

# SPPU-BE-COMP-CONTENT – KSKA Git

## BCT UNIT 1 - PYQ Answers

➤ OCT 2022

Q1)

a) List and explain advantages of ECC. [5]

### Advantages of Elliptic Curve Cryptography (ECC)

#### 1. Stronger Security with Smaller Keys

- ECC offers equivalent security to traditional algorithms (like RSA) but with much shorter key lengths.
- Example: A 256-bit ECC key  $\approx$  security of a 3072-bit RSA key.
- This makes it more resistant to brute-force attacks.

#### 2. Faster Computation

- Due to smaller keys, encryption, decryption, and key generation operations are faster.
- Improves performance in devices with limited processing power.

#### 3. Lower Storage and Bandwidth Requirements

- Shorter keys reduce the amount of data stored and transmitted.
- Suitable for constrained environments like IoT and mobile devices.

#### 4. Reduced Power Consumption

- Requires fewer computational resources, leading to lower energy usage.
- Essential for battery-powered and embedded systems.

#### 5. Scalability and Future-Proofing

- Compact design allows easy integration into modern systems.
- Provides high security even as computational power increases over time.

ECC is preferred for modern applications due to its strong security, efficiency, and suitability for resource-constrained environments.

## SPPU-BE-COMP-CONTENT – KSKA Git

### b) Differentiate asymmetric and symmetric key cryptography. [5]

Parameter	Symmetric Key Cryptography	Asymmetric Key Cryptography
<b>Number of Keys</b>	One single shared key for both encryption and decryption.	Two different keys – a public key (encryption) and a private key (decryption).
<b>Speed</b>	Faster because algorithms are simple and require less computation.	Slower due to complex mathematical operations like large integer factorization or elliptic curves.
<b>Security</b>	Less secure if the shared key is intercepted, as it compromises the system.	More secure since the private key is kept secret and never transmitted.
<b>Key Distribution</b>	Key must be shared secretly before communication, making distribution difficult.	Public key can be openly shared, making key distribution easier.
<b>Resource Usage</b>	Low computational and memory requirements.	High computational and memory usage.
<b>Examples</b>	AES, DES, 3DES, Blowfish.	RSA, ECC, DSA.
<b>Best Use Case</b>	Suitable for encrypting large volumes of data quickly.	Suitable for key exchange, authentication, and digital signatures.
<b>Confidentiality &amp; Authentication</b>	Provides confidentiality but not authentication by default.	Can provide both confidentiality and authentication.

ALTERNATIVE :

## SPPU-BE-COMP-CONTENT – KSKA Git

Symmetric Key Cryptography	Asymmetric Key Cryptography
There is only one key (symmetric key) is used, and the similar key can be used to encrypt and decrypt the message.	There are two different cryptographic keys (asymmetric keys), known as the public and the private keys, are used for encryption and decryption.
It is effective as this technique is recommended for high amounts of text.	It is inefficient as this approach is used only for short messages.
Symmetric encryption is generally used to transmit bulk information.	It is generally used in smaller transactions. It is used for making a secure connection channel before transferring the actual information.
Symmetric key cryptography is also known as secret-key cryptography or private key cryptography.	Asymmetric key cryptography is also known as public-key cryptography or a conventional cryptographic system.
Symmetric key cryptography uses fewer resources as compared to asymmetric key cryptography.	Asymmetric key cryptography uses more resources as compared to symmetric key cryptography.
The length of the keys used is frequently 128 or 256 bits, based on the security need.	The length of the keys is much higher, such as the recommended RSA key size is 2048 bits or higher.

### c) Discuss the properties of hash function [5]

A hash function is a mathematical algorithm that converts input data of any size into a fixed-size string (hash value or digest). It is widely used in cryptography to ensure data integrity, authentication, and secure storage.

#### Properties of a Hash Function

##### 1. Deterministic

- For the same input, the hash function always produces the same output.
- Ensures consistency in verification and data integrity checks.

