Unit 1

CHAPTER 1

Mathematical Foundation for Blockchain

University Prescribed Syllabus

Cryptography : Symmetric Key Cryptography and Asymmetric Key Cryptography, Elliptic Curve Cryptography (ECC), Cryptographic Hash Functions: SHA256, Digital Signature Algorithm (DSA), Merkel Trees.

1.1 CRYPTOGRAPHY : SYMMETRIC KEY CRYPTOGRAPHY AND ASYMMETRIC KEY CRYPTOGRAPHY

GQ. What is Cryptography? What is the basic purpose of it?
GQ. What are different security services that are offered by Cryptography?

- The study of secure communication methods, such as encryption, that only the message's originator and intended receiver can access, is known as cryptography. The word is derived from kryptos, a "hidden" word in Greek.
- The science of creating information security in the presence of adversaries is known as cryptography. It does so on the grounds that opponents have unrestricted access to resources.
- Data is encrypted or decrypted using ciphertext such that if it is intercepted by an enemy, it is useless to them without a secret key for decryption.

- Additionally, utilising methods like microdots or merging, cryptography includes the obscuring of data in pictures. These techniques were employed by the ancient Egyptians in their complex hieroglyphics, and Roman Emperor Julius Caesar is credited with discovering them.
- The main purpose of cryptography is to offer a secrecy service. It cannot be regarded as a whole solution on its own; rather, it functions as an essential component of a larger security system to handle a security issue. For instance, several distinct cryptographic primitives, including hash functions, symmetric key cryptography, digital signatures, and public key cryptography, are needed to secure a blockchain ecosystem.
- Cryptography offers non-repudiation, integrity, and authentication (including entity and data origin authentication) in addition to confidentiality as security services. Accountability is also offered, which is something that many security systems want.

☞ 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.
- There are two types of authentication mechanisms, namely entity authentication and data origin authentication.

☞ Entity authentication

- Entity authentication is the assurance that an entity is currently involved and active in a communication session.
- Traditionally, users are issued a username and password that

is used to gain access to the various platforms with which they are working.

☞ Data origin authentication

- Also known as message authentication, data origin authentication is an assurance that the source of the information is indeed verified. Data origin authentication guarantees data integrity because if a source is corroborated, then the data must not have been altered.
- Various methods, such as Message Authentication Codes (MACs) and digital signatures are most commonly used.

☞ Non-repudiation

- Non-repudiation is the assurance that an entity cannot deny a previous commitment or action by providing incontrovertible evidence. It is a security service that offers definitive proof that a particular activity has occurred.
- This property is essential in debatable situations whereby an entity has denied the actions performed, for example, placement of an order on an e-commerce system.
- This service produces cryptographic evidence in electronic transactions so that in case of disputes, it can be used as a confirmation of an action.

☞ Accountability

- Accountability is the assurance which states that actions affecting security can be traced back to the responsible party.
- This is usually provided by logging and audit mechanisms in systems where a detailed audit is required due to the nature of the business, for example, in electronic trading systems.
- Detailed logs are vital to trace an entity's actions, such as when a trade is placed in an audit record with the date and timestamp and the entity's identity is generated and saved in the log file.
- This log file can optionally be encrypted and be part of the database or a standalone ASCII text log file on a system.

1.1.1 Mathematical Concepts Required

GQ. What is a finite field? Why it is crucial for cryptography?

This part will expose you to some fundamental mathematical principles because the study of cryptography is dependent on mathematics.

☞ Set

A set is a grouping of unique things, such as X = {1, 2, 3, 4, 5}.

☞ Group

- A group is a commutative set that has a single operation that joins two of its elements. The group operation is finished, and it has a specific identification element attached to it. Each component of the set also has an inverse.
- If, for instance, elements A and B are in the set, then the resultant element after performing an operation on the elements is also in the set.
- This is referred to as closure (closed). Associative implies that the ordering of the elements has no affect on the outcome of the operation.

☞ A finite field

- A field with a finite set of components is said to be finite.
- These structures, also known as Galois fields, are crucial for cryptography because they can be utilised to generate precise and error-free outcomes for arithmetic operations. For instance, in Elliptic Curve Cryptography (ECC), discrete logarithm problems are created using prime finite fields.

☞ Order

The order indicates how many elements are in a field. It is often referred to as the field's cardinality.

☞ An abelian group

- If an operation on a set's elements is commutative, an abelian group is created.

- The commutative law states that the outcome of an operation, for instance, A X B = B X A, is unaffected by the arrangement of the elements.

☞ Prime fields

- A field having a prime number of elements is one that is finite. Each nonzero element in the field has an inverse, and there are precise rules for addition and multiplication.
- Operations like addition and multiplication are carried out modulo p, or prime.

☞ Ring

- An abelian group becomes a ring if more than one operation can be specified over it. There are particular requirements that must be met as well.
- Closure, associative, and distributive properties are necessary for a ring.

☞ A cyclic group

A certain kind of group known as the group generator has the ability to produce cyclic groups.

☞ Modular arithmetic

- Numbers in modular arithmetic wrap around when they reach a specific fixed number, which is also known as clock arithmetic.
- All operations are carried out with relation to this fixed number, which is a positive integer known as modulus.
- The numerals from 1 to 12 are similar to a clock. The number 1 resumes when it reaches 12.
- To put it another way, this kind of mathematics deals with the leftovers from division. For instance, 50 / 11 leaves a leftover of 6, therefore 50 mod 11 is equal to 6.

The fundamental introduction to several mathematical ideas used in cryptography is now complete. You will learn about the fundamentals of cryptography in the section that follows.

1.1.2 Generic Cryptography Model

GQ. Explain general cryptography model with suitable diagram. **(4 Marks)**

A generic cryptography model is shown in the following diagram.

It has following terms :

(1) **Entity** : Either a person or system that sends, receives, or performs operations on data

(2) **Sender** : This is an entity that transmits the data

(3) **Receiver** : This is an entity that takes delivery of the data

(4) **Adversary** : This is an entity that tries to circumvent the security service

(5) **Key** : A key is data that is used to encrypt or decrypt other data

(6) **Channel** : Channel provides a medium of communication between entities

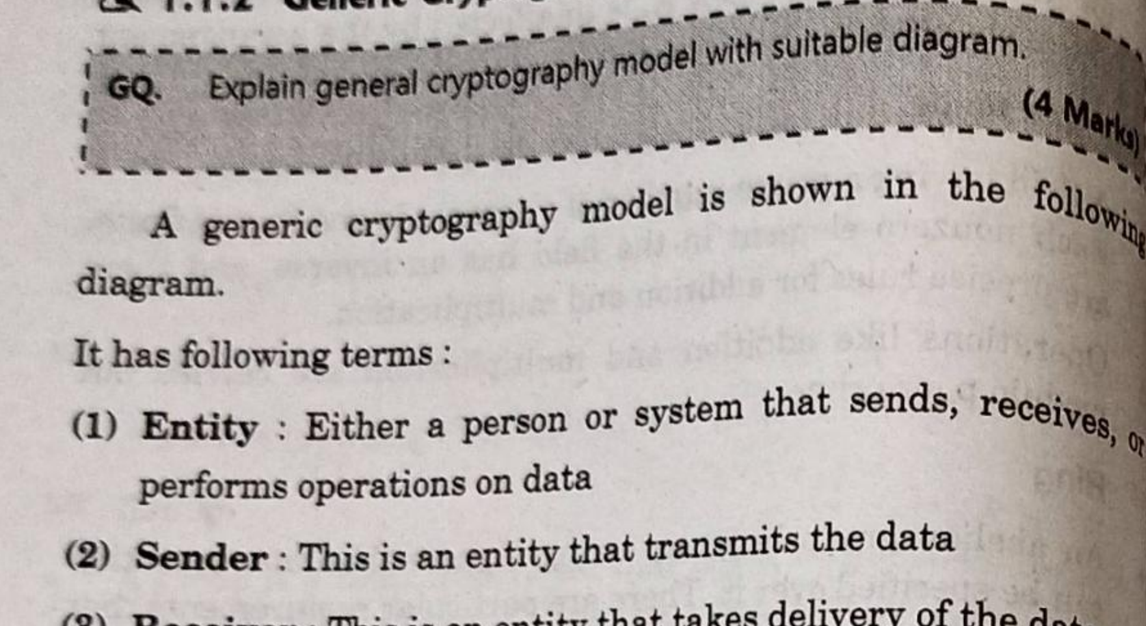

Fig. 1.1.1 : A generic encryption and decryption model

1.1.3 Cryptographic Primitives

GQ. Draw and explain the taxonomy of cryptographic primitives. **(4 Marks)**

GQ. Demonstrate symmetric key cryptography. What are pros and cons of it. **(6 Marks)**

GQ. Demonstrate asymmetric key cryptography. What are pros and cons of it. **(6 Marks)**

GQ. Compare symmetric and asymmetric key cryptography. **(4 Marks)**

GQ. Explain Stream Ciphers and block ciphers. **(4 Marks)**

- Cryptographic primitives are the basic building blocks of a security protocol or system. You will learn about cryptographic algorithms in the section that follows.
- These algorithms are crucial for creating safe protocols and systems. A security protocol is a series of actions made to use the proper security mechanisms in order to accomplish the necessary security goals.
- There are many different kinds of security protocols in use, including key management protocols, non-repudiation protocols, and authentication protocols.
- Here is an example of how the taxonomy of cryptographic primitives.

