



Subject:
(410243)
Blockchain Technology
(BT)

Scheme

Theory: BT

- **Teaching Scheme:** Lectures 3 Hrs/Week
- **Examination Scheme:**
 - In Semester Assessment: 30 Marks
 - End Semester Assessment: 70 Marks

Practical: LP-III Laboratory

- **Teaching Scheme:** Practical: 4 Hrs/Week
- **Examination Scheme:**
 - Practical : 50 Marks
 - Term Work: 50 Marks

Course Objectives:

- Technology behind Blockchain
- Crypto currency, Bitcoin and Smart contracts
- Different consensus algorithms used in Blockchain
- Real-world applications of Blockchain
- To analyze Blockchain Ethereum Platform using Solidity
- To Describe Blockchain Case Studies

Course Outcomes:

On completion of the course, student will be able to -

- **CO1:** Interpret the fundamentals and basic concepts in Blockchain
- **CO2:** Compare the working of different blockchain platforms
- **CO3:** Use Crypto wallet for crypto currency based transactions
- **CO4:** Analyse the importance of blockchain in finding the solution to the real-world problems.
- **CO5:** Illustrate the Ethereum public block chain platform
- **CO6:** Identify relative application where block chain technology can be effectively used and implemented.

Contents of BT

- **Unit I:** Mathematical Foundation for Blockchain: 06 Hrs
- **Unit II:** Feature Engineering: 07 Hrs
- **Unit III:** Blockchain Platforms and Consensus in Blockchain: 06 Hrs
- **Unit IV:** Cryptocurrency – Bitcoin, and Token: 06Hrs
- **Unit V:** Blockchain Ethereum Platform using Solidity: 06Hrs
- **Unit VI:** Blockchain Case Studies: 06Hrs

Books

Text Books:

1. Martin Quest, —Blockchain Dynamics: A Quick Beginner's Guide on Understanding the Foundations of Bit coin and Other Crypto currencies, Create Space Independent Publishing Platform, 15-May-2018
2. Imran Bashir, —Mastering Blockchain: Distributed Ledger Technology, Decentralization and Smart Contracts Explained, Second Edition, Packt Publishing, 2018
3. Alex Leverington, —Ethereum Programming, Packt Publishing, 2017

Reference Books:

1. Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, "Beginning Blockchain A Beginner's Guide to Building Blockchain Solutions", 2018
2. Chris Dannen, "Introducing Ethereum and Solidity", Foundations of Crypto currency and Blockchain Programming for Beginners
3. Daniel Drescher, "Blockchain Basics", A Non -Technical Introduction in 25Steps.
4. Ritesh Modi, "Solidity Programming Essentials", Packt Publishing, 2018
5. Chandramouli Subramanian, Asha A George, Abhilash K A and Meena Karthikeyan, "Blockchain Technology", Universities Press, ISBN-9789389211634

Books

E-Books:

1. https://users.cs.fiu.edu/~prabakar/cen5079/Common/textbooks/Mastering_Blockchain_2nd_Edition.pdf
2. https://www.lopp.net/pdf/princeton_bitcoin_book.pdf
3. <https://www.blockchainexpert.uk/book/blockchain-book.pdf>

MOOC Courses Links:

1. NPTEL Course on —Introduction to Blockchain Technology & Applications||
<https://nptel.ac.in/courses/106/104/106104220/>
2. NPTEL Course on blockchain
<https://nptel.ac.in/courses/106/105/106105184/>



Unit I

Mathematical Foundation for Blockchain

Content

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

Case Study

- Compare the Symmetric and Asymmetric Cryptography algorithms

Course Outcome Mapped

- CO1

Introduction

- The Internet or the global Internet is the internationally connected network of computers with addresses that are administrated by IANA (Internet address and Naming Authority).
- There are many aspects to provide security for many applications, ranging from secure commerce and payments to private communications and protecting passwords. One essential aspect for secure communications is that of cryptography