## SPPU-BE-COMP-CONTENT – KSKA Git

### 2. Fixed Output Length

- Regardless of input size, the hash output (digest) has a fixed length, e.g., SHA-256 always produces 256 bits.
- This simplifies storage and comparison.

### 3. Pre-image Resistance

- It should be computationally infeasible to find the original input from its hash value.
- Ensures one-way security.

### 4. Second Pre-image Resistance

- Given an input and its hash, it should be hard to find another different input with the same hash.
- Protects against forgery and substitution attacks.

### 5. Collision Resistance

- It should be extremely difficult to find two different inputs that produce the same hash value.
- Ensures uniqueness of output.

### 6. Avalanche Effect

- A small change in input causes a drastically different hash output.
- Improves unpredictability and security.

A good cryptographic hash function must be deterministic, collision-resistant, and exhibit the avalanche effect to ensure data integrity, authentication, and secure encryption in blockchain and other cryptographic applications.

Q2)

a) List and discuss benefits of Merkle tree. [5]

#### Definition:

A Merkle tree is a binary hash tree where each leaf node contains the hash of a data block, and each non-leaf node contains the hash of its child nodes. It is used to verify data integrity efficiently.

#### Benefits:

## SPPU-BE-COMP-CONTENT – KSKA Git

1. **Efficient Verification:**

Merkle trees allow verification of specific data blocks without downloading the entire dataset.

This is highly beneficial for lightweight blockchain clients (SPV nodes), saving time and computational resources.

2. **Data Integrity and Tamper Detection:**

Any modification in a data block changes its hash, which affects all parent hashes up to the root.

This property ensures that any tampering is quickly detected, maintaining strong data security.

3. **Reduced Storage and Bandwidth:**

Instead of storing or transmitting all data, only the root hash and necessary branch hashes are needed.

This significantly reduces storage requirements and network bandwidth usage in distributed systems.

4. **Faster Data Retrieval:**

The hierarchical structure allows quick verification and locating of specific data blocks without scanning the entire dataset.

This improves efficiency in handling large volumes of data.

5. **Scalability for Large Systems:**

By dividing data into smaller hashed chunks, Merkle trees support large-scale distributed systems.

This design facilitates easy data management and verification as the dataset grows over time.

6. **Supports Partial Data Verification:**

Merkle trees enable verification of subsets of data independently, which is useful in peer-to-peer networks and blockchain for validating transactions without accessing the whole ledger.

7. **Enhances Network Security:**

Because only hashes are exchanged, Merkle trees minimize exposure of raw data during verification, reducing the risk of data leaks or interception.

Merkle trees provide a secure, efficient, and scalable method for verifying data integrity, making them crucial in blockchain and other distributed applications

**b) Explain DSA key generation and verification. [5]**

## SPPU-BE-COMP-CONTENT – KSKA Git

The Digital Signature Algorithm (DSA) is a standard for creating digital signatures, used to verify the authenticity and integrity of a digital message. It involves three phases: key generation, signing, and verification.

### 1. DSA Key Generation

This process creates a public and private key pair for a user. It involves two main stages: selecting global parameters and then calculating the user-specific keys.

- **Global Public Parameters (  $p$ ,  $q$ ,  $g$  ):** These parameters are common to a group of users.
  1. A prime number  $q$  is chosen. This is the key length.
  2. A larger prime number  $p$  is chosen such that  $(p-1)$  is a multiple of  $q$ .
  3. An integer  $g$  is chosen, where  $g = h^{((p-1)/q)} \bmod p$  for some integer  $h$ .
- **User Keys (  $x$ ,  $y$  ):** These are unique to each user.
  1. **Private Key (  $x$  ):** A random integer  $x$  is chosen such that  $0 < x < q$ . This key must be kept secret. 🤐
  2. **Public Key (  $y$  ):** The public key  $y$  is calculated from the private key using the formula:

$$y = g^x \bmod p$$

The user's complete **public key** consists of (  $p$ ,  $q$ ,  $g$ ,  $y$  ), while their **private key** is  $x$ .

