CST 428 BLOCK CHAIN TECHNOLOGIES

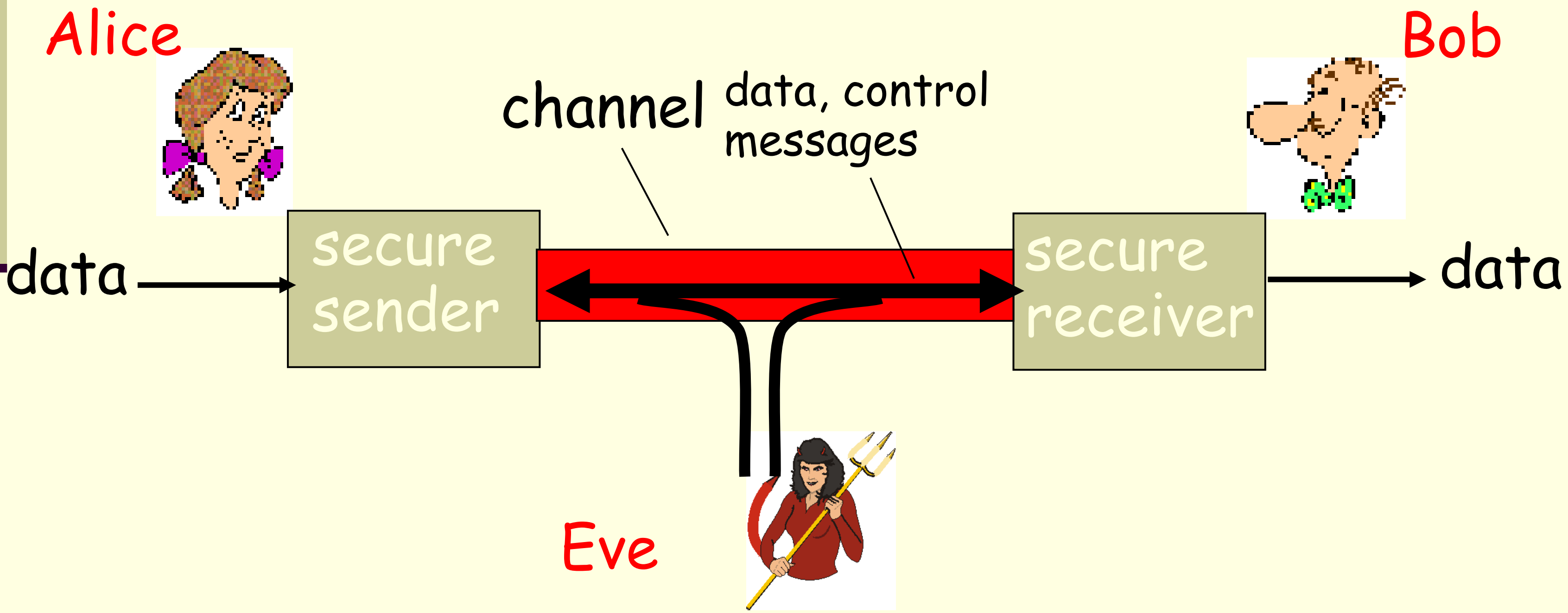
S8 CSE – ELECTIVE

MODULE – 1

Introduction to Cryptography

Definitions

- Cryptography = the science (art) of encryption
- Cryptanalysis = the science (art) of breaking encryption
- Cryptology = cryptography + cryptanalysis



Cryptography Goals

- Encryption – Prevent Eve from intercepting message
- Authentication – Prevent Eve from impersonating Alice

- **Cryptographic algorithms and protocols** can be grouped into four main areas:
  - **Symmetric encryption:** to conceal the contents of blocks or streams of data
  - **Asymmetric encryption:** to conceal small blocks of data, keys and hash function values, which are used in digital signatures
  - **Data integrity algorithms:** to protect blocks of data, such as messages from alteration
  - **Authentication protocols:** to authenticate the identity of entities

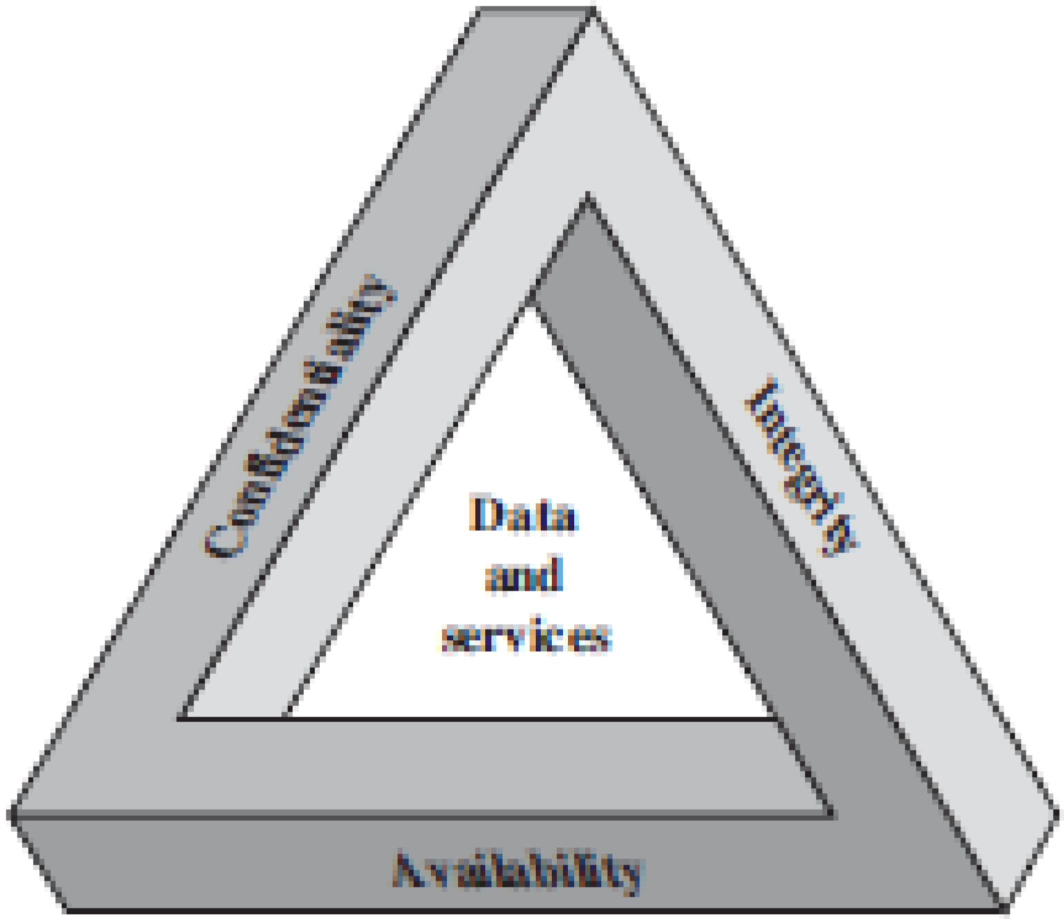


Figure 1.1 The Security Requirements Triad

Cryptographic Attacks

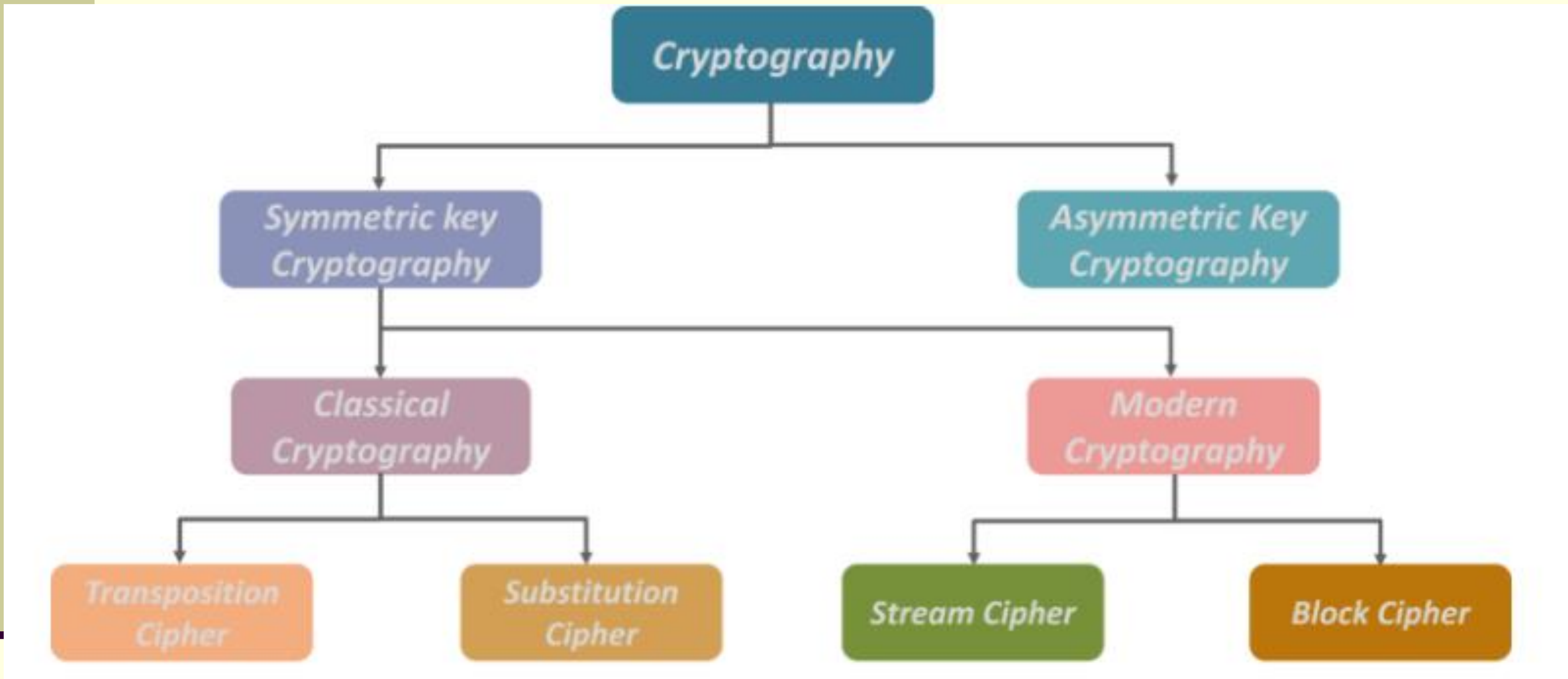
- Ciphertext only: attacker has only ciphertext.
- Known plaintext: attacker has plaintext and corresponding ciphertext.
- Chosen plaintext: attacker can encrypt messages of his choosing.
- Distinguishing attack: an attacker can distinguish your cipher from an ideal cipher (random permutation).
- A cipher must be secure against all of these attacks.

Kerckhoffs' Principle

- The security of an encryption system must depend only on the key, not on the secrecy of the algorithm.
- Nearly all proprietary encryption systems have been broken (Enigma, DeCSS, zipcrack).
- Secure systems use published algorithms (PGP, OpenSSL, Truecrypt).

Provable Security

- There is no such thing as a provably secure system.
- Proof of unbreakable encryption does not prove the system is secure.
- The only provably secure encryption is the one time pad: $C = P + K$, where K is as long as P and never reused.
- Systems are believed secure only when many people try and fail to break them.



Cryptography
Symmetric key Cryptography
Asymmetric Key Cryptography
Classical Cryptography
Modern Cryptography
Transposition Cipher
Substitution Cipher
Stream Cipher
Block Cipher

Transposition Cipher

1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		

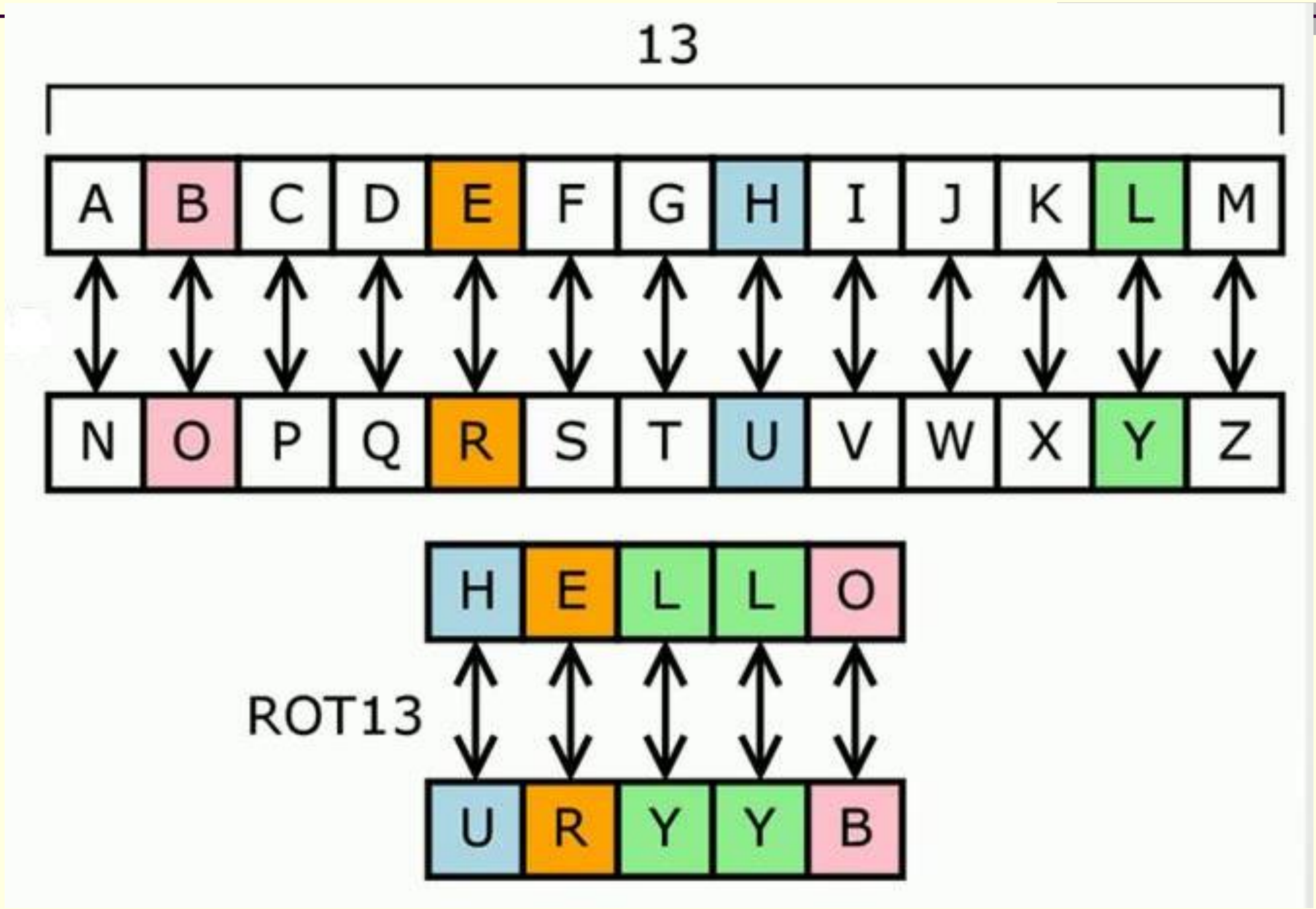
4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A		T	

Plain Text: MEET ME AFTER PARTY

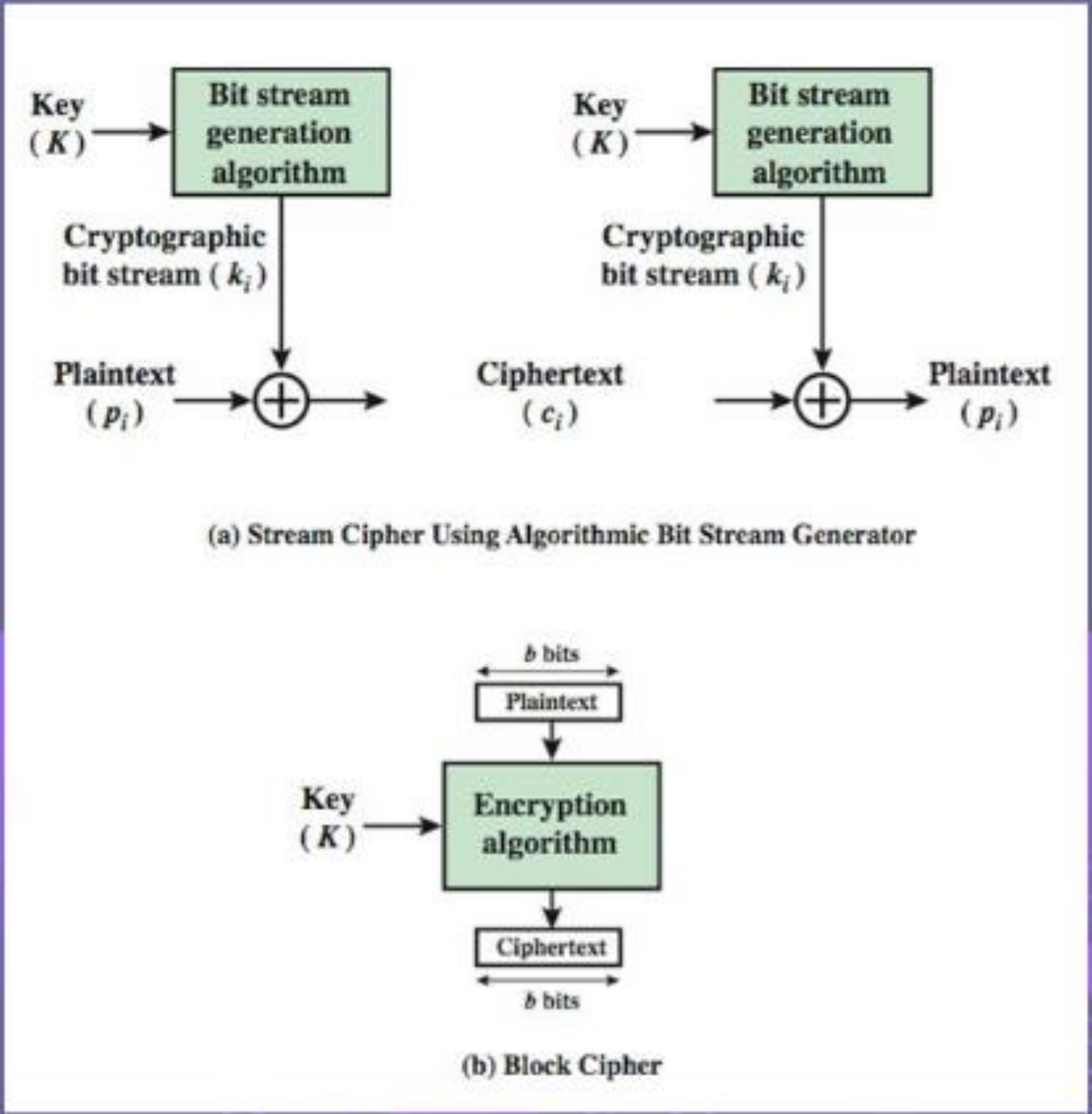
Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Substitution Cipher



Block vs Stream Ciphers



Difference ?

S.NO	Block Cipher	Stream Cipher
1.	Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	Stream Cipher Converts the plain text into cipher text by taking 1 byte of plain text at a time.
2.	Block cipher uses either 64 bits or more than 64 bits.	While stream cipher uses 8 bits.
3.	The complexity of block cipher is simple.	While stream cipher is more complex.

Symmetric Key Encryption

Introduction

- Also known as SECRET KEY, SINGLE KEY, PRIVATE KEY
- Assumption: Sender and Receiver share already a secret key
- Assumption requires solution to key-distribution problem
- Symmetric key algorithms also popular for file encryption, then

Encrypter = Decrypter

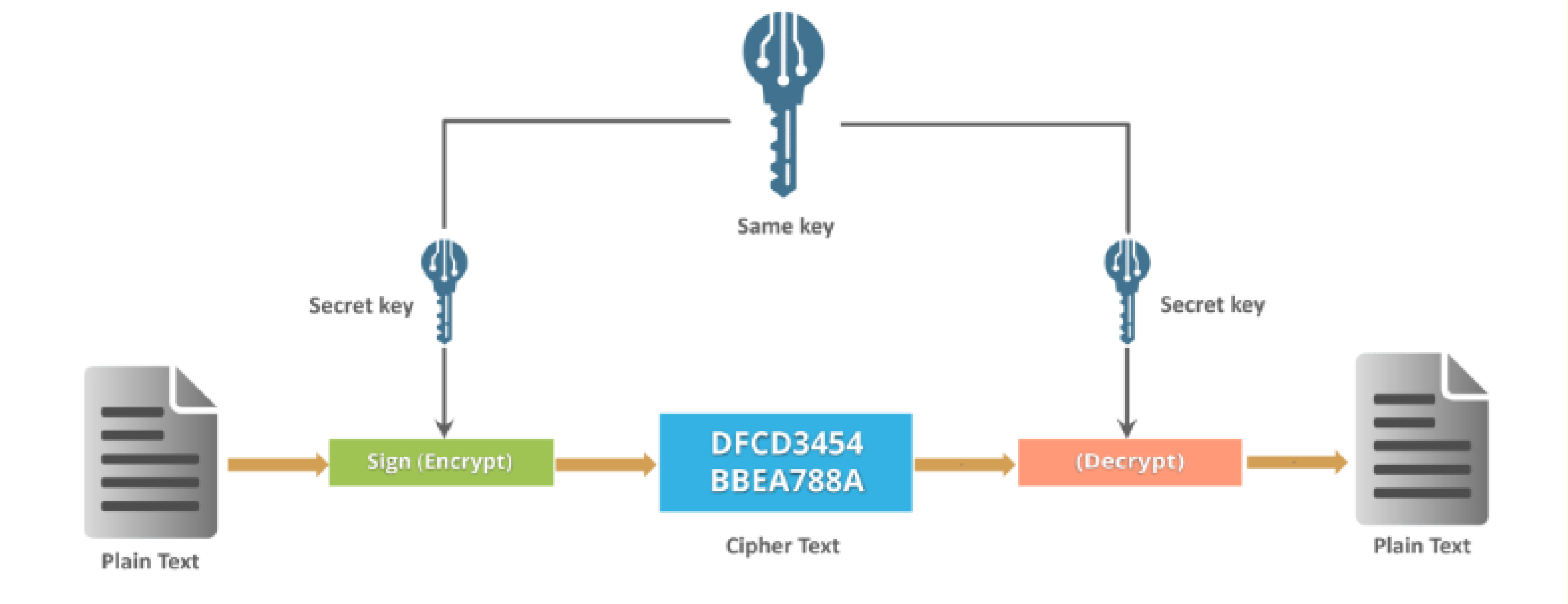
WEAK ALGORITHMS

- Classical substitution and transposition ciphers

"STRONGER" ALGORITHMS

- DES – No longer considered safe
- Triple-DES
- AES (Rijndael)
- IDEA
- RC5, RC6
- Blowfish
- Many others

Symmetric Key Cryptography



Encryption & Decryption

Key (K)

Plaintext (P) → Encrypt (E) → Ciphertext (C)

$C = E_K(P)$

Same Key (K)

Ciphertext (C) → Decrypt (D) → Plaintext (P)

$P = D_K(C)$

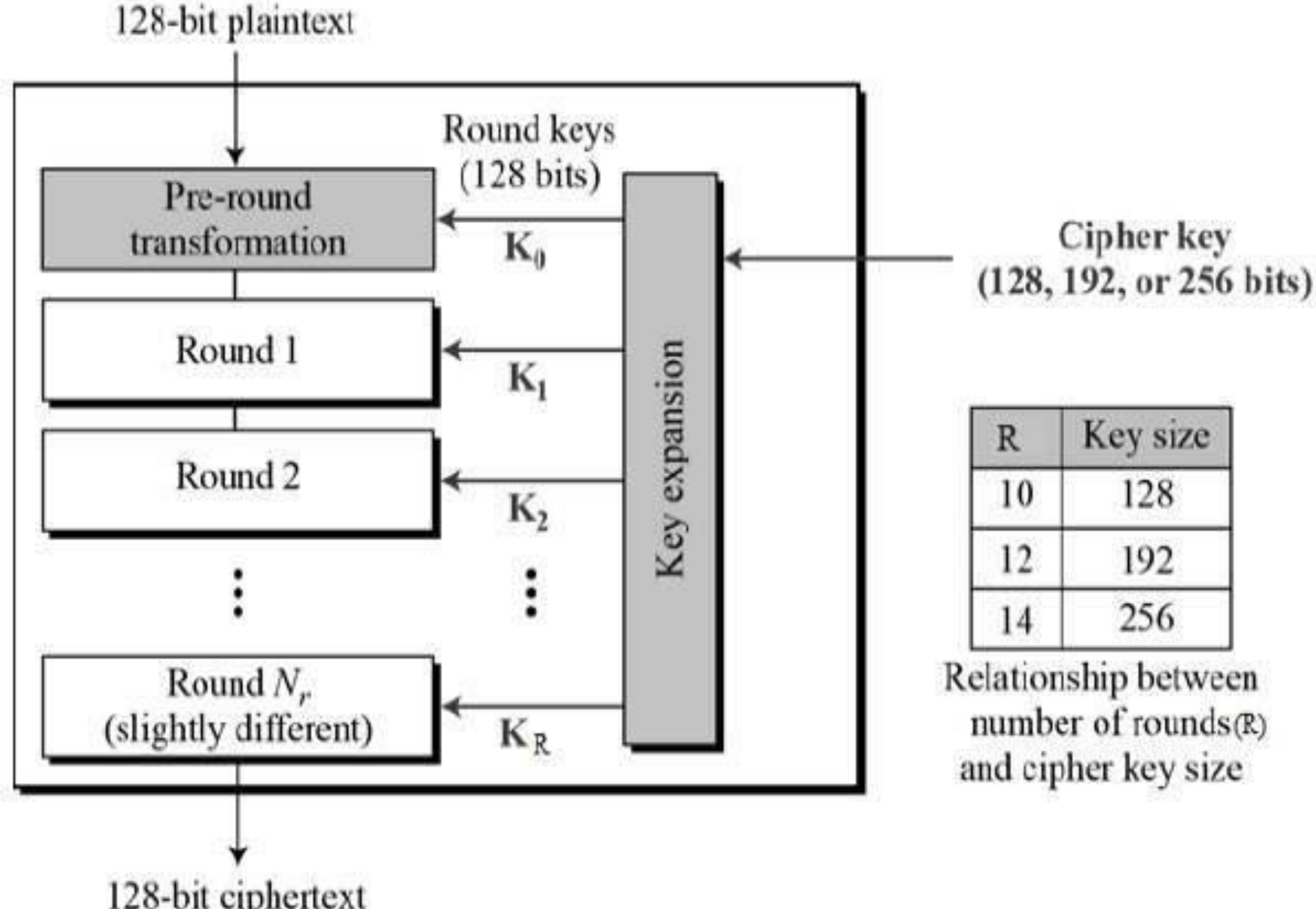
$\boxed{P = D_K(E_K(P))}$

AES- Advanced Encryption Standard

- also known as Rijndael algorithm
- symmetric block cipher algorithm
- block/chunk size of 128 bits
- converts these individual blocks using keys of 128, 192, and 256 bits
- Once it encrypts these blocks, it joins them together to form the cipher text
- based on a substitution-permutation network, known as SP network

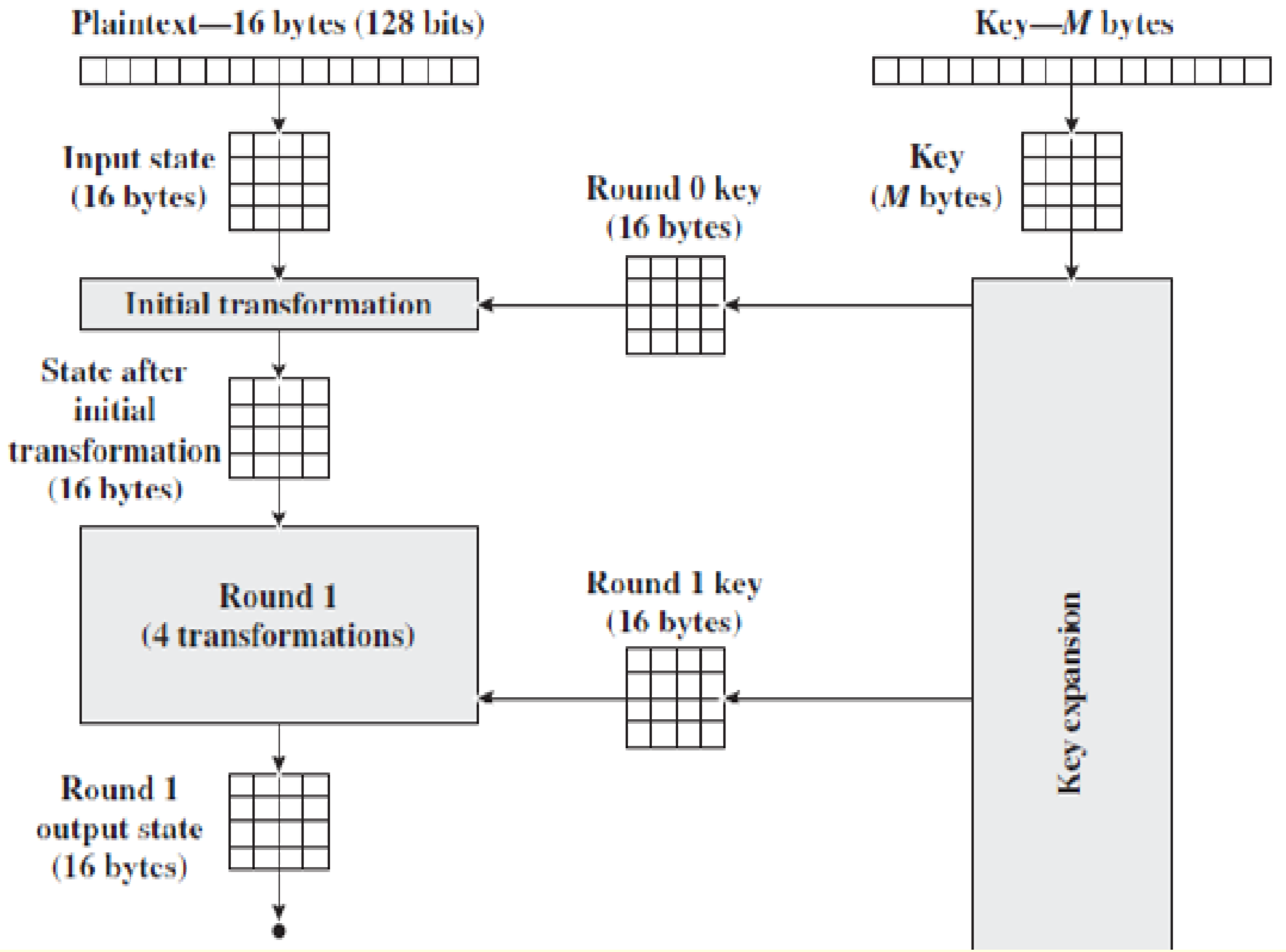
- Sub Key Generation
  - uses 128 bit Master Key
  - Key is processed in words of size 32 bit (4 words / 16 bytes)
  - Each sub key size is 32 bit / 1 word/4 bytes
  - Each round have 4 sub keys (128 bit/4 words/16 bytes)
  - For pre round calculation we use 4 sub key initially
  - Total sub key is 44

AES- Advanced Encryption Standard

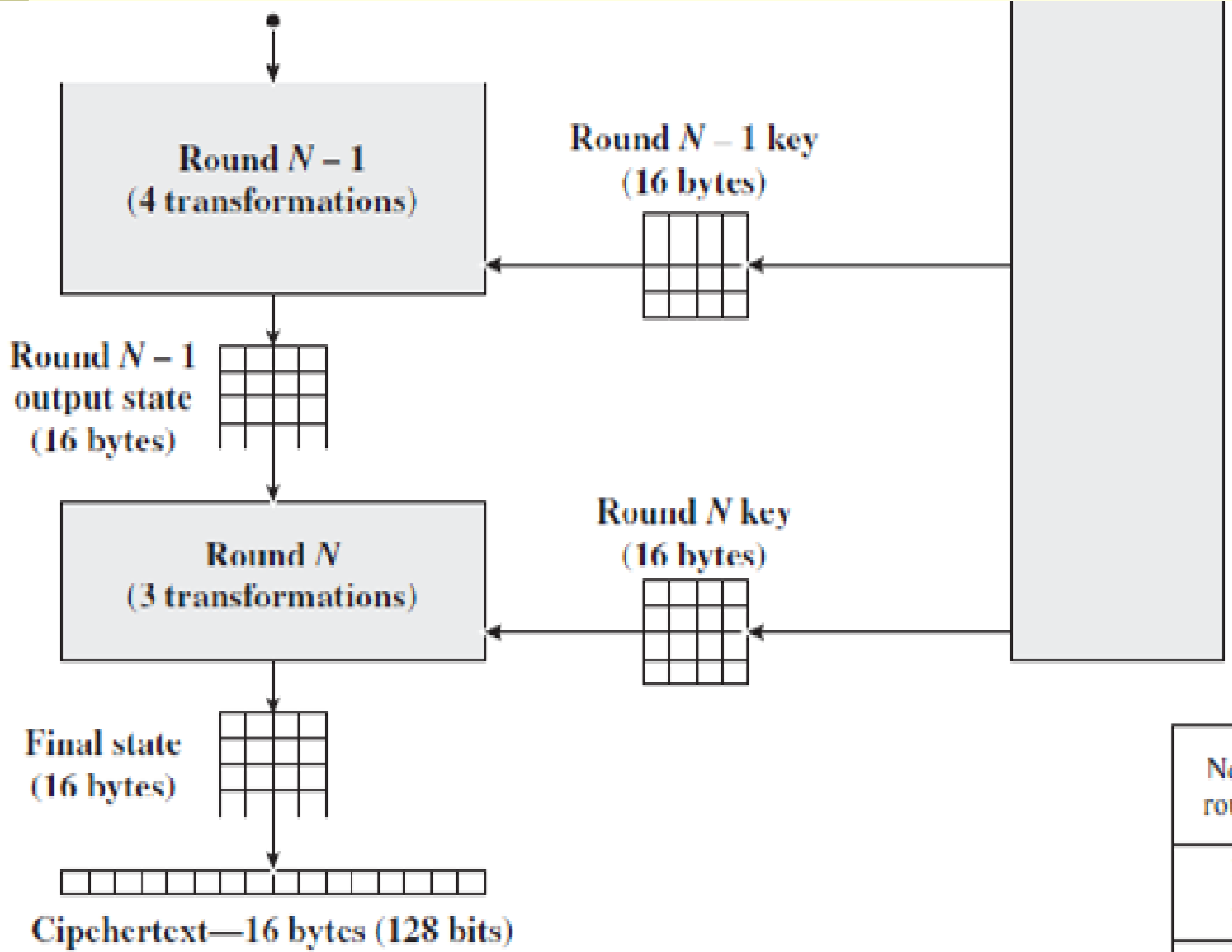


R	Key size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size

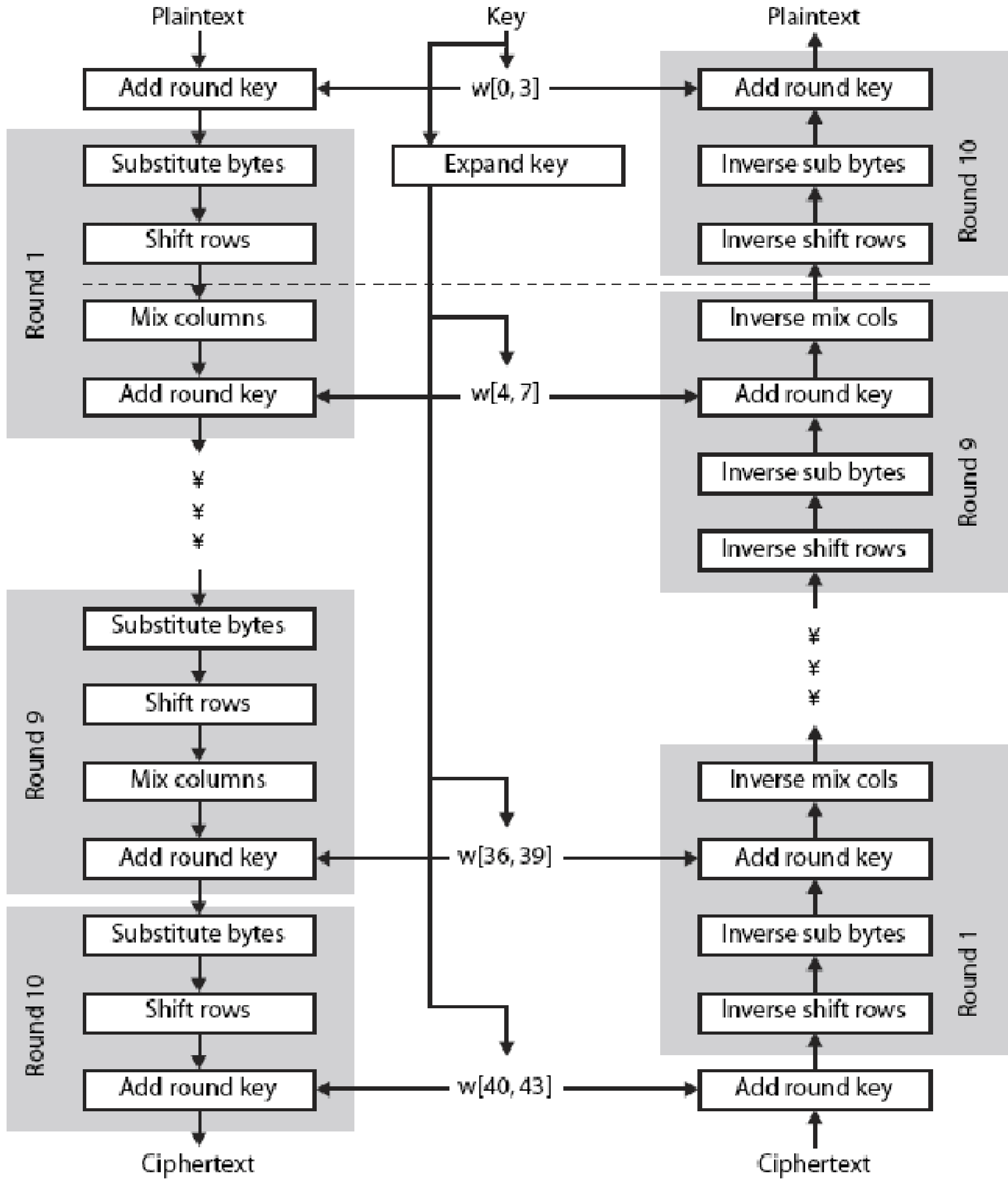


Plaintext—16 bytes (128 bits)
Key—M bytes
Input state
(16 bytes)
Key
(M bytes)
Round 0 key
(16 bytes)
Initial transformation
State after
initial
transformation
(16 bytes)
Round 1
(4 transformations)
Round 1 key
(16 bytes)
Round 1
output state
(16 bytes)
Key expansion



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 5.1 AES Encryption Process



Plaintext
Key
Plaintext
Add round key
w[0, 3]
Add round key
Substitute bytes
Expand key
Inverse sub bytes
Round 10
Shift rows
Inverse shift rows
Round 1
Mix columns
Inverse mix cols
Add round key
w[4, 7]
Add round key
Inverse sub bytes
Round 9
Inverse shift rows
Substitute bytes
Shift rows
Round 9
Mix columns
Inverse mix cols
Add round key
w[36, 39]
Add round key
Inverse sub bytes
Round 1
Substitute bytes
Inverse shift rows
Shift rows
Round 10
Add round key
w[40, 43]
Add round key
Ciphertext
Ciphertext

AES (Advanced Encryption Standard)

❑ Plain Text transform into Matrix Form

- For Example, “AES USES A MATRIX”.
- Plain text (128-bit) converts into 4x4 matrix of bytes.

	DEC	HEX		DEC	HEX
A	00	00	N	13	0D
B	01	01	O	14	0E
C	02	02	P	15	0F
D	03	03	Q	16	10
E	04	04	R	17	11
F	05	05	S	18	12
G	06	06	T	19	13
H	07	07	U	20	14
I	08	08	V	21	15
J	09	09	W	22	16
K	10	0A	X	23	17
L	11	0B	Y	24	18
M	12	0C	Z	25	19

Text	A	E	S	U	S	E	S	A	M	A	T	R	I	X	Z	Z
Hexadecimal	00	04	12	14	12	04	12	00	0C	00	13	11	08	17	19	19

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 17 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix} \text{State}$$

Substitute Byte

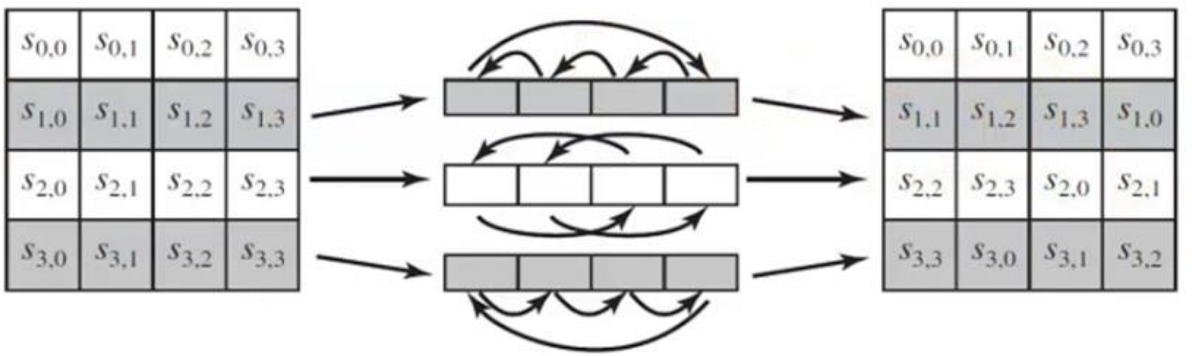
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES (Advanced Encryption Standard)

2. Shift Row transformation

- The shift row transformation is called ShiftRows.

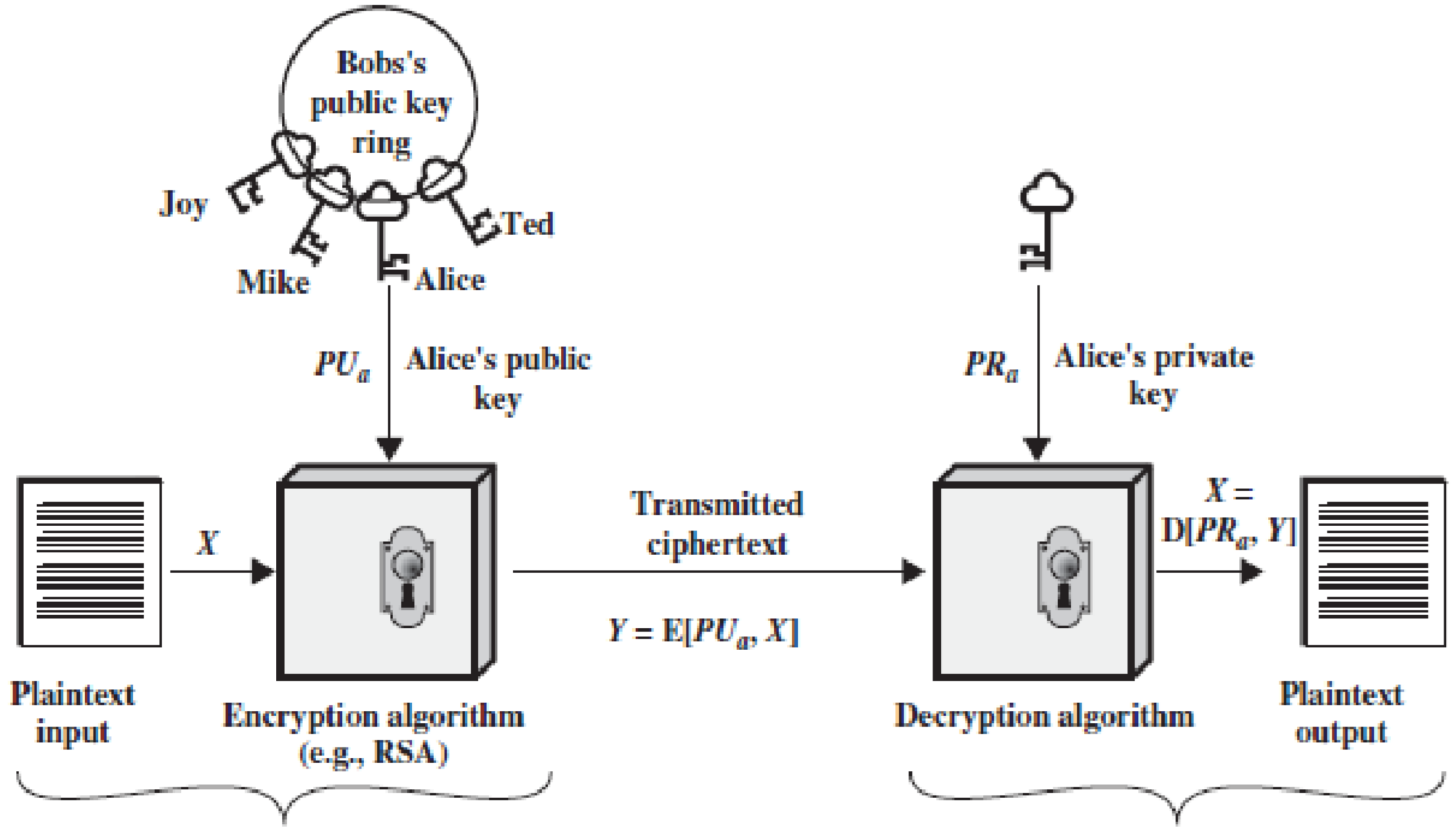
- Rules of shifting rows,
  - Row 1 → No Shifting
  - Row 2 → 1 byte left shift
  - Row 3 → 2 byte left shift
  - Row 4 → 3 byte left shift



$$\begin{pmatrix} s'0,1 \\ s'1,1 \\ s'2,1 \\ s'3,1 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} s0,1 \\ s1,1 \\ s2,1 \\ s3,1 \end{pmatrix}$$

Transform Matrix of Mix Columns

Asymmetric Cryptography



(a) Encryption with public key

Asymmetric Cryptography

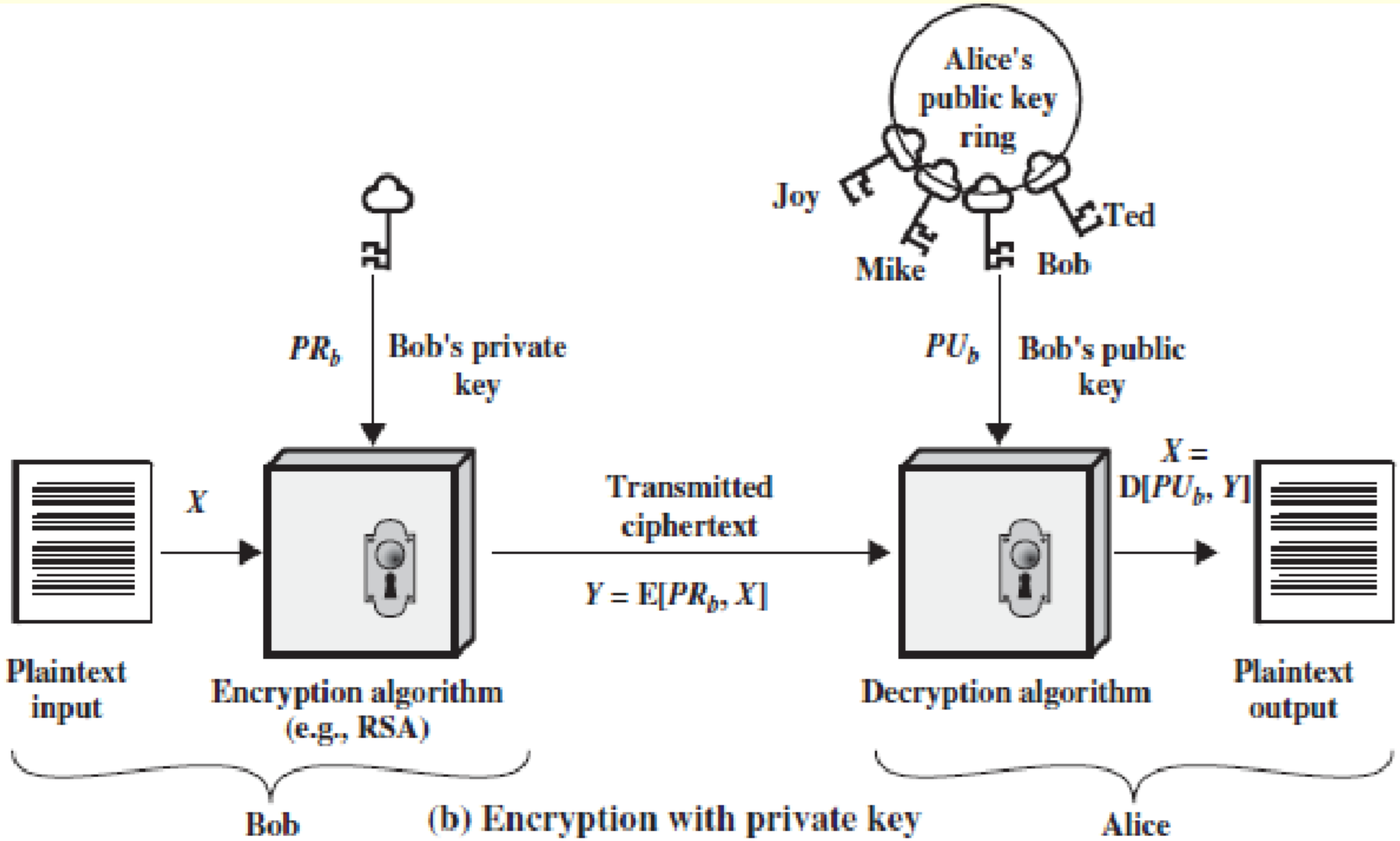


Figure 9.1 Public-Key Cryptography

Asymmetric Cryptography

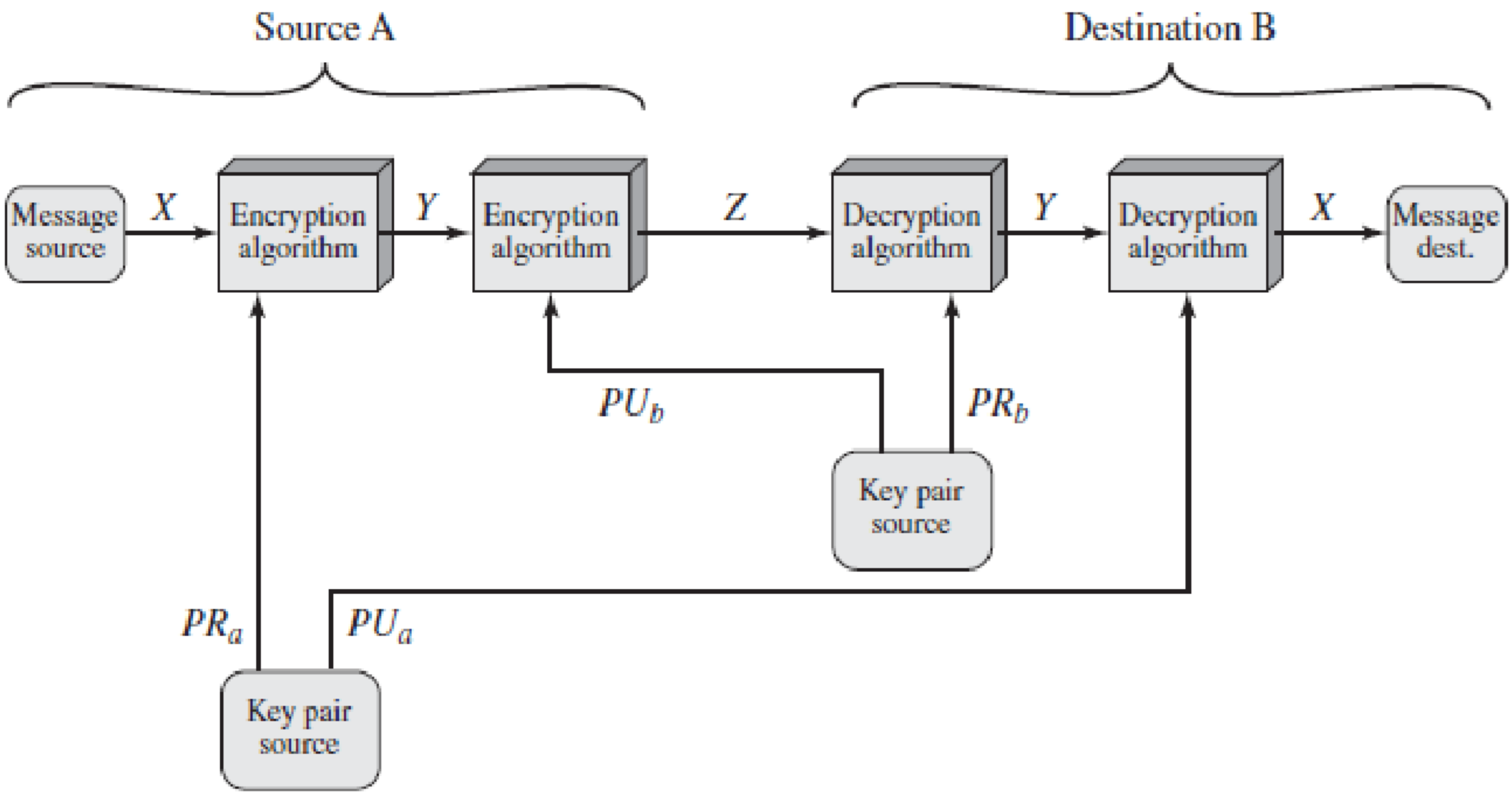


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

RSA

- **RSA** stands for **Rivest, Shamir, Adleman** - creators
- public-key encryption technique used for secure data transmission especially over the internet

RSA

p, q, two prime numbers	(private, chosen)
$n = pq$	(public, calculated)
e, with $\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$	(public, chosen)
$d \equiv e^{-1} \pmod{\phi(n)}$	(private, calculated)

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext	$M < n$
Ciphertext	$C = M^e \pmod{n}$

Decryption

Ciphertext	C
Plaintext	$M = C^d \pmod{n}$

1. Select two prime numbers, $p = 17$ and $q = 11$.
2. Calculate $n = pq = 17 \times 11 = 187$.
3. Calculate $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.
4. Select e such that e is relatively prime to $\phi(n) = 160$ and less than $\phi(n)$; we choose $e = 7$.
5. Determine d such that $de \equiv 1 \pmod{160}$ and $d < 160$. The correct value is $d = 23$, because $23 \times 7 = 161 = (1 \times 160) + 1$; d can be calculated using the extended Euclid's algorithm (Chapter 4).

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$. The example shows the use of these keys for a plaintext input of $M = 88$. For encryption, we need to calculate $C = 88^7 \bmod 187$. Exploiting the properties of modular arithmetic, we can do this as follows.

$$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \\ \times (88^1 \bmod 187)] \bmod 187$$

$$88^1 \bmod 187 = 88$$

$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$

$$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$$

$$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$$

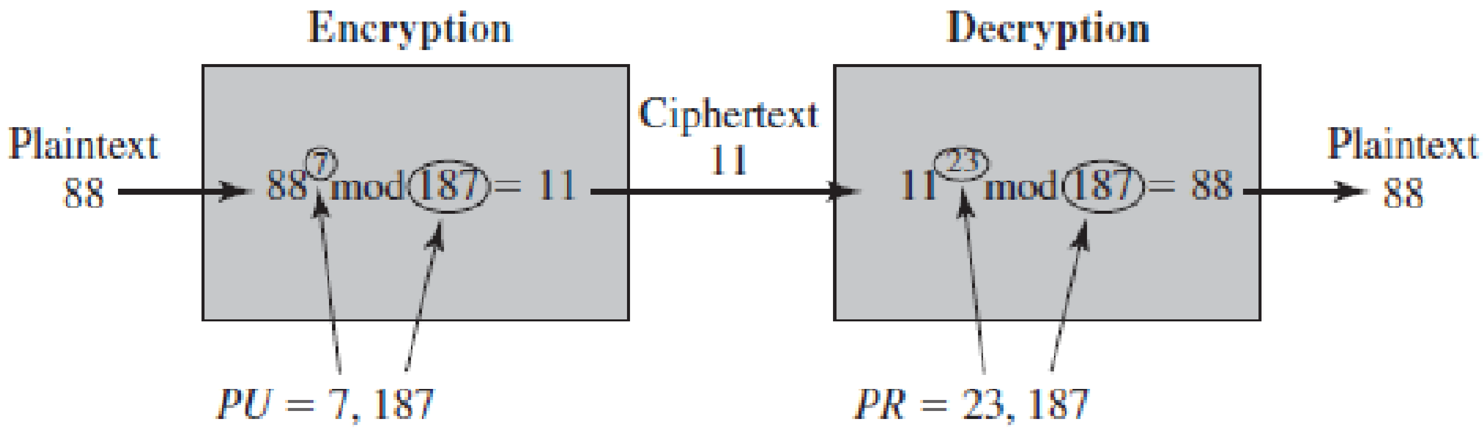


Figure 9.6 Example of RSA Algorithm

For decryption, we calculate $M = 11^{23} \bmod 187$:

$$11^{23} \bmod 187 = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \\ \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$$

$$11^1 \bmod 187 = 11$$

$$11^2 \bmod 187 = 121$$

$$11^4 \bmod 187 = 14{,}641 \bmod 187 = 55$$

$$11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33$$

$$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$$

Question

- Explain public and private keys. Perform encryption and decryption using RSA for p=3, q=11, e= 7 and M=5

Elliptic Curve Cryptography

- Asymmetric Public key cryptosystem
- Provides security with smaller key size
- alternative to the Rivest-Shamir-Adleman (RSA) cryptographic algorithm
- used for digital signatures in cryptocurrencies, such as Bitcoin and Ethereum, as well as one-way encryption of emails, data and software
- fast key generation, fast key agreement and fast signatures

Elliptic Curve Cryptography

- 2 families of Elliptic curves
  - Prime curves over Z_p
    - uses cubic equation in which variables and coefficients from 0 through p-1
    - best for software applications
  - Binary curves over $GF(2^m)$
    - variables and coefficients in $GF(2^m)$
    - best for hardware applications

Elliptic Curve Cryptography

- Makes use of Elliptic Curves $y^2=x^3 + ax + b$
- *variables and coefficients restricted to elements in a finite field*
- Properties of Elliptical Curve
  - symmetric over x axis
  - A non vertical line will intersect the curve at most 3 points

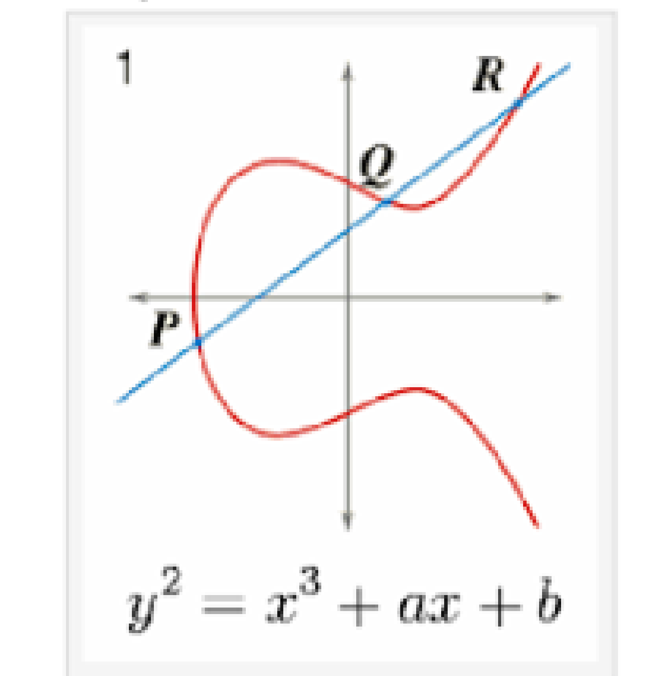
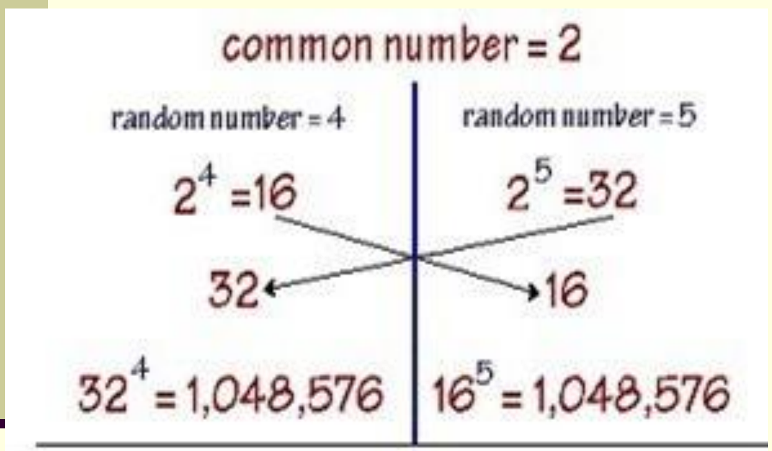


Fig. 2 shows simple elliptic curve.

ECC Diffie Hellman Key Exchange



Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private	n_A	$n_A < n$
Calculate public	P_A	$P_A = n_A * G$

User B Key Generation

Select private	n_B	$n_B < n$
Calculate public	P_B	$P_B = n_B * G$

Calculation of Secret Key by User A

$K = n_A * P_B$

Calculation of Secret Key by User B

$K = n_B * P_A$

ECC Encryption/ Decryption

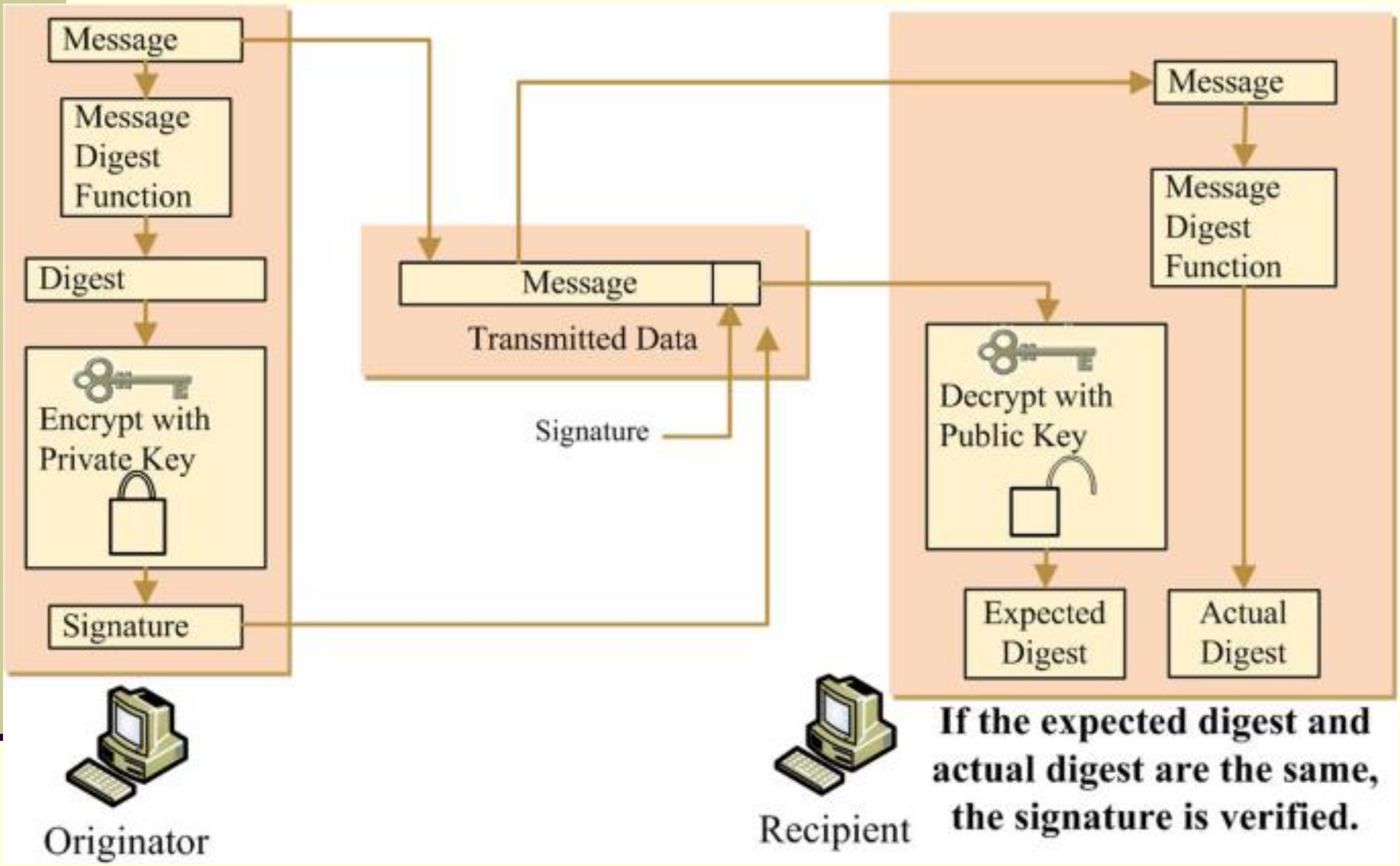
To encrypt and send a message P_m to B, A chooses a random positive integer k and produces the ciphertext C_m consisting of the pair of points:

$$Cm = \{kG, P_m + kP_b\}$$

For decryption, B multiplies the first point in the pair by B's private key and subtracts the result from the second point

Digital Signatures

- an electronic, encrypted, stamp of authentication on digital information such as email messages, macros, or electronic documents
- confirms that the information originated from the signer and has not been altered



RSA Digital Signature Process

RSA Digital Signature Example

- **RSA signature example:**
- $P = 17$, $q = 11$
- $n = 187$
- $\phi(n) = 160$
- $e = 7$
- $d = 23$
- $m = 88$
- $s = 88^{23} \bmod 187 = 11$ **Signed message (88, 11)**
- **Verification of (m , s)** : $11^7 \bmod 187 = 88$
- **Verification of forged message** (88 , 13): $13^7 \bmod 187 = 106$

Secure Hash Functions

Authentication Algorithm can be classified into

1. Message Encryption - uses encryption algorithm
2. Message Authentication Code - generates fixed length code
3. Hash Function - generates fixed length code

- Message authentication code - uses the message authentication function on the plain text along with the key to generate fixed length code. This fixed length code will be appended with the message and send to the receiver for authentication
- hash function do not use the key for generating the fixed length code

Secure Hash Functions

Characteristics :

- Fixed Length output
- Avalanche Effect
- Collision Resistant

- Commonly used Hash functions :
- MD5 (Message Digest)
- SHA (Secure Hash Function)

- Used in Blockchain and cryptocurrencies

SHA- 256

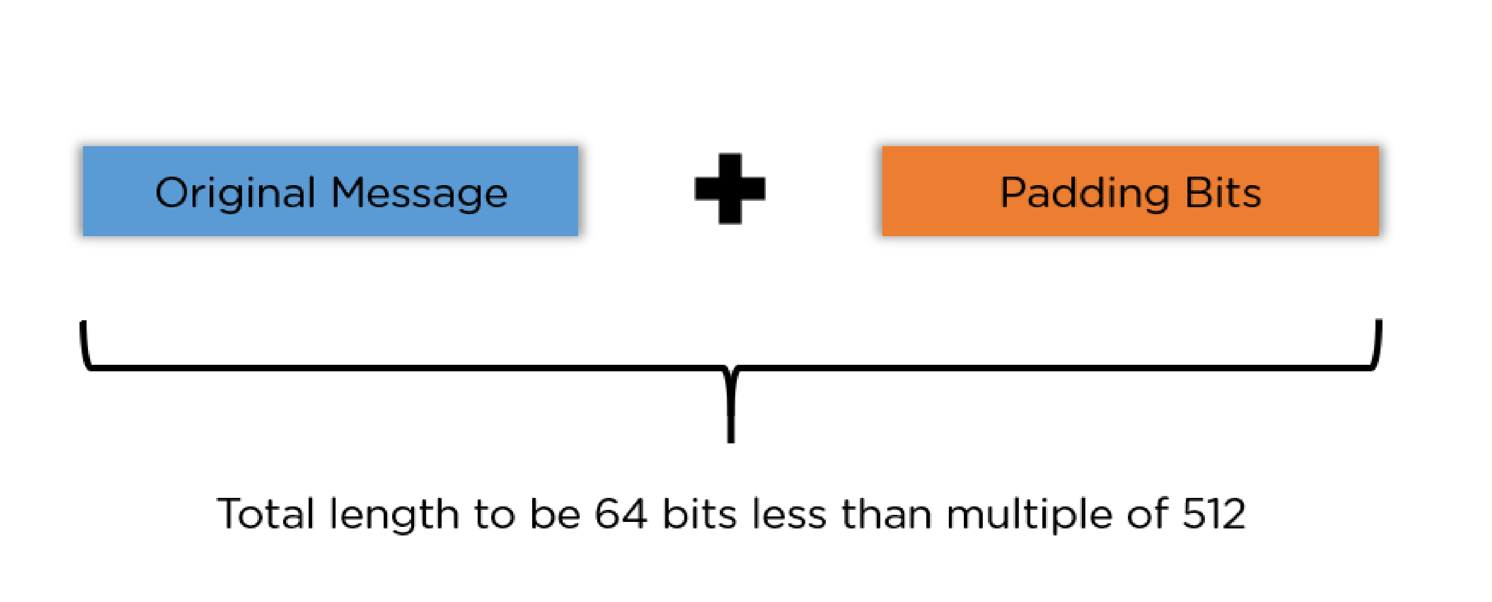
- one of the strongest hash functions available
- Secure communications for websites and web services are based on files known as certificates
- They are used to establish and authenticate secure connections
- These certificates contain cryptographic elements that are generated using algorithms such as SHA-256

SHA- 256

- hash value will always be 256 bits
- Characteristics
  - Message Length: Any length
  - Hash Length: 256 bits
    - Bigger hash suggest significantly more calculations at the cost of speed and space
  - Irreversible: all hash functions such as the SHA 256 are irreversible neither get a plaintext when you have the digest beforehand nor should the digest provide its original value when you pass it through the hash function again

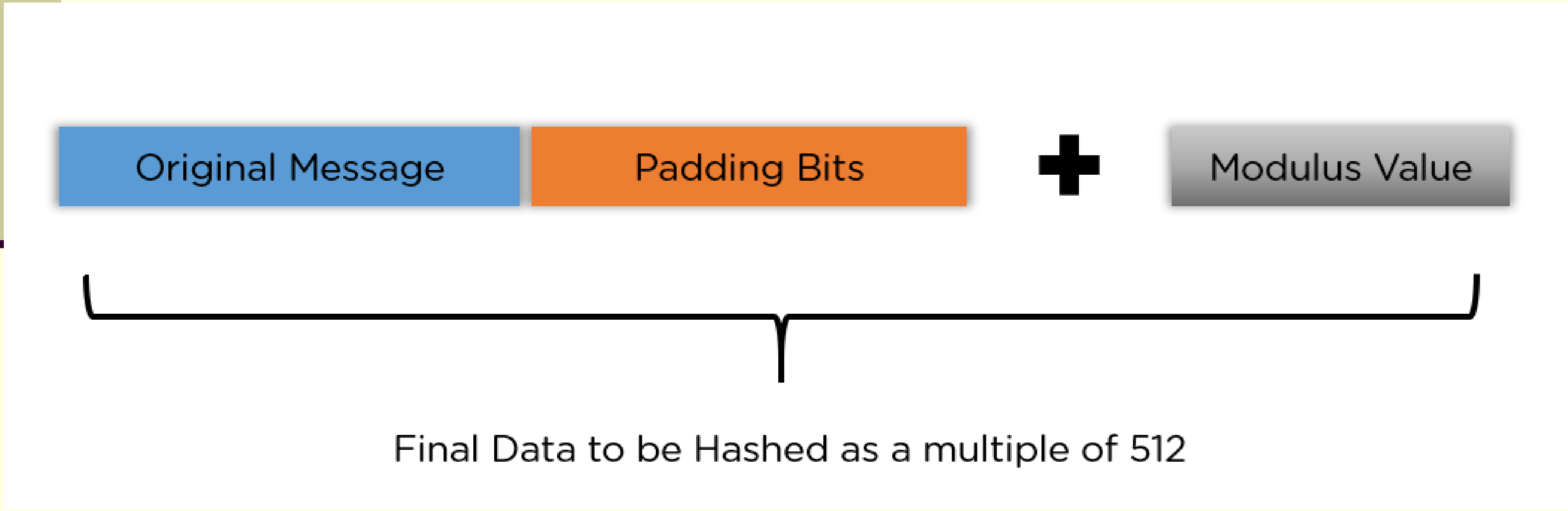
SHA- 256 Steps

- Padding
  - first bit should be one, and the rest of it should be filled with zeroes



SHA- 256 Steps

- Padding Length
  - add 64 bits of data to make the final plaintext a multiple of 512 applying the modulus to original text without the padding



SHA- 256 Steps

- Initialising buffers
  - initialize the default values for eight buffers to be used in the rounds
  - store 64 different keys in an array, ranging from K[0] to K[63]

```
k[0..63] :=
   0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5, 0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
   0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3, 0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
   0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc, 0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
   0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7, 0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
   0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13, 0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
   0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3, 0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
   0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5, 0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
   0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208, 0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2
```

a = 0x6a09e667

b = 0xbb67ae85

c = 0x3c6ef372

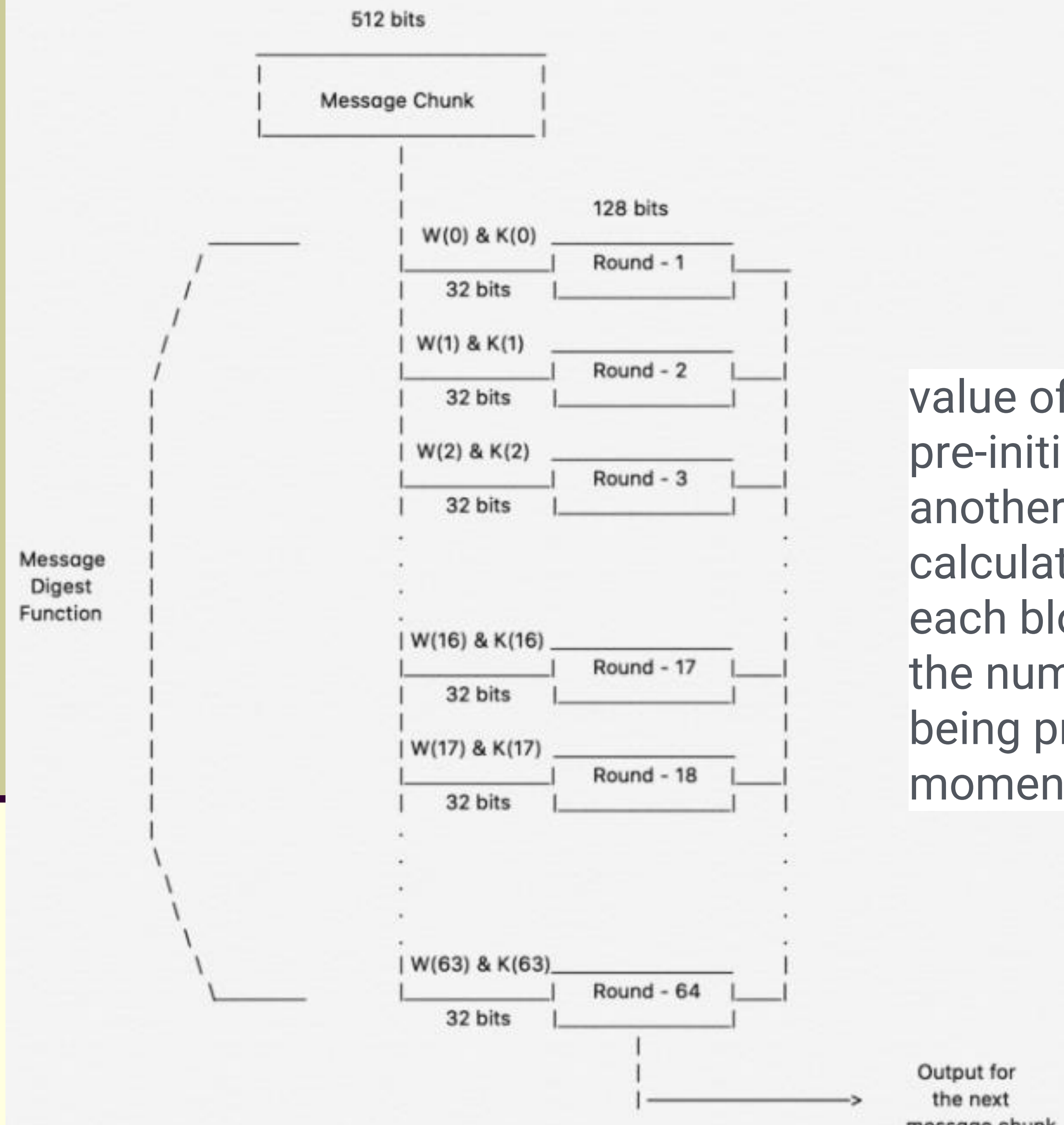
d = 0xa54ff53a

e = 0x510e527f

f = 0x9b05688c

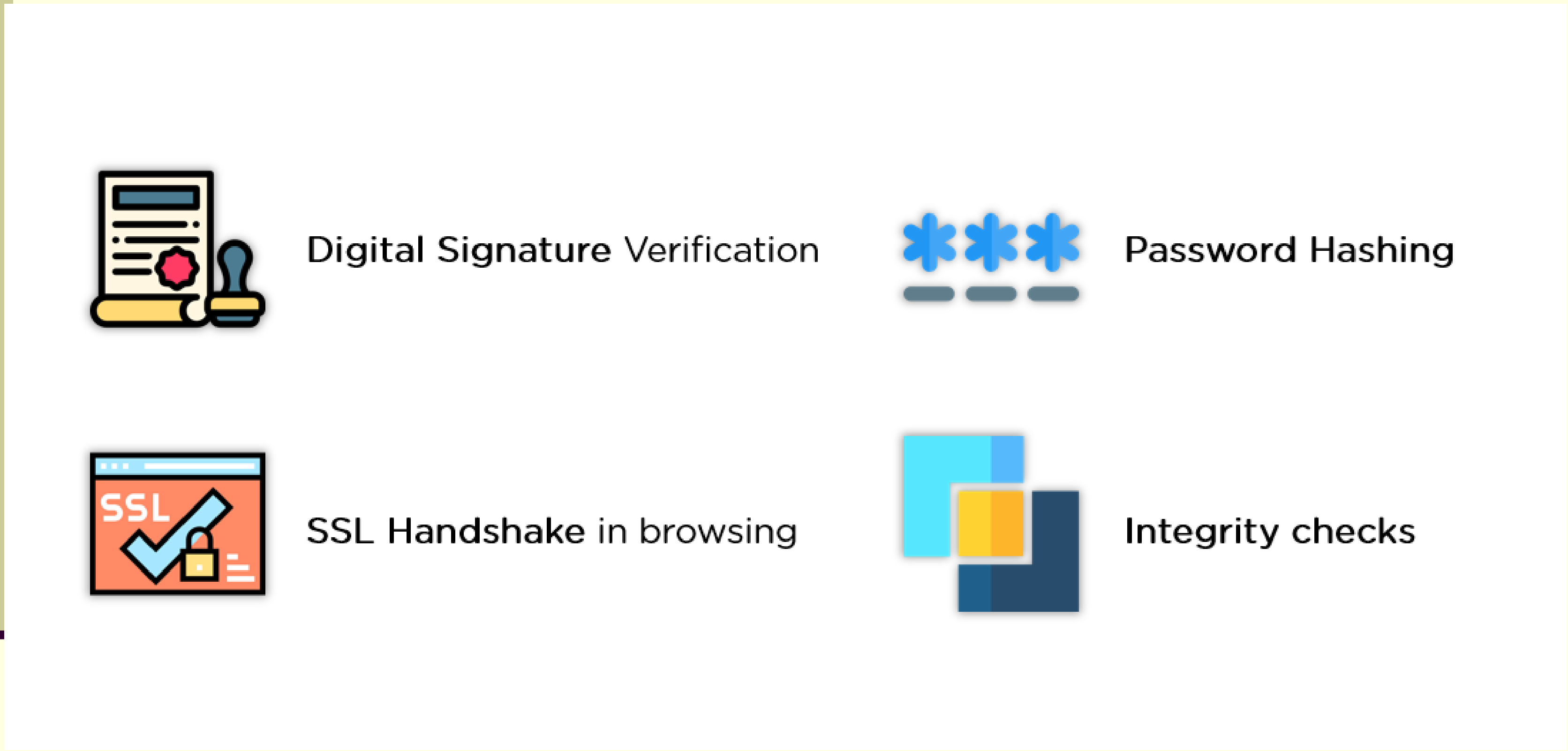
g = 0x1f83d9ab

h = 0x5be0cd19



value of K[i] in all rounds is pre-initialized, W[i] is another input that is calculated individually for each block, depending on the number of iterations being processed at the moment

