UNIT- II

Decentralization and Cryptography Technical Foundations

20CSE15- BLOCKCHAIN TECHNOLOGIES

Cryptography Technical Foundations

Introduction - Cryptography- Confidentiality - Integrity – Authentication - Cryptographic primitives - Asymmetric cryptography - Public and private keys – RSA -Discrete logarithm problem - Hash functions - Elliptic Curve Digital signature algorithm

Introduction

Set

A set is a collection of distinct objects, for example, X = {1, 2, 3, 4, 5}.

Group

- A group is a commutative set with one operation that combines two elements of the set.
- **Closure** (closed) means that if, for example, elements A and B are in the set, then the resultant element after performing an operation on the elements is also in the set.
- **Associative** means that the grouping of elements does not affect the result of the operation.

If G is a non-empty set and "$\star$" is the binary operation defined on G such that the following laws or axioms are satisfied then, (G, $\star$) is called a group.

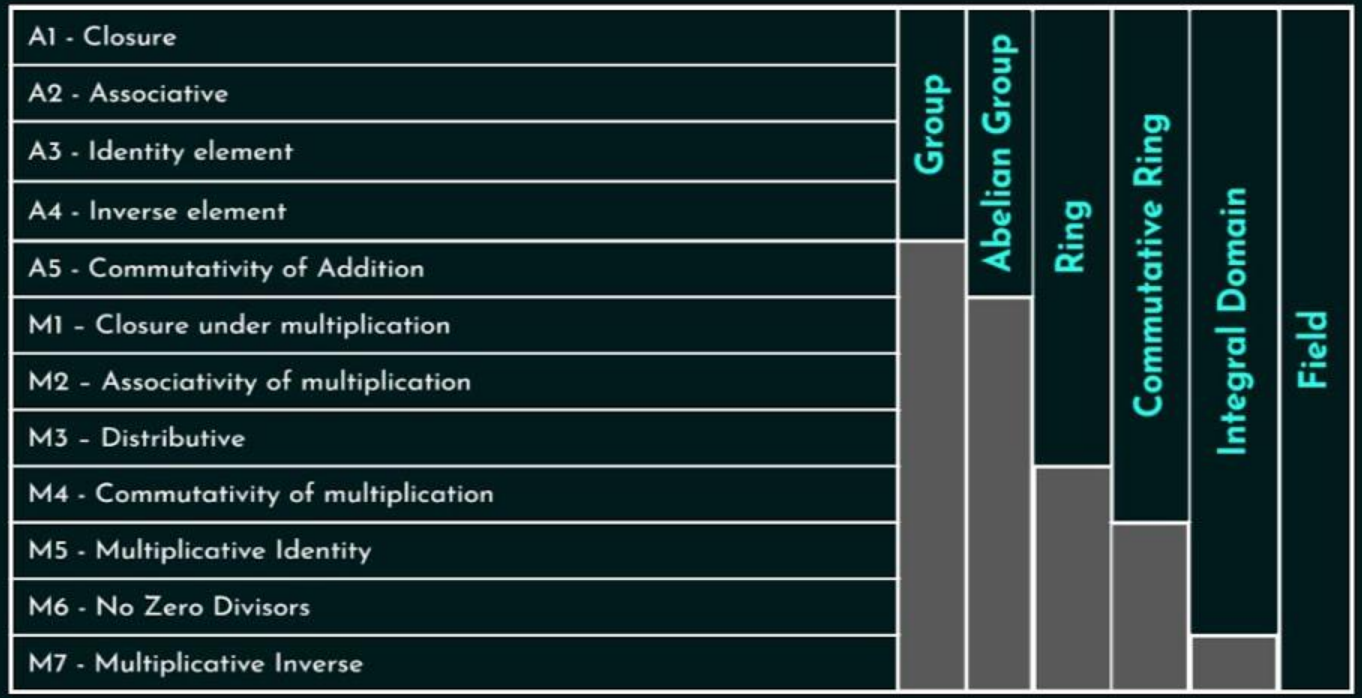
(G1) – Closure law

for a, b $\in$ G, a $\star$ b $\in$ G

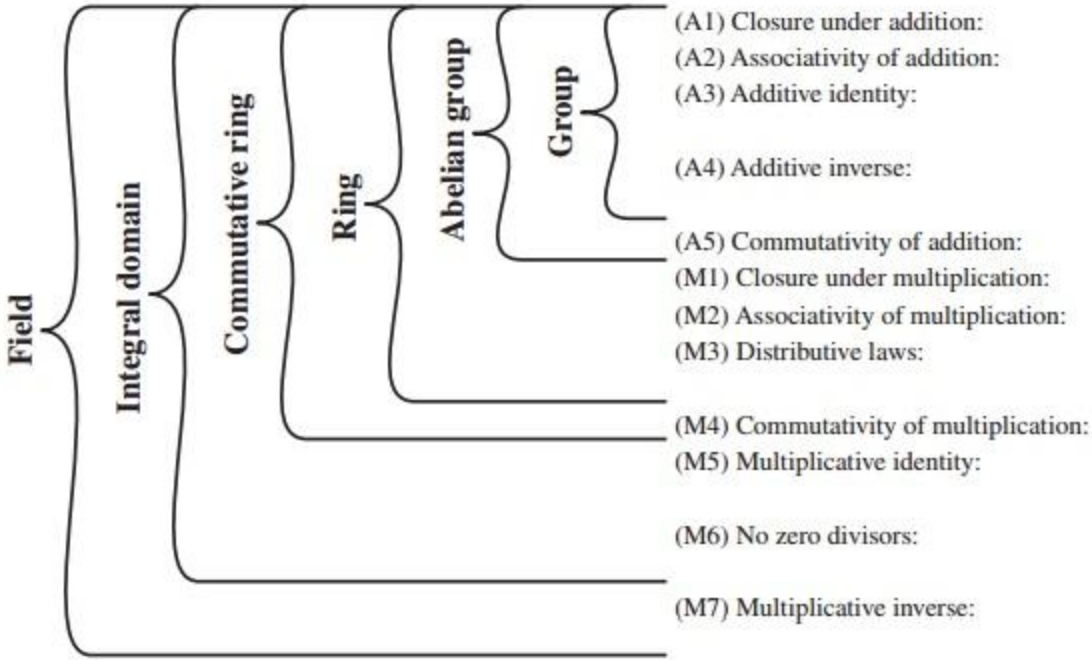
(G2) – Associative law

a $\star$ (b $\star$ c) = (a $\star$ b) $\star$ c for all a, b, c $\in$ G

Rings, Fields, & Finite Fields



56 Network Security



(A1) Closure under addition:	If a and b belong to S, then $a + b$ is also in S
(A2) Associativity of addition:	$a + (b + c) = (a + b) + c$ for all a, b, c in S
(A3) Additive identity:	There is an element 0 in R such that $a + 0 = 0 + a = a$ for all a in S
(A4) Additive inverse:	For each a in S there is an element $-a$ in S such that $a + (-a) = (-a) + a = 0$
(A5) Commutativity of addition:	$a + b = b + a$ for all a, b in S
(M1) Closure under multiplication:	If a and b belong to S, then ab is also in S
(M2) Associativity of multiplication:	$a(bc) = (ab)c$ for all a, b, c in S
(M3) Distributive laws:	$a(b + c) = ab + ac$ for all a, b, c in S $(a + b)c = ac + bc$ for all a, b, c in S
(M4) Commutativity of multiplication:	$ab = ba$ for all a, b in S
(M5) Multiplicative identity:	There is an element 1 in S such that $a1 = 1a = a$ for all a in S
(M6) No zero divisors:	If a, b in S and $ab = 0$, then either $a = 0$ or $b = 0$
(M7) Multiplicative inverse:	If a belongs to S and $a \neq 0$, there is an element a^{-1} in S such that $aa^{-1} = a^{-1}a = 1$

Figure 4.2 Groups, Ring, and Field

An abelian group

An abelian group is formed when the operation on the elements of a set is commutative.

The commutative law means that changing the order of the elements does not affect the result of the operation, for example, A X B = B X A.

Ring

If more than one operation can be defined over an abelian group, that group becomes a ring.

There are also specific properties that need to be satisfied.

A ring must have closure and associative and distributive properties.

Field

A field is a set in which all elements in the set form an additive and multiplicative group.

For all group operations, the distributive law is also applied. The law dictates that the same sum or product will be produced even if any of the terms or factors are reordered.

A finite field

- A finite field is one with a finite set of elements.
- Also known as Galois fields, they can be used to produce accurate and error-free results of arithmetic operations.
- For example, prime finite fields are used in Elliptic Curve Cryptography (ECC) to construct discrete logarithm problems.

Order

- The order is the number of elements in a field. It is also known as the cardinality of the field.

Prime fields

- A prime field is a finite one with a prime number of elements.
- It has specific rules for addition and multiplication, and each nonzero element in the field has an inverse.
- no proper subfield

A cyclic group

A cyclic group is a type of group that can be generated by a single element called the group generator.