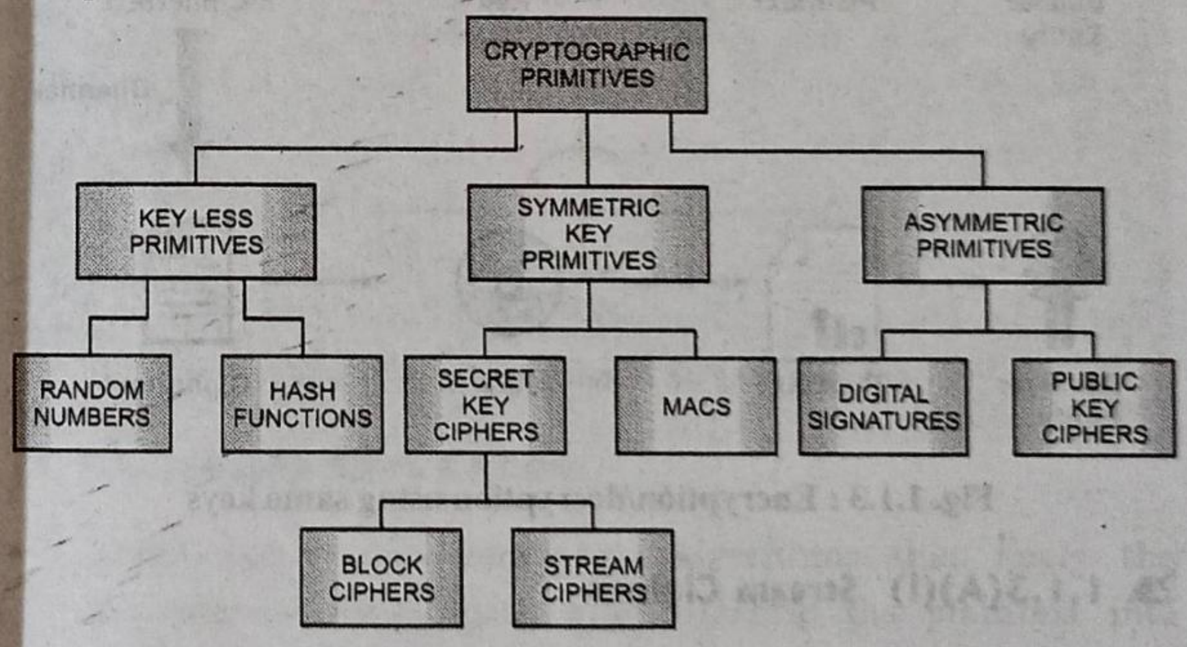

Fig. 1.1.2 : The taxonomy of cryptographic primitives

- As shown in the cryptographic primitives taxonomy diagram, cryptography is mainly divided into two categories : *symmetric cryptography* and *asymmetric cryptography*.
- These primitives are discussed further in the next section.

1.1.3(A) Symmetric Cryptography

- The term "symmetric cryptography" refers to a kind of encryption where the key used to encrypt and decrypt is the same. As a result, it is often referred to as shared key cryptography.
- The key needs to be decided upon or established before the communication parties transmit any data. Because of this, it is also known as secret key cryptography.
- Stream cyphers and block cyphers are the two categories of symmetric ciphers. Typical examples of block cyphers are Data Encryption Standard (DES) and Advanced Encryption Standard (AES), but stream ciphers like RC4 and A5 are often utilised.

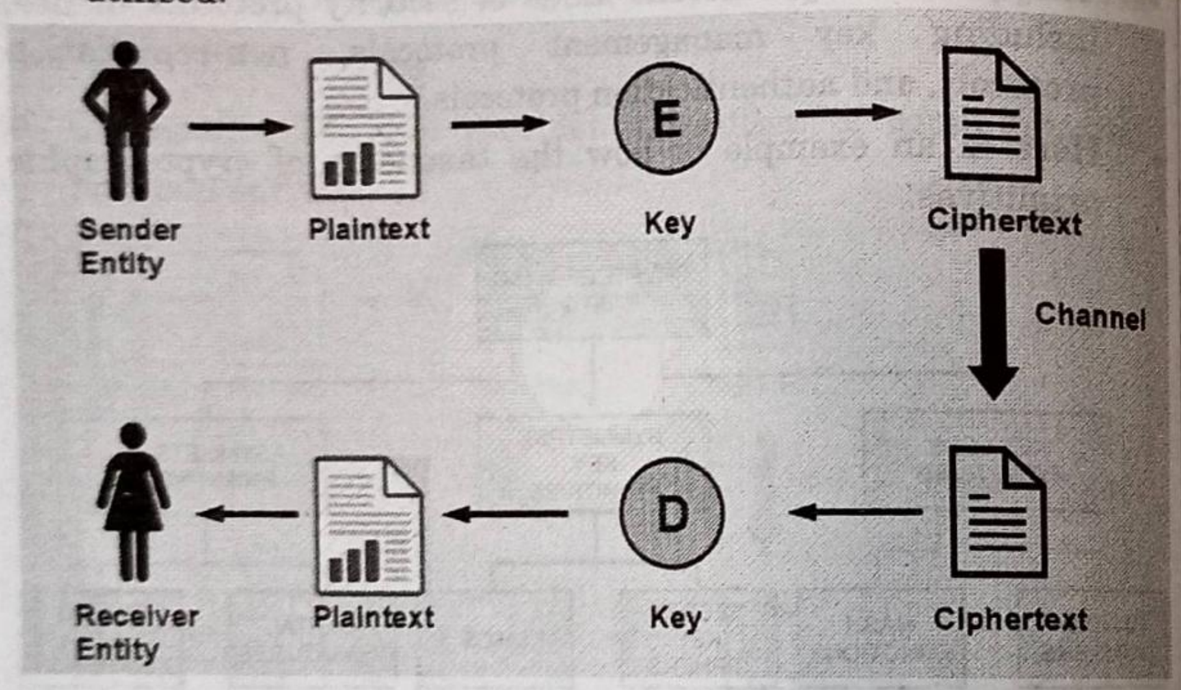

Fig. 1.1.3 : Encryption/decryption using same keys

1.1.3(A)(i) Stream Ciphers

- Using a keystream, stream ciphers encrypt plaintext by applying encryption algorithms bit-by-bit (one bit at a time) to

the data. Asynchronous and synchronous stream ciphers are the two different forms of stream ciphers.

- The keystream of a synchronised stream cipher depends exclusively on the key.
- A keystream for asynchronous stream cyphers depends on the encrypted data as well.
- Because they are just straightforward modulo-2 additions or XOR operations, encryption and decryption are the same function in stream ciphers.
- The security and unpredictability of keystreams are the essential requirements for stream ciphers. To create random numbers, a variety of methods have been devised, from hardware-implemented true random number generators to hardware-implemented pseudorandom number generators, and it is vital that all key generators be cryptographically secure.

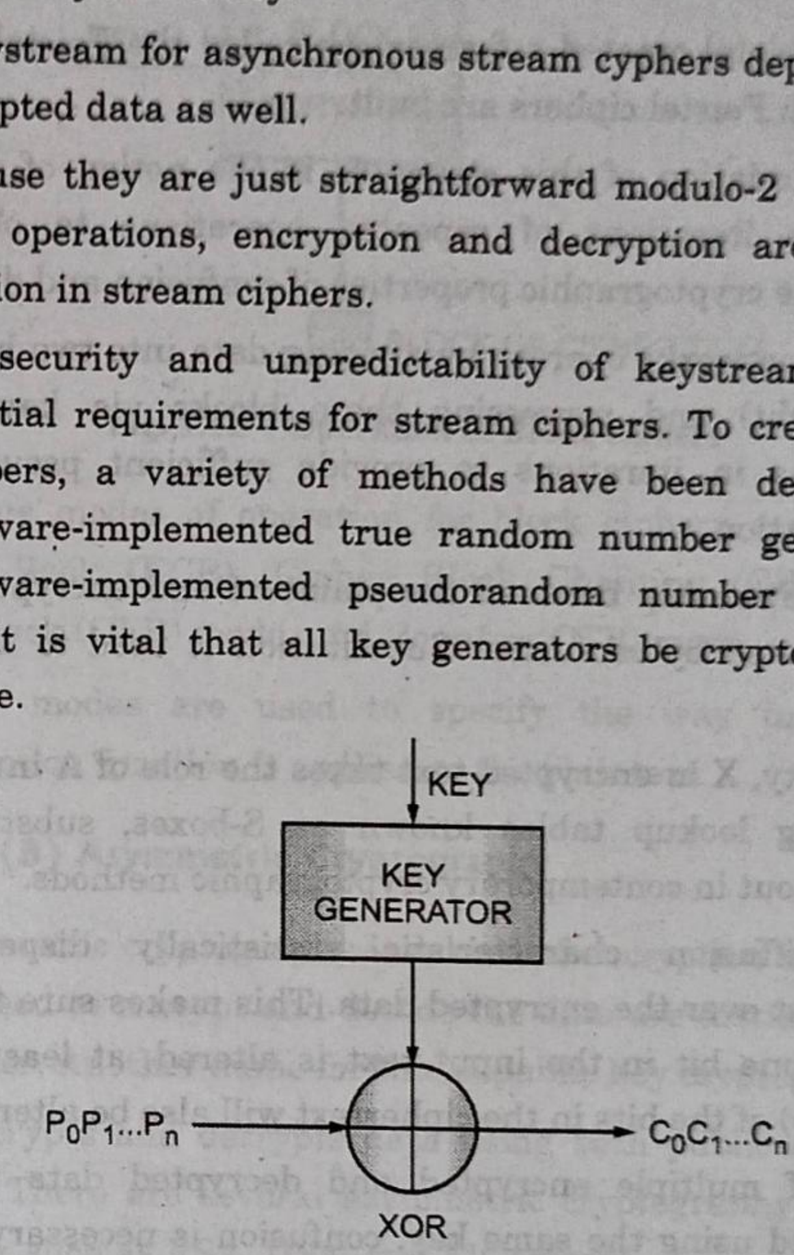

Fig. 1.1.4 : Operation of Stream ciphers

1.1.3(A)(ii) Block Ciphers

- Block cyphers are encryption algorithms that apply the encryption block-by-block after dividing the plaintext into blocks of a predetermined size.

