UNIT-3

Syllabus:-

Arithmetic for Cryptography: Pseudo random Number Generation, Prime Numbers, Euler's Theorem & CRT, Stream Ciphers: RC4 Public Key Kryptography: Principles of Public Key Cryptosystem, RSA Algorithm Security of RSA, Diffchellman key Exchange, Elleptical curve cryptography.

—x—

Pseudo Random Number Generation:-

↳ means half → Looks like random numbers but not realy random.

The Use of Random Numbers

1. key distribution and reciprocal authentication schemes

messages to distribute keys & / authenticate each other.

· The use of random no's for the nonces frustrates an opponent's efforts to determine / guess the nonce.

2. Session key generation.

· Secret key for symmetric Encryption is generated for use for a short Period of time.

3. Generation of Keys for RSA Public key Encryption Algorithm.

4. Generation of bit stream for symmetric stream Encryption

2 distinct requirements for a sequence of random no's.

1. Randomness :-

2 Criteria are used to validate that a sequence of no's is random.

→ Uniform Distribution :-

Distribution of bits in the sequence should be uniform.

i.e., Frequency of occurence of 1's & 0's = equal.

→ Independence :-

No one subsequences in the sequence can be inferred from the others.

2. Unpredictability

With "time" random sequences, each no is statistically independent of other no's in the sequence &

∴ unpredictable → opponent notable to predict future elements of sequence.

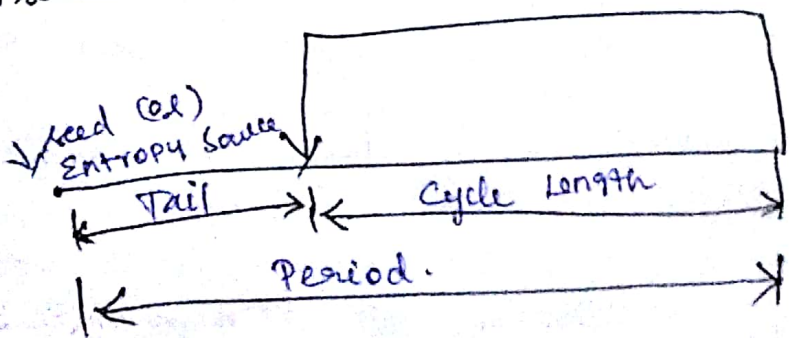
TRNGs, PRNGs, PRFs.