CYCLIC GROUPS

A cyclic group is a group that can be generated by one element.

- An element (g) generates the group if every element of the group can be obtained by repeatedly applying the group operation or it's inverse to g.

<G,*> is cyclic if we can find a generator g

for any x in G, $x = \underbrace{g*g*\cdots*g}_{n \text{ times}} = g^n \Rightarrow G = \{g^n : n \text{ in } \mathbb{Z}\}$

<G, +>

for any x in G, $x = \underbrace{g+g+\cdots+g}_{n \text{ times}} = n \cdot g \Rightarrow G = \{n \cdot g : n \text{ in } \mathbb{Z}\}$

Cryptography

Cryptography is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it.

Cryptography

Entity: Either a person or system that sends, receives, or performs operations on data

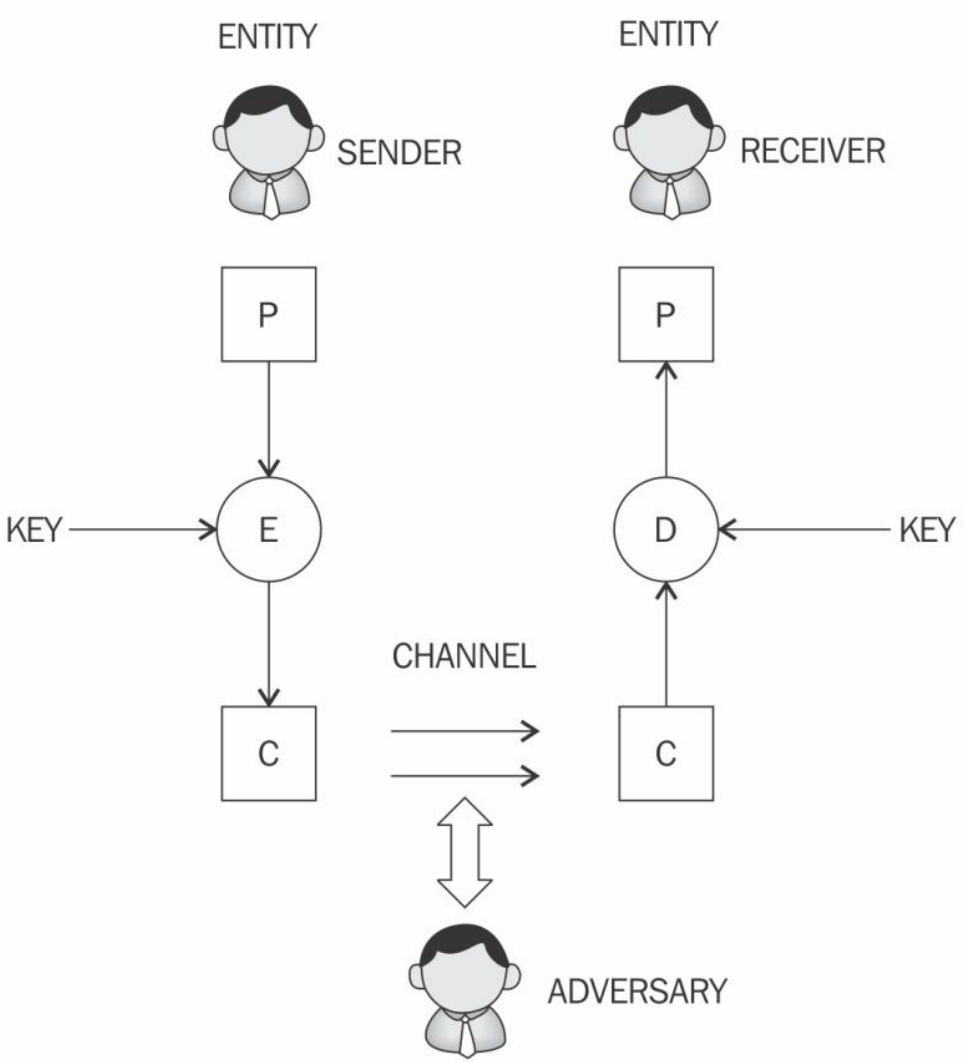
Sender: This is an entity that transmits the data

Receiver: This is an entity that takes delivery of the data

Adversary: This is an entity that tries to circumvent the security service

Key: A key is data that is used to encrypt or decrypt other data

Channel: Channel provides a medium of communication between entities


ENTITY
ENTITY
SENDER
RECEIVER
P
P
KEY
E
D
KEY
CHANNEL
C
C
ADVERSARY

Confidentiality

Confidentiality is the assurance that information is only available to authorized entities.

Integrity

Integrity is the assurance that information is modifiable only by authorized entities.

Authentication

Authentication provides assurance about the identity of an entity or the validity of a message.

two types - entity authentication and data origin authentication

Entity authentication

assurance that an entity is currently involved and active in a communication session.

single-factor authentication- only one factor involved, namely, something you know, that is, the password and username

multi-factor authentication - multiple factors

- something you have
- something you know
- something you are

Data origin authentication

- message authentication -assurance that the source of the information is indeed verified
- Various methods, such as Message Authentication Codes (MACs) and digital signatures are most commonly used

Non-repudiation

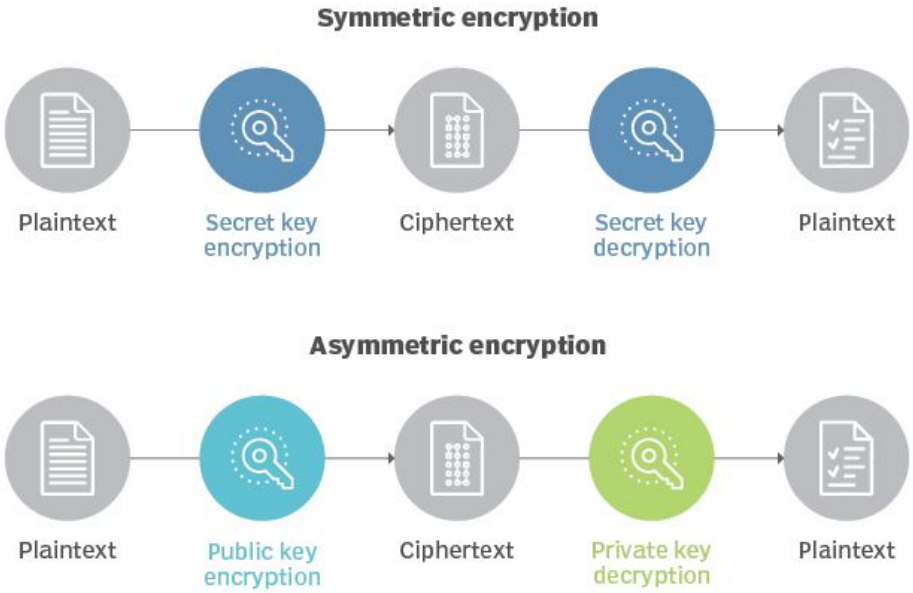
The creator/sender of the information cannot deny at a later stage their intentions in the creation or transmission of the information.

