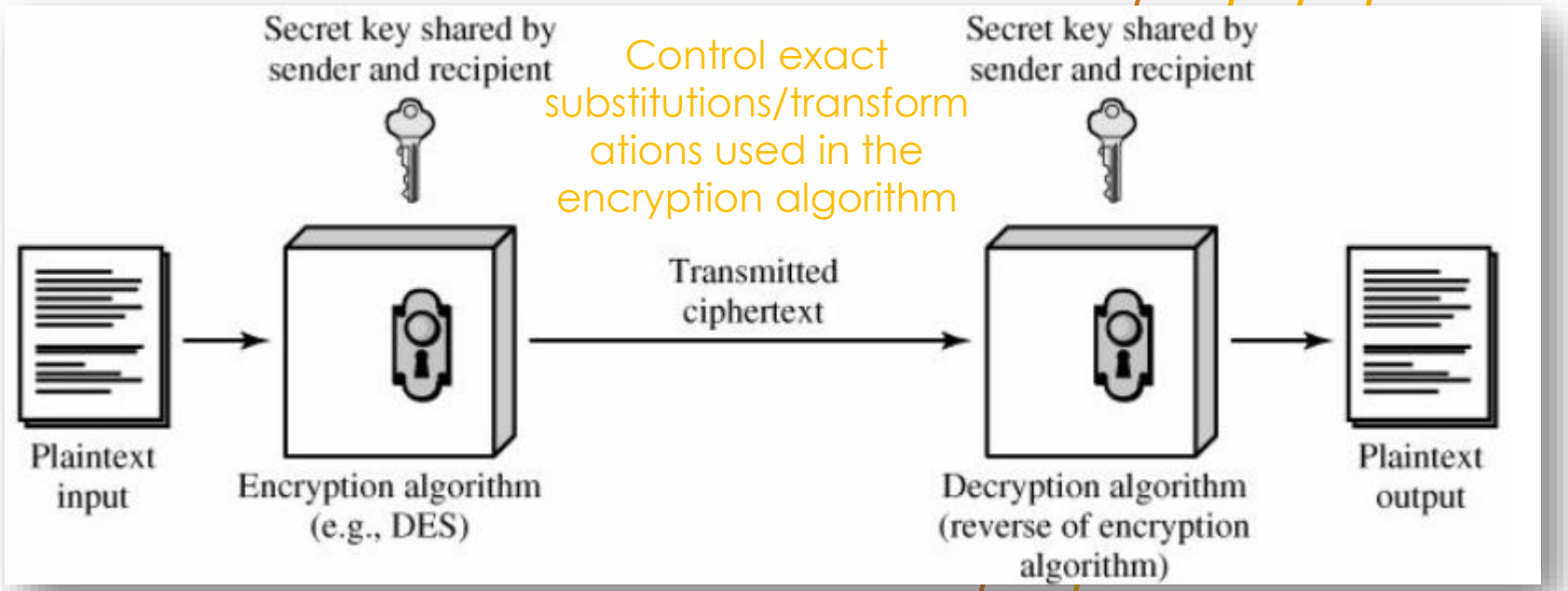
- Symmetric **encryption**, also called **conventional** or **single-key** encryption, was the only type of **encryption** in use prior to the **development** of **public-key** encryption in the **1970s**.
- It **remains** the **more widely** used **encryption** method.
- **Symmetric encryption** employs a **single key** for both **encryption** and **decryption**.
- This key is **shared** between the **sender** and **receiver**.
- Both **parties** can **encrypt** or **decrypt** messages using the **same key**.

Some Basic Terminology

- **Plaintext** - original message
- **Ciphertext** - coded message
- **Cipher** - algorithm for **transforming plaintext** to **ciphertext**
- **Key** - **info used** in cipher known only to **sender/receiver**
- **Encipher (Encrypt)** - converting **plaintext** to **ciphertext**
- **Decipher (Decrypt)** - recovering **ciphertext** from plaintext
- **Cryptography** - study of encryption **principles/methods**
- **Cryptanalysis (Codebreaking)** - study of **principles/ methods** of **deciphering** ciphertext ***without*** knowing key
- **Cryptology** - field of **both cryptography** and **cryptanalysis**

Symmetric Cipher Model

original message



original message

Performs substitutions/transfor mations on plaintext

Performs an inverse action of the encryption algorithm

Main Requirements

- We assume that it is **impractical** to **decrypt** a message on the basis of the **cipher text** plus knowledge of the **encryption/decryption algorithm** and do not need to keep the **algorithm secret**; rather, we only need to keep the **key secret**.
- This feature of **symmetric encryption** is what makes it **feasible** for **widespread use**. It allows easy **distribution** of s/w and h/w implementations.
- two requirements for secure use of symmetric encryption:
- a **strong** encryption algorithm
- a **secret** key known only to **sender/receiver**
- **mathematically** have:

  $Y = E_K(X)$

  $X = D_K(Y)$
- It can be considered a pair of functions with **plaintext X**, **ciphertext Y**, **key K**, **encryption algorithm EK**, and **decryption algorithm DK**.

