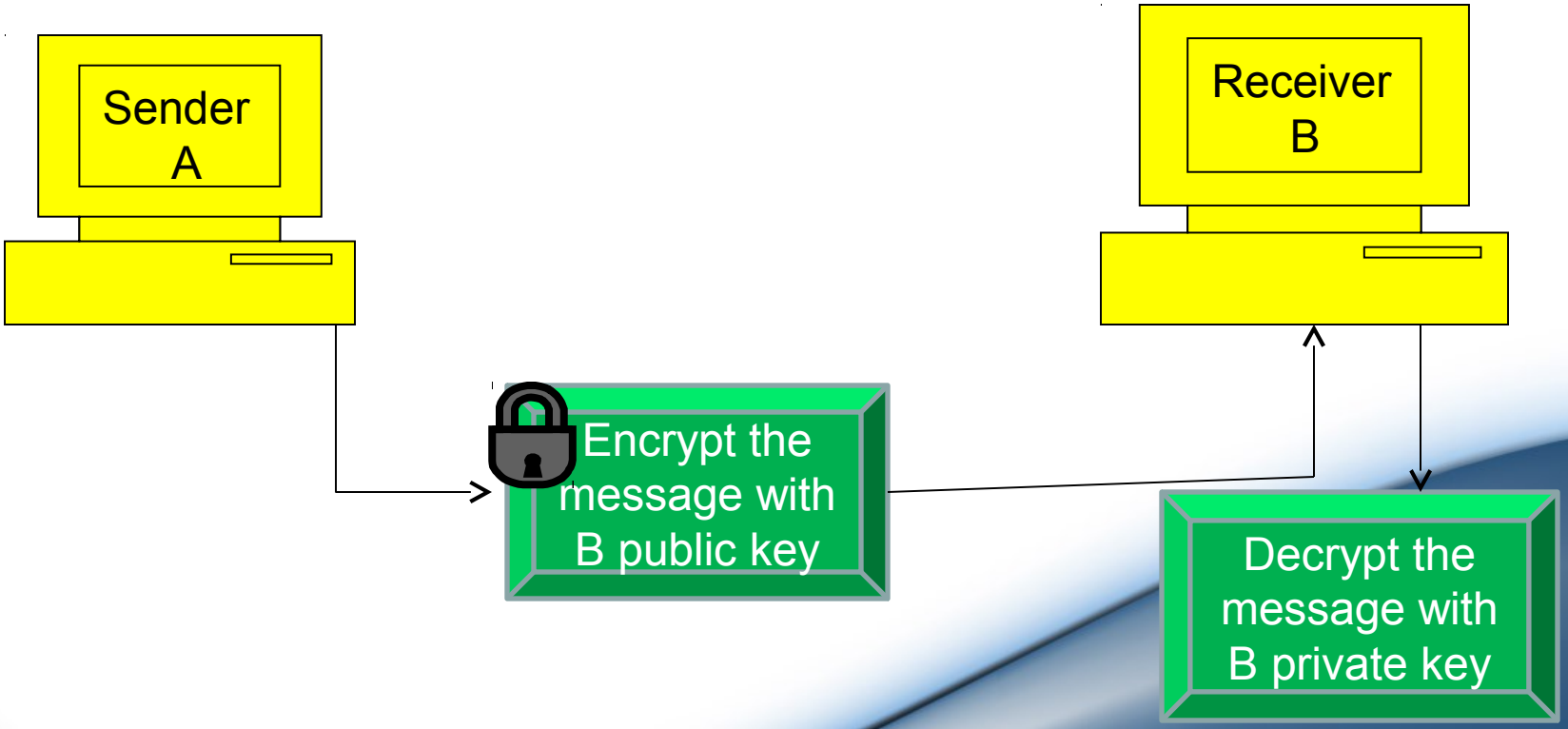
Network Security
Unit-III

ASymmetric Key Algorithms

Overview of asymmetric –key cryptography



One key is used for **locking**. Other is used for **unlocking**. **Public key** is used for locking. **Private key** is used for unlocking.

