

Block Cipher Modes - Symmetric Encryption

Summary

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	•General-purpose block-oriented transmission •Useful for high-speed requirements

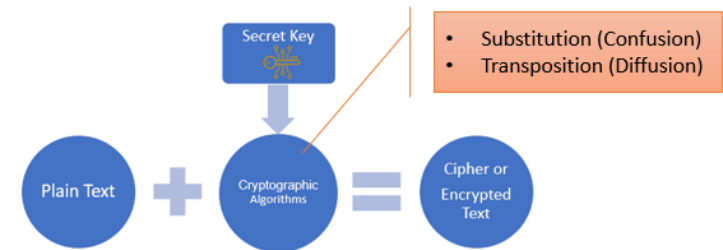
Attack Strategies on Symmetric Encryption

Brute-force Attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
- On average half of all possible keys must be tried to achieve success
- Number of Keys are dictated by the key size.

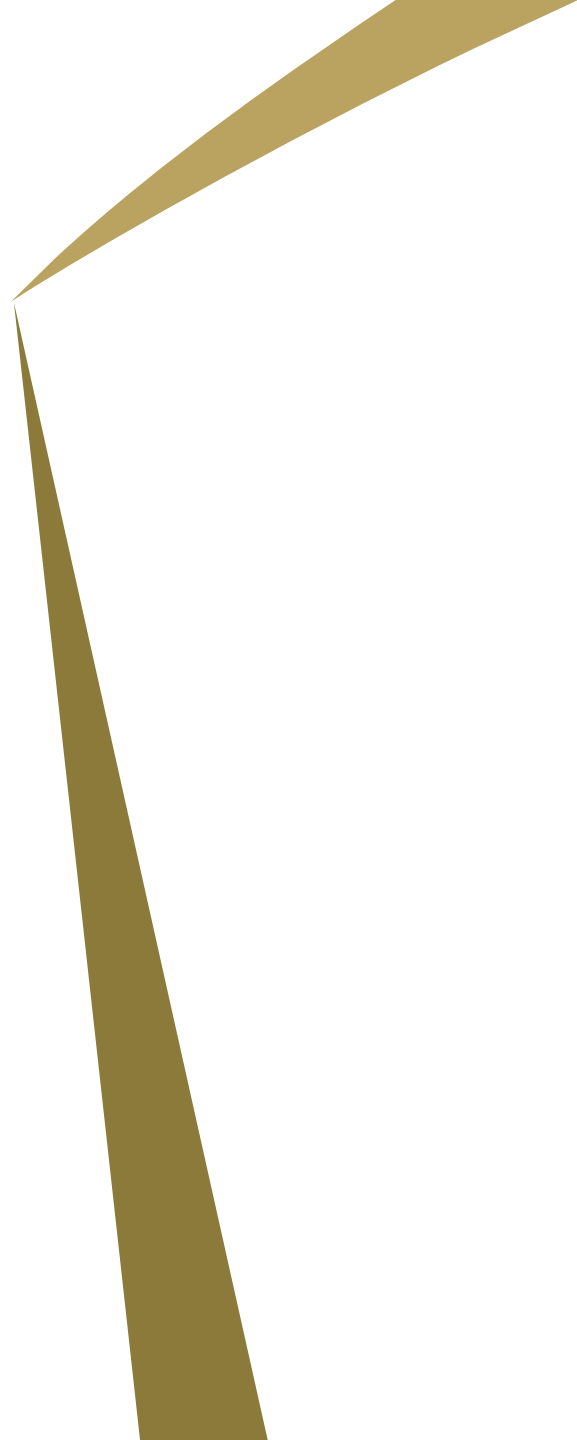
Cryptanalytic Attacks

- Rely on:
 - Nature of the algorithm
 - Some knowledge of the general characteristics of the plaintext
 - Some sample plaintext-ciphertext pairs
- Exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
 - If successful, all future and past messages encrypted with that key are compromised



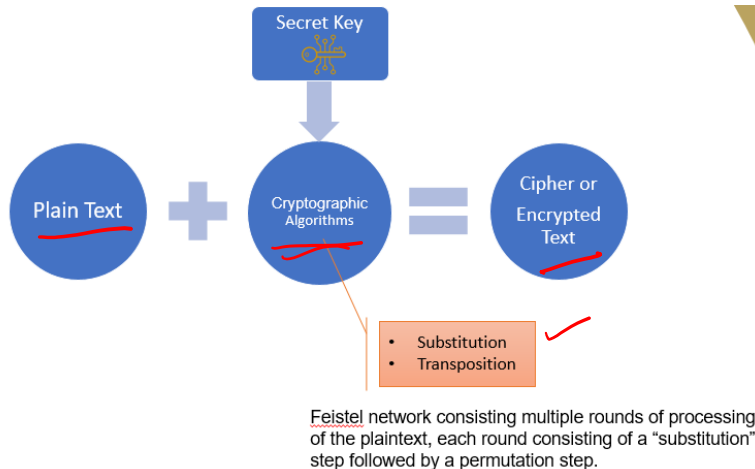
Feistel Network

Symmetric Encryption

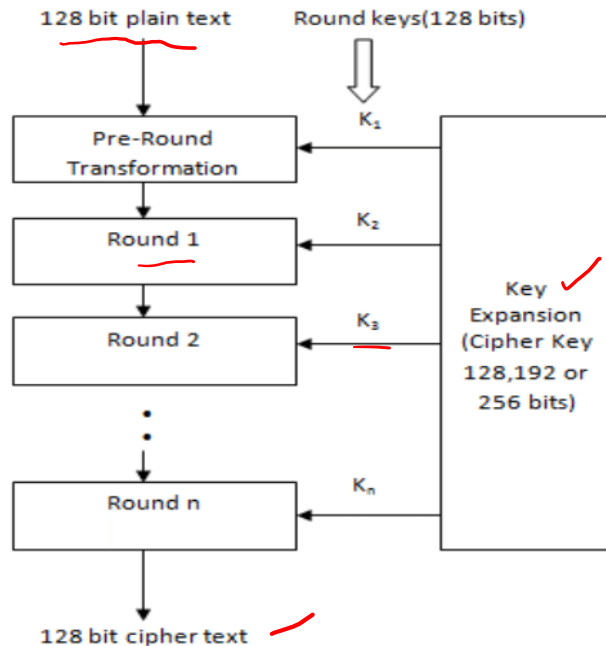
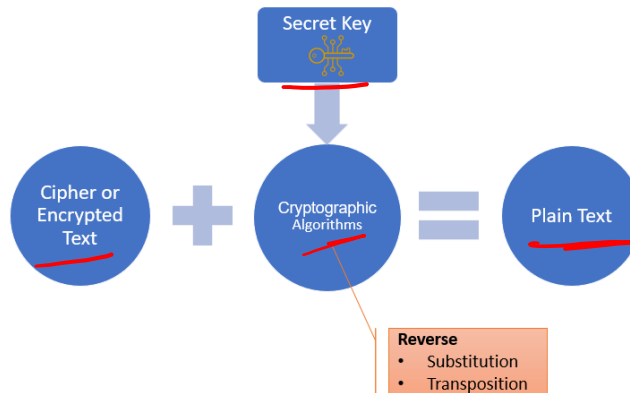


Encryption/Decryption Process

Encryption Process



Decryption Process



We want to make encryption process hard for attacker to break our algorithm...How?

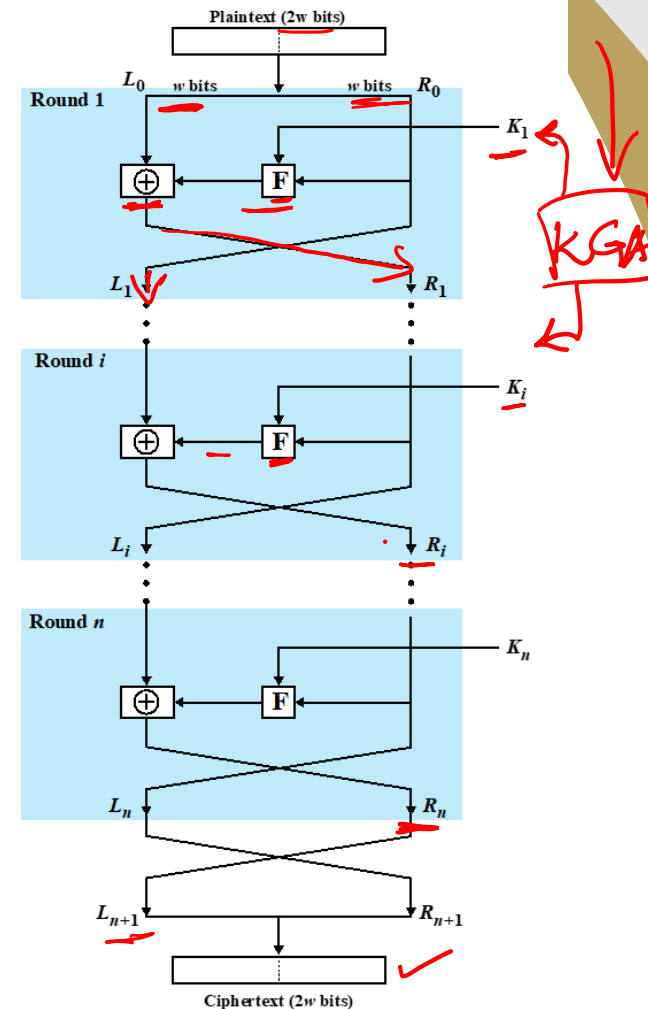
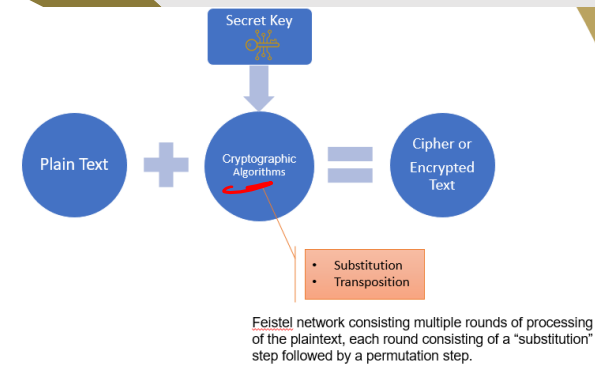
More rounds, complex substitution and transposition...

But, users with the Key should be able to decrypt it easily...How?

Feistel Network

Feistel Network - Symmetric Encryption

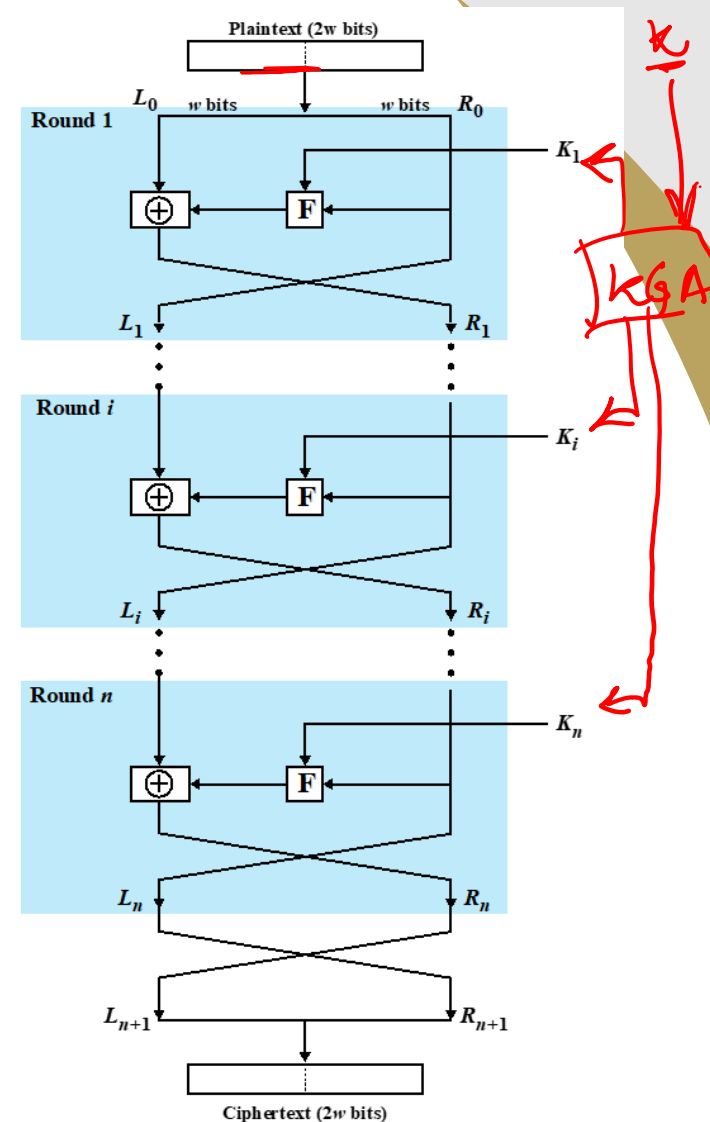
- Many encryption algorithms incorporate a structure first proposed by Horst Feistel of IBM (1973); a **Feistel Network**.
- Feistel structure consisting multiple rounds of processing of the plaintext, each round consisting of a “substitution” step followed by a permutation step.
- Many advance encryption algorithms are designed using this Feistel structure
 - This includes the DES standard.



Feistel Network –

Brief explanation

- The inputs to the encryption algorithm are a plaintext block of length $2w$ bits and a key K .
- The plaintext block is divided into two halves, L^0 and R^0 .
- The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block.
- Each round i has as inputs L^{i-1} and R^{i-1} , derived from the previous round, as well as a subkey K^i , derived from the overall K .
- In general, the subkeys K^i are different from K and from each other and are generated from the key by a subkey generation algorithm.



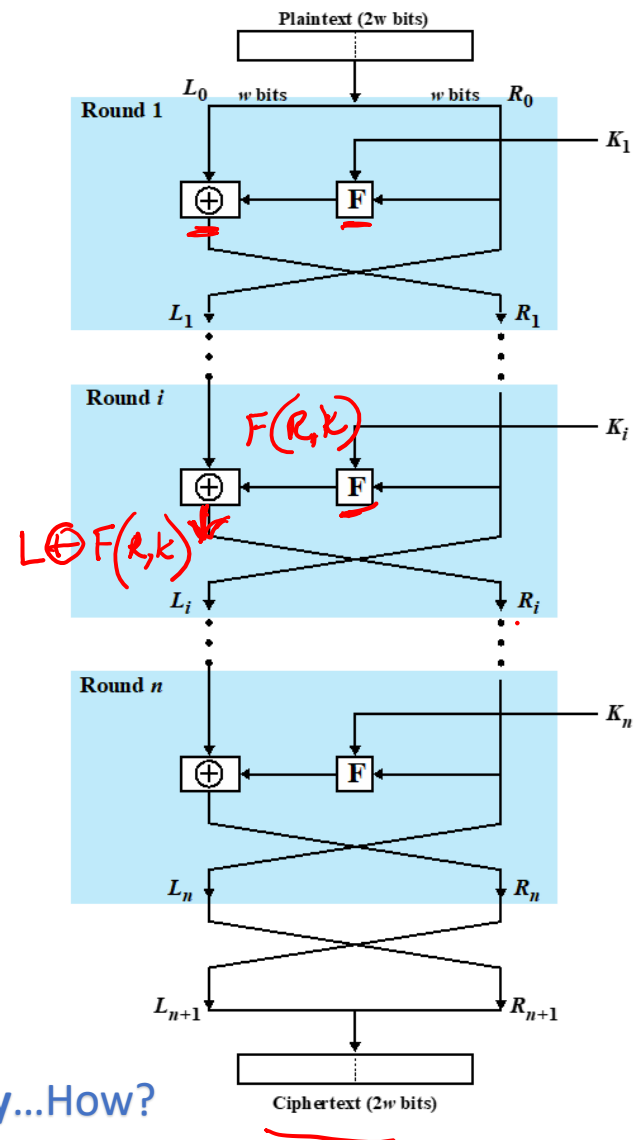
Feistel Network –

Brief explanation

- All rounds have the same structure. A substitution is performed on the left half of the data.
- This is done by applying a round function F to the right half of the data and then taking the exclusive-OR (XOR) of the output of that function and the left half of the data.
- The round function has the same general structure for each round but is parameterized by the round subkey K^i .
- Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.

What's happening inside round function “F”?

Users with the Key should be able decrypt it **easily**...How?



Feistel Network

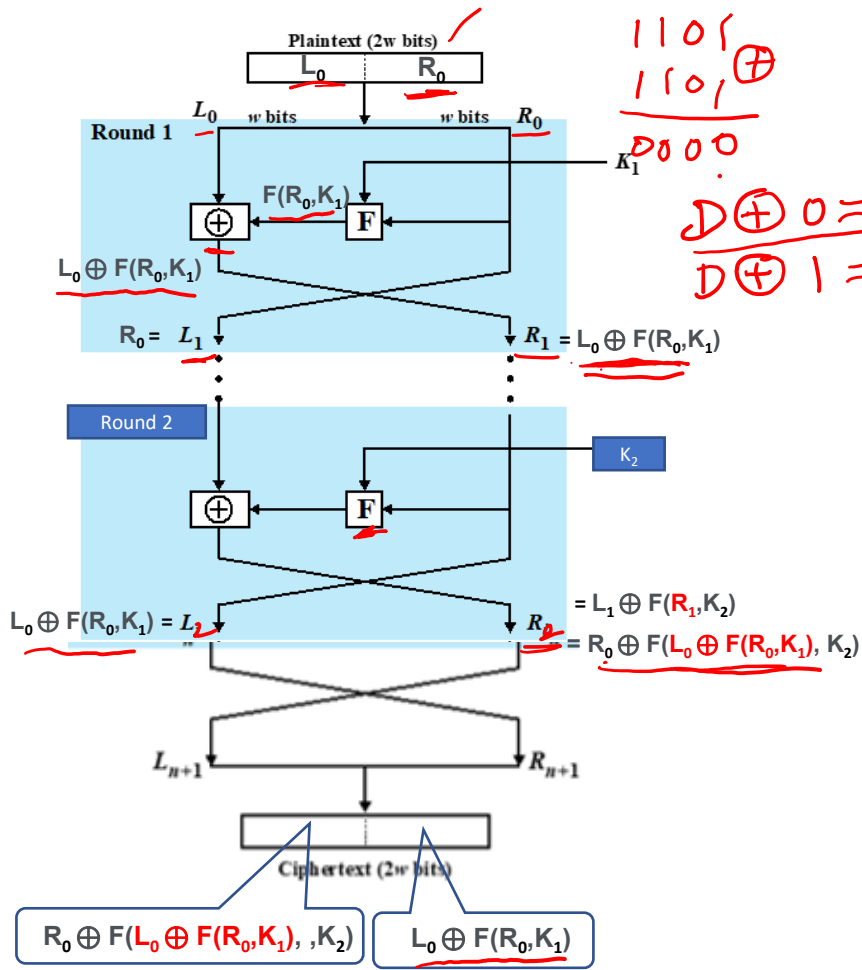
Encryption/Decryption Process

What's happening inside round function “F”?