Cryptographic primitives

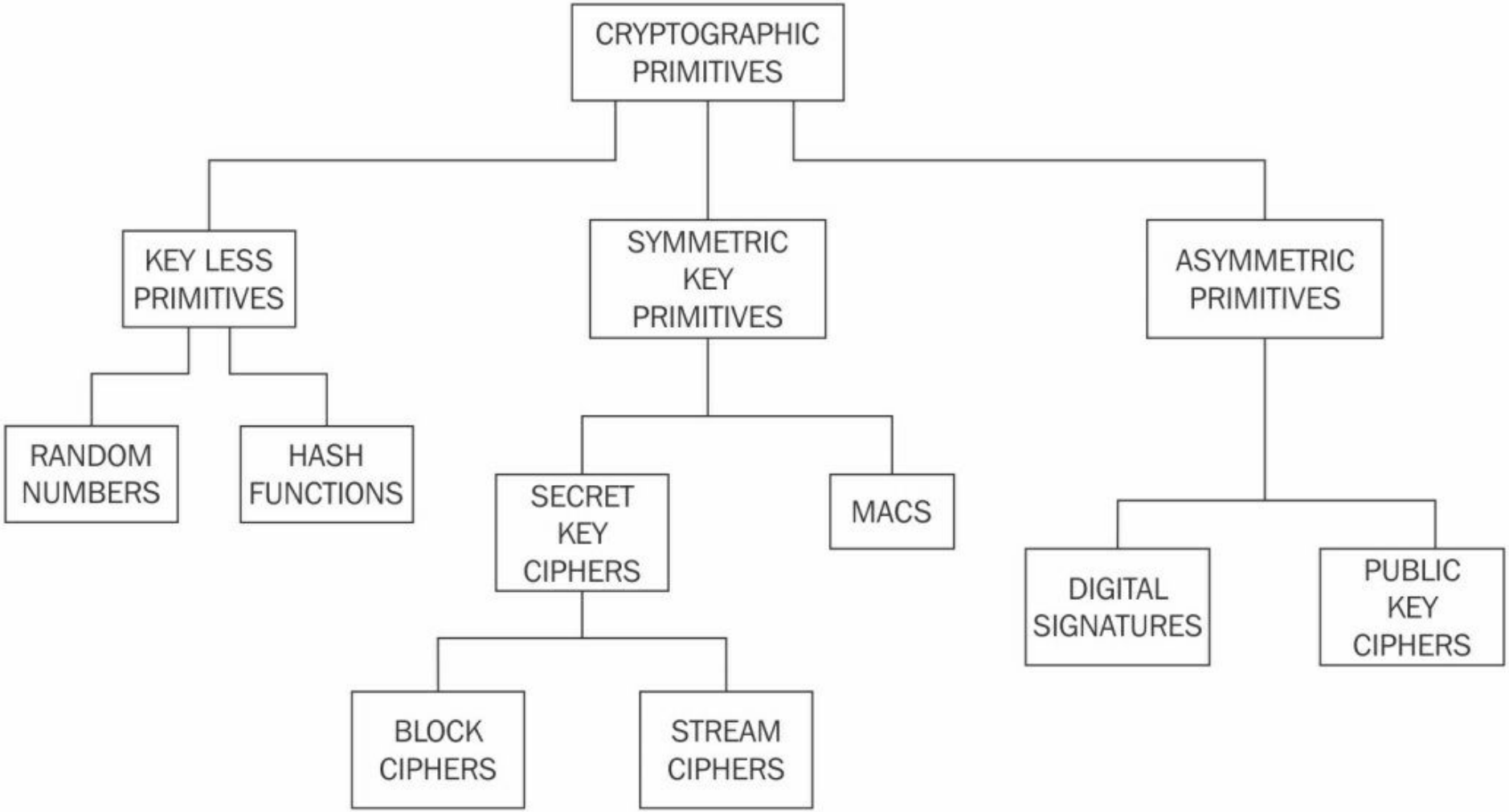
Cryptographic primitives are the basic building blocks of a security protocol or system.

A security protocol is a set of steps taken to achieve the required security goals by utilizing appropriate security mechanisms.

Symmetric vs. asymmetric encryption

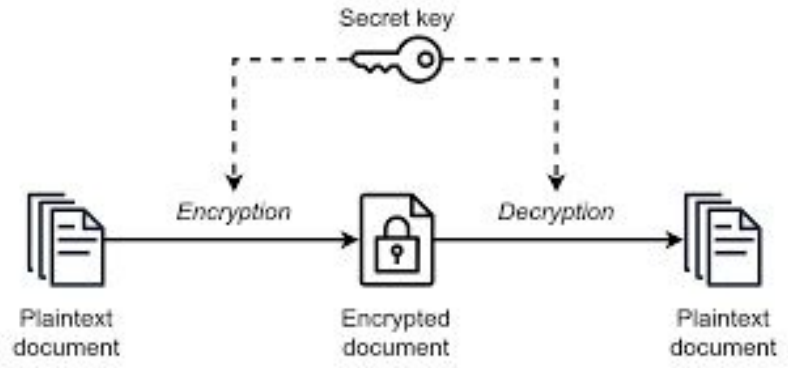


ICONS: ALEXDNDZ/ADOBE STOCK


CRYPTOGRAPHIC PRIMITIVES
KEY LESS PRIMITIVES
SYMMETRIC KEY PRIMITIVES
ASYMMETRIC PRIMITIVES
RANDOM NUMBERS
HASH FUNCTIONS
SECRET KEY CIPHERS
MACS
DIGITAL SIGNATURES
PUBLIC KEY CIPHERS
BLOCK CIPHERS
STREAM CIPHERS

Symmetric cryptography/shared key cryptography/secret key cryptography.

- Same key is used for encryption and decryption
- key must be established or agreed upon before the data exchange occurs between the communicating parties.

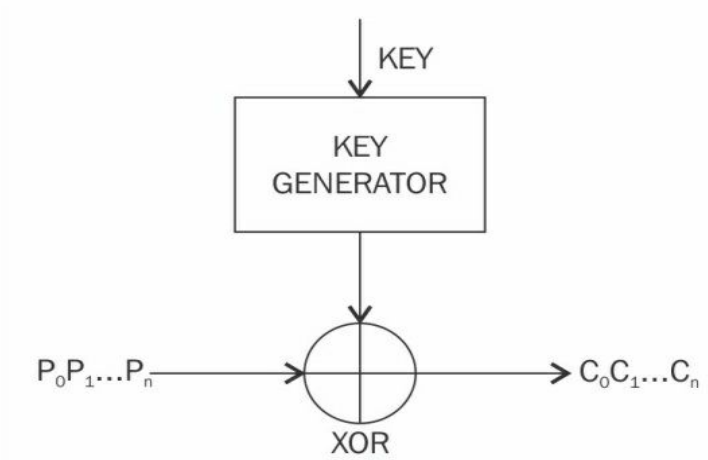


Symmetric cryptography (Contd..)

- Two types of symmetric ciphers: **stream ciphers and block ciphers**
- Eg: **block ciphers**- Data Encryption Standard (DES) and Advanced Encryption Standard (AES), **stream ciphers**- RC4 and A5

Stream ciphers

apply encryption algorithms on a bit-by-bit basis (one bit at a time) to plaintext using a keystream

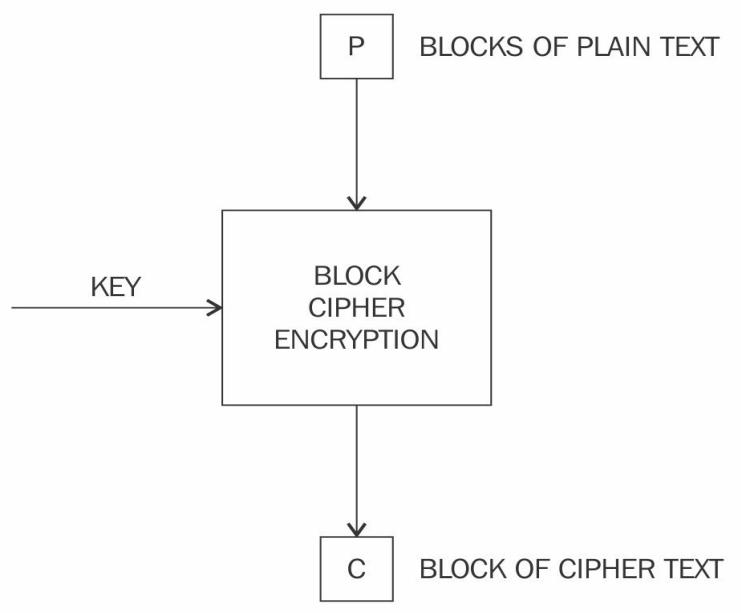


Two types of stream ciphers

- **Synchronous stream ciphers**- keystream is dependent only on the key, strict synchronization
- **Asynchronous stream ciphers**- keystream is dependent on the encrypted data, Lack of strict synchronization - there may be delays or disruptions in the communication.

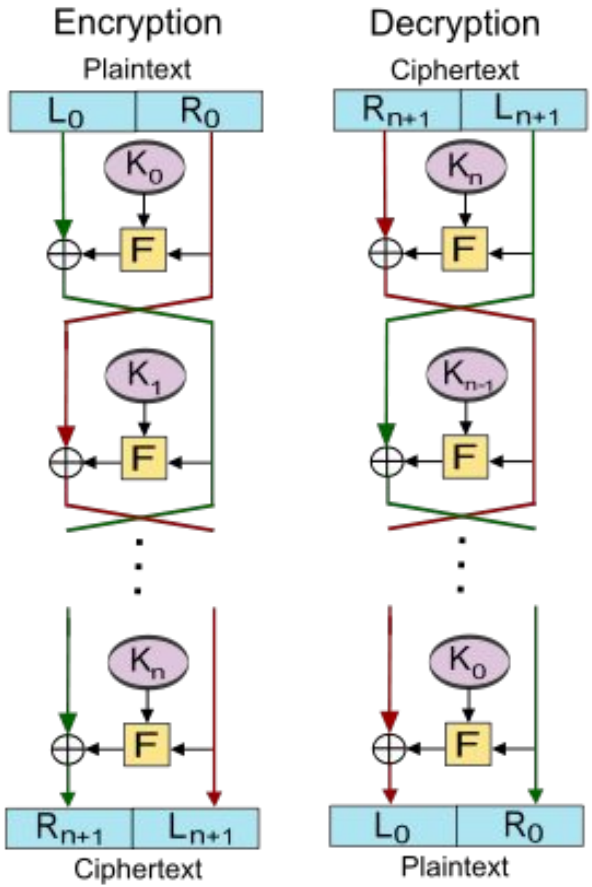
Block ciphers

- break up the text to be encrypted (plaintext) into blocks of a fixed length and apply the encryption block-by-block
- generally built using a design strategy known as a **Feistel cipher**
- Recent block ciphers, have been built using a combination of substitution and permutation called a Substitution-Permutation Network (SPN)