RSA Algorithm

- Invented by Rivest, Shamir & Adleman of MIT in 1977
- Choose two large prime numbers P and Q
- Calculate N=PxQ
- Select Public key E as not a factor of (P-1)x(Q-1)
- Select Private key D as (DxE) mod (P-1)x(Q-1)=1
- For Encryption, $CT=PT^E \bmod N$
- Send CT to the receiver
- For Decryption, $PT=CT^D \bmod N$

Example:

- P=7 and Q=17
- Calculate N=7x17=119
- Select Public key E as not a factor of (P-1)x(Q-1) i.e.
- (7-1)x(17-1) = 6x16 = 96.
- The factors of 96 are: 2,3,4,15,6.
- So, let us say, we keep E = 5
- Select Private key D as (DxE) mod (P-1)x(Q-1)=1
- (Dx5) mod (7-1)x(17-1)=1
- (Dx5) mod 6x16=1.
- After some calculations, we say D=77
- For Encryption, $CT=PT^E \bmod N$. let us say PT=10
- $CT=10^5 \bmod 119 = 40$
- Send CT to the receiver
- For Decryption, $PT=CT^D \bmod N$
- $PT= 40^{77} \bmod 119 = 10$

Elgamal Cryptography

- The **ElGamal encryption system** is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie-Hellman Exchange Agreement.
- The system parameters consist of a prime ***p*** and an integer ***g***, whose powers modulo *p* generate a large number of elements, as in Diffie-Hellman.
- A has a private key ***a*** and a public key ***y***, where $y = g^a \pmod p$.
- Suppose B wishes to send a message ***m*** to A. B first generates a random number ***k*** less than ***p***.
- He then computes :-

$$y_1 = g^k \pmod p \text{ and } y_2 = m \text{ xor } y^k,$$

B sends (y_1 ,y_2) to A.

Upon receiving the ciphertext,

A computes :-

- $$m = (y_1^a \bmod p) \text{ xor } y_2 .$$

Elliptic curve algorithm

- Elliptical curve cryptography (ECC) is a public key encryption technique based on *elliptic curve theory* that can be used to create faster, smaller, and more efficient cryptographic keys.

- ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.

- ECC was introduced by Victor Miller and Neal Koblitz in 1985.
- It's new approach to Public key cryptography.
- ECC requires significantly smaller key size with same level of security.
- Benefits of having smaller key sizes : faster computations, need less storage space.
- ECC ideal for : Pagers ; PDAs ; Cellular Phones ; Smart Cards.

