CRYPTOGRAPHY

Index

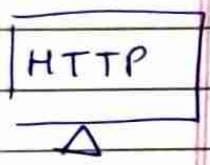
Moral of Story: See if Website is

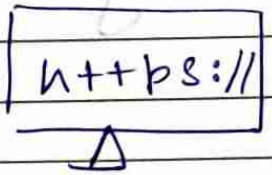
HTTP ✗ HTTPS ✓

Importance of CRYPTOGRAPHY

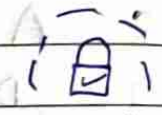
HTTP Websites are not secure and the data is visible to everyone trying to look for it.

HTTPS Websites are much more secure & the bank transactions are encrypted





Not Easy to steal



Not Secure.

1. What is CRYPTOGRAPHY?

Cryptography is the science of encrypting and decrypting information to prevent unauthorized access. The decryption process should be known to both the sender & the receiver.

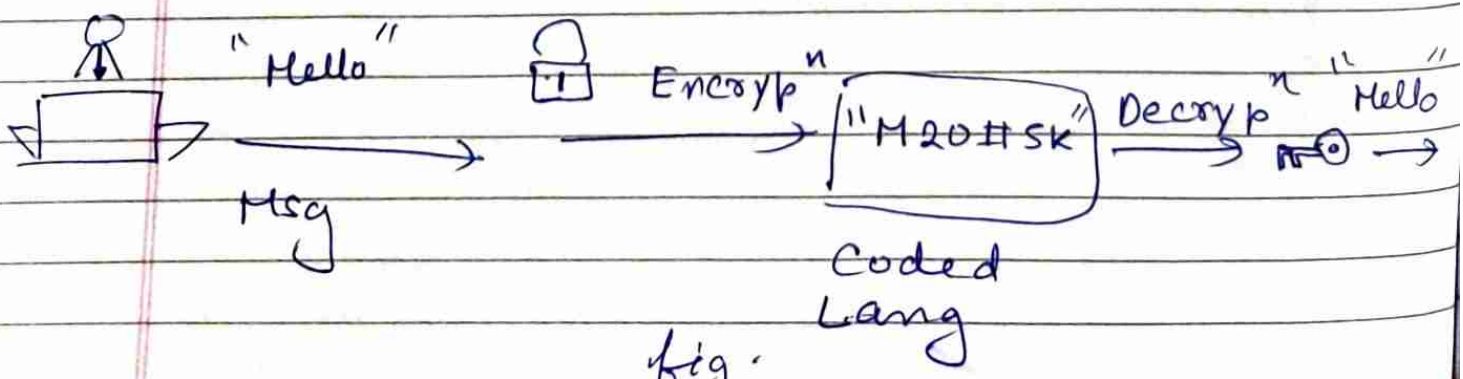
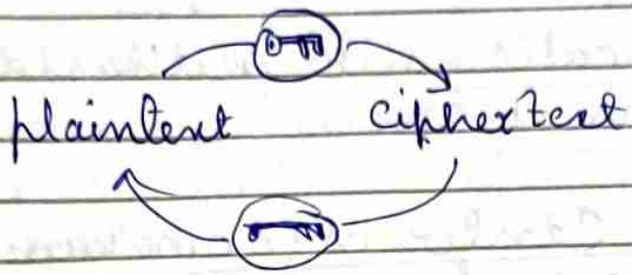


fig.

Encrypⁿ	Decrypⁿ
Making normal readable text difficult to understand.	Reversing the encrypⁿ process to retrieve normal message



Applicaⁿ :—

1. SSL / TLS Encryption
2. Digital Signatures
3. Safe Online Banking
4. Secure Chatting Services
5. Encrypted Emails
6. CryptoCurrency

Terms :

Cipher Ciphertext

Types of Encryption

(i) Symmetric key Cryptography

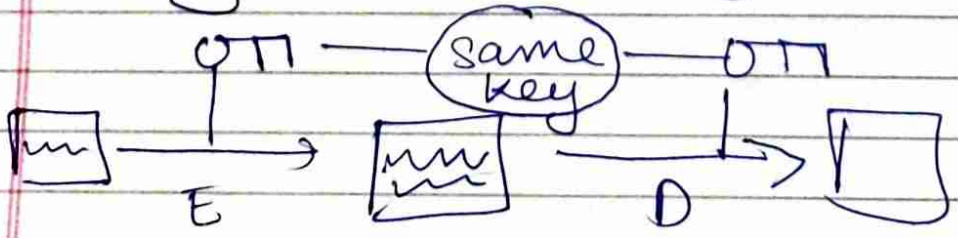
(ii) Asymmetric key Cryptography

Applicaⁿˢ of Symmetric key Crypto

- Banking applications to authenticate ID & transactions
- Secure / Data Center information can be encrypted at rest.
- HTTPS encryption with secure all around browsing.

What is Symmetric key Cryptography?

Symmetric key cryptography relies on a single key for encryption and decryption of informaⁿ. The key need to be kept decret & be available with both the sender & the receiver. Strength of encryption depends on the key size being used.



DES → old
AES → industry Standard.

Private key cryptography

- Same key for encryption & decryption means a single point of failure.
- Key needs to be always kept secret.
- Receiver / Third party can also generate message with the same key, so authentication issue will arise should the secret key is leaked.

Types of Encryption → Stream cipher

- Encrypt information one bit/byte at time
- Quick format of encryption
- Data is converted to binary digits and encrypted sequentially
- Popular algo → RC4, Salsa20

Binary Data + Encryp^n key → cipher text

Binary data → 10010101

Algo Fnc. → $S(x)$

Random → 01110010

E K → 10111001 → CT

2) Block Ciphers

- Informa" broken down to chunks /blocks of fixed size
- Size of block depends on key size
- The chunks are encrypted & later chained together
- Popular algorithms - AES, DES, 3DES

Binary Data

01001000 01100101 01101100 01011100

block1 block2 block3 Block4

Each block + Encrypted key → Encrypted Block.

B1 + B2 + B3 + B4

↓

[B1 B2 B3 B4 B5]

↓

Cipher text.

Advantages of Symmetric Key Cryptography

- Faster than Asymmetric
- Better performance Metrics
- Optimized for bulk amounts of data.

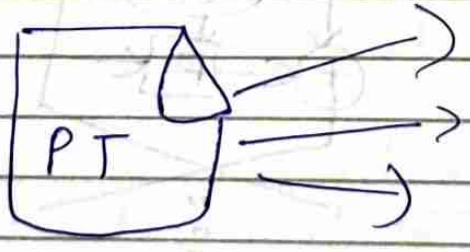
D1101111
Blocks

- Easier to set-up & implement

DES Encryption Algorithm

oldest

- Symmetric key algo.
- Stands for Data Encryption Standard
- Block size is 64 bits & key size is 48 bits
- Follows the feistal cipher structure



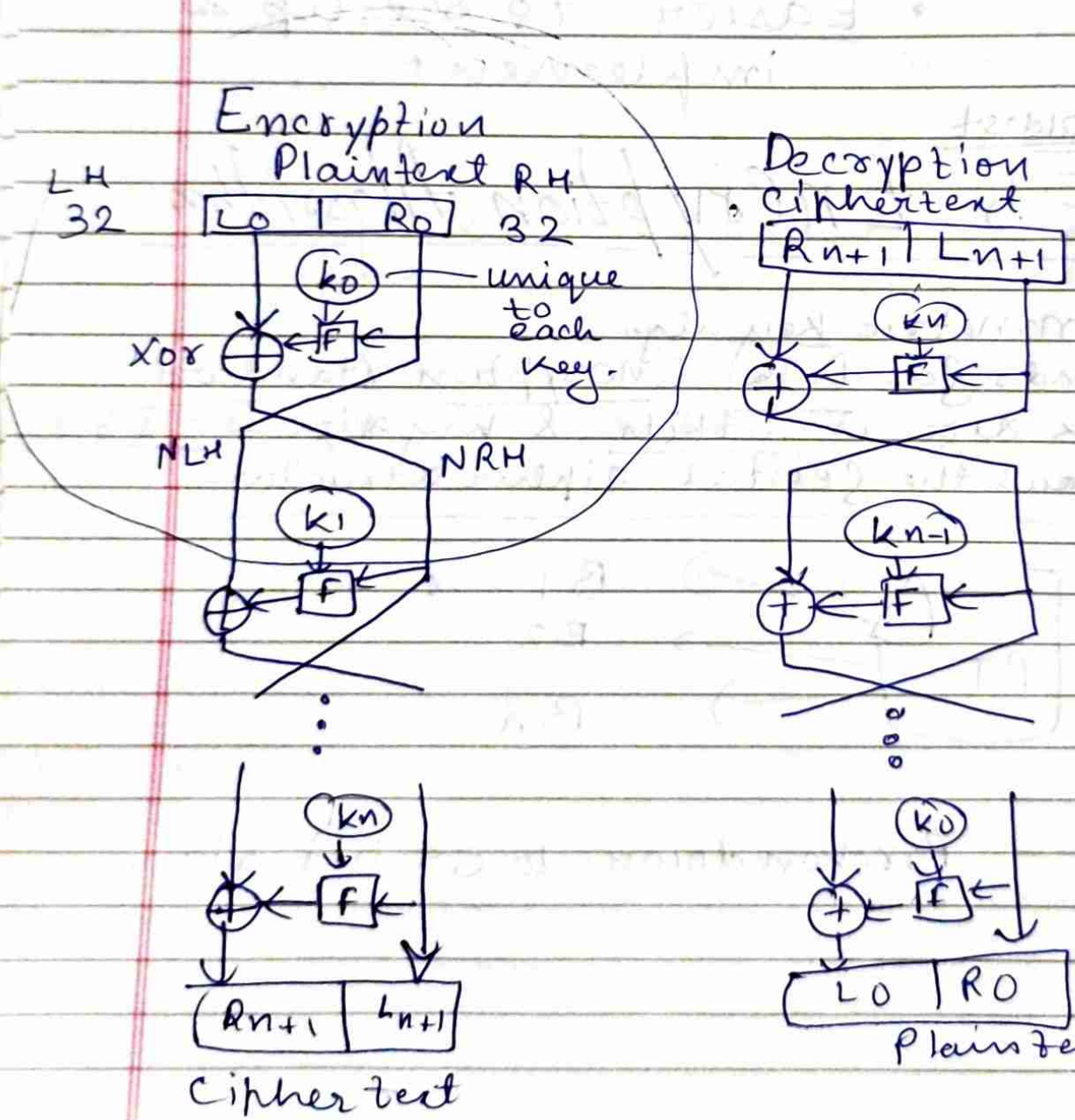
Broken down to 64 bit blocks

Origin of DES

Fiestal dev in 1971 → DES in 1976 → Triple DES introduced too slow → Rijndael algo 1998

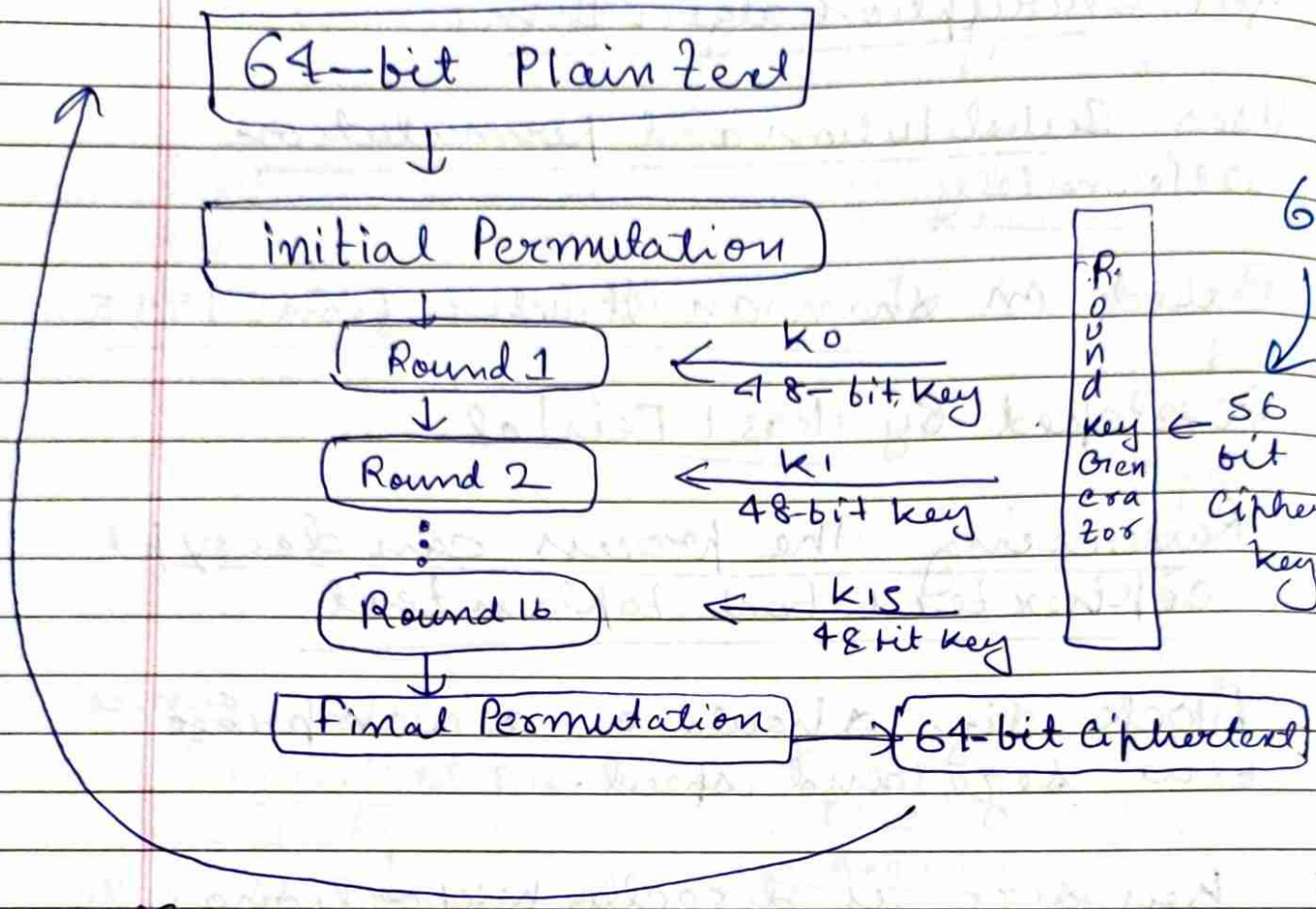
↓

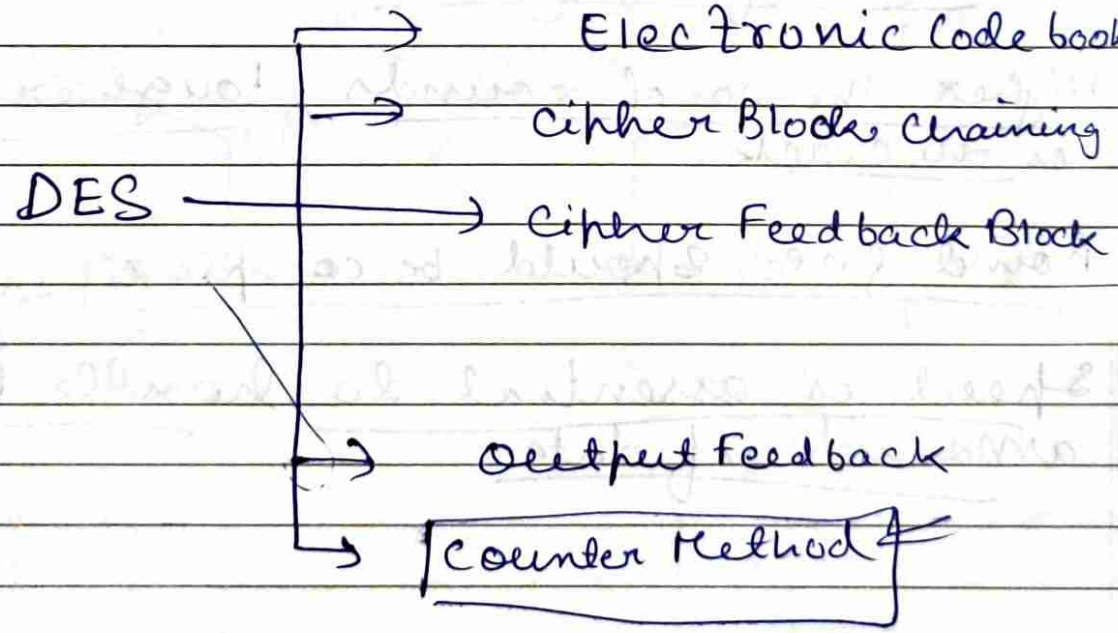
Replaced by AES in 2002



- Block Cipher that is used as a structure for encryption algorithms.
- Uses Substitution and permutation alternately
- Based on Shannon Structure from 1945
- Developed By Horst Feistel
- Reversing the process can decrypt ciphertext back to plain text.
- Block size should be a compromise b/w size and speed.
- Key size is directly proportional to strength of encryption.
- Higher the no. of rounds, tougher it is to crack.
- Round fnc should be complex
- Speed is essential to handle large amounts of data.

How DES Works - Key Generation





- Replaced by AES in 2002 as the world standard for encryption
- 56-bit key size easily broken by new generation computers
- withdrawn support for official purposes in 2005
- Triple-DES still allowed for important data till 2030

AES — Advanced Encryption Standard.

The AES algorithm (also known as the Rijndael algorithm) is a symmetric block cipher algorithm that takes a block size of 128 bits and converts them into ciphertext using keys of 128, 192 and 256 bits.
10 12 14

1. It uses (Substitution and Permutations) also called SP Networks.
2. A single key is expanded to be used in multiple rounds.

3. AES performs on byte data, instead of bit data.

4. No. of rounds is dependent on key length.

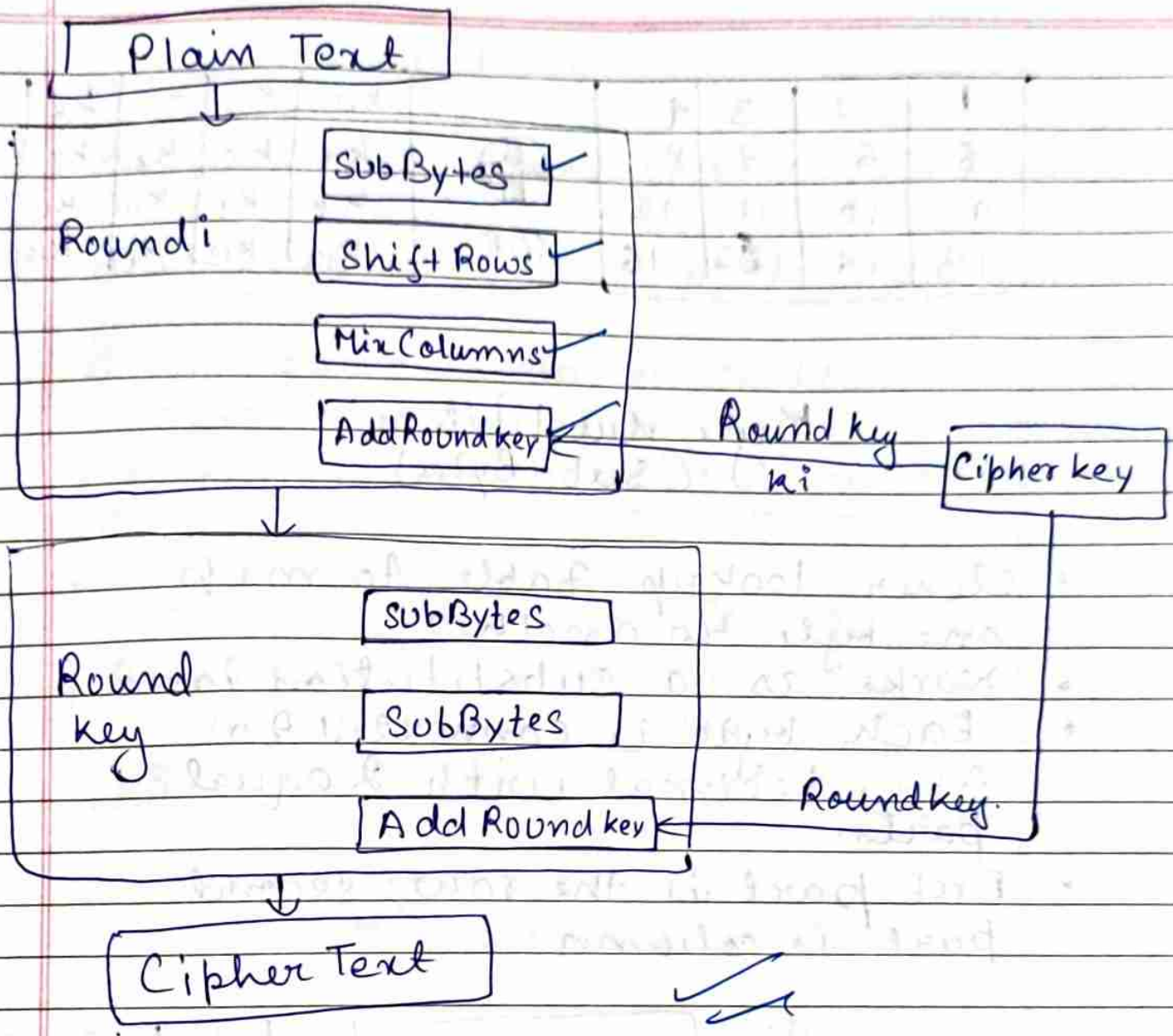
128 bit key length → 10 rounds
192 bit key length → 12 rounds
256 bit key length → 14 rounds

How Does AES Work?

Manner of Storage

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

- Everything is stored in a 4×4 matrix format.
- Known as state array.
- Each round takes state array as input and gives similar output.
- 16-byte matrix, with each cell representing one byte
- 4 bytes = 1 word, so each state array has 4 words.



Adding the Round key

- Plaintext is stored in a state array and XOR'd with k0.
- Performed only once per block.
- Will be performed once again at the end of each round.

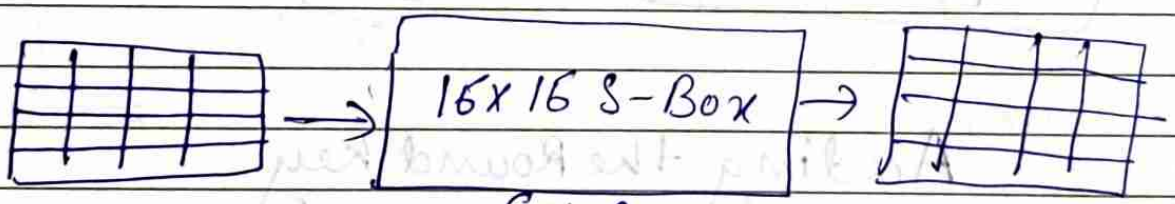
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

XOR

k_0	k_1	k_2	k_3
k_4	k_5	k_6	k_7
k_8	k_9	k_{10}	k_{11}
k_{12}	k_{13}	k_{14}	k_{15}

Byte substitution (Sub-Bytes)

- Clever lookup table to map one byte to another.
- Works as a substitution table
- Each byte is converted to hexadecimal with 2 equal parts.
- First part is the row, second part is column



Initial StateArray

Sub-Bytes

Shift Rows

- Shifting row elements among each other
- Increases complexity of the algorithm
- First row is to be skipped, second row moves 1 place, third row moves 2 places and last row moves 3 places

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→

1	2	3	4
6	7	8	5
11	12	9	10
16	13	14	15

Mix Columns

- Multiply each column with a constant matrix.
- Resultant matrix forms the new column
- Not to be done in the last round

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

X

C_0
C_1
C_2
C_3

=

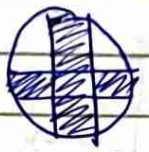
NC_0
NC_1
NC_2
NC_3

Constant Matrix Old Column New Column

Add Round key

- The expanded key is used with the resultant matrix.
- If its last round, the result is ciphertext.
- If not the last round, result is input for next round

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



XOR

k_0	k_1	k_2	k_3
k_4	k_5	k_6	k_7
k_8	k_9	k_{10}	k_{11}
k_{12}	k_{13}	k_{14}	k_{15}

Applications

- Wireless security against hackers
- General file Encryption
- Encrypted browsing sessions for better protection against hackers
- Processor security to prevent hijacking

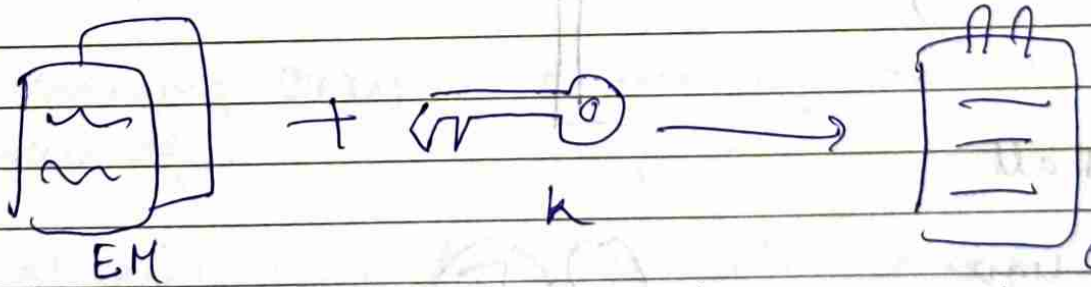
Diff

DES	AES
Key length – 56 bits	key length – 128 / 192 / 256 bits
Block size – 64 bits	Block size – 128 bits
Fixed no. of – 16 rounds	No. of rounds dependent on key length
Slower	Faster
old	New

The decrypⁿ key along with encrypted msg will reveal all the contents & Ryan will lose his privacy

Joe

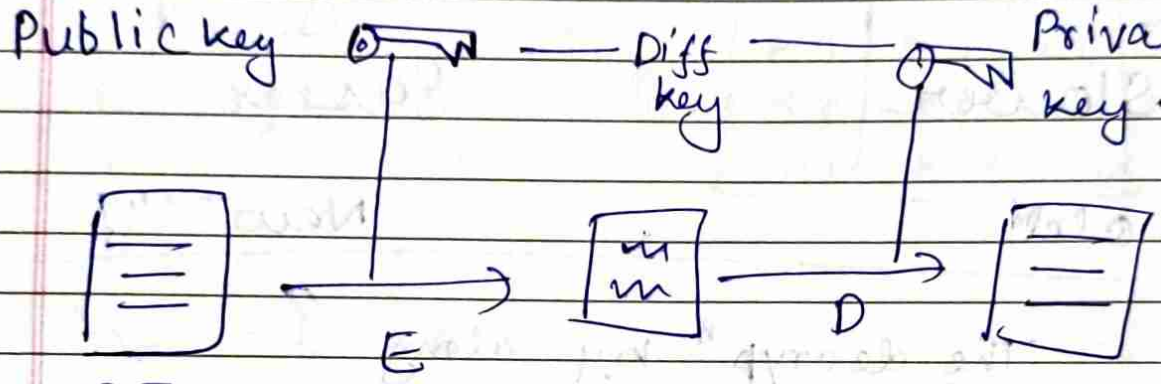
Ryan

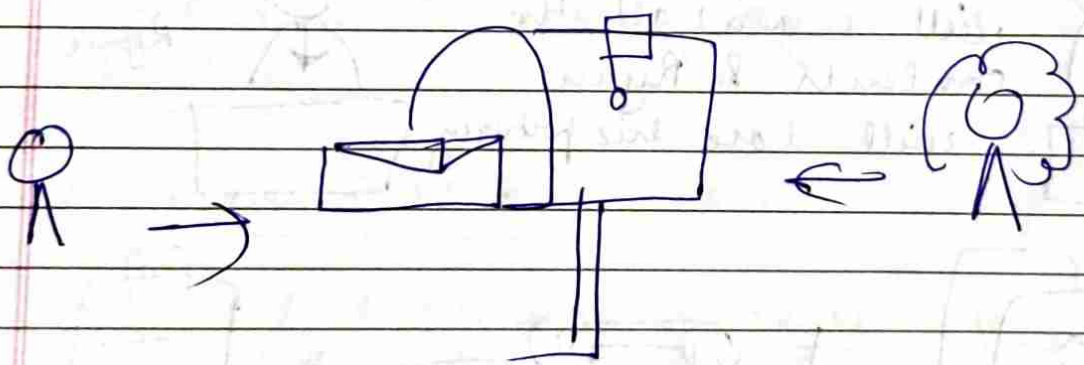


Sol: Asymmetric key Cryptography

What is Asymmetric key CRYPTOGRAPHY?

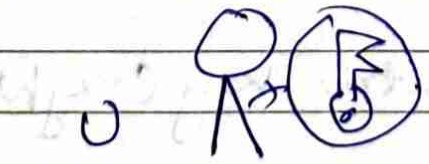
Asymmetric key Cryptography uses two different keys for encryption and decryption. The key used for encryption is the public key and the key used for decryption is the private key.

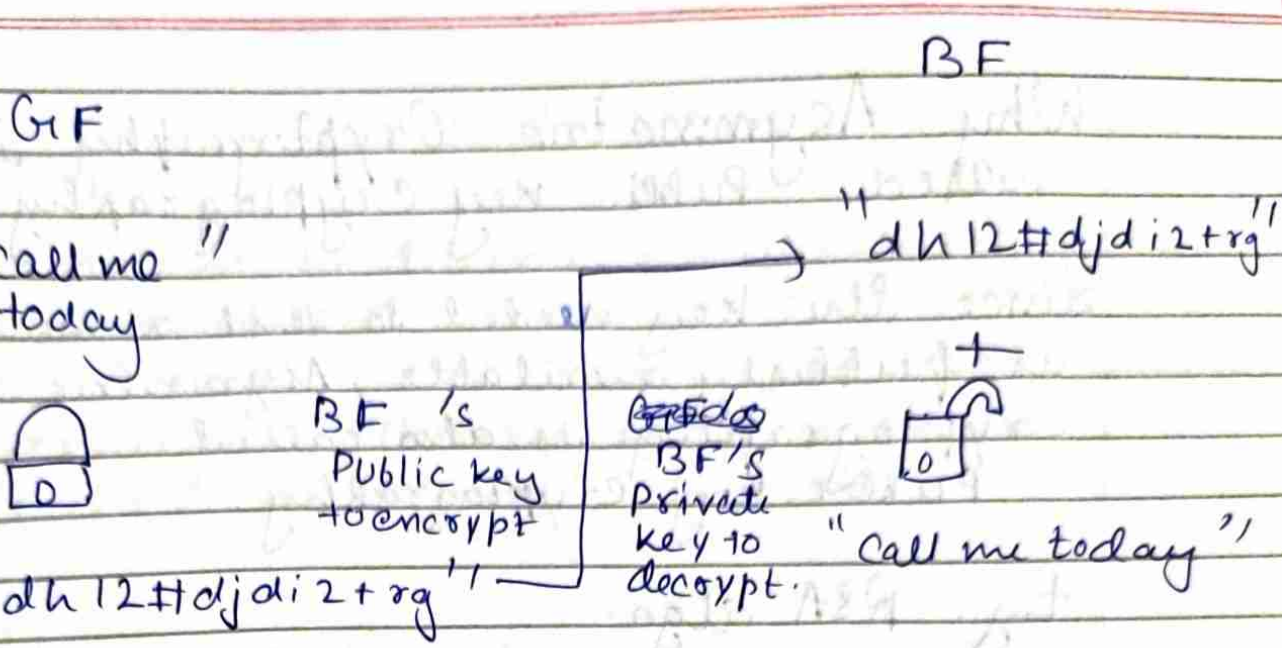




address all

key — you only





Applications:

: Digiatal Signatures to maintain authenticity of documents.

: Encrypted browsing sessions for better protection against hackers

: Managing crypto-currency transactions Securely.

: Sharing keys for Symmetric key cryptography

Why Asymmetric Cryptography is called Public key cryptography

Since the key needed to send a message is publicly available, Asymmetric cryptography is also called Public key cryptography.

Eg: RSA algo.

- Based on Asymmetric algorithm
- Designed by Ron Rivest, Adi Shamir, Leonard Adleman
- Most valuable when encrypting data in transit.
- Used with VPN services, email transfer, messaging applications, etc

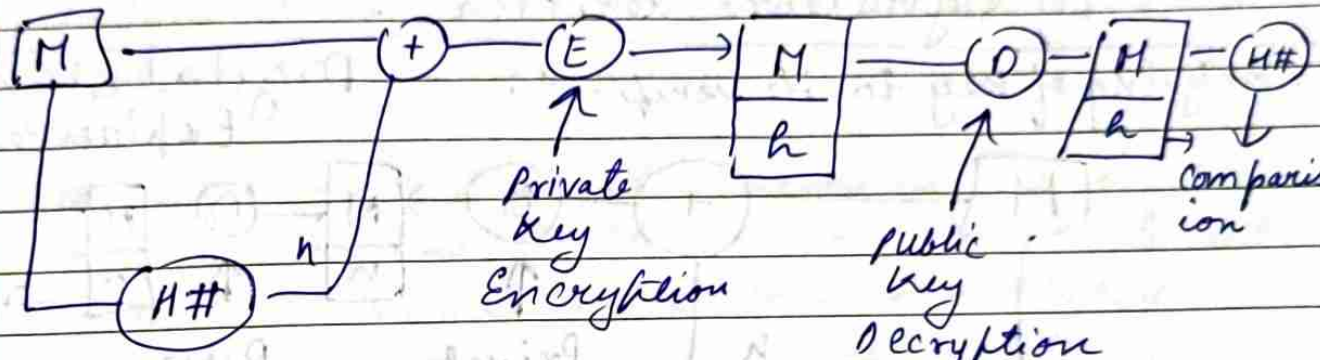
Advantage over SKC:

- No need of sharing secret keys
- Proof of owner's authenticity
- Longer key lengths mean stronger encryption.
- Data can't be modified in transit.

DSA

Digital Signature

- Mechanism to determine authenticity of a document file
- Uses public key cryptography mechanism
- Helpful to authenticate long distance official communicaⁿ channels.



M – Plaintext
h – digest

DS
Implementaⁿ

DSA RSA.

Benefits

(I) Message Authenticaⁿ

(II) Integrity verificaⁿ

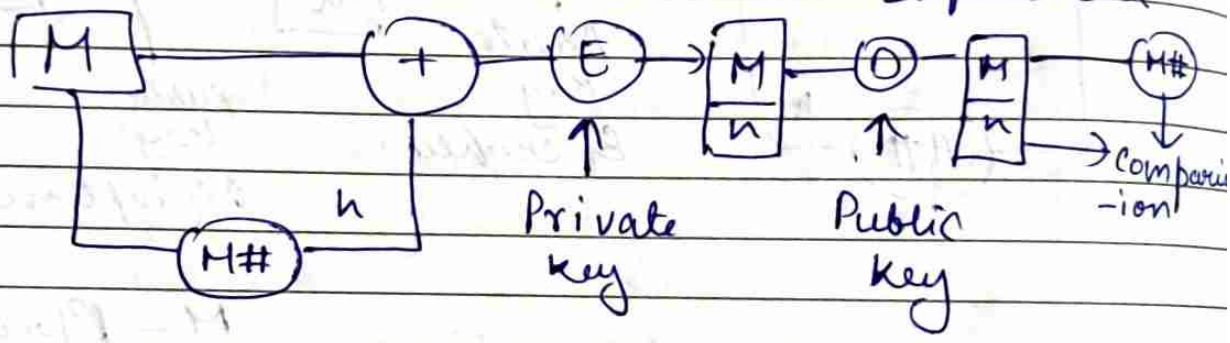
(III) Non Repudiation.

What is DSA?

- Federal Information Processing Standard for digital signatures.
- Proposed in 1991, Standardized in 1994.
- National Institute of Standards & Technology made it royalty free.
- Covers the process from key generation to signature verificaⁿ.

from signing of key to its verificaⁿ.

Digital Signature Explained



M – Plaintext
H – Hash function
h – Hash digest
'+' – Bundle both plaintext and digest
E – Encryption
D – Decryption

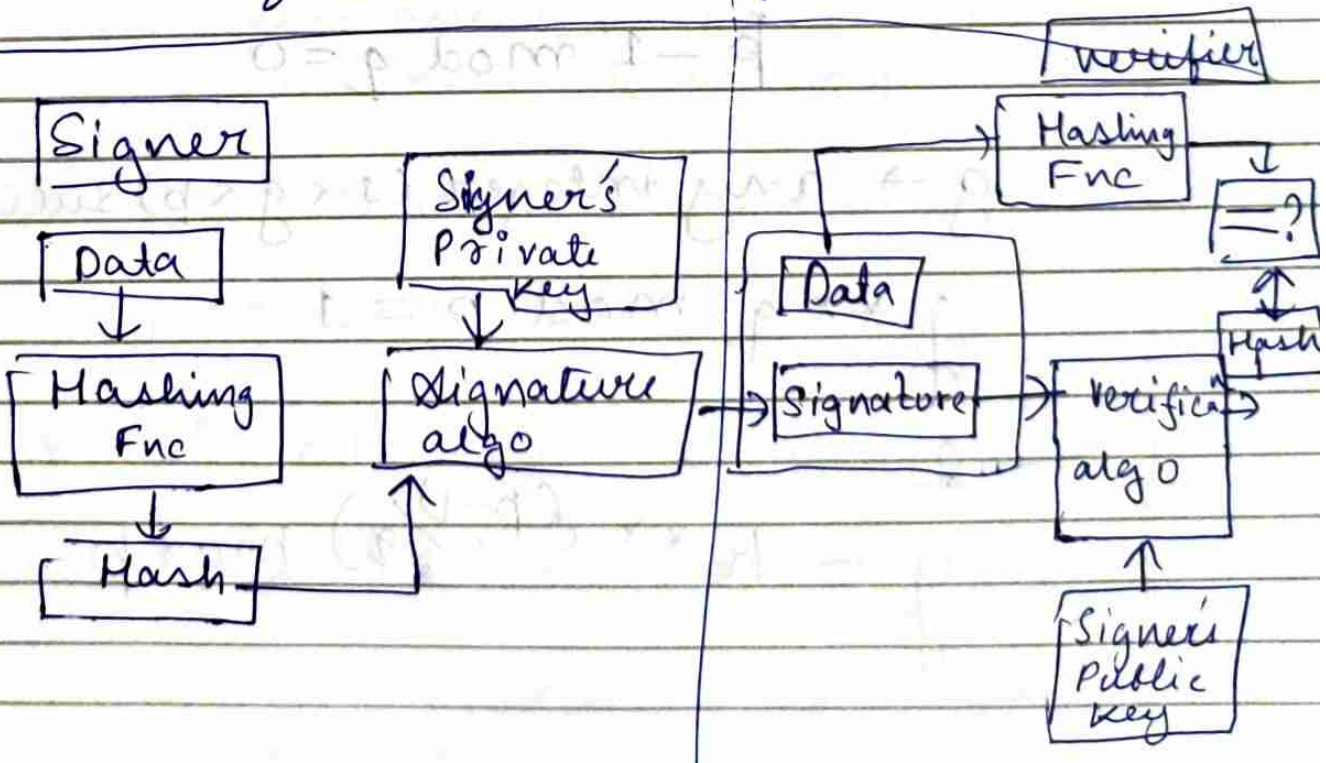
Step 1 : M, the original message is first passed to a hash function denoted by H# to create a digest.

Step 2: Next, it bundles the message together with the hash digest h and encrypts it using the sender's private key.

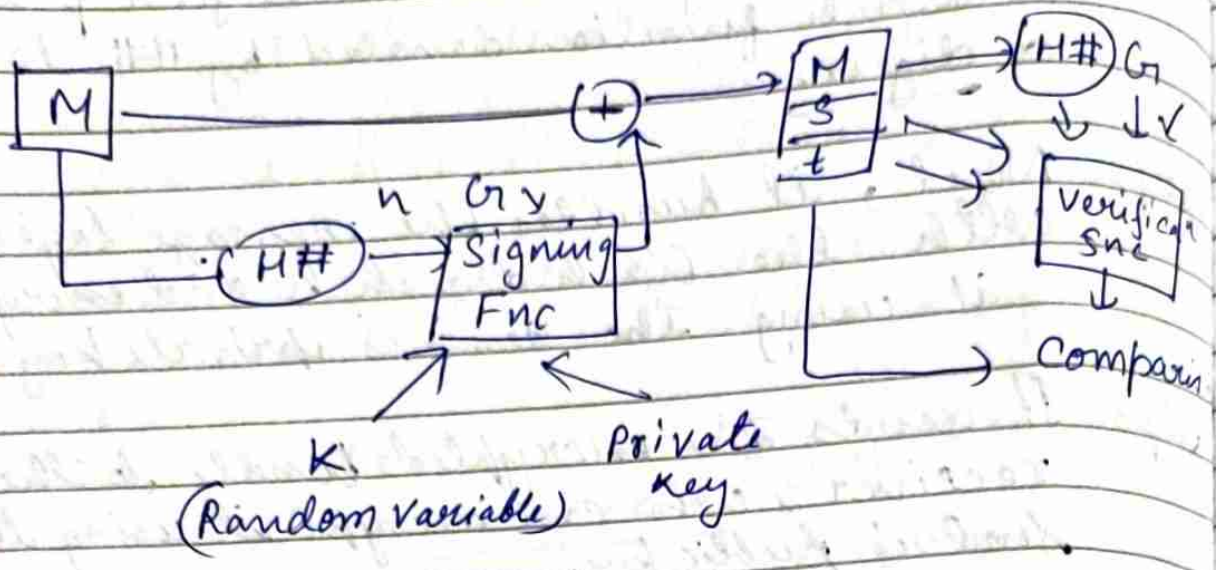
Step 3: It sends the encrypted bundle to the receiver, who can decrypt it using the sender's public key.

Step 4: Once it decrypts the message, it is passed through the same hash function (H#), to generate a similar diglst.

Step 5: It compares the newly generated hash with the bundled hash value received along with the message. If they match, it verifies data inligrity.



DSA Algo Diagram:



(3)

Step 1: Key Generation

1. Pre-requisites for the key generation formulas:

q → Prime Divisor

p → prime number, such that

$p-1 \bmod q = 0$

g → any integer $(1 < g < p)$ such that

$g^{**}q \bmod p = 1$

&

$g = h^{**}((p-1)/q) \bmod p$

Public key

- x (private key) → random integer such that : $0 < x < q$
- y (public key) can be calculated as : $y = g^x \bmod p$
- private key can be packaged as : $\{p, q, g, x\}$
- Public key can be packaged as : $\{p, q, g, y\}$

on.

Step 2 : Signature Generation

1. Message is passed through a hash function to generate a digest (h).
2. Choose any random integer k such that : $0 < k < q$
3. To calculate the value of r :

   $(g^k \bmod p) \bmod q$.

4. To calculate the vale of s :

   $[k^{-1}(h + x \cdot R) \bmod q]$

The signature can be packaged as $\{r, s\}$

Step 3 : Signature Verification

1. Calculate the message digest using same hash function.
2. Compute the value of w such that:
   $s * w \bmod q = 1$.
3. Compute the value of $v1$ as
   $v1 = h * w \bmod q$
4. Compute the value of $u2$ as
   $v2 = r * w \bmod q$
5. Finally, the verificaⁿ component v.
   $v = [(g^{v1} \cdot y^{v2}) \bmod p) \bmod q]$

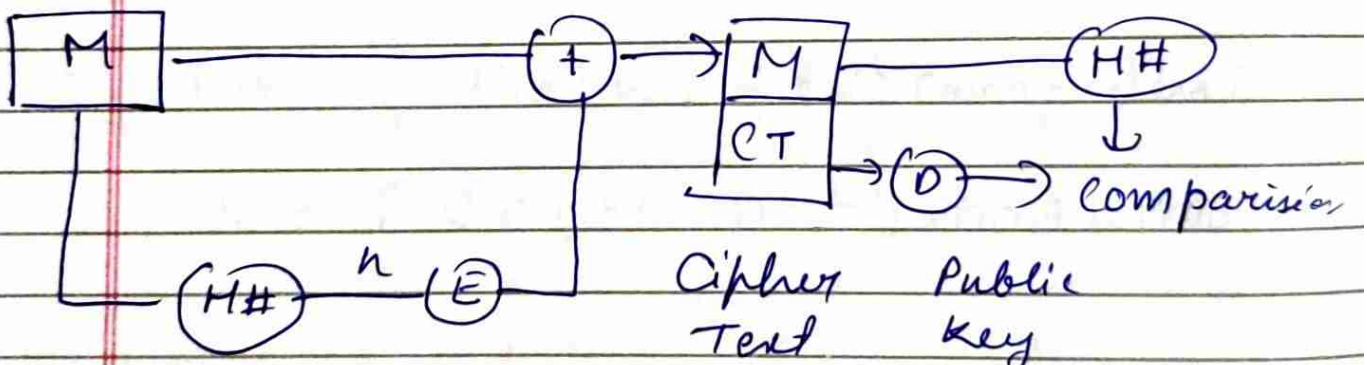
If $v == r$, the signature verificaⁿ is successful.

Advantages of DSA

- Highly Robust Standard
- Better speed in key computaⁿ

What is RSA?

- Rivest – Shamir – Adleman algorithm, named after its 3 founders.
- First Published in 1977
- Along with signature verification, it can be used for encryption and decryption of standard Data.
- Below figure is the process of verifying signatures using RSA



RSA in Data Encryption

- Key scope is reversed
- Public key of receiver is used to encrypt data.
- Private key of receiver is used to decrypt data
- Key exchange not necessary.

Two main components
→ Key generation
→ Encryption / Decryption Fnc.

Key Generation

1. Two large prime numbers are chosen (p & q)
2. compute $n = p * q$ and $z = (p-1)(q-1)$
3. Choose a no. e where $1 < e < (p-1)(q-1)$
4. A no. d is selected so that $ed \bmod z = 1$ & calculated as $d = e^{-1} \bmod (p-1)(q-1)$
5. Public key is (n, e) & Private key is (n, d)

Encrypⁿ and Decrypⁿ

If plaintext is m, encrypted ciphertext c is calculated as

$$c = m^e \bmod n$$

~ the plain text m can be calculated as

$$m = c^d \bmod n$$

Ex.

1. $p = 7$, $q = 13$

$$n = p * q = 91$$

2. $1 < e < (p-1)(q-1)$

$$\left\{ \begin{array}{l} 1 < e < 6 \times 12 \\ 1 < e < 60 \end{array} \right\}$$

let $e = 5$

a mod b

calcia ÷ b → with this ☐ x b

3. value of d.

$e \times d \bmod z = 1$

$5 \times d \bmod 91 = 1$

$d = e^{-1} \bmod (p-1)(q-1) = 29$

4. Public key = (91, 5), private key = (91, 29)

5. Let plaintext m be 10.

Ciphertext (c) = $m^e \bmod n = 82$

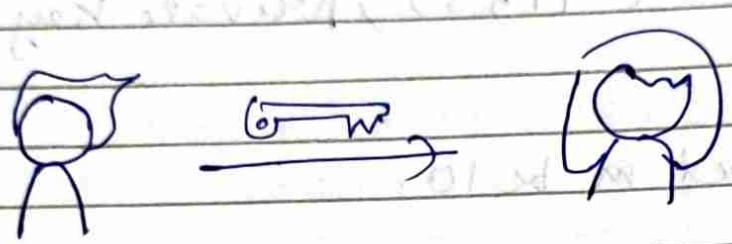
Plaintext = $c^d \bmod n = 10$

Advantages of RSA

- No need of sharing secret keys
- Proof of owner's authenticity.
- Faster Encryption than DSA
- Data can't be modified in transit.

Diffie Hellman Key Exchange Algorithm.

Story: Henry & Stella want to exchange messages securely using encrypted messages.



Prob': sending this key directly makes it easy for hackers to capture it in transit.

Sol^n: This is where the Diffie Hellman Key Exchange algorithm can help us in securely exchanging keys.

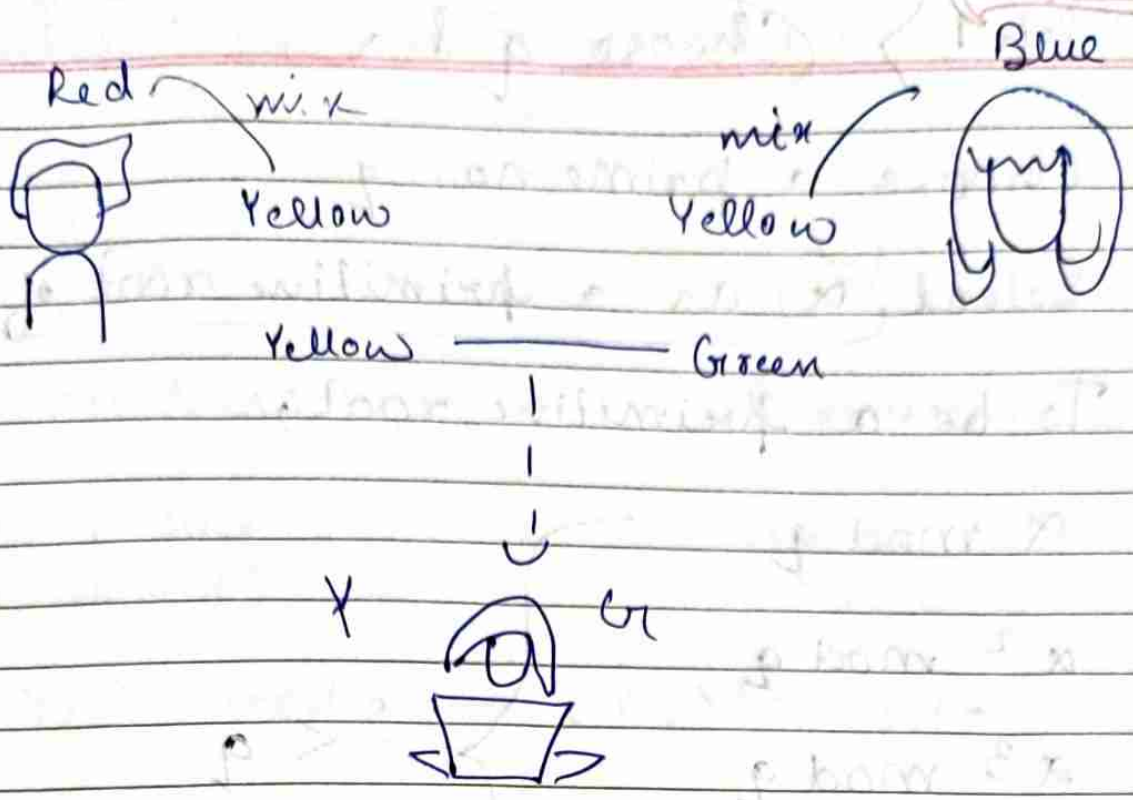
Diffie Hellman Key Exchange:

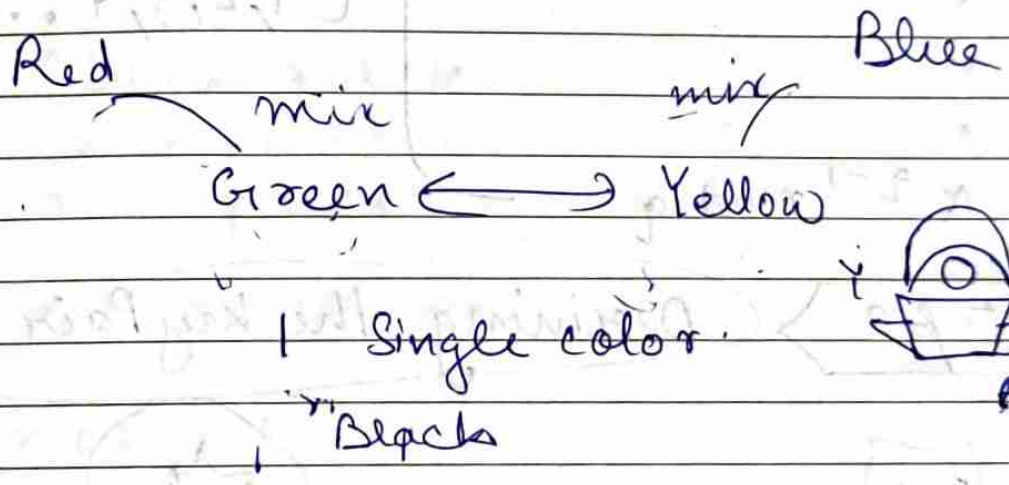
→ Algo to securely exchange keys

→ Proposed in 1976 by Whitfield Diffie and Martin Hellman

→ Communicated over "insecure" channel

→ Used as a precursor to asymmetric cryptographic algorithm





Applicaⁿˢ:

- Public key infrastructure.
- SSL/TLS Handshake
- SSH Secure Shell Access

Step 1 — Choose q & α

a. Choose a prime no. q

b. Select (α) as a primitive root of q

To be a primitive root,

$\alpha \bmod q$

$\alpha^2 \bmod q$

$\alpha^3 \bmod q$

.

.

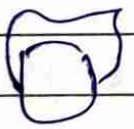
.

$\alpha^{q-1} \bmod q$

$< q$

$(1, 2, 3, 4 \ldots, q-1)$

Step 2 — Deriving the key Pair

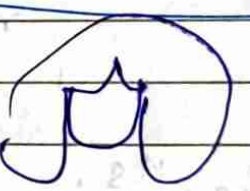


Assume private key = X_a where $X_a < q$

Public key (Y_a) becomes

$Y_a = \alpha^{X_a} \bmod q$

key pair : $\{X_a, Y_a\}$



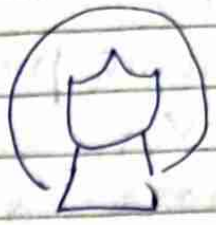
Assume private key = X_b where $X_b < q$

Public key (Y_b) becomes

$Y_b = a^{X_b} \bmod q$

key pair : $\{X_b, Y_b\}$

Step 3 Key Generation.



Parameters: X_a, Y_b, q

Secret key generated

Parameters: X_b, Y_a, q

Secret key generated

$$(Y_b)^{X_a} \bmod q = (Y_a)^{X_b} \bmod q$$

Q.

I: Choose q & α

a. Choose a prime no. $q = 17$

b. α as a primitive root of q, $\alpha = 3$.

To be primitive root

$3 \bmod 17 = 3$

$3^2 \bmod 17 = 9$

$3^3 \bmod 17 = 10$

.

.

.

$3^{16} \bmod 17 = 1$

< 17

II. Deriving the key pair

Boy

Assume private key $= X_a$
where $X_a = 15$

Public key (Y_a) becomes

$Y_a = 3^{15} \bmod 17 = 6$

key pair : {15, 6}

Girl

Assume private key $= X_b$ where $X_b = 13$

Public key (Y_b) becomes

$Y_b = 3^{13} \bmod 17 = 12$

key pair : {13, 12}

III. Key Generation

Boy

Parameters : X_a, Y_b, q

Secret key generated

$k = 12^{15} \bmod 17 = 10$

Girl

Parameters : X_b, Y_a, q

Secret key generated

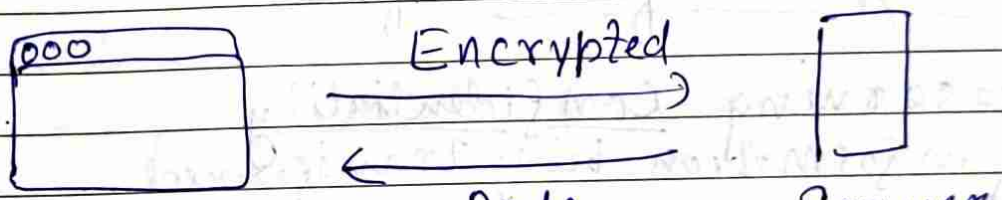
$k = 6^{13} \bmod 17 = 10$

SSL Hand Shake

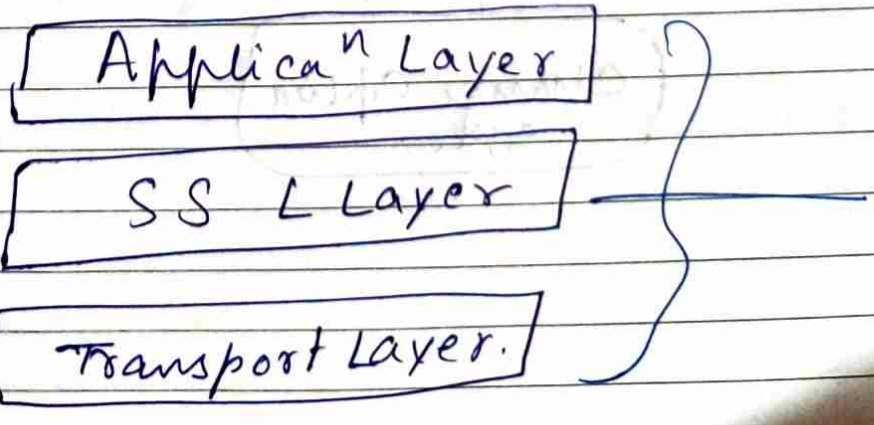
Moral: To preserve our data online from Hackers, SSL handshake was introduced.

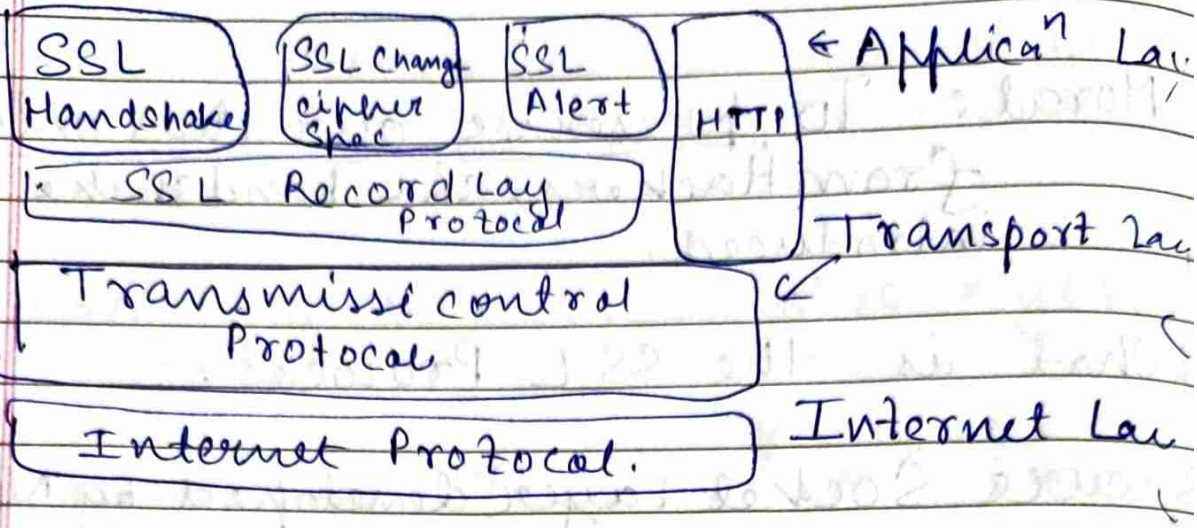
What is the SSL Protocal.

- Secure Socket layer developed by Netscape in 1995
- Security protocol to provide security & privacy
- used to encrypt information b/w a client and a server.

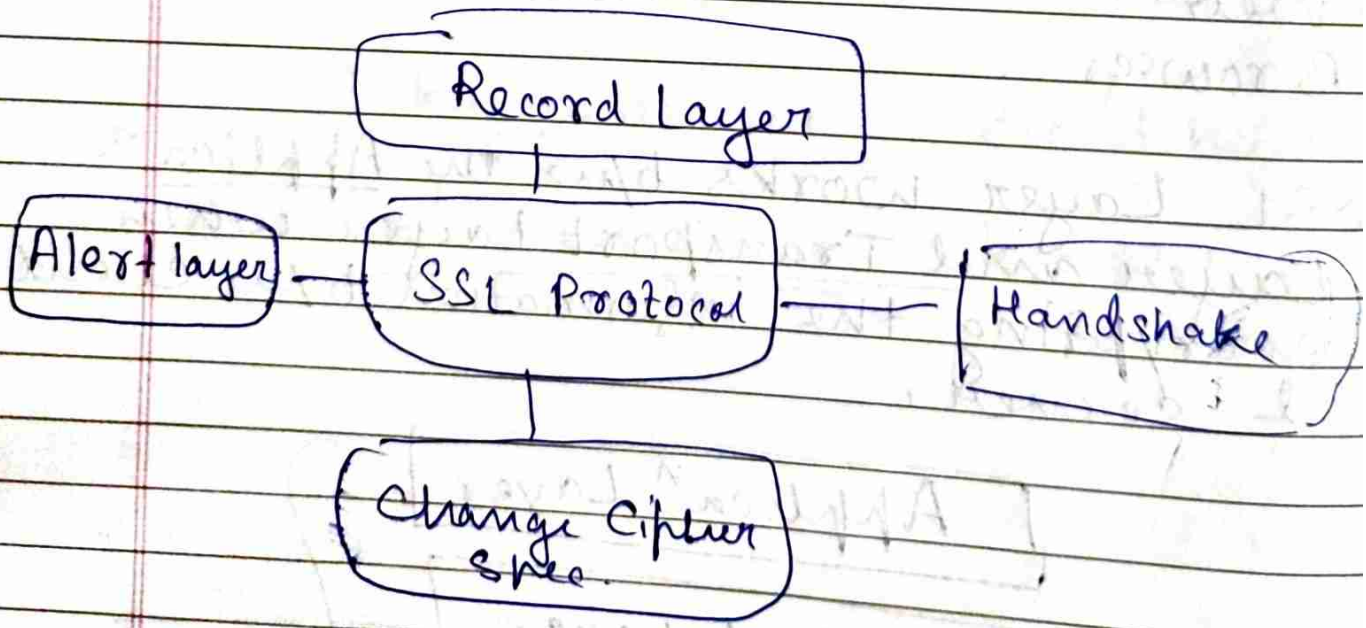


SSL Layer works b/w the Applicaⁿ Layer and Transport Layer when encrypting the information b/w client & server.





- Authenticity via securing the right client & Server connection
- Ensuring Integrity of data by encryption of data flow
- Preserving Confidentiality of information being transferred.



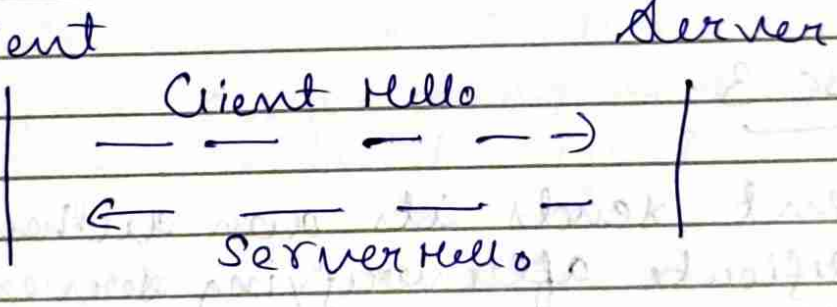
To Which layer is the data passed to after being encrypted in SSL?

Ans: Transport Layer.

Steps in SSL

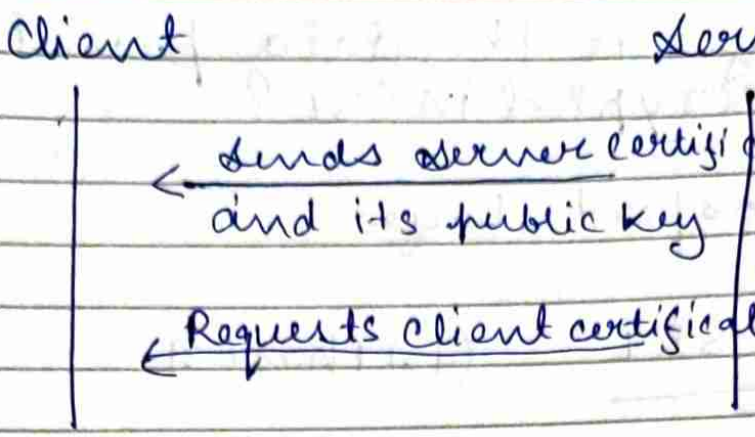
(1.) Phase 1

- Client & Server get acquainted with a hello signal each.
- Client sends SSL version, cipher suite, session ID, etc.
- Server returns a common encryption algorithm chosen from the cipher suite and compression algorithm.



(2.) Phase 2.

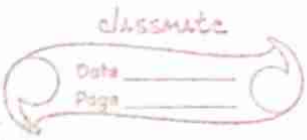
- Server sends its own authentication certificate and requests for client authentication
- Server also send its own public encryption key.
- The phase with a 'Server hello done' message.

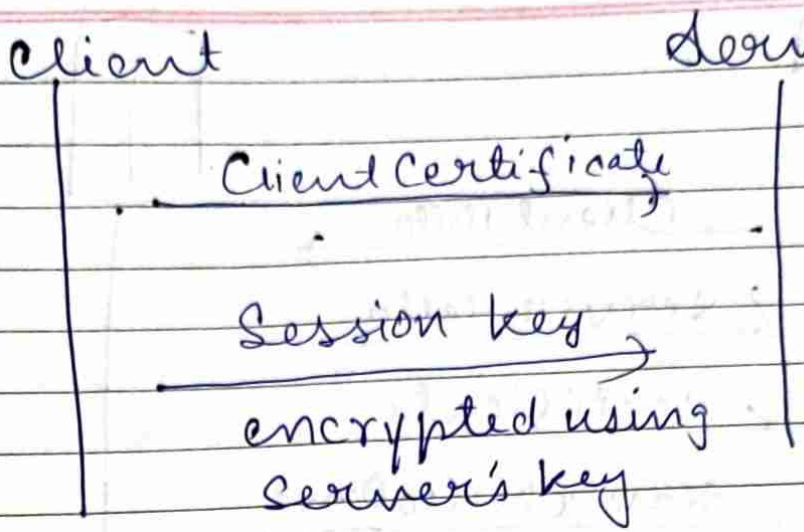


- Client sends the states of the cipher functions along with a finished message to end the handshake from its side.
- Server.

③ Phase 3.

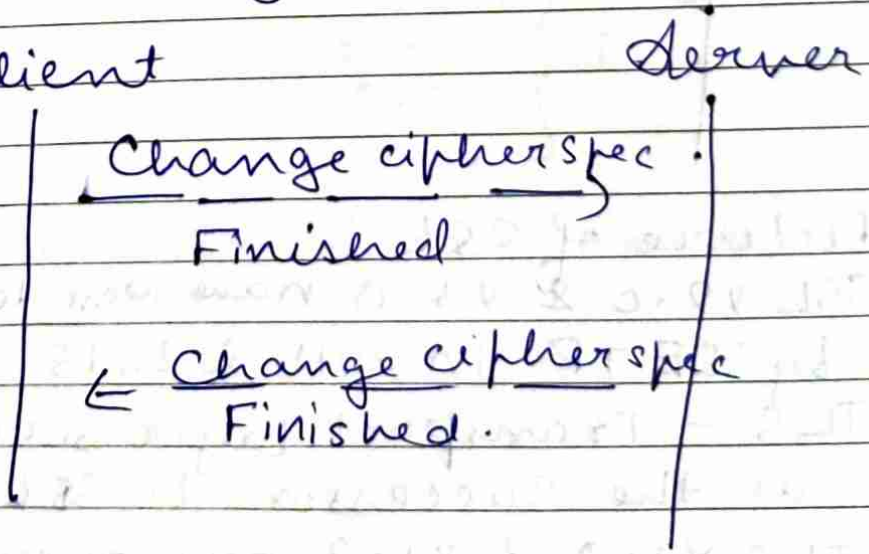
- Client sends its own authentication certificate after verifying server with respective certificate authorities (CAs)
- Client also send a secret private key encrypted using the server's previously received public key

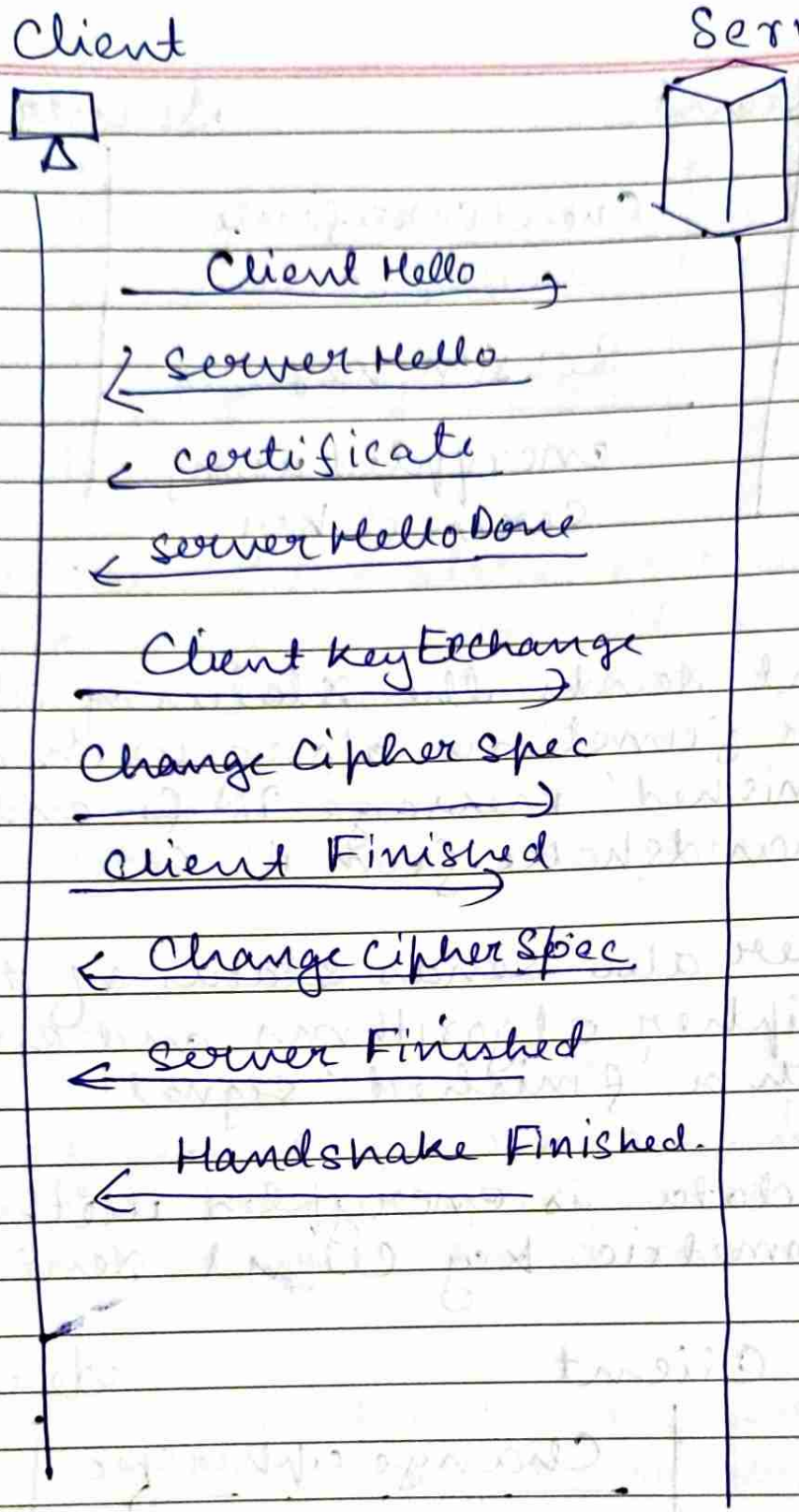




(4.) Phase 4

- Client sends the status of the cipher functions along with a 'finished' message to ~~f~~ end the handshake from its side.
- Server also sends status of the & cipher algorithms and ends with a 'finished' signal.
- The data is encrypted with the Symmetric key client sent in Ph 3.





Future of SSL

1. SSL V2.0 & V3.0 have been deprecated by IETF in 2011 & 2015, respectively
2. TLS - Transport Layer Security is the successor to SSL protocol
3. TLS V1.2 & V1.3 are now the global standards for securing internet traffic from client to server