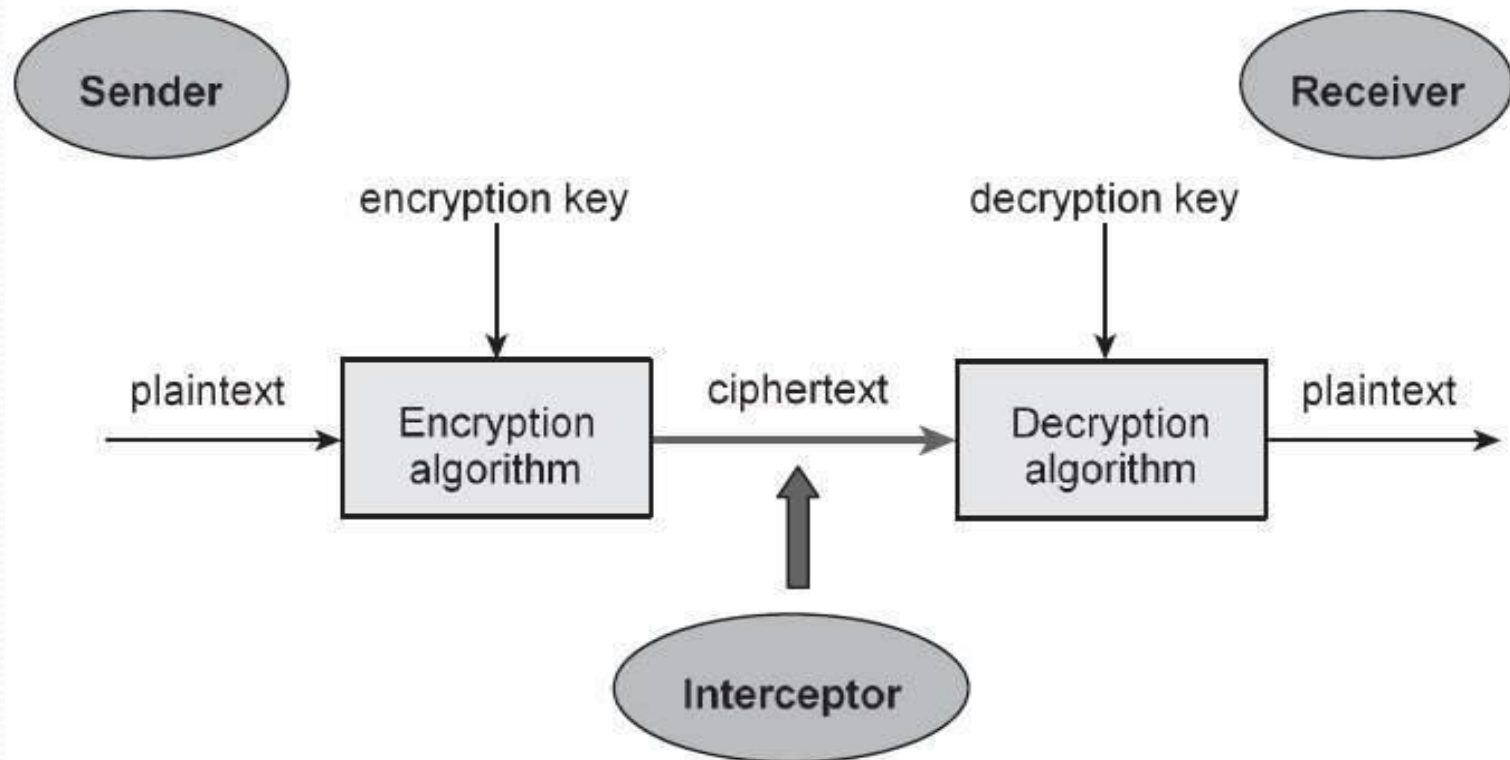
Cryptography

- Cryptography derived its name from a Greek word called “krypto’s” which means “Hidden Secrets”.
- Cryptography is the practice and study of hiding information. It is the Art or Science of converting a plain unintelligible data into an intelligible data and again retransforming that message into its original form.
- It provides Confidentiality, Integrity, and Accuracy

Purpose of Cryptography

- **Authentication:** The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.)
 - Entity authentication
 - Data origin authentication
- **Privacy/confidentiality:** Ensuring that no one can read the message except the intended receiver.
- **Integrity:** Assuring the receiver that the received message has not been altered in any way from the original.
- **Non-repudiation:** A mechanism to prove that the sender really sent this message.

Architecture of Cryptography



Types of Cryptography

Secret Key Cryptography

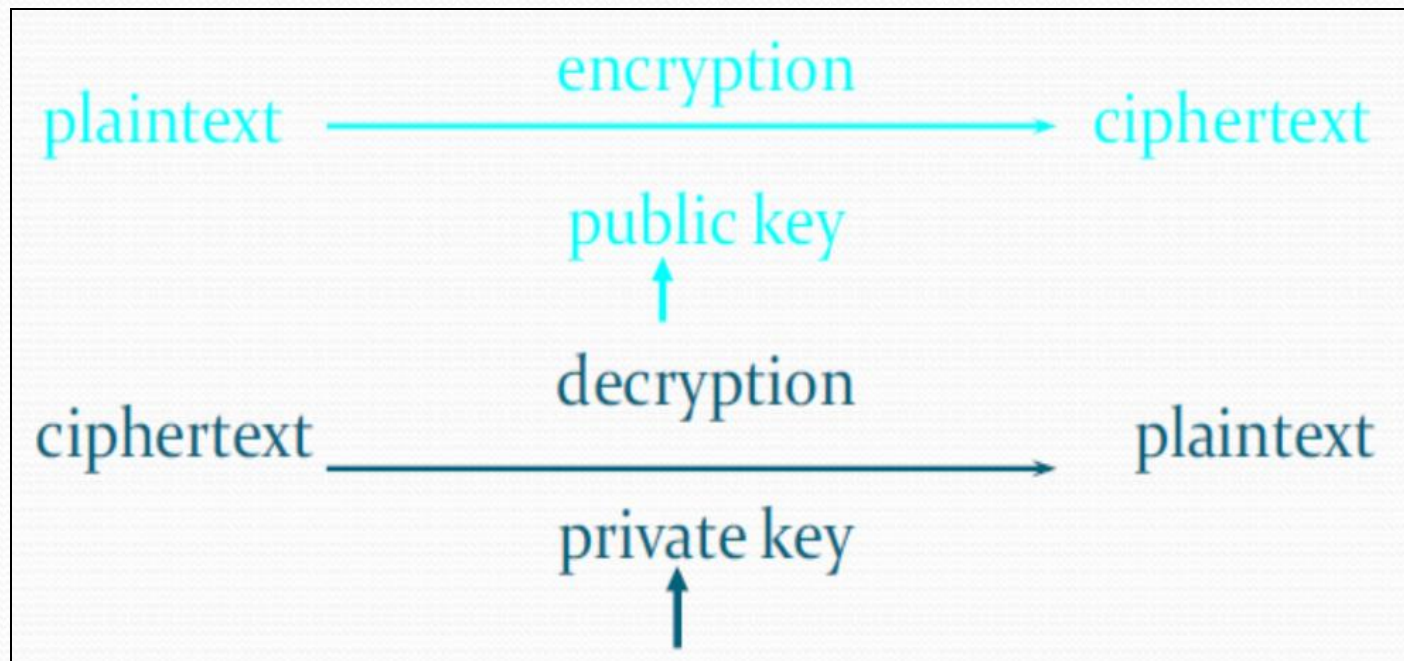
- Single key used to encrypt and decrypt.
- Key must be known by both parties.