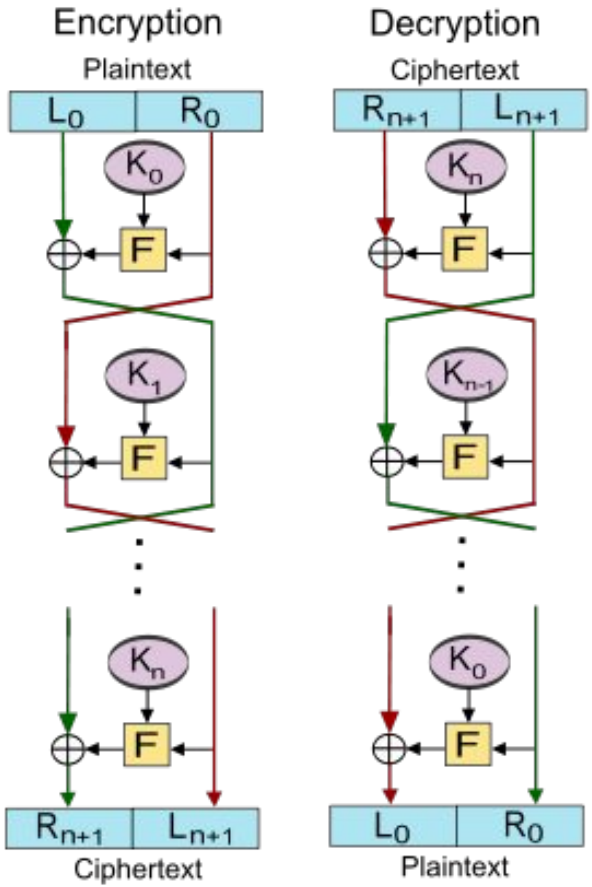
Feistel cipher

- The Feistel cipher is a specific design structure used to construct symmetric block ciphers.
- A key advantage of using a Feistel cipher is that encryption and decryption operations are almost identical and only require a reversal of the encryption process to achieve decryption.
- DES is a prime example of Feistel-based ciphers:



Feistel cipher

- **Key Expansion:** The secret encryption key is expanded into a set of round keys, one for each round of encryption.
- **Split and Concatenate:** The plaintext block is split into two equal-sized parts, typically referred to as the left and right halves.
- **Round Function:** The round function takes the current round's key as input and produces an output that is then XORed with the left half of the plaintext.
- **Swap:** After the XOR operation, the left and right halves are swapped.
- **Repeat:** The process of applying the round function and swapping the halves is repeated for a fixed number of rounds (typically 16 or 32) to create the final ciphertext.



Block encryption mode

- the plaintext is divided into blocks of fixed length depending on the type of cipher used
- the encryption function is applied to each block

Various modes of operation for block ciphers

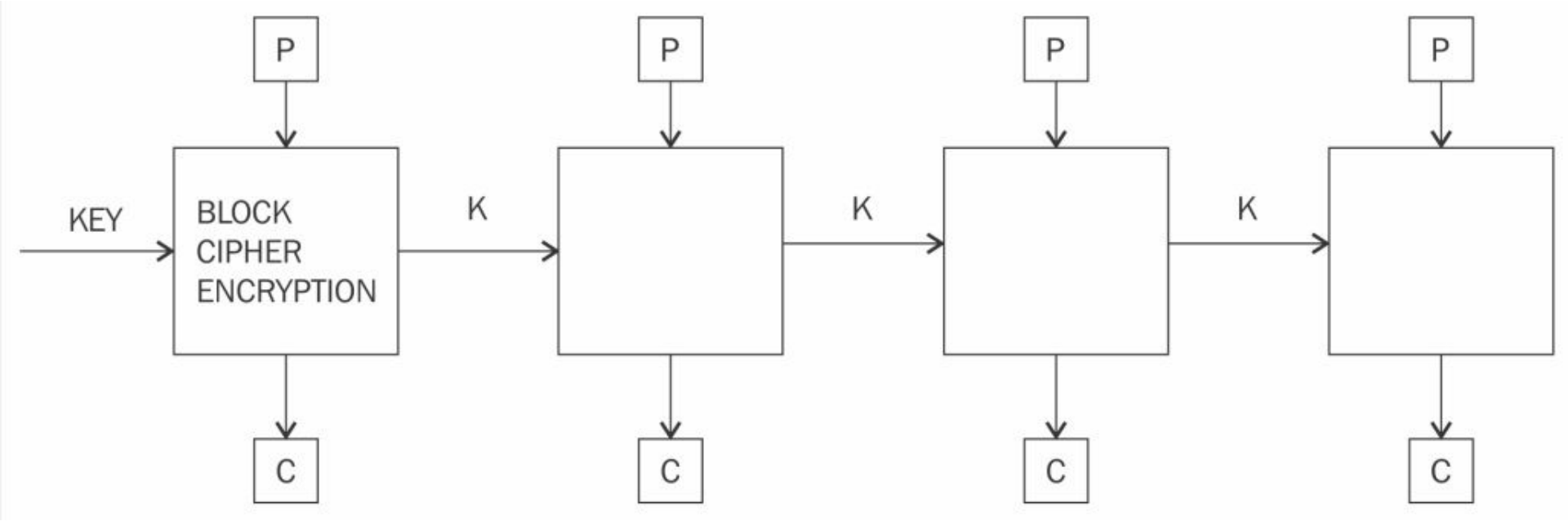
modes are used to specify the way in which an encryption function is applied to the plaintext

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Output Feedback (OFB) mode
- Counter (CTR) mode

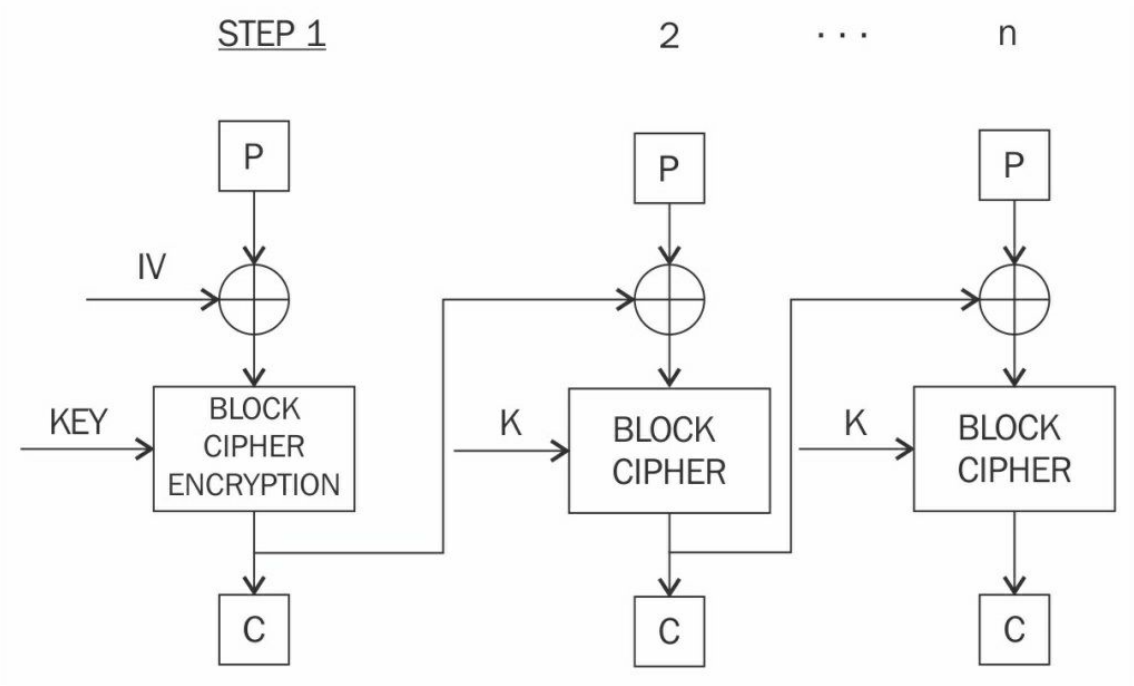
Electronic Code Book

- basic mode of operation in which the encrypted data is produced by applying the encryption algorithm one-by-one to each block of plaintext.
- most straightforward mode
- not be used in practice as it is insecure and can reveal information

