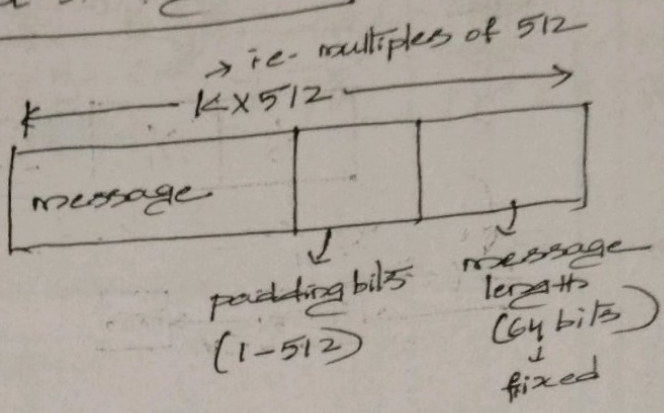
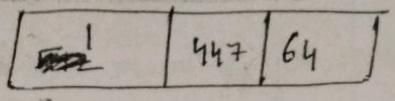
V.V. MD5 SHA PGP

Unit - IV

V.V. V.V.

Q:- Explain message Digist 5 algorithm. (MD5)



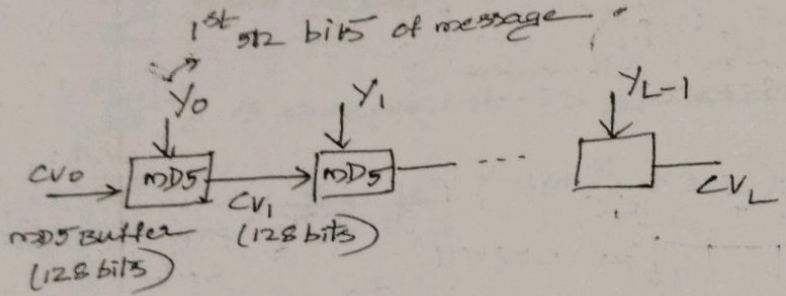


1-447
2-446
3-445

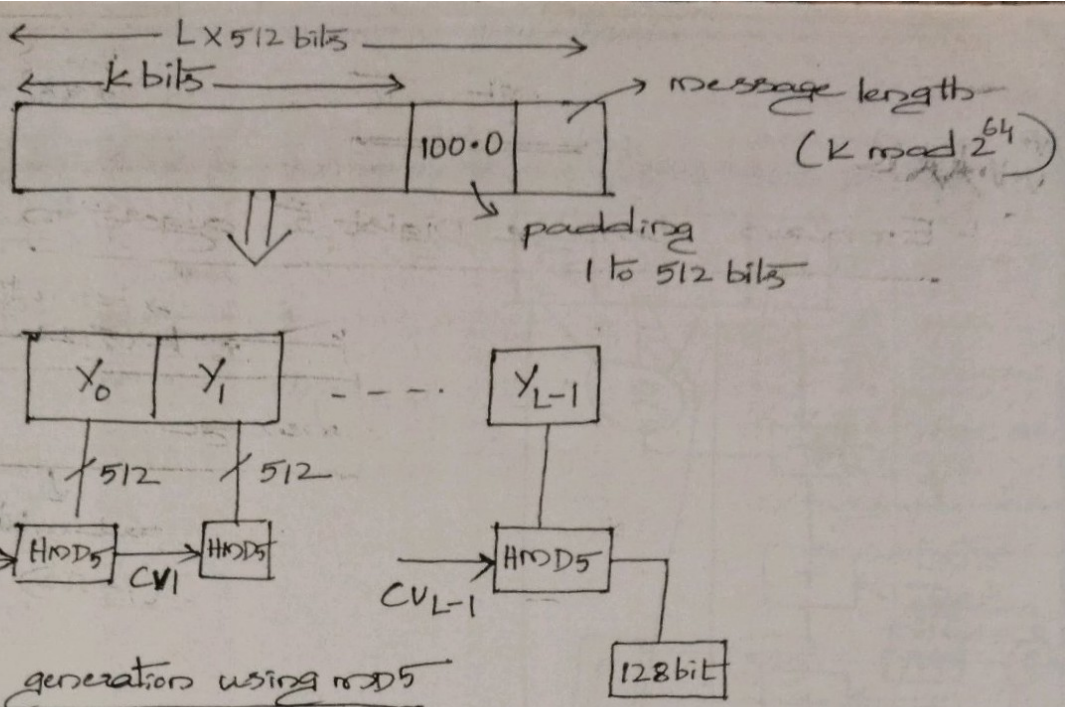
449-?

449-448 = 1

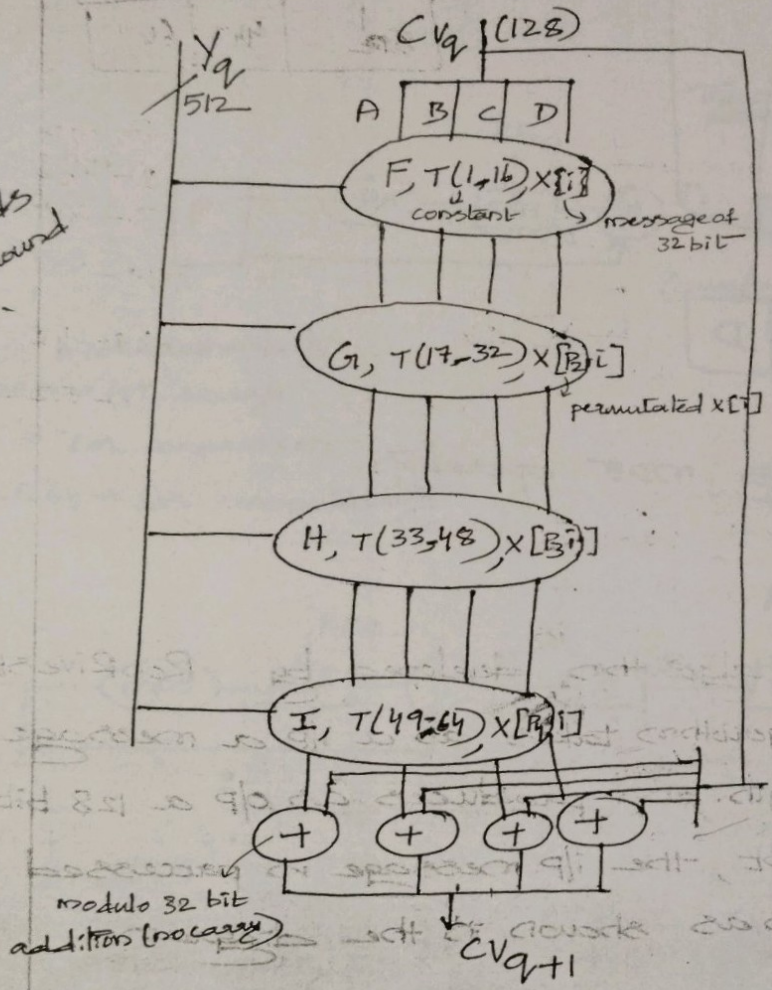
1-512 = 511



The MD5 algorithm is developed by RonRivest at MIT. The algorithm takes as a i/p a message of arbitrary length and produces as o/p a 128 bit message digist, the i/p message is paccessed in 512 bit blocks as shown in the diagram.



