Avalanche effect:

The desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext. In particular a change in one bit of plaintext or one bit of the key should produce a change in atleast half of many bits of the cipher text. This effect is known as avalanche effect.

Block cipher design principles:

1) No of rounds
2. Design of function F
3 Keyschedule alg.

strict avalanche criteria.
bit independent criteria.

UNIT - 2

→ Des
→ Diff
→ Engine
→ fig

Scanned by CamScanner

UNIT-2

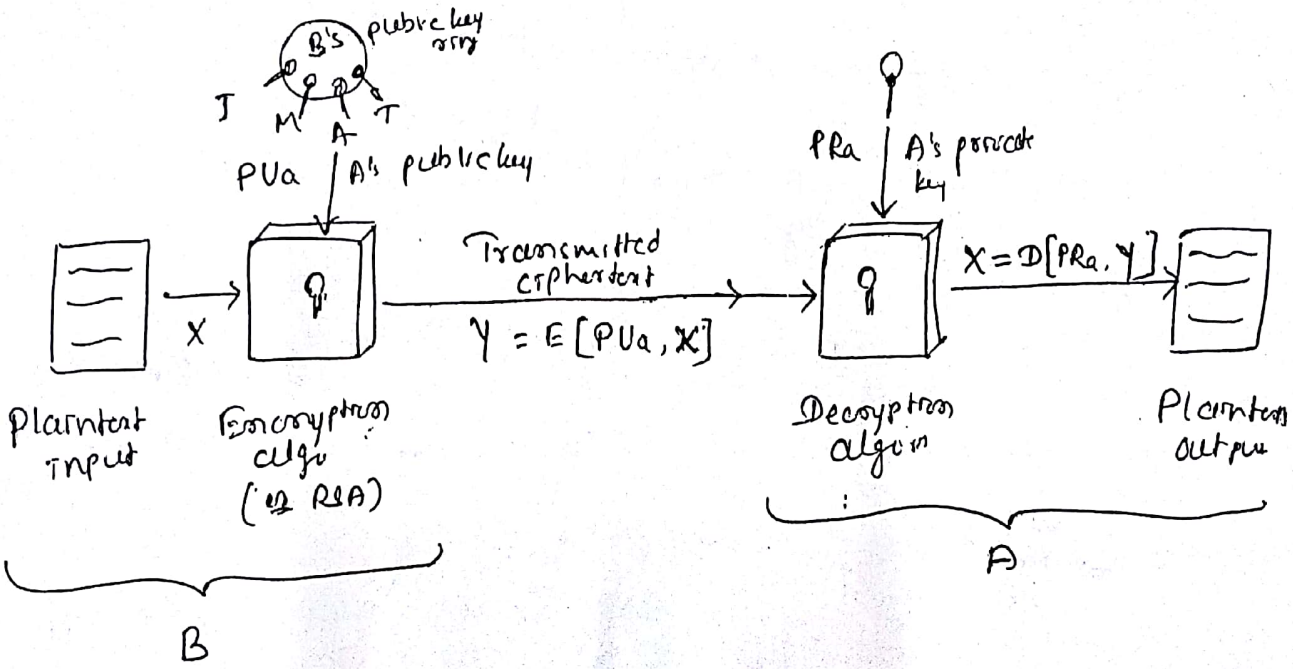
Public Key Cryptography and Message Authentication

