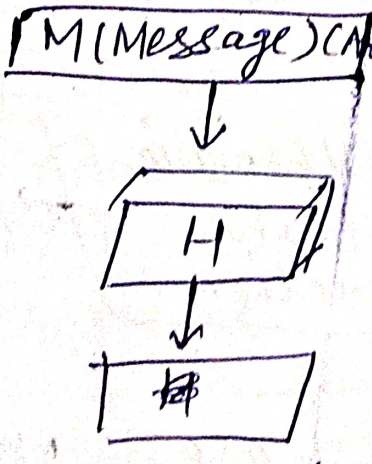
DETAILED LECTURE NOTES

Hash-functions –

A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length. Values returned by a hash function are called Message Digest or simply hash values.



Features of Hash functions -

The typical features of hash functions are

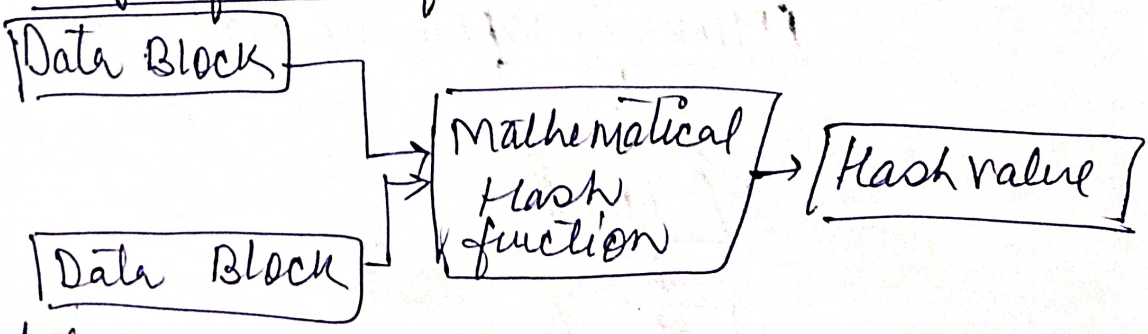
i) fixed length output → Hash function converts data of arbitrary length to a fixed length. This is known as hashing the data.

ii) The hash is much smaller than the input data, so it is called compression function.

iii) A hash is a smaller representation of a larger data, known as digest.

iv) Hash function with n bit output is referred to as an n-bit hash function.

Design of Hash function -



Popular hash functions are - MD5, SHA, RIPEMD (Race Integrity Primitives Evaluation Msg Digest)

POORNIMA
COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

PAGE NO.

Requirements and security

The generally accepted requirements for a cryptographic hash functions are shown in table

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed Output size	H produces a fixed length output
Efficiency	H(x) is relatively easy to compute making both H/W & S/W implementations practical.
Preimage resistant (one-way property)	for any given hash value h, it is computationally infeasible to find y such that H(y) = h.
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find $y \neq x$ with H(y) = H(x).
Collision resistant	It is computationally infeasible to find any pair (x, y) such that H(x) = H(y)
Pseudorandomness	output of H meets standard tests for pseudorandomness.

- The preimage resistant is the one way property. it is easy to generate a code given a message. This property is important if the authentication technique involves the use of a secret value.
- The second preimage resistant guarantees that it is impossible to find an alternative message with same hash value as given in message.
- The collision resistant is referred as a strong has function. It protects against an attack in which one party generates a message for another party to sign.
- The pseudorandomness requirement has not traditionally been listed as a requirement of cryptographic has functions but it is more or less implied.

POORNIMA

COLLEGE OF ENGINEERING

LECTURE NOTES

Campus: ... Course: ... Class/Section: ... Year: ...

Name of Faculty: ... Name of Subject: ... Code: ...

Date (Prep): ... Date (Del): ... Unit No./Topic: ... Lect. No.: ...

OBJECTIVE: To be written before taking the lecture (Pl. write in bullet points about what will be covered in the lecture)

Hash functions based on Cipher Block chaining.
Secure Hash Algorithm

IMPORTANT & RELEVANT QUESTIONS:

FEED BACK QUESTIONS (AFTER 20 MINUTES):

OUTCOME OF THE DELIVERED LECTURE: To be written after taking the lecture (Pl. write in bullet points about students' feedback on this lecture, level of understanding of this lecture by students etc.)

REFERENCES: Text/Ref. Book with Page No. and relevant Internet resources

Afsal Kahati

Hash functions based on Cipher Block chaining -

A number of proposals have been made for hash functions based on using a cipher Block chaining technique, but without using secret key:

Rabin divide a message M into fixed size blocks $M_1, M_2, \ldots, M_N$ and use a symmetric encryption system such as DES to compute the hash code as G as -

H_0 = initial value.

$H_i = E(M_i, H_{i-1})$

$G = H_N$

This is similar to the CBC technique, but in this case there is no secret key. As with any hash code, this scheme is subject to the birthday attack and if the encryption algo is DES and only a 64-bit hash code is produced, then the system is vulnerable.

Another version of birthday attack can be used even if the opponent has access to only one message and its valid signature and cannot obtain multiple signings we assume that the opponent intercepts a message ~~and its~~ with a signature in the form of an encrypted hashcode and that the unencrypted hash code is m bits long.

1) Use the algorithm defined at the beginning to calculate the unencrypted hashcode G.

2) Construct any desired message in the form $Q_1, Q_2 \dots Q_{N-2}$.

3) Compute

$$H_i = E(Q_i, H_{i-1}) \text{ for } 1 \dots i \dots (N-2).$$

4) Generate $2^{m/2}$ random blocks, for each block X, compute $E(X, H_{N-2})$. Generate an additional $2^{m/2}$ random blocks, for each block Y, compute $D(Y, G)$ where D is the decryption function corresponding to E.

5) Based on birthday Paradox with high probability there will be an X and Y such that

$$E(X, H_{N-2}) = D(Y, G)$$

6) Form the message $Q_1, Q_2, A, Q_{N-2}, X, Y$. This message has the hash code G and therefore can be used with the intercepted encrypted signature.

This form of attack is known as meet-in the middle attack.

More generally, it can be shown that some form of birthday attack will succeed against any hash scheme involving the use of cipher-block-chaining without a secret key, provided that either the resulting hash code is small enough or that a larger hash code can be decomposed into independent subcodes.

MACs Based on Block Ciphers -

PAGE NO. .....

① DAA and ② CMAC

Data Authentication Algorithm (DAA)

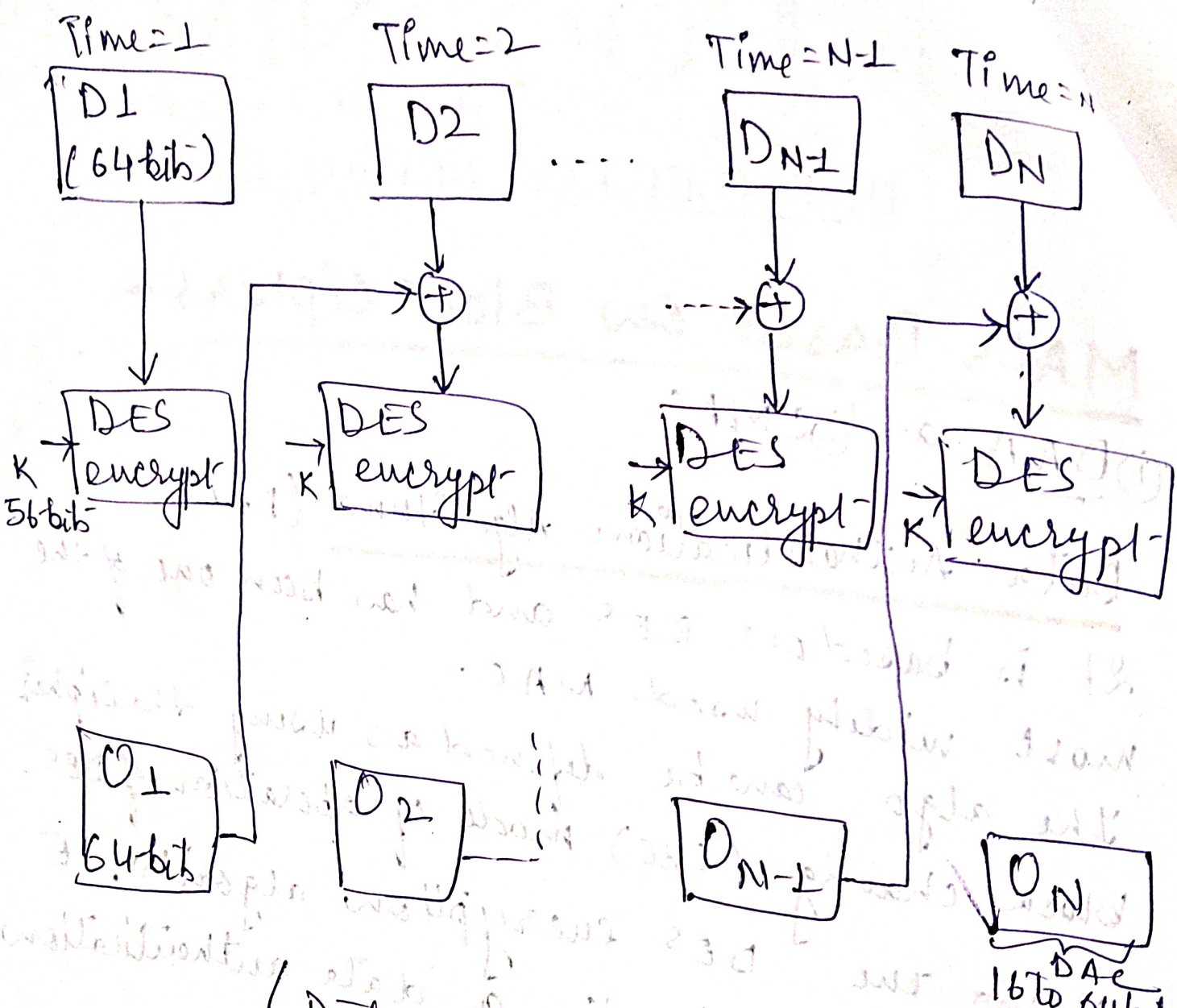
It is based on DES and has been one of the most widely used MAC.

