Public Key Cryptography — chapter 9 in Stallings book (7th Ed)

Principles of Public-Key Cryptosystems

↓

evolved from an attempt to solve two problems associated with symmetric-key cryptography.

① Key Distribution ② Digital Signature.

① Key Distribution

↳ How to have secure communications in general without having to trust KDC with your key.

KDC — Key Distribution Center.

② Digital Signature

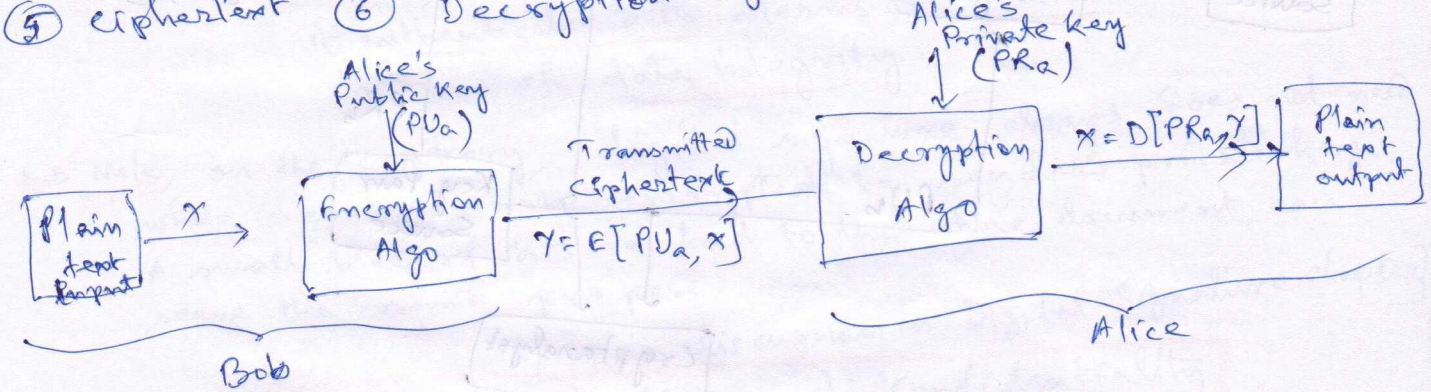
↳ How to verify that a message comes intact from the claimed sender.

↳ this requirement is the superset to user authentication.

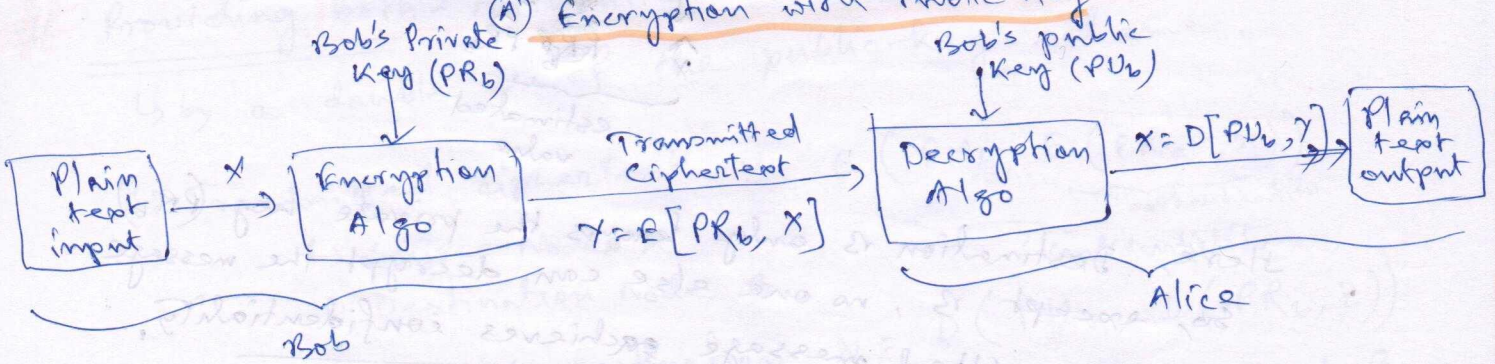
↳ Public-key cryptosystem has six ingredients

① Plaintext ② Encryption Algorithm ③④ Public and Private keys

⑤ ciphertext ⑥ Decryption Algorithm



(A) Encryption with Public Key



(B) Encryption with Private key

Application: (A) Many-to-One scenario e.g. customer → Bank

(B) One-to-Many scenario e.g. Bank → customer.