* Public key cryptographic principles

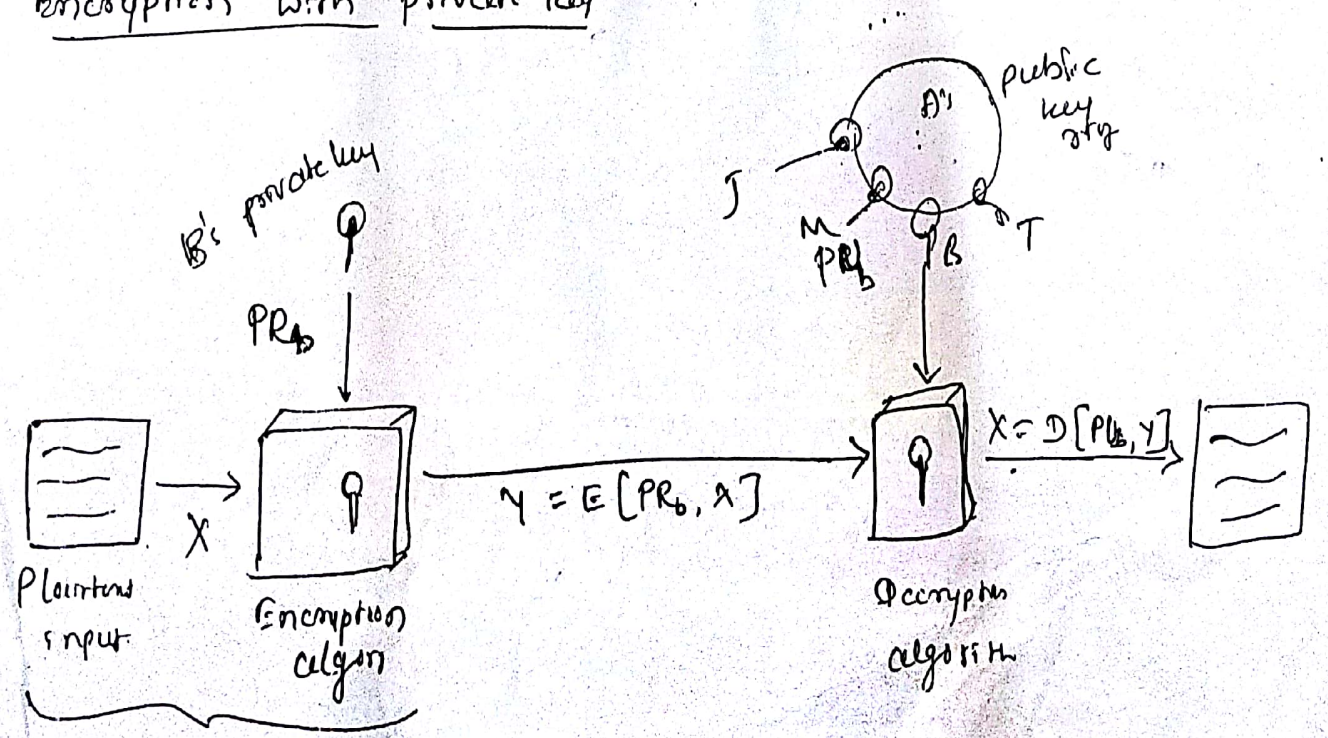
Slide-4th

Public key encryption structure :- Public key algorithms are based on mathematical functions rather than on simple operations on bit patterns.

Encryption with public key



Encryption with private key



Scanned by CamScanner

A public key encryption scheme has 6 ingredients

* Plaintext : This is the readable message or data that is fed into the algorithm as input
* Encryption algorithm : The encryption algorithm performs various transformations on the plaintext
* Public and Private keys : This is a pair of keys that have been selected so that one is used for encryption, the other is used for decryption. The exact transformation performed by the encryption algorithm depend on the public or private key that is provided as input.
* Ciphertext : This is the scrambled message produced as ou It depends on the plaintext and the key.
* Decryption algorithm : This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The public key of the pair is made public for others to use, while the private key is known only to its owner A general purpose public key cryptographic algorithm relies on one key for encryption and a different but related key for decryption.

The essential steps are the following.

1) Each user generates a pair of keys to be used for the encryption and decryption of messages
2) Each user places one of the two keys in a public register or other accessible file ie public key. The companion k is kept private. In the first figure each user maintains

Scanned by CamScanner

3) If B wishes to send a private message to A, B encrypts the message using A's public key.

4) When A receives the message, A decrypts it using private key. No other recipient can decrypt the message because only A knows A's private key.

As long as a user protects his or her private key incoming communication is secure. At any time a user can change the private key and publish the companion public key to replace the old public key.

Applications for Public-key Cryptosystems

Public key cryptosystems can be categorized into 3 types

* Encryption / Decryption: The sender encrypts a message with the recipient's public key.
* Digital signature: The sender signs a message with its private key. Signing is achieved by cryptographic algorithm applied to a small block of data or message.
* Key exchange: Two sides cooperate to exchange a session key.

Some algorithms are suitable for all 3 application, whereas others can be used only for one or 2 of these application.

Algorithm	Encryption/Decryption	Digital signature	Key exchange
RSA	✓	✓	✓
Diffie Hellman	X	X	✓
DSS	X	✓	X
Elliptic curve	✓	✓	✓

Scanned by CamScanner

Requirements for public-key cryptography

Diffie Hellman postulated it as DIFF & H

The algorithm must fulfill the following points

1) It is computationally easy for a party B to generate a pair (Public key PU_b, Private key PR_b)

2) It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext.

$$C = E(PU_b, M)$$

3) It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message.

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

4) It is computationally infeasible for an opponent, knowing the public key PU_b to determine the private key PR_b

5) It is computationally infeasible for an opponent knowing the public key, PU_b and a ciphertext C to recover the original message M.

By adding one more requirement

6) Either of the two related keys can be used for encryption with the other for decryption

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

i.e. $M = D[PU_b, E(PR_b, M)]$ ⇒ encrypting using Private key

$M = D[PR_b, E(PU_b, M)]$ ⇒ " " Public key.

Scanned by CamScanner

* Public key Cryptographic Algorithms (RSA, Diffie Hel...