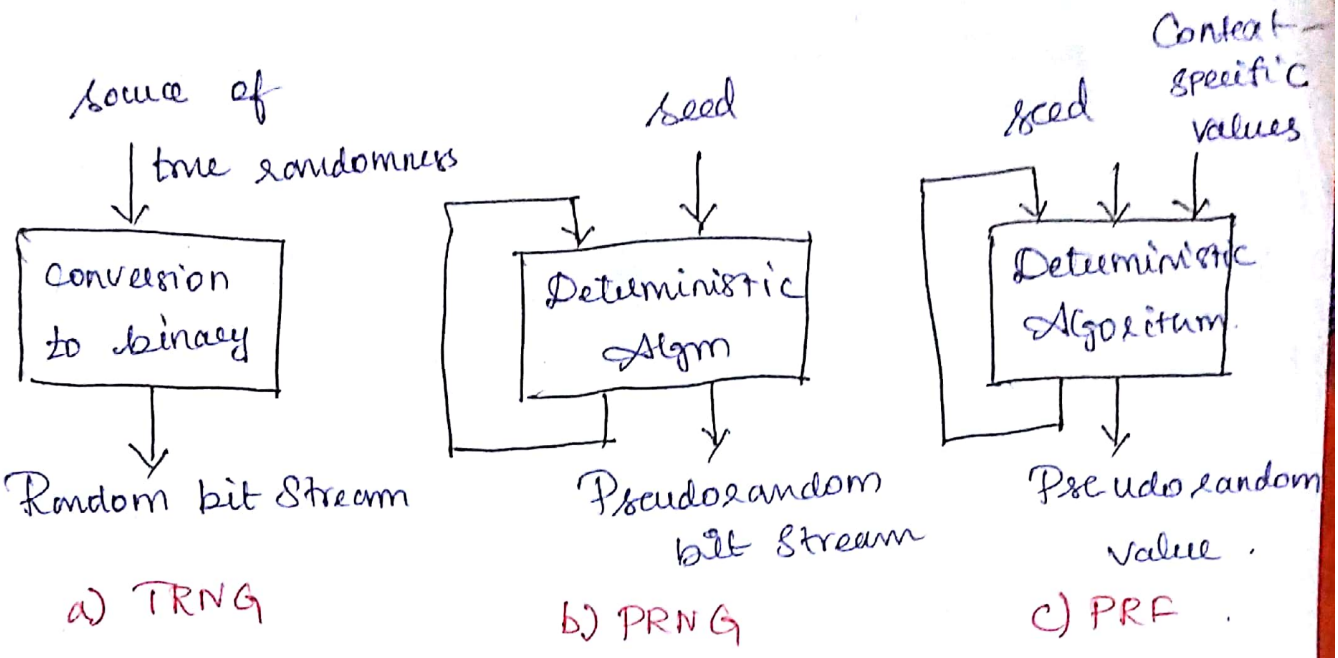
TRNGs → True Random Number Generators

PRNGs → Pseudo Random Number Generators

PRFs → Pseudo Random Functions.

TRNG Terminology





→ TRNGs takes I/P as a fixed value → called Seed.

→ Seed = x_0

→ Pseudo - Random : Deterministic yet would pass randomness tests.

→ Fully - Random : Non Repeatable

→ Cycle length, Tail, Period.

PSRNs → sequence → produce → good looking sequence will Pass many reasonable tests of randomness such numbers are referred to PSRNs.

TRNG → with → 2 forms of PRNGs.

→ A TRNG takes I/P as source ie. effectively random.

→ The some often referred to as an entropy source.

→ Entropy Source is drawn from the physical environment of the computer & includes things such as

1. Keystroke timing patterns,
2. Disk electrical activity,
3. Mouse movements &
4. Instantaneous values of the system clock.

Source / combination of source (I/P)

↓

Random binary O/P.

1) TRNG —Convert→ Analog source to Binary O/P.

→ In contrast PRNG takes I/P fixed value called seed —Produces→ sequence of O/P bits using a deterministic Algm.

→ PRNGs has 2 diff forms based on appln.

2) PRNGs → Algm i.e., used to produce an open ended sequence of bits is referred to as a PRNG.

→ Common Appln —for→ open ended sequence of bits is I/P to symmetric stream cipher. ~~Sequence~~

3) PRF → Pseudo Random Function

↳ produced Pseudo Random string of bits of some fixed Length.

↳ Examples → symmetric encryption keys & nonces.

→ PRF takes as i/p seed + some context specific values, such as a user ID / an application ID.

→ Other than the no. of bits produced, there is no difference b/w PRNG & PRF. The same Algms can be used in both Appns.

→ Both needs seed &
Both must exhibit randomness & unpredictability.

PRNG Requirements

→ In general requirement of secrecy of the o/p of PRNG (or) PRF leads to specific Req in the areas of randomness, Unpredictability & the characteristics of Seed.

(1.) Randomness :-

↳ It specifies that the tests should seek to establish the following 3 Characteristics.

1. Uniformity
2. Scalability
3. Consistency.

→ to give flavor for the test, we list 3 of the test.

1. Frequency Test.

Purpose → To determine wether the no. of ones & 0's in a sequence is same for a truely random sequence

2. Runs Test :-

Purpose → To determine wether the no. of runs of 1's & 0's of various lengths is as expected for a random sequence.

3. Maurer's Universal Statistical test :-

Purpose :-

→ To detect wether or not the sequence can be significantly compressed without loss of information.

(2.) P Unpredictability

→ exhibit 2 forms.

1. Forward Unpredictability

→ If the seed unknown, next op sequence is unpredictable by previous bits in the sequence

2. Backward Unpredictability

→ Infeasible to determine seed.

→ No correlation b/w seed & value generated from seed should be evident.

(5) Seed Requirements.