Public Key Cryptography

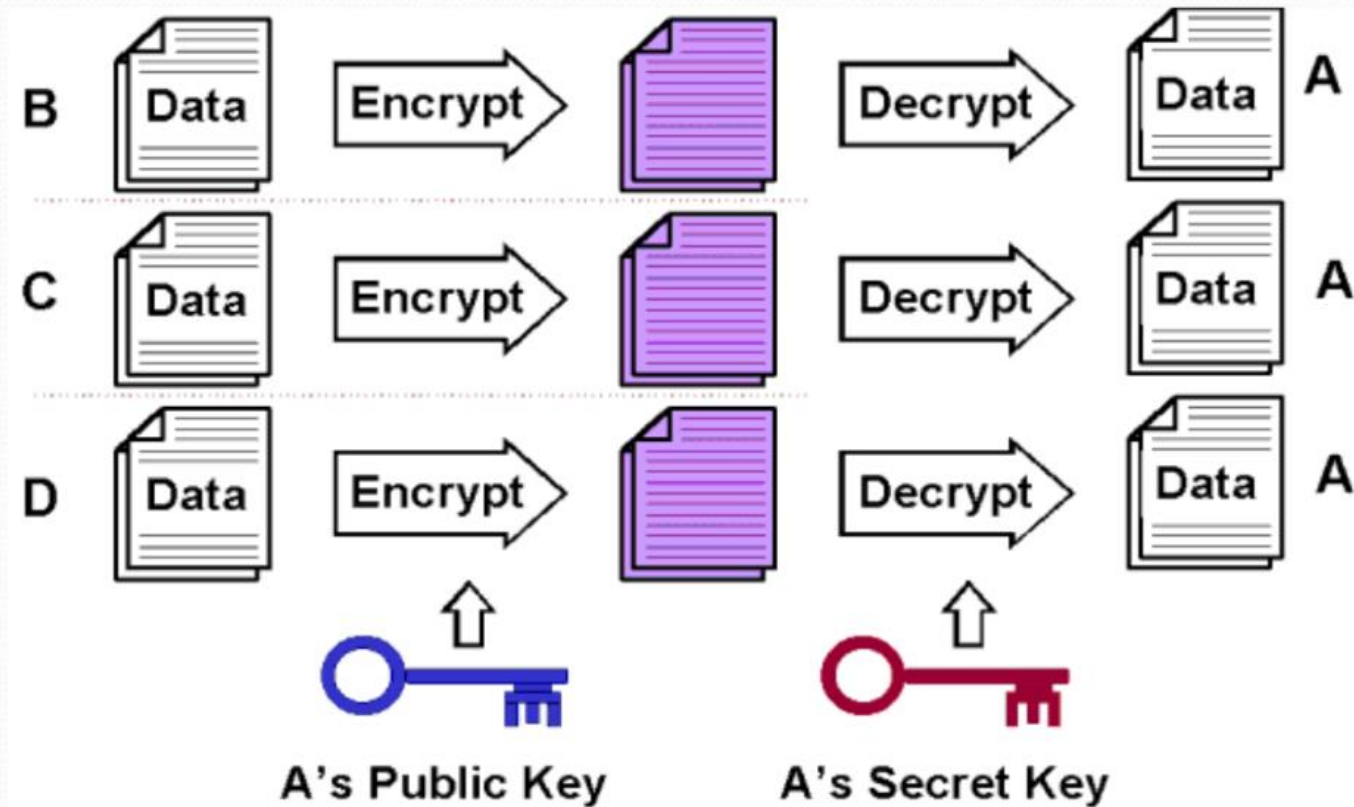
- One of the keys allocated to each person is called the "public key", and is published in an open directory somewhere where anyone can easily look it up, for example by email address.
- Each entity has 2 keys:
 - Private Key (a secret)
 - Public key (well known)

Use of key in Cryptography

- Private keys are used for decrypting.
- Public keys are used for encrypting



Process of Cryptography



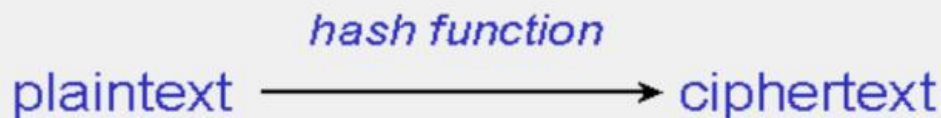
Types of Cryptography Algorithms



A) Secret key (symmetric) cryptography. SKC uses a single key for both encryption and decryption.



B) Public key (asymmetric) cryptography. PKC uses two keys, one for encryption and the other for decryption.



C) Hash function (one-way cryptography). Hash functions have no key since the plaintext is not recoverable from the ciphertext.