RSA Public key encryption algorithm

stall 4th

RSA was developed in 1977 by Ron Rivest, Adi Shamir and Len Adleman at MIT and first published in 1978.

RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.

The RSA algorithm can be written as follows,

Key generation

large prime nos to make RSA Robust

Select p, q	p and q both prime, $p \neq q$
Calculate	$n = p \times q$
calculate	$\phi(n) = (p-1) \times (q-1)$
Select integer e	$\gcd(\phi(n), e) = 1 \; ; 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption

Plaintext	$M < n$
ciphertext	$C = M^e \bmod n$

Decryption

Ciphertext	C
Plaintext	$M = C^d \bmod n$

Scanned by CamScanner

wrt RSA encryption and decryption are of the following form period for some plaintext block M and ciphertext block C.

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Both sender and receiver must know the values of n and e and only the receiver knows the value of d

Hence RSA is a public key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$.

example 1

[select e and d] through inspection

① $p = 3 \quad q = 11$

$n = p \times q = 3 \times 11 = 33$

$\phi(n) = (p-1) \times (q-1) = 2 \times 10 = 20$

Choose e such that $1 < e < \phi(n)$ e and n are coprime

$gcd(\phi(n), e) = 1 \Rightarrow gcd(20, 7) = 1$

$e = 7$

② $0 < e < 3$

$de \bmod \phi(n) = 1 \Rightarrow d \times 7 \bmod 20 = 1$

$d = 3$

Note:-
During the selection of e ; Min value of e = 3
∵ e = 2 is not prime (2 is the only even prime)
∵ p & q are prime
(p-1) * (q-1) is even
∴ 0.
Realistic $e = 2^{16} + 1 = 65537$

Public key $PU = \{e, n\} \Rightarrow \{7, 33\}$

Private key $PR = \{d, n\} \Rightarrow \{3, 33\}$

$M = 2$

$C = M^e \bmod n = 2^7 \bmod 33 = 29$

$= 128 \bmod 33$

$128 \bmod 33 = 29$

① $128 \div 33 = 3.8787$

② $3.8787 - 3 = 0.8787$

③ $0.8787 \times 33 = 29$

$M = C^d \bmod n = (29)^3 \bmod 33 = 2$

$= 24389 \bmod 33 = 2$

3, 5, 17, 257, 65537

Scanned by CamScanner

example 2

$p = 17 \quad q = 11 \quad e = 7 \quad d = 23$

example 3

$p = 7 \quad q = 11 \quad e = 13 \quad d = 37$

example 4

$p = 257 \quad q = 337 \quad e = 17 \quad d = 65777$

Strength of DH is based on difficulty in computing discrete logarithms for $b < p$ $b = a^i \bmod p$ i = discrete logarithm

a - primitive root
p - prime no.

Diffie - Hellman Key exchange.

The purpose of the algorithm is to enable two users to exchange a secret key securely that can be used for subsequent encryption of messages.

The algorithm is limited to exchange of keys.

Algorithm

Global public elements q prime number

α $\alpha < q$ and α is primitive root of q

User A key Generation

Select private X_A $X_A < q$ — random integers

Calculate public Y_A $Y_A = \alpha^{X_A} \bmod q$

User B key Generation

Select private X_B $X_B < q$

Calculate public Y_B $Y_B = \alpha^{X_B} \bmod q$

Generation of secret key by User A

$K = (Y_B)^{X_A} \bmod q$

Generation of secret key by user B

$K = (Y_A)^{X_B} \bmod q$

Alice wishes to set up a connection with Bob

$\{Y_A, Y_B\}$ Public keys are public — though

$\{X_A \; X_B\}$ Private do not leave localhosts

getting x from Ys is computationally infeasible ∵ log discrete

Scanned by CamScanner

for the algorithm there are 2 publicly known numbers

$q \rightarrow$ prime number

$\alpha \rightarrow \alpha < q$ and it is primitive root of q

→ User A selects a random integer $X_A < q$ and computes

$$Y_A = \alpha^{X_A} \bmod q.$$

→ User B selects a random integer $X_B < q$ and computes

$$Y_B = \alpha^{X_B} \bmod q$$

→ X_A and X_B are private keys and Y_A and Y_B are public keys

→ User A computes the key as $K = (Y_B)^{X_A} \bmod q$ and user B computes the key as $K = (Y_A)^{X_B} \bmod q$

The two calculations produce same result as shown below.

$$K = (Y_B)^{X_A} \bmod q \quad \text{// A can compute}$$

$$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$$

$$= (\alpha^{X_B})^{X_A} \bmod q$$

$$= \alpha^{X_B X_A} \bmod q$$

$$= (\alpha^{X_A})^{X_B} \bmod q$$

$$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$$

$$= (Y_A)^{X_B} \bmod q \quad \text{// B can compute}$$

∴ Two sides have exchanged a secret value. This secret value is used as shared symmetric secret value.