The algo. can be defined as using the Cipher block chaining (CBC) mode of operation of DES.

Using the DES encryption algorithm E and a secret key K, a data authentication code (DAC) is calculated as follows -

$$O_1 = E(K, D_1)$$
$$O_2 = E(K, [D_2 \oplus O_1])$$
$$O_3 = E(K, [D_3 \oplus O_2])$$
$$\vdots$$
$$O_N = E(K, [D_N \oplus O_{N-1}])$$

The DAC consists of either the entire block O_N or the left most M bits of the block, with $16 <= M <= 64$.



(Data Authentication Algorithm)

Secure Hash Algorithm (SHA-1)

PAGE NO.

NIST along with NSA developed - SHA 1

↓

National Institute of Standards & Technology.

- SHA is modified version of MD5 and its design closely resembles MD5.
- I/P message in SHA 1 of length → less than 2^{64} bits in length.
- O/P of SHA → message digest which is 160 bits in length.

SHA is designed to be computationally infeasible to –

(i) obtain the original message, given its message digest.

(ii) find two messages producing the same message-digest.

Working -

① Padding - The first step in SHA is to add padding to the end of original message in such a way that the length of message is 64 bits short of a multiple of 512. Like MD5, the padding is always added, even if the message is already 64 bits short of a multiple of 512.

② Append length — The length of the message excluding the length of the padding is now calculated & appended to the end of the padding as a 64-bit block.

③ Divide the input into 512 bit blocks - The input message is now divided into blocks each of length 512 bits. These blocks become the input to the message digest processing logic.

④ Initialize chaining variables. — Now, five chaining variables A through E are initialized

A	Hex	01	23	45	67
B	Hex	89	AB	CD	EF
C	Hex	FE	DC	BA	98
D	Hex	76	54	32	10
E	Hex	C3	D2	E1	F0

(5) Process Blocks - Now, the actual algorithm begins -

(5.1) copy the chaining variables A-E into variables a-e. The combination of a-e. called as abcde will be considered as a single register for storing the temporary intermediate as well as final results.

(5.2) Divide the current 512 bit block into 16 sub-blocks, each consisting of 32 bits.

(5.3) SHA has four rounds, each round consists of 20 steps. Each round takes the current 512-bit block, register abcde and a constant K[t] (where t = 0 to 79) as three inputs. It then updates the contents of register abcde using SHA algorithm steps

values of K[t]

Round	values of t between	K[t] in hexadecimal	K[t] in decimal
1	1 and 19	5A 92 79 99	$2^{30} \times \sqrt{2}$
2	20 and 39	6E D9 EB A1	$2^{30} \times \sqrt{3}$
3	40 and 59	9F 1B BC DC	$2^{30} \times \sqrt{5}$
4	60 and 79	CA 62 C1 D6	$2^{30} \times \sqrt{10}$

(5.4) SHA consists of four rounds, each round containing 20 iterations. This makes it a total of 80 iterations.

$$abcde = (e + Process P + s^5(a) + W[t] + K[t]),$$

where $a, s^{30}, (b), c, d$.

abcde – register made up of 5 variables a, b, c, d, e

Process P – logical operation

s^t – circular shift of 32 bit sub block by t bits

W[t] – A 32-bit derived from current 32-bit sub block

K[t] – One of the five additive constants

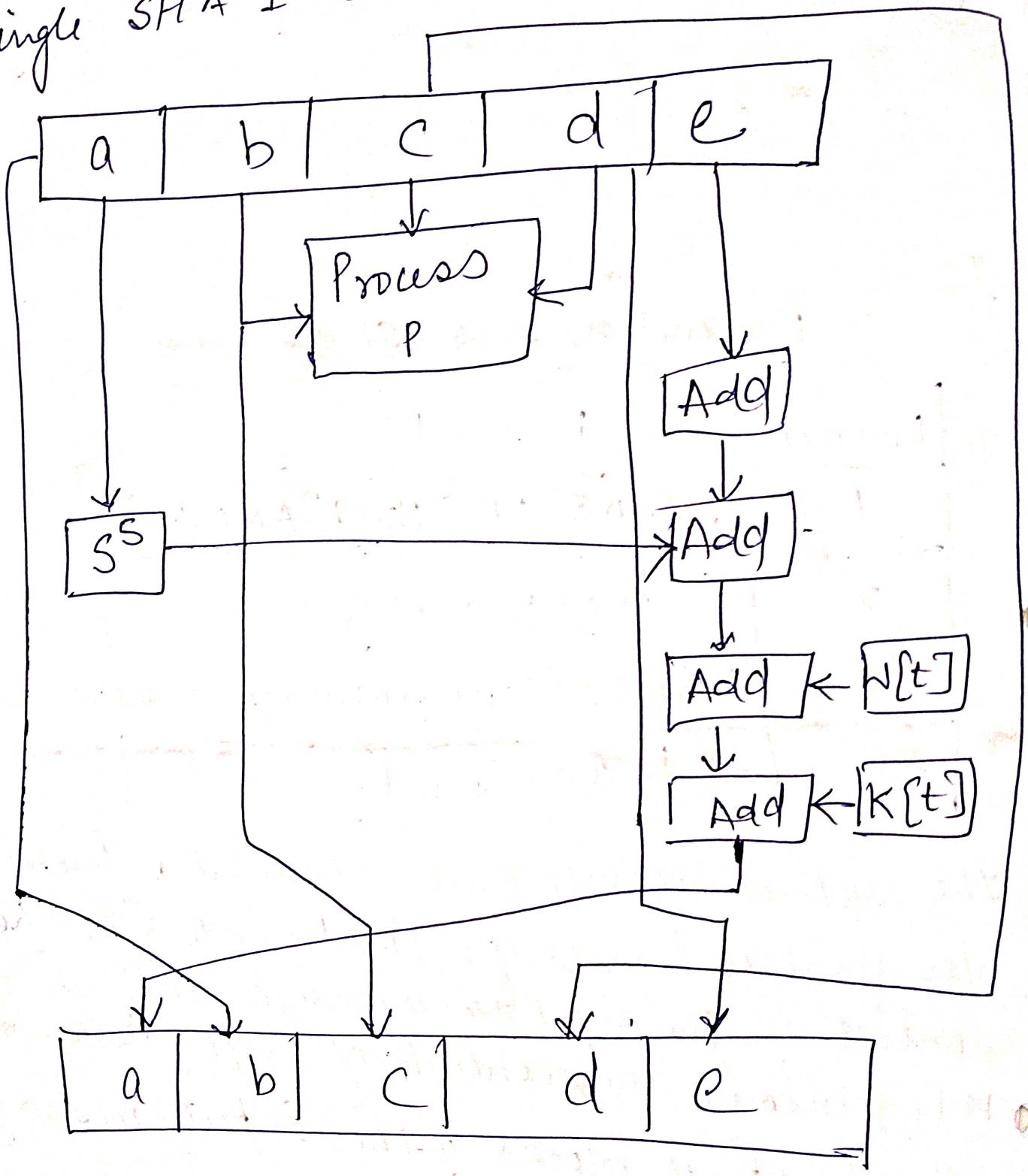
Processing in each SHA-1 Round

Round	Process P
1	(b AND c) OR ((NOT b) AND (d))
2	b XOR c XOR d
3	(b AND c) OR (b AND d) OR (c AND d)
4	b XOR c XOR d.

The values of W[t] are calculated as follows —
for first 16 words of W (i.e t = 0 to 15), the contents of the input message sub-block M[t] become the contents of W[t] straightaway.
The first 16 blocks of the input message M are copied to W.
The remaining 64 values of W are defined using the equation —

$$W[t] = S^1(W[t-16] \text{ XOR } W[t-14] \text{ XOR } W[t-8] \text{ XOR } W[t-3]).$$

Single SHA-1 iterations are as follows -



POORNIMA

COLLEGE OF ENGINEERING

LECTURE NOTES

Campus: ..... Course: ..... Class/Section: ..... Date: .....

Name of Faculty: ..... Name of Subject: ..... Code: .....

Date (Prep.): ..... Date (Del.): ..... Unit No./Topic: ..... Lect. No: .....

OBJECTIVE: To be written before taking the lecture (Pl. write in bullet points the main topics/concepts which will be taught in this lecture)

Message Authentication code
Requirements and security.
MAC based Hash functions & Block ciphers

IMPORTANT & RELEVANT QUESTIONS

FEED BACK QUESTIONS (AFTER 20 MINUTES)

OUTCOME OF THE DELIVERED LECTURE: To be written after taking the lecture (Pl. write in bullet points about students' feedback on this lecture, level of understanding of this lecture by students etc.)

REFERENCES: Text/Ref. Book with Page No. and relevant Internet Websites:

Atul Kahate

Message Authentication Codes

PAGE NO.

- Message Authentication is a mechanism or service used to verify the integrity of a message. It assures that data received are exactly as sent by and that the purported identity of the sender is valid.

- MAC is an algorithm that requires the use of a secret key. A MAC takes a variable-length message and a secret key as input and produces an authentication code. A recipient in possession of secret key can generate an authentication code to verify the integrity of the message.

- MAC is to use a symmetric block cipher in such a way that it produces a fixed length output for a variable length input

Message Authentication Requirements.

In the context of communication across a network, the following attacks can be identified

1. Disclosure - Release of message contents to any person or process not possessing the appropriate cryptographic key.

2. Traffic Analysis - Discovery of the pattern of traffic between parties. In a connection oriented applications, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.

3. Masquerade - Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgement of message receipt or non receipt by someone other than the message recipient.

PAGE NO. .....

(4) Content Modification – Changes to the contents of a message including insertion, deletions, transposition and modification.

(5) Sequence Modification – Any modification to a sequence of messages between parties including insertion, deletions and reordering.

(6) Timing Modification – Delay or replay of messages. In a connection-oriented application an entire session or sequence of message could be a replay of some previous valid session or individual messages in the sequence could be delayed or replayed: In a connectionless application, an individual message could be delayed or replayed.

(7) Source repudiation – Denial of transmission of message by source.

(8) Destination Repudiation – Denial of receipt of message by destination.

In summary, message authentication is a procedure to verify that received messages come from the alleged source and have not been altered.

Message Authentication may also verify sequencing and timeliness.

POORNIMA

COLLEGE OF ENGINEERING

LECTURE NOTES

Campus: ..... Course: ..... Class/Section: ..... Date: .....

Name of Faculty: ..... Name of Subject: ..... Code: .....

Date (Prep.): ..... Date (Del.): .....Unit No./Topic:.....Lect. No: .....

OBJECTIVE: To be written before taking the lecture (Pl. write in bullet points the main topics/concepts which will be taught in this lecture)

Digital Signatures -
Requirements and security -

IMPORTANT & RELEVANT QUESTIONS:

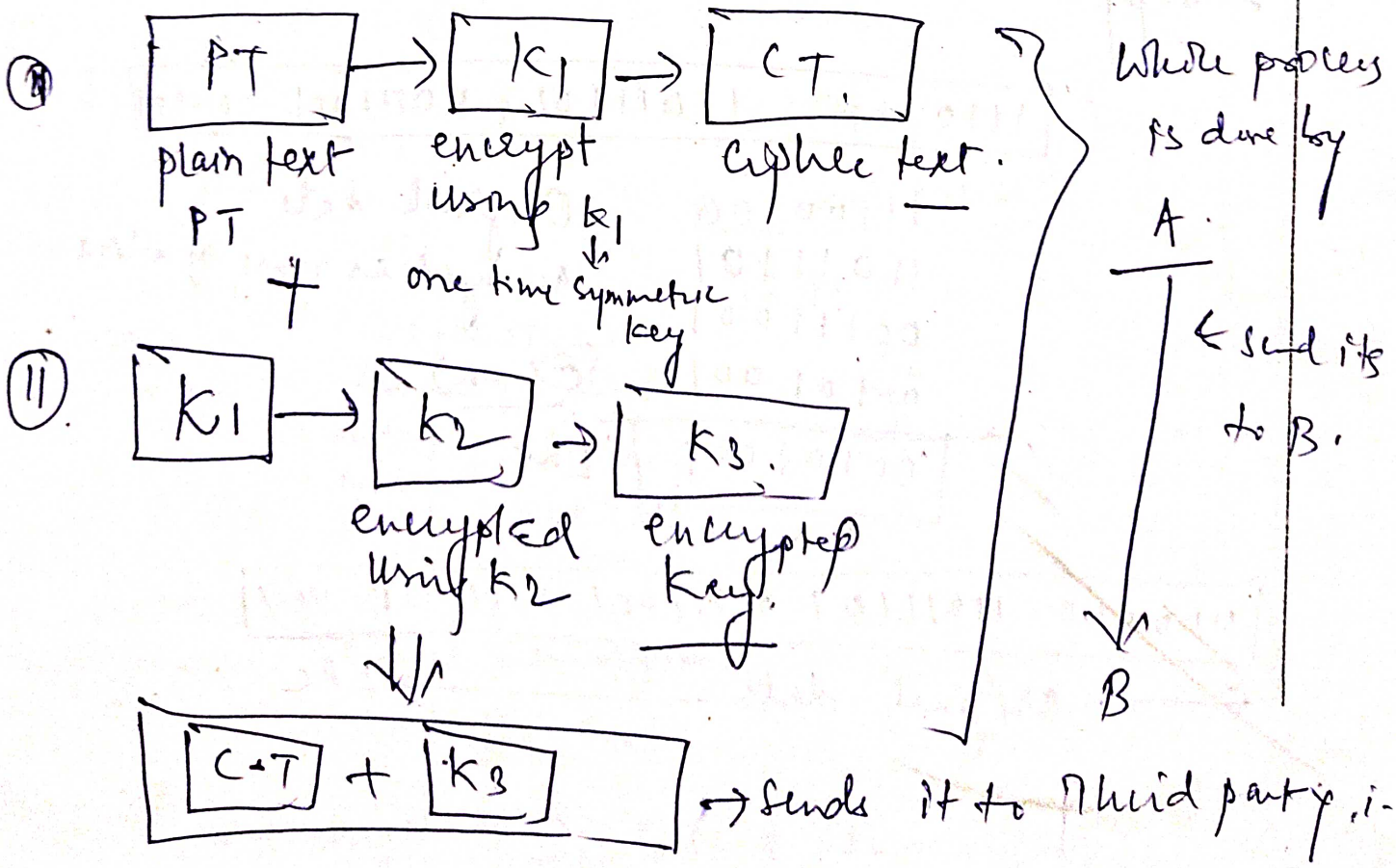
FEED BACK QUESTIONS (AFTER 20 MINUTES):

OUTCOME OF THE DELIVERED LECTURE: To be written after taking the lecture (Pl. write in bullet points about students' feedback on this lecture, level of understanding of this lecture by students etc.)

Message Digest :-

→ The conceptual process of Digital signatures, it doesn't deal with the problems associated with asymmetric key, namely slow operation & large cipher text. This is due to we are encrypting the whole of the original plain text message with the sender's private key.

→ We can tackle this issue by digital envelope.



However in real practice, a more efficient scheme used. It involves the usage of a message digest (a called as a hash)

⇒ **What is message Digest:**

A message digest is a fingerprint or the Summary of a message. It is similar to the concepts of Longitudinal Redundancy check (LR) or Cyclic Redundancy Check (CRC). That is used to verify the integrity of the data

→ **LRC Calculation**

→ list of bits are organised in the rows.

⇒ Suppose we need to send 32 bits of data.

1110010	1101110	0011100	0010100

11100100	Original data
11011101	arranged as rows of a list.
00111001	
00101001	(EX-OR)
00101001.	LRC.

1110010	1101110	0011100	0010100	0010100

← Original data → LRC

PAGE NO: .....

Idea of message Digest

→ Suppose we have number 4000 divide it by 4

$$4'000/4 = 1000$$

↓ finger print of a number 4000

(1). either we change 4000, or 4 it will not produce =1000.

⇒ (2). i.e Number 4 doesn't tell anything about the original number.

Eg: Message Digest of a No. 7391743.

original message

7391743

Multiply 7×3	21
Discard first digit	1
Multiply it by 9×1	9
Multiply 7×9	63
Discard first digit	3
Multiply 9×3	12
Discard first digit	2
Multiply 2×3	6.

Note:- If next digit is 0, discard it.

Requirements of a Message Digest :-

1. Original data → Message digest algorithm (Should be possible and the result should always be the same) → Message digest.

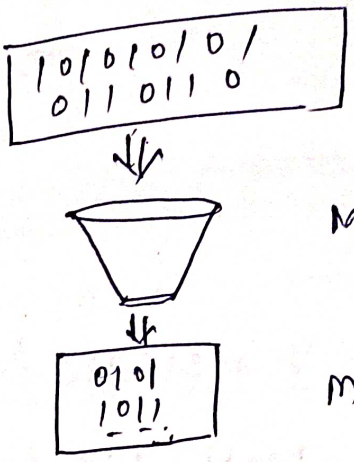


Fig:1:- Message Digest Concept.

Fig 2: Message digest for the same original data should always be same.

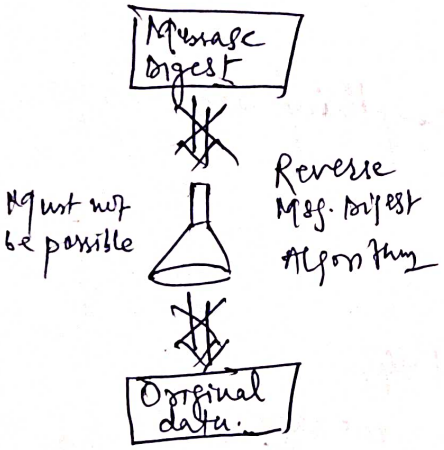
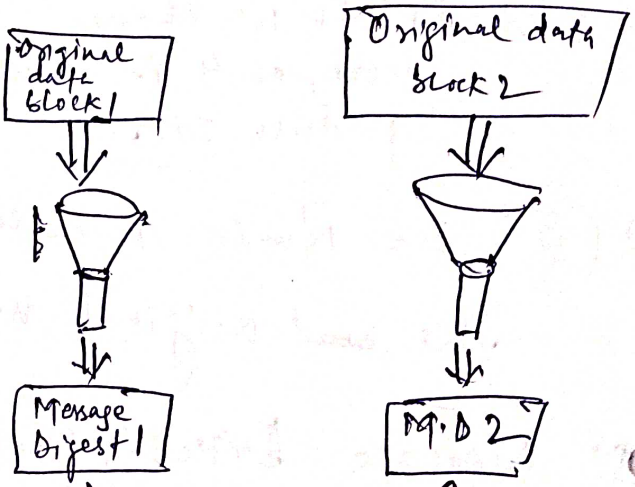


Fig 3. Message digest should not work in the opposite direction.



These 2 message digests must be different.

Fig 4. Message Digests of two different messages must be different.

1. For a Given message, it should be very easy to find corresponding message digest. ie Fig. 2,1.
2. Given a message digest, it should be very difficult to find the original message for which digest was created.
3. Given any two messages, if we calculate their message digest, two message digests must be different. ie Fig - Fig-4.

DETAILED LECTURE NOTES

Campus: ..... Course: ..... Class/Section: ..... Date: .....

Name of Faculty: ..... Name of Subject: ..... Code: .....

→ If any two messages produce the same message digest, thus violating our principle, it is called as a collision. That is, if two message digests collide, they meet at the digest! ~~As we shall study soon.~~

→ The chances of any two message digests ~~of length 128 b~~ being the same are one in 2^{128} or 2^{160} respectively.

→ A specific type of security attack called as birthday attack, is used to defeat collisions in message digest algorithms.

→ It is based on the principle of Birthday Paradox, which states if there are 23 people in a room, chances are more than 50% that two of the people will share the same birthday.

→ Birthday attack is most often used to attempt discover collisions in has functions such as MD5 &

How MD5 works?

Step1: Padding!

The first step in MD5 is to add padding bits to the original message. The aim of this step is to make the length of the original message equal to a value, which is 64 bits less than an exact multiple of 512.

Ex. If message length = 1000

Add = 472

Total = 1472 bits.

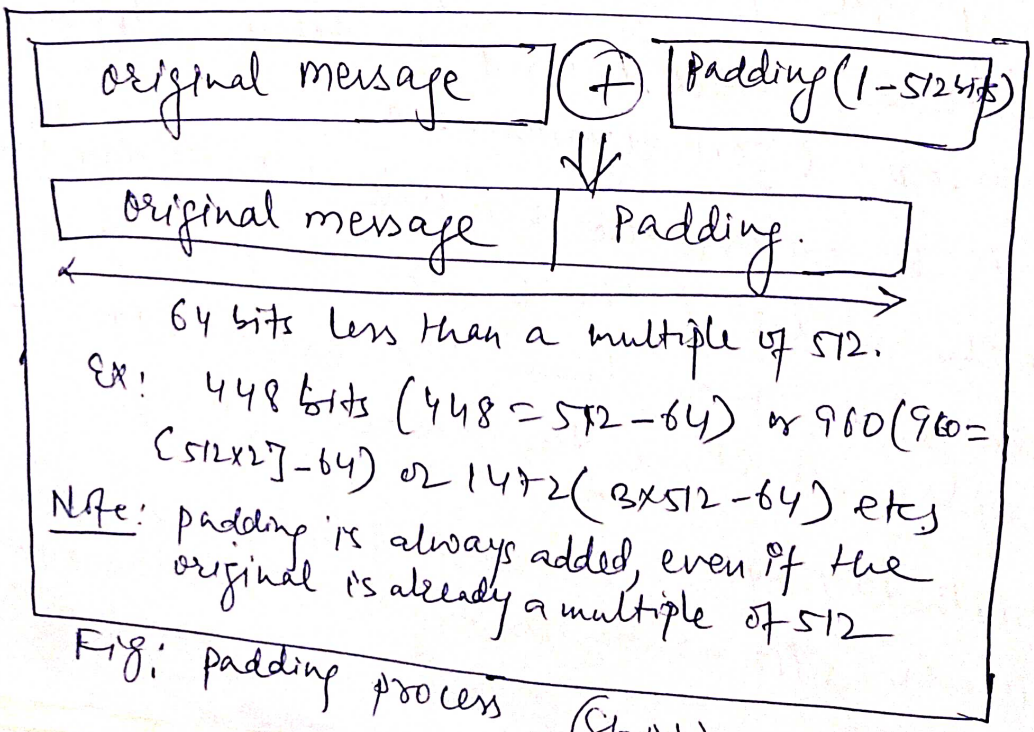
512x3 = 1536

−64

1472

Step2: Append length

- To calculate original length of the message & add it to the end of the message, after padding



Ex: 448 bits (448 = 512 − 64) or 960 (960 = [512x2] − 64) or 1472 (3x512 − 64) etc.

Note: padding is always added, even if the original is already a multiple of 512

Fig: padding process (Step1)

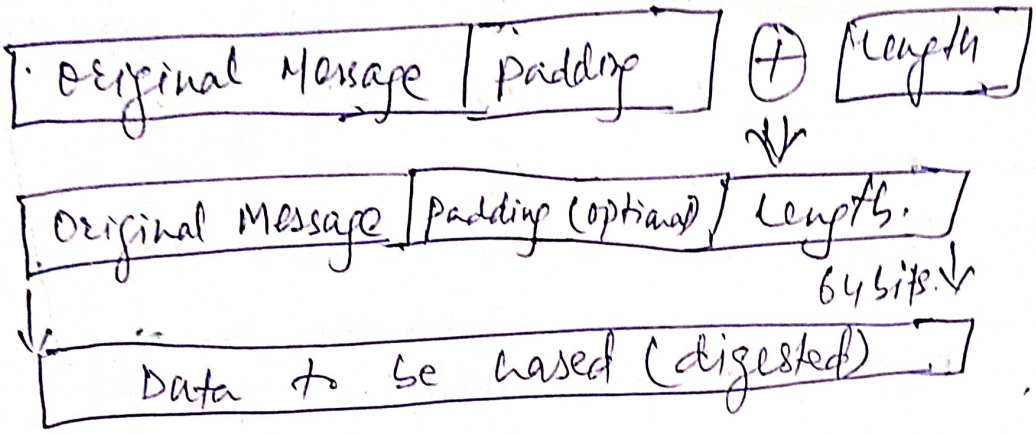


Fig: Append length. (Step 2).

Steps: Divide the input into 512-bit blocks:

Now we divide the input message into blocks, each of length 512 bits.

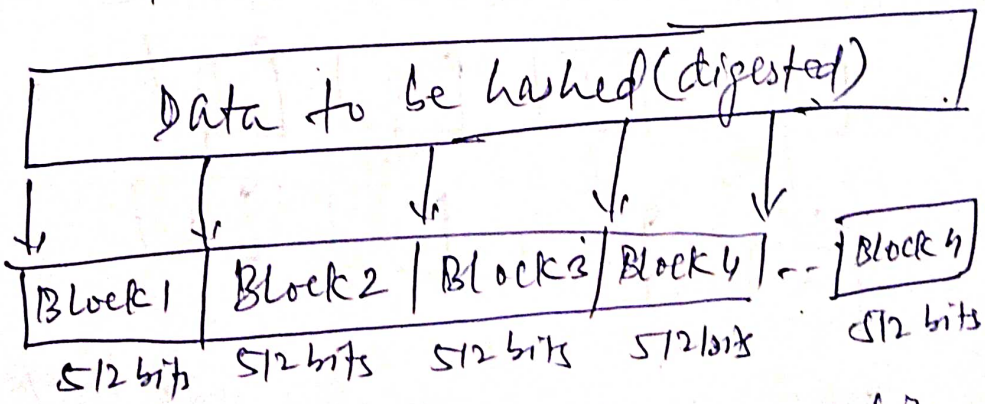


Fig - Data is divided into 512 bits.

Step 4: Initialize chaining variables:

In this step, four variables (called as chaining variables) are initialized. They are called as A, B, C & D. Each of these is a 32 bit No.

A	HEX	01	23	45	67
B	HEX	89	AB	CD	EF
C	HEX	FE	DC	BA	98
D	HEX	76	54	32	10

Fig: Chaining variables.

Step 5: Process Blocks: After all initializations, the real algorithm begins.

Step 5.1: Copy the four chaining variables a, b, c & d.

Thus, we now have a = A, b = B, c = C, d = D

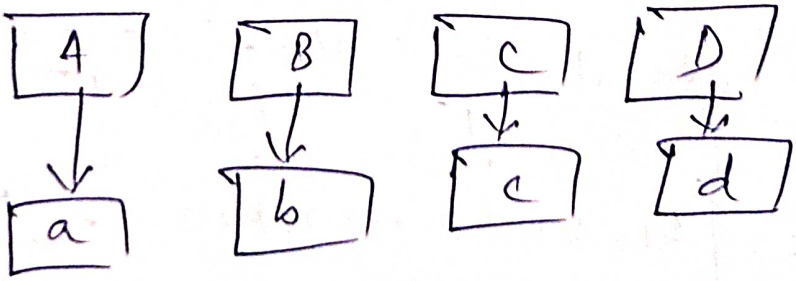


Fig: Copying chaining variables into temporary variables.

Campus: ..... Course: ..... Class/Section: ..... Date: .....

Name of Faculty: ..... Name of Subject: ..... Code: .....

Step 5.2: Divide the current 512-bit block into 16 sub-blocks. Thus, each sub-block contains 32 bits.

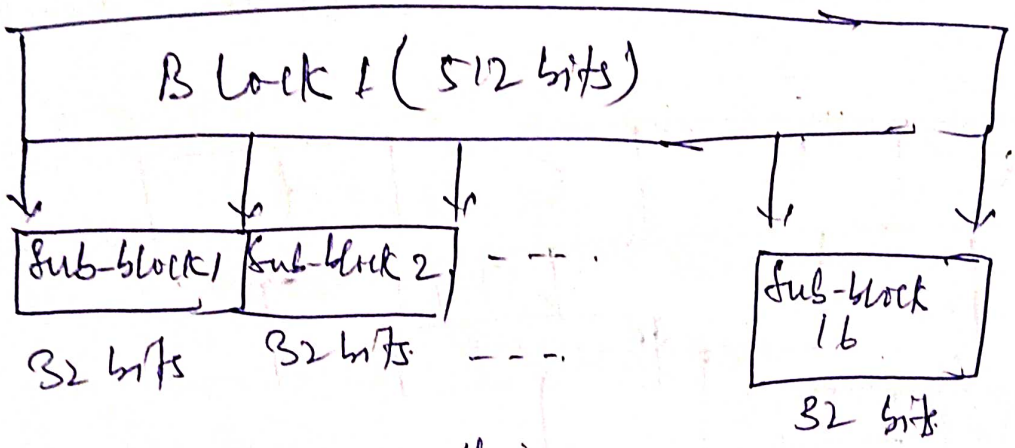


Fig: Sub-Blocks within a block.

Step 5.3: Now, we have four rounds, in each round we process all the 16 sub-blocks (6)

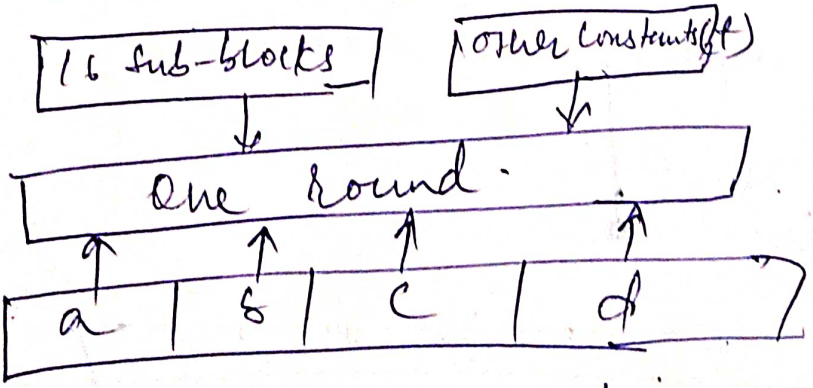


Fig: Conceptual process with in a round.

→ In each round, we have 16 input sub-blocks named M[0], M[1], ---- M[15] or in General M[i] where i varies from 0 to 15. each subblock consists of 32 bits.

→ Also, t is an array of constants. It contains 64 elements. we denote the elements of this array t as t[1], t[2], --- t[64] or in general t[k], where k varies from 1 to 64.

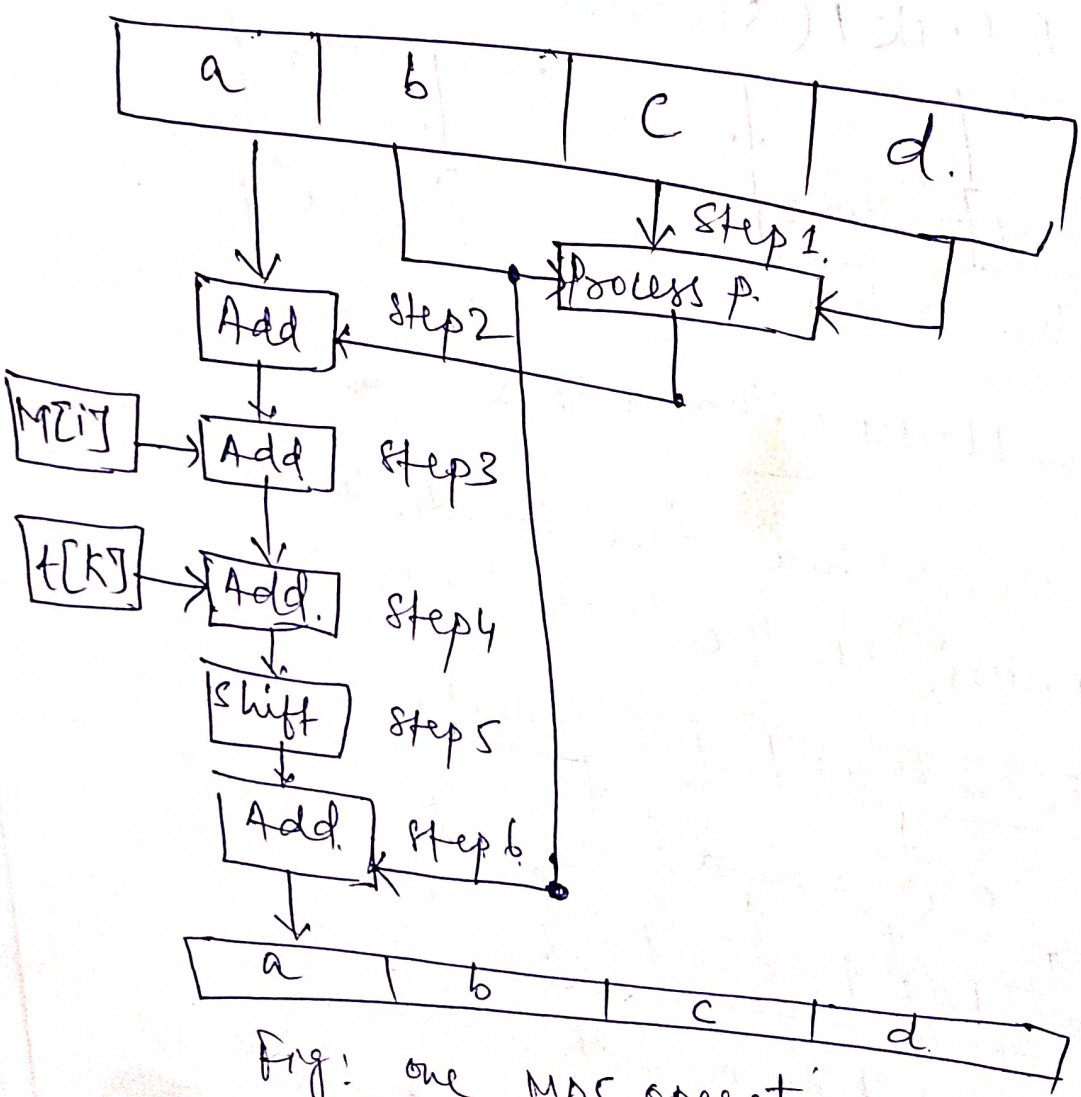


Fig: one MD5 operation.

We can mathematically express a single MD5 operation as follows:

$$a = b + ((a + \text{process } P(b,c,d) + M[i] + T[k]) <<< s$$

Where

a, b, c, d = Chaining variables,

Process P = A non-linear operation.

M[i] = M[j × 16 + i], which is the ith 32-bit word in the jth 512 bit block of the message.

t[k] = A constant

<<< s = Circular left shift by s-bits.

Table. Process P in each round.

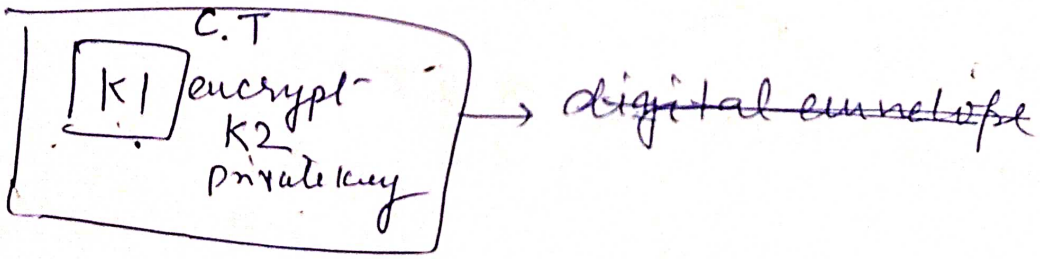
Round	Process
1	(b AND c) OR ((NOT b) AND (d))
2	(b AND d) OR (c AND (NOT d))
3	B XOR c XOR d)
4	C XOR (b OR (NOT d))

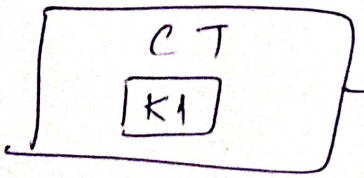
Message Digest -

(1) A encrypt-
↓
PT. → with one time key K1 → form CT ~~Private key K2~~

(2) A then ~~creates~~ encrypts the one time symmetric key K1 with her private key (K2). She creates a digital envelope containing CT and K1 encrypted with K2 and send this digital envelope to B.

(3)





→ digital envelope.

B. open:

A - public key K3.

↓

decrypt.

↓

obtains K1

now uses K1 to

decrypt the CT and obtain PT.

B uses A's public key ⇒ decrypt the key K1

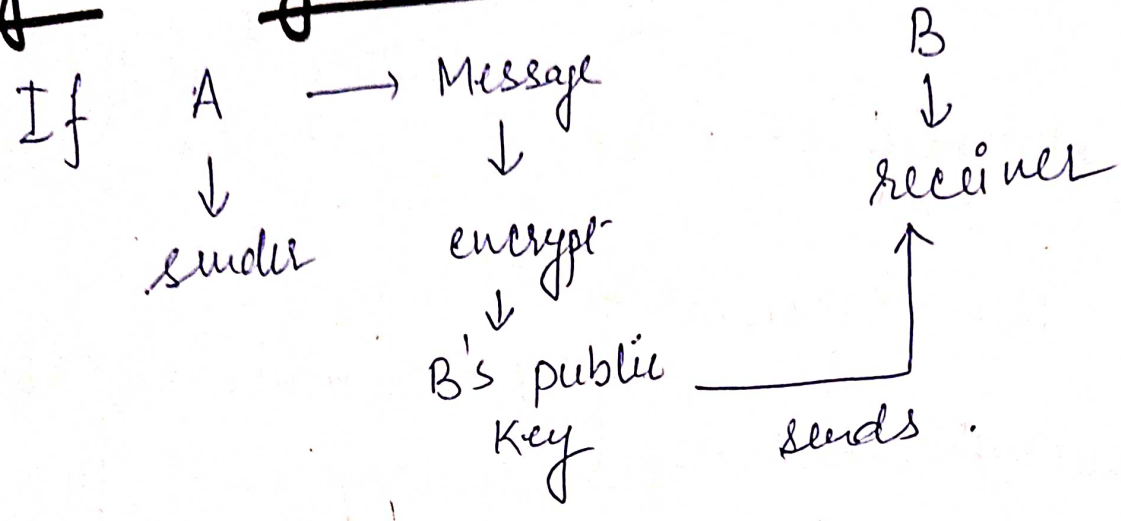
B can be assured that only A's private key could have encrypted K1.

Thus B can be assured that the digital envelope came from A.

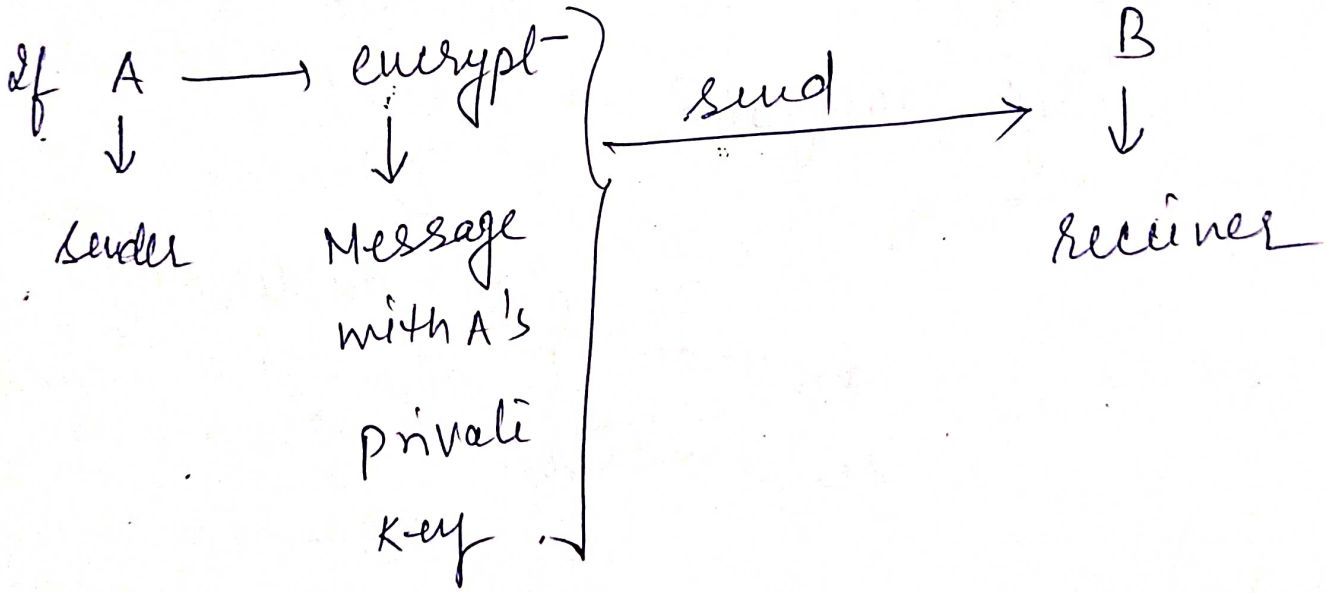
And in current scenario it also involves message digest.