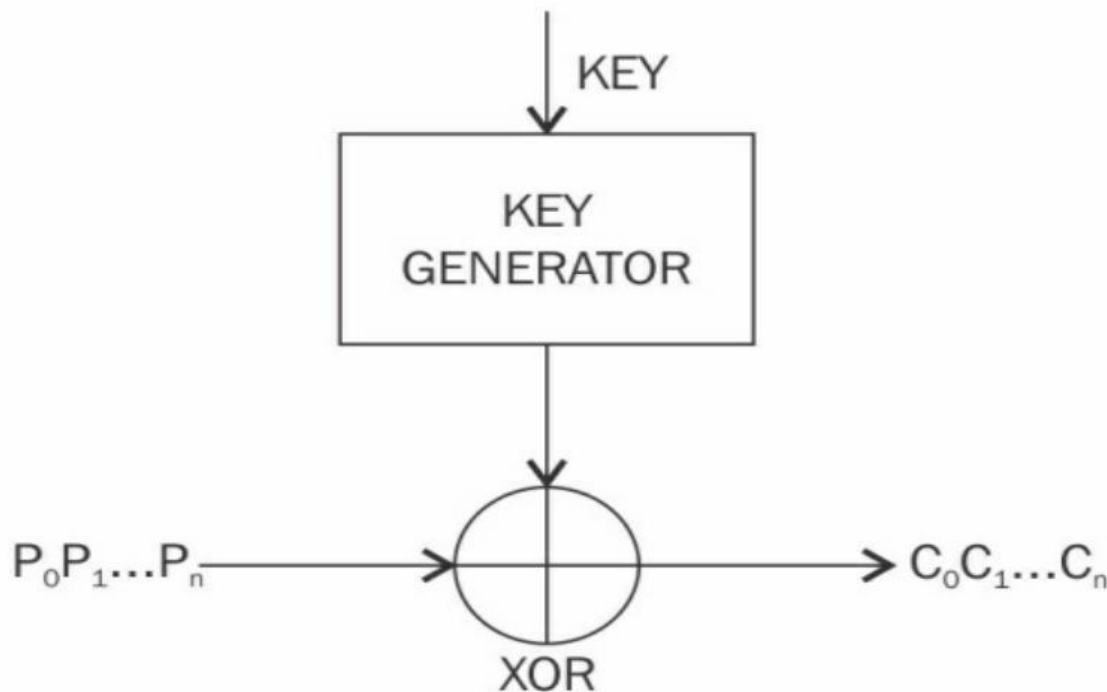
Types of Cryptography Algorithms

A) Symmetric Key Cryptography

- Symmetric cryptography refers to a type of cryptography where the key that is used to encrypt the data is the same one that is used for decrypting the data. Thus, it is also known as shared key cryptography.
- The key must be established or agreed upon before the data exchange occurs between the communicating parties. This is the reason it is also called secret key cryptography.
- There are two types of symmetric ciphers: **stream ciphers** and **block ciphers**.

Types of Cryptography Algorithms

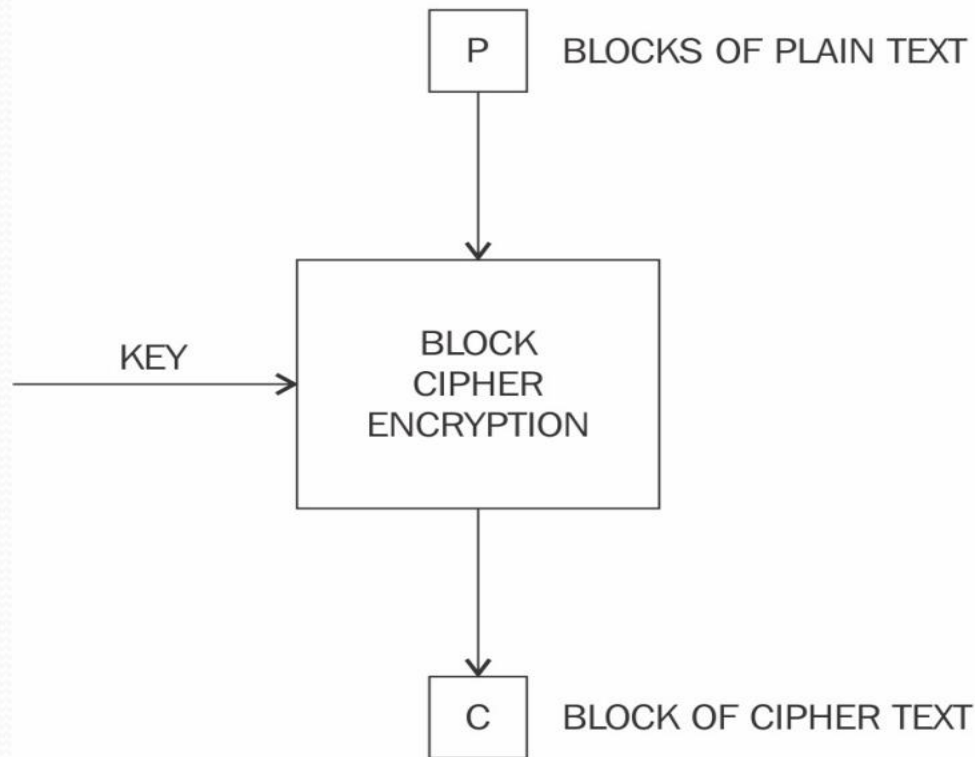
- **Stream ciphers** are encryption algorithms that apply encryption algorithms on a bit-by-bit basis (one bit at a time) to plaintext using a keystream.
- RC4 and A5 are example of stream ciphers



Operation of a stream cipher

Types of Cryptography Algorithms

- **Block ciphers** are encryption algorithms that break up the text to be encrypted (plaintext) into blocks of a fixed length and apply the encryption block-by-block
- Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are typical examples of block ciphers