Electronic Code Book



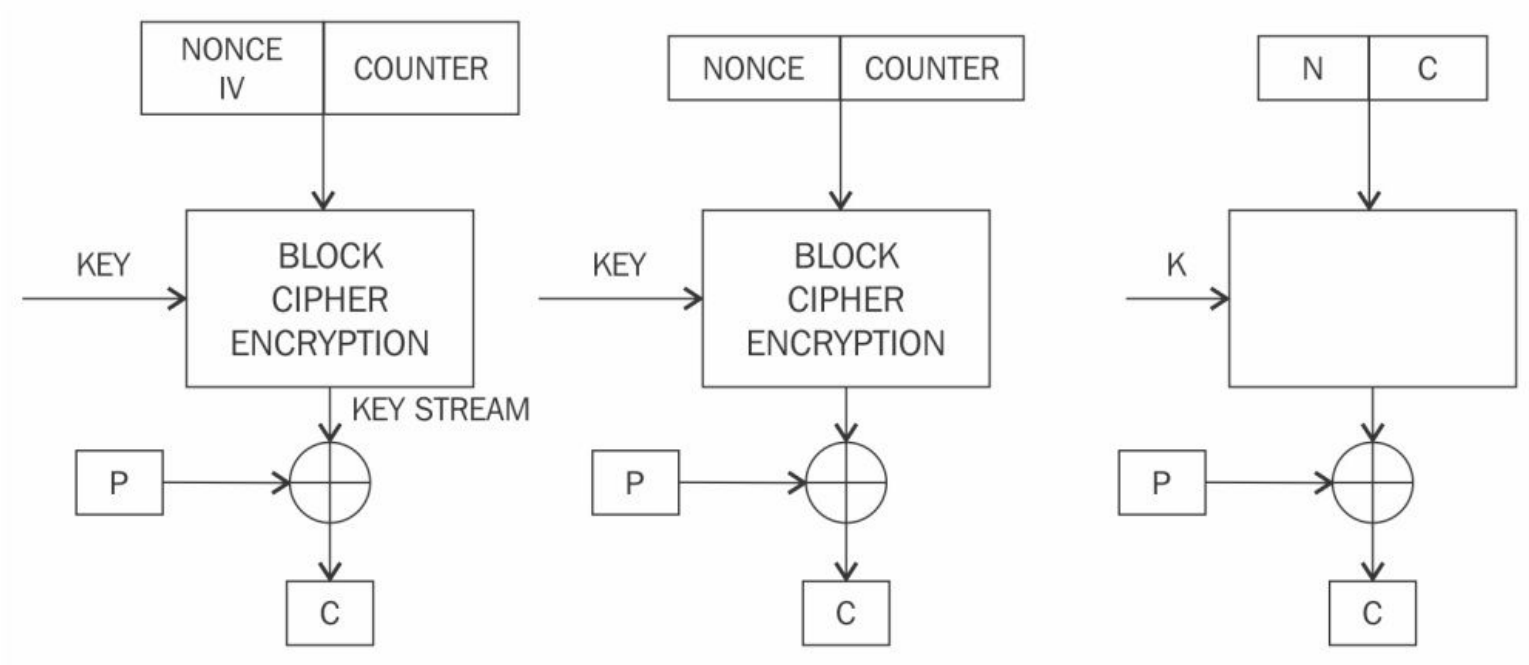
Cipher Block Chaining



Cipher Block Chaining

- In Cipher Block Chaining (CBC) mode, each block of plaintext is XOR'd with the previously-encrypted block.
- CBC mode uses the Initialization Vector (IV) to encrypt the first block.
- It is recommended that the IV be randomly chosen.

Counter mode



Counter mode

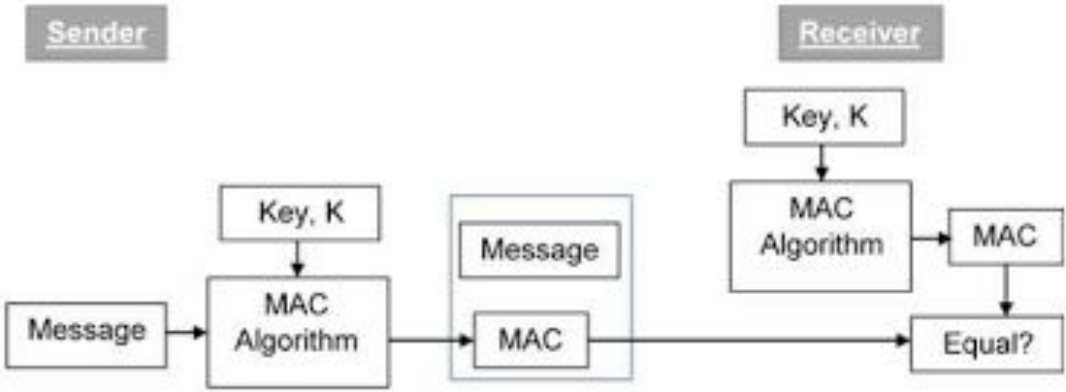
- The Counter (CTR) mode effectively uses a block cipher as a stream cipher.
- In this case, a unique nonce is supplied that is concatenated with the counter value to produce a keystream

Keystream generation mode

the encryption function generates a keystream that is then XOR'd with the plaintext stream to achieve encryption.

Message authentication mode

- MAC is a cryptographic checksum that provides an integrity service.
- common method to generate a MAC using block ciphers is CBC-MAC, where a part of the last block of the chain is used as a MAC



Cryptographic hash mode

- Hash functions are primarily used to compress a message to a fixed-length digest.
- In cryptographic hash mode, block ciphers are used as a compression function to produce a hash of plaintext.

Data Encryption Standard

- introduced by the U.S. National Institute of Standards and Technology (NIST) as a standard algorithm for encryption, and it was in widespread use during the 1980s and 1990s.
- not very resistant to brute force attacks
- DES uses a key of only 56 bits and this problem was addressed with the introduction of Triple DES (3DES), which proposed the use of a 168-bit key by means of three 56-bit keys and the same number of executions of the DES algorithm, thus making brute force attacks almost impossible.
- limitations- slow performance and 64-bit block size

Advanced Encryption Standard

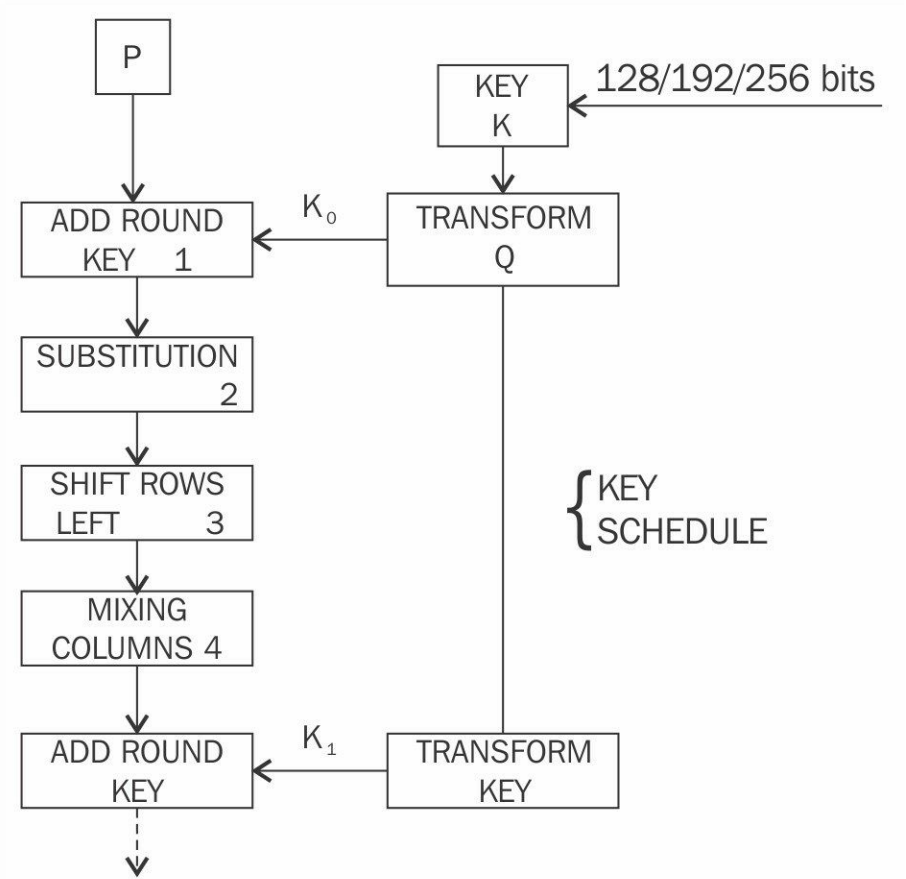
- The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm and one of the most secure cryptographic standards.
- It was selected as the official encryption algorithm by the U.S. National Institute of Standards and Technology (NIST) in 2001 after a public competition to replace the aging Data Encryption Standard (DES).
- AES operates on fixed-size blocks of data, and the standard defines three key sizes: AES-128, AES-192, and AES-256. These key sizes correspond to 128-bit, 192-bit, and 256-bit keys, respectively

Advanced Encryption Standard

- During AES algorithm processing, a 4 x 4 array of bytes known as the state is modified using multiple rounds.
- Full encryption requires 10 to 14 rounds, depending on the size of the key.