Cryptography Dimensions

1. **The type of operations used for transforming plaintext to ciphertext**:

- **Encryption algorithms** rely on two main principles: substitution and transposition.
  - **Substitution** involves **mapping** each element in the **plaintext** (i.e., **bit**, **letter**, **group of bits**, or **letters**) to another element.
  - **Transposition** **rearranges** elements within the **plaintext**.
- The **primary requirement** is to ensure **reversibility**, meaning no information **loss**.
- Most **systems**, known as product **systems**, incorporate multiple **stages** of **substitutions** and **transpositions**.

Cryptography Dimensions

2. The number of keys used

- When both **sender** and **receiver** utilize the **same** key, it's called **symmetric**, **single-key**, or **conventional** encryption.
- If the **sender** and **receiver** use **different keys**, it's termed **asymmetric** or **public-key encryption**.

3. The way in which the plaintext is processed

- A **block cipher** operates on **input blocks**, generating an **output block** for each **input block**.
- In contrast, a **stream cipher** processes **input elements continuously**, producing **output one element** at a time.

Cryptanalysis and Brute-force Attack

Typically, the objective is to **recover** the key in use rather than simply to **recover** the plaintext of a single ciphertext.

There are two general approaches:

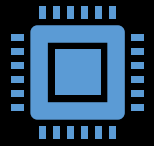
Cryptanalytic attacks rely on the **nature** of the **algorithm** plus perhaps some **knowledge** of the **general characteristics** of the **plaintext** or even some sample **plaintext-ciphertext** pairs.

Brute-force attacks try **every possible key** on a piece of ciphertext until an **intelligible translation** into plaintext is obtained. On average, half of **all possible keys** must be **tried** to achieve success.

Cryptanalysis and Brute-force Attack

Types of **Attacks** on **Encrypted** Messages

Ciphertext only



- Encryption algorithm
- Ciphertext

Known plaintext

- Encryption algorithm
- Ciphertext
- One or more plaintext-ciphertext pairs formed with the secret key

Chosen plaintext

- Encryption algorithm
- Ciphertext
- Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key

Chosen ciphertext

- Encryption algorithm
- Ciphertext
- Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Chosen text

- Encryption algorithm
- Ciphertext
- Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
- Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Cryptanalysis and Brute-force Attack

Two additional definitions of **encryption** schemes are **either** they take **too long** or are **too expensive** to break the cipher.

Unconditionally secure

No matter how **much computer power** or **time** is available, the cipher cannot be **broken since** the **ciphertext provides insufficient** information to uniquely **determine** the **corresponding** plaintext

Computationally secure

Given **limited computing resources** (e.g., the **time needed** for calculations is greater than the age of the universe), the cipher cannot be broken.

Brute Force Search

Always possible to simply **try every key**

Most basic attack, **proportional to key size**

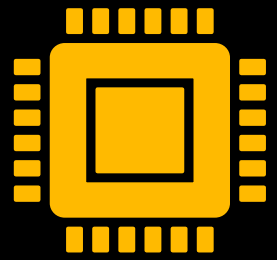
Assume either know/**recognise** plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	**2.15 milliseconds**
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	**10.01 hours**
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	**5.4×10^{18} years**
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	**5.9×10^{30} years**
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	**6.4×10^6 years**

Substitution Techniques

Classical Substitution Ciphers

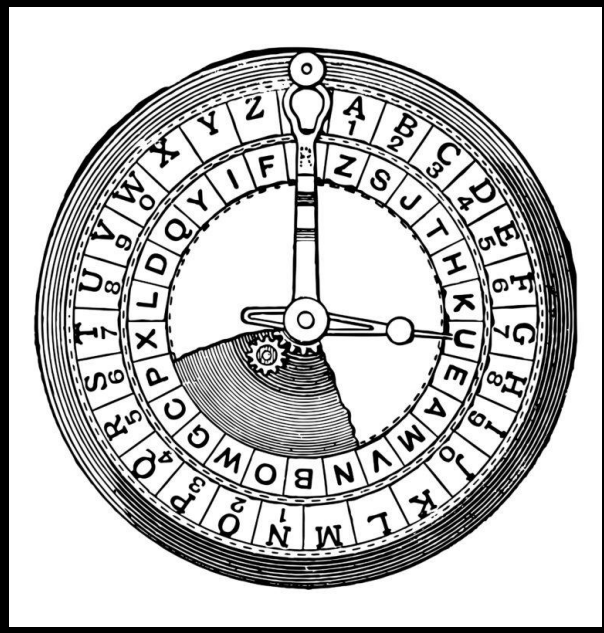
where letters of **plaintext** are **replaced** by other **letters** or by **numbers** or symbols



or if **plaintext** is viewed as a **sequence** of **bits**, then **substitution** involves **replacing plaintext** bit patterns with **ciphertext** bit **patterns**

Caesar Cipher

- The **Caesar Cipher** is one of the **earliest** encryption techniques, **attributed** to **Gaius Julius Caesar**, involving **replacing** each letter in a text with **another** letter a **fixed number** of **positions** down the alphabet.
- For instance, shifting each letter by "**1**" would change **A** to **B** and **B** to **C**.
- Traditionally, the shift value is **3**, but any number of **shifts** can be applied.
- **Decryption** involves **reversing** the shift by the **same number of positions**.
- While the **Caesar Cipher** is not considered **strong encryption** due to its ease of **decoding**, it remains a part **of more complex encryption methods**.
- Despite its simplicity, this encryption was **valuable** during **Caesar's military campaigns**, preventing intercepted messages from **being** easily understood by **adversaries**.