Simplified operation of a block cipher

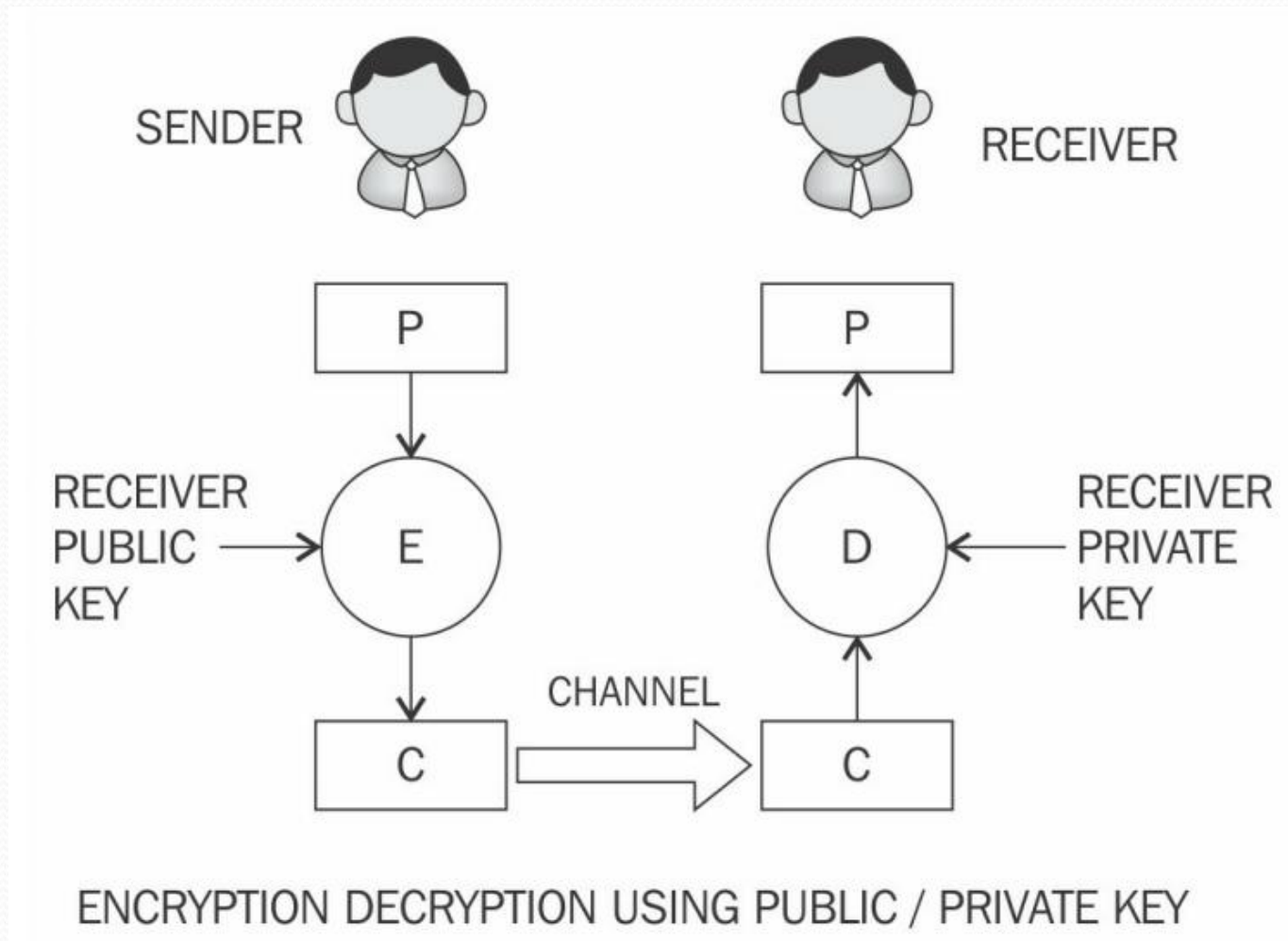
Types of Cryptography Algorithms

B) Asymmetric Key Cryptography

- Asymmetric cryptography refers to a type of cryptography where the key that is used to encrypt the data is different from the key that is used to decrypt the data.
- This is also known as public key cryptography.
- It uses both public and private keys to encrypt and decrypt data, respectively.
- Various asymmetric cryptography schemes are in use, including RSA, DSA, and ElGamal.

Types of Cryptography Algorithms

B) Asymmetric Key Cryptography



Types of Cryptography Algorithms

B) Asymmetric Key Cryptography

- Public key cryptosystems provide key setup, digital signatures, identity, identification, encryption and decryption.
- These algorithms are slow in computation than symmetric key algorithms.
- It is mostly not used in encryption of large files or the actual data.
- This is mostly used to exchange keys for symmetric algorithms. Once the keys are established securely, symmetric algorithm can be used to encrypt the data.

Types of Cryptography Algorithms

C) Elliptic Curve Cryptography (ECC)

- Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.
- ECC allows smaller keys to provide public key equivalent security.
- Elliptic curves are applicable for key agreement, digital signatures, pseudo-random generators and other tasks.
- Indirectly, they can be used for encryption by combining the key agreement with a symmetric encryption scheme

Types of Cryptography Algorithms

D) Cryptographic Hash Functions

- Hash functions are used to create fixed-length digests of arbitrarily-long input strings.
- Hash functions are keyless, and they provide the data integrity service.
- They are usually built using iterated and dedicated hash function construction techniques.
- Various families of hash functions are available, such as MD, SHA-1, SHA-2, SHA-3, RIPEMD, and Whirlpool.
- Hash functions are commonly used for to offer data integrity services, digital signatures and Message Authentication Codes

Types of Cryptography Algorithms

D) Cryptographic Hash Functions

- Cryptographic hash functions are one of the key component of blockchain.
- It is part of building block functions which provides security, privacy and consensus on blockchain platform.
- These are the mathematical functions/algorithms used to perform required conversion.

Types of Cryptography Algorithms

Characteristics of Cryptographic Hash Functions

1. Fixed output size
2. Deterministic
3. Easy to compute
4. Avalanche effect
5. Pre-image resistance
6. Second pre-image resistance
7. Collision resistance

Secure Hash Algorithms

Secure Hash Algorithms (SHA)

SHA0 : This is a 160-bit function introduced by NIST in 1993.

SHA1 : 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.

SHA2 : This category includes four functions defined by the number of bits of the hash: SHA-224, SHA-256, SHA-384, and SHA-512.

SHA3 : 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.

Design of Secure Hash Algorithms

Secure Hash Algorithms (SHA 256)

- SHA-256 has the input message size $< 2^{64}$ 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.
- SHA-256 is an iterated hash function producing a 256-bit output. The SHA-256 compression function takes a *512-bit message block* and a *256-bit intermediate hash value* as an **input** and produces a *256-bit new intermediate hash value* as an **output**.