Elliptic Curves

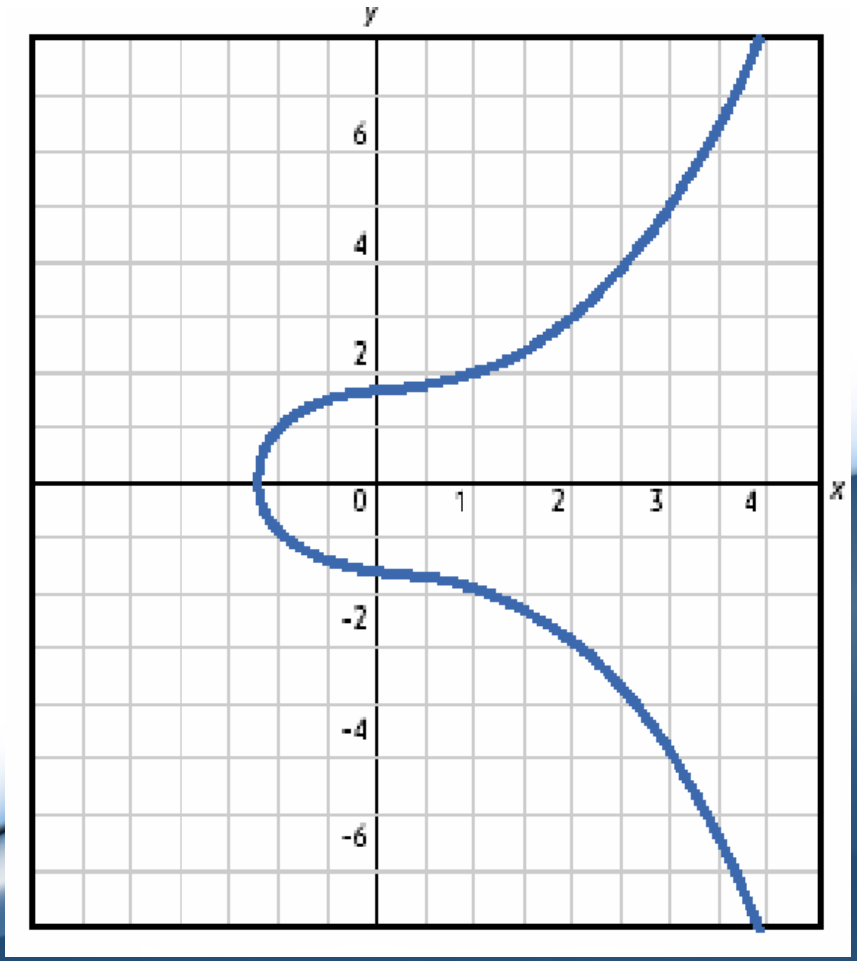
- Standard Form Equation

  $y^2 = x^3 + a.x + b$

  where x, y, a and b are

  real numbers.

- x,y,a form the public key whereas b forms a private key.



KNAPSACK ALGORITHM

- That is, if M1, *M2, ..., Mn are the given values and S is the sum, find out bi so that:*
- *S = b1M1 + b2M2 + ... + bnMn*
- Each *bi can be 0 or 1.*
- 1 indicates that the item is in the knapsack
- 0 indicates that it is not

Plain text	0 1 1 0 1 1	1 1 1 0 0 0	0 1 0 1 1 0
Knapsack	1 7 8 12 14 20	1 7 8 12 14 20	1 7 8 12 14 20
Cipher text	7 + 8 + 14 + 20 = 49	1 + 7 + 8 = 16	7 + 12 + 14 = 33

Message Digest

- A message digest is a fingerprint or the summary of a message.
- It is used to verify the integrity of data to ensure that the data is not altered in any way.

MD5

- Steps:

1.) Padding (64 bits less than a multiple of 512)

2.) Append Length (of length 64 bits)

3.) Divide the input into 512-bit blocks

4.) Initialize chaining variables

5.) Process Blocks

5.1> copying chaining variables into temporary variables

5.2> Divide current 512 block into 16 subblocks of 32-bit

5.3> perform 4 rounds

Chaining variables :-

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

Each of A, B, C, D is a 32-bit register.

Functions used in 4 rounds:-

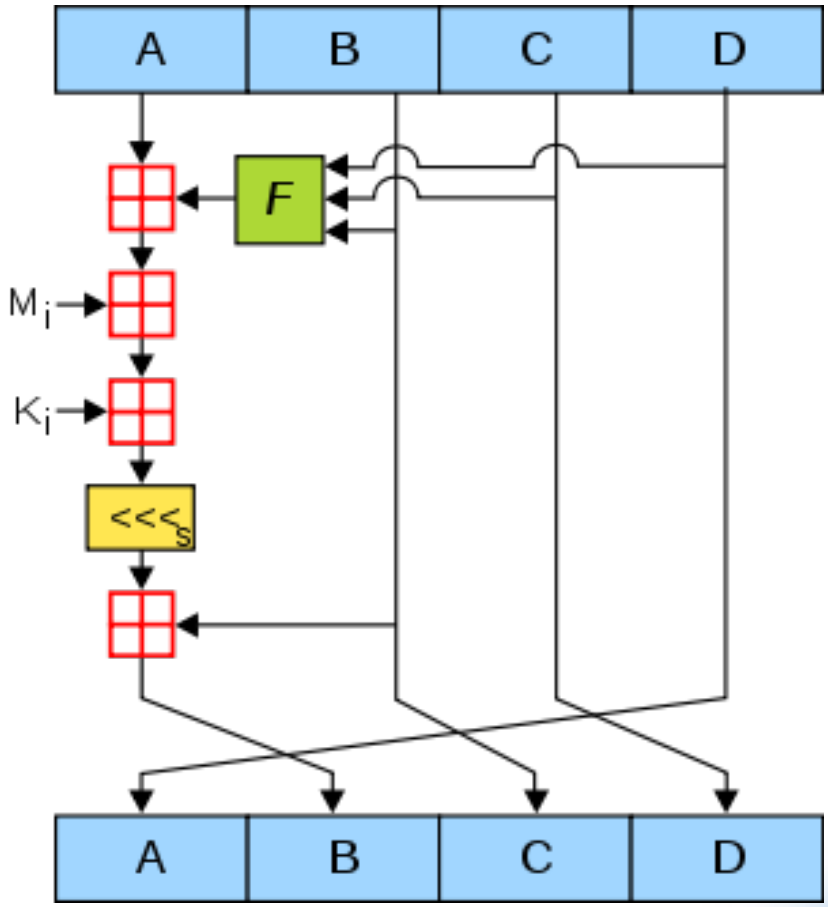
F(X,Y,Z) = (X & Y) | (~(X) & Z)

G(X,Y,Z) = (X & Z) | (Y & ~(Z))

H(X,Y,Z) = X ^ Y ^ Z

I(X,Y,Z) = Y ^ (X | ~(Z))

Four functions will be defined such that each function takes an input of three 32-bit words and produces a 32-bit word output.



A mathematical expression of a single MD5 operation:
a = b + ((a + F (b, c, d) + X [k] + T [i]) <<< s).

Secure Hash Algorithm (SHA)

- SHA works with any input message that is less than 2^{64} *bits in length.*
- *The output of SHA is a message* digest, which is 160 bits in length (32 bits more than the message digest produced by MD5).
- The word *Secure in SHA was decided based on two features. SHA is designed to be computationally infeasible to*
  - obtain the original message, given its message digest, and
  - find two messages producing the same message digest.

The Working of SHA

Step 1: Padding : such a way that the length of the message is 64 bits short of a multiple of 512

Step 2: Append Length: appended to the end of the padding as a 64-bit block

Step 3: Divide the Input into 512-bit Blocks: each of length 512 bits

Step 4: Initialize Chaining Variables: five *chaining variables A through E are initialized*

Step 5: Process Blocks:

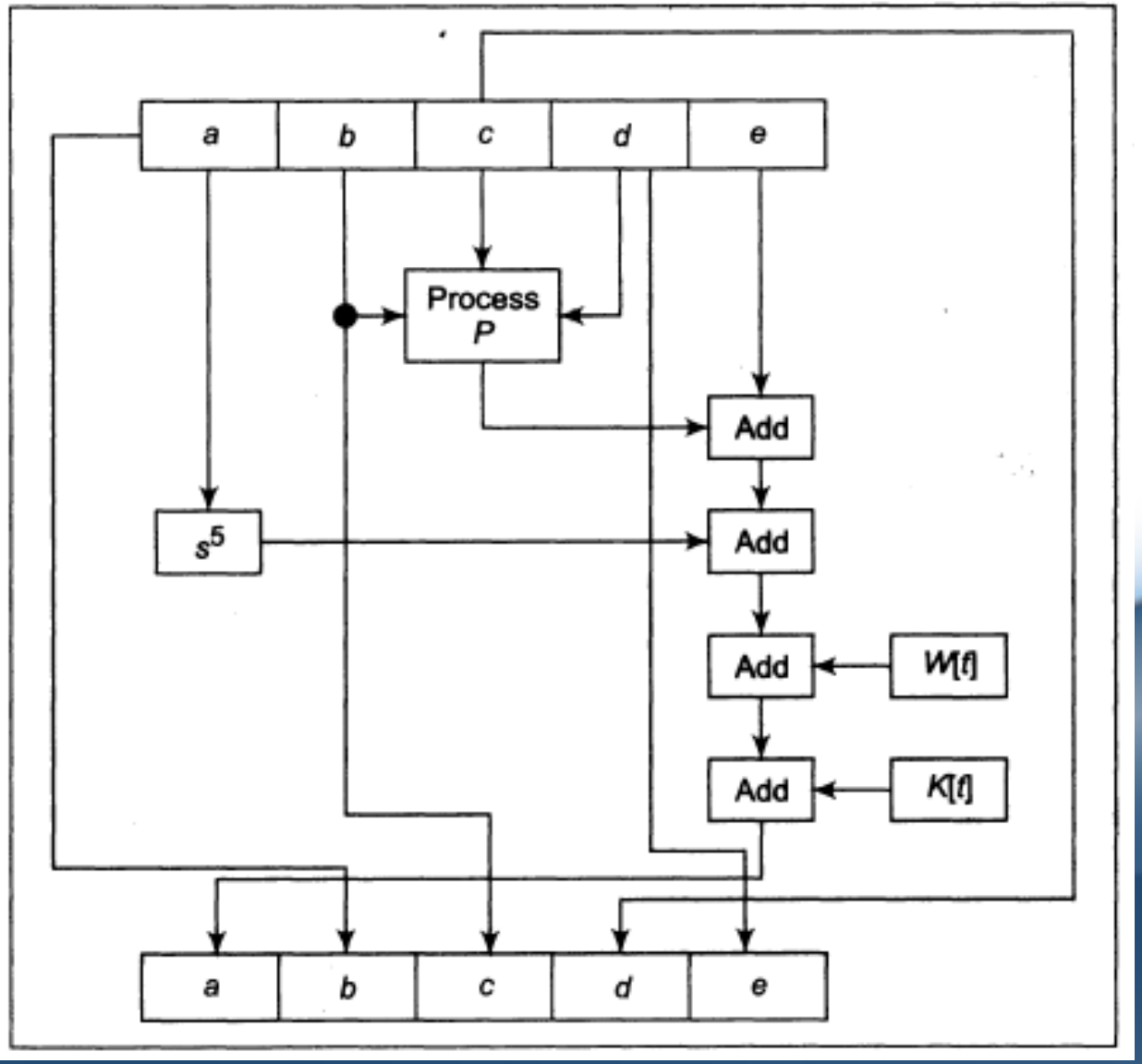
Step 5.1 Copy the chaining variables A-E into variables a-e.

Step 5.2 Now, divide the current 5 12-bit block into 16 sub-blocks, each consisting of 32 bits.

Step 5.3 SHA has four rounds, each round consisting of 20 steps.

Each round takes the current 512-bit block,
the register *abcde and a constant K[t] (where t = 0 to 79) as the three inputs.*

Single SHA-1 iteration



Mathematically, an iteration consists of the following operations:
abcde = (e + Process P + s5 (a) + W[t] + K[t]), a, s30 (b), c, d
where,
abcde = The register made up of the five variables a, b, c, and e
Process *P* =

Round	Process P
1	(b AND c) O R ((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

st = Circular-left shift of the 32-bit sub-block by t bits
W[t] = A 32-bit derived from the current 32-bit sub-block,
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*
W[t] = s1 (W [t- 16] XOR W[t - 14] XOR W[t - 8] XOR W[t - 3])
s1 indicates a circular-left shift BY 1 bit
K[t] = One of the five additive constants.

SHA-512

The SHA-512 algorithm takes a message of length 2128 bits, and produces a message digest of size 512 bits. The input is divided into blocks of size 1024 bits each.

Step 1: Padding : Like MD5 and SHA-1, the first step in SHA is to add padding to the end of the original message

Step 2: Append Length : length of the message is exactly a multiple of 1024 bits

Step 3: Divide the Input into 1024-bit Blocks

Step 4: Initialize Chaining Variables :Now, eight *chaining variables, a through h, are initialized.*

A = 6AQ9E661F3BCC908	B = BB67AE85S4CAA73B
C = 3C6EF372FE94F82B	D = A54FF53A5F1D36F1
E = 51QE521FADE6B2D1	F -9B05688C2B3E6C1F
G = 1F83D9ABFB41BD6B	H = 5BE0CD19131E2H9

Step 5: Process Blocks

Step 5.1 Copy the chaining variables *A-H into variables a-h. The combination of a-h, called abcdefgh*

Step 5.2 Now, divide the current 1024-bit block into 16 sub-blocks, each consisting of 64 bits.

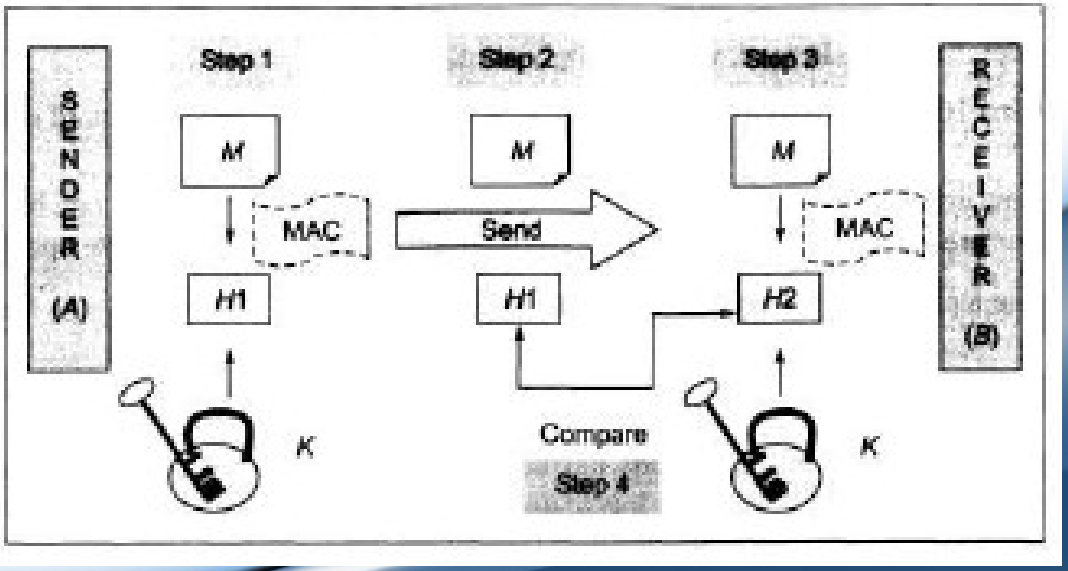
Step 5.3 SHA-512 has 80 rounds. Each round takes the current 1024-bit block, the register *abcdefgh* and a constant *K[t] (where t = 0 to 79) as the three inputs.*

Message Authentication Code (MAC)

The concept of **Message Authentication Code (MAC) is quite similar to that of a message digest.** However, there is one difference. As we have seen, a message digest is simply a fingerprint of a message.

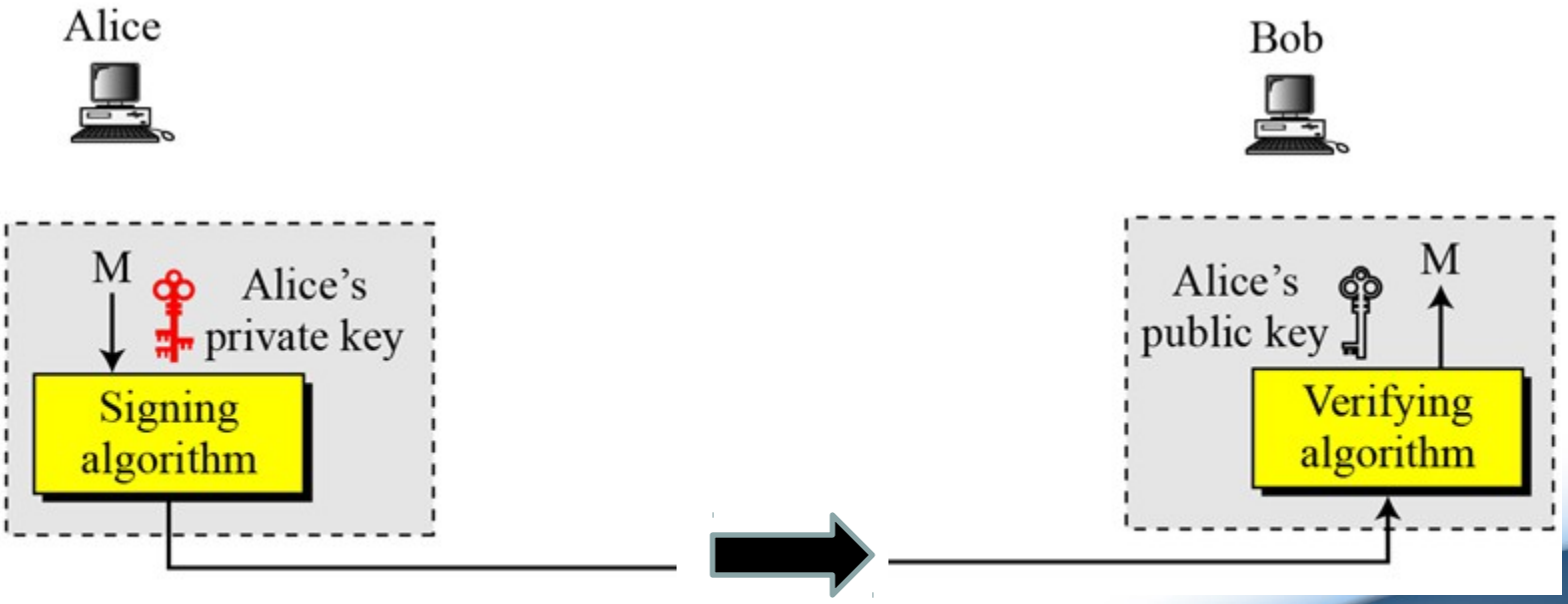
There is no cryptographic process involved in the case of message digests.

In contrast, a MAC requires that the sender and the receiver should know a shared symmetric (secret) key, which is used in the preparation of the MAC. Thus, MAC involves cryptographic processing.



Digital Signature

- To provide Authenticity, Integrity and Non -repudiation to electronic documents
- A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document that has been sent is unchanged.
- Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped

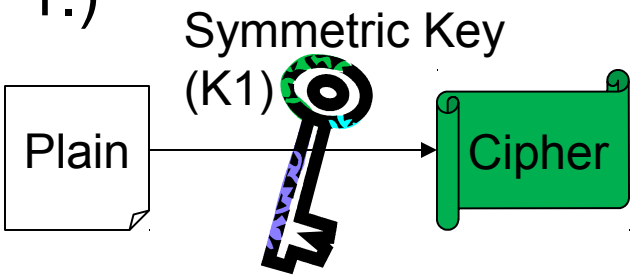

Alice
Bob
M
Alice's
private key
Signing
algorithm
Alice's
public key
M
Verifying
algorithm