DETAILED LECTURE NOTES

Digital Signatures -



opposite :



Reaction -

A's public key is known to everyone. anybody is accessible the ~~acce~~ message.

but when A's encrypts the message with his private key.

A
↓ private key ⇒ her intention is not to hide the contents of message.
Encrypt.

Confidentiality acheive ✗.

but it will something else.

If B → receives
↓
encrypted message. } A's public key open. Message will read when it is open by A's public key.

Authentication.

C → private key } encrypted → B send
↓
proved Authentication { A's public key open - ✗.

DETAILED LECTURE NOTES

In case of Dispute → A cannot refuse that she had sent this message as message is decrypted by her public key.

- Non-repudiation →

Interception → Message → n/w → change.

C wants to interrupt the message in n/w transit. he will not able to do. why.

A's private key - X.

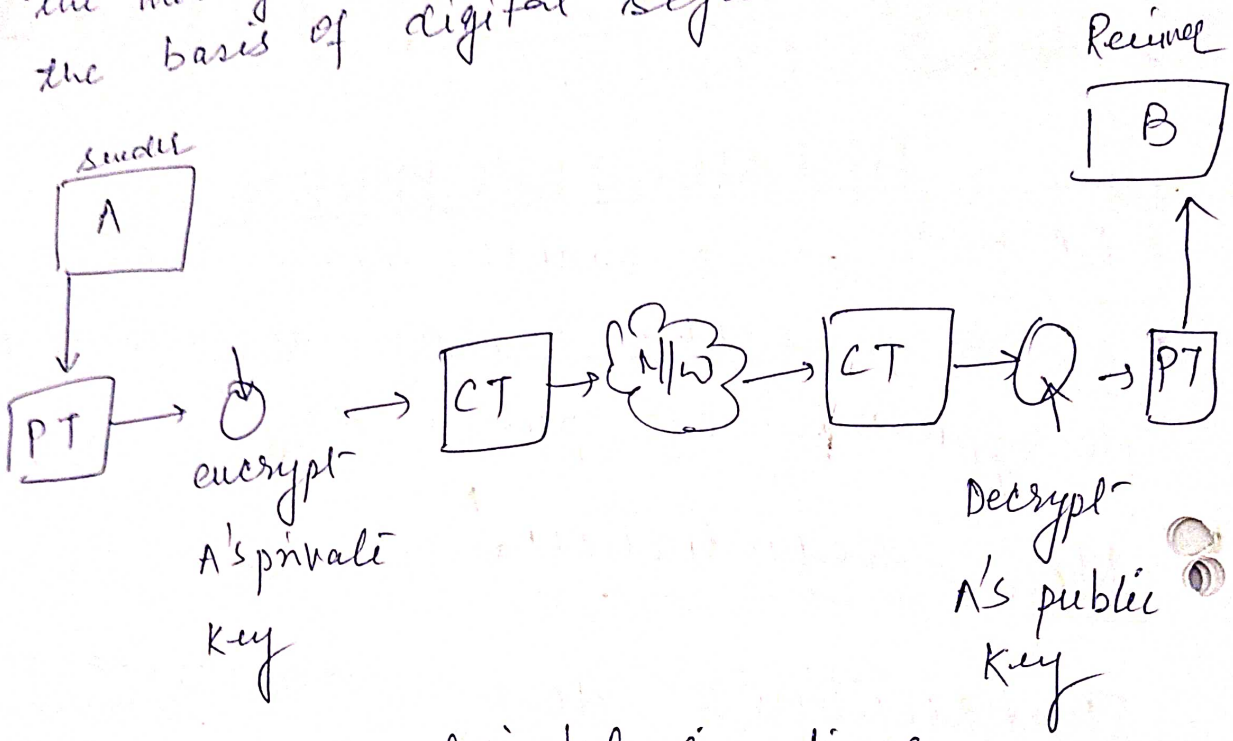
C will not able to change the message. why because.

C → encrypted message → A's public key → Decrypt.

B cannot be fooled. ← if he will do the same. ← not able to again encrypt

B's has A's public key.

Such a scheme where the sender encrypts the message with her private key, forms the basis of digital signatures.



Basis of digital signatures.

Digital signatures will not deal with the problems of Asymmetric key encryption named slow operation and large cipher text size; because are encrypting the whole plain text with sender's private key. As the size of PT can be quite large. then the encryption process can be really very slow.

So, this will resolve using digital envelope.

POORNIMA

COLLEGE OF ENGINEERING

LECTURE NOTES

Campus: Course: Class/Section: Date:

Name of Faculty: Name of Subject: Code:

Date (Prep.): Date (Del.): Unit No./Topic: Lect. No:

OBJECTIVE: To be written before taking the lecture (Pl. write in bullet points the main topics/concepts etc. to be taught) will be covered in this lecture

Elgamal and Schnorr digital certificate Schemes.

IMPORTANT & RELEVANT QUESTIONS:

FEED BACK QUESTIONS (AFTER 20 MINUTES):

OUTCOME OF THE DELIVERED LECTURE: To be written after taking the lecture (Pl. write in bullet points about students' feedback on this lecture, level of understanding of this lecture by students etc.)

REFERENCES: Text/Ref. Book with Page No. and relevant Internet Websites:

PAGE NO. .....

Elgamal Digital signature Scheme -

- It involves the use of private key for encryption and public key for decryption.

→ The global elements of Elgamal digital signature are a prime number q and α, which is a primitive root of q.

User A generates a private / public key pair as follows -

1. Generate a random integer X_A, such that $1 < X_A < q-1$
2. compute $Y_A = \alpha^{X_A} \bmod q$
3. A's private key is X_A; A's public key is $\{q, \alpha, Y_A\}$.

To sign a message M, user A first computes the hash $m = H(M)$, such that m is an integer in the range $0 \le m \le q-1$. A then forms a digital signature as follows -

(1) choose a random integer K such that $1 \le K \le q-1$ and gcd $(K, q-1) = 1$ That is, K is relatively prime to $q-1$.

(2) compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.

(3) compute $K^{-1} \bmod (q-1)$. That is compute the inverse of K modulo $q-1$.

(4) compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q-1)$

(5) The signature consists of the pair (S_1, S_2).

Any user B can verify the signature as follows

(1) compute $V_1 = \alpha^m \bmod q$

(2) compute $V_2 = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$.

The signature is valid if $V_1 = V_2$. Let us demonstrate that this is so. Assume that the equality is true. Then we have-

$\alpha^m \bmod q = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$. assume $V_1 = V_2$

$\alpha^m \bmod q = \alpha^{X_A S_1} \alpha^{KS_2} \bmod q$. substitute for Y_A and S_1

$\alpha^{m - X_A S_1} \bmod q = \alpha^{KS_2} \bmod q$ rearranging terms.

$m - X_A S_1 \equiv KS_2 \bmod (q-1)$ property of primitive roots

$m - X_A S_1 \equiv \cdot KK^{-1} (m - X_A S_1) \bmod (q-1)$ substituting for S_2.

Schnorr Digital Signature :

- It is based on discrete logarithms.
- It minimizes the message-dependent amount of computation required to generate a signature
- The main work for signature generation does not depend on the message and can be done during the idle time of processor.
- The scheme is based on using prime modulus p with p-1 having prime factor q of appropriate size, i.e.-

$$p-1 \equiv (\text{mod } q).$$

The first part is the generation of a private/public key pair —

① choose primes p and q, such that q is a prime factor of $p-1$

② choose an integer a such that $a^q = 1 \bmod p$.

③ choose a random integer s with $0 < s < q$. This is user's private key.

④ Calculate $v = a^{-s} \bmod p$. user publickey

A user with private key s and public key v generates a signature as follows-

① choose a random integer $0 < r < q$. and compute $x \equiv a^r \bmod p$.