Scanned by CamScanner

→ The primitive root of a prime number p is the one whose power generates all the integers from 1 to p-1 ie α is a primitive root of the prime number p, then the numbers

$\alpha \bmod p,\ \alpha^2 \bmod p,\ \alpha^3 \bmod p \ \ldots\ \alpha^{p-1} \bmod p$

are distinct and consists of the integers from 1 through p-1.

example 1

$q = 353 \qquad \alpha = 3$

$X_A = 97 \qquad X_B = 233$

$Y_A = \alpha^{X_A} \bmod q = 3^{97} \bmod 353 = 40$

$Y_B = \alpha^{X_B} \bmod q = 3^{233} \bmod 353 = 248$

$K_A = (Y_B)^{X_A} \bmod q$

$= (248)^{97} \bmod 353$

$= 160$

$K_B = (Y_A)^{X_B} \bmod q$

$= (40)^{233} \bmod 353$

$= 160$

A & B shares 160 secret key

$= \left[(3^{20} \bmod 353)^8 \times (3^{17} \bmod 353) \times (3^7 \bmod 353)\right] \bmod 353$

$= \left[(73)^8 \times 98 \times 69\right] \bmod 353$

example 2

$q = 23 \qquad \alpha = 5$

$Y_A = 6 \quad X_A = 6 \qquad X_B = 15$

$Y_B = 20 \quad X_B = 9$

$K_A = K_B = 3$

$Y_A = \alpha^{X_A} \bmod q = 5^6 \bmod 23 = 8$

$Y_B = \alpha^{X_B} \bmod q = 5^{15} \bmod 23 = 19$

$K_A = (Y_B)^{X_A} \bmod q$

$= (19)^6 \bmod 23$

$K_B = (Y_A)^{X_B} \bmod q$

$= (8)^{15} \bmod 23$

$= 2$

Example 3

$q = 23 \qquad \alpha = 7$

$x_A = 3 \qquad x_B = 6$

$Y_A = 7^3 \bmod 23$ } Alice

$= 21$

$Y_B = 7^6 \bmod 23$ } Bob

$= 4$

Alice sends 21 to Bob

Bob sends 4 to Alice

Alice calculate

$K_A = 4^{3 \text{ Priv}}_{\text{Pub}} \bmod 23$

$= 18$

$K_B = 21^6 \bmod 23$

$= 18$

$K_A = K_B$

Scanned by CamScanner

*) $n = 33 \quad e = 3 \quad d = 7 \quad \text{Block} = 5$

Problems on RSA

01010 | 01001 | 00101 | 10100

10 ↓ | 9 ↓ | 5 ↓ | 20 ↓

$C = [10^3 \bmod 33]\ [9^3 \bmod 33]\ [5^3 \bmod 33]\ [20^3 \bmod 33]$

$C = 10 \quad 3 \quad 26 \quad 14$

$M = C^7 \bmod n$

$= 10 \quad 9 \quad 5 \quad 20$

*) $P = 3 \quad q = 13 \quad e = 5 \quad d = 5$

$(P-1)(q-1) = 2 \times 12 = 24 \quad \phi(n)$

$P \cdot q = 39 = n$

$e = 5 \quad \because \gcd(24, 5) = 1$

$d\,e \bmod \phi(n) = 1$

$d(5) \bmod 24 = 1$

$d = 5$

$M = 3$

$c = 3^5 \bmod 39$

$= 9^5 \bmod 39$

$= 3$

$C = 9$

$M = 9^5 \bmod 39$

$= 3$

*) $P = 7, \quad q = 11 \quad m = 5$

$e = 13, \quad d = 37$

$c = 26$

Scanned by CamScanner

*) $p = 17 \quad q = 11 \quad e = 7 \quad d = 23$

$p * q = 187 = n$

$\phi(n) = 160 \quad (p-1) \times (q-1) = 160$

$e = 7 \quad$ [chosen]

$d\, e \bmod 160 = 1$

$d * (7) \bmod 160 = 1$

$d = 23$

$C = m^7 \bmod 160$

If $\boxed{M = 88}$

$C = 88^7 \bmod 187$

$C = \left[(88^4 \bmod 187) * (88^2 \bmod 187) \times (88 \bmod 187)\right] \bmod 187$

$= [\, 132 \times 77 \times 88] \bmod 187$

$\boxed{C = 11}$

$M = c^d \bmod n$

$= 11^{23} \bmod 187$

$= \left[(11^7 \bmod 187)^3 \times 11^2 \bmod 187\right] \bmod 187$

$= (88)^3 \times 121$

$\boxed{M = 88}$

Scanned by CamScanner

$a \bmod p, a^2 \bmod p \text{ ---- } a^{p-1} \bmod p$ —①

Primitive root of prime no P is one whose powers generate all the integers from 1 to p-1

ie if 'a' is a primitive root of the prime no. P, then the nos

$a \bmod p, a^2 \bmod p \text{ ---- } a^{p-1} \bmod p$ are distinct and consists of integers from 1 to p-1.

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	5	10	9	7	3	6	1
3	3	9	5	4	1	3	9	5	4	1
4										
5										

→ 1, 2, --- 10

⇒ not distinct

↖ for a mod p where p = 1

p = 11

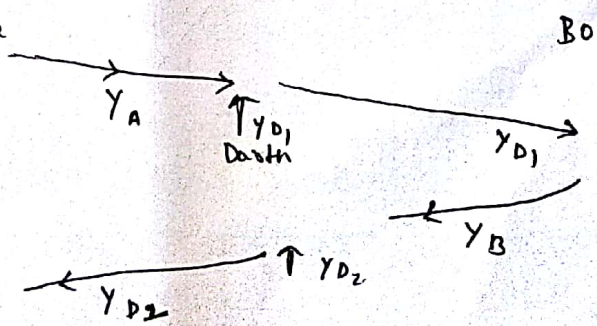
$2^1 \bmod 11, 2^2 \bmod 11, 2^3 \bmod 11, 2^4 \bmod 11$

2

Man in the middle

1) Dasth with an intention of adversary generates Y_{D1} & Y_{D2}

2) Alice Bob



3) Dasth calculates $K_2 = (Y_A)^{X_{D2}} \bmod q$

Bob calculates $K_1 = (Y_{D1})^{X_B} \bmod q$

Scanned by CamScanner

Perspective of an adversary

→ Known information q, α, Y_A, Y_B

→ With the known info adversary is forced to take a discrete logarithm to determine a key

ie to compute X_B, an adversary must compute

$$X_B = d\log_{\alpha,q}(Y_B) \quad -①$$

→ Difficulty in computing discrete log than calculating exponentials modulo.

→ for large prime nos, eq① is infeasible

Key exchange protocols:-

→ central directory (with α, q, Y_A & Y_B)

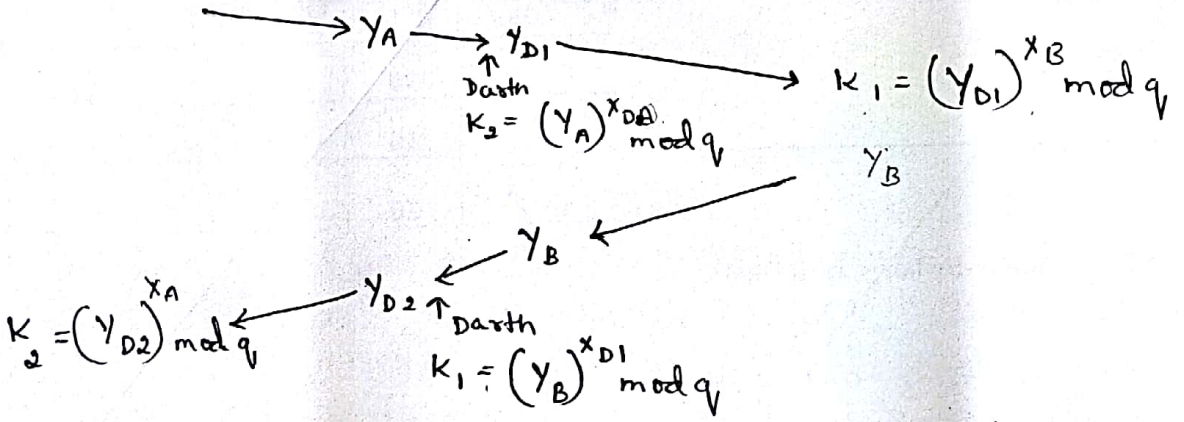
→ Maintenance of confidentiality & authenticity

Man-in-the-middle-Attack

→ Darth X_{D1} & X_{D2} → public keys Y_{D1} & Y_{D2}

X_{DB} X_{DA}

→ Alice Bob



→ Alice sends $E(K_2, M)$ ⇒ Darth intercepts → causes drop

→ Darth sends $E(K_1, M)$ or $E(K_1, M')$ → modification

Scanned by CamScanner

Man in the middle attack :- Diffie-Hellman key exchange protocol is insecure against man-in-the-middle attack. The attack occurs as in the following way.