SHA- 256 Applications



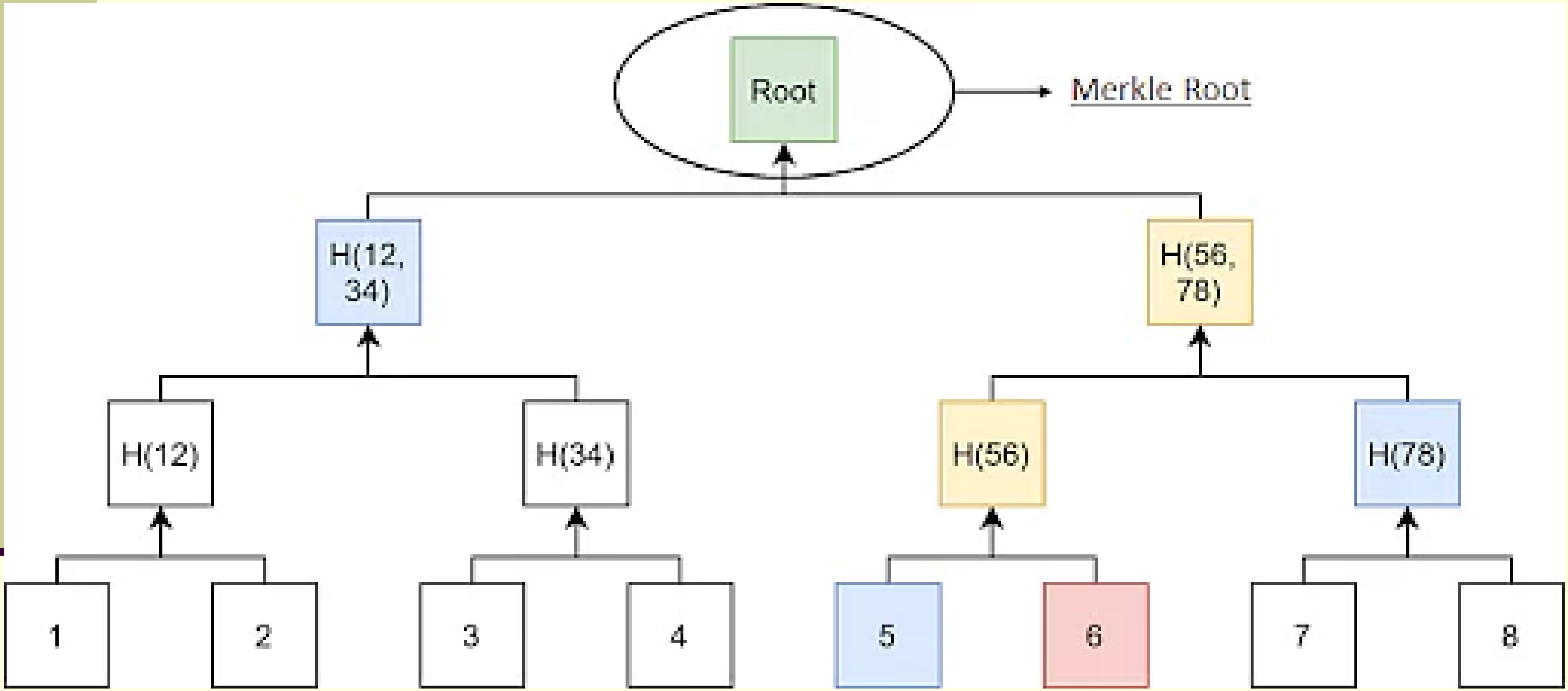
Merkle Tree

- fundamental part of blockchain technology
- mathematical data structure composed of hashes of different blocks of data, and which serves as a summary of all the transactions in a block
- allows for efficient and secure verification of content in a large body of data

Merkle Tree

- Both Bitcoin and Ethereum use Merkle Trees structure
- also known as Binary Hash Tree
- used to encrypt blockchain data more efficiently and securely
- enables quick and secure content verification across big datasets and verifies the consistency and content of the data

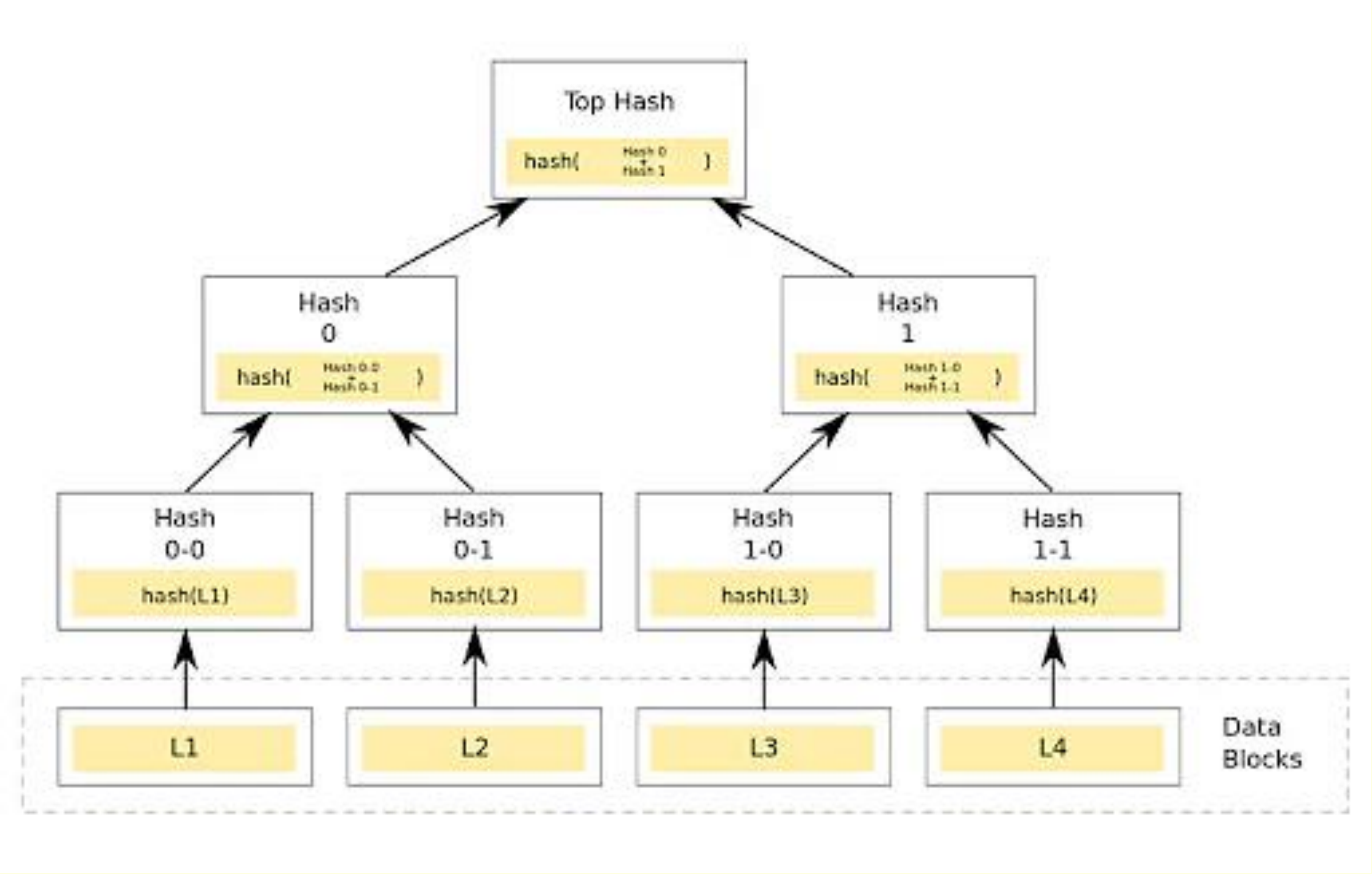
Merkle Root



Merkle Tree Working

- totals all transactions in a block and generates a digital fingerprint of the entire set of operations, allowing the user to verify whether it includes a transaction in the block
- Each non-leaf node is a hash of its previous hash, and every leaf node is a hash of transactional data

Merkle Tree



Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks

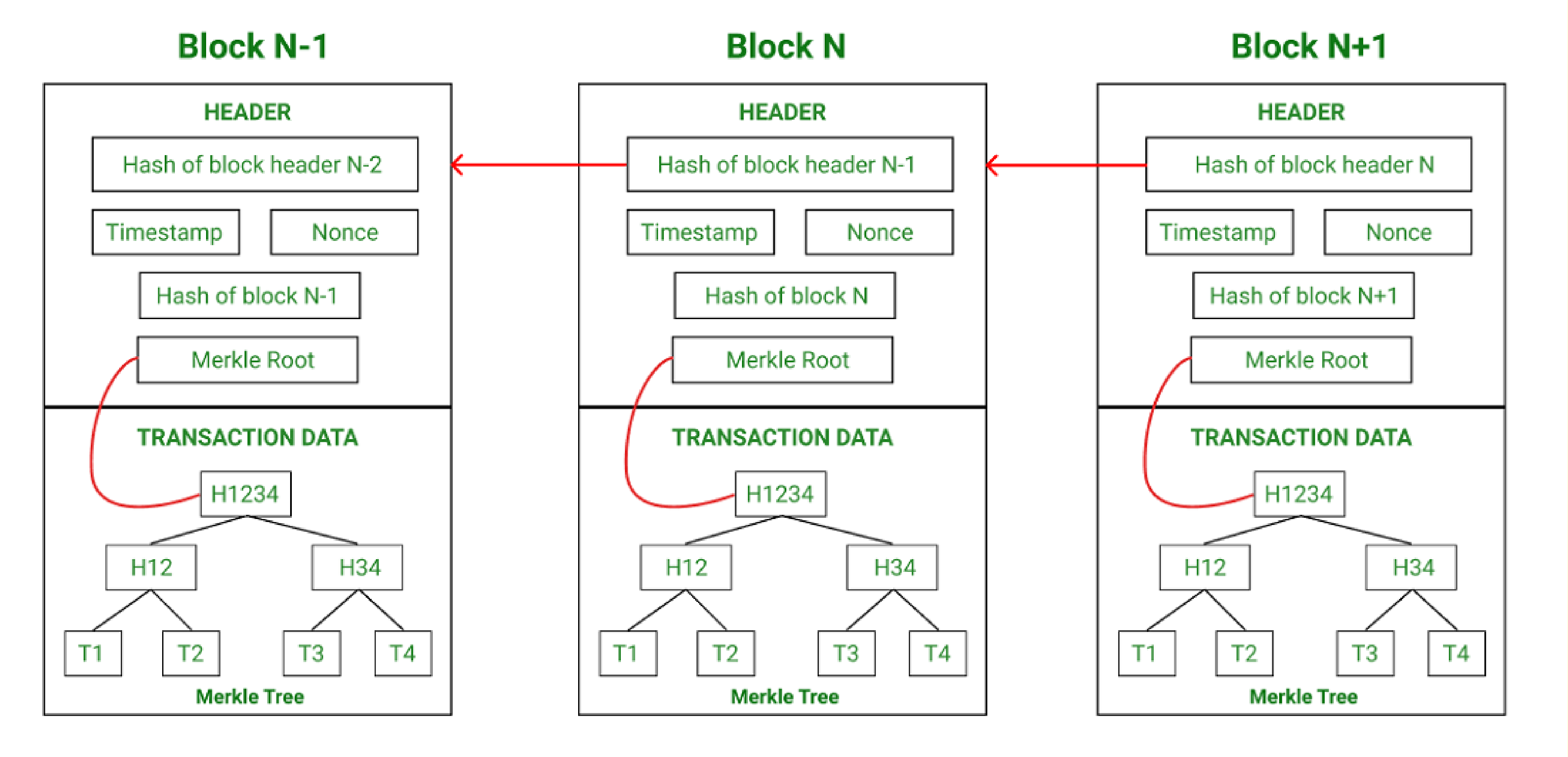
Merkle Tree Working

- Merkle Root is stored in the block header
- block header is the part of the bitcoin block which gets hash in the process of mining
- It contains the hash of the last block, a Nonce, and the Root Hash of all the transactions in the current block in a Merkle Tree

Merkle Tree Working

- having the Merkle root in block header makes the transaction tamper-proof
- As this Root Hash includes the hashes of all the transactions within the block, these transactions may result in saving the disk space

Merkle Tree Working



Merkle Tree Benefits

- Validates the data's integrity effectively
- Compared to other data structures, the Merkle tree takes up very little disk space
- can be broken down into small pieces of data for verification
- data format is efficient, and verifying the data's integrity takes only a few moments.

Distributed Hash Tables

- a decentralized data store based on key-value pairs
- Every node is responsible for a set of keys and their associated values
- The key is a unique identifier for its associated data value, created through a hashing function
- The data values can be any form of data

Distributed Hash Tables

- provide an easy way to find information in a large collection of data
- each node stores the key partitioning scheme so that if it receives a request to access a given key, it can quickly map the key to the node that stores the data
- It then sends the request to that node

Distributed Hash Tables

- nodes can be easily added or removed