### 2. DSA Signature Verification

This process allows anyone with the sender's public key to verify the signature on a received message. Assume the verifier has received the original message (M) and the digital signature, which consists of two values (r, s).

The verifier also has the sender's public key (y) and the global parameters (p, q, g).

1. **Hash the Message:** First, the verifier calculates the hash of the received message using the same hash function (e.g., SHA-256) as the sender:  $H(M)$ .
2. **Calculate w :** The verifier computes an intermediate value w from the s part of the signature:

$$w = s^{-1} \bmod q$$

## SPPU-BE-COMP-CONTENT – KSKA Git

3. **Calculate  $u_1$  and  $u_2$**  : Two more values are calculated using the message hash, the public key, and  $w$  :

$$u_1 = (H(M) \cdot w) \mod q$$

$$u_2 = (r \cdot w) \mod q$$

4. **Calculate  $v$**  : The verifier then computes the final value  $v$  :

$$v = ((g^{u_1} \cdot y^{u_2}) \mod p) \mod q$$

5. **Final Check**: The signature is considered **valid** if and only if  $v = r$  .

If  $v$  and  $r$  match, it proves that the message was signed by the legitimate holder of the private key  $x$  and that the message has not been altered since it was signed.

... "उत्तर बरोबर आहे का माहित नाही, एकदा चेक करा."

### c) Discuss role of hashing in Blockchain. [5]

Hashing is a fundamental process in blockchain technology that converts data into a fixed-size unique value called a hash. It ensures the security, integrity, and smooth operation of the blockchain network.

#### Role of Hashing in Blockchain

##### 1. Data Integrity and Immutability:

- Each block contains a hash of its own data and the hash of the previous block.
- Any change in a block's data changes its hash, breaking the chain and alerting the network.
- This ensures the blockchain is tamper-proof and data remains immutable.

##### 2. Efficient Data Verification:

- Hashes provide a fixed-size, compact representation of data.
- Nodes verify data integrity by comparing hashes instead of the entire data, speeding up consensus.

##### 3. Proof of Work and Mining:

- Mining involves finding hashes with specific properties (e.g., starting with zeros).

## SPPU-BE-COMP-CONTENT – KSKA Git

- This process, called Proof of Work, uses hashing to secure the network by making data alteration computationally expensive.

### 4. Address Generation and Digital Signatures:

- Hashing helps generate unique user addresses and secure digital signatures, enabling authentication and non-repudiation.

### 5. Ensuring Privacy:

- Sensitive data can be stored as hashes rather than plain text, adding a privacy layer while maintaining verifiability.

Hashing is the backbone of blockchain's security and trustworthiness. It enables a decentralized, tamper-proof system essential for blockchain's widespread applications.

## ➤ SEP 2023

### Q1)

#### a) Illustrate Elliptic Curve Cryptography. [6]

Elliptic Curve Cryptography (ECC) is a public-key cryptographic system based on the algebraic structure of elliptic curves over finite fields. It offers strong security with smaller key sizes compared to other systems like RSA.

#### 1. Elliptic Curve Equation:

- An elliptic curve is defined by the equation:

$$y^2 = x^3 + ax + b$$

where  $a$  and  $b$  are constants satisfying  $4a^3 + 27b^2 \neq 0$  (to avoid singularities).

- The curve forms a smooth, continuous shape when plotted on a coordinate plane.

#### 2. Points on the Curve:

- The set of all points  $(x, y)$  satisfying the curve equation, along with a special point at infinity, form an abelian group.
- This group supports an addition operation where two points  $P$  and  $Q$  can be added to get another point  $R = P + Q$  on the curve.



## SPPU-BE-COMP-CONTENT – KSKA Git

### 3. Public and Private Keys:

- A private key is a randomly chosen integer  $d$ .
- The corresponding public key is a point  $Q = d \times G$ , where  $G$  is a fixed base point (generator) on the curve.
- Multiplying  $G$  by  $d$  means adding  $G$  to itself  $d$  times.

### 4. Security Basis (ECDLP):

- The security of ECC relies on the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally hard:  
Given  $Q$  and  $G$ , finding  $d$  such that  $Q = d \times G$  is infeasible.

### 5. Advantages of ECC:

- Provides equivalent security with much smaller keys than RSA or DSA (e.g., 256-bit ECC  $\approx$  3072-bit RSA).
- Faster computations, lower power consumption, and reduced storage requirements, making it suitable for mobile and embedded devices.

... "उत्तर बरोबर आहे का माहित नाही, एकदा चेक करा."

### b) Justify the importance of Hashing in Block Chain. [4]

#### Importance of Hashing in Blockchain

Hashing is a critical process in blockchain that transforms any input data into a fixed-length unique hash value. It ensures the security and integrity of the blockchain network.

#### 1. Data Integrity and Immutability:

- Hashing links blocks by including the previous block's hash in the current block.
- Any modification in a block changes its hash, breaking the chain and alerting the system to tampering.
- This makes blockchain tamper-resistant and immutable.

#### 2. Efficient Verification:

- Hashes provide a compact representation of data, allowing quick comparison and validation without examining entire data.
- This speeds up consensus and verification among nodes.

#### 3. Security through Proof of Work:

- Mining requires finding a hash that meets certain criteria, making the process computationally difficult.

## SPPU-BE-COMP-CONTENT – KSKA Git

- This protects the network from attacks by making it expensive to alter data.

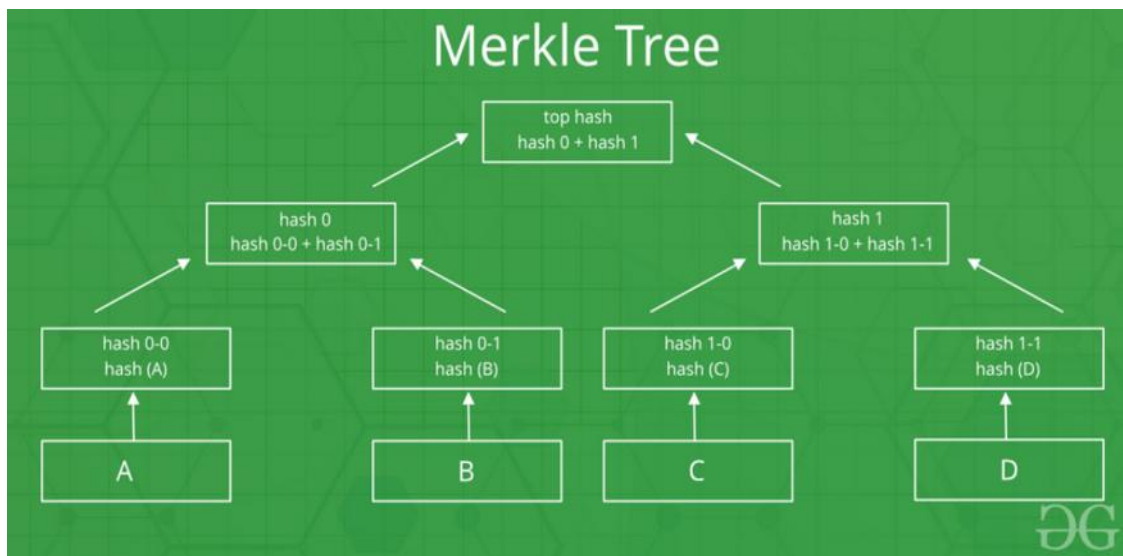
#### 4. Authentication and Privacy:

- Hashing generates unique digital signatures and user addresses.
- Sensitive information can be hashed to protect privacy while maintaining verifiability.