Design of Secure Hash Algorithms

Secure Hash Algorithms Steps:

- SHA-256 works in 8 steps and divided into two phases as pre-processing (*step 1-3*) and hash computation (*step 4-8*).

Pre-processing

1. Padding of the message is used to adjust the length of a block to 512-bits if it is smaller than the required block size of 512-bits.
2. Parsing the message into message blocks, which ensures that the message and its padding is divided into equal blocks of 512-bits.
3. Creating the initial hash value, which consists of 8 words(32-bits) of the square roots of the first eight prime numbers.

These initial values are composed of eight (32 bit each) words. These initial parameters are picked at random.

Design of Secure Hash Algorithms

Secure Hash Algorithms Steps:

- SHA-256 works in 8 steps and divided into two phases as pre-processing (*step 1-3*) and hash computation (*step 4-8*).

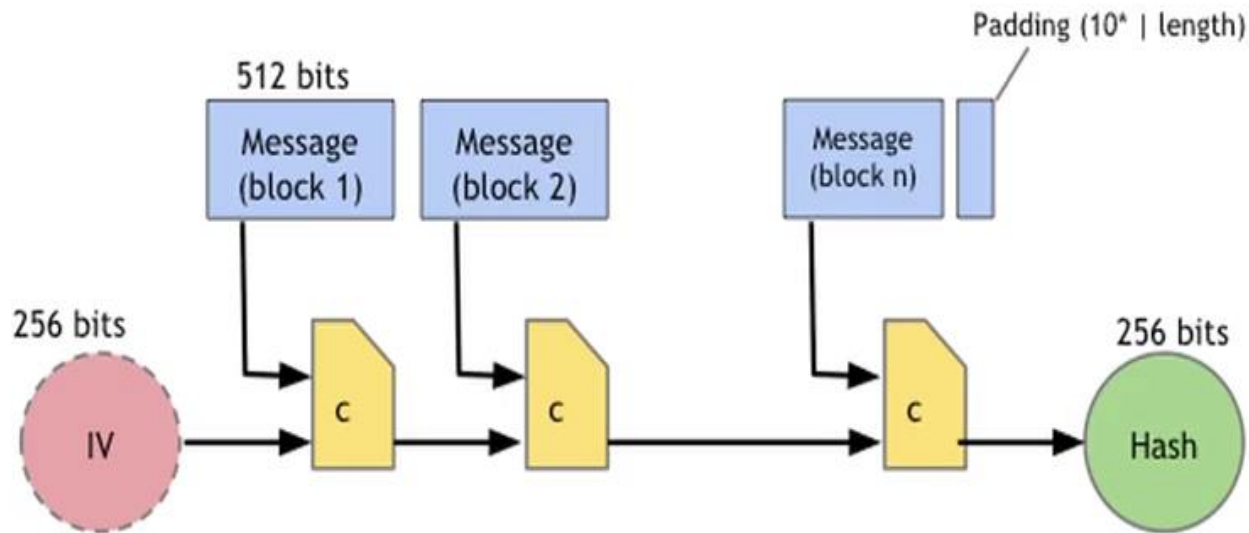
Hash-Computation

4. Each message block is then processed in a sequence, and it requires 64 rounds to compute the full hash output. Each round uses slightly different constants to ensure that no two rounds are the same.
5. The message schedule is prepared.
6. Eight working variables are initialized.
7. The intermediate hash value is calculated.
8. Finally, the message is processed, and the output hash is produced.

Design of Secure Hash Algorithms

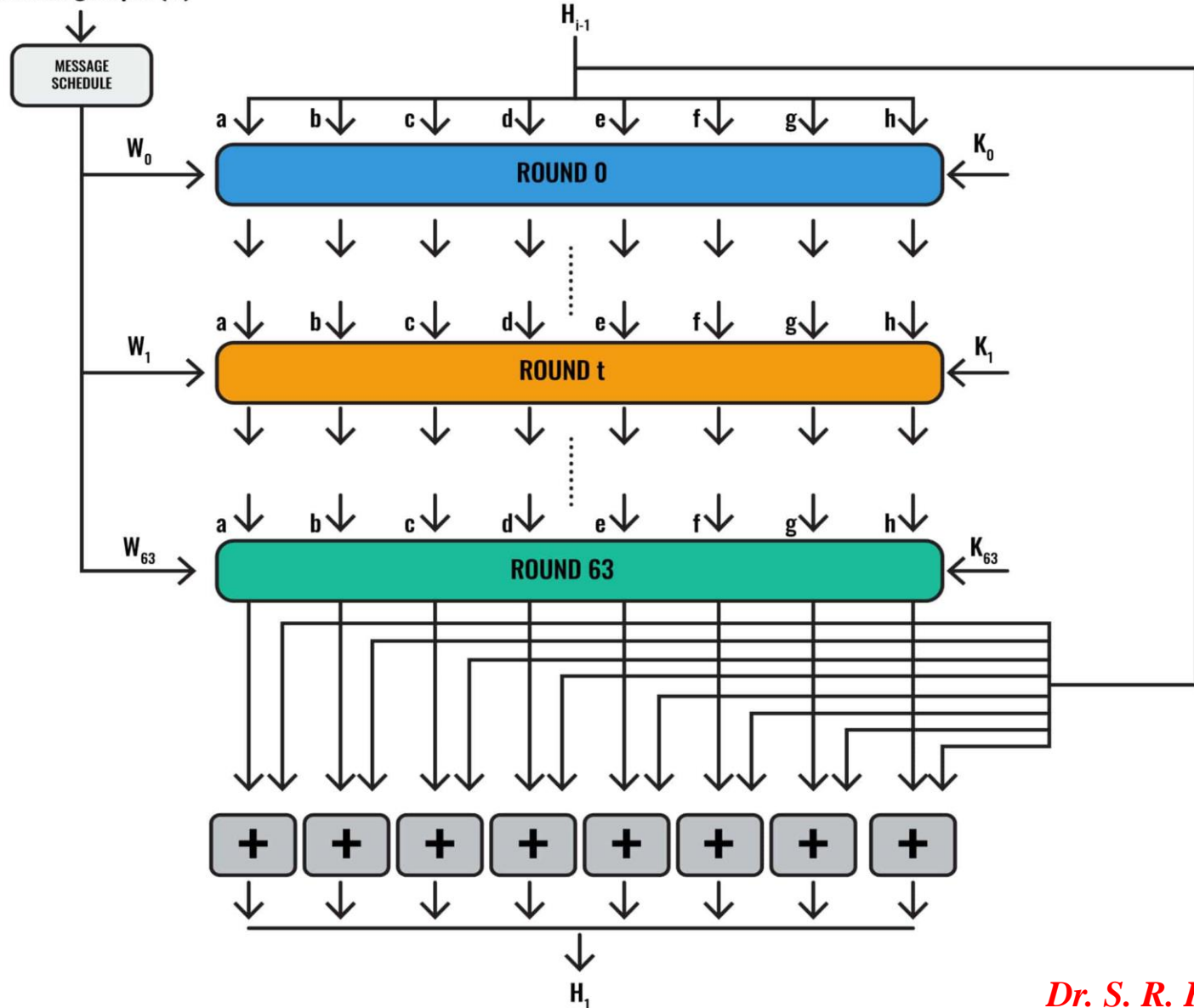
Secure Hash Algorithms :