MD generation using MD5

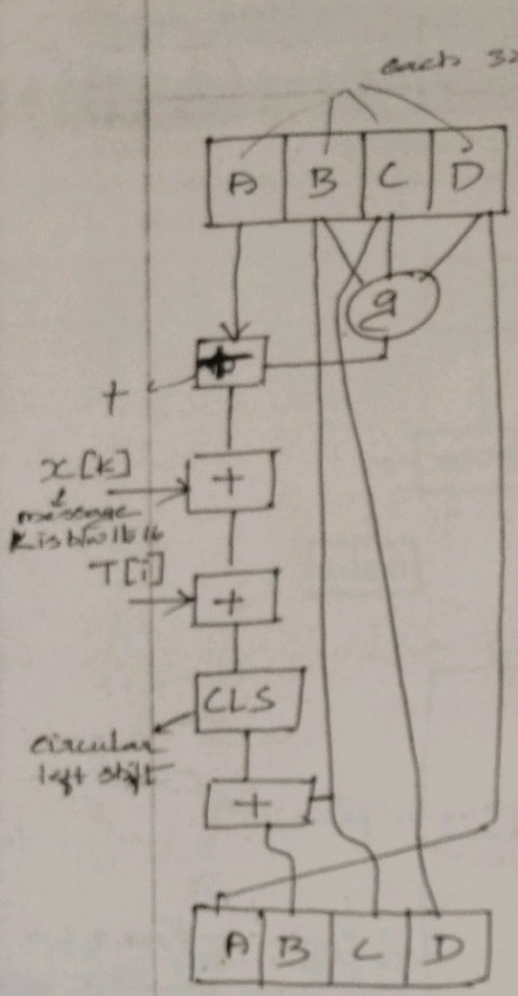


64 rounds & each round 16 steps.

F, G, H, I → primitive logical bodian fns.

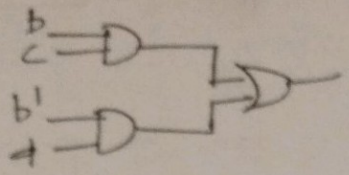
MD5 processing of single 512 bit block

2^{64}



Elementary MD5 operations

if b then c else d



H, I →
ie logical
, fns.

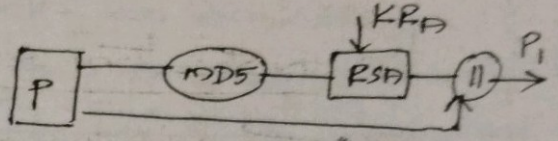
block

⇒ E-mail Security :-

Vulnerable → How resources are effected

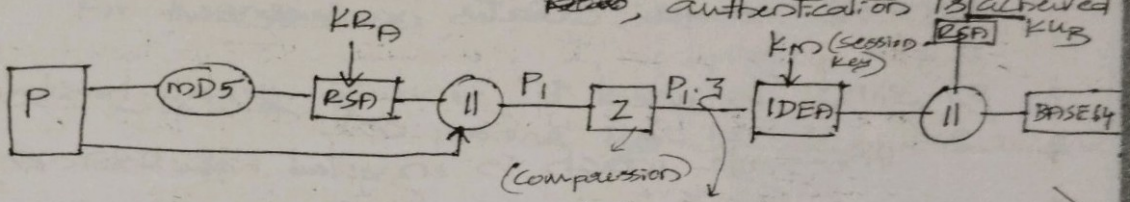
I. DES, Attacks, Techniques
II. RSA
III. Kerberos
IV. MD5, SHA

message + hash value ⇒ signature.

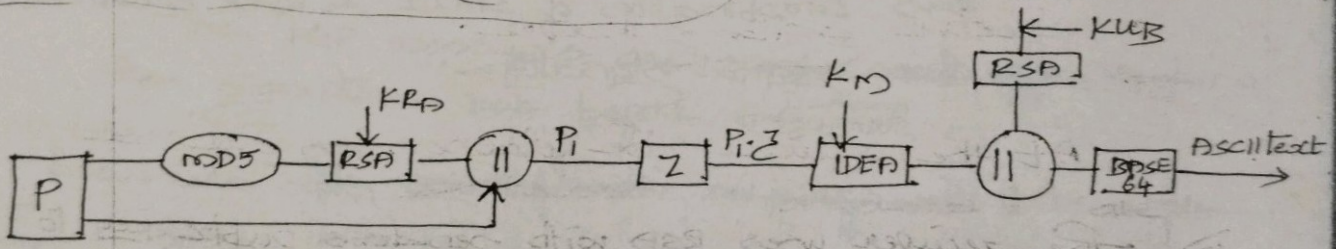


P_1 = M + Signature

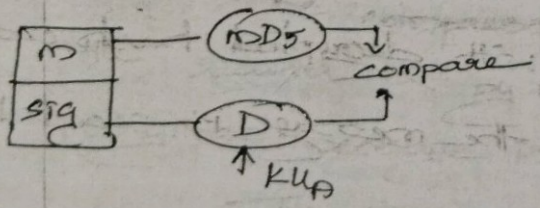
Thus, authentication is achieved



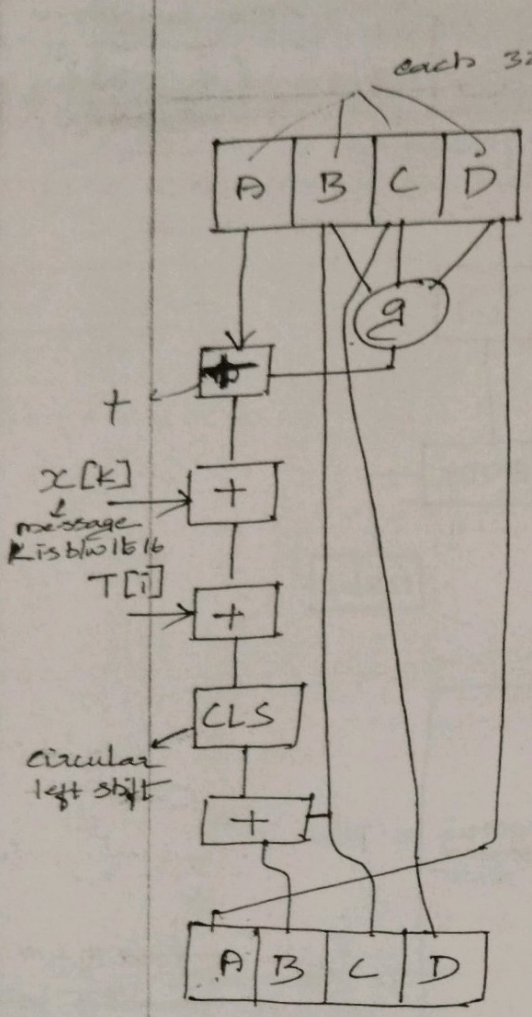
RSA → for authenticity
IDEA → for secrecy
Z → for compression
BASE 64 → for compatability



in decryption

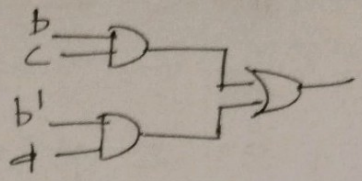


2^{64}



Elementary MD5 operations.

if b then c else d



H, I → the logical functions.

block

⇒ Explain how PGP (pretty Good Privacy) provides authentication, security, key management functions in e-mail. (page-359)

It is an effort of single person Zimmermann, provides confidentiality, authentication, compression, key management & compatability for e-mail applications.

It is developed for DOS, windows, Unix & commercial versions are also made as products.

Authentication service :-

The following steps are used.

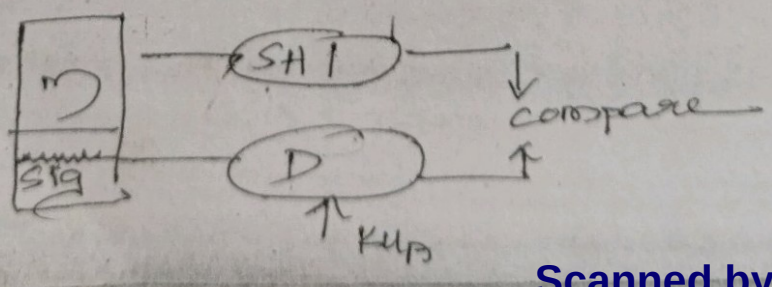
1) The sender creates a message
2) SHA-1 is used to generate 160 bit hash code of the message which is encrypted with RSA using sender's private key & the result is appended to the message.

Thus combination of SHA-1 & RSA provides an effective digital signature.

At the receiver the following steps are used.

1) The receiver uses RSA with sender's public key to decrypt & recover the hash code.
2) The receiver generates a new hash code for the message & compares with decrypted hash code.

If both are same the message is accepted as authentic.



Confidentiality

Encrypting the messages to be transmitted or to be stored locally as seperate files.

The message is encrypted by using IDEA algorithm along with a session key is generated is used for encrypting the message. Thus the message is encrypted by using IDEA with session key.

For key management the session key is encrypted with RSA using receiver's public key & is appended to the message.

At the receiver,

1) it uses RSA with it's private key to decrypt & recover the session key.
2) Then the session key is used to decrypt the message.

Compression :-

The PGP messages are compressed after applying the signature but before encryption.

The compression is performed to save the space both for event transmission & also for file storage.

The compressed message is decrypted at the receiver & by using uncompress algorithm the original message is obtained.

The compression algo used is Zip algo

E-mail compatability

The message of PGP is converted into ASCII text format by using RADIX 64 algorithm.

& converts into ASCII text format.

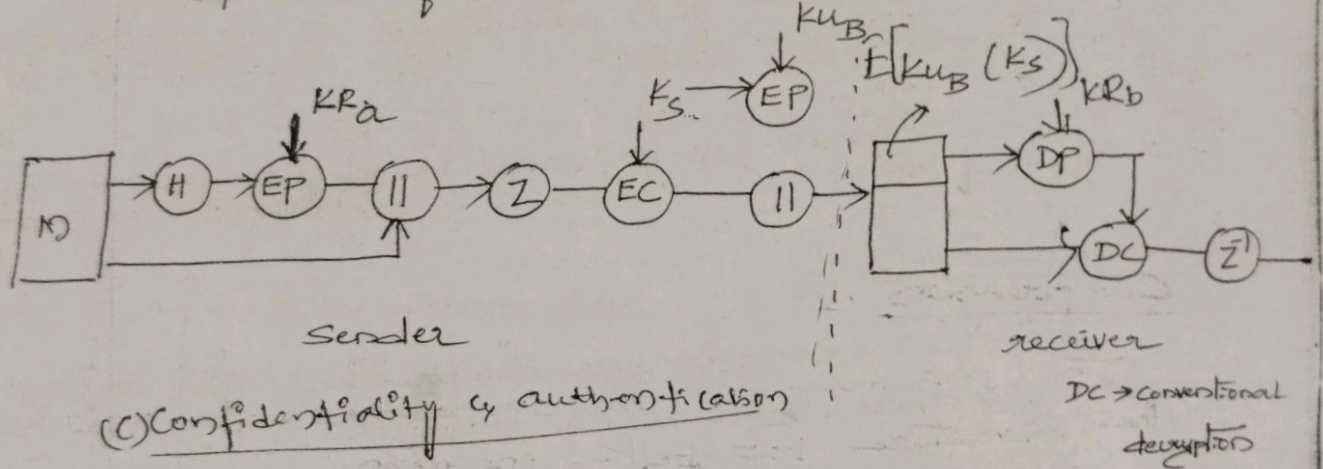
On the reception the incoming block is 1st converted from ASCII to binary by using RADIX64 format.

If the message is encrypted, the receiver recovers the session key & decrypts the message. Then the resulting block is decompressed.

If the message is signed the receiver recovers the transmitted hash code & compares it's own calculation of hash code.

The following diagram gives overview operation of PGP.