#### b) What is Merkle tree? Explain the structure of merkle tree. [5]

A Merkle Tree is a cryptographic data structure used in blockchain and other systems to efficiently and securely verify the integrity of large sets of data. It is also called a hash tree.



## SPPU-BE-COMP-CONTENT – KSKA Git

### Structure of Merkle Tree:

#### 1. Leaf Nodes:

- Each leaf node holds the hash of an individual data block or transaction.
- For example, if there are 4 transactions (A, B, C, D), each leaf node contains the hash of one transaction.

#### 2. Non-Leaf (Internal) Nodes:

- Each internal node contains the hash of the concatenation of its two child nodes' hashes.
- This process repeats upward until the top hash (Merkle Root) is formed.

#### 3. Merkle Root:

- The topmost node of the tree, representing a single hash summarizing all underlying data.
- Used to verify any transaction's integrity with minimal information.

### Advantages:

- Enables efficient and secure verification of large data sets without downloading the entire dataset.
- Commonly used in blockchain to verify transactions quickly and securely.

## Q2

### a) Explain working of SHA 256 Algorithm [6]

**SHA-256** (Secure Hash Algorithm 256-bit) is a cryptographic hash function that takes an input message and produces a fixed-size 256-bit (32-byte) hash. It is widely used for data integrity and security applications.

#### How SHA-256 Works :

SHA-256 takes any input data and processes it to produce a fixed 256-bit hash, which acts like a unique digital fingerprint of the data.

## SPPU-BE-COMP-CONTENT – KSKA Git

1. **Input Preparation:** The data is padded by adding a '1' bit, followed by enough '0' bits, and finally a 64-bit representation of the original data length. This padding makes the total length a multiple of 512 bits, suitable for block processing.
2. **Initial Setup:** The algorithm begins with eight constant hash values, derived from the fractional parts of the square roots of the first eight prime numbers. These serve as the starting point for hash computation.
3. **Block Processing:** The padded data is divided into 512-bit blocks. Each block is split into 16 chunks of 32 bits, which are then expanded to 64 chunks through a series of logical bitwise operations like shifts and XORs.
4. **Compression Function:** For each block, SHA-256 performs 64 rounds of complex operations including bitwise logical functions (AND, OR, XOR), modular addition, and bit shifts. These steps thoroughly mix the data to create a secure and unpredictable output.
5. **Final Hash:** After processing all blocks, the final eight hash values are combined to produce the 256-bit hash output. This final hash acts as a unique fingerprint of the input data, ensuring strong security by making it infeasible to reverse or find collisions.

SHA-256 transforms any input message into a fixed 256-bit hash by processing the message in 512-bit blocks with a series of bitwise operations and modular additions, ensuring high sensitivity to input changes (avalanche effect) and resistance to collisions and preimage attacks.

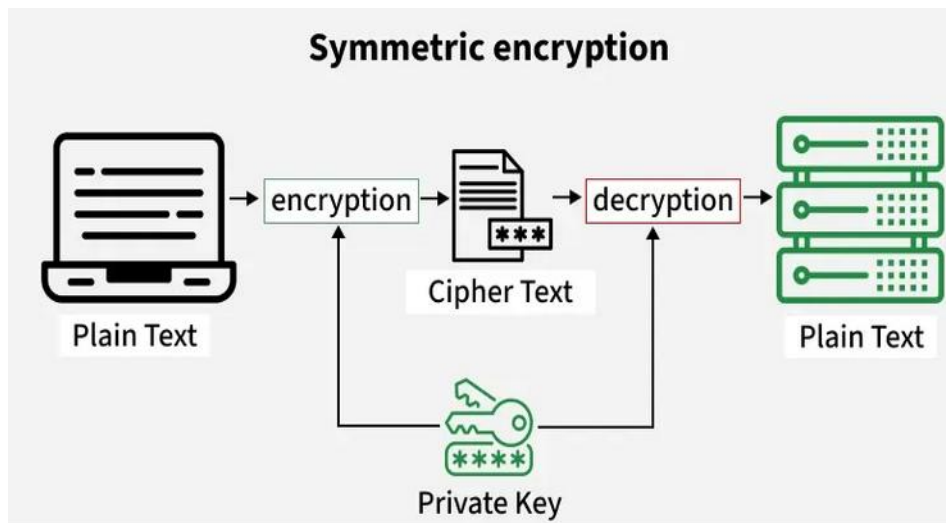
This makes SHA-256 widely used for digital signatures, certificates, blockchain, and password hashing.