- The design method known as a Feistel cypher is typically used to create block ciphers. A Substitution-Permutation Network, which combines substitution and permutation, has been used to create modern block cyphers like AES (Rijndael) (SPN).
- Horst Feistel created a framework called the Feistel network, on which Feistel ciphers are built.
- The foundation of this structure is the notion of combining multiple iterations of repeated operations to obtain the desirable cryptographic properties of confusion and diffusion.
- Feistel networks operate by dividing data into two blocks (left and right) and processing these blocks via keyed *round functions* in iterations to provide sufficient pseudorandom permutation.
- The relationship between the plaintext and encrypted text is complicated by confusion. Substitution is used to accomplish this.
- In reality, X in encrypted text takes the role of A in plaintext. Utilizing lookup tables known as S-boxes, substitution is carried out in contemporary cryptographic methods.
- The diffusion characteristic statistically disperses the plaintext over the encrypted data. This makes sure that, even if just one bit in the input text is altered, at least half (on average) of the bits in the ciphertext will also be altered.
- Even if multiple encrypted and decrypted data pairs are produced using the same key, confusion is necessary to make locating the encryption key highly challenging.
- In reality, this is accomplished by permutation or transposition.
- 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 :

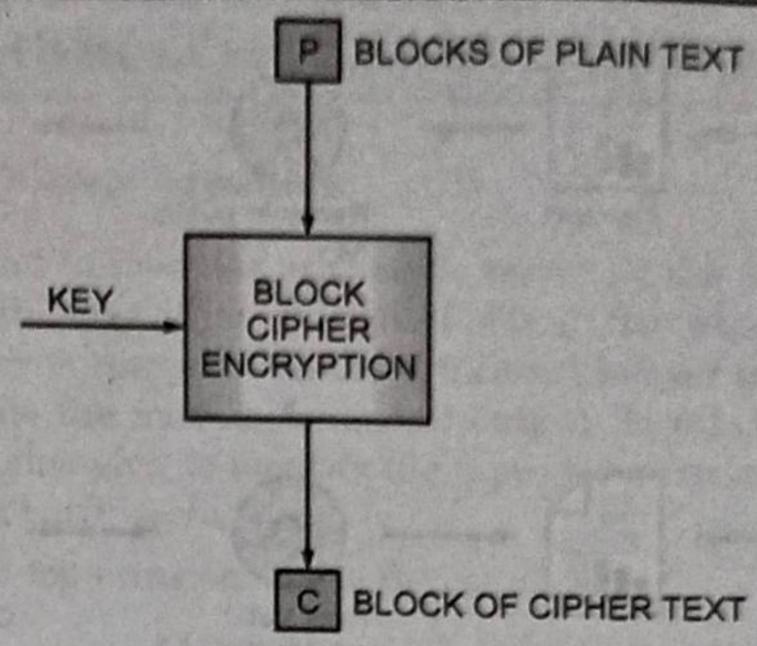

Fig. 1.1.5 : Operation of Block ciphers

- Various modes of operation for block ciphers are Electronic Code Book (ECB), Cipher Block Chaining (CBC), Output Feedback (OFB) mode, and Counter (CTR) mode.
- These modes are used to specify the way in which an encryption function is applied to the plaintext.

1.1.3(B) Asymmetric Cryptography

- Asymmetric cryptography is a kind of encryption in which the key used to encrypt and decode the data are distinct from one another. Another name for this is public key cryptography.
- It encrypts and decrypts data using both public and private keys. There are several asymmetric cryptography algorithms in use, such as RSA, DSA, and ElGammal.
- In the Fig. 1.1.6, public key cryptography is depicted in general terms :

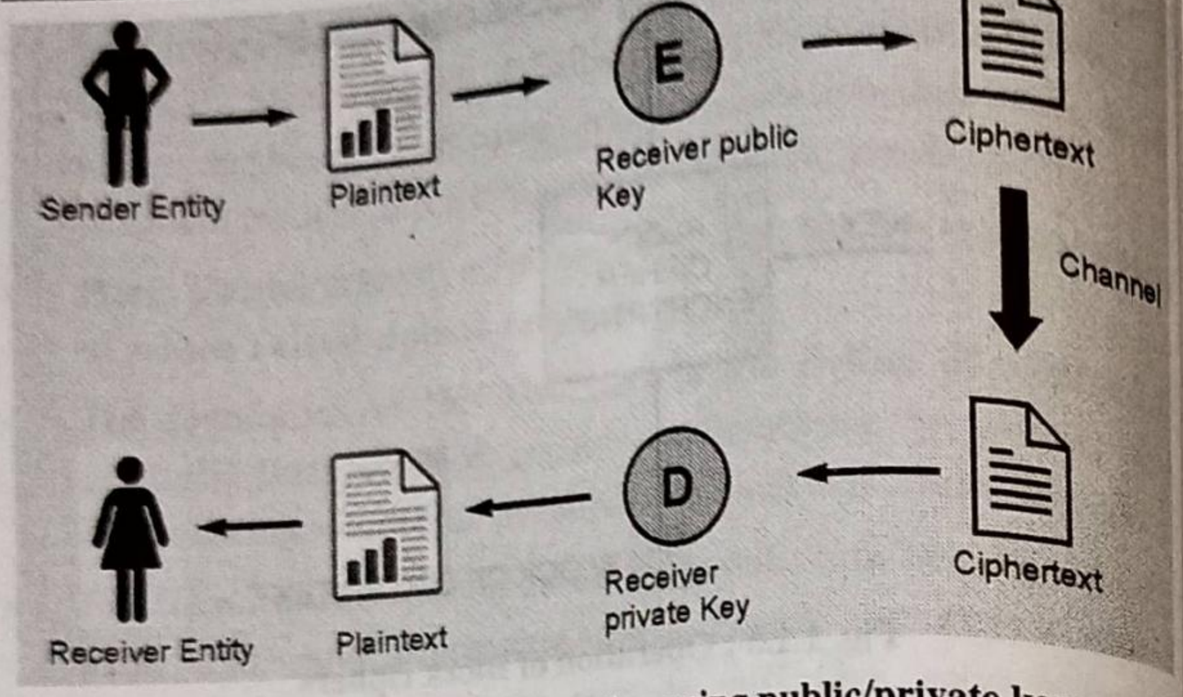

Fig. 1.1.6 : Encryption/decryption using public/private keys

- The Fig. 1.1.6 shows how a sender can encrypt plaintext using the recipient's public key and encryption function E to create ciphertext, which is subsequently sent over the network to the recipient.
- If the data is fed into function D, which produces plaintext, it may be decrypted using the receiver's private key once it reaches the recipient.
- In this approach, unlike symmetric encryption, where keys must be shared in order to accomplish encryption and decryption, the private key stays on the side of the receiver.
- Public key cryptosystems provide key setup, digital signatures, identity, identification, encryption, and decryption. Public key algorithms are slower in terms of computation than symmetric key algorithms.
- Therefore, they are not commonly used in the encryption of large files or the actual data that requires encryption. They are usually used to exchange keys for symmetric algorithm.
- Once the keys are established securely, symmetric key algorithms can be used to encrypt the data.

1.2 ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

1.2.1 Discrete Logarithm

- A problem in modular arithmetic serves as the foundation for a discrete logarithm system. Finding the exponent of the generator is computationally difficult, however it is simple to determine the modulo function's output. In other words, it is quite challenging to identify the input from the output. It only works in one direction.
- Take the following equation, for instance:

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

- Now, it is quite difficult to identify the exponent of the generator 3 in the prior question, which is the outcome of the preceding equation finding 2 given 9.
- The Diffie-Hellman key exchange and digital signature algorithms both employ this challenging problem.

1.2.2 Elliptic curves

GQ. Write a short note on elliptic curve. **(4 Marks)**

- The collection of points that fulfil a specific mathematical equation is known as an elliptic curve. Because the curve is non-singular, it lacks cusps and self-intersections. It includes the two variables a and b as well as the infinite point. An elliptic curve's equation seems kind of like this :

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

- In this case, the numbers a and b's values are components of the field that the elliptic curve is defined on. Over real numbers, rational numbers, complex numbers, or finite fields, elliptic curves can be defined.
- An elliptic curve over prime finite fields is used in place of real numbers for cryptography reasons. The prime should also be bigger than 3. a and/or b's values can be changed to produce various curves.
- The most prominently used cryptosystems based on elliptic

curves are the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Elliptic Curve Diffie-Hellman (ECDH) key exchange.