Key size	Number of rounds required
128-bit	10 rounds
192-bit	12 rounds
256-bit	14 rounds

Advanced Encryption Standard



Once the state is initialized with the input to the cipher, four operations are performed in four stages to encrypt the input.

1. In the **AddRoundKey** step, the state array is XOR'd with a subkey, which is derived from the master key

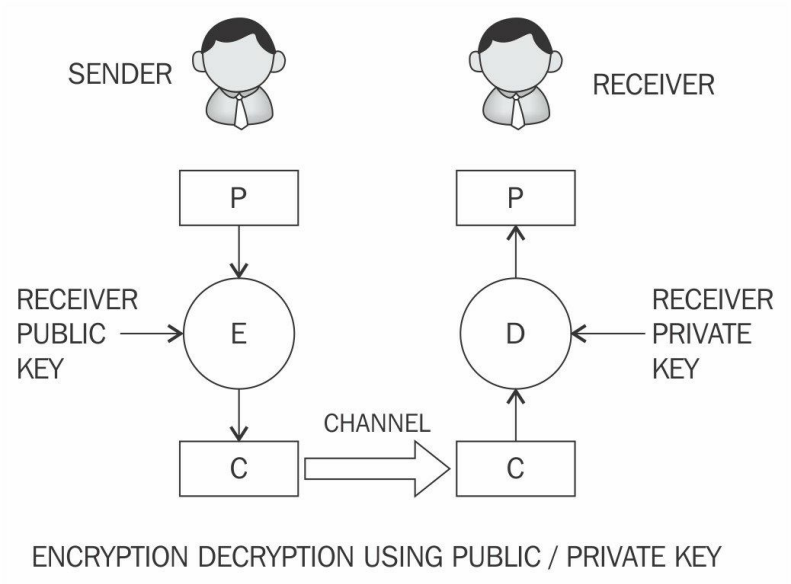
2. **SubBytes** is the substitution step where a lookup table (S-box) is used to replace all bytes of the state array

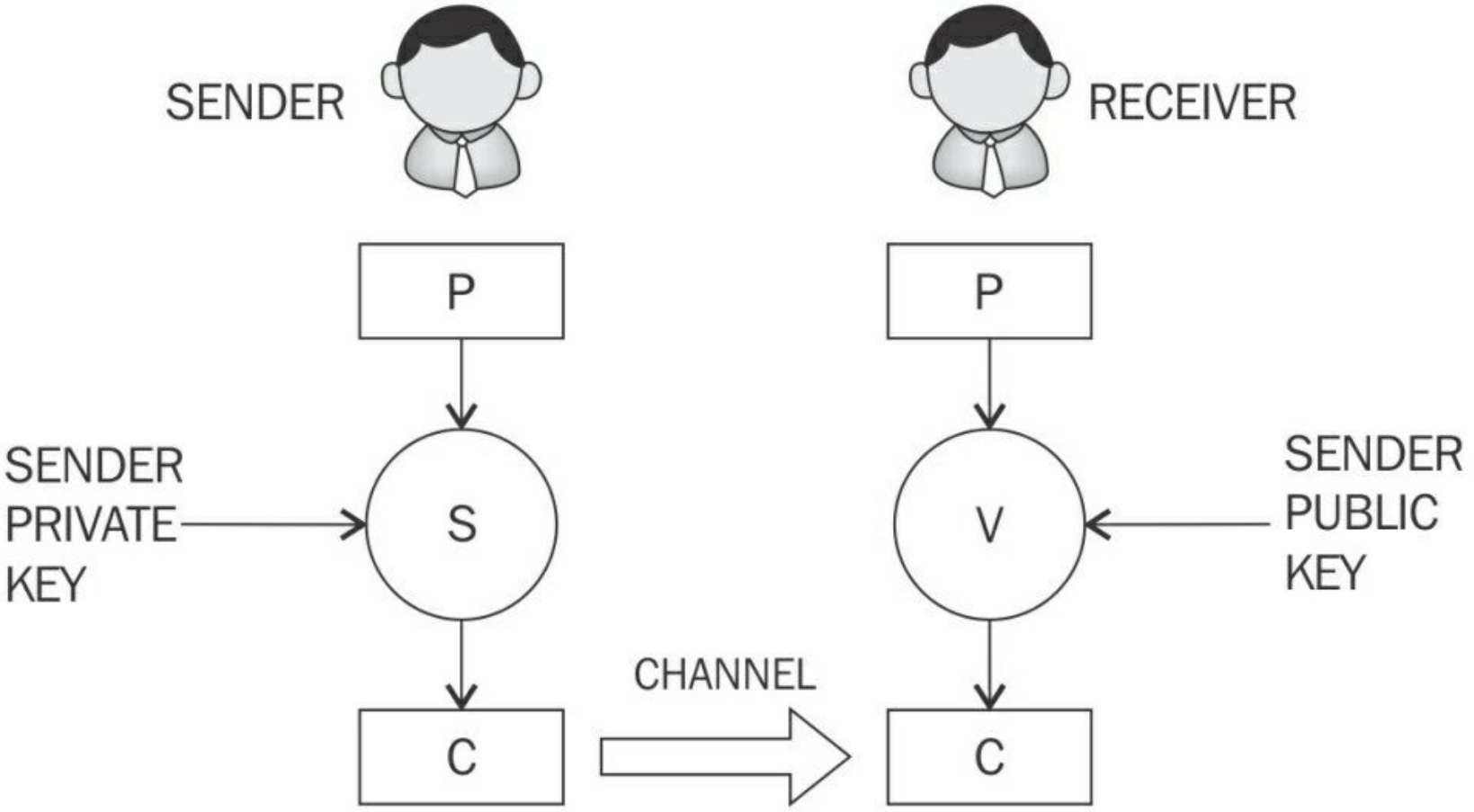
3. The **ShiftRows** step is used to shift each row to the left, except for the first one, in the state array to the left in a cyclic and incremental manner

4. Finally, all bytes are mixed in the **MixColumns** step in a linear fashion, column-wise

Asymmetric cryptography/ public key cryptography

- the key that is used to encrypt the data is different from the key that is used to decrypt the data.
- It uses both public and private keys to encrypt and decrypt data, respectively.
- Various asymmetric cryptography schemes are in use, including **RSA, DSA, and ElGammal**





SIGNING & VERIFICATION USING PUBLIC / PRIVATE KEY

Integer factorization

- Integer factorization schemes are based on the fact that large integers are very hard to factor (the multiplication of two large prime numbers is easy, but it is difficult to factor it (the result of multiplication,product))back to the two original numbers.
- RSA uses Integer factorization

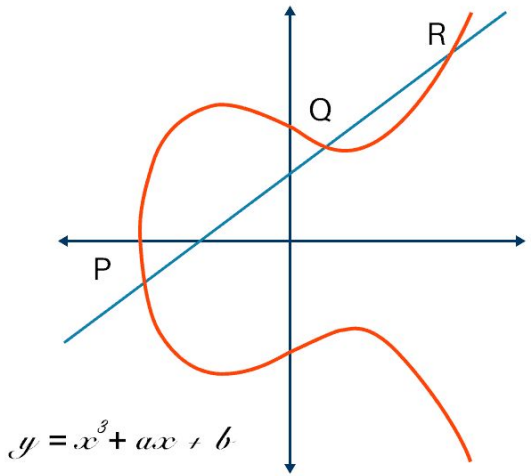
Discrete logarithm

- A discrete logarithm scheme is based on a problem in modular arithmetic.
- It is easy to calculate the result of modulo function, but it is computationally impractical to find the exponent of the generator.
- This difficult problem is commonly used in the Diffie-Hellman key exchange and digital signature algorithms.

$$3^2 \bmod 10 = 9$$

Elliptic curves

- algebraic cubic curve over a field
- prominently used cryptosystems based on elliptic curves are the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Elliptic Curve Diffie-Hellman (ECDH) key exchange.