SHA-256 hash function



Design of Secure Hash Algorithms

The message input (M)



Digital Signature

- Digital signatures provide a means of associating a message with an entity from which the message has originated.
- Digital signatures are used to provide data origin authentication and non-repudiation.
- 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.
- This digital signing, proves they are the rightful owner of the asset, for example, bitcoins.
- These transactions are verified again by other nodes on the network to ensure that the funds indeed belong to the node (user) who claims to be the owner.

Digital Signature Algorithm using (RSA)

- The following is the RSA digital signature algorithm
- This algorithm works in two phases as explain below:

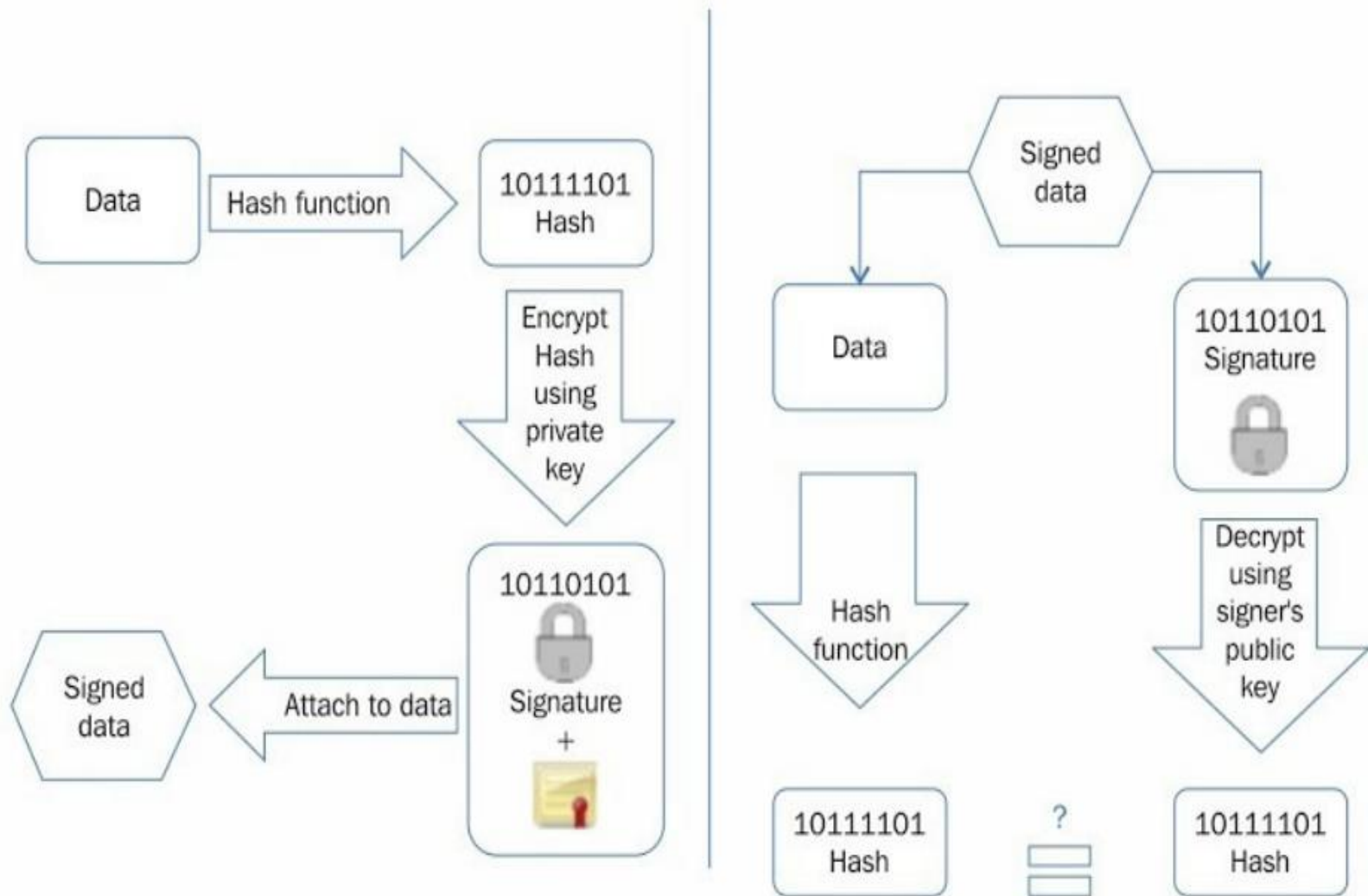
1. Calculate the hash value of the data packet

This will provide the data integrity guarantee as the hash can be computed at the receiver's end again and matched with the original hash to check whether the data has been modified in transit. Technically, message signing can work without hashing the data first, but is not considered secure.