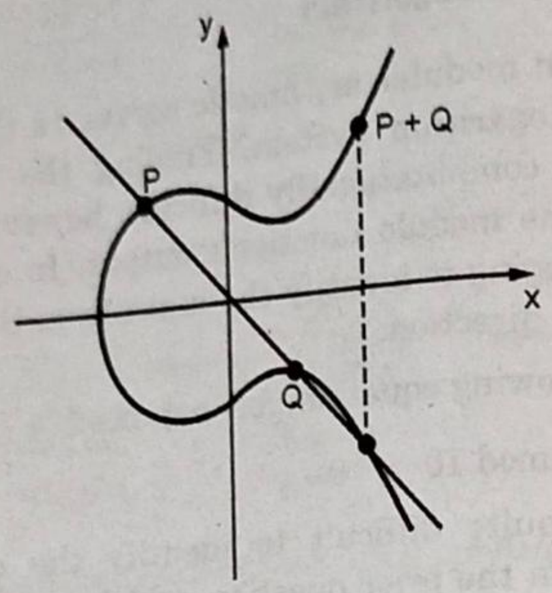

Fig. 1.2.1 : Elliptic curve

- There are other representations of elliptic curves, but technically an elliptic curve is the set points satisfying an equation in two variables with degree two in one of the variables and three in the other.
- An elliptic curve is not just a pretty picture, it also has some properties that make it a good setting for cryptography.

1.3 PUBLIC AND PRIVATE KEYS

GQ. Explore the concept of public and private keys. **(4 Marks)**

- To understand public key cryptography, the key concept that needs to be explored is the concept of public and private keys.
- As the name implies, a private key is a number that has been produced randomly and is retained secretly and privately by its users. Since this is the key used to decode communications, private keys must be kept secure and no unauthorized access should be permitted; otherwise, the entire concept of public key cryptography is put in jeopardy.
- Depending on the kind and class of algorithms being used, private keys might have different lengths.

- For instance, RSA commonly uses keys of 1024 or 2048 bits.
- A minimum key size of 2048 bits is advised rather than the outdated 1024-bit key size, which is no longer regarded as safe.
- The owner of the private key publishes and makes a public key publicly accessible.
- The message can then be encrypted with the published public key and sent to the owner of the private key by anybody who wants to send the publisher of the public key an encrypted message.
- Because the accompanying private key is securely held by the intended receiver, no one else can decrypt the communication.
- The recipient can use the private key to decrypt the communication after receiving it once it has been public key encrypted. However, there are certain issues with public keys.
- These include the veracity of the public keys and the identification of their publisher.
- In the following section, we will introduce two examples of asymmetric key cryptography: RSA and ECC. RSA is the first implementation of public key cryptography whereas ECC is used extensively in blockchain technology.

1.4 RSA

GQ. RSA is not the best cryptographic method for the near future. Why ? **(4 Marks)**

GQ. Describe the steps to generate public and private keys in RSA. **(4 Marks)**

GQ. How encryption and decryption is carried out in RSA? **(4 Marks)**

- RSA was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adelman, hence the name Rivest–Shamir-Adleman (RSA).
- This type of public key cryptography is based on the integer factorization problem, where 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.

The crux of the work involved with the RSA algorithm is during the key generation process. An RSA key pair is generated by performing the following steps :

(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:

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

- Usually, the extended Euclidean algorithm is used to calculate **d.** This algorithm takes **p, q,** and **e** and calculates **d**.
- The key idea in this scheme is that anyone who knows **p** and **q** can easily calculate private key **d** by applying the extended Euclidean algorithm.
- However, someone who does not know the value of **p** and **q** cannot generate **d.** This also implies that **p** and **q** should be large enough for the modulus **n** to become extremely difficult (computationally impractical) to factor.

1.4.1 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***.
- As the size of the numbers being factored increases, these algorithms become more effective. As the quantity (i.e., the key's bit length) grows greater, the difference in complexity between multiplying large numbers and factoring large numbers is less.
- The size of the keys must expand even more quickly as the amount of computing power available to decipher numbers rises. For portable, low-power devices with constrained processing capability, this is an unsustainable position. Factoring and multiplication are separated by an insurmountable distance.
- All of this simply indicates that RSA is not the best cryptographic method for the near future.
- In an ideal Trapdoor Function, the easy way and the hard way get harder at the same rate with respect to the size of the numbers in question.

1.5 ELLIPTIC CURVE CRYPTOGRAPHY

GQ. Write a short note on Elliptic Curve Cryptography. **(4 Marks)**
GQ. What are the advantages of ECC over other public key algorithms? **(2 Marks)**
GQ. Explain the concept of point addition in ECC. **(4 Marks)**
GQ. Explain the concept of point doubling in ECC. **(4 Marks)**
GQ. How to determine public and private key in ECC? **(4 Marks)**

- After the introduction of RSA and Diffie-Hellman, researchers explored other mathematics-based cryptographic solutions looking for other algorithms beyond factoring that would serve as good Trapdoor Functions.
- In 1985, cryptographic algorithms were proposed based on an esoteric branch of mathematics called elliptic curves.
- Elliptic Curve Cryptography (ECC) is based on the discrete logarithm problem founded upon elliptic curves over finite fields (Galois fields).
- ECC has the advantage of requiring a smaller key size while yet offering the same level of security as, say, RSA, as compared to other public key algorithms. ECDSA for digital signatures and ECDH for key exchange are two well-known ECC-derived protocols.
- ECC implements encryption, signatures, and key exchange, which are the three main asymmetric cryptosystem features.
- Although ECC may be used for encryption, it is not frequently done in reality. Instead, it's frequently utilised for key exchange and digital signatures.
- As ECC needs less space to operate, it is becoming very popular on embedded platforms and in systems where storage resources are limited. By comparison, the same level of security can be achieved with ECC only using 256-bit operands as compared to 3072-bits in RSA.
- The following table shows that ECC is able to provide the same level of cryptographic strength as an RSA based system with smaller key sizes :

Table 1.5.1

RSA key sizes (bits)	Elliptic curve key sizes (bits)
1024	160
2048	224
3072	256
7680	384
15360	521

- The ECC uses integer private keys that fall inside the field size range of the curve, which is generally 256 bits.
- The following is an example of a 256-bit ECC private key in hexadecimal format :

  0x51897b64e85c3f714bba707e867914295a1377a7463a9dae8ea6a8b914246319.
- ECC cryptography is incredibly quick since the key creation is as easy as safely producing a random integer within a given range. An ECC private key that falls inside the range is valid.
- The public keys in the ECC are what are known as EC points, which are pairs of x, y-coordinates. EC points may be reduced to only one coordinate plus one bit because of their unique characteristics (odd or even).
- Thus the **compressed public key**, corresponding to a 256-bit ECC private key, is a **257-bit** integer.
- Example of ECC public key (corresponding to the above private key, encoded in the Ethereum format, as hex with prefix 02 or 03) is :

  0x02f54ba86dc1ccb5bed0224d23f01ed87e4a443c47fc690d7797a13d41d2340e1a.
- In this format the public key actually takes 33 bytes (66 hex digits), which can be optimized to exactly 257 bits.
- Different underlying elliptic curves can be used with ECC cryptographic algorithms. Different curves offer various levels of security (cryptographic strength), performance (speed), and key length, as well as perhaps using various methods.
- In addition to having a name (named curves, for example secp256k1 or Curve25519), a field size (which defines the key length, for example 256 bits), security strength (typically the field size / 2 or less), performance (operations/sec), and many other parameters, ECC curves are widely used in cryptographic libraries and security standards.
- The length of ECC keys is closely related to the underlying curve. The default key length for the ECC private keys in the majority of programmes (including OpenSSL, OpenSSH, and Bitcoin) is 256 bits, however many alternative ECC key sizes are feasible depending on the curve: 233-bit (curve sect233k1), 192-bit (curve secp192r1), and many more.

- Elliptic-curve cryptography (ECC) provides several groups of algorithms, based on the math of the elliptic curves over finite fields :
  (1) ECC digital signature algorithms like ECDSA (for classical curves) and EdDSA (for twisted Edwards curves).
  (2) ECC encryption algorithms and hybrid encryption schemes like the ECIES integrated encryption scheme and EEECC (EC-based ElGamal).
  (3) ECC key agreement algorithms like ECDH, X25519 and FHMQV.
- All these algorithms use a curve behind (like secp256k1, curve25519 or p521) for the calculations and rely of the difficulty of the ECDLP (elliptic curve discrete logarithm problem).
- All these algorithms use public / private key pairs, where the private key is an integer and the public key is a point on the elliptic curve (EC point).