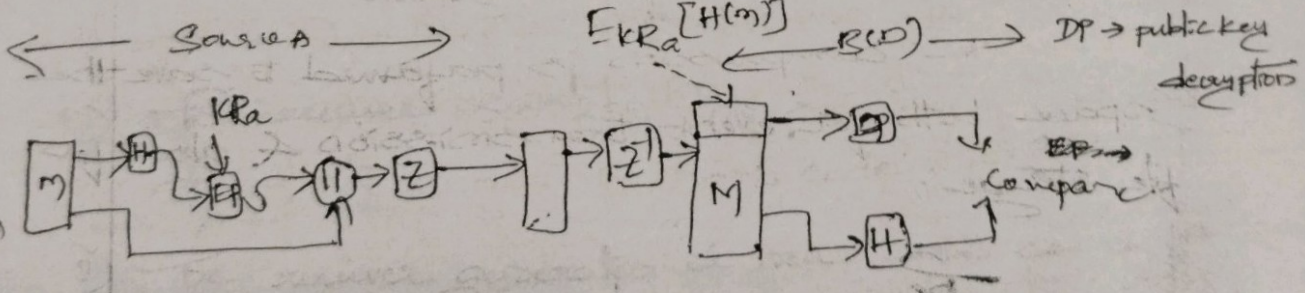
Pg no 35



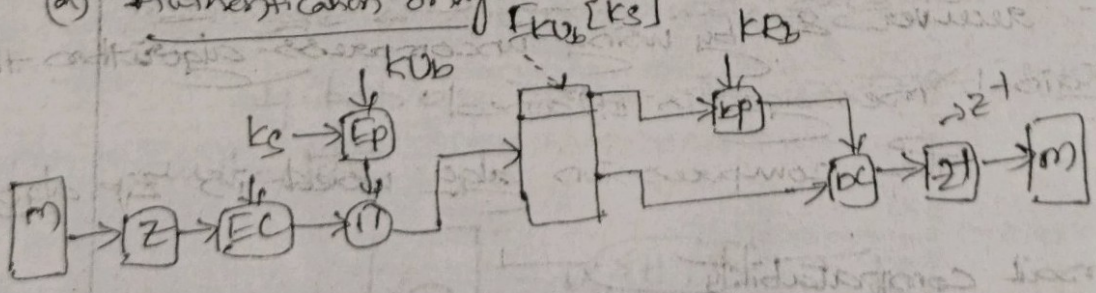
(c) Confidentiality & authentication

DC → conventional decryption

DP → public key decryption



(a) Authentication only



(b) Confidentiality only

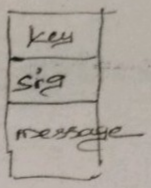
⇒ Segmentation & Reassembly :-

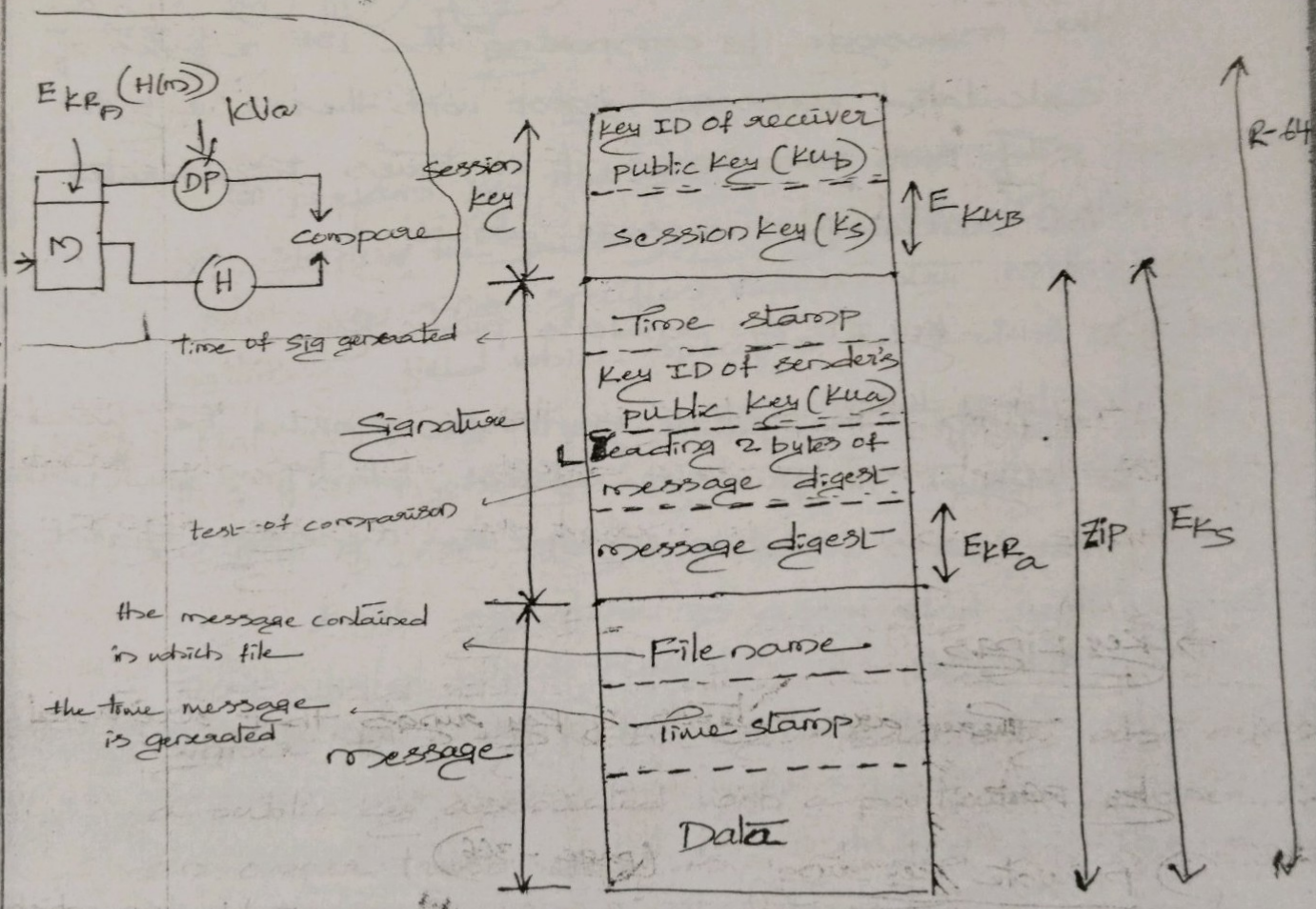
E-mail facilities are restricted to a maximum message length of upto 50,000 bytes. PGP allows subdividing the message into smaller segments & sending each segment seperatly.

At the destination these segments are reassembled & given as single message.

⇒ PGP message format :-

PGP contains 3 parts





PGP message contain 3 parts.

1. session key
2. Signature
3. message.

The session key component contains key ID of receiver's public key so that the corresponding private key is used for decrypting the session key.

The signature component contains,

1) Time stamp

specifies the time at which Sig was generated

2) message digest

160 bit message digest encrypted with sender's private key

3) Leading 2 bytes of message digest

To enable the recipient to determine if the correct public key was used to decrypt the message by comparing the 1st 2 bytes of calculated message digest with these 2 bytes. If both are same it continues to generate the remaining message digest.

4) Key ID of sender's public key

Identifies the public key that should be used to decrypt the message digest, identifies the private key that was used to encrypt the message digest.

⇒ Key Rings

There are 2 types of key rings that are used by PGP.

1) private key ring (page - 366)

one can view the ring as a table, in which each row represents one of public/private key pairs owned by the user, known as private key ring

The entries in private key ring contains time stamp, key ID, public key, encrypted private key & user ID.

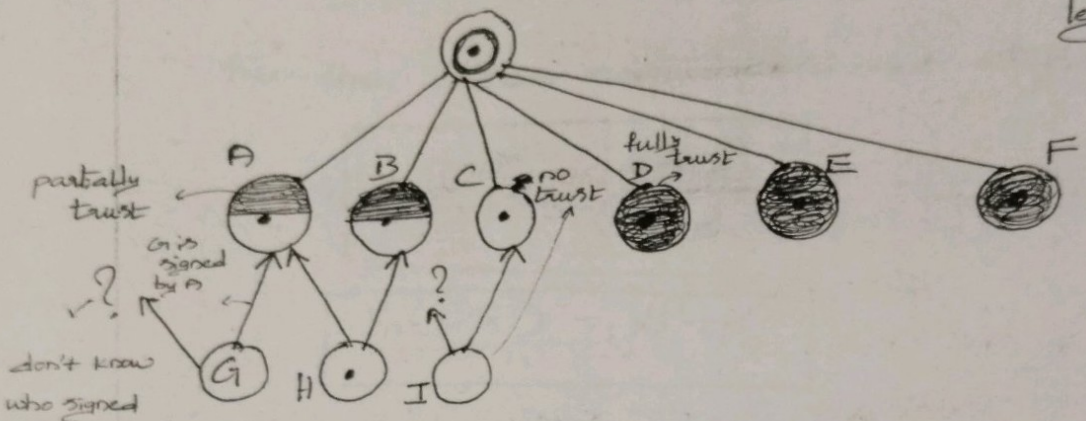
2.) public key ring :-

The public key ring contains all others public keys. The table contains timestamp, key ID, public key, owner trust, user ID, key legitimacy, signature & signature trust.

Extra (out of syllabus SMIME - page 374)

⇒ public key management :-

⊙ → key is deemed legitimated by you



To perform key management PGP uses owner trust for signing the public key certificates. In the public key ring specifies each entry with key-legitimacy field which specifies the extent to which PGP will trust that it is a valid public key for this user, higher the level of trust the stronger is binding of user Id.

Each signature is associated with signature trust field which indicates the user trust the signer to certify public keys. Each entry also defines a public key associated with a particular owner & an owner trust field is included which indicates the degree to which this public key is trusted to sign other public key certificates.

As shown in the diagram PGP trust model specifies fully trusted, partial trust, no trust & deemed legitimate because the certificate is signed by 2 people which are trusting the user.

Lucas Sequences :-

Select 2 integers P,Q, and also consider the quadratic equation $x^2 - Px + Q = 0$.

Assume that the roots of the quadratic eq are α, β then,

$$\alpha + \beta = P$$
$$\alpha\beta = Q$$

Then the Lucas sequences are defined as,

$$U_n(P,Q) = \frac{\alpha^n + \beta^n}{\alpha - \beta}$$

$$V_n(P,Q) = \alpha^n + \beta^n$$

$n = 0$ then,

$$V_0(P,Q) = \alpha^0 + \beta^0 = 2$$

$$U_0(P,Q) = \frac{\alpha^0 + \beta^0}{\alpha - \beta} = 0$$

$n = 1$ then,

$$V_1(P,Q) = \alpha + \beta = P$$

$$U_1(P,Q) = \frac{\alpha - \beta}{\alpha - \beta} = 1$$

We can rewrite the sequences as,

$$P(\alpha^{n-1} + \beta^{n-1}) - Q(\alpha^{n-2} + \beta^{n-2})$$

$$= \alpha^{n-2}(P\cdot\alpha + Q) + \beta^{n-2}(P\cdot\beta - Q)$$

$$= \alpha^{n-2}((\alpha+\beta)\cdot\alpha - \alpha\beta) + \beta^{n-2}((\alpha+\beta)\cdot\beta - \alpha\beta)$$

$$= \alpha^n + \beta^n$$

$$V_n(P,Q) = P V_{n-1}(P,Q) - Q V_{n-2}(P,Q)$$

Similarly,

$$U_n(P,Q) = P U_{n-1}(P,Q) - Q U_{n-2}(P,Q)$$

→ Calculate Lucas sequences for $P=3$, $Q=1$ for upto $n=3$.

Sol:- Given that, $P=3, Q=1$

The Lucas sequences $U_n(P,Q)$ & $V_n(P,Q)$ are calculated as,

$n=0$

$V_0(P,Q) = 2$

$U_0(P,Q) = 0$

$n=1$

$V_1(P,Q) = P = 3$

$U_1(P,Q) = 1$

$n=2$

By using recursive formula,

$$U_2(P,Q) = P \cdot U_1(P,Q) - Q\, U_0(P,Q)$$
$$= 3\times1 - 1\times0 = 3$$

$$V_2(P,Q) = P \cdot V_1(P,Q) - Q \cdot V_0(P,Q)$$
$$= 3\times3 - 1\times2 = 7$$

$V_{11} = 3\times18 - 1\times7$
$= 54 - 7$
$= 47$

$n=3$

$$U_3(P,Q) = P \cdot U_2(P,Q) - Q \cdot U_1(P,Q)$$
$$= 3\times3 - 1\times1 = 8$$

$$V_3(P,Q) = P \cdot V_2(P,Q) - Q \cdot V_1(P,Q) = 3\times7 - 1\times3$$
$$= 18$$

Properties of $V_n(P,Q)$:-

1) $V_n(P \bmod N, Q \bmod N) = V_n(P,Q) \bmod N$

2) Consider the Lucas sequence defined in terms of the roots of the equation as P & Q.

Assume, $P = V_k(P,Q)$

$Q = Q^k$

Then the quadratic equation is,

$$x^2 - V_k(P,Q) \cdot x + Q^k = 0$$

and the roots are, α' & β'.

Then, $\alpha' + \beta' = V_k(P,Q) = \alpha^k + \beta^k$ ($\because \alpha' = \alpha^k$, $\beta' = \beta^k$)

$\alpha'\beta' = Q^k$

$x^2 - Px + Q = 0$
$V_n(P,Q) = \alpha^n + \beta^n$
$\alpha + \beta = P$, $\alpha\beta = Q$
$\Rightarrow P = V_k(P,Q)$, $Q = Q^k$
$= \alpha^k + \beta^k$

The Lucas sequence based on these 2 integers can be written as,

$$V_n(V_k(P,Q), Q^k) = (\alpha^k)^n + (\beta^k)^n$$

$$= (V_k(P,Q))^n$$

$$= \alpha^{kn} + \beta^{kn}$$

$$= V_{nk}(P,Q)$$

Now assume that $Q=1$,

$$V_{nk}(P,1) = V_n(V_k(P,1), 1)$$

decryption encryption

By using above results, you can use Lucas sequence for public key cryptography as, by finding 2 integers e,d in such a way

$V_{de}(X \bmod N, 1) = V_{de}(X,1) \bmod N$ so that we can take a plain text of 'x' & encrypt the message as, $Y = V_e(X \bmod N, 1)$.

Then you can decrypt & obtain the P.T. as

$X = V_d(Y \bmod N, 1)$

Properties of $V_n(P,Q)$:-

1) $V_n(P \bmod N, Q \bmod N) = V_n(P,Q) \bmod N$

2) Consider the Lucas sequence defined interms of the roots of the equation as P & Q.

Assume, $P = V_k(P,Q)$

$Q = Q^k$

Then the quadratic equation is,

$$x^2 - V_k(P,Q) \cdot x + Q^k = 0$$

and the roots are, α' & β'.

Then, $\alpha' + \beta' = V_k(P,Q) = \alpha^k + \beta^k$ ($\because \alpha' = \alpha^k$, $\beta' = \beta^k$)

$\alpha'\beta' = Q^k$

$x^2 - Px + Q = 0$
$V_n(P,Q) = \alpha^n + \beta^n$
$\alpha + \beta = P$, $\alpha\beta = Q$
$\Rightarrow P = V_k(P,Q)$, $Q = Q^k$
$= \alpha^k + \beta^k$

The Lucas sequence based on these 2 integers can be written as,

$$V_n(V_k(P,Q), Q^k) = (\alpha^k)^n + (\beta^k)^n$$

$$= (V_k(P,Q))^n$$

$$= \alpha^{kn} + \beta^{kn}$$

$$= V_{nk}(P,Q)$$

Now assume that $Q = 1$,

$$V_{nk}(P,1) = V_n(V_k(P,1), 1)$$

(decryption, encryption)

By using above results, you can use Lucas sequence for public key cryptography as, by finding 2 integers e,d in such a way

$V_{de}(X \bmod N, 1) = V_{de}(X,1) \bmod N$ so that we can take a plaintext of 'X' & encrypt the message as, $Y = V_e(X \bmod N, 1)$.

Then you can decrypt & obtain the P.T. as

$$X = V_d(Y \bmod N, 1)$$

This is true by using second property.

$$V_{de}(X \bmod N, 1) = V_d(V_e(X \bmod N, 1), 1)$$

(finding the Lucas value for P.T. with respect to e
⇒ encryption

finding the Lucas value for C.T. with respect to
'd' ⇒ decryption)

$P = 3, e = 5, d = 7$

⇒ (C.T.) $C = V_5(3,1)$

$(= 123)$

C.T.

$V_5(3,1)$ — P.T.

public key

$P = V_7(123, 1)$ — C.T.

private key

$P \times q(p-q) - 4$

$3 \times 47 - 1 \times 15$

$= 141 - 15$

$= 123$

⇒ Lucas public key algorithm :-

The plain text is encrypted in the blocks, each block having a binary value less than the number 'N', encryption & decryption are,

$$C = V_e(P,1) \bmod N$$

$$P = V_d(C,1) \bmod N$$

where e, d are public key & private key.

The following procedure is used for Lucas public key cryptography.

Step1 :- Consider 2 large prime numbers P, q.

Step2 :- Calculate $n = P \times q$.

Step3 :- Select an integer 'e' as $\gcd((P-1)(q-1)(P+1)(q+1), e)$

Step4 :- Also calculate $D = P^2 - 4$.

Step 5:- Calculate,

$$\lambda(n) = LCM\left[\left(p-\frac{D}{p}\right), \left(q-\frac{D}{p}\right)\right]$$

Step 6:- Calculate

$$d = e^{-1} \bmod \lambda(n)$$

public key

$K_u = \{e, n\}$

K_R (private key) $= \{d, n\}$

Step 7 :- Encryption :

Assume that A to B

then (C to T) $C = V_e(m, D) \bmod N$

Decryption :-

$m = V_d(C, D) \bmod N$ ✓

RSA

1) p, q

2) $n = pq$

3) $\phi(n) = (p-D)(q-1)$

4) select d ⇒

$\gcd(d, \phi(n)) = 1$

5) $e = d^{-1} \bmod \phi(n)$

or $e \times d \bmod \phi(n) = 1$

$K_u = \{e, n\}$ & $K_R = \{d, n\}$

$C = m^e \bmod n$

$P = C^d \bmod n$

⇒ Privacy Enhanced mail (PEM)

(page - 664 to 667 Terminology page - 670 difference b/w PGP & PEM)

$(q+1), e) = 1$

Step5:- Calculate,

$$\lambda(n) = LCM\left[\left(p-\frac{D}{p}\right), \left(q-\frac{D}{p}\right)\right]$$

Step6:- Calculate

$$d = e^{-1} \bmod \lambda(n)$$

public key

$$K_u = \{e, n\}$$

K_a (private key) $= \{d, n\}$

Step7 :- Encryption:

Assume that a P.T. is 'M'

then (C.T.) $C = V_e(M, I) \bmod N$

Decryption :-

$M = V_d(C, I) \bmod N$. ✓

RSA

1) P, q

2) $n = pq$

3) $\phi(n) = (p-1)(q-1)$

4) select $d \Rightarrow$

$\gcd(d, \phi(n)) = 1$

5) $e = d^{-1} \bmod \phi(n)$

or $e \times d \bmod \phi(n) = 1$

$K_u = \{e, n\} \Rightarrow K_a = \{d, n\}$

$C = m^e \bmod n$

$P = C^d \bmod n$

⇒ Privacy Enhanced mail (PEM)

(page-666 to 669 Tenenbaum)

page-670 differences b/w PGP & PEM.

$(q+1), e) = 1$

Description	PGP → PGP	PEM
1) Supports encryption	Yes	Yes
2) Supports authentication	Yes	Yes
3) Supports Non-repudiation	Yes	Yes
4) Supports compression	Yes	NO
5) Supports Standardisation	NO	Yes
6) Supports mailing list	NO	Yes
7) Uses Base-64 coding	Yes	Yes
8) Data encryption algorithm	IDEA	DES
9) Username space	User defined	X.400
10) X.509 conformant (distribution of certificates by using X.509)	NO	Yes
11) Trust any one	NO	Yes
12) Key certification	Adhoc ↓ temporary	IPRA (Internet Registration)
13) Internet Standard	NO	Yes
14) Designed by	small team	Society (committee)

Unit-3

(page-303)

Authentication protocol

Unit-3

⇒ Authentication protocol :- (i)

1) Mutual authentication which uses conventional approach & key distribution centre

2) public key encryption approach which performs authentication by using public key techniques

3) one way authentication commonly used in e-mail system so that the recipient will authenticate the sender.

policy

Authority)

1, 14, 15, 27, 29, 35, 36, 37, 39, 42, 43, 49, 50, 53, 56, 58, 5

Unit - I

1) DES
2) Attacks, services, security model
3) conventional encryption principles.

short

1) Tripple DES & double DES
2) key distribution
3) crypt analysis techniques. (linear crypt analysis)
4) Bivalanche effect
5) random number generator

Unit - II

1) RSA
2) public key cryptography with security & authentication

short

3) Key management.
4) distribution of secret key using public key
5) Differences b/w conventional & public key encryptions.

Unit - III

1) Kerberos
2) Hash functions
3) Digital signatures (DSS)

short

1) Authentication protocols
2) X.509
3) Diffi - Helman

(Unit - IV)

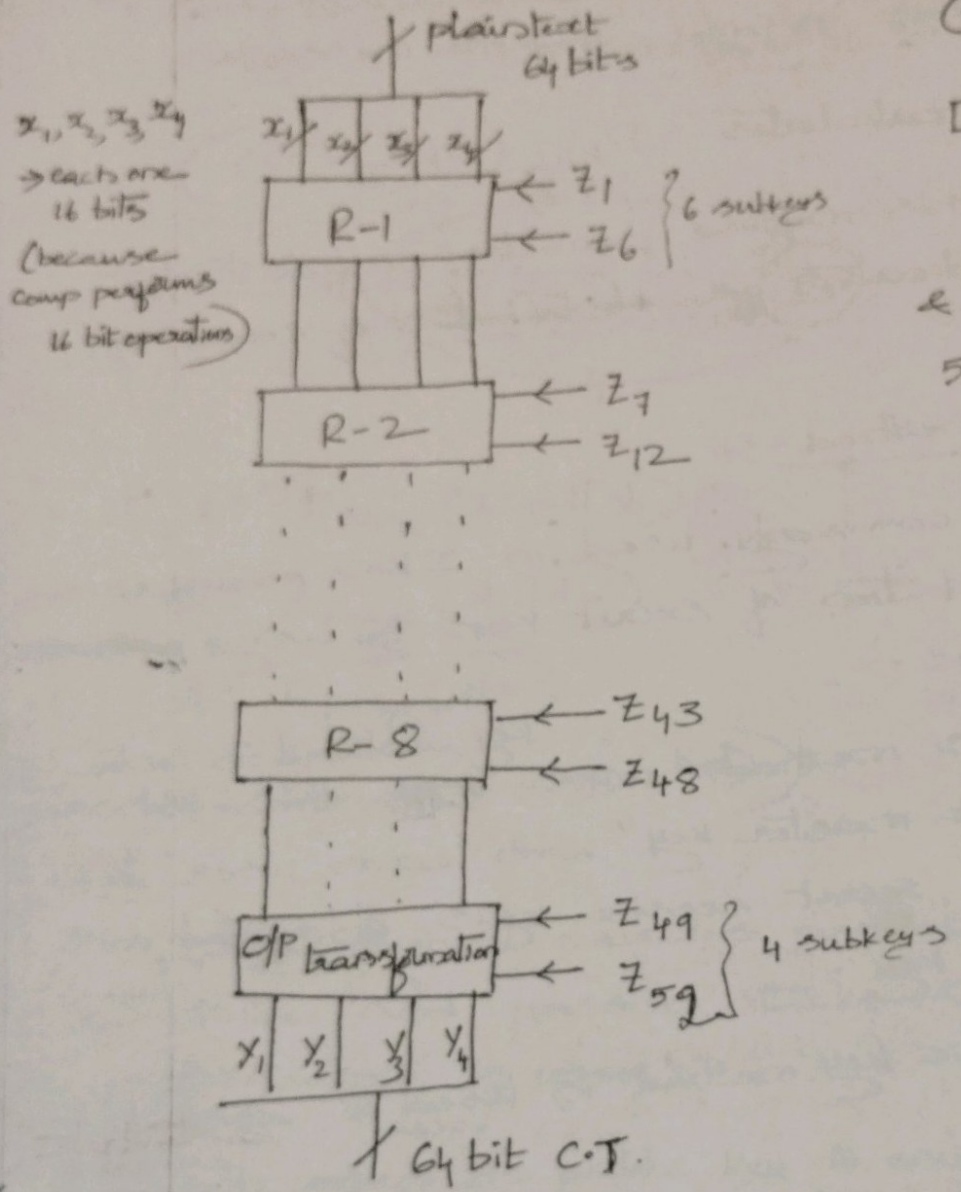
⇒ International data encryption algorithm:

(IDEA)

⊕ → EX-OR

⊙ → modulo multiplication

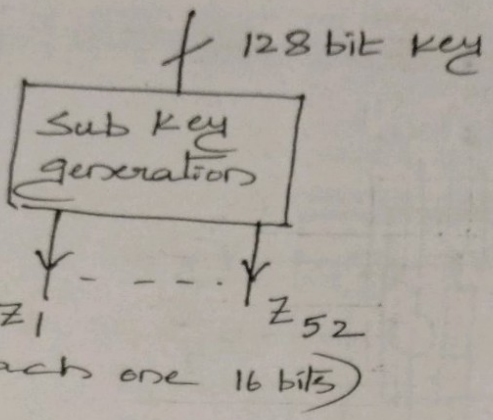
⊞ → modulo addition



Key → 128
& converting into
52 x 16 bits = 832 bits

Block diagram of IDEA encryption

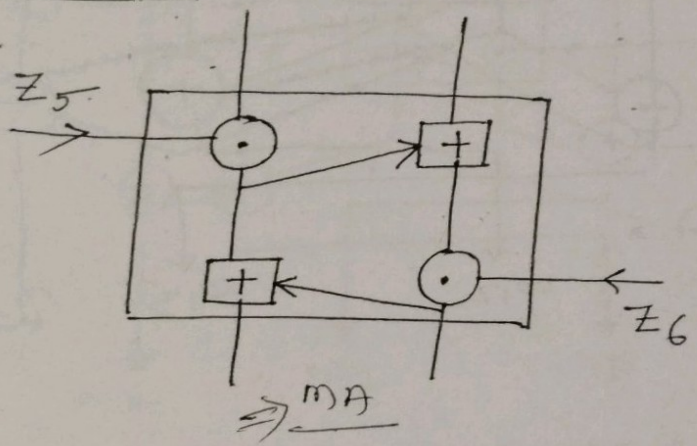
→



→ IDEA uses a special structure called multiplication addition (MA) to make more confusion during the encryption process.