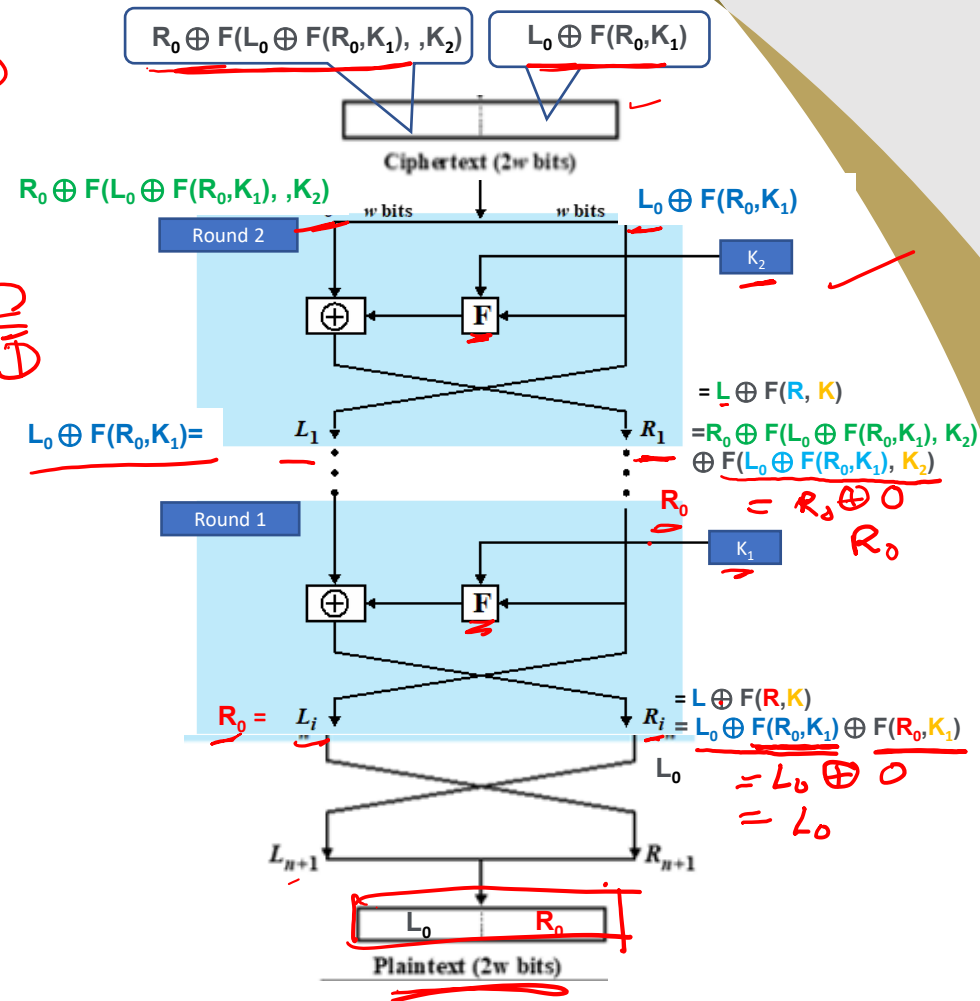
Users with the Key should be able decrypt it
easily...How?

Feistel Network

Encryption Process



Decryption Process



Data Encryption Standard (DES)

Symmetric Encryption Algorithm

Data Encryption Standard (DES)

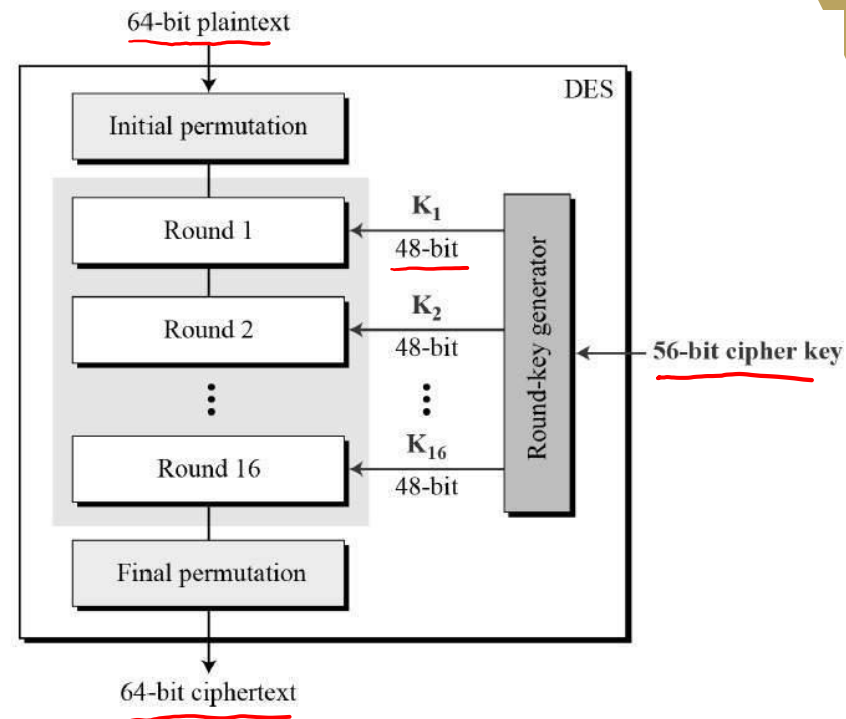
Symmetric Encryption Algorithm

Until recently was the most widely used encryption scheme

- Federal Information Processing Standards FIPS PUB 46
- Referred to as the Data Encryption Algorithm (DEA)
- DES is the most studied encryption algorithm
- DES is a Feistel Cipher.

Working

- **64 bit plaintext block**
- **56 bit key**
- to produce a **64 bit ciphertext block**
- Total **16 rounds**
- Concerns about the use of a 56-bit key
 - Prone to brute-force attack...



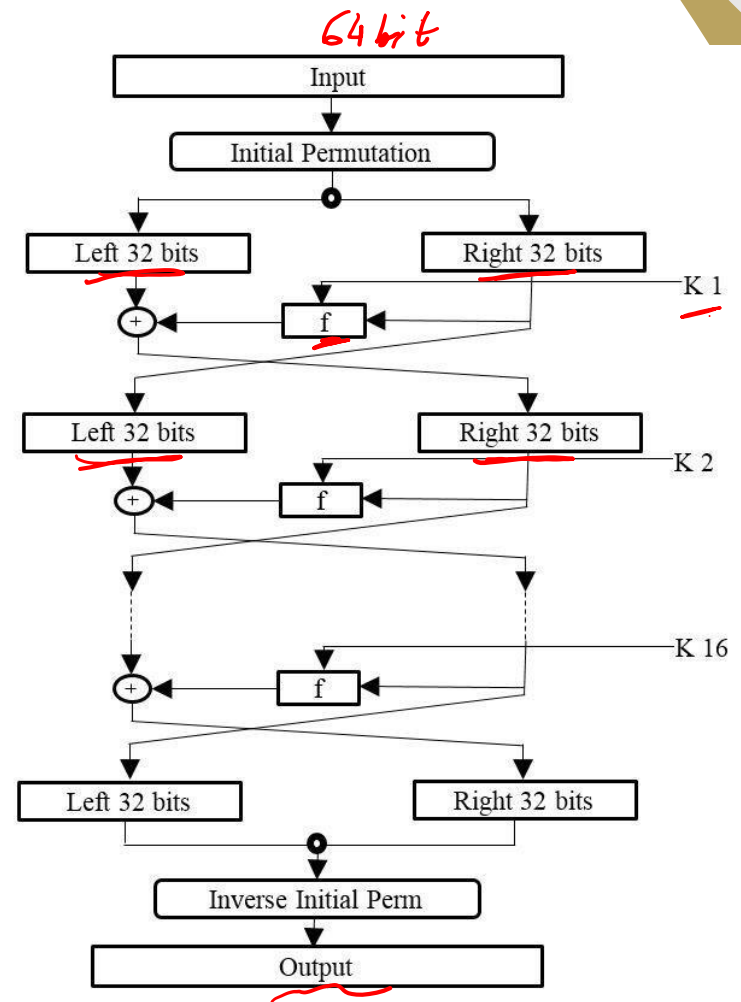
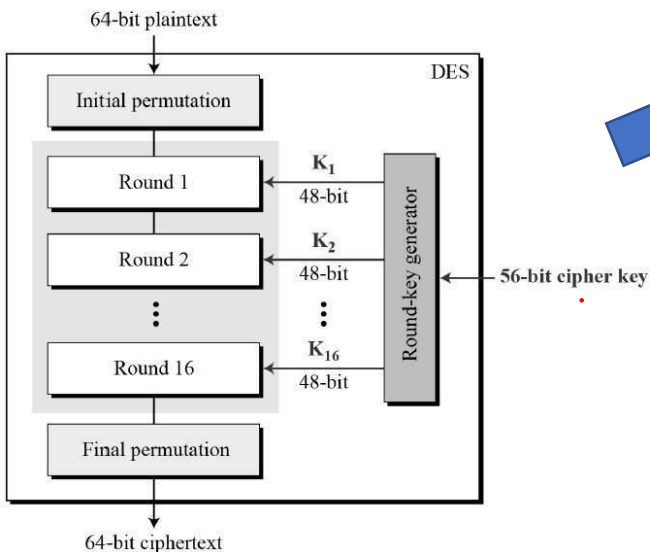
Data Encryption Standard (DES)

Symmetric Encryption Algorithm

- DES is the most studied encryption algorithm
- DES is a Feistel Cipher.

Working

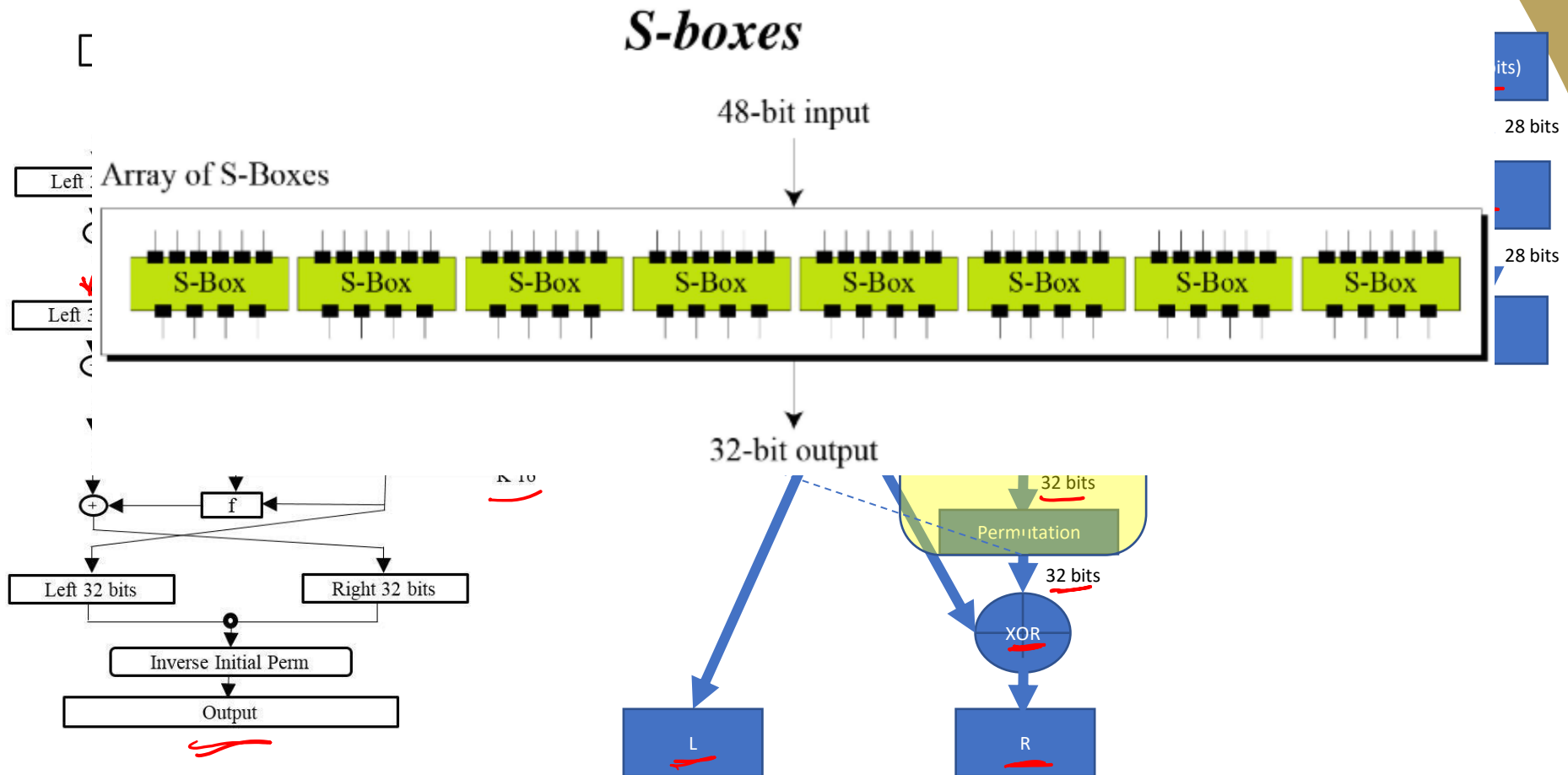
- **64 bit plaintext block**
- **56 bit key**
- to produce a **64 bit ciphertext block**
- **Total 16 rounds**



Data Encryption Standard (DES)

What happens in each round?

- 16 Rounds.
- In each round we apply:
 - ➔ Expansion Box. ✓
 - ➔ Substitution Box.
 - ➔ XOR with the round key.



Data Encryption Standard (DES)

How strong?

Concerns about the use of a 56-bit key

Brute-force attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
- On average half of all possible keys must be tried to achieve success
- Number of Keys are dictated by the key size. 2^{56} 2^{55}

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
<u>56</u>	<u>DES</u>	<u>$2^{56} \approx 7.2 \times 10^{16}$</u>	<u>2^{55} ns = 1.125 years</u>	<u>1 hour</u>

$$\begin{aligned} &= \frac{2^{16} \times 3.6 \times 10^{10}}{10^9} = \frac{3.6 \times 10^{27} \text{ sec}}{3.2 \times 10^{27}} \end{aligned}$$

Think...what can you do to make it strong??

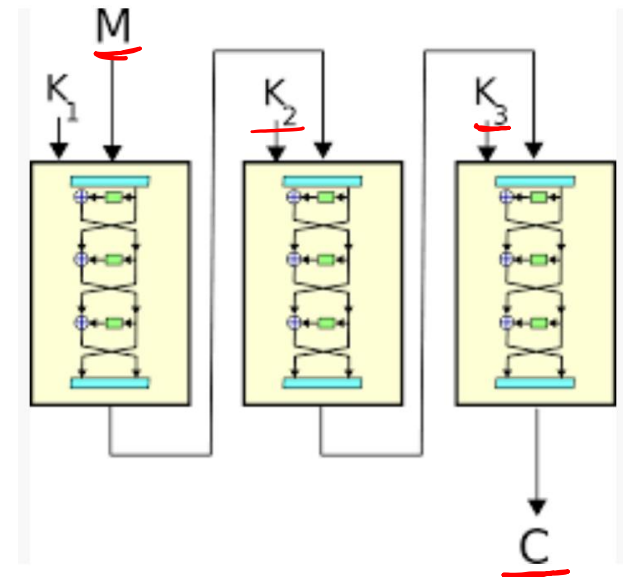
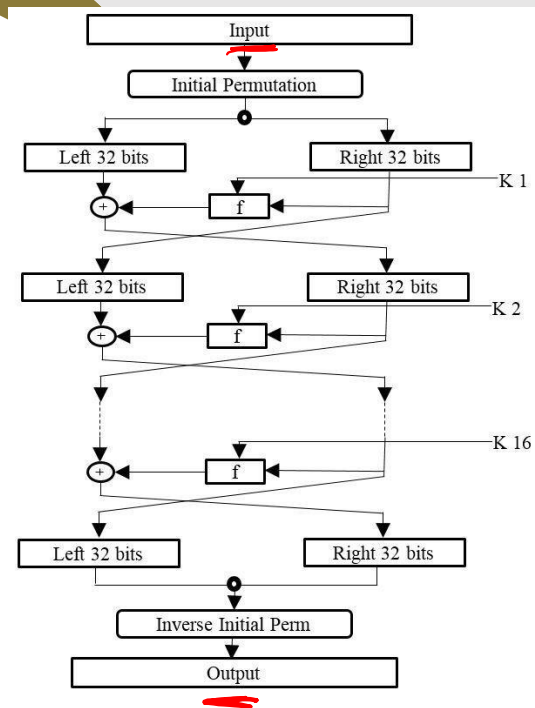
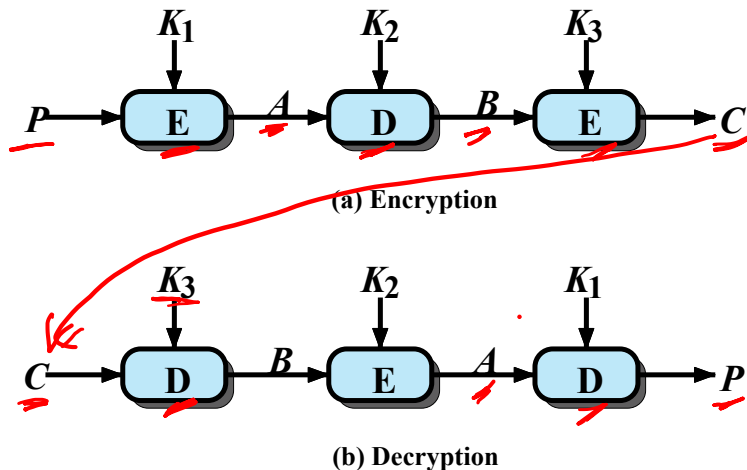
Triple DES (3DES)

Symmetric Encryption Algorithm

Triple DES (3DES)

Symmetric Encryption Algorithm

- Designed to address security concerns over DES
 - Original DES soon become prone to brute-force attack
- Need more secure encryption algorithms
- Repeats basic DES algorithm three times using either two or three unique keys
- First standardized for use in financial applications in ANSI standard X9.17 in 1985



Triple DES (3DES)

Symmetric Encryption Algorithm

Attractions:

- With three distinct keys, 3DES has an effective key length of 168 bits. = ~~3~~ × 56
- 168-bit key length overcomes the vulnerability to brute-force attack of DES
- Underlying encryption algorithm is the same as in DES

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
<u>168</u>	Triple DES	<u>$2^{168} \approx 3.7 \times 10^{50}$</u>	2^{167} ns = 5.8×10^{33} years	<u>5.8×10^{29} years</u>

Drawbacks:

- Algorithm is sluggish in software
- Uses a 64-bit block size

Advanced Encryption Standard (AES)

Symmetric Encryption Algorithm

Advanced Encryption Standard (AES)

Symmetric Encryption Algorithm



Needed a replacement for 3DES

3DES was not reasonable for long term use

NIST called for proposals for a new AES in 1997

Should have a security strength equal to or better than 3DES

Significantly improved efficiency

Symmetric block cipher

128 bit data and 128/192/256 bit keys

Proposal submissions

In a first round of evaluation, 15 proposed algorithms were accepted.

Second round narrowed the field to 5 algorithms.

Selected Rijndael in November 2001

Published as FIPS 197

AES is now widely available in commercial products.