1.5.1 Mathematics Behind ECC

GQ. Explain the concept of point addition in ECC. (4 Marks)
GQ. Explain the concept of point doubling in ECC. (4 Marks)
GQ. How to determine public and private key in ECC? (4 Marks)

- A fundamental introduction to the underlying mathematics is required in order to understand ECC. An elliptic curve is basically a type of polynomial equation known as the Weierstrass equation, which generates a curve over a finite field.
- The field where all arithmetic operations are carried out modulo a prime p is the most often used field. Elliptic curve groups are made up of curve points over a finite field.
- An elliptic curve is defined in the following equation:

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

- Here, 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.
- Furthermore, a condition also needs to be met that ensures that the equation mentioned earlier has no repeated roots. This means that the curve is non-singular.
- The condition is described in the following equation, which is a standard requirement that needs to be met. More precisely, this ensures that the curve is non-singular:

$$4a^3 = 27b^2 \neq 0 \bmod P$$

- 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 are point addition and point doubling. Point addition is a process where two different points are added, and point doubling means that the same point is added to itself.

(A) Point addition

- Point addition is shown in the following diagram. This is a geometric representation of point addition on elliptic curves.
- In this method, 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.
- This point is mirrored as P+Q, which represents the result of the addition as R.
- This is shown as P+Q in the following diagram:

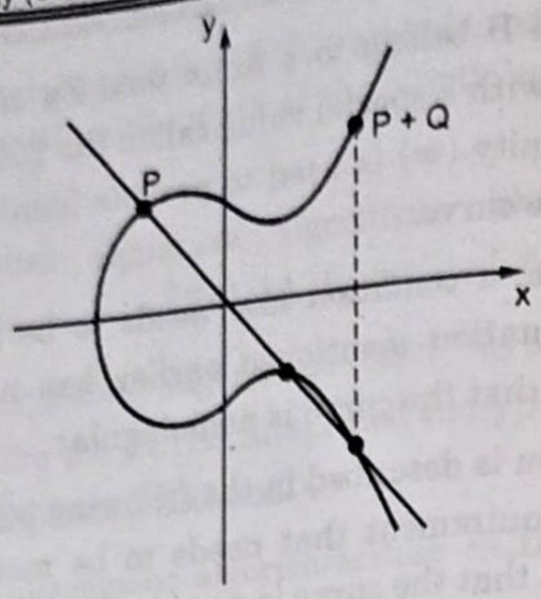

Fig. 1.5.1

- The group operation denoted by the + sign for addition yields the following equation:

$$P + Q = R$$

- In this case, two points are added to compute the coordinates of the third point on the curve:

$$P + Q = R$$

- More precisely, this means that coordinates are added, as shown in the following equation:

$$(x1, y1) + (x2, y2) = (x3, y3)$$

- The equation of point addition is as follows:

$$x3 = s2 - x1 - x2 \bmod p$$

$$y3 = s(x1 - x3) - y1 \bmod p$$

- Here, we see the result of the preceding equation:

$$S = ((y2-y1)/(x2-x1)) \bmod p$$

S in the preceding equation depicts the line going through P and Q.

- An example of point addition is shown in the following diagram. It was produced using Certicom's online calculator. This example shows the addition and solutions for the equation over finite field F_{23}.

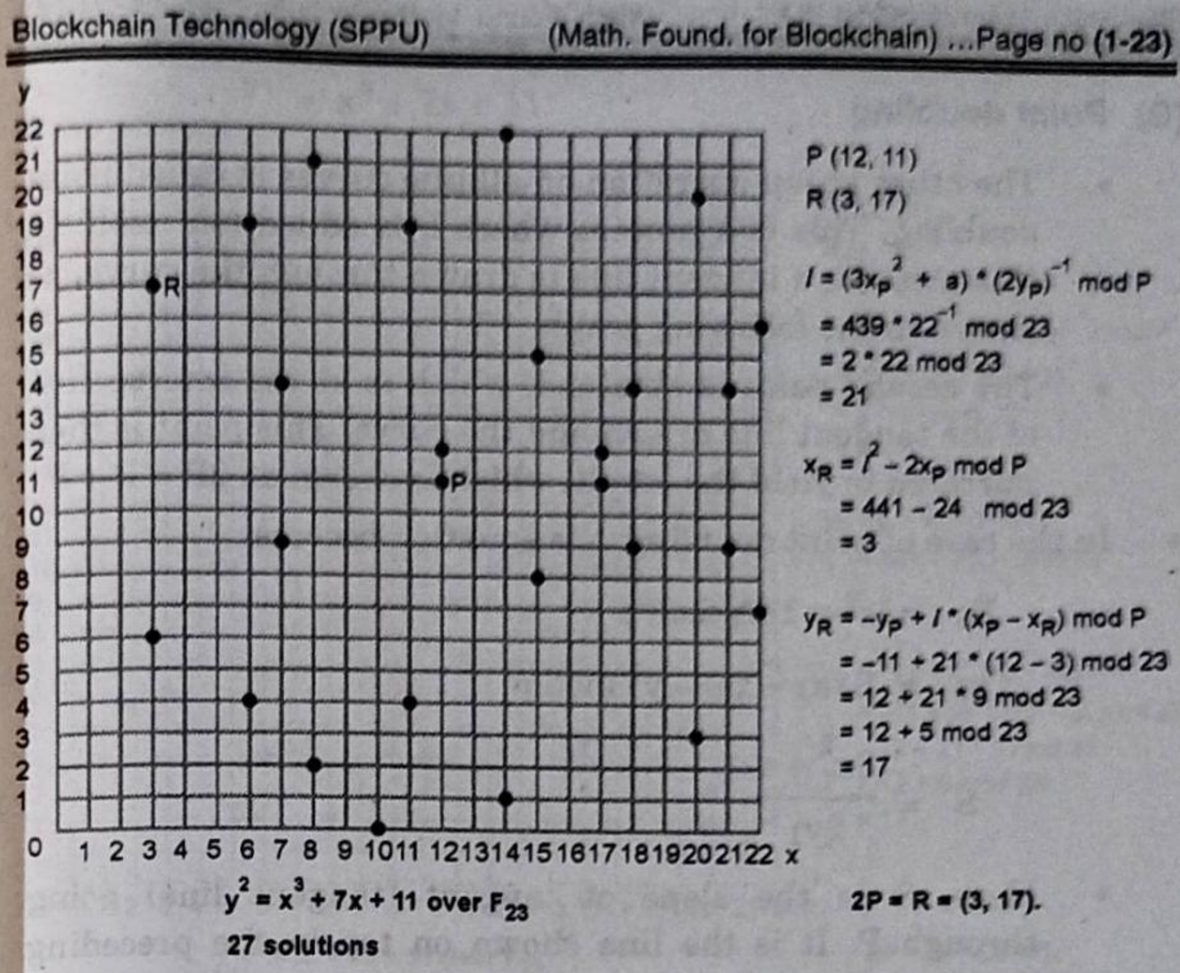

Fig. 1.5.2 : Example of point addition

- In the preceding example, the graph on the left side shows the points that satisfy this equation:
- There are 27 solutions to the equation shown earlier over finite field *F23*. *P* and *Q* are chosen to be added to produce point *R*. Calculations are shown on the right side, which calculates the third point *R*. Note that here, *l* is used to depict the line going through *P* and *Q*.
- As an example, to show how the equation is satisfied by the points shown in the graph, a point (x, y) is picked up where $x = 3$ and $y = 6$.
- Using these values shows that the equation is indeed satisfied:

$$y^2 \bmod 23 = x^3 + 7x + 11 \bmod 23$$

$$6^2 \bmod 23 = 3^3 + 7(3) + 11 \bmod 23$$

$$36 \bmod 23 = 59 \bmod 23$$

$$13 = 13$$

(B) Point doubling

- The other group operation on elliptic curves is called point doubling. This is a process where P is added ton itself. In this method, a tangent line is drawn through the curve, as shown in the following graph.
- The second point is obtained, which is at the intersection of the tangent line drawn and the curve. This point is then mirrored to yield the result, which is shown as $2P = P + P$.
- In the case of point doubling, the equation becomes:

$$X_3 = s^2 \cdot x\ 1 - x_2 \bmod p$$

$$y_3 = S(x_1 - x_3) - y_1 \bmod p$$

$$S = \frac{3x_1^2 + a}{2y_1}$$

- Here, S is the slope of tangent (tangent line) going through P. It is the line shown on top in the preceding diagram. In the preceding example, the curve is plotted over real numbers as a simple example, and no solution to the equation is shown.