MA Structure:

(2 i/ps & 2 o/ps)

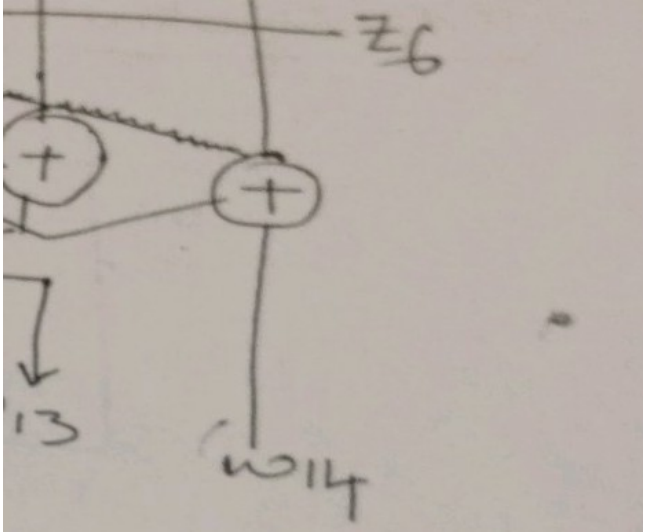


⇒ MA

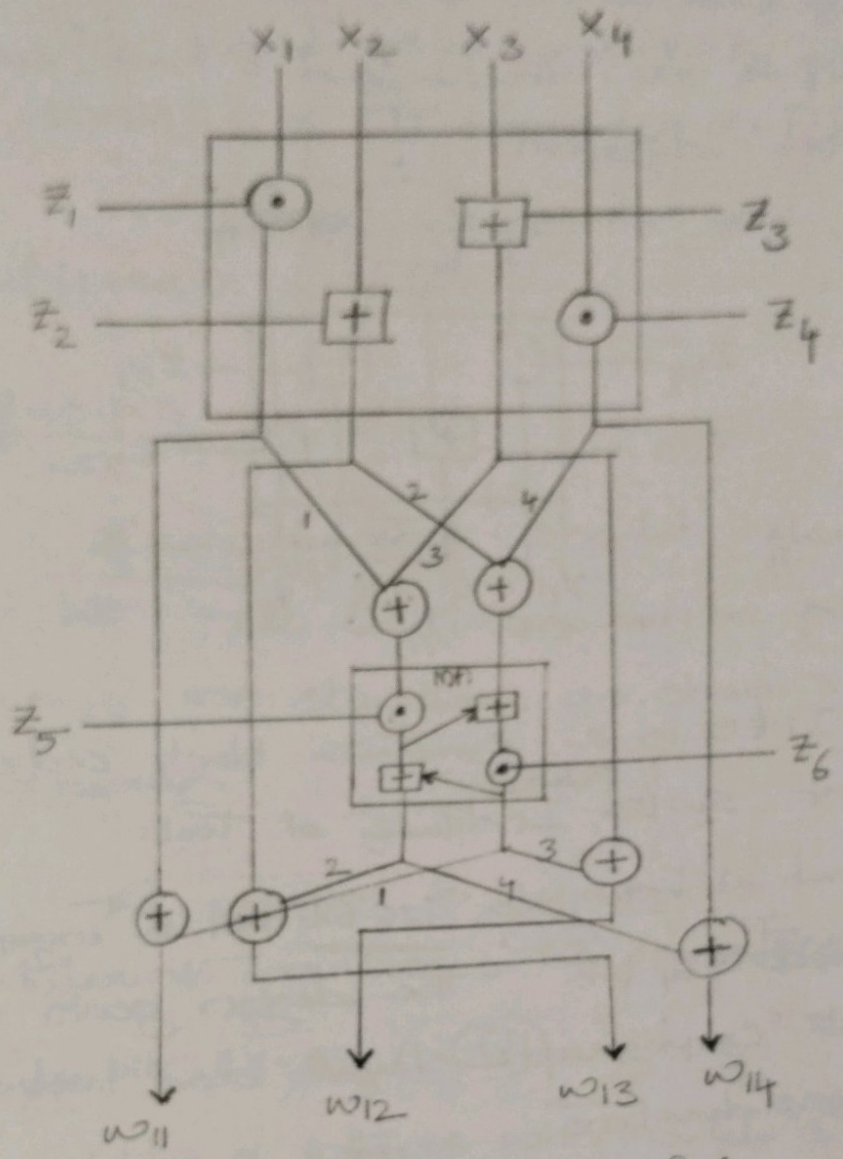
(for 2nd round Z_{11} & Z_{12} like this take the last 2 subkeys for every round)

→ Single iteration of IDEA encryption :-

P.T.O.

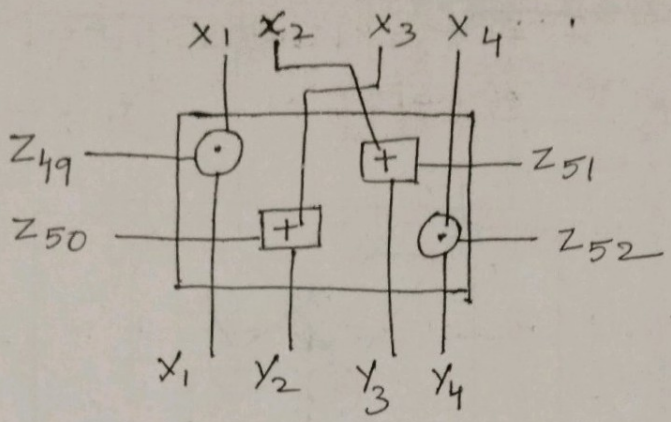


vided by the basic
known as (MA
(above previous fig
two 16-bit values
o 16-bit subkeys
duces two 16-bit



Single round of IDEA.

The o/p transformation is same as upperpart of each round but only difference is the 2nd & 3rd inputs are interchanged to have compatability during the decryption.



$x_1 - w_{81}$
$x_2 - w_{82}$
$x_3 - w_{83}$
$x_4 - w_{84}$

IDEA is a symetric block cipher developed at Swiss Federal Institute of Tech.

It uses 128 bit key to encrypt data in the blocks of 64 bits. The design goals are grouped into cryptographic strength and implementation considerations.

The cryptographic strength characteristics are

1) Block length

The length of the block (64 bits) is long enough to find statistical analysis. The complexity of algorithm increases exponentially with block size Therefore 64 bit block size is recognized as sufficiently strong.

2) Key length

128 Key length is sefficient to prevent brute force attacks.

3.) Confusion

It should make complex to determine the statistics of C.T. depending on P.T.

IDEA acheives this goal by using 3 different operations.

4.) Diffusion

Each plaintext bit should influence every C.T. bit. The diffusion can be provided by using MA structure as shown in the diagram.

The confusion is acheived by combining 3 different operations. The operations are,

(i) bit by bit EX-OR ($\oplus$)

(ii) an addition of integers with modulo 2^{16} ($\boxplus$)

(iii) multiplication of integers with modulo $2^{16}+1$ ($\odot$)

⇒ Encryption :-

There are 2 i/p's to the encryption fn, the P.T. to be encrypted (64 bits) & key which is 128 bits

IDEA algorithm consists of 8 rounds followed by o/p transformation. The algorithm divides the i/p into 4, 16 bit subblocks, each round takes 4, 16 bit subblocks & produces 4, 16 bit o/p blocks.

Each round also takes 6, 16 bit subkeys whereas the final transformation takes 4 subkeys with a total of 52 subkeys.