Advanced Encryption Standard (AES)

Symmetric Encryption Algorithm

- An iterative rather than Feistel cipher
 - Processes data as block of 4 columns of 4 bytes
 - Operates on entire data block in every round

- Designed to be:
 - Resistant against known attacks
 - Speed and code compactness on many CPUs
 - Design simplicity

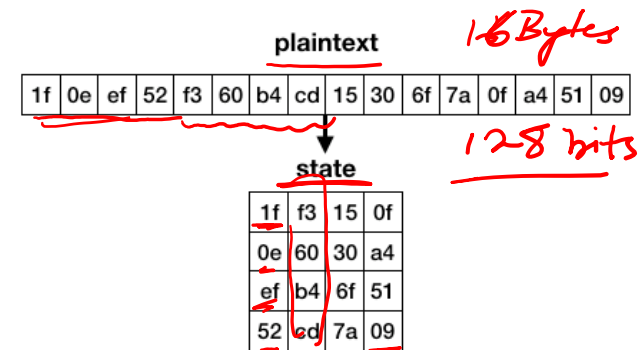
- Data block viewed as 4-by-4 table of bytes
 - Such a table is called the current state

- Key is expanded to array of words

- Has 10 rounds in which state the following transformations (called 'layers'):

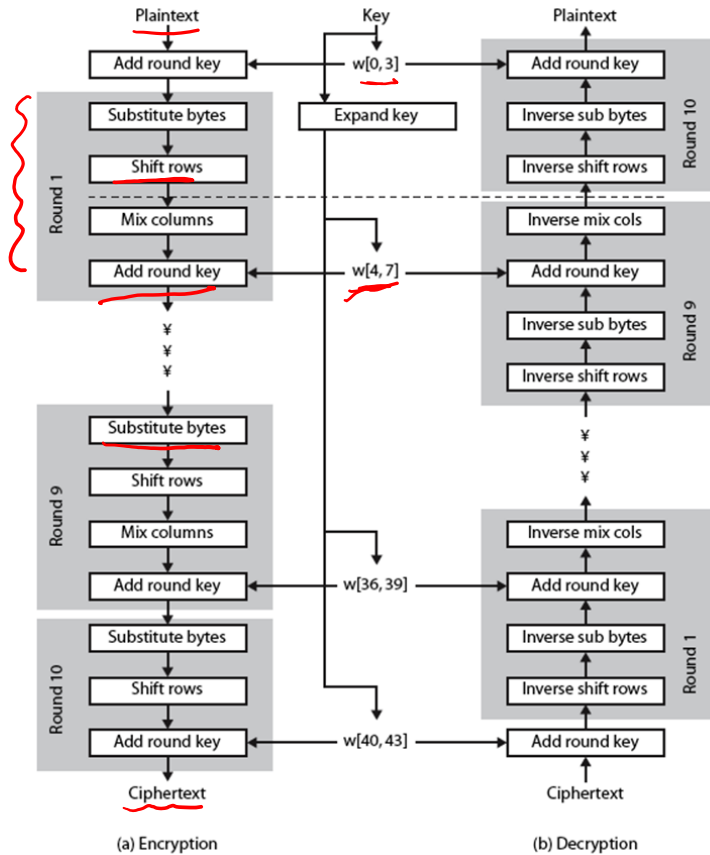
- BS- byte substitution (1 S-box used on every byte)
- SR- shift rows (permute bytes between groups/columns)
- MC- mix columns (uses matrix multiplication in GF(256))
- ARK- add round key (XOR state with round key)

- First and last round are a little different

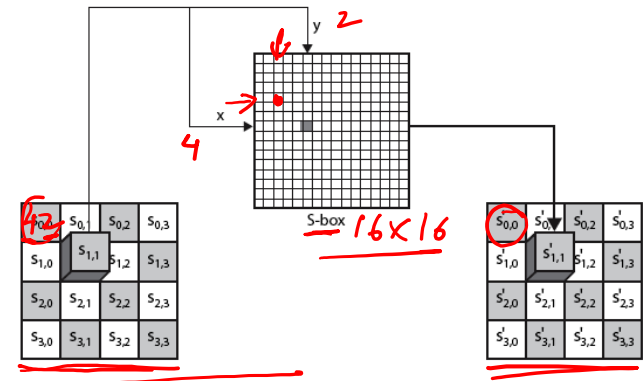


Algorithm	Key length (bits)	Block size (bits)	No of round
AES- 128	128	128	10
AES- 192	192	128	12
AES- 256	256	128	14

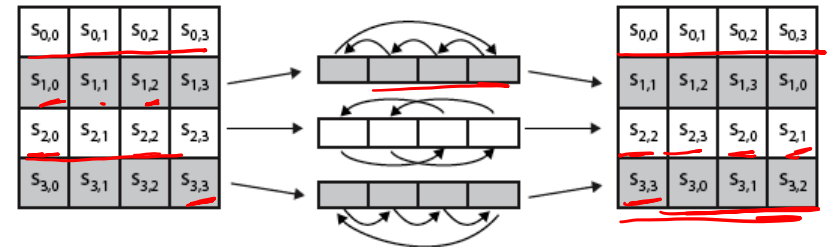
AES Working



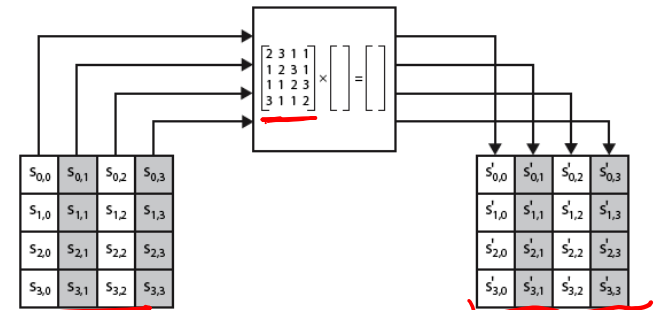
BS



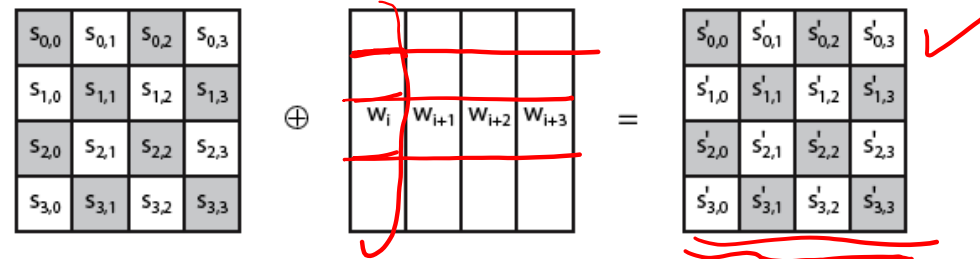
SR



MC

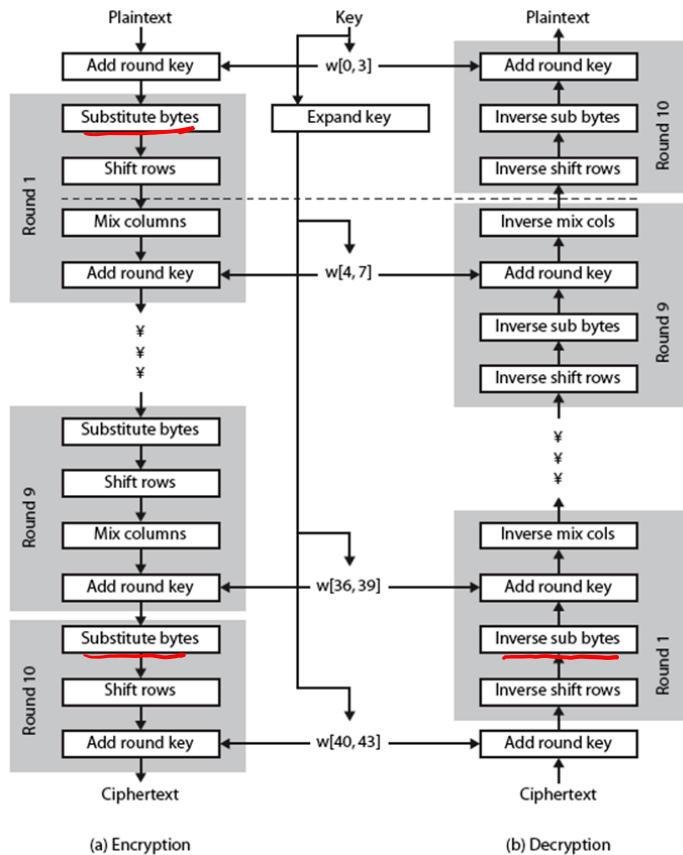


ARK



AES Working

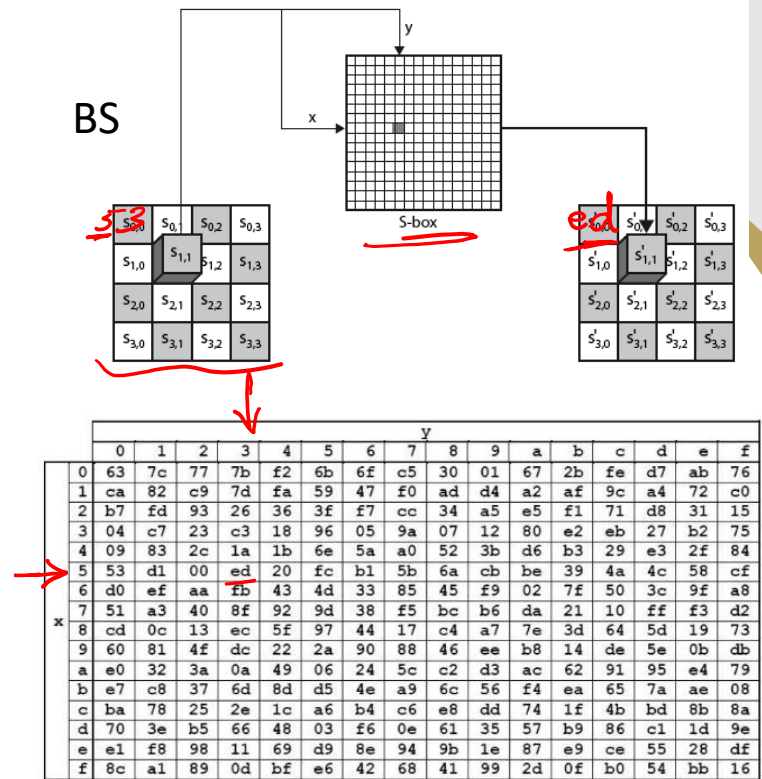
Byte Substitution



(a) Encryption

(b) Decryption

BS



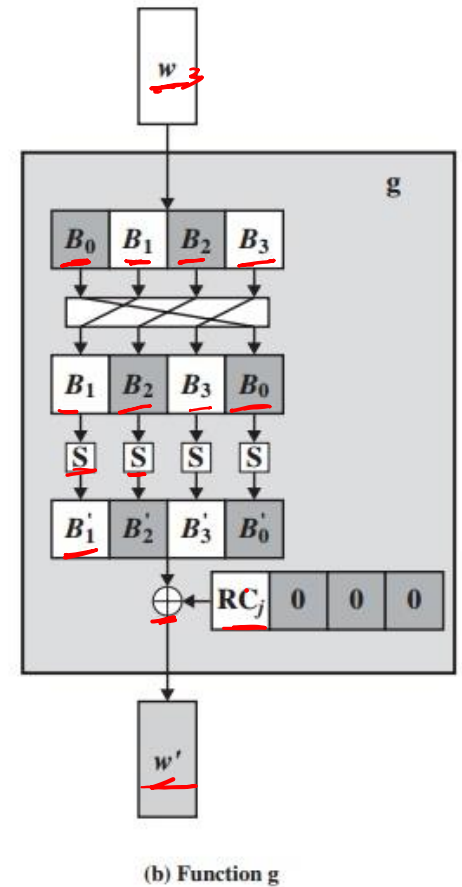
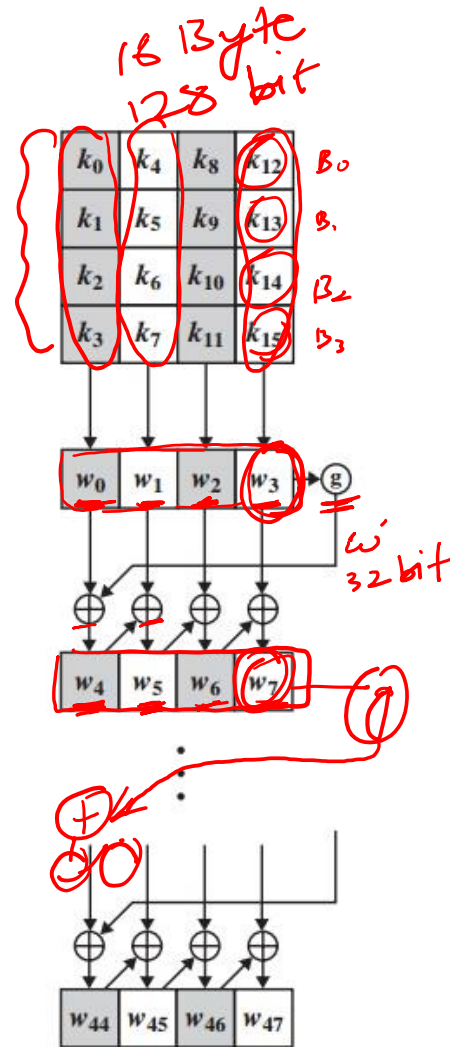
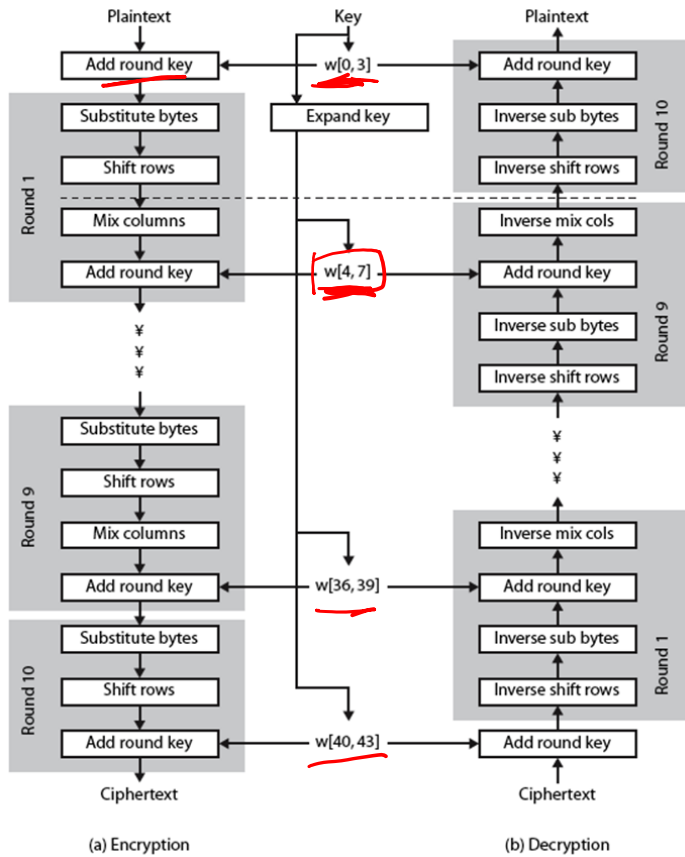
AES S-box

		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

AES – Inverse S-box

AES Sub-Key Generation

Key Expansion



Public-Key Cryptography

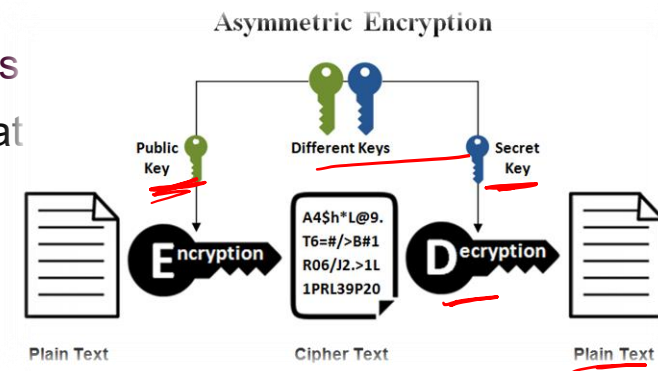
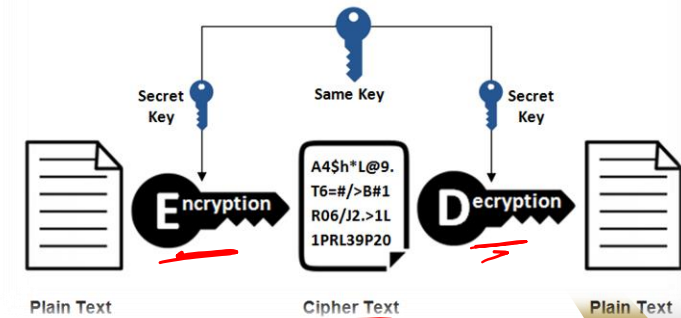
Asymmetric Encryption

Public-Key Cryptography

Asymmetric Encryption

Why Public-Key?

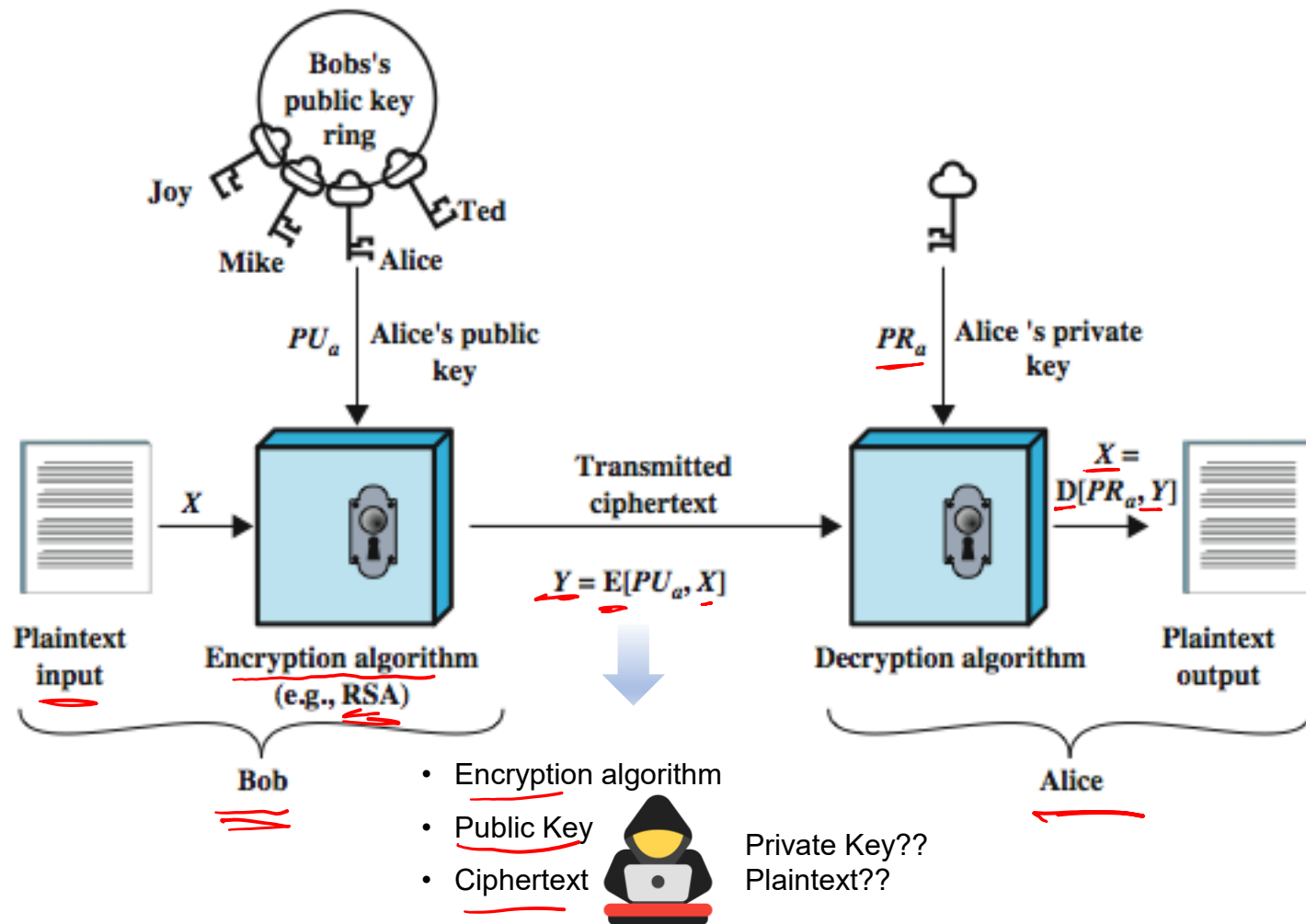
- Developed to address two major issues:
 - key distribution** – how to have secure key exchange among communicating parties
 - digital signatures** – how to verify a message comes intact from the claimed sender
- Public-Key Cryptography was designed to address these concerns
- Public invention due to Whitfield Diffie & Martin Hellman at
 - known earlier in classified community



- Probably most significant advancement in the history of cryptology
- Uses two keys – a public & a private key
- Asymmetric since different keys are used for encryption and decryption
- Uses clever application of number theoretic concepts to function
- This is not meant to replace Symmetric Encryption (private key crypto) but rather to complement...