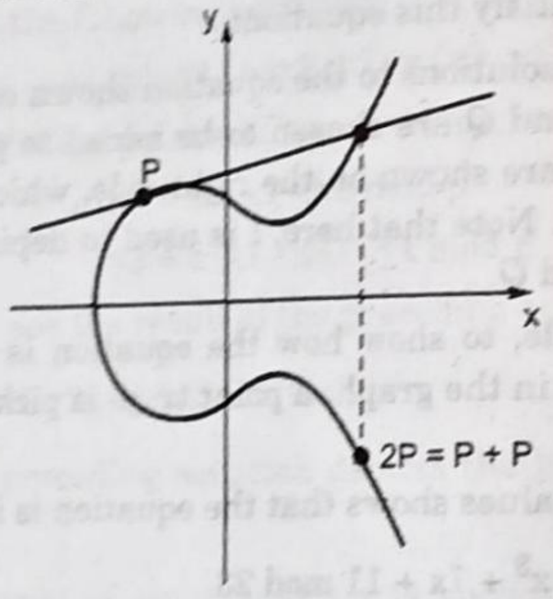

Fig. 1.5.3

- The following example shows the solutions and point doubling of elliptic curves over finite field F23. The graph on the left side shows the points that satisfy the equation:

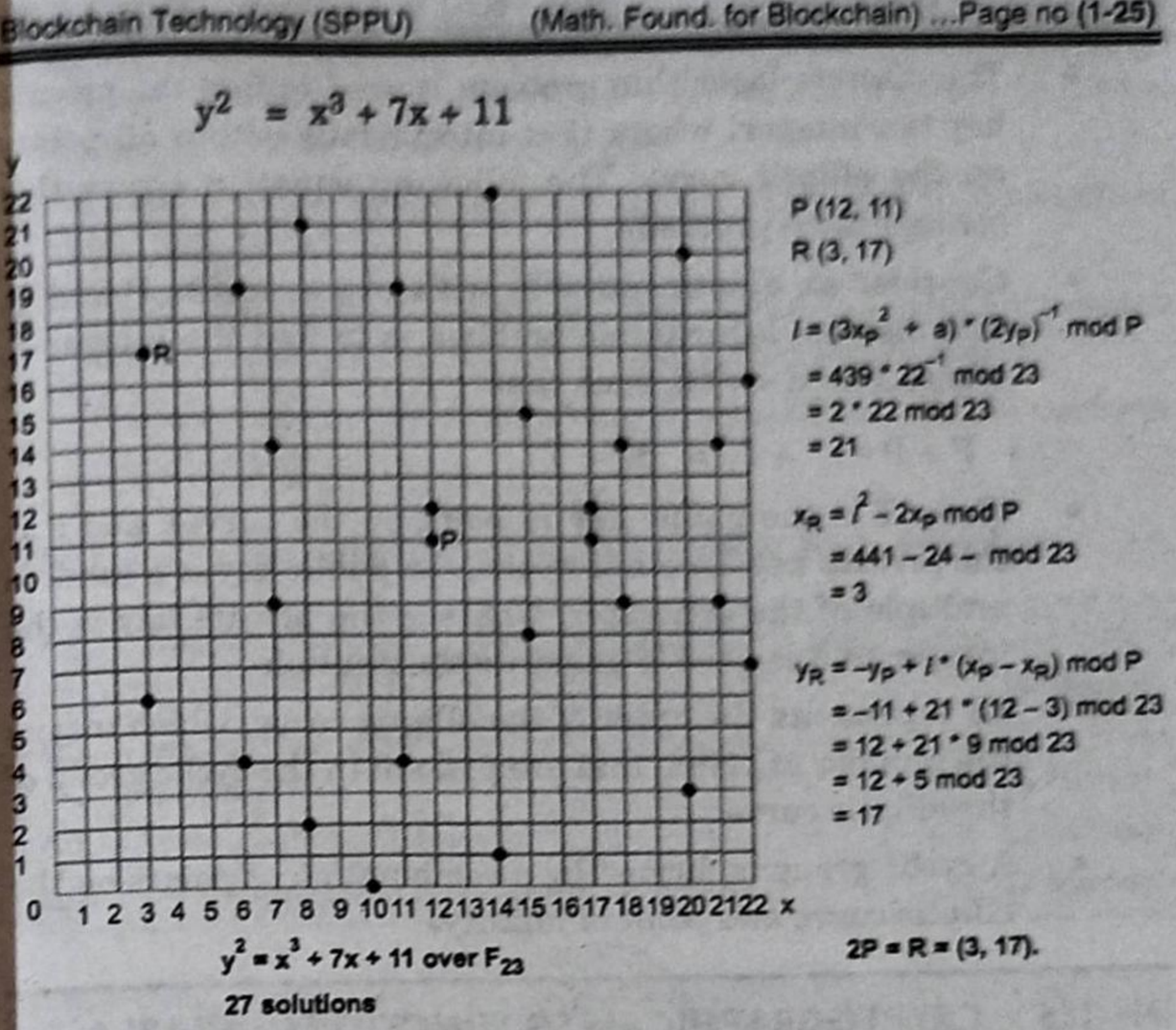

Fig. 1.5.4

- As shown on the right side in the preceding graph, the calculation that finds the *R* after *P* is added into itself (point doubling).
- There is no *Q* as shown here, and the same point *P* is used for doubling. Note that in the calculation, *l* is used to depict the tangent line going through *P*.

(C) Discrete logarithm problem in ECC

- The discrete logarithm problem in ECC is based on the idea that, under certain conditions, all points on an elliptic curve form a cyclic group.
- On an elliptic curve, the public key is a random multiple of the generator point, whereas the private key is a randomly chosen integer used to generate the multiple.
- In other words, a private key is a randomly selected integer, whereas the public key is a point on the curve.

- The discrete logarithm problem is used to find the private key (an integer) where that integer falls within all points on the elliptic curve. The following equation shows this concept more precisely.
- Consider an elliptic curve E, with two elements P and T. The discrete logarithmic problem is to find the integer d, where $1 <= d <= \#E$, such that:

$$P + P + \ldots + P = dP = T$$

- Here, T is the public key (a point on the curve), and d is the private key. In other words, the public key is a random multiple of the generator, whereas the private key is the integer that is used to generate the multiple.
- #E represents the order of the elliptic curve, which means the number of points that are present in the cyclic group of the elliptic curve.
- A cyclic group is formed by a combination of points on the elliptic curve and point of infinity.

1.6 CRYPTOGRAPHIC HASH FUNCTIONS : SHA256

1.6.1 Hash Functions

GQ. State and explain hash function? (2 Marks)

- Hash functions are used to convert infinitely lengthy input texts into digests of a fixed length. Hash functions offer the data integrity service and are keyless. Dedicated and iterated hash function creation strategies are often used to create them.
- There are several different families of hash functions, including MD, SHA-1, SHA-2, SHA-3, RIPEMD, and Whirlpool. Digital signatures and Message Authentication Codes (MACs), like HMACs, frequently employ hash functions.
- Additionally, hash functions are frequently utilised to offer data integrity services. These may be used to create additional cryptographic primitives like MACs and digital signatures as well as one-way functions.

- Hash functions are sometimes used in applications to create pseudo-random numbers (PRNGs).
- Cryptographic hash functions are one of the key components of blockchain.
- These are part of building block functions which provides security, privacy and consensus on blockchain platform.
- These functions are mathematical algorithms used to perform required conversion.

1.6.2 Characteristics of Cryptographic Hash Functions

GQ. Enlist the features of cryptographic hash functions? Explain any two. (4 Marks)

GQ. Demonstrate pre image and second pre image resistance with example. (4 Marks)

GQ. What is collision in hash? (2 Marks)

GQ. Explain Avalanche effect of hash functions. (2 Marks)

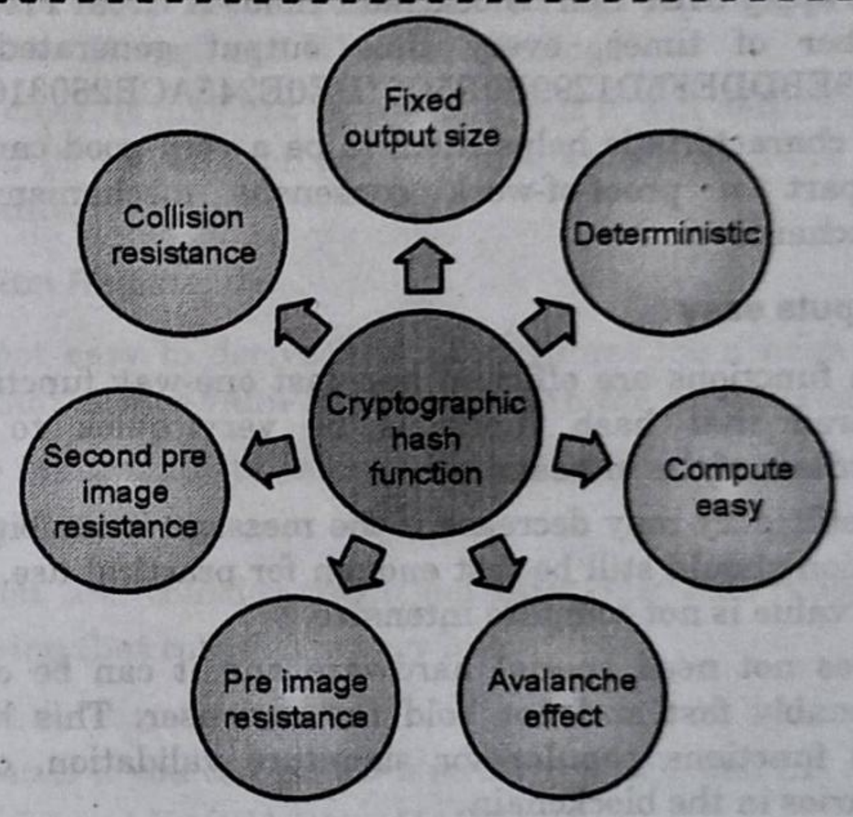

Fig. 1.6.1 : Characteristics of Cryptographic Hash Functions

Fixed output size

- Usually, if the size of input changes then size of output

changes. E.g Compression

- However, in case of cryptographic has functions, output string size remains same irrespective of input string size.