Details of single round :—

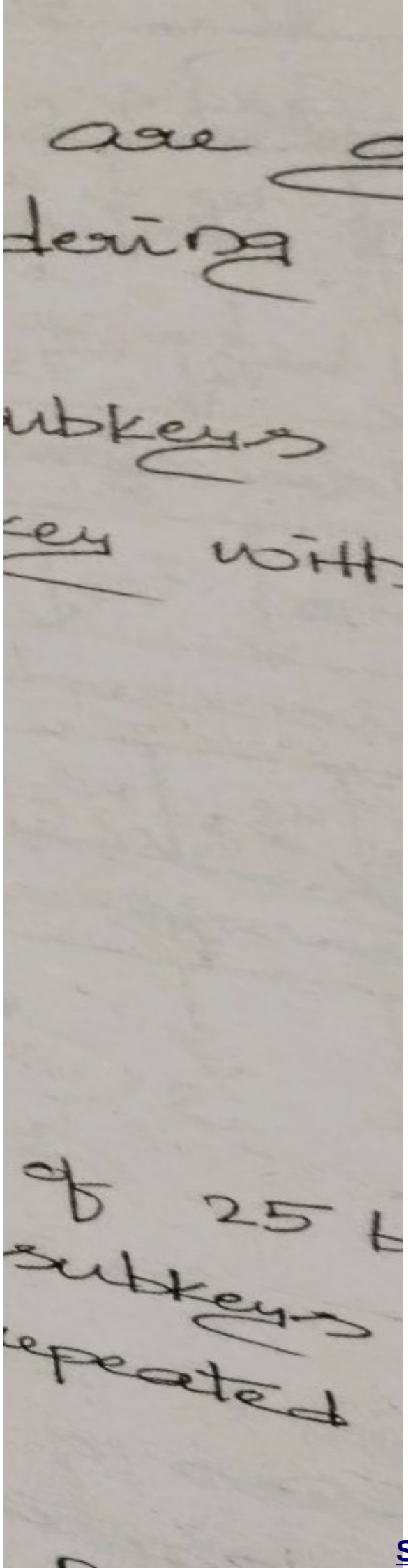
The round begins with a transformation that combines the 4 i/p subblocks with 4 subkeys using addition & multiplication operation. The 4 o/p blocks of this transformation are combined by using EX-OR operation to generate 2, 16 bit blocks which are i/p to MA structure. The MA structure also takes 2 subkeys as i/p & combines these i/ps to produce 2, 16 bit o/ps.

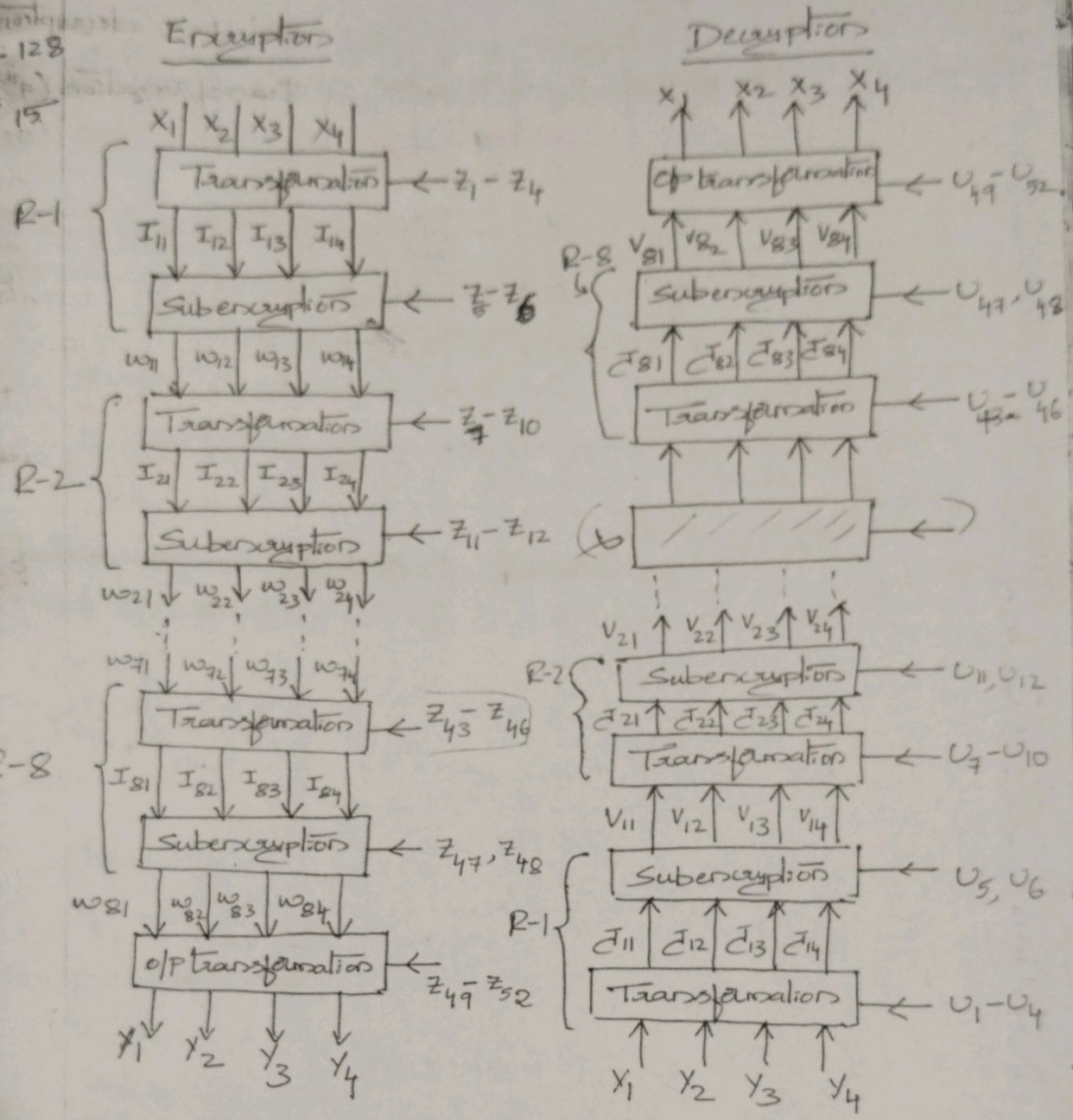
Finally the 4 o/p blocks from the upper transformation are combined with the 2 o/p blocks of MA structure by using EX-OR operations to produce the 4 o/p blocks for each round. The 2 o/ps that are produced by 2nd & 3rd i/ps are interchanged to produce as 3rd & 2nd o/p's.

The 9th stage of the algorithm is the o/p transformation which is same as upper portion of preceeding round except that 2nd & 3rd i/p's are interchanged.

⇒ Key generation

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
1---16	17--32	33---48	49--64	65---80	81---96	97--112	113--128





$(V_{11} = I_{81}, V_{12} = I_{82}, I_{11} = w_{81}$ ---)

The process of decryption is same as encryption process, decryption is acheived by giving the C.T. as I/P to the same structure of encryption, but with a different selection of subkeys. The decryption subkeys are U_1 ---- U_{52} which are derived from the encryption keys as,

Step1:- The 1st 4 subkeys of decryption round-i are derived from 1st 4 subkeys of encryption round (10-i)

For example, for round-1 of decryption 1st 4 keys are same as o/p transformation (9th round 10-1=9) 4 keys of encryption. i.e.

$$U_1 = Z_{49},\ U_2 = Z_{50},\ U_3 = Z_{51},\ U_4 = Z_{52}.$$

But for round-2 to 8 the 2nd & 3rd decryption subkeys are equal to 3rd & 2nd encryption subkeys, for rounds 1 & 9, 2nd & 3rd decryption subkeys are same as 2nd & 3rd encryption subkeys

Step2 :- For the 1st 8 rounds the last 2 subkeys of decryption round-i are equal to the last 2 subkeys of encryption round (9-i).

1st round $\begin{cases} U_5 = Z_{47} \\ U_6 = Z_{48} \end{cases}$

2nd round

$U_7 = Z_{43}$
$U_8 = Z_{45}$
$U_9 = Z_{44}$
$U_{10} = Z_{46}$

formula is, (10-i) (⇒ 10-2 = 8)

because for 2 to 8 rounds the 2nd & 3rd subkeys are interchanged

$U_{11} = Z_{41}$
$U_{12} = Z_{42}$

formula is (9-i) (Here 9-2 = 7)

For decryption multiplicative inverse & additive inverse are used.