1. Intruders prepares for the attack by generating two random private keys X_{D1} & X_{D2} & compute corresponding public keys Y_{D_1} & Y_{D_2}.
2. When Alice transmits Y_A to Bob, Intruder (Darth) intercepts Y_A and transmits Y_{D_1} to Bob. Bob calculates key K_2

$$K_2 = (Y_A)^{X_{D2}} \bmod q.$$

3. Bob receives Y_{D_1} & calculates $K1 = (Y_{D1})^{X_B} \bmod q$
4. Bob transmits Y_B to Alice
5. Darth intercepts Y_B and transmits Y_{D_2} to Alice.
   Also Darth calculates $K_1 = (Y_B)^{X_{D1}} \bmod q$.
6. Alice receives Y_{D2} & calculats $K_2 = (Y_{D2})^{X_A} \bmod q$.

Here Bob & Alice think that they share a secret key. Instead Bob & Darth share secret key K_1, & & Alic & darth share the secret key K_2.

All fatd future comm^n gets Intercepted as follows. M from Alice is decrypted by Darth (as he has the key) then Darth sends Bob as $E(M', K_1)$ where M' can be any msg.

Note :- Key exchange protocol is vulnerable to such an attack because it does not authenticate participants. which is overcome by digital signatures & public-key certificates.

Scanned by CamScanner

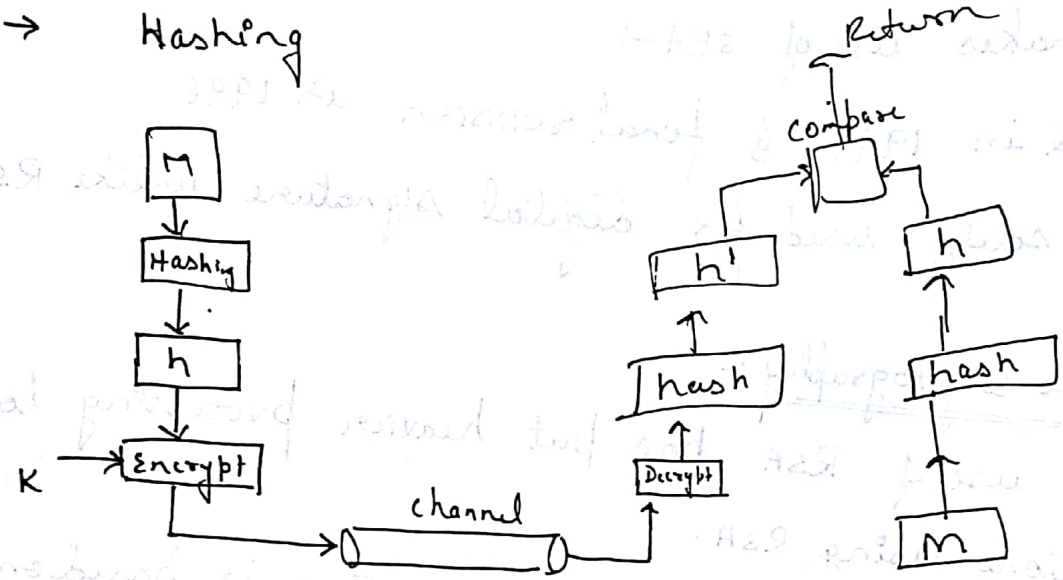
Digital signature

→ Using private key

→ Storage concern

→ Function of the content ⇒ authenticator

→ Hashing



Scanned by CamScanner

Other public-key cryptography algorithms.

Dss - Digital signature standard (DSS)

Elliptic curve cryptography

DSS:-

* NIST (National Institute of standards & Technology) has FIPS PUB 186 → knowns as DSS.
* Dss makes use of SHA-1
* Proposed in 1991 & final revision in 1996.
* It is solely used for digital signature unlike RSA.

Elliptic curve cryptography:-

* Extensive use of RSA has put heavier processing load on applications using RSA.
* A competing system is the ECC. ECC is based on mathematical constructs known as elliptic curve.
* ECC appears to offer equal security for a smaller bit size
* Cryptanalytic interest in probing the weakness of ECC is still in progress.