② Concatenate the message with x and hash the result to compute the value e

$$e = H(M||x)$$

(3) Compute $y = (k + se) \bmod q$.
signature consists of pair (e, y)

Any other user can verify the signature as -

(1) Compute $x' = a^y v^e \bmod p$

(2) Verify $e = H(M || x')$

(3) To see that verification works -

$$x' \equiv a^y v^e \equiv a^y a^{-se} \equiv a^{y-se} \equiv a^k \equiv x \pmod p$$

Hence

$$H(M || x') = H(M || x).$$

POORNIMA

COLLEGE OF ENGINEERIN

LECTURE NOTES

Campus: ..... Course: ..... Class/Section: ..... Date: .....

Name of Faculty: ..... Name of Subject: ..... Code: .....

Date (Prep.): ..... Date (Del.): .....Unit No./Topic:.....Lect. No: .....

OBJECTIVE: To be written before taking the lecture (Pl. write in bullet points the main topics/concepts etc., which will be taught in this lecture)

NIST digital Signature Algo.

IMPORTANT & RELEVANT QUESTIONS:

FEED BACK QUESTIONS (AFTER 20 MINUTES):

OUTCOME OF THE DELIVERED LECTURE: To be written after taking the lecture (Pl. write in bullet points about students' feedback on this lecture, level of understanding of this lecture by students etc.)

REFERENCES: Text/Ref. Book with Page No. and relevant Internet Websites:

NIST Digital Signature Standard -

NIST → National Institute of standards and Technology.

- It makes use of secure hash Algorithm (SHA) and presents a new digital signature techn DSA (Digital Signature Algorithm)

First study the DSS and then DSA Algo.

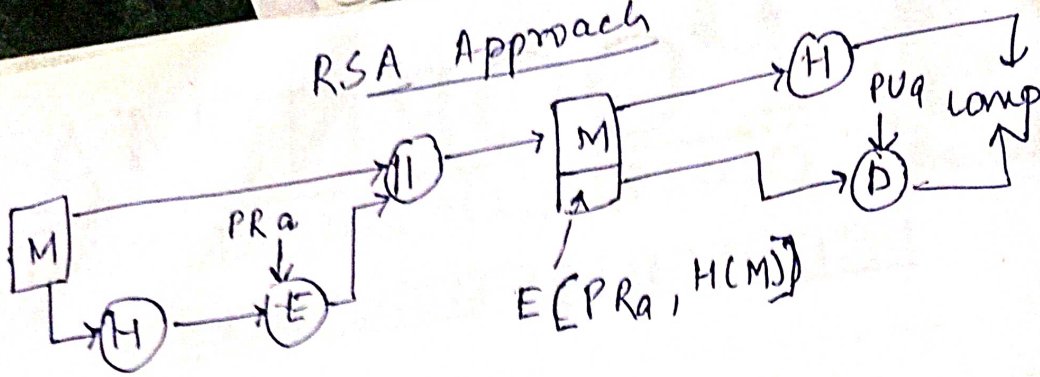
DSS Approach -

- designed to provide only the digital signature function.
- public key Technique.
- It cannot used for Encryption or Key Exchange.

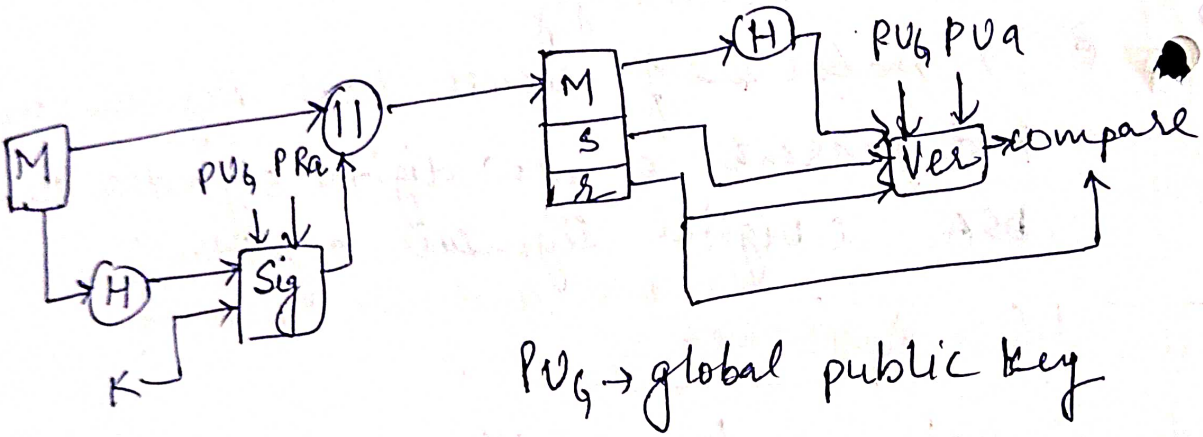
Two approaches of Digital signature

- RSA
- DSS

RSA Approach



DSS Approach.



PU_g → global public key

~~DSA~~

Digital Signature Algo –

based on

- Difficulty of computing discrete logarithms.
- based on schemes originally presented by Elgamal and Schnorr.
-

DSA → Algorithm.

PAGE NO.

Global Public key components.

$P \rightarrow$ prime no. where $2^{L-1} < p < 2^L$

$q \rightarrow$ prime divisor of $(p-1)$

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

User's private key -

$x \rightarrow$ random no. $0 < x < q.$

User's public key -

$y = g^x \bmod p$

User's per-Message Secret No -

$k =$ random no. $0 < k < q.$

Signing.

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

$s = [k^{-1} (H(M) + x r)] \bmod q.$

Signature $= (r, s)$

Verifying -

$w = (s')^{-1} \bmod q$.

$u_1 = [H(M')w] \bmod q$

$u_2 = (r')w \bmod q$.

$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$

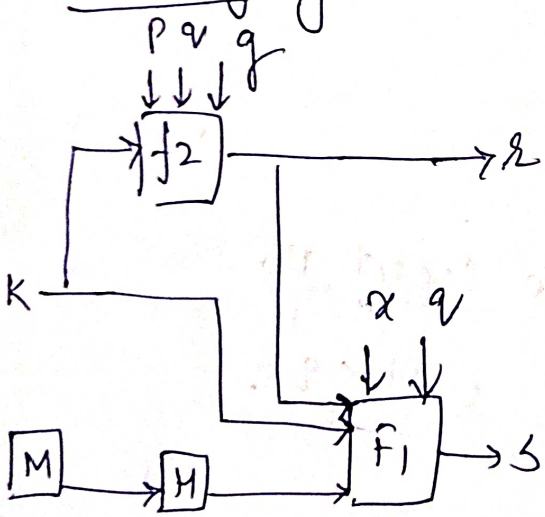
TEST : $v = r'$

Here - M → Message to be signed

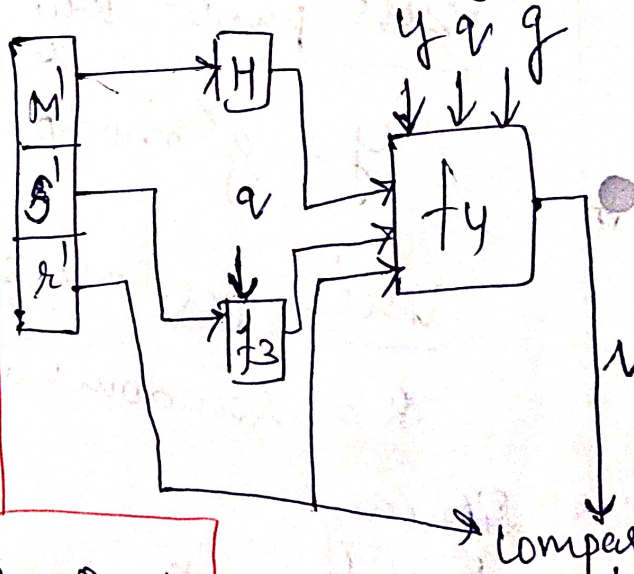
$H(M)$ → hash of M using SHA-1

M', r', s' → received versions of M, r, s.

DSS Signing:



DSS verifying:



$s = f_1(H(M), K, r, x, q) = (K^{-1}(H(M) + x \cdot r)) \bmod q$

$r = f_2(K, p, q, g) = (g^K \bmod p) \bmod q$

$w = f_3(s', q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r')$

$= ((g^{(H(M')w) \bmod q} \cdot y^{r' w \bmod q}) \bmod p) \bmod q$