Public-Key Cryptography

How public and private keys are used?



Public-Key Cryptography

Example of public and private keys

Symmetric

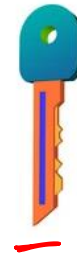


$10 \times 10 \times 10 \times 10$
 $= 10,000$



private

public



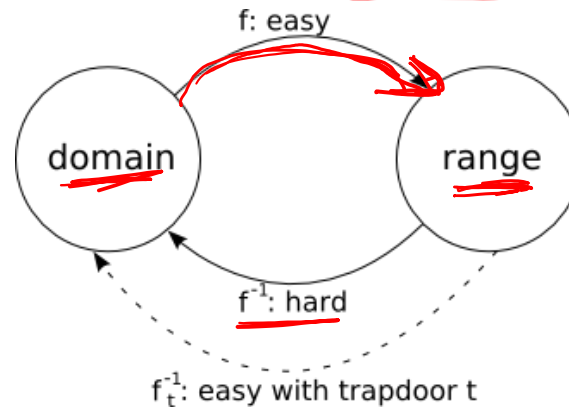
One-way Functions

Base for Modern Cryptography

One-way Functions

Base for Modern Cryptography

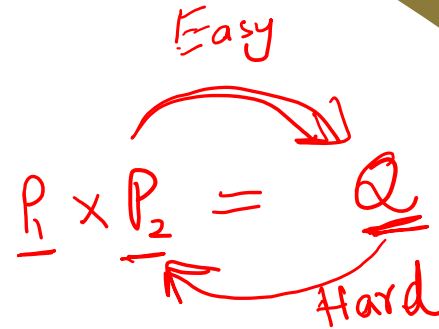
- A **one-way function** is “easy” to compute and “difficult” to reverse.



- The process of **exponentiation** means raising numbers to a power.
- Raising **2** to the power **3**, normally denoted 2^3 just means multiplying **2** by itself **3** times. In other words:
 - $2^3 = (2 \times 2 \times 2) = 8$
 - $a^x = \underbrace{a \times a \times a \times \dots \times a}$
- Modular exponentiation** means computing a^x mod some other number n .
 - $a^x \bmod n$.
- In other words, given a number a and a power x , the function $f(x) = \underbrace{a^x \bmod n}$ is a one-way function.
 - For $x=b$, we can easily compute $y=f(b)=\underbrace{a^b \bmod n}$. But its reverse is : (
 - For a particular $y=c$, its difficult to get respective x Such that $c=f(b)=a^x \bmod n$

One-way Functions

Base for Modern Cryptography



Another Example

- It is easy to take two prime numbers and multiply them together.
- If the numbers are small, we can do this in our heads, on a piece of paper.
- What if numbers get bigger and bigger?
- Multiplication of two big prime numbers is **believed** to be a one-way function.
- Check Wikipedia for RSA Numbers
- https://en.wikipedia.org/wiki/RSA_numbers#:~:text=In%20mathematics%2C%20the%20RSA%20numbers,prime%20factors%20of%20each%20number.
- One of the modern asymmetric algorithm RSA is based on this idea

RSA Algorithm

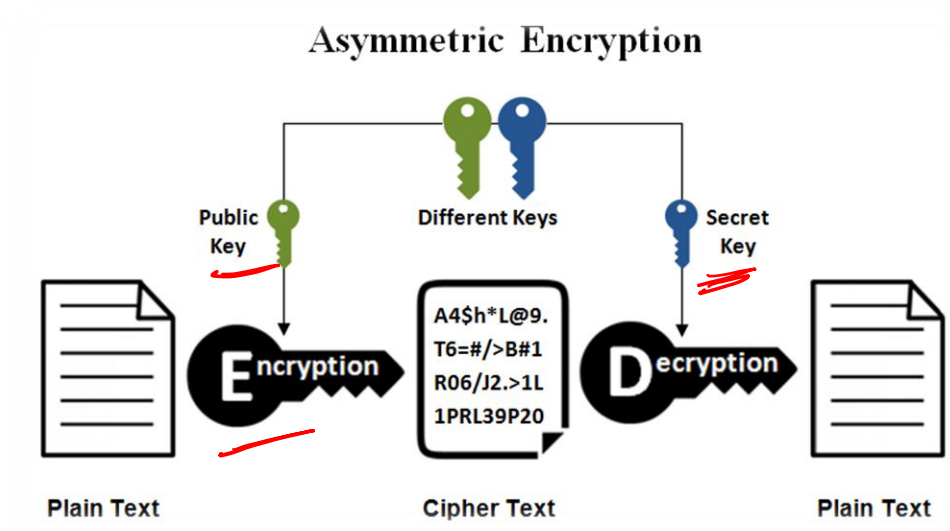
Asymmetric Encryption Algorithm

RSA Algorithm

Asymmetric Encryption Algorithm



- The **RSA** algorithm was the first practical implementation of public key encryption.
- It is named after the three researchers Ron **R**ivest, Adi **S**hamir and Len **A**dleman



RSA Algorithm

How the public and private keys are selected?

- Let n be the product of two large primes p and q (i.e. $n = p \times q$)
 - at least 512 bits.
- Select a special number e $\in \{1, (p-1)(q-1)\}$
 - greater than 1 and less than $(p-1)(q-1)$. The precise mathematical property that e must have is that there must be no numbers that divide neatly into e and into $(p-1)(q-1)$, except for 1.
- Publish the pair of numbers (n, e)
- Compute the private key d from p , q and e
- The private key d is computed to be the unique inverse of e modulo $(p-1)(q-1)$.
- In other words, d is the unique number less than $(p-1)(q-1)$ that when multiplied by e gives you 1 modulo $(p-1)(q-1)$.
 - $ed = 1 \text{ mod } (p-1)(q-1)$

Public key: (e, n)
Private key: (d, n)

RSA Algorithm

How the message is encrypted and decrypted?

Encryption

Given a message M, $0 < M < n$ $M \in \mathbb{Z}_n - \{0\}$

use public key (e, n)

compute C = M^e mod n $C \in \mathbb{Z}_n - \{0\}$

Decryption

Given a ciphertext C, use private key (d)

Compute C^d mod n = (M^e mod n)^d mod n = M^{ed} mod n = M

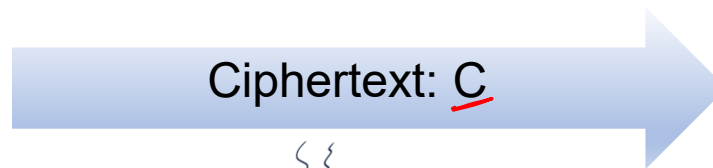
$$ed = 1 \bmod (p-1)(q-1)$$

Sender Side

Plaintext: M

Encryption

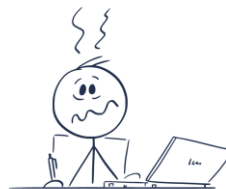
C = M^e mod (n=pq)



Receiver Side

Ciphertext: C

Plaintext: M = C^d mod n



From n, difficult to figure out p, q

From (n, e), difficult to figure d.

From (n, e) and C, difficult to figure out M s.t. C = M^e

RSA Algorithm

Can we see the public and private keys?

- Let **n** be the product of two large primes **p** and **q** (i.e. $n=pq$)
 - at least 512 bits.
- Select a special number **e**
 - greater than 1 and less than $(p-1)(q-1)$. The precise mathematical property that **e** must have is that there must be no numbers that divide neatly into **e** and into $(p-1)(q-1)$, except for 1.
- Publish the pair of numbers (**n**,**e**)
- Compute the private key **d** from **p**, **q** and **e**
- The private key **d** is computed to be the unique inverse of **e** modulo $(p-1)(q-1)$.
- In other words, **d** is the unique number less than $(p-1)(q-1)$ that when multiplied by **e** gives you 1 modulo $(p-1)(q-1)$.
 - $ed = 1 \text{ mod } (p-1)(q-1)$ ✓

Public key: (**e**, **n**)
Private key: (**d**, **n**)

A Digital Certificate is **an electronic file that is tied to a cryptographic key pair and authenticates the identity of a website, individual, organization, user, device or server**. It is also known as a public key certificate or identity certificate.

<https://www.boxentriq.com/code-breaking/hex-to-decimal-converter>

RSA Algorithm - Working Example

Asymmetric Encryption Algorithm

RSA Algorithm Example

How are the public and private keys generated?

- Assume we have generated the RSA public-private key pair:

- Choose two "big" primes $p=11$ and $q=13$
- Compute the number $n = p \cdot q = 143$

- Number $e=? \in [1, (p-1)(q-1)] = [1, 120]$
 - Relative prime with $(p-1)(q-1)$

- $e=7$

- private exponent $d = ?$

- $ed = 1 \bmod (p-1)(q-1)$

- $d=103$

Public key: (e, n) $(7, 143)$ ✓
Private key: (d, n) $(103, 143)$

- Let n be the product of two large primes p and q (i.e. $n=pq$)

- at least 512 bits.

- Select a special number e ✓

- greater than 1 and less than $(p-1)(q-1)$.
The precise mathematical property that e must have is that there must be no numbers that divide neatly into e and into $(p-1)(q-1)$, except for 1.

- Publish the pair of numbers (n, e)

- Compute the private key d from p, q and e

- The private key d is computed to be the unique inverse of e modulo $(p-1)(q-1)$.

- In other words, d is the unique number less than $(p-1)(q-1)$ that when multiplied by e gives you 1 modulo $(p-1)(q-1)$.

- $ed = 1 \bmod (p-1)(q-1)$ ✓

Public key: (e, n)
Private key: (d, n)

RSA Algorithm Example

How the message is encrypted and decrypted?

- number n say 143
- $e=7$
- $d=103$

Public key: (e, n) $(7, 143)$
Private key: $(\underline{d}, n)$ $(\underline{103}, 143)$

- Let's **encrypt** a secret message $msg = 83$

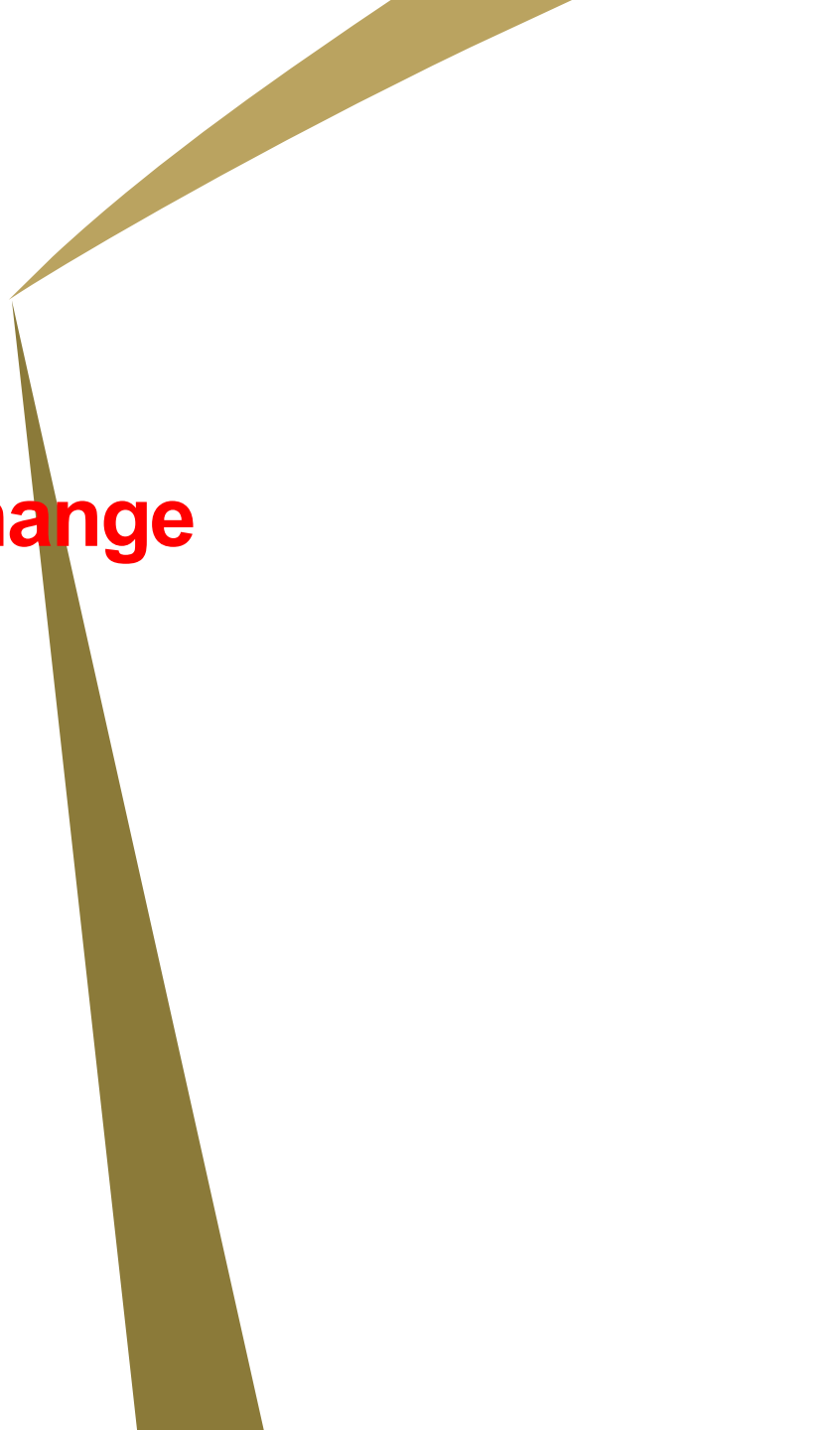
$$C = 83^7 \bmod 143 = \underline{8}$$

Encryption
 $C = \underline{M}^e \bmod (n=pq)$

- Let's now try **decryption** of the secret message $C=8$

$$M = 8^{103} \bmod 143 = \underline{\underline{83}}$$

Plaintext: $M = \underline{C}^d \bmod n$



Diffie–Hellman Key Exchange Protocol

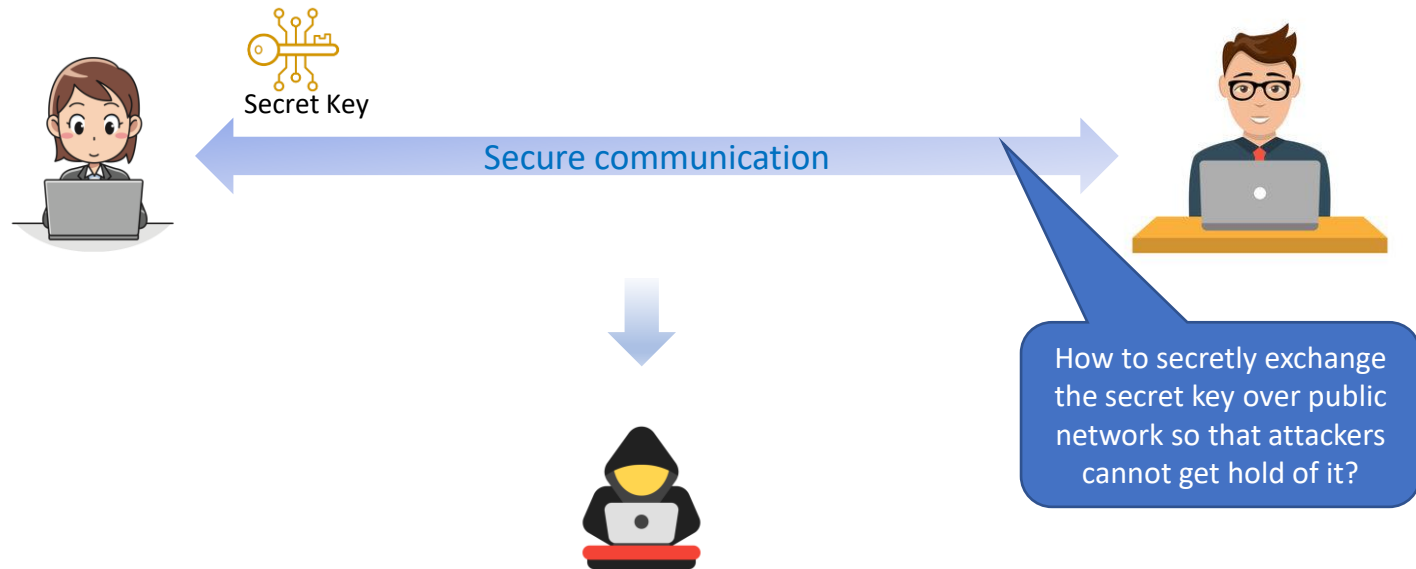
For Secure Key Exchange

Diffie–Hellman Key Exchange Protocol

For Secure Key Exchange

Two major issues with symmetric encryptions algorithms:

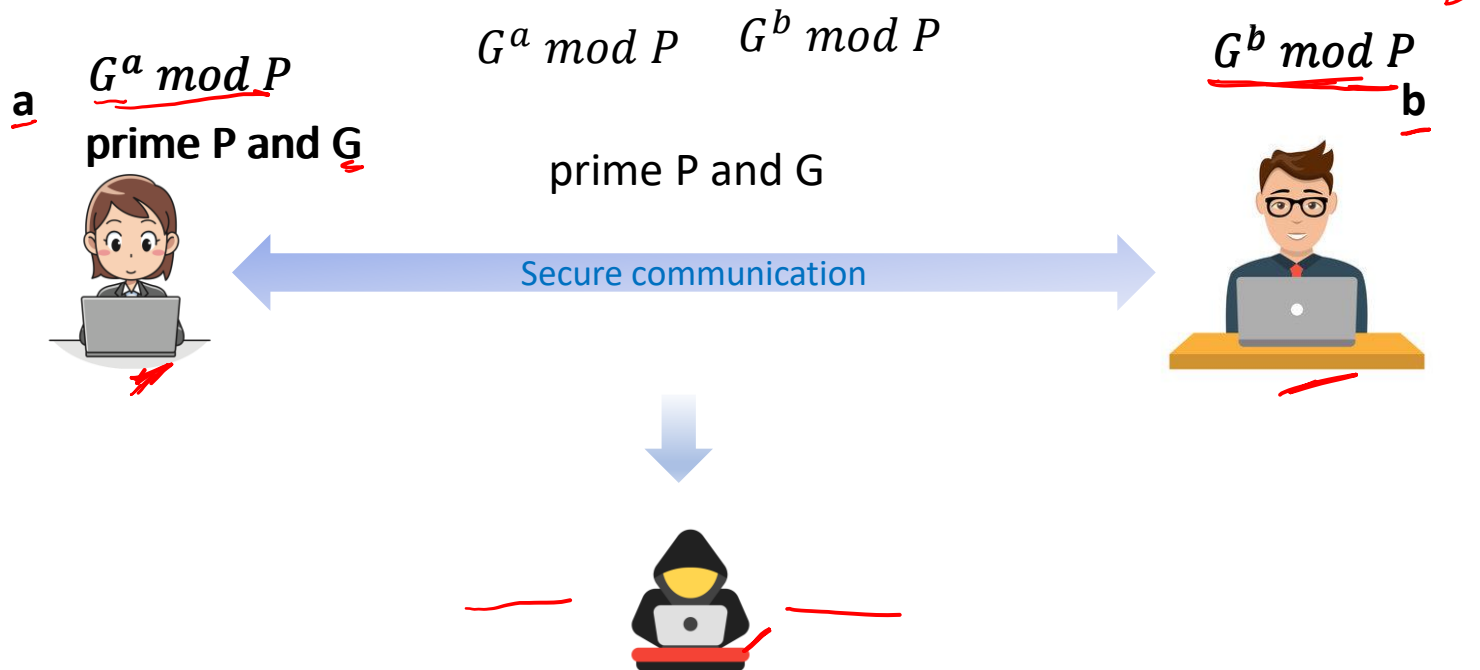
- **key distribution** – how to have secure key exchange among communicating parties
- **digital signatures** – how to verify a message comes intact from the claimed sender



Diffie–Hellman Key Exchange Protocol

How Keys are secretly exchanged?

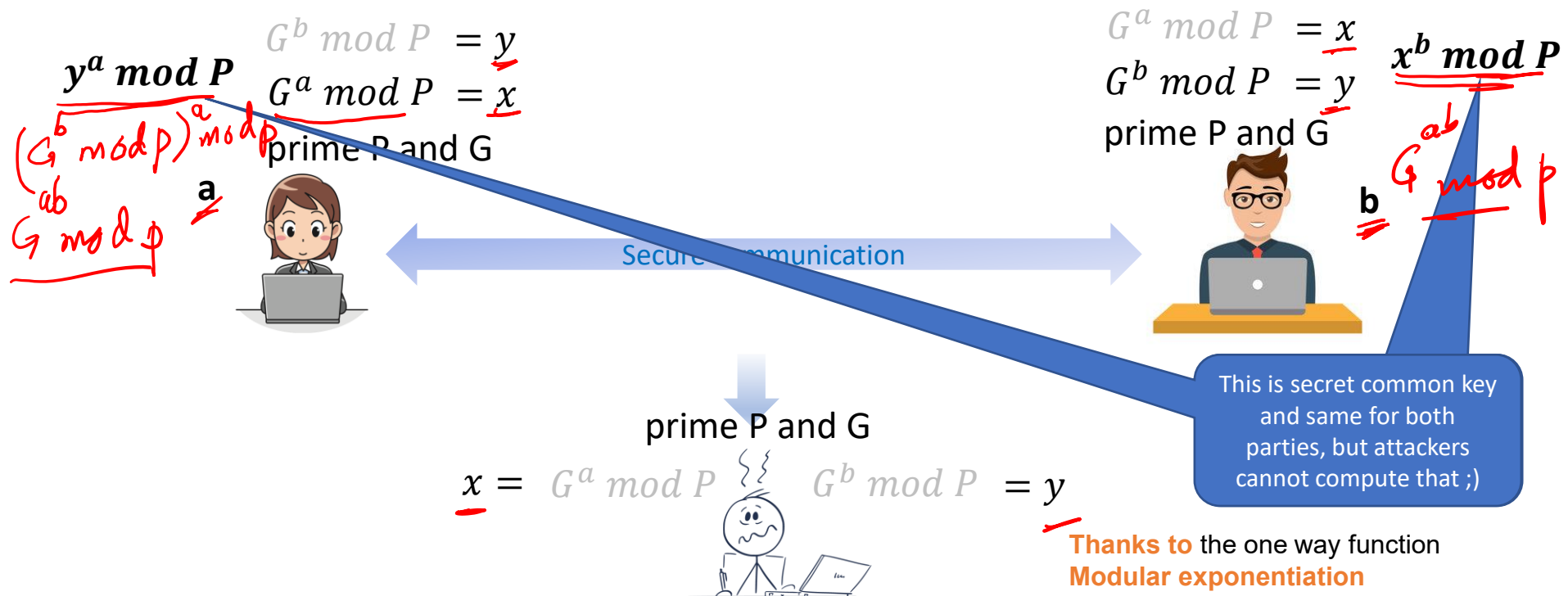
- Let's see, how two users can secretly exchange the key in presence of attacker
- Either party can choose one prime P and G (a primitive root of P) and exchange with the other...attacker can also see that
- Next, each party choose their private values
 - let's say, user-1 choose "a" and user-2 choose "b".
- Afterwards, both parties compute a number $G^x \bmod P$ and send it to the other
- Finally, both parties can generate the common secret key

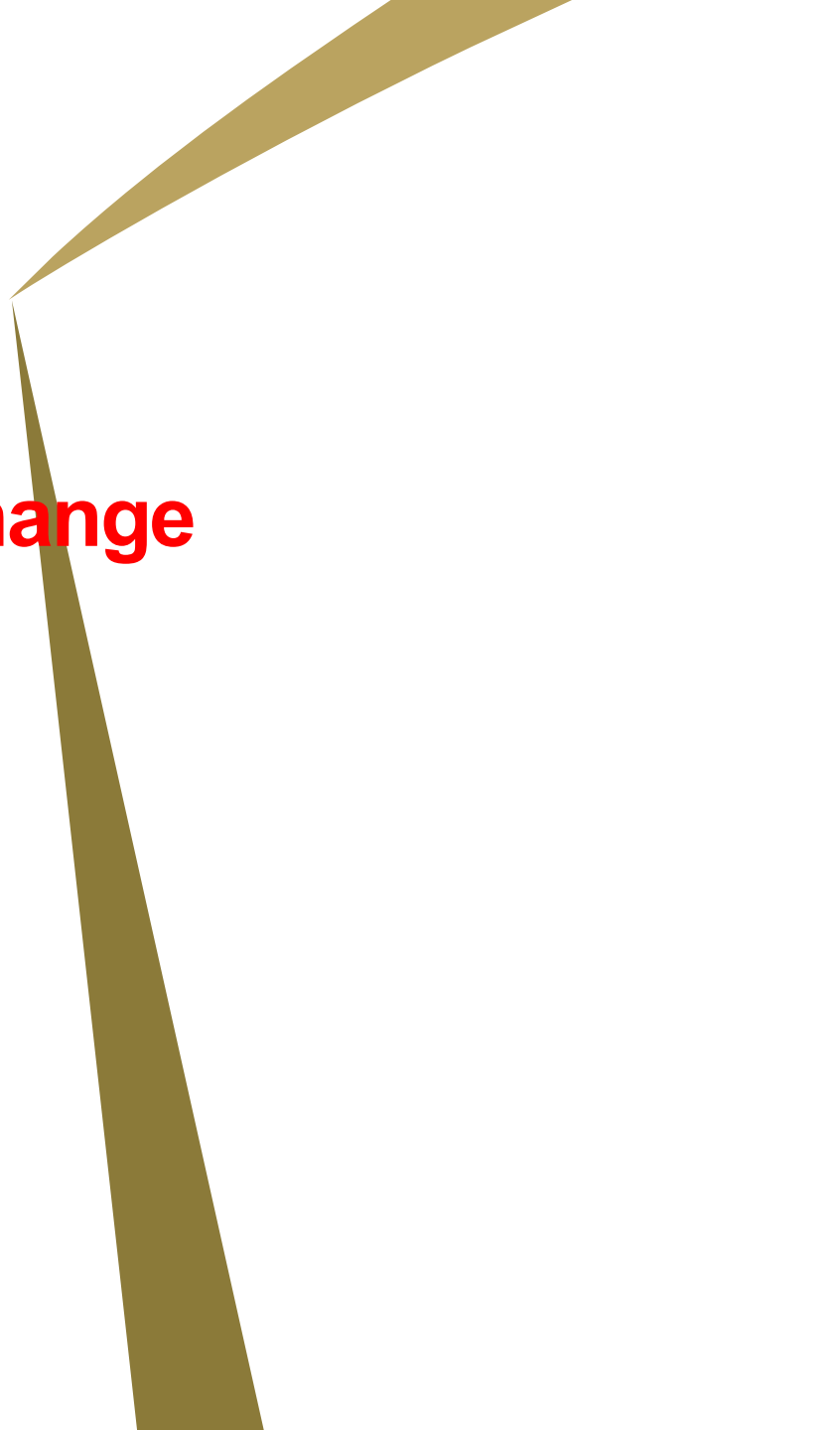


Diffie-Hellman Key Exchange Protocol

How Keys are secretly exchanged?

- Lets see, how two users can secretly exchange the key in presence of attacker
- Either party can choose one prime P and G (a primitive root of P) and exchange with the other...attacker can also see that
- Next, each party choose their private values
 - lets say, user-1 choose "**a**" and user-2 choose "**b**".
- Afterwards, both parties compute a number $G^x \bmod P$ and send it to the other
- Finally, both parties can generate the common secret key





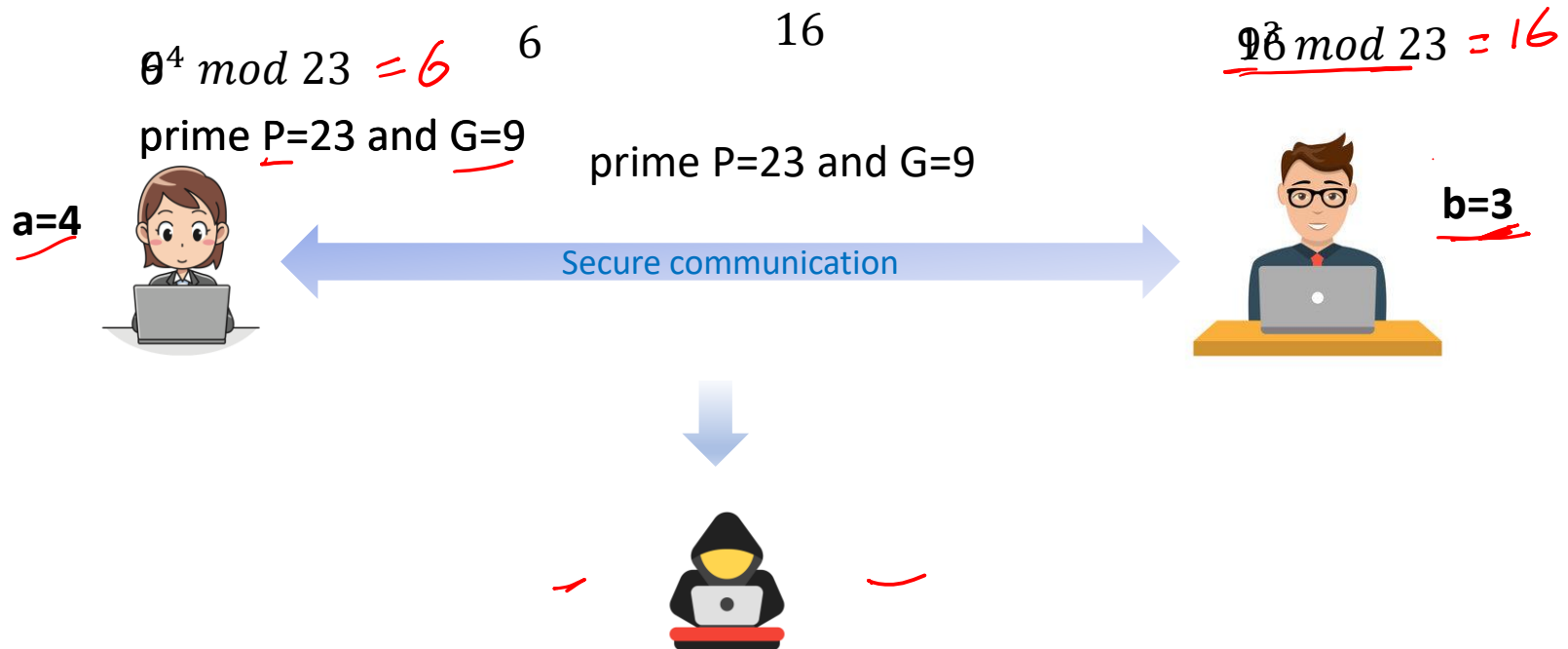
Diffie–Hellman Key Exchange Working Example

Secure Key Exchange

Diffie-Hellman Key Exchange - Example

How Keys are secretly exchanged?

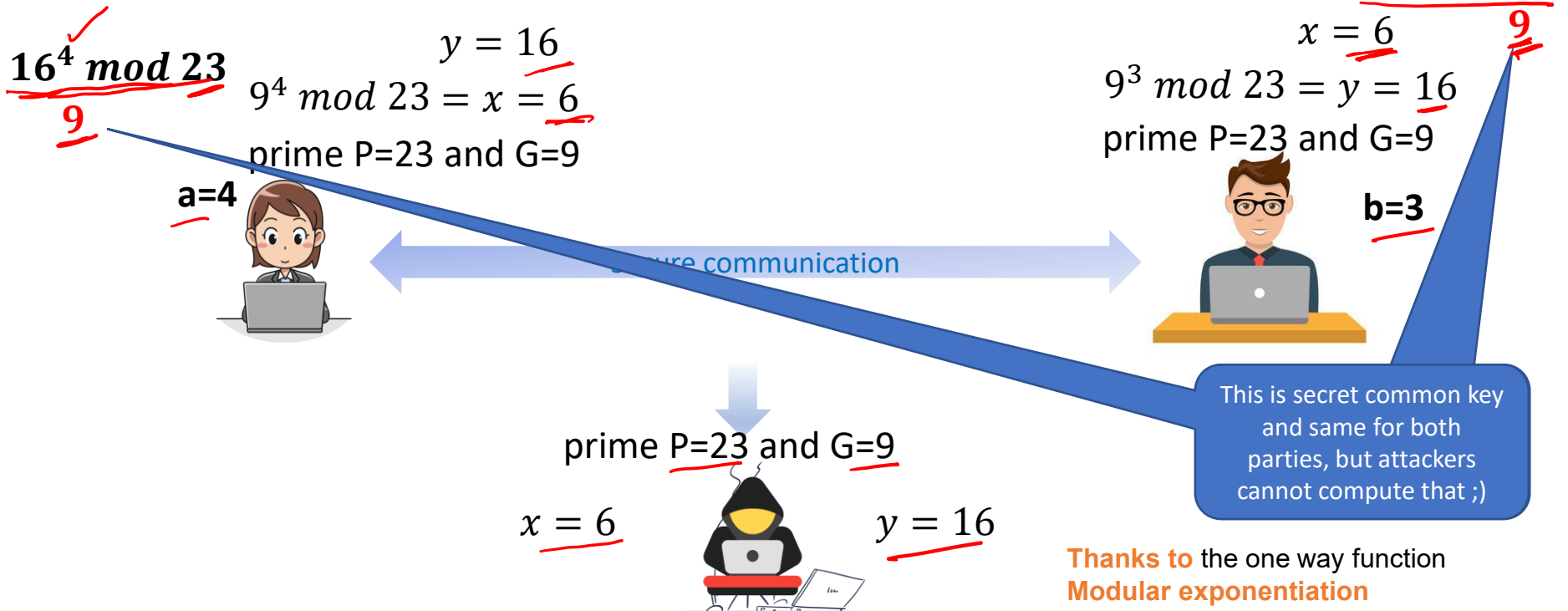
- Let's see, how two users can secretly exchange the key in presence of attacker
- Either party can choose one prime $P=23$ and $G=9$ (a primitive root of P) and exchange with the other...attacker can also see that
- Next, each party chooses their private values
 - let's say, user-1 choose " $a=4$ " and user-2 choose " $b=3$ ".
- Afterwards, both parties compute a number $G^x \bmod P$ and send it to the other
- Finally, both parties can generate the common secret key



Diffie-Hellman Key Exchange - Example

How Keys are secretly exchanged?

- Lets see, how two users can secretly exchange the key in presence of attacker
- Either party can choose one prime P and G (a primitive root of P) and exchange with the other...attacker can also see that
- Next, each party choose their private values
 - lets say, user-1 choose "**a**" and user-2 choose "**b**".
- Afterwards, both parties compute a number $G^x \bmod P$ and send it to the other
- Finally, both parties can generate the common secret key



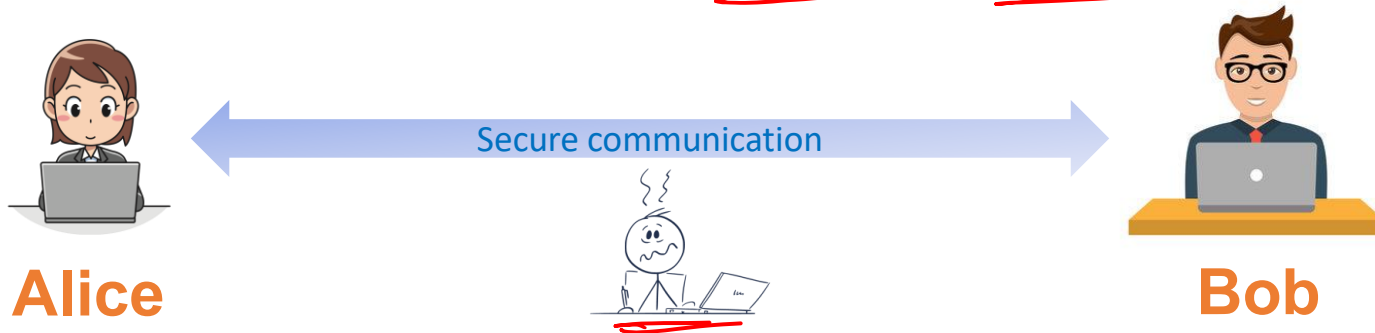
Security Issue in the Diffie-Hellman

Can we trust the Diffie-Hellman Algorithm?

Security Issue in the Diffie-Hellman

Can we trust the Diffie-Hellman Algorithm for secure Key exchange?

- Diffie-Hellman Algorithm is safe against sniffing attack...interception



- But what if attacker sits in the middle?



Man in the Middle Attack



Network Devices

Introduction

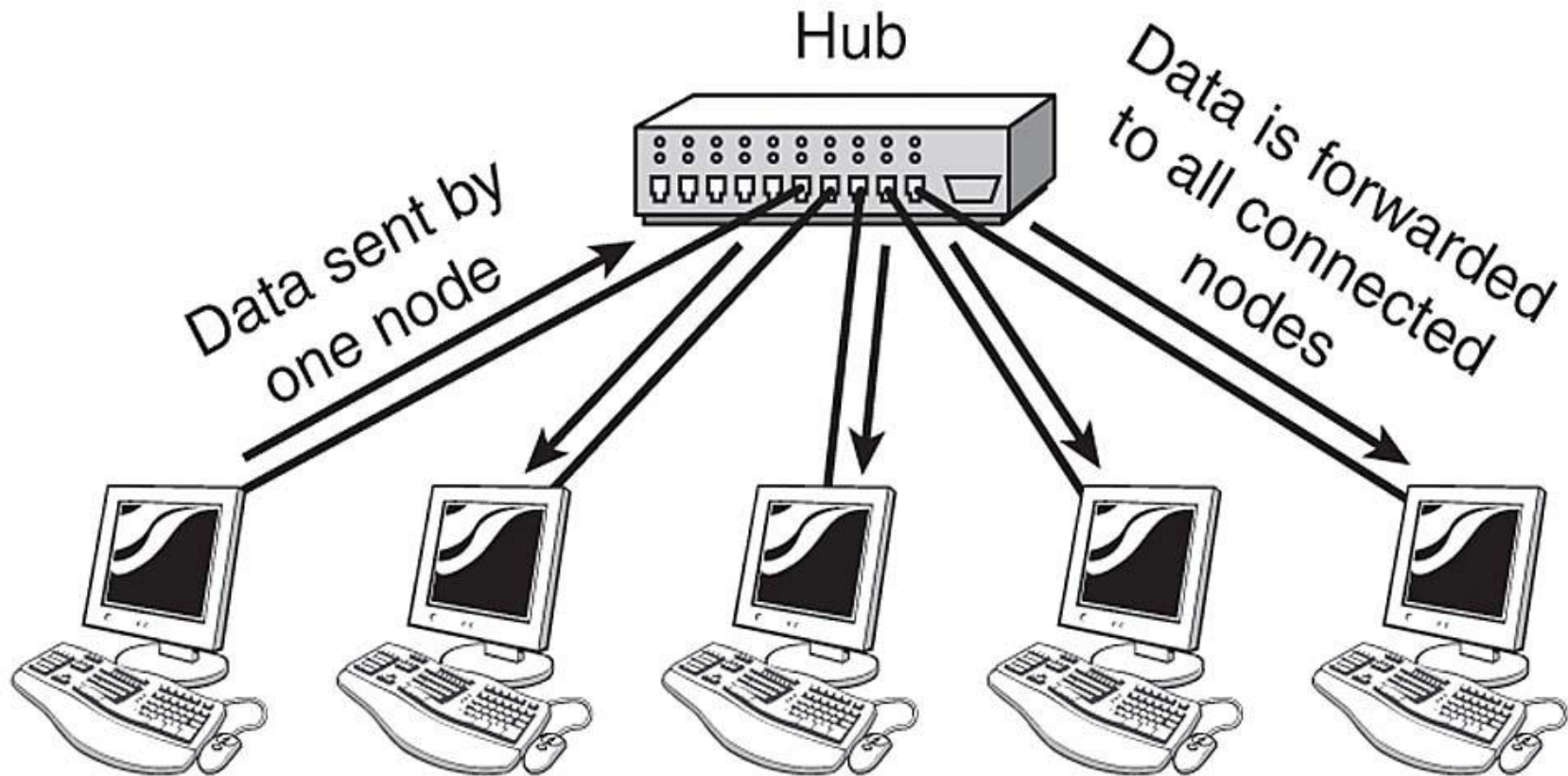
- In this topic we will cover the basics of network devices and their functions.
- We will discuss different types of network devices, including routers, switches, firewalls, and more.

Hub

- A network hub is a device that broadcasts data to every computer or Ethernet-based device connected to it.
- Network hubs are best suited for small, simple local area network (LAN) environments.



Working of Hub

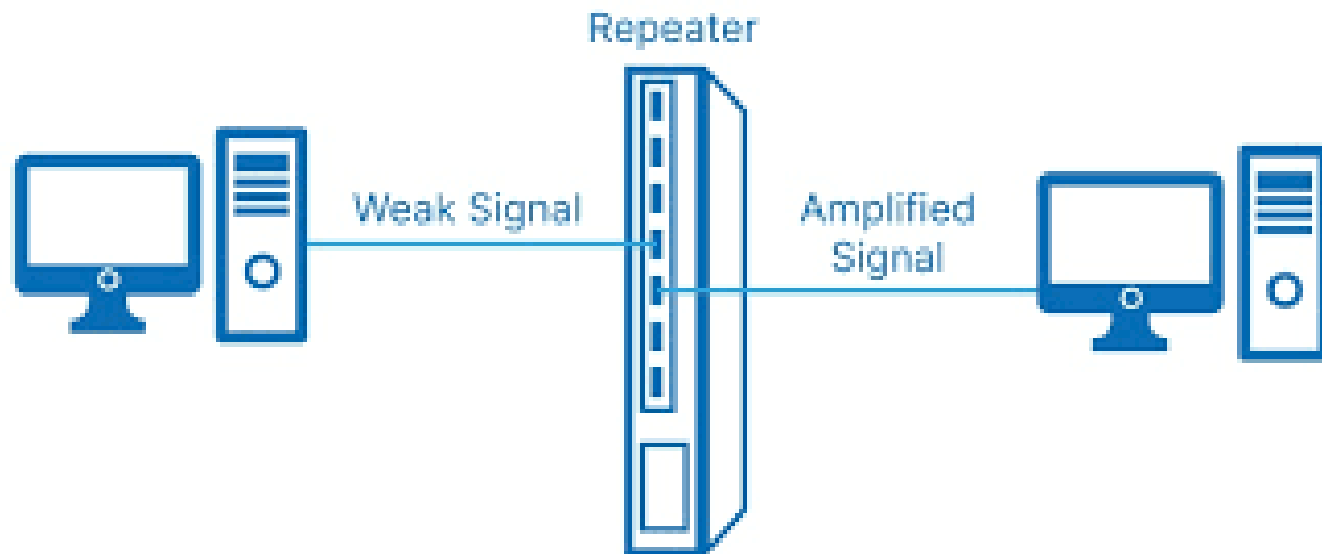


Repeater

- Repeaters are network devices operating at physical layer of the OSI model that amplify or regenerate an incoming signal before retransmitting it.
- They are incorporated in networks to expand its coverage area. They are also known as signal boosters.



Working of Repeater



Switch

- A switch is a network device that connects devices on a local area network (LAN).
- It operates at the data link layer of the OSI model and uses MAC addresses to direct traffic to the appropriate device.



Switch's working

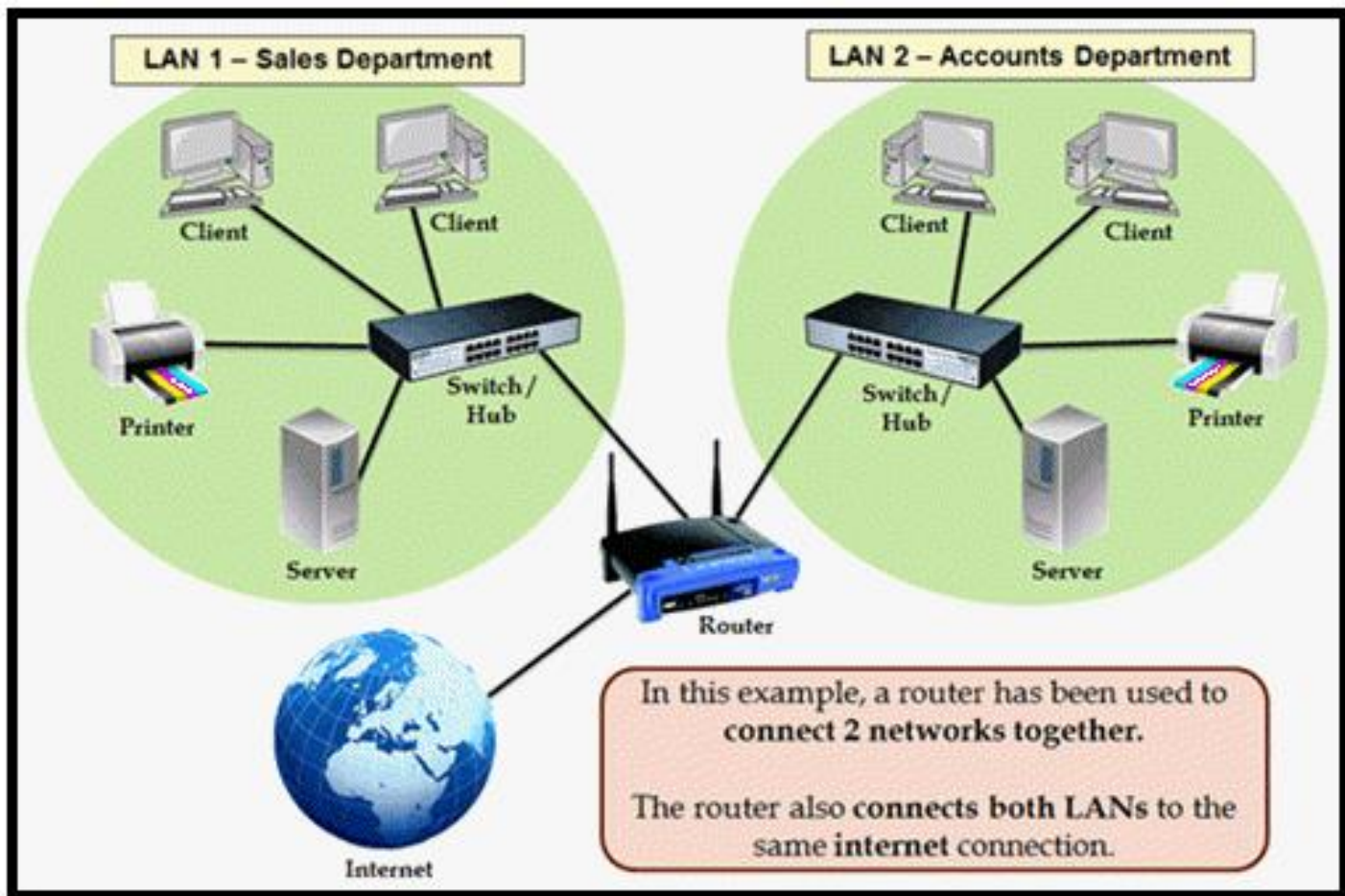


Routers

- A router is a network device that connects multiple networks and routes traffic between them.
- It operates at the network layer of the OSI model and uses routing tables to determine the best path for data to travel.

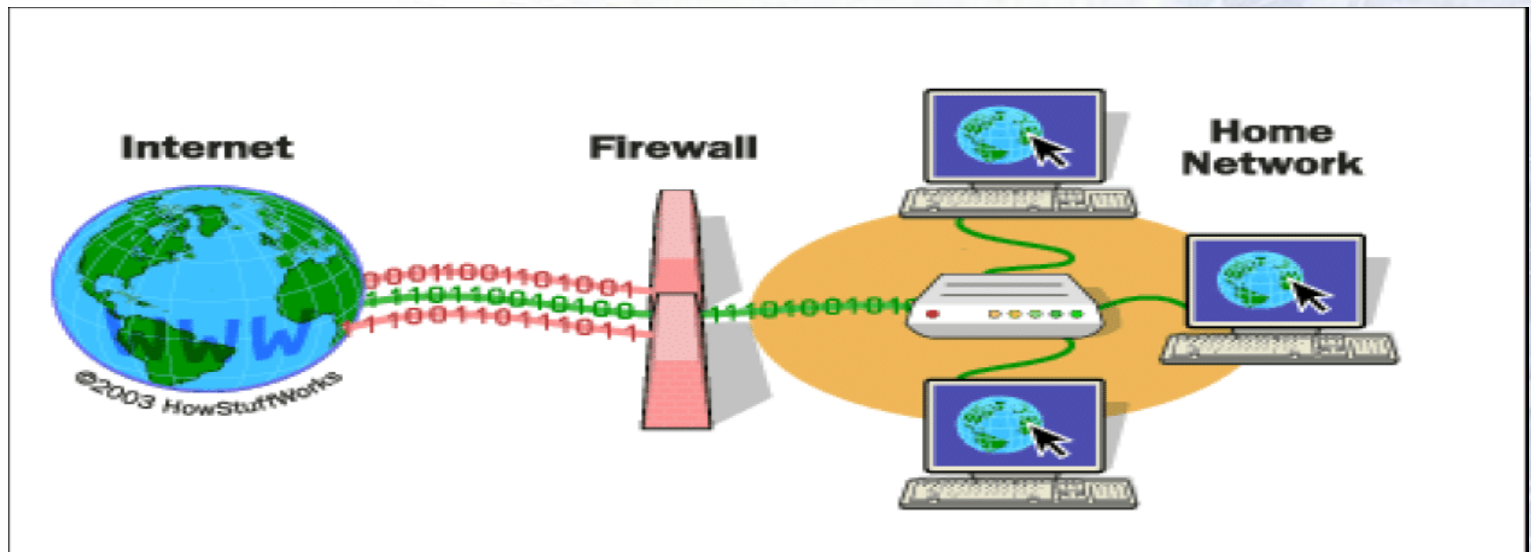


Router's working



Firewalls

- A firewall is a network security device that monitors and controls incoming and outgoing network traffic.
- It can block unauthorized access to a network and prevent malicious traffic from entering or leaving the network.

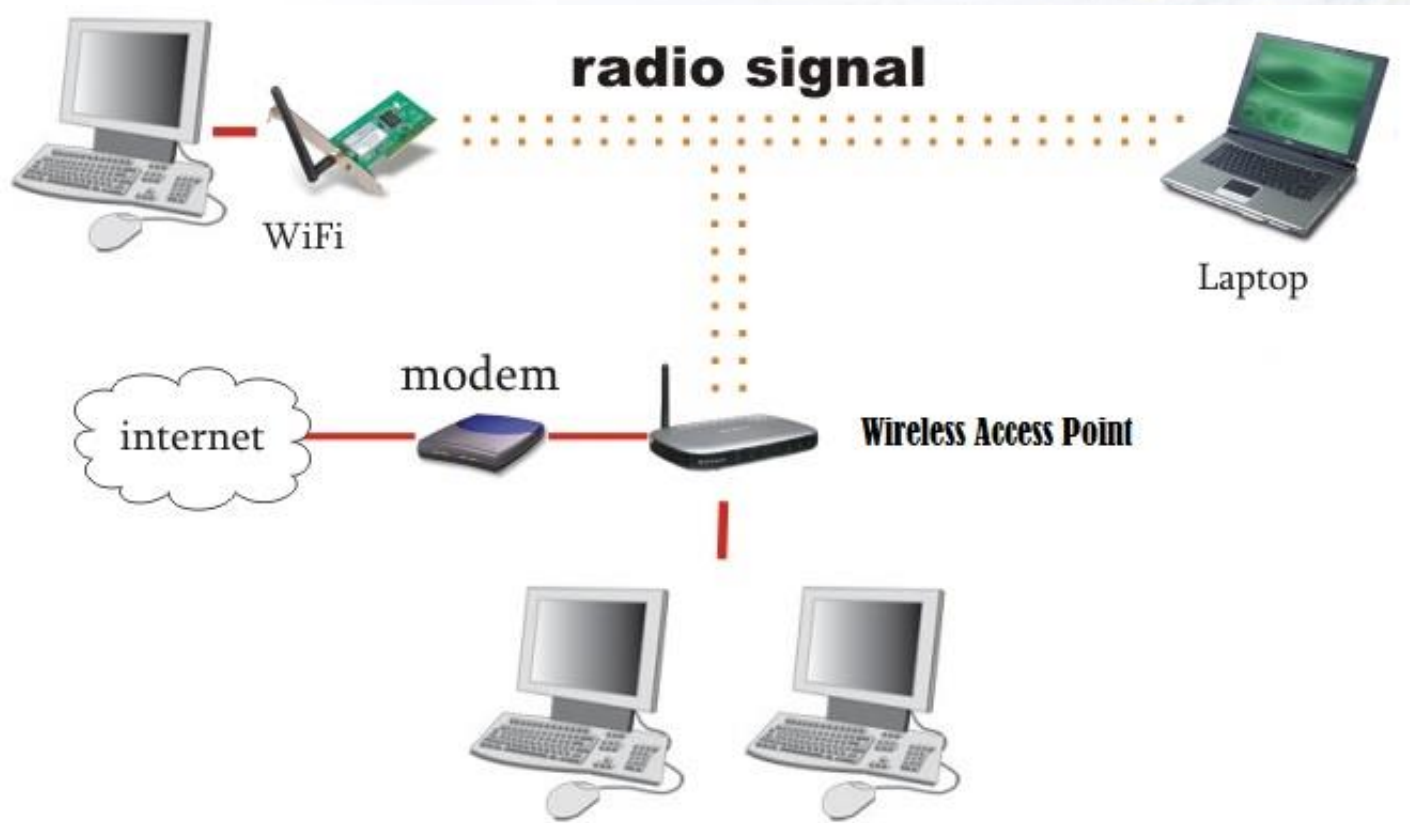


Wireless Access Points

- A wireless access point (WAP) is a network device that allows wireless devices to connect to a wired network.
- It provides a wireless signal for devices to connect to and can support multiple wireless devices simultaneously.



Wireless Access Point's working

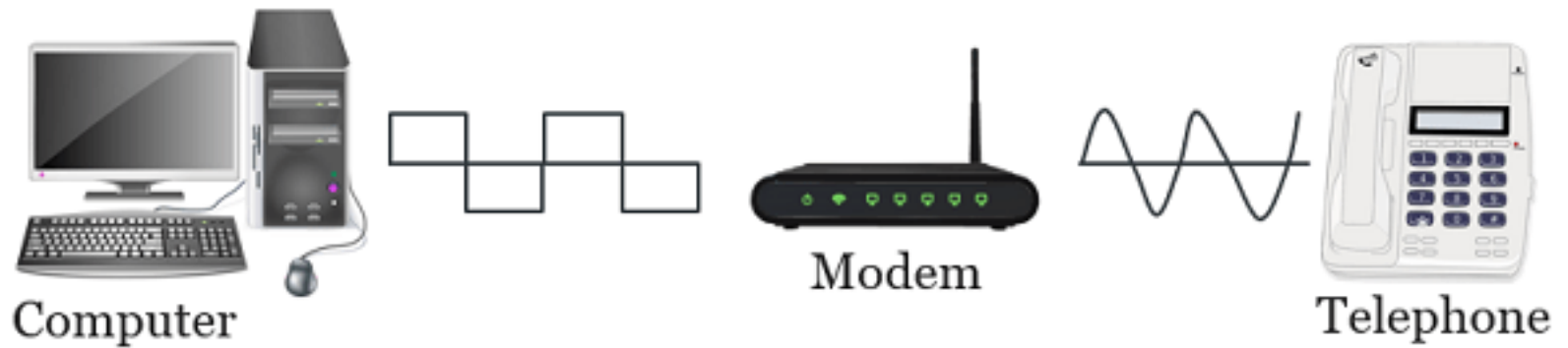


Modems

- A modem is a network device that modulates and demodulates signals between a computer and the internet service provider (ISP).
- It converts digital signals from a computer into analog signals that can travel over phone lines or cable lines, and vice versa.



Working of Modem

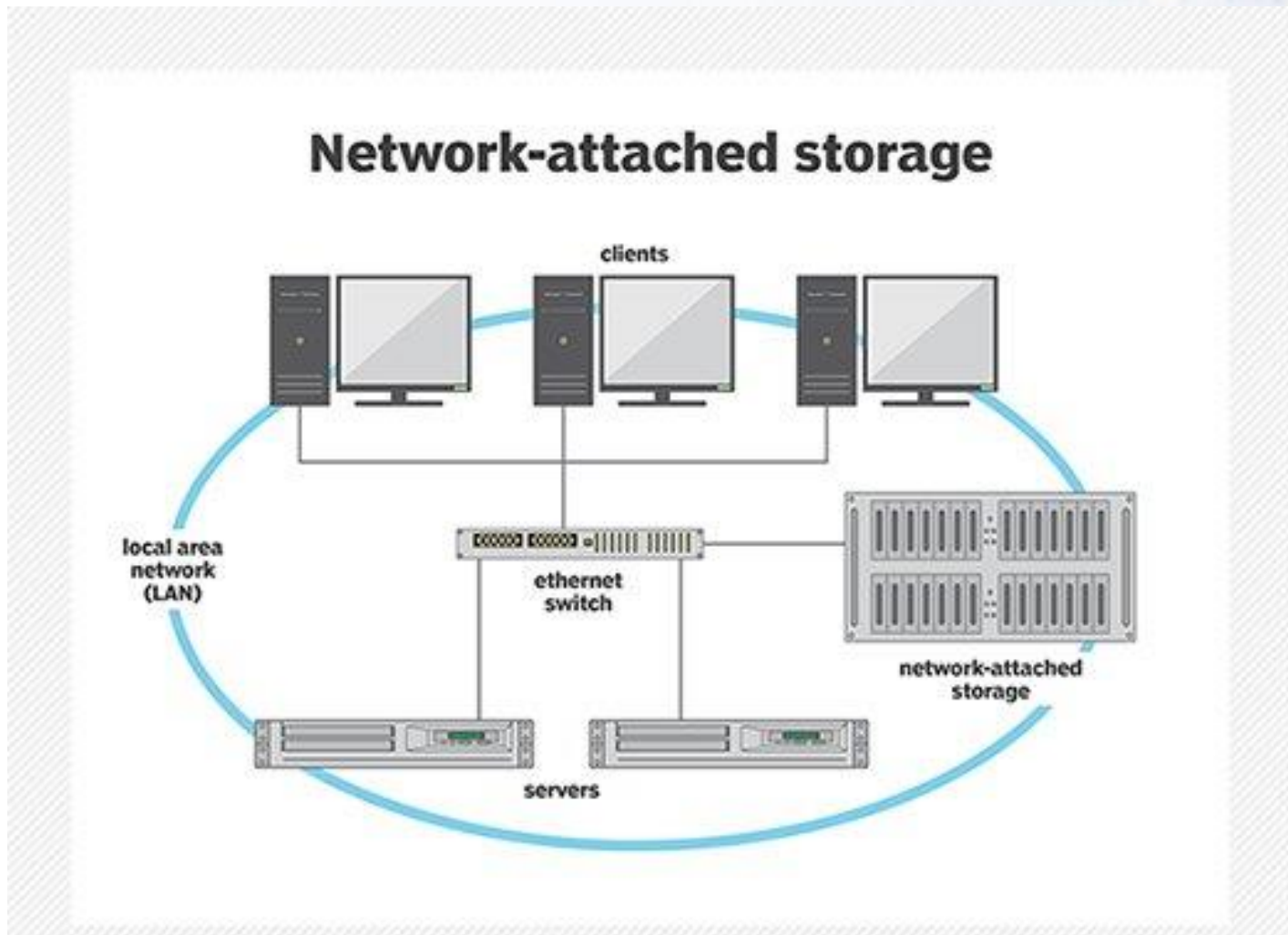


Network Attached Storage (NAS)

- Network Attached Storage (NAS) is a network device that allows multiple devices to share and access files from a central location.
- It can provide data redundancy, backup, and storage capabilities for a network.



Network Attached Storage working



Conclusion

- In conclusion, network devices are essential components of a computer network.
- Understanding the functions and capabilities of each device can help network administrators design, configure, and maintain a secure and efficient network infrastructure.



Perimeter Security

Introduction

- In this topic we will cover the basics of perimeter security and its importance in protecting a network.
- We will discuss different types of perimeter security, including physical and logical measures.

Physical Perimeter Security

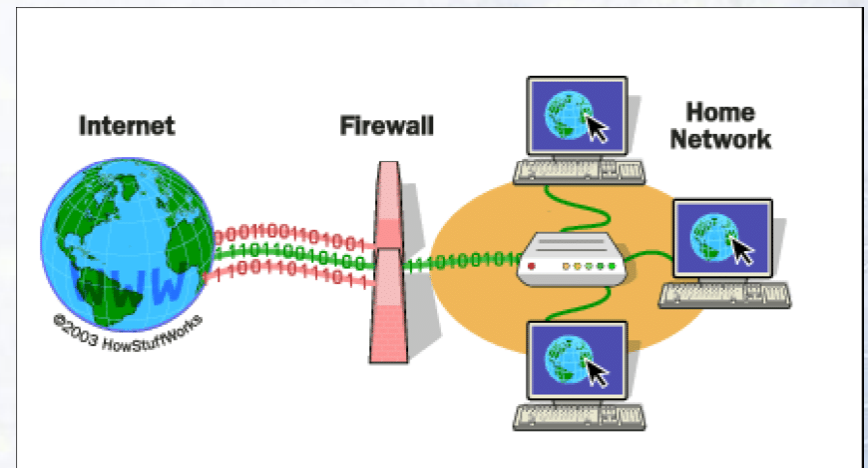
- Physical perimeter security includes measures such as fences, gates, walls, and security cameras.
- These measures restrict access to a physical location and can prevent unauthorized entry.

Logical Perimeter Security

- Logical perimeter security includes measures such as firewalls, intrusion detection systems, and virtual private networks (VPNs).
- These measures control access to a network and can prevent unauthorized access.

Firewalls

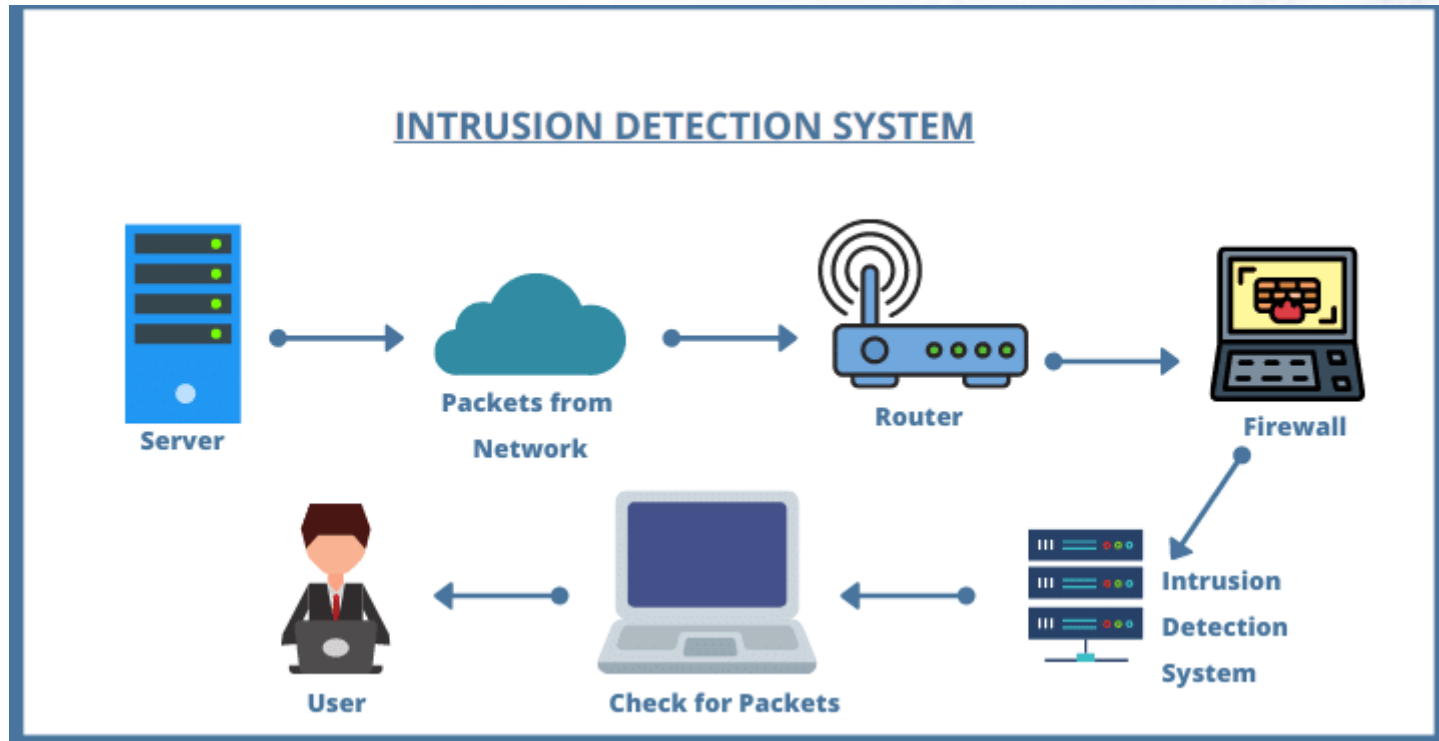
- A firewall is a logical perimeter security device that monitors and controls incoming and outgoing network traffic.
- It can block unauthorized access to a network and prevent malicious traffic from entering or leaving the network.



Intrusion Detection Systems (IDS)

- An intrusion detection system (IDS) is a logical perimeter security device that monitors a network for suspicious activity.
- It can detect and alert network administrators of potential security breaches.

Working of Intrusion Detection Systems (IDS)

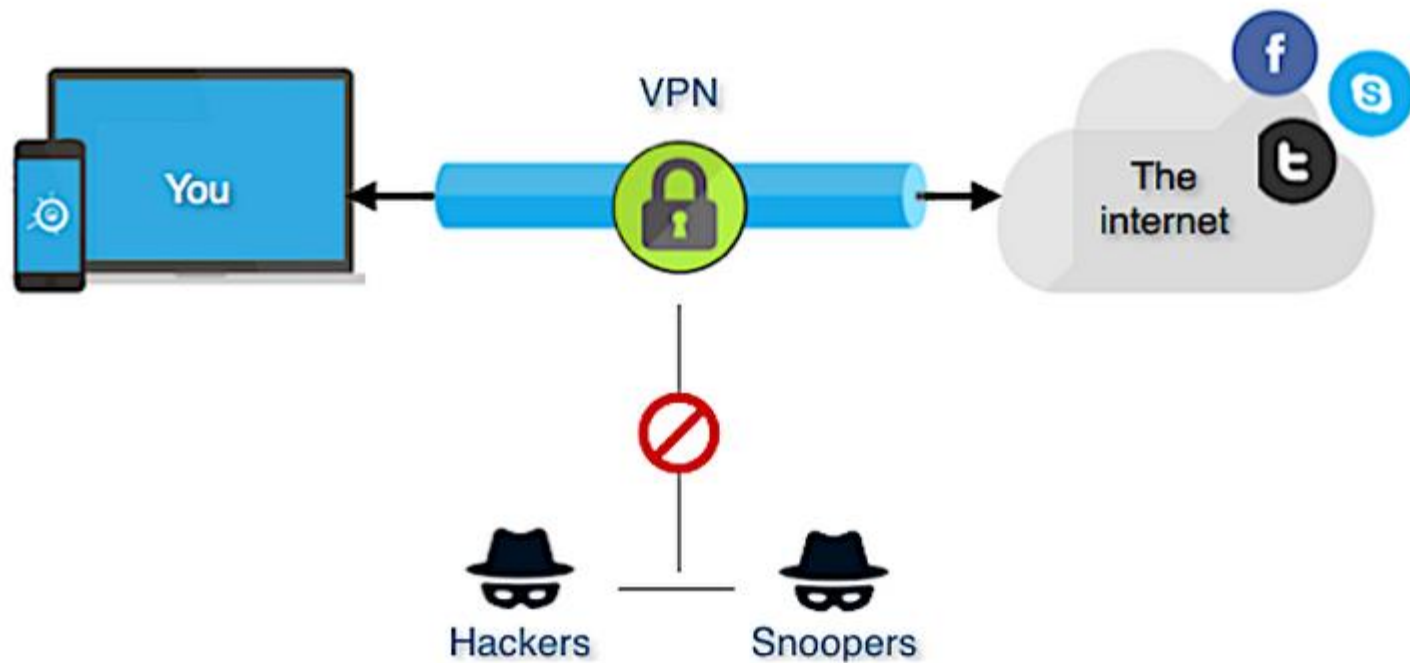


Virtual Private Networks (VPNs)

- A virtual private network (VPN) is a logical perimeter security measure that allows remote users to securely connect to a network.
- It encrypts traffic between the remote user and the network, providing secure access to network resources.



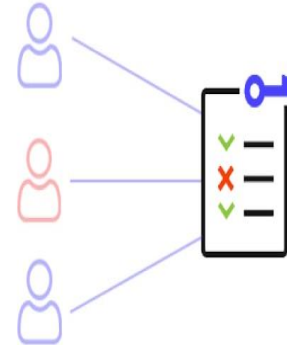
Working of VPN



Access Control

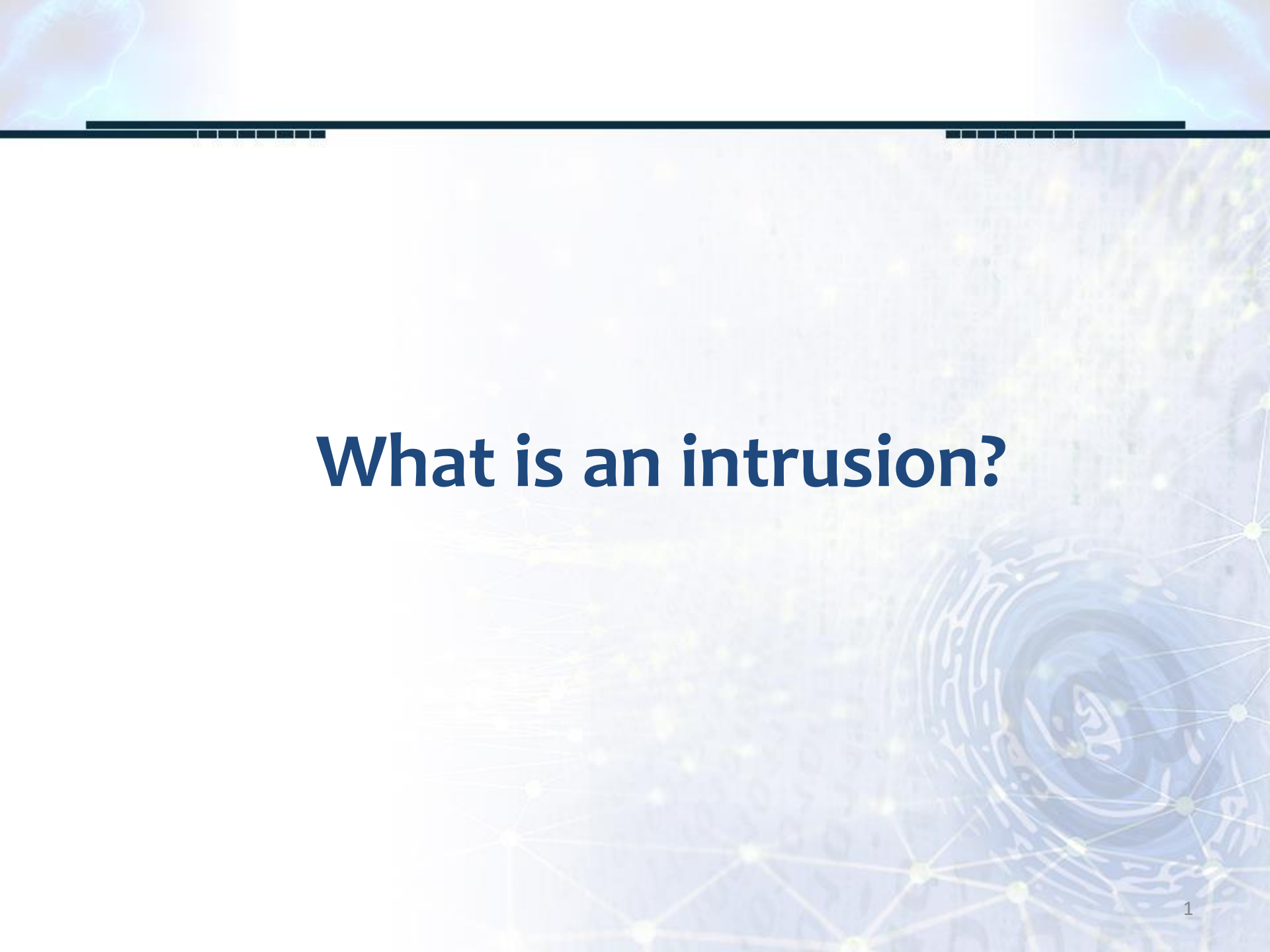
- Access control is a logical perimeter security measure that restricts access to network resources based on user credentials and permissions.
- It ensures that only authorized users have access to sensitive information.

Access Control List



Conclusion

- In conclusion, perimeter security is a critical component of network security.
- Physical and logical measures can be used to protect a network from unauthorized access and malicious attacks.
- Understanding the various types of perimeter security measures and their functions can help network administrators design, configure, and maintain a secure network infrastructure.



What is an intrusion?

Introduction

- In this topic we will cover the basics of intrusion and its impact on a network.
- We will discuss what an intrusion is, types of intrusions, and how to prevent them.

What is an intrusion?

- An intrusion is an unauthorized attempt to access, use, or modify a network, system, or device.
- It can occur through various means, including malware, phishing attacks, and brute-force attacks.

Types of Intrusions

There are different types of intrusions, including:

- **External Intrusion:** when an attacker attempts to gain access to a network from outside.
- **Internal Intrusion:** when an attacker gains access to a network from within, such as an employee with malicious intent.
- **Insider Threat:** when an employee or contractor intentionally or unintentionally causes harm to a network.
- **Distributed Denial of Service (DDoS):** when an attacker floods a network with traffic to disrupt normal operations.

Impact of Intrusions

Intrusions can have severe impacts on a network, including:

- [?]Data theft or loss
- [?]Financial loss
- [?]Reputation damage
- [?]Disruption of operations

Prevention Measures

To prevent intrusions, organizations can implement various measures, including:

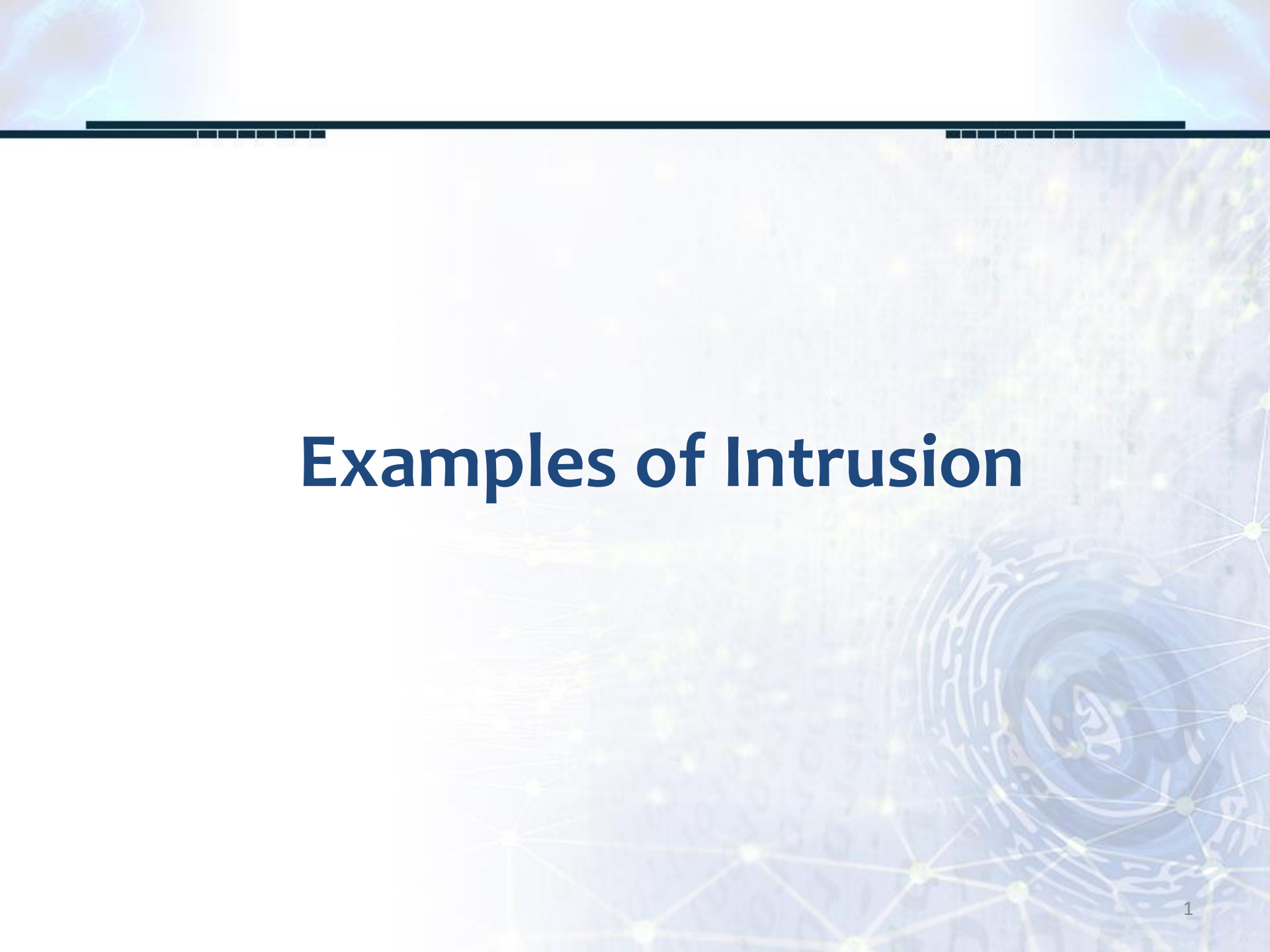
- Firewalls and intrusion detection/prevention systems
- Regular software updates and patching
- Strong password policies
- Employee training on phishing and social engineering attacks
- Access control policies

Incident Response

- Despite preventative measures, intrusions can still occur.
- Organizations should have an incident response plan in place to quickly and effectively respond to a security breach.

Conclusion

- In conclusion, intrusions are unauthorized attempts to access, use, or modify a network, system, or device.
- They can have severe impacts on an organization, but prevention measures and incident response plans can help mitigate the risks.



Examples of Intrusion

Introduction

- In this topic we will cover some common examples of intrusions.
- We will discuss the types of attacks used in each example and the impact on the affected network.

Malware Infection

- Malware is a type of software designed to harm a computer system or network.
- Examples of malware include viruses, worms, and Trojan horses.
- A malware infection can result in stolen data, financial loss, and network downtime.



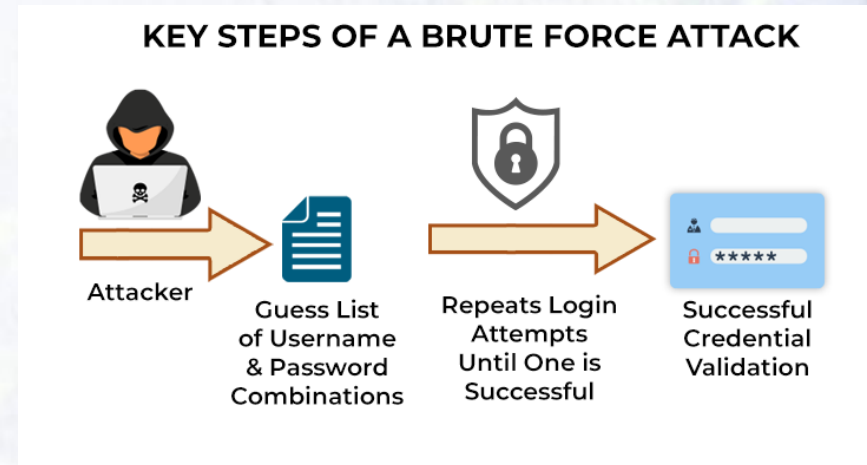
Phishing Attacks

- Phishing attacks are attempts to trick individuals into revealing sensitive information.
- Attackers often use email or social media to deceive victims into clicking a malicious link or providing personal information.
- A successful phishing attack can result in stolen login credentials or sensitive data.



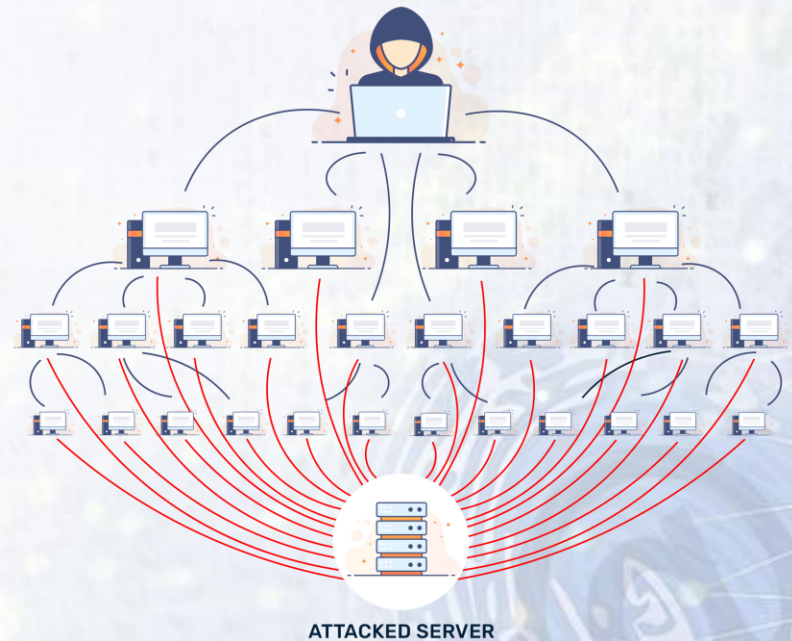
Brute-Force Attacks

- A brute-force attack is an attempt to gain unauthorized access to a system or network by guessing passwords.
- Attackers use automated tools to try multiple password combinations until they find the correct one.
- A successful brute-force attack can result in unauthorized access to sensitive data.



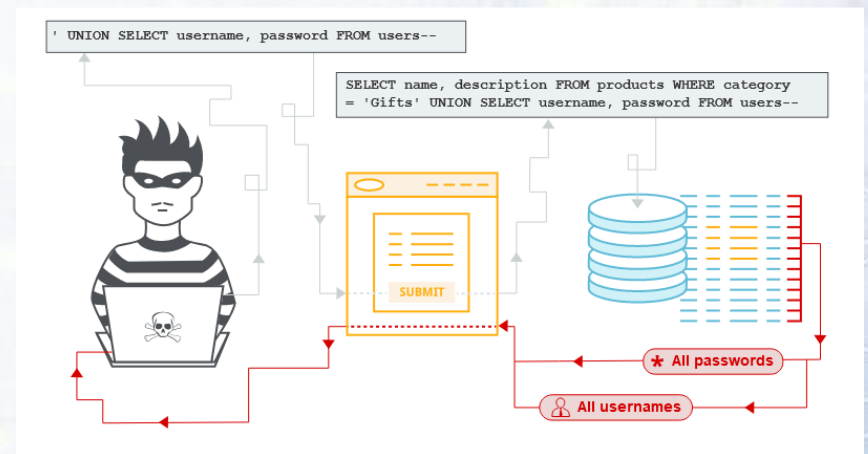
Distributed Denial of Service (DDoS)

- A DDoS attack is an attempt to disrupt normal network operations by overwhelming the network with traffic.
- Attackers use a botnet, a group of compromised devices, to flood the network with traffic.
- A successful DDoS attack can result in network downtime and financial loss.



SQL Injection

- An SQL injection is an attack that targets the database of a web application.
- Attackers input malicious SQL code into a web form to access or modify sensitive information stored in the database.
- A successful SQL injection can result in unauthorized access to sensitive data.



Conclusion

- In conclusion, there are various examples of intrusions that can impact a network.
- Attackers use different types of attacks, including malware infections, phishing attacks, brute-force attacks, DDoS attacks, and SQL injections.
- Organizations must implement strong security measures, such as firewalls, intrusion detection systems, and access control policies, to prevent and mitigate the impact of intrusions.



Intruder Behavior

Introduction

- In this topic we will cover intruder behavior and how it can impact a network.
- We will discuss the different types of intruders and their motivations for attacking a network.

Types of Intruders

There are different types of intruders, including:

Script Kiddies: inexperienced attackers who use pre-written scripts to launch attacks.

- **Hacktivists:** attackers who target a network to promote a political or social cause.
- **Insiders:** employees or contractors who intentionally or unintentionally cause harm to a network.
- **Cybercriminals:** attackers who target a network for financial gain.
- **Nation-state Actors:** attackers who are sponsored by a government to gain access to sensitive information.

Motivations for Intruder Behavior

Intruders may have various motivations for attacking a network, including:

- Financial gain
- Political or social reasons
- Espionage
- Personal challenge
- Revenge

Intruder Methods

Intruders use various methods to gain access to a network, including:

- Exploiting vulnerabilities in software or hardware
- Social engineering, such as phishing attacks
- Brute-force attacks
- Distributed denial of service (DDoS) attacks
- Insider threats

Impact of Intruder Behavior

Intruder behavior can have severe impacts on a network, including:

- Data loss or theft
- Financial loss
- Reputation damage
- Disruption of operations
- Legal consequences

Prevention and Mitigation

To prevent and mitigate the impact of intruder behavior, organizations can implement various measures, including:

- [?] Strong passwords and access control policies
- [?] Regular software updates and patching
- [?] Employee training on social engineering attacks
- [?] Firewalls and intrusion detection/prevention systems
- [?] Incident response plans

Conclusion

- In conclusion, intruder behavior can have severe impacts on a network, and there are different types of intruders with varying motivations for attacking a network.
- Organizations must implement strong security measures to prevent and mitigate the impact of intruder behavior, including strong passwords, access control policies, regular software updates, employee training, and incident response plans.



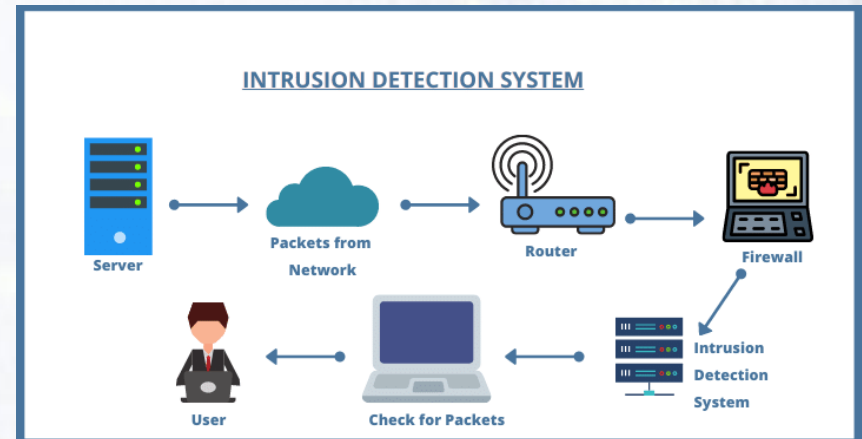
Intrusion Detection System (IDS)

Introduction

- In this topic we will cover Intrusion Detection System (IDS) and its importance in network security.
- We will discuss the types of IDS and how they work to detect intrusions.

What is an IDS?

- An IDS is a security tool designed to detect and alert on malicious activity on a network.
- IDS can detect unauthorized access, malware infections, phishing attacks, and other types of intrusions.



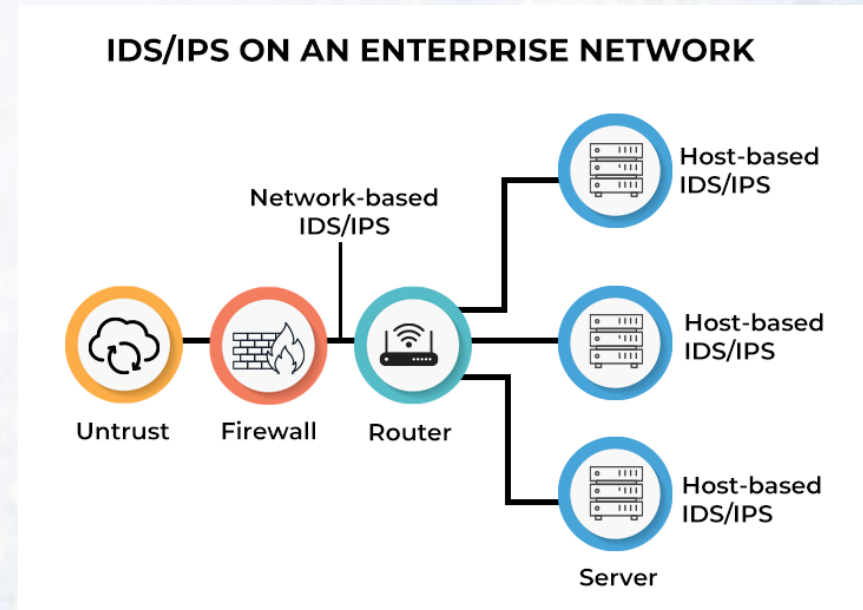
Types of IDS

There are two types of IDS:

- **Network-based IDS (NIDS):** These monitor network traffic for suspicious activity, such as unusual traffic patterns or known attack signatures.
- **Host-based IDS (HIDS):** These monitor activity on individual devices, such as servers or workstations, to detect unauthorized access or system-level attacks.

How IDS Works?

- IDS works by comparing network traffic or system activity against known attack signatures, behavior patterns, or other indicators of compromise (IoCs).
- IDS can detect intrusions by analyzing packet headers and payloads, system logs, and other data sources.
- IDS alerts security teams or generates automated responses, such as blocking traffic or quarantining affected devices.



Benefits of IDS

IDS provides several benefits, including:

- Early detection and alerting of intrusions, reducing the impact of an attack.
- Faster incident response by identifying the affected devices and the scope of the attack.
- Improved visibility into network traffic and activity, allowing security teams to identify vulnerabilities and potential attack vectors.

Limitations of IDS

IDS has some limitations, including:

- IDS may generate false positives or false negatives, leading to unnecessary alerts or missed intrusions.
- IDS requires maintenance and updates to stay current with the latest attack signatures and IoCs.
- IDS may not detect new or unknown attack vectors.

Conclusion

- In conclusion, IDS is an important tool in network security that can detect and alert on intrusions, reducing the impact of an attack.
- Organizations must choose the right type of IDS, NIDS or HIDS, based on their network architecture and security needs.
- IDS has limitations, and organizations must implement complementary security measures, such as firewalls, access control, and endpoint security, to achieve a comprehensive security posture.



IDS Analysis Approaches

Introduction

- In this topic we will cover IDS analysis approaches and how they can be used to detect and respond to network intrusions.
- We will discuss the different types of IDS analysis approaches and how they work.

Types of IDS Analysis Approaches

There are three main types of IDS analysis approaches:

- [?]Signature-based analysis
- [?]Anomaly-based analysis
- [?]Hybrid analysis

Signature-based Analysis

- Signature-based analysis compares network traffic against known attack signatures or patterns.
- IDS with signature-based analysis can identify known attacks with high accuracy but may miss new or unknown attacks.
- Signature-based analysis requires regular updates to stay current with new attack signatures.

Anomaly-based Analysis

- Anomaly-based analysis detects deviations from normal network traffic or user behavior.
- IDS with anomaly-based analysis can identify new or unknown attacks but may generate more false positives than signature-based analysis.
- Anomaly-based analysis requires a baseline of normal network traffic or user behavior to be established and updated regularly.

Hybrid Analysis

- Hybrid analysis combines signature-based and anomaly-based analysis to achieve a more comprehensive detection capability.
- IDS with hybrid analysis can identify known and unknown attacks with fewer false positives but require more resources to operate effectively.

IDS Analysis Approaches in Action

IDS analysis approaches can be applied to detect various types of network intrusions, including:

- Malware infections
- Unauthorized access attempts
- Data exfiltration
- Denial of service attacks

Conclusion

- In conclusion, IDS analysis approaches are essential for detecting and responding to network intrusions effectively.
- Organizations must choose the right type of IDS analysis approach based on their network architecture and security needs.
- IDS analysis approaches require regular maintenance, updates, and baseline establishment to operate effectively.



Host-Based IDS

Introduction

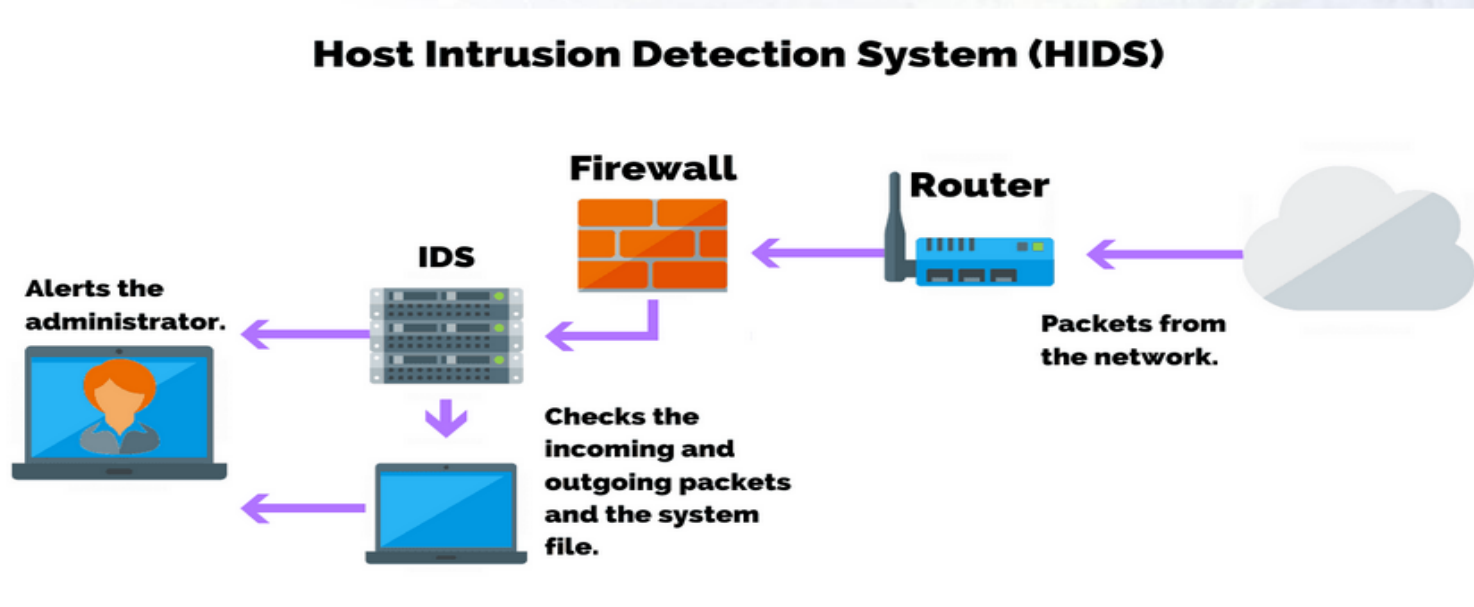
- In this topic we will cover Host-Based Intrusion Detection System (HIDS) and its importance in host-level security.
- We will discuss how HIDS works and its benefits and limitations.

What is HIDS?

- HIDS is an intrusion detection system that monitors activity on individual devices, such as servers or workstations, to detect unauthorized access or system-level attacks.
- HIDS can detect malicious activity that network-based IDS (NIDS) may not capture, such as file modifications or process-level attacks.

How HIDS Works?

- HIDS works by collecting and analyzing system activity logs, file integrity, and system performance metrics to identify suspicious activity.
- HIDS can detect unauthorized access attempts, malware infections, and other types of host-level attacks.
- HIDS alerts security teams or generates automated responses, such as terminating malicious processes or isolating affected devices.



Benefits of HIDS

HIDS provides several benefits, including:

- Granular visibility into host-level activity, allowing for faster incident response and reducing the impact of an attack.
- Detecting threats that may evade NIDS, such as fileless malware or process-level attacks.
- Enhancing compliance by monitoring and reporting on system-level activity.

Limitations of HIDS

HIDS has some limitations, including:

- HIDS can generate a high number of alerts, leading to alert fatigue and missed intrusions.
- HIDS requires maintenance and updates to stay current with the latest attack patterns and IoCs.
- HIDS may not detect sophisticated or zero-day attacks.

HIDS Deployment

HIDS can be deployed in several ways, including:

- **Agent-based:** HIDS agent installed on each monitored device.
- **Agentless:** HIDS uses system logs and network packets for analysis.
- **Hybrid:** HIDS uses a combination of agent-based and agentless approaches.

Conclusion

- In conclusion, HIDS is an essential tool in host-level security that can detect and alert on intrusions, reducing the impact of an attack.
- Organizations must choose the right type of HIDS deployment based on their network architecture and security needs.
- HIDS has limitations, and organizations must implement complementary security measures, such as firewalls, access control, and endpoint security, to achieve a comprehensive security posture.



Network-Based IDS

Introduction

- Network-based Intrusion Detection Systems (NIDS) are security tools that monitor and analyze network traffic to detect suspicious activity or malicious attacks.
- NIDS is one of the most important security measures in modern-day networks.
- NIDS provides a higher level of security by detecting and alerting the network administrator of potential threats.

Types of NIDS

- **Signature-based NIDS:** Uses a pre-defined database of known attack patterns, and alerts when a network packet matches a signature in the database.
- **Anomaly-based NIDS:** Analyzes the network traffic for abnormal behavior or traffic patterns, and alerts the administrator when such activity is detected.
- **Hybrid NIDS:** Combines both signature-based and anomaly-based detection methods to provide a more comprehensive approach to intrusion detection.