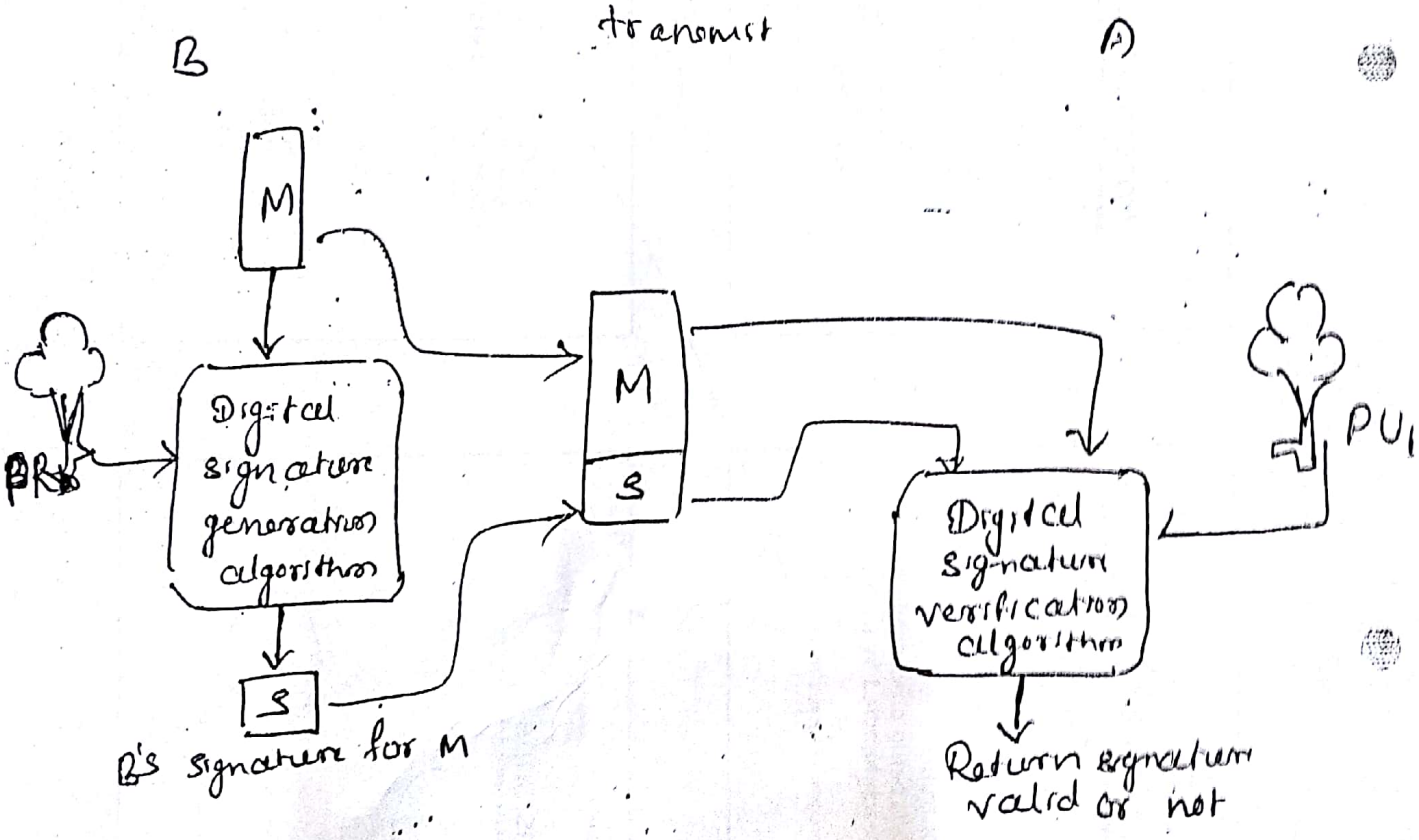
Scanned by CamScanner

Digital Signatures

6th edition

The important development from the work on public-key cryptography is the digital signature. The digital signature provides a set of security capabilities that would be difficult to implement.

Generic model of digital signature process

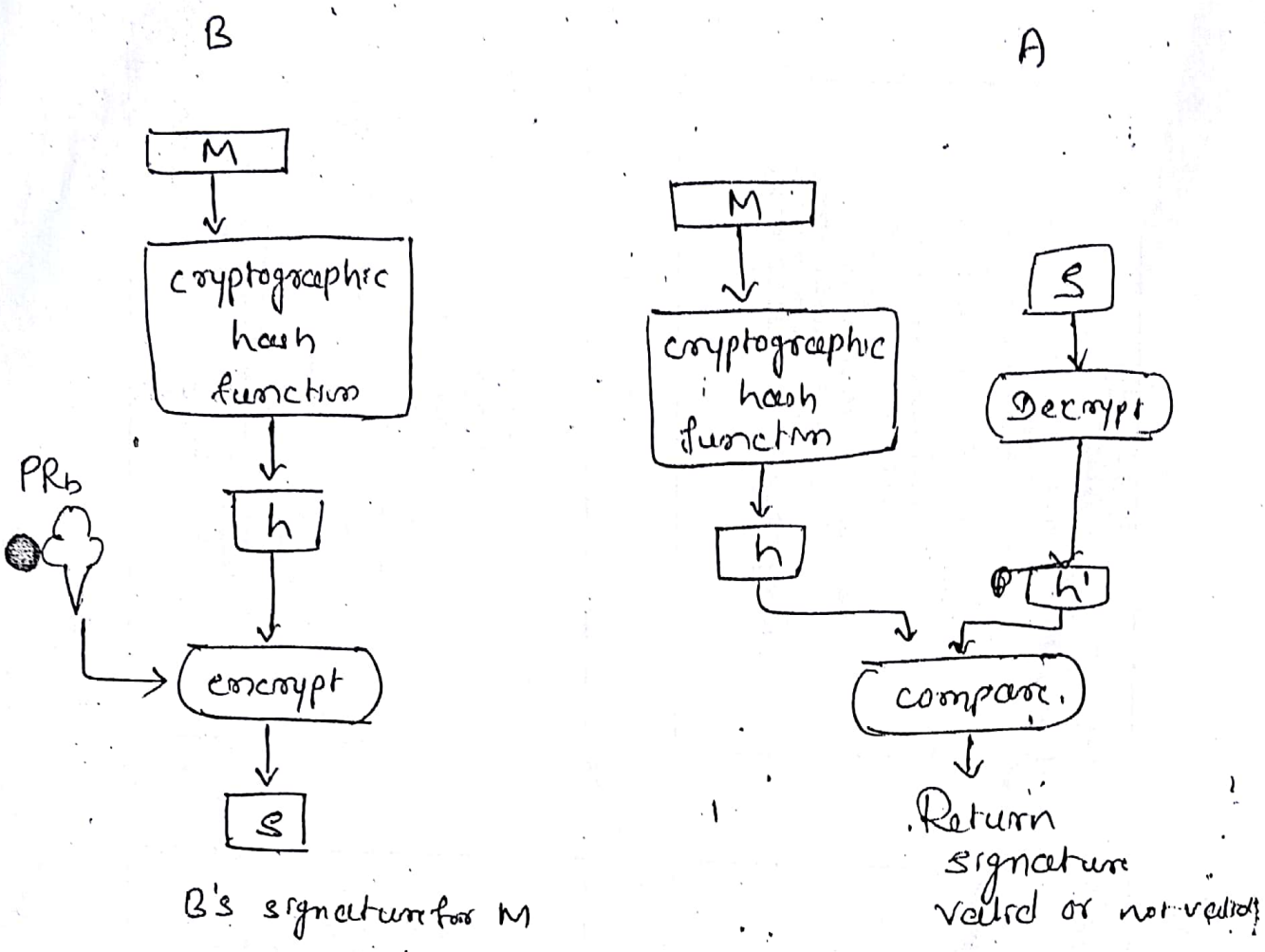


→ B can sign a message using a digital signature generation algorithm. The inputs to the algorithm are the B's private key and the message.

→ Any other user, A can verify the signature using a verification algorithm, whose inputs are the message, signature & public key of the sender B.

Scanned by CamScanner

Simplified Depiction of essential elements of Digital signature process.



Message authentication protects 2 parties who exchange message from any 3rd party. Several forms of argument or disagreement may happen between the two.

Eg Suppose A sends an authenticated message to B. Consider the following argument that could arise

1) B may forge a different message and claim that it came from A. B would simply have to create a message & append an authentication code using the key A & B share.

Scanned by CamScanner

2) A can deny sending the message. Because it is possible for B to forge a message., there is no way to prove that A did in fact send the message.

for the first scenario

An electronic funds transfer takes place, and the receiver increases the amounts of funds transfered and claims that larger amount had arrived from the sende

for second scenario

An electronic mail message contains instructions to a stockbroker for a transaction that turns out badly. The sender pretends that the message was never sent.

In situation where there is not complete trust between sender & receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature must have the following properties

* It must verify the author and the date and time of the signature
* It must authenticate the contents at the time of the signature
* It must be verifiable by third parties to resolve disputes

Hence digital signature function includes the authentication function

*) P = 7 q = 11 e = 13

n = P * q = 77

$\phi(n) = (p-1)(q-1) = 60$

e = 13

$de \bmod \phi(n) = 1$

$d * 13 \bmod 77 = 1$

d = 6, choosing m = 10

$C = 10^{13} \bmod 77$

$= [10^6 \bmod 77 \times 10^6 \bmod 77 \times 10^1 \bmod 77] \bmod 77$

$= [100 \times 100 \times 10] \bmod 77$

C = 54

$M = C^d \bmod 77$

$= 54^6 \bmod 77$

$= [54^3 \bmod 77 \times 54^3 \bmod 77] \bmod 77$

$= [76 \times 76] \bmod 77$

=

verification 60) 65 (1
60
5

$m^{ed} \bmod n = m$

$10^{13 \times 6} \bmod 77$

d = 5

*) p = 3 q = 13

e = 5

d = 5

p =

x mod y

① $x \div y = A.B$

② $A.B - A = .B$

③ $.B \times y =$

65 mod 60

65 / 60 = 1.0833

1.0833 − 1 = 0.833

0.833 × 60 = 5

Scanned by CamScanner