Private key

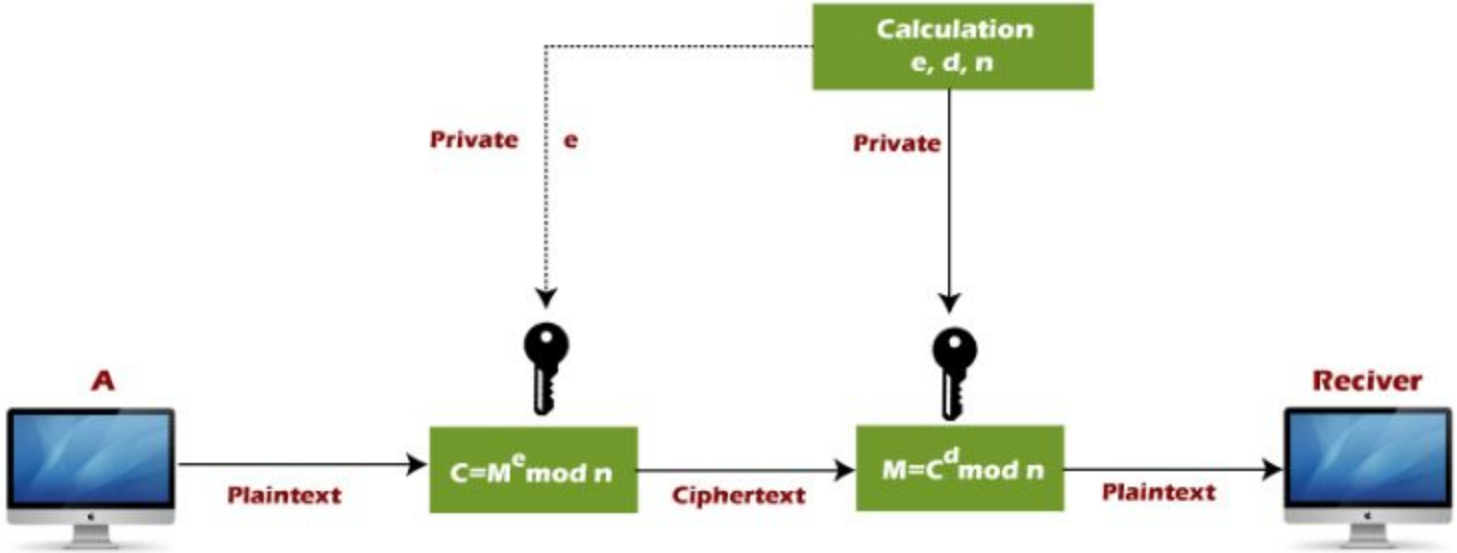
- randomly generated number that is kept secret and held privately by its users
- need to be protected and no unauthorized access should be granted to that key

Public key

- A public key is freely available and published by the private key owner
- Anyone who would then like to send the publisher of the public key an encrypted message can do so by encrypting the message using the published public key and sending it to the holder of the private key

RSA

- RSA was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adelman, hence the name Rivest–Shamir– Adleman (RSA).
- based on the integer factorization problem


Calculation
e, d, n
Private
e
Private
A
Reciver
Plaintext
C=M^e mod n
Ciphertext
M=C^d mod n
Plaintext

1. **Modulus generation**:

   - Select p and q, which are very large prime numbers
   - Multiply p and q, $n=p.q$ to generate modulus n

2. **Generate co-prime**:

   - Assume a number called e.
   - e should satisfy a certain condition; that is, it should be greater than 1 and less than $(p-1)\ (q-1)$. In other words, e must be a number such that no number other than 1 can divide e and $(p-1)\ (q-1)$. This is called **co-prime**, that is, e is the co-prime of $(p-1)\ (q-1)$.

3. **Generate the public key**:

   The modulus generated in step 1 and co-prime e generated in step 2 is a pair together that is a public key. This part is the public part that can be shared with anyone; however, p and q need to be kept secret.

4. **Generate the private key**:

   The private key, called d here, is calculated from p, q, and e. The private key is basically the inverse of e *modulo* $(p-1)\ (q-1)$. In the equation form, it is this as follows:

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

Encryption and decryption using RSA

RSA uses the following equation to produce ciphertext:

$$C = P^e \bmod n$$

This means that plaintext P is raised to e number of times and then reduced to modulo n. Decryption in RSA is provided in the following equation:

$$P = C^d \bmod n$$

This means that the receiver who has a public key pair (n, e) can decipher the data by raising C to the value of the private key d and reducing to modulo n.

https://drive.google.com/file/d/1Ss4Abi8FcttVI3f-JR0r_W2FWLpUmHUi/view?usp=drive_link

PROBLEM https://www.javatpoint.com/rsa-encryption-algorithm

Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

https://www.uobabylon.edu.iq/eprints/paper_1_17152_649.pdf

Elliptic Curve Cryptography

- The key length for secure RSA use has increased over recent years, and this has put a heavier processing load on applications using RSA
- A competing system challenges RSA: Elliptic Curve Cryptography (ECC)
- The principal attraction of ECC, compared to RSA, is that it appears to offer

  equal security for a far smaller key size, thereby reducing processing overhead
- ECC is commonly used for key exchange and digital signatures

Elliptic curves are not ellipses. They are so named because they are described by cubic equations, similar to those used for calculating the circumference of an ellipse.

In general, cubic equations for elliptic curves take the following form, known as a **Weierstrass equation**:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

ECC operates over a finite field, meaning the coordinates of points on the curve are taken from a finite set of numbers.

Such equations are said to be cubic, or of degree 3, because the highest exponent they contain is a 3.

An elliptic curve is defined in the following equation:

$$y^2 = x^3 + Ax + B \bmod P$$

A and B belong to a finite field Zp or Fp (prime finite field) along with a special value called the point of infinity.

The point of infinity (∞) is used to provide identity operations for points on the curve

A condition also needs to be met that ensures that the curve is non-singular

$$4a^3 + 27b^2 \neq 0 \, mod \; p$$

To construct the discrete logarithm problem based on elliptic curves, a large enough cyclic group is required.

- First, the group elements are identified as a set of points that satisfy the previous equation.
- After this, group operations need to be defined on these points.

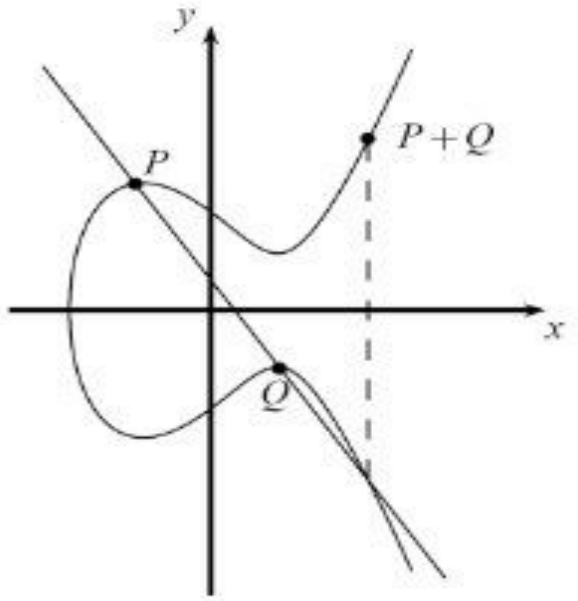
Group operations on elliptic curves

Point addition - process where two different points are added

Point doubling - the same point is added to itself

Point addition

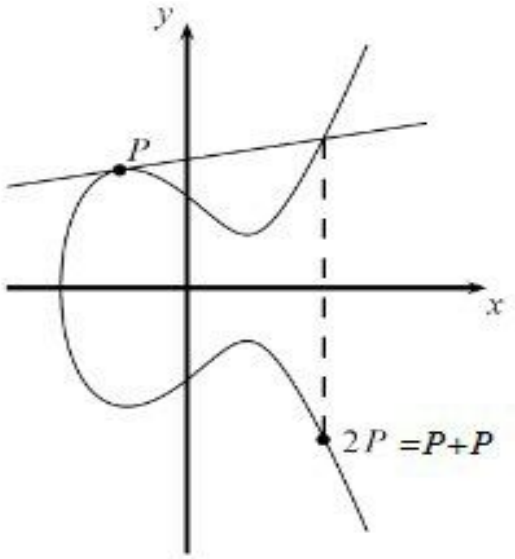
a diagonal line is drawn through the curve that intersects the curve at two points P and Q, as shown in the diagram, which yields a third point between the curve and the line



geometric representation of point addition on elliptic curves

Point doubling

a tangent line is drawn through the curve and the second point is obtained, which is at the intersection of the tangent line drawn and the curve



Graph representing point doubling over real numbers

Discrete logarithm problem in ECC

under certain conditions, all points on an elliptic curve form a cyclic group

Hash functions

- Create fixed-length digests of arbitrarily-long input strings
- Commonly used for Digital Signatures and Message Authentication Codes

  (MACs), such as HMACs

Three security properties

- Preimage resistance
- Second preimage resistance
- Collision resistance

Practical Properties of Hash function

Compression of arbitrary messages into fixed-length digest

hash function must be able to take an input text of any length and output a fixed-length compressed message.