What is a Caesar Cipher? | IT Wiki | GateKeeper Proximity Passwordless 2FA Solution (gkaccess.com)

Caesar Cipher

- Can define transformation as:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- Mathematically give each letter a number

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Then have Caesar cipher as:

  $c = E(p) = (p + k) \bmod (26)$

  $p = D(c) = (c - k) \bmod (26)$

plaintext letter p, substitute the ciphertext letter C, k takes on a value in the range 1 to 25

Monoalphabetic Techniques

Monoalphabetic Cipher Security

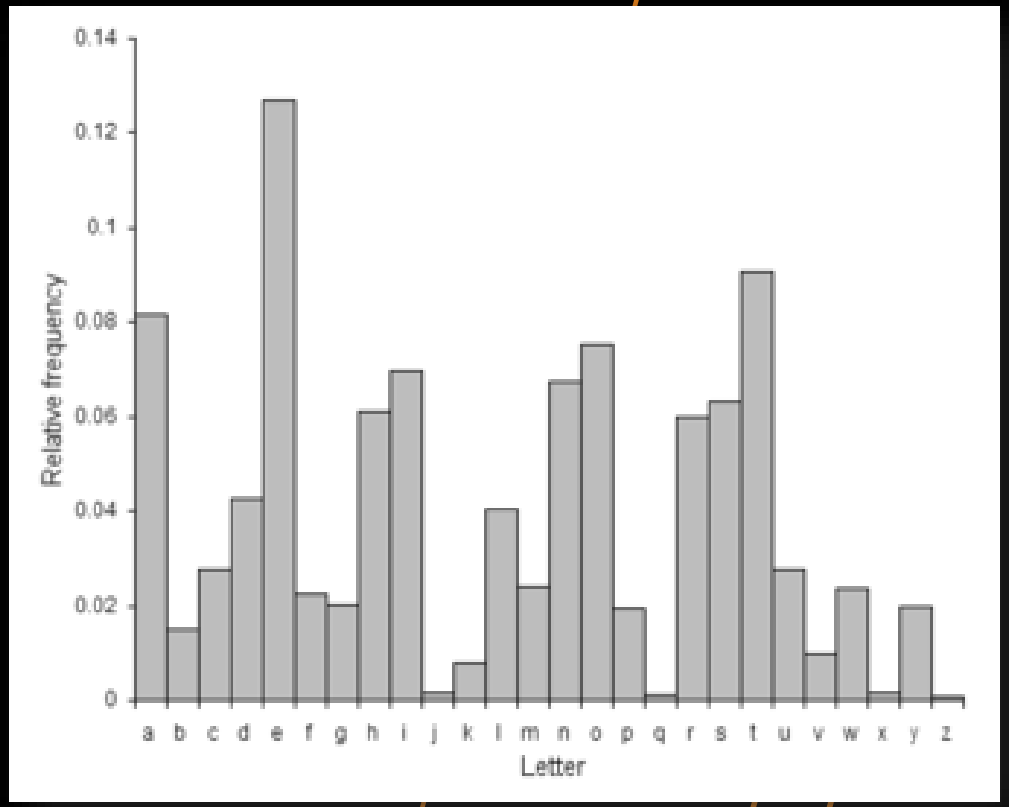
- Rather than just **shifting** the alphabet
- Could **shuffle** (jumble) the **letters arbitrarily**
- Each **plaintext letter** maps to a **different** random ciphertext letter
- Hence key is 26 letters long

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

Plaintext: if we wish to replace letters
Ciphertext: WI RF RWAJ UH YFTSDVF SFUUFYA

Monoalphabetic Cipher Security

- The "cipher" line can be any **permutation** of the **26** alphabetic characters, then there are **26!** or **greater than 4 x 10^{26}** possible keys.
- With so **many keys**, might think is secure
- but it would be **!!!WRONG!!!**



- The problem is language **characteristics**
- We don't actually need all the **letters** in order to understand written ***English text***.
- Human languages are **redundant**
- In English **E** is by far the most **common letter**
  - followed by **T**, **R**, **N**, **I**, **O**, **A**, **S**
- Other letters like **Z**, **J**, **K**, **Q**, **X** are fairly rare

William Stallings, Cryptography and Network Security: Principles and Practice, Fifth Edition

Example Cryptanalysis

Given ciphertext:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMET
SXAIZVUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZ
UHSXEPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

- Count relative letter frequencies

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

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
2	2	0	6	6	4	2	7	1	1	0	0	8	0	9	16	3	0	10	3	0	5	4	5	2	14

- Guess **P** & **Z** are **e** and **t**
- Guess **ZW** is **th** and hence **ZWP** is **the**
- Proceeding with trial and error finally get:

```
it was disclosed yesterday that several informal
but direct contacts have been made with political
representatives of the viet cong in moscow
```

Other Techniques

01. Playfair Cipher

02. Hill Cipher

03. Polyalphabetic Ciphers

04. One-Time Pad

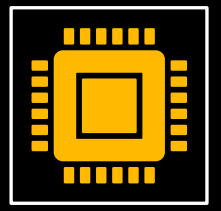
Transposition Ciphers

Transposition Ciphers

Previously discussed techniques examined so far involve the **substitution** of a **ciphertext** symbol for a plaintext symbol.