### b) Describe Digital Signature & Verification steps in Digital Signature Algorithm. [4]

→ svatah shodha ani abhyas kara

### c) Describe Symmetric Key Encryption with neat diagram. [5]

**Symmetric Key Encryption** is a cryptographic method where the same secret key is used for both encryption and decryption of data. It is one of the oldest and most widely used encryption techniques for securing data.



**Description:**

- In symmetric key encryption, the sender and receiver share a single secret key beforehand.
- The sender uses this shared key to **encrypt** the plaintext (original message) into ciphertext (scrambled message) using an encryption algorithm.
- The ciphertext is then transmitted to the receiver.
- The receiver uses the **same secret key** to **decrypt** the ciphertext back into the original plaintext.
- Since the key is shared, the security depends entirely on keeping the key secret.

**Key Characteristics:**

- Faster and efficient for large amounts of data.
- Key distribution and management can be challenging because the same key must be securely shared.
- Common symmetric algorithms include AES (Advanced Encryption Standard), DES (Data Encryption Standard), and Blowfish.

➤ SEP 2024

Q1)

a) Differentiate between symmetric and asymmetric key cryptography. [5]

➔ Already done !

**b) What is hashing? Explain role of hashing in Blockchain [5]**

➔ Already done !

**c) What is Merkle tree? Explain with diagram [5]**

➔ Already done but can write this too

Merkle tree also known as hash tree is a data structure used for data verification and synchronization.

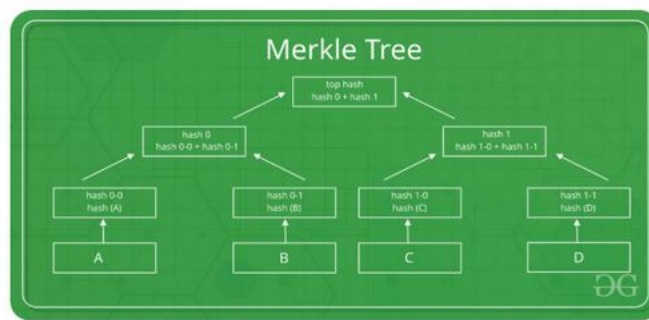
It is a tree data structure where each non-leaf node is a hash of its child nodes. All the leaf nodes are at the same depth and are as far left as possible.

It maintains data integrity and uses hash functions for this purpose.

**Hash Functions:**

So before understanding how Merkle trees work, we need to understand how hash functions work.

A hash function maps an input to a fixed output and this output is called hash. The output is unique for every input and this enables fingerprinting of data. So, huge amounts of data can be easily identified through their hash.



This is a **binary merkel tree**, the top hash is a hash of the entire tree.

- This structure of the tree allows efficient mapping of huge data and small changes made to the data can be easily identified.
- If we want to know where data change has occurred then we can check if data is consistent with root hash and we will not have to traverse the whole structure but only a small part of the structure.
- The root hash is used as the fingerprint for the entire data.

**Q2)**

**a) Explain digital signature algorithm. [5]**

➔ Already done

**b) What are the benefits of Merkle tree in Blockchain [5]**

➔ Already Done !

**c) Discuss elliptic curve cryptography. [5]**

➔ Already DONE !

"Check/Verify Answer – Read at Your Own Risk"