Easy to compute

hash functions be very quick to compute regardless of the message size

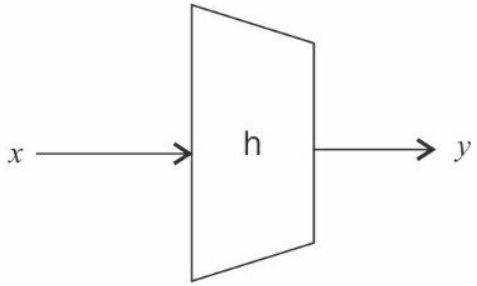
Preimage resistance/one-way property

$$h(x) = y$$

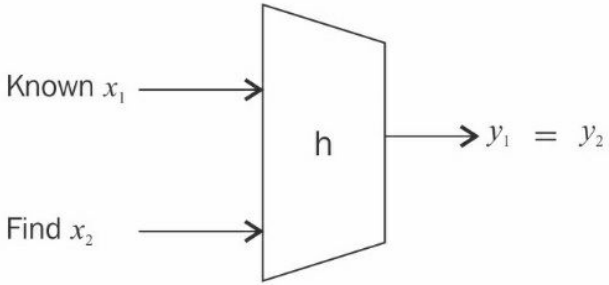
- y cannot be reverse-computed to x
- x is considered a preimage of y, hence the name preimage resistance

Second preimage resistance/ weak collision resistance

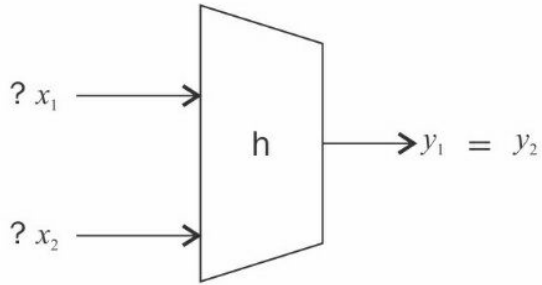
- given x and h(x), it is almost impossible to find any other message m, where m != x and hash of m = hash of x or h(m) = h(x)



1- PRE - IMAGE RESISTANCE



2- SECOND PRE IMAGE RESISTANCE

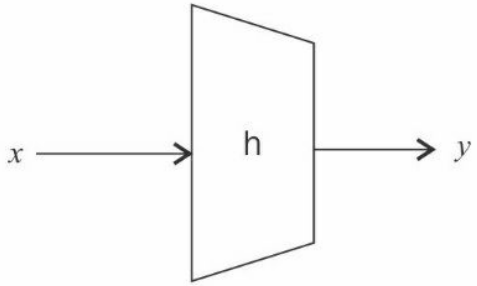


3- STRONG COLLISION RESISTANCE

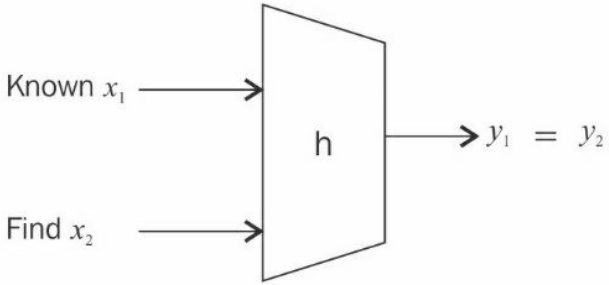
Collision resistance/ strong collision resistance

h(x) != h(z)

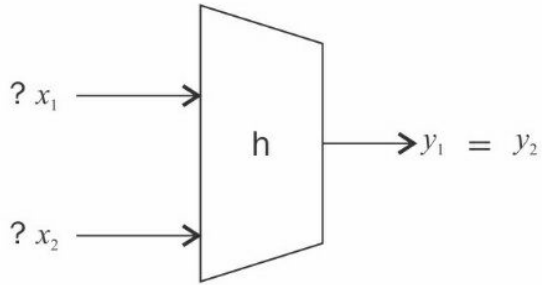
two different input messages should not hash to the same output



1- PRE - IMAGE RESISTANCE



2- SECOND PRE IMAGE RESISTANCE



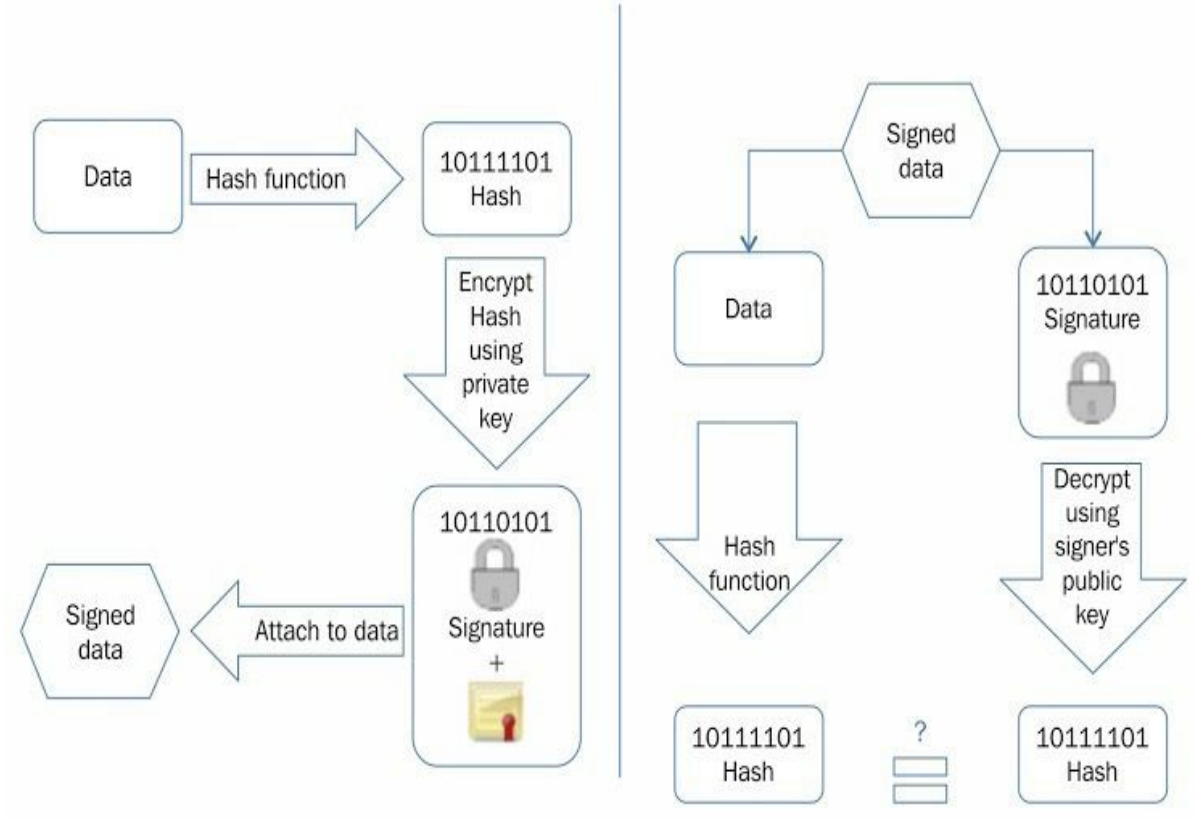
3- STRONG COLLISION RESISTANCE

Multiple categories of hash functions

- Message Digest
- Secure Hash Algorithms- SHA-0, SHA-1,SHA-2,SHA-3, RIPEMD, Whirlpool

Digital signatures

Digital signatures are used in blockchain where the transactions are digitally signed by senders using their private key before broadcasting the transaction to the network.



Elliptic Curve Digital Signature Algorithm

key pair needs to be generated

1. First, define an elliptic curve E:
   - With modulus P
   - Coefficients a and b
   - Generator point A that forms a cyclic group of prime order q
2. An integer d is chosen randomly so that $0 < d < q$.
3. Calculate public key B so that $B = d\,A$.

The public key is the sextuple in the form shown here:

$$Kpb = (p,a,b,q,A,B)$$

The private key, d is randomly chosen in step 2:

$$Kpr = d$$

Now the signature can be generated using the private and public key.

4. First, an ephemeral key K_e is chosen, where $0 < K_e < q$. It should be ensured that K_e is truly random and that no two signatures have the same key; otherwise, the private key can be calculated.
5. Another value R is calculated using $R = K_e\,A$; that is, by multiplying A (the generator point) and the random ephemeral key.
6. Initialize a variable r with the x coordinate value of point R so that $r = xR$.
7. The signature can be calculated as follows:

$$S = \left(h(m) + dr\right) K_{e^{-1}} \bmod q$$

Here, m is the message for which the signature is being computed, and $h(m)$ is the hash of the message m.

Signature verification is carried out by following this process:

1. Auxiliary value w is calculated as $w = s\text{-}1 \bmod q$.
2. Auxiliary value $u1 = w.\ h(m) \bmod q$.
3. Auxiliary value $u2 = w.\ r \bmod q$.
4. Calculate point P, $P = u1A + u2B$.

5. Verification is carried out as follows:

   r, s is accepted as a valid signature if the x coordinate of point P calculated in step 4 has the same value as the signature parameter $r \bmod q$; that is: