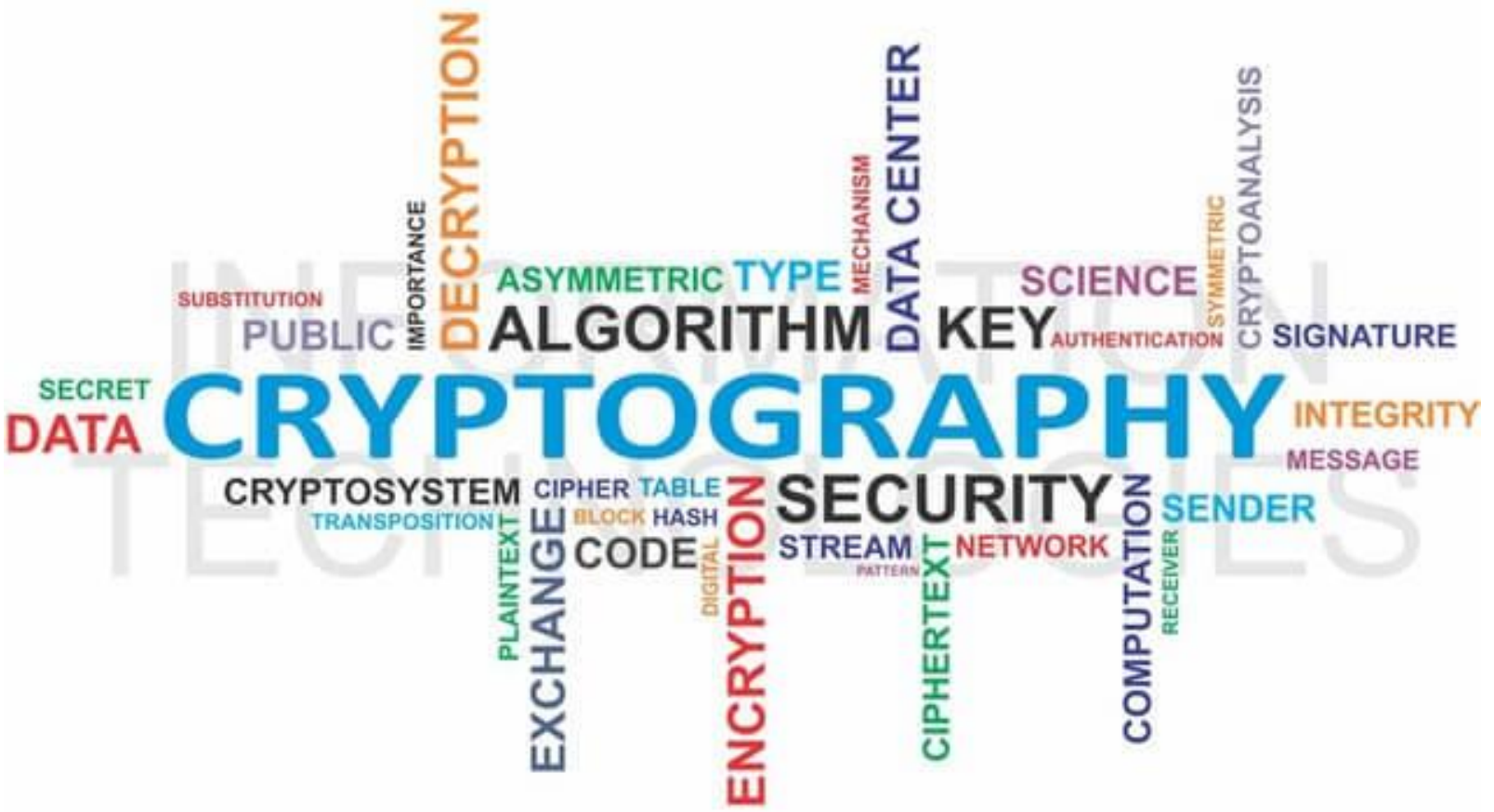
Introduction to Cryptography

Learning Objectives

At the end of this learning activity, you should be able to:

1. Articulate cryptography and its uses.
2. List different cryptography techniques and articulate the application of these techniques