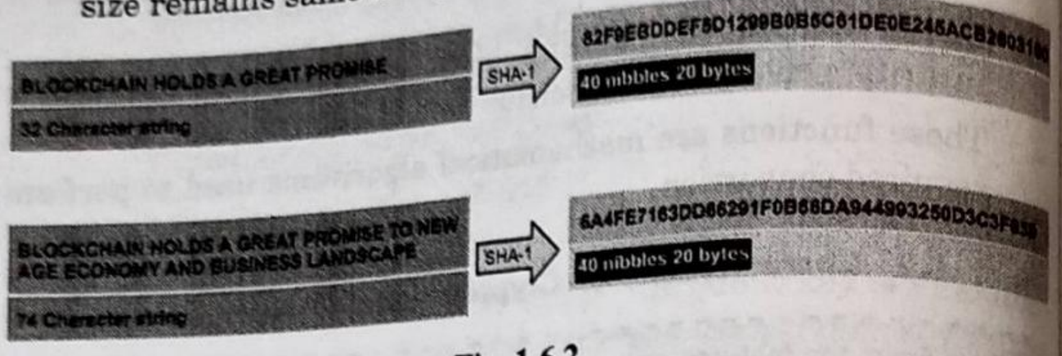

Fig. 1.6.2

☞ Deterministic

- If the input is same, the output will always be same.
- If the function is applied on the same input any number of times, the resultant answer will always be same.
- If we apply SHA-1 on "Blockchain Holds A Great Promise" any number of times, every time output generated will be "82F9EBDDEF5D1299B0B5C61DE0E245ACB2603160".
- This characteristic helps them to be a very good candidate to be part of proof-of-work consensus mechanism in the blockchain.

☞ Compute easy

- Hash functions are efficient and fast one-way functions. It is required that hash functions be very quick to compute regardless of the message size.
- The efficiency may decrease if the message is too big, but the function should still be fast enough for practical use. Creating hash value is not compute intensive.
- It does not need special hardware and it can be completed reasonably fast and not hold the end user. This has made these functions popular for signature validation, consensus scenarios in the blockchain.

☞ Pre-image resistance

- It is not computationally easy to derive input for a given output.

Say, H1 = hash(string 1)

- If H1 is given, it is not computationally easy or practical to find string1.
- It is possible but just not easy.
- The stronger the algorithm, the more computation it may require and it may have better pre-image resistance.
- Next, even if two characteristics are similar to pre image resistance, they are not the same.
- It is not easy to find source based on data, making these a kind of one way function. This is also called a one-way property.

☞ Second pre-image resistance

- It is possible to find another input that can give the same hash value as a given input.

Let's take String1 and H1 = hash (String1).

- It is not easy to find another string (String2) such that String1! = String2 and H1 = hash(String2).
- To be clear, it may not be possible, but it will definitely not be easy to do so. This property is also known as weak collision resistance.

☞ Collision Resistance

- It is not easy to derive two input values for a hash function such that output value is same for both the input values.
- If two input maps to same output, then the function is said to have collision.
- It is not that collision might not exist, just that probability of occurring that collision is very less.
- So, hash algorithm,HASH1 is said to not have collision resistance if you can find two strings, say S1 and S2, such that S1 ! = S2 and HASH1(S1) = HASH1(S2).
- It is different than second pre-image resistance as here only hash function is given while input strings can be anything.

☞ **Avalanche Effect**

A small change in input causes a large change in output.

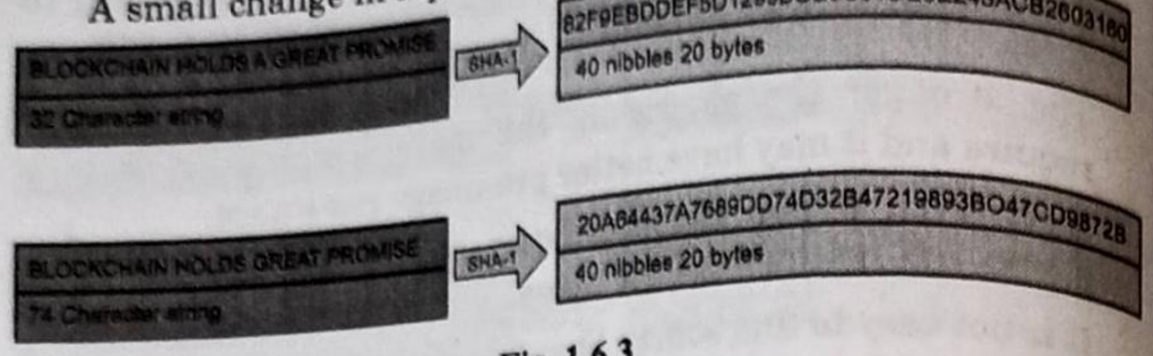

Fig. 1.6.3

1.7 SECURE HASH ALGORITHMS

The following list describes the most common **Secure Hash Algorithms (SHAs)** :

- **SHA-0** : This is a 160-bit function introduced by NIST in 1993.
- **SHA-1** : SHA-1 was introduced in 1995 by NIST as a replacement for SHA-0. This is also a 160-bit hash function. SHA-1 is used commonly in SSL and TLS implementations. It should be noted that SHA-1 is now considered insecure, and it is being deprecated by certificate authorities. Its usage is discouraged in any new implementations.
- **SHA-2** : This category includes four functions defined by the number of bits of the hash: SHA-224, SHA-256, SHA-384, and SHA-512.
- **SHA-3** : This is the latest family of SHA functions. SHA-3-224, SHA-3-256, SHA-3-384, and SHA-3-512 are members of this family. SHA-3 is a NIST-standardized version of Keccak. Keccak uses a new approach called **sponge construction** instead of the commonly used Merkle-Damgard transformation.
- **RIPEMD** : RIPEMD is the acronym for **RACE Integrity Primitives Evaluation Message Digest**. It is based on the design ideas used to build MD4. There are multiple versions of RIPEMD, including 128-bit, 160-bit, 256-bit, and 320-bit.
- **Whirlpool :** This is based on the W Rijndael cipher, a modified

variation of the Rijndael encryption. It makes use of the one-way function known as the Miyaguchi-Preneel compression function, which compresses two fixed-length inputs into a single fixed-length output. It does a single block length compression.

- Numerous real-world uses exist for hash functions, from straightforward file integrity checks and password storage to inclusion in cryptographic protocols and algorithms.
- They are utilised in several applications, including peer-to-peer file sharing, virus fingerprinting, distributed hash tables, bloom filters, and hash tables.
- Hash operations are essential to blockchain. In order to confirm the amount of computing work used by miners, the PoW algorithm employs SHA-256 twice. RIPEMD 160 is used to produce Bitcoin addresses.

1.8 DESIGN OF SHA-256

GQ. Explain the working of SHA-256 algorithm. **(4 Marks)**

- SHA-256 has the input message size < 264-bits. Block size is 512-bits, and it has a word size of 32-bits. The output is a 256-bit digest.
- The compression function processes a 512-bit message block and a 256-bit intermediate hash value. There are two main components of this function: the compression function and a message schedule.
- The algorithm works as follows, in eight steps :

☞ **Pre-processing**

(1) If a block's length is less than the requisite 512 bits, padding of the message is used to increase it to 512 bits.

(2) Parsing the message into blocks of 512 bits, which separates the message and any padding into equal blocks.

(3) Creating the initial hash value, which is made up of the first 32 bits of the fractional portions of the square roots of the first

eight prime integers. The first hash value is composed of eight 32-bit words. These initial parameters are picked at random to start the procedure off, and they give some amount of assurance that the method has no backdoors.

☞ Hash computation

(1) Following that, each message block is analysed one at a time and it takes 64 rounds to compute the whole hash result. To ensure that no two rounds are same, each round uses slightly different constants.

(2) The message schedule has been set.

(3) Eight operational variables are initialised.

(4) A calculation is made for the intermediate hash value.

(5) After the message has been processed, the output hash is generated:

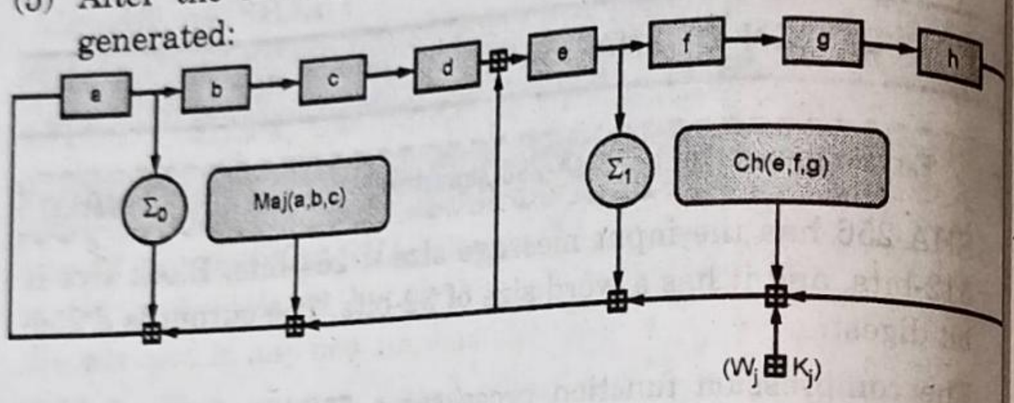

Fig. 1.8.1 : One round of SHA-256 compression function

- In the preceding diagram, a, b, c, d, e, f, g, and h are the registers. Maj and Ch are applied bitwise. Σ0 and Σ1 performs bitwise rotation. Round constants are Wj and Kj, which are added, mod 232.