Symmetry and Asymmetry key together

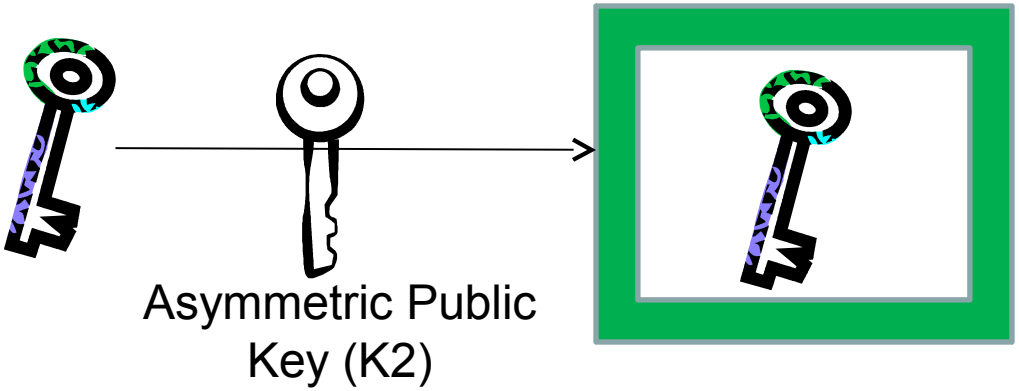
- In practice, symmetric & asymmetric key cryptography are combined to have a very efficient security solution.

Sender's side

1.)

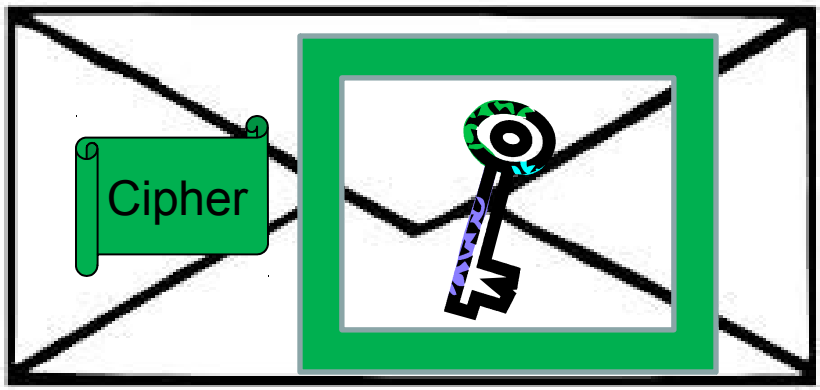


2.)



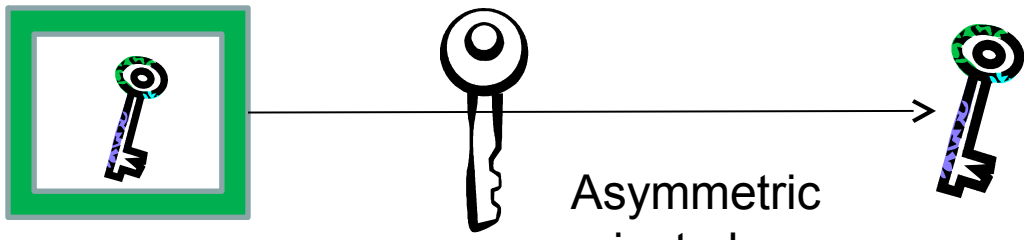
This process is called as "**Key Wrapping**"

3.)

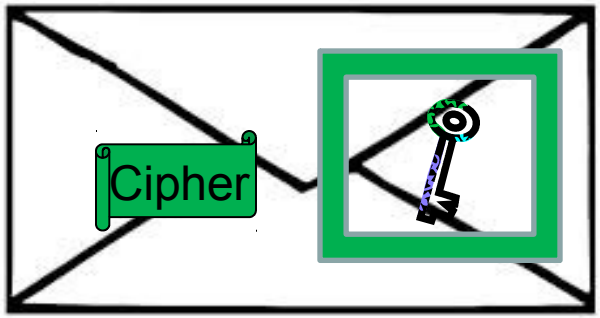


This is called as **Digital Envelope**

Receiver's side



Asymmetric private key (K3)



Symmetric key (K1)

Problems with public key exchange

Alice
Public key =17

Tom

A=20, B=13

Bob

Public key=17

Public key=20

Intercept

Public key=17

Public key=13

Public key=17

Intercept

Thank You