2. Signs the hash value with the signer's private key

As only the signer has the private key, the authenticity of the signature and the signed data is ensured.

Digital Signature Algorithm using (RSA)



Digital signing (left) and verification process (right) (Example of RSA digital signatures)

Digital Signature Algorithm using (RSA)

Digital signatures important properties

- **Authenticity:** It means that the digital signatures are verifiable by a receiving party.
- **Unforgeability:** This property ensures that only the sender of the message can use the signing functionality using the private key. In other words, no one else can produce the signed message produced by a legitimate sender.
- **Nonreusability:** It means that the digital signature cannot be separated from a message and used again for another message

Digital Signature Algorithm using (RSA)

Authentication message from sender to receiver can be passed using two approaches

- **Sign then encrypt**

In 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:**

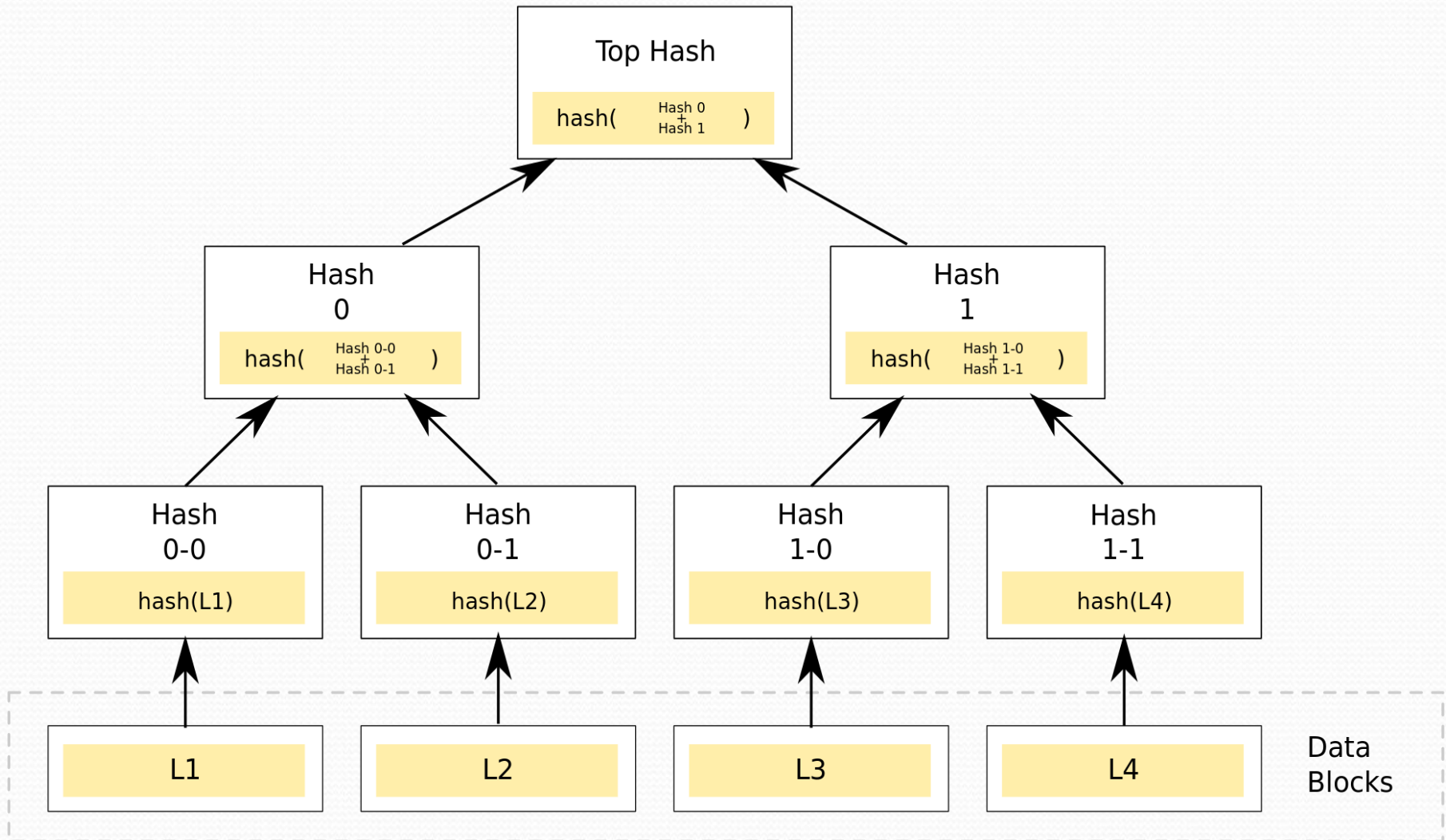
In this method, the sender encrypts the data using the receiver's public key and then digitally signs the encrypted data.

Merkel Trees

Merkel Trees

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

Merkel Trees



Merkel Trees

- An input of the data broken into number of blocks. Each block data is 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 of root node is called as Merkel root.
- While data sharing, data along with Merkel root is share so that receiver can calculate hash function and check authenticity of data.
- After receiving entire document Merkel root of sender and receiver is compared for verification and integrity.
- In blockchain Merkel trees serve to encode blockchain data more efficiently and securely. They are referred to as “binary hash trees”.

*End
of
Unit I*