A very **different kind** of mapping is achieved by performing some sort of **permutation** on the **plaintext** letters.

This technique is referred to as a **transposition** cipher and forms the **second** basic **building** block of ciphers.



The core idea is to **rearrange** the order of basic units (**letters**/**bytes**/**bits**) without **altering** their actual values.

Rail Fence Cipher

- The **rail fence** technique is one of the **simplest** transposition ciphers used for **encryption**.
- In this **method**, the plaintext **message** is arranged in a **zigzag pattern** across multiple "**rails**" or **rows**.
- For instance, to encrypt the message "**meet me after the toga party**" using a rail **fence depth** of **2**, the plaintext is written **diagonally** across two rows.

```
m e m a t r h t g p r y
e t e f e t e o a a t
```

- The encrypted ciphertext is then formed by reading the characters **row** by **row**.
- This technique provides a **basic level** of encryption but may **not** be secure against **advanced cryptanalysis methods**.

MEMATRHTGPRYETEFETEOAAT

Row Transposition Ciphers

- A more complex method **involves arranging** the message in a **rectangular grid**, filling it **row** by **row**.
- The ciphertext is then **generated** by **reading** the **grid column** by **column**, with the order of **columns** altered according to a **predetermined** key.
- Example
  - Plain text: Attack postponed until two am

```
Key 3 4 2 1 5 6 7
```

```
Column number: 1 2 3 4 5 6 7
Plaintext:     a t t a c k p
               o s t p o n e
               d u n t i l t
               w o a m x y z
```

```
Key:           3 4 2 1 5 6 7
Plaintext:     t a t a c k p
               t p s o o n e
               n t u d i l t
               a m o w x y z
```

- then reorder the columns **according** to some key **before reading off** the rows
  - Ciphertext: **TTNAAPTMTSUOAODWCOIXKNLYPETZ**

Rotor Machines

Rotor Machines

- Before modern **ciphers**, **rotor machines** were most common **complex ciphers** in use
- Widely used in **WW2**

German Enigma

Source: Enigma machine - Wikipedia

Allied Hagelin

Source: Hagelin BC-543 (cryptomuseum.com)

Japanese Purple

Secrets Abroad: A History of the Japanese Purple Machine - Wonders & Marvels (wondersandmarvels.com)

- Implemented a very complex, varying **substitution** cipher
- Used a series of **cylinders**, each giving one **substitution**, which **rotated** and **changed** after each letter was encrypted
- With 3 cylinders have **26^3=17576** alphabets

William Stallings, Cryptography and Network Security: Principles and Practice, Fifth Edition

Claude Shannon and Substitution-Permutation Ciphers

- Claude **Shannon** introduced idea of **substitution-permutation** (S-P) networks in **1949** paper
- form basis of modern **block ciphers**
- **S-P** nets are based on the two primitive **cryptographic operations** seen before:
  - ***substitution*** (S-box)
  - ***permutation*** (P-box)
- provide ***confusion*** & ***diffusion*** of message & key

Confusion and Diffusion

- The terms **diffusion** and **confusion** were introduced by Claude Shannon to capture the **two basic building blocks** for any cryptographic system.
- Every **block** cipher involves a transformation of a **block of plaintext** into a **block of ciphertext**, where the transformation depends on the key.
- The **mechanism of diffusion** seeks to make the **statistical relationship** between the **plaintext** and **ciphertext** as **complex** as possible in order to **thwart attempts** to deduce the key.
- Confusion seeks to make the relationship between the **statistics** of the **ciphertext** and the value of the **encryption key** as **complex as possible**, again to **thwart attempts to discover the key**.
- So successful are **diffusion** and **confusion** in capturing the essence of the desired attributes of a block cipher that they have become the **cornerstone** of modern block cipher design.