⏭ 1.9 DIGITAL SIGNATURES

GQ. How the digital signing and verification is carried out with RSA digital signature algorithm? **(6 Marks)**

GQ. Explain the two approaches to using digital signatures with encryption. **(2 Marks)**

- With the use of digital signatures, it is possible to link a communication to the source of its creation. Non-repudiation and data origin authentication are provided via digital signatures.
- Blockchain technology makes use of digital signatures, with senders utilising their private key to sign transactions before publishing them to the network.
- This digital signature demonstrates that they are the legal owners of the item, like bitcoins. To confirm that the money truly belong to the node (user) that claims to be the owner, these transactions are once again validated by other nodes on the network.
- Two processes are used to calculate digital signatures. The high-level phases of an RSA digital signature method are shown in an example below.

☞ Digital signature algorithm using RSA

The RSA digital signature algorithm is as follows :

(1) **Determine the data packet's hash value :** This will ensure data integrity since the hash can be recalculated at the receiver's end and compared with the original hash to determine whether the data has been altered during transit. Although message signing technically is possible without first hashing the contents, this method is not thought to be safe.

(2) **Uses the signer's private key to tamper-proof the hash value :** The legitimacy of the signature and the signed material is guaranteed since only the signer has access to the private key.

- Important characteristics of digital signatures are authenticity, invulnerability, and nonreusability.
- Authenticity refers to the ability of a receiving party to confirm the validity of digital signatures. The unforgeability attribute makes sure that only the message's sender may use the private key signing feature. To put it another way, only the genuine sender can create a signed message.

- The concept of nonreusability states that a digital signatur cannot be taken out of one communication and applied another.
- The following diagram illustrates how a generic digit signature function works :

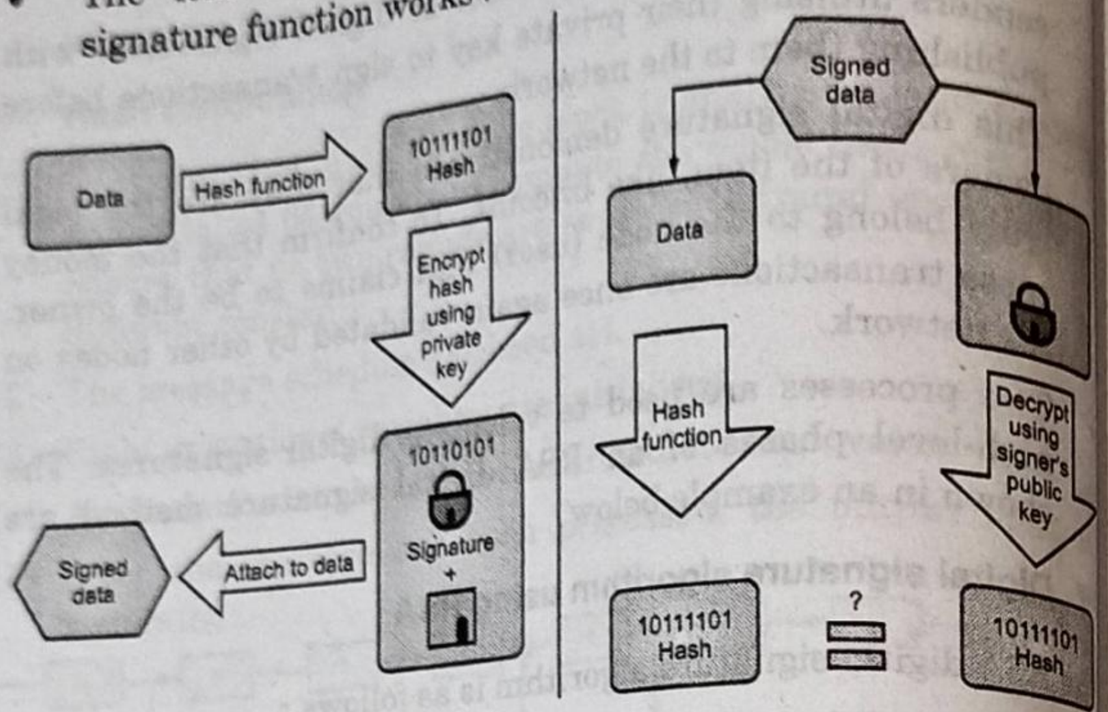

Fig. 1.9.1 : Digital signing (left) and verification process (right) (Example of RSA digital signatures)

- If a sender wants to send an authenticated message to receiver, there are two methods that can be used: sign the encrypt and encrypt then sign.

These two approaches to using digital signatures with encryption are a follows.

Sign then encrypt

- With this approach, the sender digitally signs the data using the private key, appends the signature to the data, and then encrypts the data and the digital signature using the receiver's public key.
- This is considered a more secure scheme as compared to the *encrypt then sign* scheme .

Encrypt then sign

- With this method, the sender encrypts the data using the receiver's public key and then digitally signs the encrypted data.

1.10 MERKEL TREES

GQ. What are Merkle trees ? Explain the structure of a Merkle tree. **(4 Marks)**

- A Merkle tree, named after its creator Ralph Merkle, is a binary tree containing hash pointers. The Merkle tree, commonly referred to as the hash tree, is a kind of data structure used for data synchronisation and verification.
- Each non-leaf node of the tree data structure is a hash of the nodes it contains as children.
- The leaf nodes are all equally deep and as far to the left as they can be.
- It employs hash functions to preserve the integrity of the data.

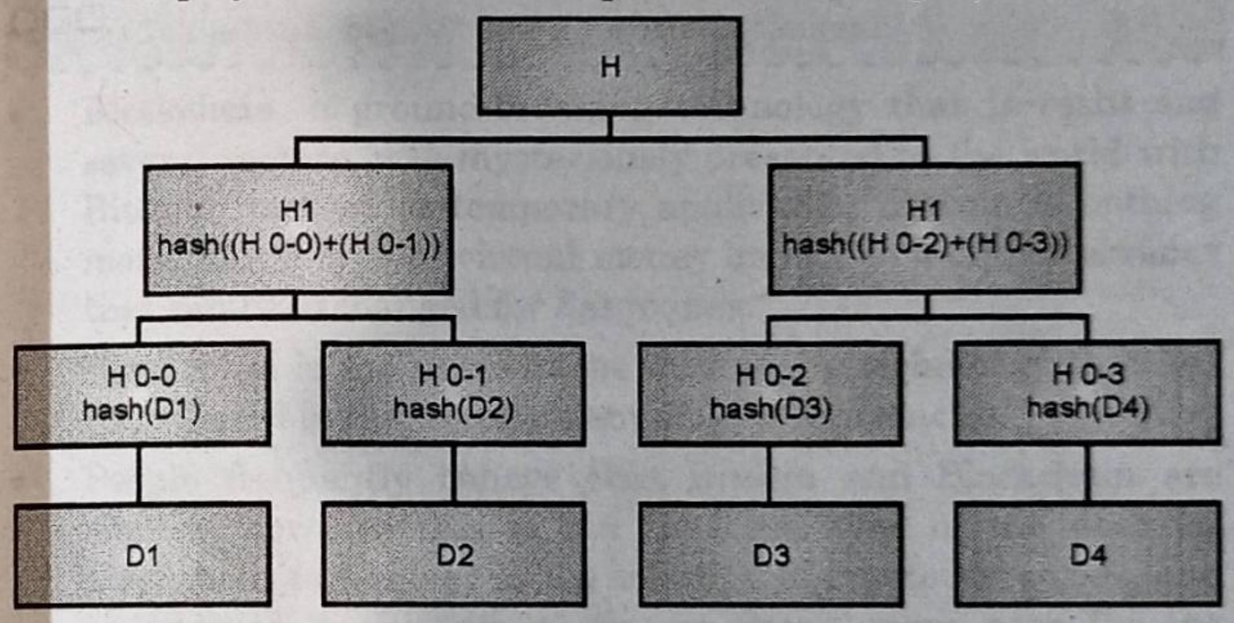

Fig. 1.10.1 : Merkel tree

- An input of data broken up into blocks labeled D1 though D4. Each of these blocks are hashed using some hash function.
- Then each pair of nodes are recursively hashed until we reach the root node, which is a hash of all nodes below it.

- The hash value at root node is called as Merkle root.
- When data is shared among parties, data is shared fro regular channels and Merkle root is shared from secu channel.
- Intermediate hash values can be shared from regular or secu channels as per requirements. As data starts coming in, th verifier can verify if the data is valid by calculating hash valu of the data received and comparing it with the hash valu received for that segment.
- After the entire document is received, merkle root is compare with the merkle root received from the secure source. If data i manipulated in the process, then value of hash and in turn merkle root would change and this will help verifier to chec authenticity of the data.
- In bitcoin and other cryptocurrencies, Merkle trees serve t encode blockchain data more efficiently and securely. They ar also referred to as "binary hash trees."

Chapter Ends..

❑❑❑