Cryptography

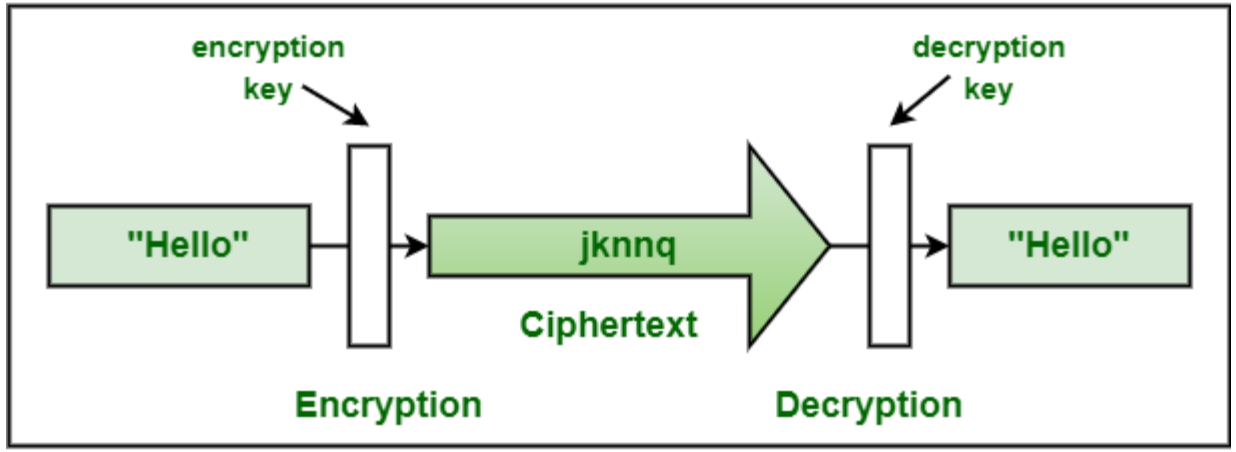


Cryptography

What is a Cryptography?

- It is the study and practice of techniques for secure communication in the presence of third parties called *adversaries*;
- It deals with developing and analyzing protocols that prevents malicious *third parties* from retrieving information being shared between two entities thereby following the various aspects of information security;
- *Crypto*' means secret or hidden;
- It is the science of secret writing with the intention of keeping the data secret.

Cryptography



Cryptography

Cryptography

Basic Terminologies:

- ***Secure Communication*** - refers to the scenario where the message or data shared between two parties can't be accessed by an *adversary*.
- ***Adversary*** - a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security, data confidentiality, data integrity;
- ***Confidentiality*** - refers to certain rules and guidelines usually executed under confidentiality agreements which ensure that the information is restricted *to certain people or places*.

Cryptography

Basic Terminologies:

- ***Data integrity* -** refers to maintaining and making sure that the data stays accurate and consistent over its entire life cycle;
- ***Authentication*** - is the process of making sure that the piece of data being claimed by the user belongs to it;
- ***Non-repudiation* -** refers to the ability to make sure that a person or a party associated with a contract or a communication cannot deny the authenticity of their signature over their document or the sending of a message.

Cryptography

Example:

```
Alice (Sender)          Bob (Receiver)
C = E (m, k)  ----->     m = D (C, k)
```

ciphertext

message

key

Encryption and
Decryption Algorithms

Cryptography

Example:

```
Plaintext : Geeksforgeeks
Ciphertext : Jhhnvirujhhnv
```

As the name suggests, in Caesar's Cipher each character in a word is replaced by another character under some defined rules. Thus, if A is replaced by D, B by E and so on. Then, each character in the word would be shifted by a position of 3

Cryptography

Types of Cryptography:

1. Symmetric-key cryptography
2. Asymmetric-key cryptography

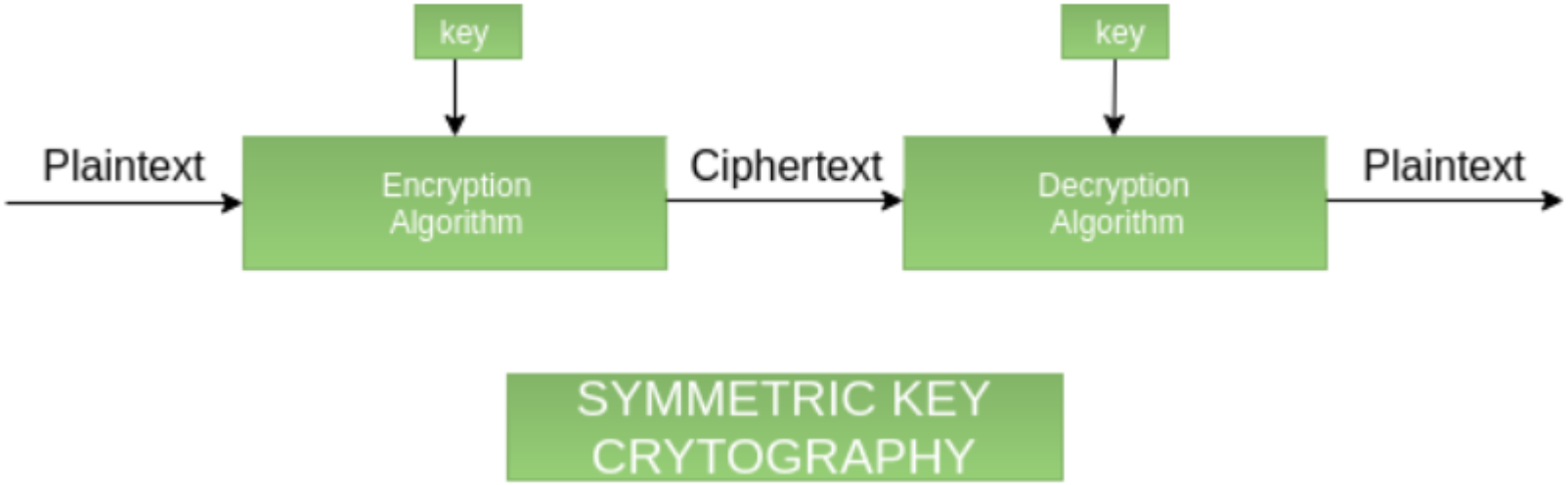
Types of Cryptography

Symmetric key cryptography:

- It involves the usage of *one secret key* along with *encryption* and *decryption* algorithms which help in securing the contents of the message.
- The strength of symmetric key cryptography depends upon the *number of key bits*.
- It is relatively *faster* than asymmetric key cryptography.
- There arises a key distribution problem as the key has to be transferred from the sender to the receiver through a secure channel.

Types of Cryptography

Symmetric key cryptography:



Types of Cryptography

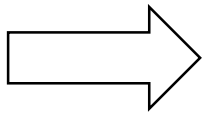
Symmetric key cryptography Example:

Scenario:

*Imagine Alice wants to send a secret message to Bob, such as "MEET AT 5 PM". They both agree to use a shared secret key, say: K = "mysecretkey123"They use a symmetric encryption algorithm, such as **AES** (Advanced Encryption Standard).*

Encryption Process (Sender - Alice):

1. Alice inputs the plaintext: **MEET AT 5 PM**
2. She uses the **AES** algorithm with the key **mysecretkey123**
3. The algorithm converts the message into an unreadable ciphertext: Encrypted Message: **a8f9b2c7d09e**... (*sample format*)



Decryption Process (Sender - Alice):

1. Bob receives the ciphertext **a8f9b2c7d09e**...
2. He uses the same key: **mysecretkey123**
3. **AES** decrypts the message and recovers the original plaintext: Decrypted Message: **MEET AT 5 PM**

Symmetric key cryptography

Examples of Symmetric key cryptography:

Algorithm	Description	Use Case
DES (Data Encryption Standard)	Outdated, replaced by AES	Historical use
AES (Advanced Encryption Standard)	Most widely used, strong and fast	VPNs, disk encryption, Wi-Fi security (WPA2)
3DES (Triple DES)	Applies DES three times	Banking systems (legacy)
Blowfish	Fast, flexible block cipher	Password management tools (e.g., bcrypt)

Symmetric key cryptography

Data Encryption Standard (DES)

- It is a symmetric block cipher.
- By ‘symmetric’, we mean that the size of input text and output text (ciphertext) is same (64-bits).
- The ‘block’ here means that it takes group of bits together as input instead of encrypting the text bit by bit.
- Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, it was replaced by Advanced Encryption Standard (AES).

Symmetric key cryptography

Characteristics of Data Encryption Standard (DES)

- It is a block cipher that encrypts data in 64 bit blocks.
- It takes a 64-bit plaintext input and generates a corresponding 64-bit ciphertext output.
- The main key length is 64-bit which is transformed into 56-bits by skipping every 8th bit in the key.
- It encrypts the text in 16 rounds where each round uses 48-bit subkey.
- This 48-bit sub key is generated from the 56-bit effective key.
- The same algorithm and key are used for both encryption and decryption with minor changes.

Symmetric key cryptography

Advanced Encryption Standard (AES)

- is a highly trusted encryption algorithm used to secure data by converting it into an unreadable format without the proper key;
- It is developed by the National Institute of Standards and Technology (NIST) in 2001;
- It is widely used today as it is much stronger than DES and triple DES despite being harder to implement;
- AES encryption uses various key lengths (128, 192, or 256 bits) to provide strong protection against unauthorized access

Symmetric key cryptography

Application of Advanced Encryption Standard (AES)

- Wireless security
- Database Encryption
- Secure communications
- Data storage
- Virtual Private Networks (VPNs)
- Secure Storage of Passwords
- File and Disk Encryption:

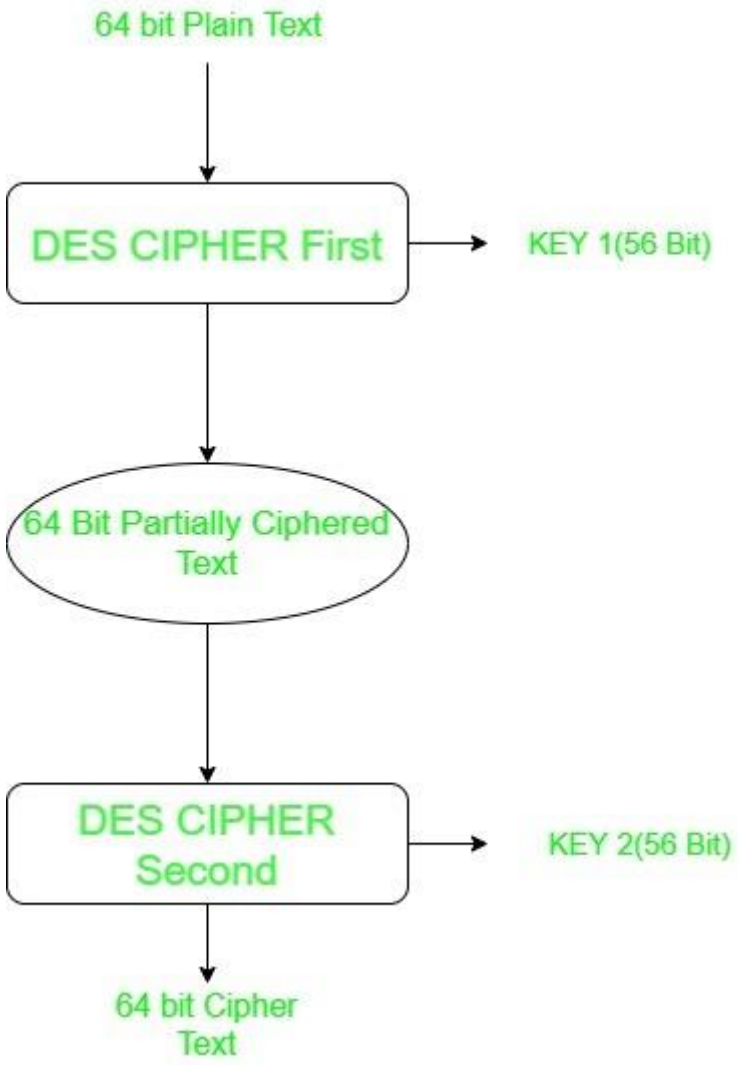
Symmetric key cryptography

Double Advanced Encryption Standard (AES)

- Double DES is a encryption technique which uses two instance of DES on same plain text.
- In both instances it uses *different keys* to encrypt the plain text.
- Both keys are *required at the time of decryption*.
- The 64 bit plain text goes into first DES instance which then converted into a 64 bit middle text using the first key and then it goes to second DES instance which gives 64 bit cipher text by using second key.

Symmetric key cryptography

Double Advanced Encryption Standard (AES)



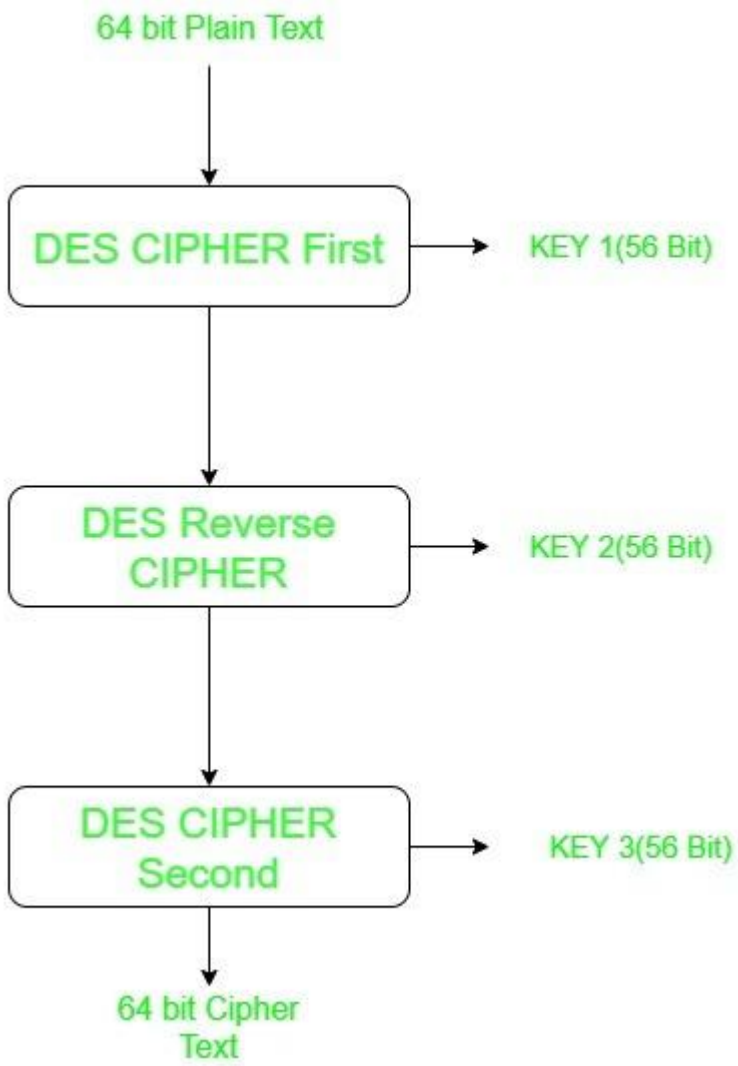
Symmetric key cryptography

Triple Advanced Encryption Standard (AES)

- Triple DES is a encryption technique which uses three instance of DES on same plain text.
- It uses there different types of key choosing technique in first all used keys are different and in second two keys are same and one is different and in third all keys are same.

Symmetric key cryptography

Triple Advanced Encryption Standard (AES)



Symmetric key cryptography

Blowfish

- It is an encryption technique designed by **Bruce Schneier** in **1993** as an alternative to DES Encryption Technique.
- It is significantly faster than DES and provides a good encryption rate with no *effective cryptanalysis technique* found to date.
- It is one of the first, secure block cyphers not subject to any patents and hence freely available for anyone to use.
- It is symmetric block cipher algorithm.

Symmetric key cryptography

Characteristics of Blowfish

1. **blockSize**: 64-bits
2. **keySize**: 32-bits to 448-bits variable size
3. **number of subkeys**: 18 [P-array]
4. **number of rounds**: 16
5. **number of substitution boxes**: 4 [each having 512 entries of 32-bits each]

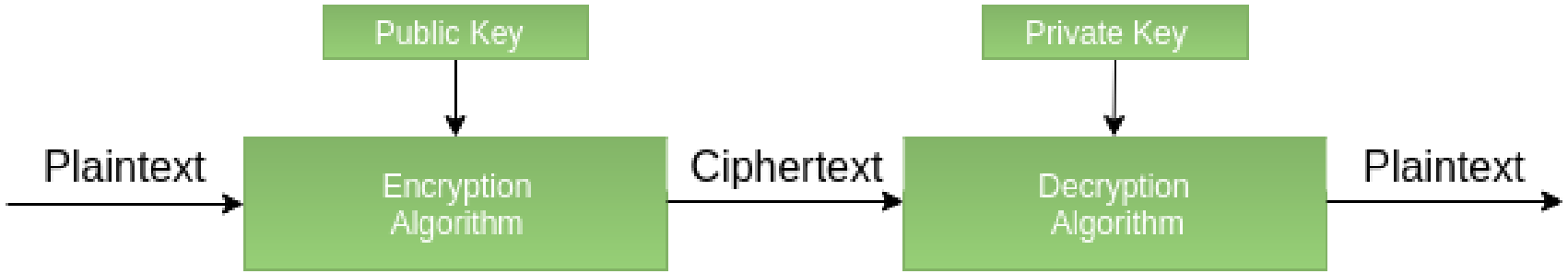
Types of Cryptography

Asymmetric key cryptography:

- It is also known as ***public-key cryptography*** because it involves the usage of a *public key* along with the secret key.
- It solves the problem of key distribution as both parties use *different keys* for encryption/decryption.
- It is not feasible to use for decrypting bulk messages as it is *very slow* compared to symmetric key cryptography.

Types of Cryptography

Asymmetric key cryptography:



ASYMMETRIC KEY CRYTOGRAPHY

Types of Cryptography

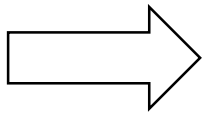
Asymmetric key cryptography Example:

Scenario:

*Imagine Alice wants to send a confidential message to Bob, such as "TRANSFER ₱5000".Instead of sharing a key, Bob gives Alice his public key. Alice uses it to encrypt the message, and only Bob can decrypt it using his private key.They use an asymmetric encryption algorithm, such as **RSA** (Rivest-Shamir-Adleman).*

Encryption Process (Sender - Alice):

1. Alice inputs the **plaintext**: **TRANSFER ₱5000**
2. She uses Bob's **public key** to encrypt the message.
3. The RSA algorithm converts the message into an unreadable ciphertext: **Encrypted Message: f82c9d03b7a2...** (this is a sample format)



Decryption Process (Sender - Alice):

1. Bob receives the **ciphertext:f82c9d03b7a2...**
2. He uses his private key to decrypt the message.
3. RSA decrypts the ciphertext and recovers the original message: **Decrypted Message: TRANSFER ₱5000**

Types of Cryptography

Examples of Asymmetric key cryptography:

Algorithm	Description	Use Case
RSA (Rivest–Shamir–Adleman)	Widely used, secure with large key sizes	HTTPS (SSL/TLS), email encryption
ECC (Elliptic Curve Cryptography)	Stronger security with smaller keys	Mobile devices, IoT, cryptocurrency wallets
ElGamal	Based on discrete logarithms	Secure key exchange
DSA (Digital Signature Algorithm)	Used for digital signatures only	Authentication and document verification

Asymmetric key cryptography

RSA (Rivest–Shamir–Adleman)

- *RSA(Rivest-Shamir-Adleman)* Algorithm is an asymmetric or public-key cryptography algorithm which means it works on two different keys: *Public Key and Private Key.*
- The Public Key is used for encryption and is known to everyone, while the Private Key is used for decryption and must be kept secret by the receiver.
- RSA Algorithm is named after **Ron Rivest**, **Adi Shamir,** and **Leonard Adleman**, who published the algorithm in **1977**.
- It is commonly utilized to ensure secure communication and for creating digital signatures. It Uses large integer prime numbers for key generation. It Encrypts data with the public key and decrypts with the private key. It is Slower than some other algorithms but offers strong security.

Asymmetric key cryptography

RSA Algorithm

RSA Algorithm is based on factorization of large number and modular arithmetic for encrypting and decrypting data. It consists of three main stages:

1. ***Key Generation***: Creating Public and Private Keys
2. ***Encryption***: Sender encrypts the data using Public Key to get cipher text.
3. ***Decryption***: Decrypting the cipher text using Private Key to get the original data.

Asymmetric key cryptography

Elliptic Curve Cryptography (ECC)

- It gives equal protection to RSA with *shorter key sizes*.
- The concept behind this is based on the mathematical properties of elliptic curves.
- It is Faster and more efficient for resource-constrained devices.
- It Gaining popularity in mobile security and the Internet of Things (IoT).

Asymmetric key cryptography

Elliptic Curve Cryptography (ECC)

Key Generation

- Select an elliptic curve over a finite field.
- Choose a base point on the curve and a large prime order.
- Select a private key, which is a random integer k.
- Now, Using the elliptic curve scalar multiplication, you need to find the public key by multiplying the base point by the private key.

Encryption and Decryption

- ECC is primarily used for key exchange, such as in the Elliptic Curve Diffie-Hellman (ECDH) algorithm, rather than directly for encryption/decryption.

Asymmetric key cryptography

ElGamal Encryption Algorithm

- **ElGamal Encryption** is a public-key cryptosystem.
- It uses asymmetric key encryption to communicate between two parties and encrypt the message.
- This cryptosystem is based on the difficulty of finding **discrete logarithms** in a cyclic group that is even if we know g^a and g^k, it is extremely difficult to compute g^{ak}.
- It was invented by **Taher ElGamal** in **1985**.
- The algorithm is widely used for secure data transmission and has digital signatures and encryption applications

Asymmetric key cryptography

Digital Signature Algorithm (DSA)

- The Digital Signature Algorithm (DSA) is a *Federal Information Processing Standard* that governs digital signatures.
- It is used to ensure the *validity* and *integrity* of a messages, software, or digital document.
- The **National Institute of Standards and Technology (NIST)** suggested DSA in **August 1991** for use in the Digital Signature Standard (DSS).
- **Digital Signature** is a Federal Information Processing Standard(FIPS) which defines algorithms that are used to generate digital signatures with the help of Secure Hash Algorithm(SHA)

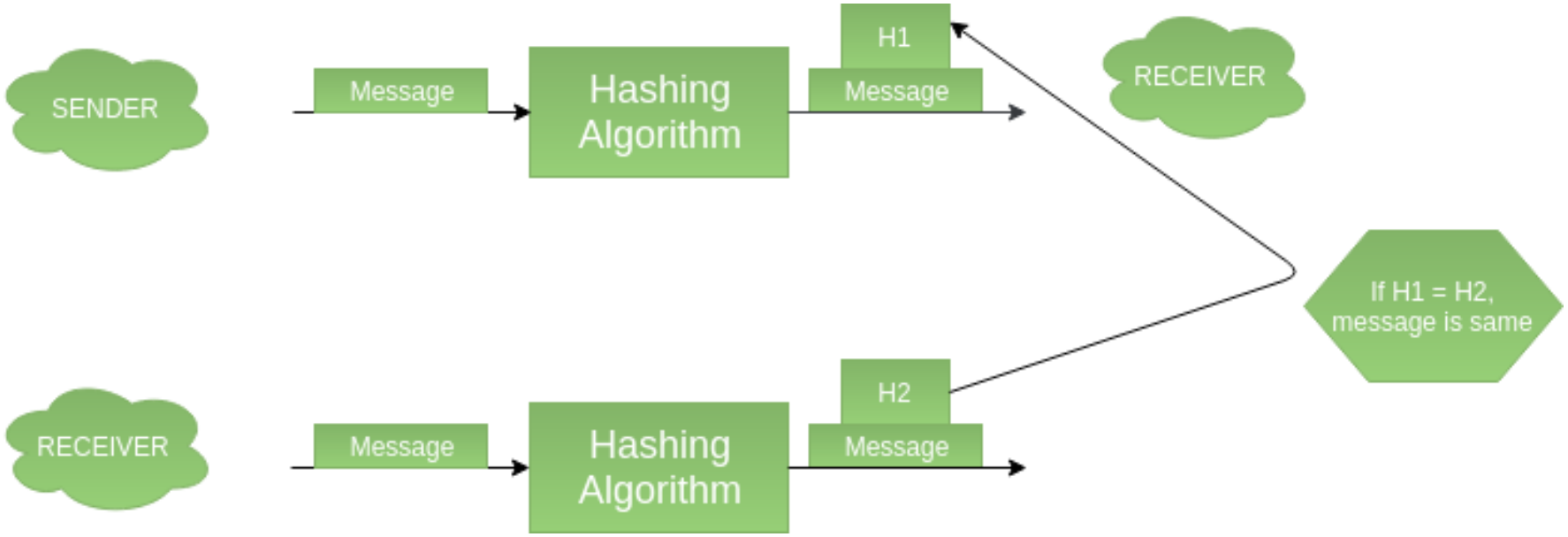
Hashing

Hashing:

- It involves taking the plain text and converting it to a hash value of fixed size by a hash function;
- This process ensures the integrity of the message as the hash value on both, the sender's and receiver's sides should match if the message is unaltered.
- **Hash functions:** A hash function is a mathematical algorithm that converts data of any size into a fixed-size output. Hash functions are often used to verify the integrity of data and ensure that it has not been tampered with.

Types of Cryptography

Hashing:



Hashing

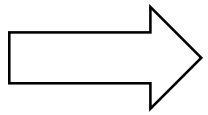
Hashing Example:

Scenario:

*A website needs to store user passwords securely. Instead of saving the password directly, it uses a hashing algorithm, such as **SHA-256**, to convert the password into a fixed, unreadable format.*

Hashing Process (User Registration):

1. Alice creates a password: **mySecureP@ssw0rd**
2. The system hashes this password using **SHA-256**.
3. The algorithm generates a unique, fixed-length hash: Hashed Password:*aee949757a2e698417463d47acac93df30cda0b1...*(This is a sample SHA-256 hash)
4. This hashed value is stored in the database — not the actual password.



Verification Process (User Login):

1. Alice logs in **with:mySecureP@ssw0rd**
2. The system hashes her input using **SHA-256** again.
3. It compares the newly generated hash to the one stored in the database.
4. ✓ If the hashes match → Access granted✗ If they don't → Access denied

Types of Cryptography

Examples of Hashing Algorithms:

Algorithm	Description	Use Case
MD5 (Message Digest 5)	Fast, outdated	Checksums, file verification (non-secure)
SHA-1 (Secure Hash Algorithm 1)	More secure than MD5, now weak	Older SSL certs, Git version control
SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)	Secure, widely used	Password hashing, SSL, blockchain
SHA-3	Latest NIST standard	Future-proof applications

Hashing Algorithms

Message Digest 5 (MD5)

- MD5 is a cryptographic hash function algorithm that takes the message as input of any length and changes it into a fixed-length message of **16 bytes**.
- MD5 algorithm stands for the Message-Digest algorithm.
- MD5 was developed in **1991** by **Ronald Rivest** as an improvement of *MD4*, with advanced security purposes.
- The output of MD5 (Digest size) is always **128 bits**.
- It is still the most commonly used message digest for non-cryptographic functions, such as used as a *checksum* to verify data integrity, compressing large files into smaller ones securely, etc.

Hashing Algorithms

Applications of Message Digest 5 (MD5)

- MD5 is used as a *checksum* to verify the integrity of files and data by comparing the hash of the original file with the file received to check if the files or data has been altered.
- MD5 is used for data security and encryption e.g. *Secure password* of users in database and non-sensitive data.
- It is used in version control systems to manage different versions of files.
- It was earlier used in digital signatures and certificate but due it's vulnerabilities, it has been replaced by more secure algorithms like SHA-256.

Hashing Algorithms

Secure Hash Algorithm 1 (SHA-1)

- It is a cryptographic algorithm that takes an input and produces a **160-bit (20-byte**) hash value.
- This hash value is known as a **message digest** which is usually then rendered as a hexadecimal number which is **40** digits long.
- It is a U.S. Federal Information Processing Standard and was designed by the United States **National Security Agency**. SHA-1 is been considered insecure since **2005**.
- However, major tech giants browsers like Microsoft, Google, Apple, and Mozilla have stopped accepting SHA-1 SSL certificates by 2017.

Hashing Algorithms

Applications of Secure Hash Algorithm 1 (SHA-1)

1. Cryptography
2. Data Integrity
3. Digital Signatures
4. Digital Forensics
5. Password Storage
6. Software Updates

Hashing Algorithms

Secure Hash Algorithm 2 (SHA-2)

- It is a cryptographic hash function designed by U.S National Security Agency to replace SH1;
- It was published in 2001;
- It produces 224, 256, 384 or 512 bits hash value;

Hashing Algorithms

Secure Hash Algorithm 3 (SHA-3)

- It is known to be the latest member of the SHA family of the secure hash algorithm stands it is published by the **NIST** on the year **2015**.
- the SHA-3 is different when we look at the internal structure of the SHA-3 because the MD-5 structure is different from the SHA-1 and SHA-2.
- The SHA-3 is designed in order to provide a "**random mapping**" from a particular string of binary data to a "**message digest**" that is fixed in size, the SHA-3 also helps to achieve more security properties.

Learning Activity # 1

1. **Explain the importance of cryptography in modern communication**. In your answer, include how cryptography supports confidentiality, integrity, and secure communication in digital environments.
2. **Differentiate between symmetric and asymmetric key cryptography**. Use practical examples (such as Alice and Bob scenarios) to illustrate how each encryption method works.
3. **Discuss the role of hashing in data security**. Provide an example of how a hashing algorithm like SHA-256 ensures the integrity of a user's password.
4. **Compare the characteristics and uses of DES, AES, and Blowfish encryption algorithms**. Highlight their key sizes, speed, and current relevance in secure communications.
5. **What is a digital signature, and how does it ensure non-repudiation?** Include in your explanation how public-key cryptography is used in this context.

Homework # 1

1. Choose one of the following cryptographic algorithms:
   - SHA-1 (Secure Hash Algorithm 1)
   - MD5 (Message Digest 5)
   - DES (Data Encryption Standard)
2. Find a sample code written in either Java or Python that demonstrates the use of your chosen algorithm. You may:
   - Search for open-source code on GitHub, tutorial sites, or documentation.
   - Write your own simple implementation (optional for bonus points).
3. Copy the code into your document or attach it as a separate file. Be sure to:
   - Add brief comments to explain each step of the code (either your own or the one you found).
   - Highlight which part is performing encryption, decryption, or hashing.
4. Answer the following questions below the code:
   a) What does this code do?
   b) What inputs are required and what outputs does it produce?
   c) Why is this algorithm considered important in cryptography?
   d) Are there any security concerns or limitations with this algorithm?
5. Cite your sources if you used any website, book, or article.