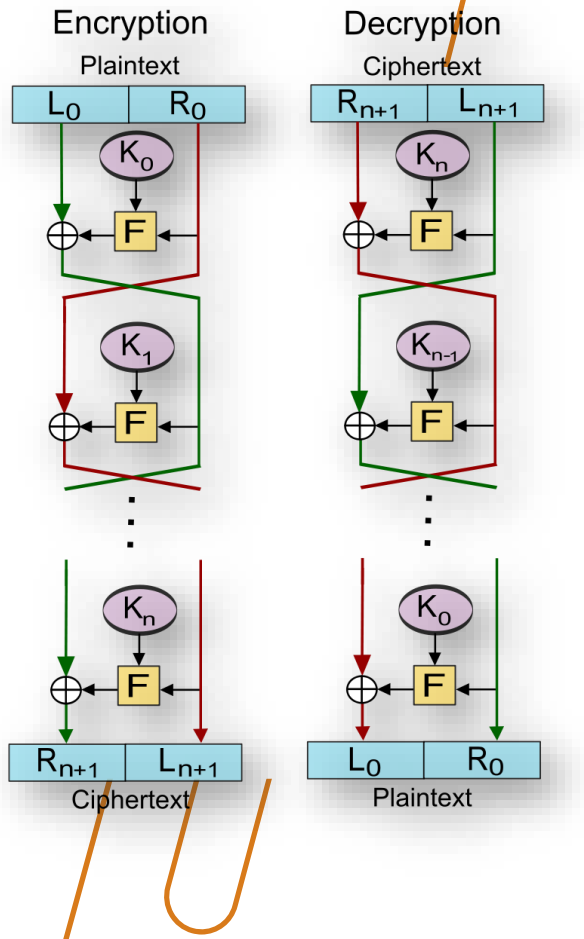
Feistel Cipher Structure

- Horst Feistel, working at **IBM Thomas J Watson Research Labs**, devised a suitable **invertible cipher structure** in the early 70's.
- One of Feistel's main contributions was the **invention** of a **suitable structure** that adapted **Shannon's S-P** network in an easily inverted structure.
- It **partitions** the **input block** into **two halves**, which are processed **through multiple rounds** that perform a **substitution** on the **left data half**, based on the **round function** of the **right half** & **subkey**, and then have **permutation swapping halves**.
- Essentially the same **h/w** or **s/w** is used for **both encryption** and **decryption**, with just a **slight change** in how the **keys are used**.
- One layer of **S-boxes** and the following **P-box** are used to form the round function.

Feistel Cipher Encryption/Decryption

The exact realization of a Feistel network depends on the choice of the following parameters and design features:

- **block size** - increasing size improves security but slows cipher
- **key size** - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **number of rounds** - increasing the number improves security, but slows cipher
- **subkey generation** algorithm - greater complexity can make analysis harder, but slows cipher
- **round function** - greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption -** more recent concern for practical use
- **ease of analysis - for easier validation &** testing of strength



DES Algorithm – Symmetric Cipher

DES Decryption

The **Data Encryption Standard** (**DES**) is a block cipher widely used for **data security**, characterized by a **56-bit** key length.

Despite its historical significance, **DES** has faced increasing vulnerabilities to **powerful attacks over time**.

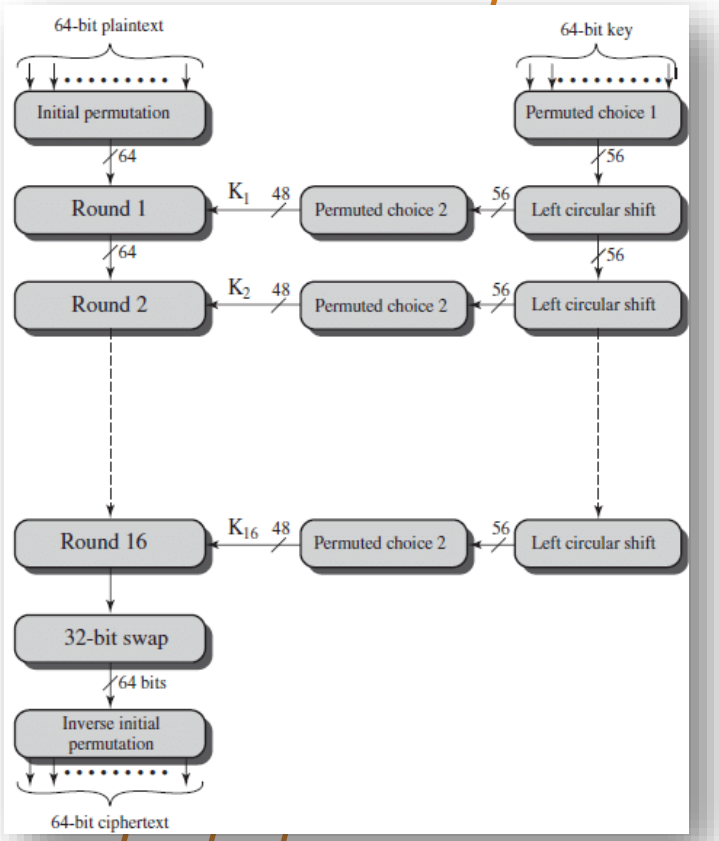
Consequently, the popularity of **DES** has declined due to these security concerns.

DES operates by encrypting data in blocks of **64 bits** each, meaning that **64 bits** of **plaintext** are inputted to produce **64** bits of ciphertext.

Both **encryption** and **decryption** processes in DES utilize the same **algorithm** and **key**, with **minor** differences.

The key length in DES is fixed at **56 bits**.