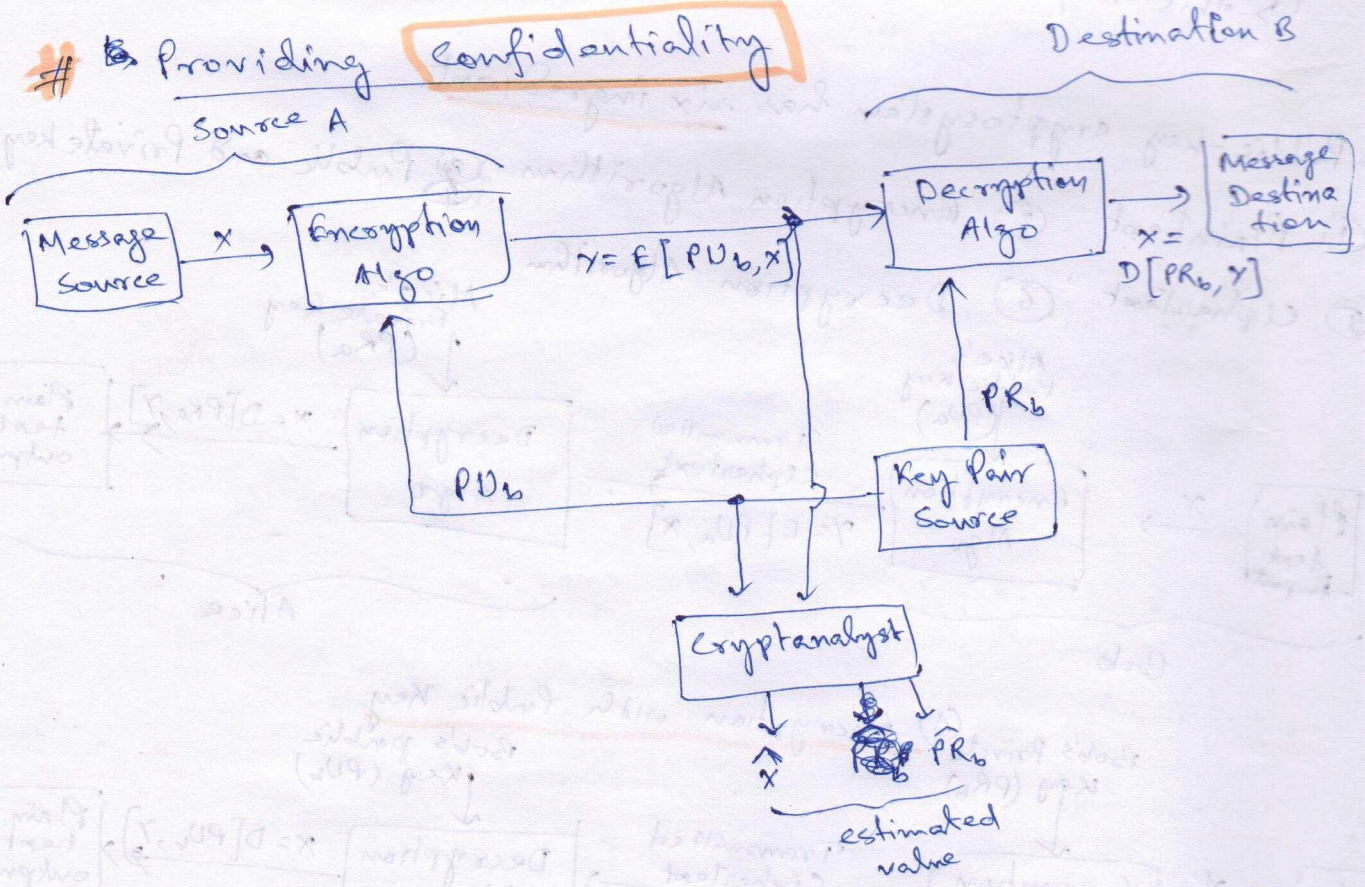
↳ Important Characteristics of

↳ It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algo and the encryption key.

↳ In many algo, either of the two related keys can be used for encryption, with the other used for decryption.
e.g. RSA.

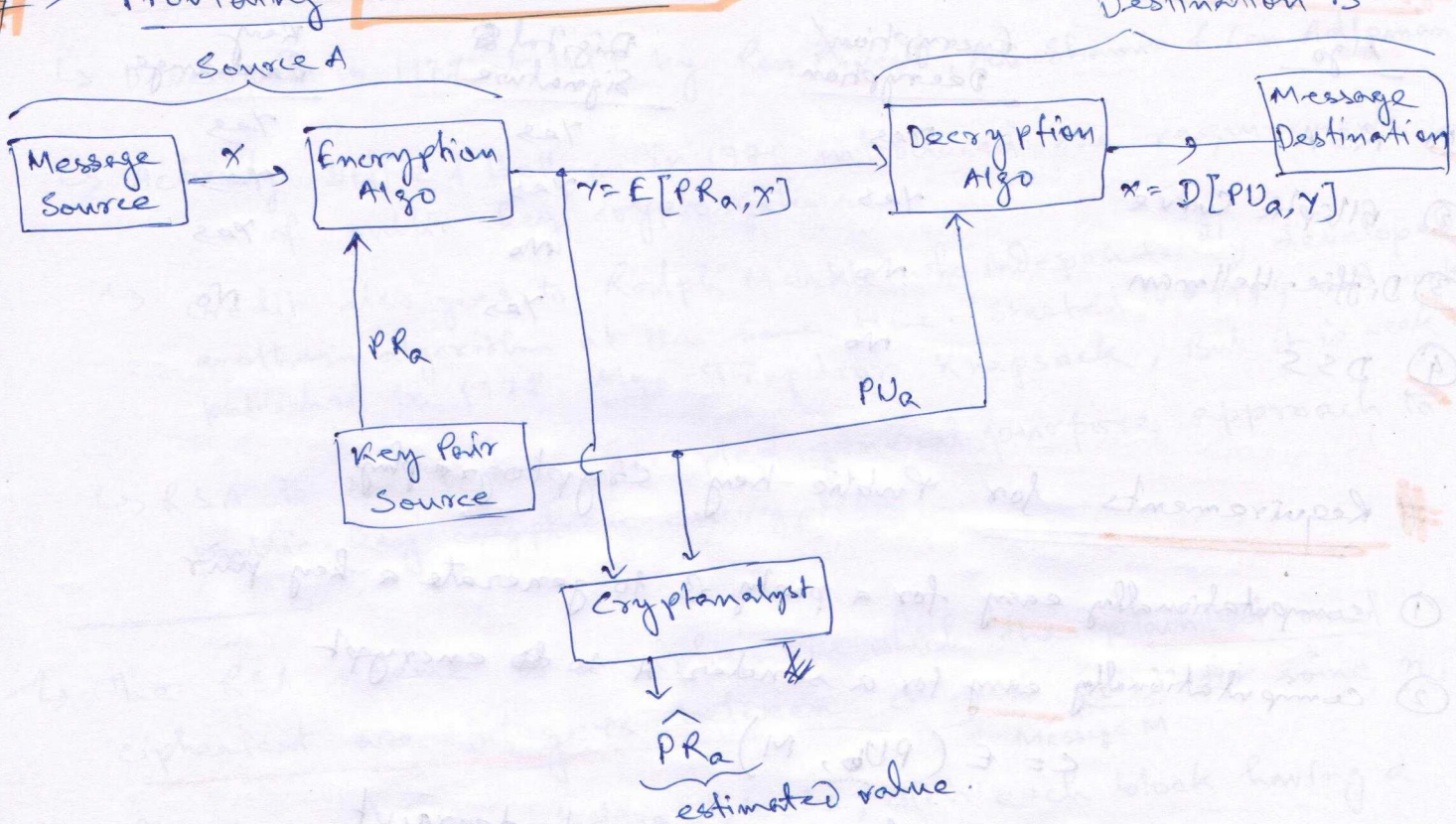
↳ Key convention
- ↳ In symmetric-key cryptography
  - ↳ secret key
- ↳ In Public-key cryptography
  - ↳ Public key & Private key

Providing Confidentiality



Here, Destination B only knows the private key (PR_b)
So, except B, no one else can decrypt the message.
So, transmitted message achieves confidentiality.

Providing Authentication



↳ In this case, any node who intercept Y can also decrypt it as PU_a is available openly.

↳ Here, destination B can ensure who has sent the message,

↳ Here, the entire encrypted message serves as a digital signature.
↓
it authenticates both interms of source and interms of data integrity.

↳ Note, on the contrary, digital signature always does not need whole message to be encrypted by sender's private key. A small block of bits related to the whole document can serve the same purpose.
[More discussion in Digital Signature chapter]

Providing both - Authentication and confidentiality

↳ by a double use of the public-key scheme.

↳ Transmitted ciphertext $Z = E(PU_b, E(PR_a, X))$
authentication
confidentiality

↳ Source Node A

↳ Destination Node B.

↳ Decryption at destination $X = D(PU_a, D(PR_b, Z))$

↳ Disadvantage: Public-key algorithm must be exercised four times per communication.

Applications of Public-Key Cryptosystems

Algo	Encryption/ Decryption	Digital Signature	Key Exchange
(1) RSA	Yes	Yes	Yes
(2) Elliptic Curve	Yes	Yes	Yes
(3) Diffie-Hellman	No	No	Yes
(4) DSS	No	Yes	No

Requirements for Public-Key Cryptography

(1) computationally easy for a party B to generate a key pair

(2) computationally easy for a sender A to encrypt

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

(3) Computationally easy for the receiver to decrypt

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

(4) Computationally infeasible for an adversary, knowing the public key PU_b, to determine the private key PR_b

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

(6) [optional] the two keys can be applied in either order

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

Public-Key Cryptanalysis

(1) Brute-force attack — use all combination to decrypt

(2) Key Prediction attack — find some way to compute private key given the public key

(3) Probable-message attack — eg: if a message is sent to convey 56-bit DES key. Now, an adversary could encrypt all possible 56-bit DES keys using the public key and then could discover the encrypted DES key by matching the transmitted ciphertext.

The RSA Algorithm

↳ Developed in 1977 at MIT by Ron Rivest, Adi Shamir & Len Adleman

↳ Actually Diffie & Hellman in 1976 introduced the requirements of public-key cryptosystems.

↳ Credit also goes to Ralph Merkle who independently developed another algorithm at the same time. Started in 1974 and work published in 1978. Algo Trapdoor Knapsack., But it is weak.

↳ RSA is the widely accepted general purpose approach to public-key ~~cryptosy~~ encryption.

↳ The RSA scheme is a cipher in which the plaintext and ciphertext are integers between 0 and $(n-1)$ for some n.
Message M

↳ Plaintext is encrypted in blocks, with each block having a binary value less than some number n.

i.e. block size ~~must~~ $\leq \log_2(n) + 1$

if, block size is i bits, then $2^i < n$ ~~2^{i+1}~~

Encryption & Decryption in RSA

↳ 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 value of n.

↳ Sender knows the value of e,

↳ Only the receiver knows the value of d,

↳ Thus, Public key $PU = \{e, n\}$
Private key $PR = \{d, n\}$

↳ For this algorithm to be satisfactory for public-key encryption it is needed to find out the values of e, d and n such that $M^{ed} \bmod n = M$ for $\forall M < n$.

↳ we need to find a relationship of the form

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

↳ this relationship holds if e and d are multiplicative inverse modulo $\phi(n)$, where $\phi(n)$ is the Euler totient function.

↳ Euler's totient function $\phi(n)$

↳ It is defined as the number of positive integers less than n and relatively prime to n.
↳ only common positive integer factor is 1 i.e. $\gcd(\phi(n), n) = 1$
↑ elements of $\phi(n)$

(a) ↳ For any prime number p

$$\phi(p) = p - 1$$

e.g. $\phi(37) = 36$ as all the positive integers from 1 through 36 are relatively prime to 37 (as 37 itself is prime).

(b) ↳ Exception: $\phi(1) = 1$

(c) ↳ Now suppose we have two prime numbers p and q with $p \neq q$. Then, for $n = pq$,

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

e.g. $\phi(21) = \phi(3) \times \phi(7) = (3-1) \times (7-1) = 2 \times 6 = 12$
(3 and 7 are prime)

where 12 integers are $\{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$

See chapter 2 (sec 2.5) in Stallings book (7th Ed).

~~↳ Multipli~~

↳ e & d are multiplicative inverse modulo $\phi(n)$

↳ means $ed \bmod \phi(n) = 1$

i.e. $ed \equiv 1 \bmod \phi(n)$

$\Rightarrow d \equiv e^{-1} \bmod \phi(n)$

↑ congruent modulo

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

or $ed \bmod \phi(n) = 1 \bmod \phi(n)$

↳ Note that, according to modular arithmetic, this is true only if d (and therefore e) is relatively prime to $\phi(n)$.

i.e. $\gcd(\phi(n), d) = 1$

The RSA Algorithm

Let Bob wants to send to Alice.

(A) Key Generation by Alice

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 — $\begin{cases} d \equiv e^{-1} \bmod \phi(n) \Rightarrow d \bmod \phi(n) = e^{-1} \bmod \phi(n) \\ 1 < d < \phi(n) \end{cases}$

Public key — $PU = \{e, n\}$

Private key — $PR = \{d, n\}$

(B) Encryption by Bob with Alice's Public Key

Plaintext — $M < n$

Ciphertext — $C = M^e \bmod n$

(C) Decryption by Alice with Alice's Private Key

Ciphertext — C (intact $C < n$)

Plaintext — $M = C^d \bmod n$

An example

Let $p = 17$, $q = 11$

so, $n = pq = 187$

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

Select e such that $\gcd(160, e) = 1$ and $e < 160$.

Let $e = 7$

So, get d

$de \equiv 1 \bmod 160$ and $d < 160$

$\Rightarrow d = 23$.

from this, d is computed using Extended Euclidean Algorithm.

See chapter 2 (Sec. 2.3) stallings book (7th Ed).

Now the resulting keys are

$PU = \{7, 187\}$, $PR\{23, 187\}$.

Let plaintext input of $M = 88$.

So, $C = 88^7 \bmod 187$

$\Rightarrow 88^7 \bmod 187$

$= \left[(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)\right] \bmod 187$

$\Leftarrow$ using modular arithmetic properties.

$88^1 \bmod 187 = 88$

$88^2 \bmod 127 = 77$

$88^4 \bmod 127 = \left[(88^2 \bmod 187) \times (88^2 \bmod 187)\right] \bmod 187$

$= [77 \times 77] \bmod 127$

$= 132$

$\Rightarrow 88^7 \bmod 187 = (132 \times 77 \times 88) \bmod 187$

$= 11$

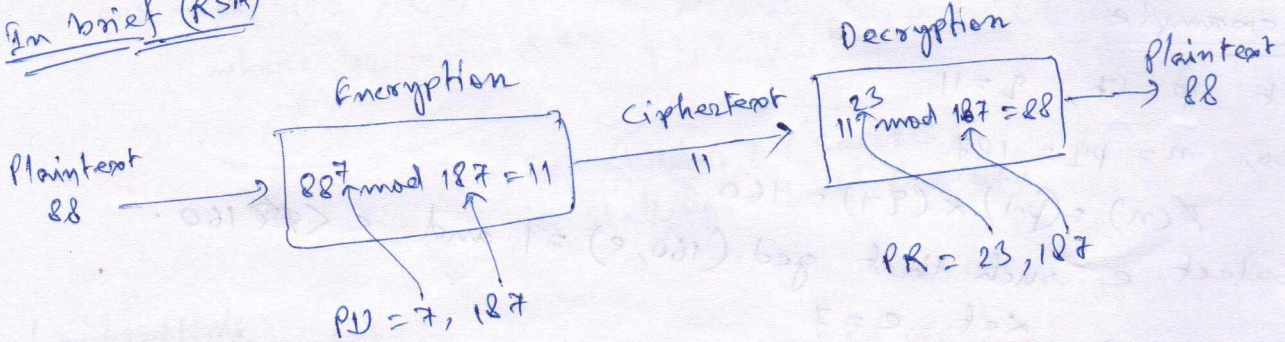
for decryption.

$11^{23} \bmod 187 = \left[(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187)\right] \bmod 127$

$= [11 \times 121 \times 55 \times 33 \times 33] \bmod 187$

$= 88$

In brief (RSA)



The Security of RSA

Five possible approaches to attacking the RSA

(1) Brute-force — trying all possible private keys

(2) Mathematical attack — factoring the product of two primes

(3) Timing attack — depends on the running time of the decryption algorithm

(4) Hardware fault-based attack — involves inducing hardware faults in the processor that is generating digital signature.

(5) Chosen ciphertext attack — exploits the properties of RSA

Few common misconceptions related Public-key Cryptography:

(1) Public-key encryption is more secure than is symmetric encryption.
— In fact, there is no proof for this claim.

(2) Public-key encryption is ~~more~~ a general purpose technique that has made symmetric encryption obsolete.
— It is also wrong.

(3) There is a feeling that key distribution is trivial when using public-key encryption.
— In fact the procedure involved are not simpler nor any more efficient.