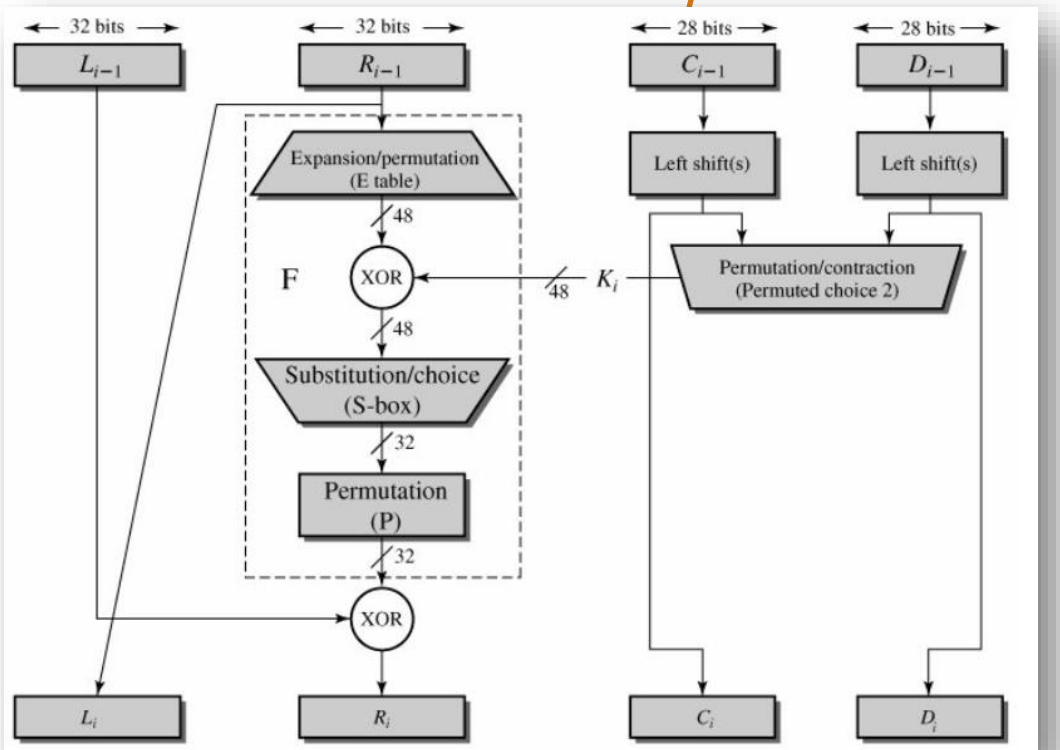
The function expects a **64-bit** key as input. However, only **56** of these bits are ever used; the other 8 bits can be used as parity bits or simply set arbitrarily



William Stallings, Cryptography and Network Security: Principles and Practice, Fifth Edition
Data encryption standard (DES) | Set 1 – GeeksforGeeks, accessed in February 2024.

DES Round Structure

- uses two **32-bit L & R** halves
- as for any Feistel cipher can describe as:
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
  - F takes **32-bit R** half and **48-bit subkey**:
- expands **R** to **48-bits** using perm E
- adds to subkey using **XOR**
- passes through **8 S-boxes** to get **32-bit** result
- finally permutes using **32-bit** perm P



Strength of DES – Key Size

- **56-bit** keys have **2^{56} = 7.2 x 10^{16} values**
- **Brute force** search looks hard
- However **DES** was finally and definitively proved **insecure** in **July 1998**, when the **Electronic Frontier Foundation** (**EFF**) announced that it had broken a DES encryption using a special-purpose "**DES cracker**" machine that was built for less than **$250,000**.
- The attack took **less than three days**.
- We must clearly consider **alternatives** to **DES**, the most important of which are **AES** and **triple DES**.

DES vs Triple DES (3DES)

- **Triple DES** (**TDES** or **3DES**) is an encryption algorithm that applies the **Data Encryption Standard** (**DES**) cipher **three times** successively to encrypt data.
- While DES performs encryption in **16** rounds for each **data block**, **3DES increases** the number of rounds to **48**, enhancing its cryptographic strength.
- Despite being somewhat **stronger** than **DES**, **3DES** has demonstrated **vulnerabilities** in **securing** data transmissions.
- Due to its **susceptibility** to **brute force attacks**, the **National Institute of Standards and Technology** (**NIST**) has **officially prohibited** the use of **3DES beyond 2023**.
- As a result, the **cryptography community** has shifted its **focus** towards the **Advanced Encryption Standard** (**AES**) as a more secure **alternative** to **3DES**.

What Is DES Encryption? A Look at the DES Algorithm - InfoSec Insights (sectigostore.com), accessed in February 2024

Asymmetric Encryption

What is Asymmetric Encryption?

- **Asymmetric encryption**, or **public-key cryptography**, utilizes **two keys** – a **public key**, **shared openly**, and a **private key**, **kept secret.** This method allows for **secure data transmission** without a **shared secret key**.
- The **sender employs** the recipient's **public key** to **encrypt the data**, while the recipient uses their **private key** for **decryption**, ensuring secure communication.
- **Unlike** symmetric encryption, which requires the **exchange of secret keys**, **asymmetric encryption eliminates this need**, simplifying the process, especially in **multi-party** communication.
- Additionally, **asymmetric encryption enables** the **creation** of **digital signatures**, crucial for **verifying data authenticity**.
- Common applications of asymmetric encryption include **secure online communication**, **digital signatures**, and **secure data transfer**.
- Examples of asymmetric encryption algorithms include **RSA**, **Diffie-Hellman**, and **Elliptic Curve Cryptography** (**ECC**).

What is Asymmetric Encryption? – GeeksforGeeks, accessed in February 2024

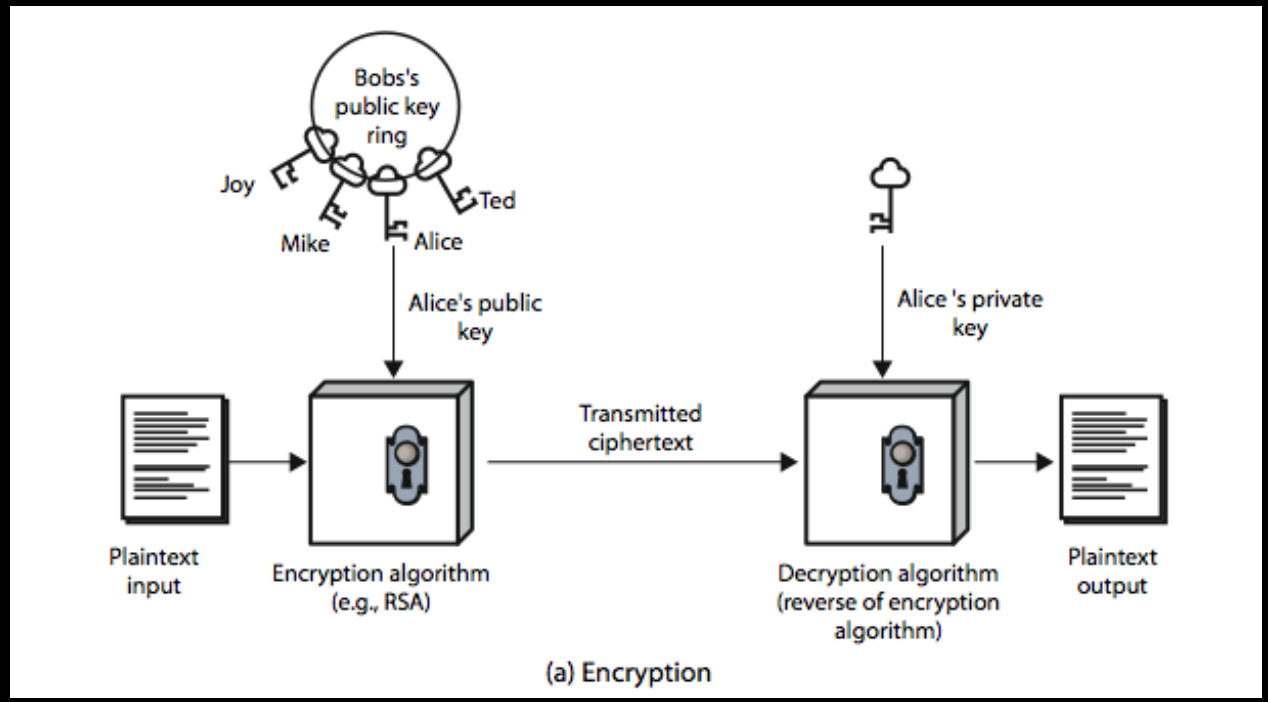
Advantages of Asymmetric Encryption

- **Enhanced Security**: **Unlike symmetric encryption**, where **one key** is used for both **encryption** and **decryption**, asymmetric encryption utilizes **different keys** for each process. The **private key,** used for **decryption**, **remains secret**, making it **challenging for attackers** to **intercept** and **decrypt data**.
- **Authentication**: **Asymmetric encryption** facilitates **authentication** by allowing the **receiver** to verify the **sender's identity**. The **sender encrypts** a **message with their private key,** which can only be decrypted using their **public key. Successful decryption confirms** the **sender's identity**.
- **Non-repudiation**: Asymmetric encryption **ensures non-repudiation**, preventing the **sender** from **denying sending a message** or **altering** its content. **Messages encrypted** with the **sender's private key** can only be **decrypted with their public key**, providing assurance of the sender's identity and message integrity.
- **Key Distribution**: Unlike **symmetric encryption**, which requires a secure **key distribution** system, **asymmetric encryption** eliminates this need. The public key can be **openly shared**, while the **private key remains** secret, simplifying key management.
- **Versatility**: Asymmetric encryption finds applications in various **fields**, including secure **email communication**, **online banking transactions**, **e-commerce**, and securing SSL/TSL connections for **internet traffic**.

What is Asymmetric Encryption? – GeeksforGeeks, accessed in February 2024

Public-Key Cryptography

A public-key encryption scheme has six ingredients: plaintext, encryption algorithm, public & private keys, ciphertext & decryption algorithm.



Public-Key Applications

Can classify uses into 3 categories:

- **Encryption/decryption:** The sender **encrypts** a message with the **recipient's public key**.
- **Digital Signatures:** The sender "**signs**" a message with its **private key**, either to the **whole message** or to a **small block** of **data** that is a function of the message.
- **Key Exchange:** Two sides cooperate to exchange a **session key**. Several different approaches are possible involving the **private** key(s) of **one** or **both** parties.

Some **algorithms** are suitable for all **three applications**, whereas others can be used only for one or two of these applications.

Security of Public Key Schemes

- Like **private key** schemes brute force **exhaustive search** attack is always theoretically possible
- But keys used are **too large** (>**512bits**)
- Security relies on a **large enough** difference in difficulty between **easy** (**en/decrypt**) and **hard** (**cryptanalysis**) problems
- More generally the **hard** problem is known, but is made hard enough to be **impractical** to break
- Requires the use of **very large numbers**
- Hence is **slow** compared to **private key schemes**

RSA Algorithm – Asymmetric Cipher

RSA

- RSA is the best known, and by far the most widely used **general public key encryption algorithm**, and was first published by **Rivest**, **Shamir** & **Adleman** of **MIT** in **1978**.
- Since that time, RSA has reigned supreme as the **most widely accepted** and implemented **general-purpose** approach to **public-key encryption**.
- Uses large integers (eg. **1024** bits).
- Its security is due to the cost of factoring large numbers.
- **Prime Numbers** play an essential role in **RSA**.
- If we have the number **30**, which numbers can be multiplied to give the same result?
  - 15 x 2
  - 3 x 10
  - 5 x 6
  - So we have multiple options to reach 30.
- However, if I repeat the same question with **35**.
- **5 x 7** is the only answer, according to "**A prime number is a natural number greater than 1 that has no positive integer divisors other than 1 and itself**."

Prime Numbers | Brilliant Math & Science Wiki, accessed in February 2024
William Stallings, Cryptography and Network Security: Principles and Practice, Fifth Edition

RSA Idea

- The **RSA encryption scheme** is founded on the challenge of **factoring large integers**, making it hard to **decipher**.
- Public keys in **RSA** comprise **two numbers**, one of which is the product of **two large prime numbers**, while the **private key** is derived from the **same primes**.
- Security in RSA hinges on the difficulty of factoring the large number, ensuring **private key protection**.
- Encryption strength is **directly proportional** to the **key size**, with **doubling** or **tripling** exponentially **increasing** encryption **robustness**.
- RSA keys are typically **1024** or **2048** bits in length, with concerns over the **vulnerability** of **1024-bit** keys in the future.
- Despite **experts' predictions**, breaking **1024-bit keys** remains an **unfeasible** task at present.

RSA Algorithm in Cryptography – GeeksforGeeks, accessed in February 2024

The Crucial Roles of RSA in Internet Security

- RSA plays two crucial roles in today's internet.
  - Firstly, it's used in over **90% of internet connections** during the **SSL handshake**, which initiates secure communication.
  - This handshake is a key moment where an **attack** could **jeopardize** the **entire session**, potentially **exposing** sensitive **information** such as **personal data**, **financial records**, and **intellectual property**.
  - Another critical function of RSA is generating cryptographic **digital signatures**. These signatures are used for various purposes, such as **authenticating emails**, **documents**, and **software updates**. When a **file** or **program** is **digitally signed**, it is trusted by **computers** and **mobile** devices. Failure at this point could lead to serious consequences.

RSA Is Dead — We Just Haven't Accepted It Yet (forbes.com), accessed in February 2024

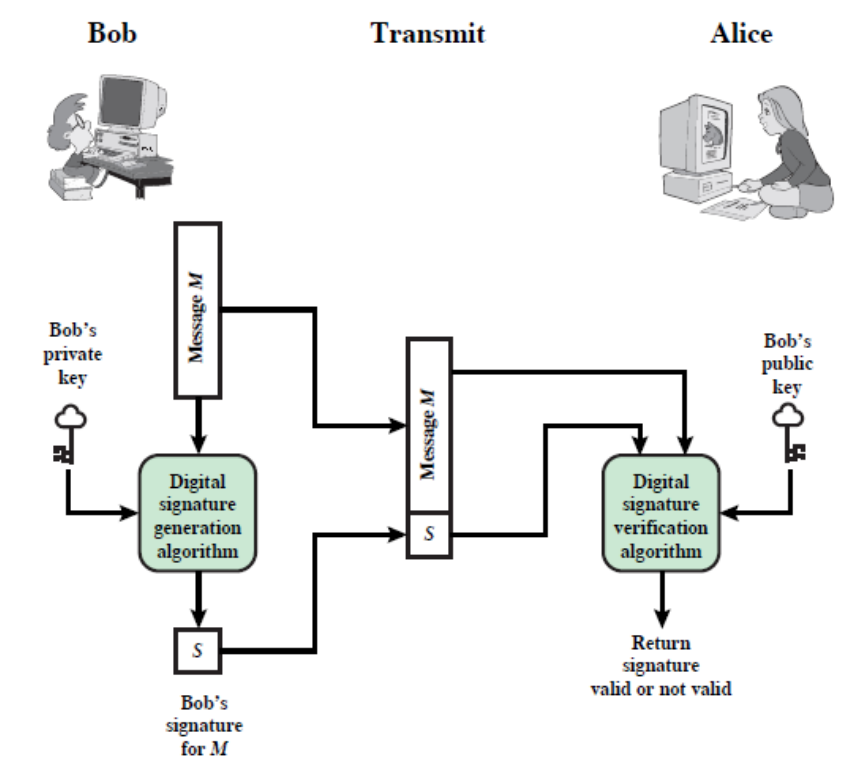
Digital Signature

Digital Signature

- The most important development from the work on public-key cryptography is the digital signature.
- A digital signature is **analogous to a handwritten** signature and provides a set of **security capabilities** that would be difficult to implement in any other way.
- A **digital signature** serves as an **authentication** method, allowing the message creator to **attach a code functioning as a signature**.
- It is generated by hashing the message and then encrypting it with the **creator's private key**.
- The signature **ensures** the message's **source** and **integrity**.

Digital Signature Model

- Bob can generate a **digital signature** for a message using an algorithm.
- This process involves **Bob's private key** and the message as **inputs**.
- Other users, like **Alice**, can verify the **signature**.
- Verification requires the **message**, **signature**, and **Bob's public key** as **inputs**.



Connect with CyberSecPro:
How to register and other practical information

1. Website:
   www.cybersecpro-project.eu
2. X (Twitter):
   https://twitter.com/CyberSecPro_eu
3. LinkedIn:
   https://www.linkedin.com/company/cybersecpro-euproject/

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

Thank you

Please send all questions to:
Abdelkader Shaaban
abdelkader.Shaaban@ait.ac.at
Stefan Schauer
Stefan.Schauer@ait.ac.at