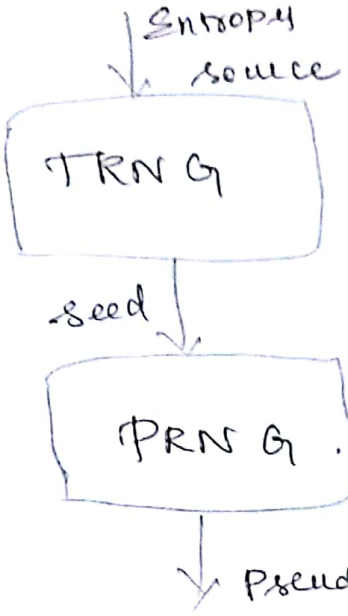


Fig: Generation of seed I/p to PRNG

Algorithm Design:-

→ Cryptographic PRNGs Algms falls roughly into 2 categories.

1. Gen purpose - built Algorithms ex → stream ciphes RC4
2. Algorithms based on Existing Cryptographic Algms.
   ↳ 3 categories of cryptographic Algms are commonly used to create PRNGs
   i) Symmetric block Ciphers
   ii) Asymmetric Ciphers
   iii) Hash functions & Message Authentication Codes

For Ex:

$$x_n = 5x_{n-1} + 1 \mod 16$$

seed $x_0 = 5$

$n = 1$

$$x_1 = 5x_{1-1} + 1 \mod 16$$

$$= 5x_0 + 1 \mod 16$$

$$= 5 \times 5 + 1 \mod 16$$

$$= 26 \mod 16.$$

$$\boxed{x_1 = 10}$$

⇒ The first 32 numbers obtained by the above Procedures.

{0, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4
5, 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5}

⇒ To avoid the predection of values in the sequence. will divide x's by 16.

So that the above sequence becomes as follows.

{0.6250, 0.1875, 0.0000, 0.0625, 0.3750, 0.9375,
0.7500, 0.8125, 0.1250, 0.6875, 0.5000, 0.5625, 0.8750,
0.4375, 0.2500, 0.3125, 0.6250, 0.1875, 0.0000,
0.0625, 0.3750, 0.9375, 0.7500, 0.8125, 0.1250,
0.6875, 0.5000, 0.5625, 0.8750, 0.4375, 0.2500,
0.3125}.

Pseudo Random Number Generators :— 2 types of Algms

1. Linear Congruential Generators (LCM)
2. Blum Blum Shub Generators (BBS)

1. LCM :—

→ Discovered by D.H Lehmer.

→ The residues of successive Powers of a number have good randomness properties.

→ A widely used technique for PRNG.

→ The Algm is Parametrized with 4 no's.

m — the modulus $m > 0$

a — the multiplier $0 < a < m$

c — the increment $0 \leq c < m$

X_0 — the starting value/seed $0 \leq x_0 < m$.

→ The sequence of Random no's (X_n) is obtained using this equation.

$$X_{n+1} = (a X_n + C) \bmod m .$$

→ If m, a, c & X_0 are integers, then it Produces sequence of integers with the range $0 \leq X_n < m$.

For ex:-

Consider $a = 7$, $c = 0$, $m = 32$ & $X_0 = 1$.

$n=0$ $\boxed{X_{n+1} = (aX_n + c) \bmod m}$

$X_{0+1} = (7 \times X_0 + 0) \bmod 32$

$X_1 = 7 \times 1 \bmod 32$

$\boxed{X_1 = 7}$

$n=1$

$X_{1+1} = (7 \times X_1 + 0) \bmod 32$

$X_2 = 7 \times 7 \bmod 32$

$= 49 \bmod 32$

$\boxed{X_2 = 17}$

$n=2$

$X_{2+1} = (7 \times X_2 + 0) \bmod 32$

$X_3 = 7 \times 17 \bmod 32$

$= 119 \bmod 32$

$\boxed{X_3 = 23}$

∴ This generates the sequence.

{ 7, 17, 23, ... etc }.

→ which is also clearly unsatisfactory.

Of the 32 possible values, only 4 are used.

→ Thus the sequence is said to have a period of 4.

→ If instead, we change the value of $a \to 5$. then the sequence is. $\{8, 25, 29, 17, 21, 9, \ldots \text{etc}\}$.

$n=0$

$$X_{0+1} = 5 \times X_0 + 0 \mod 32$$

$$= 5 \times 1 \mod 32$$

$$\boxed{X_1 = 5}$$

$n=1$

$$X_{1+1} = 5 \times X_1 + 0 \mod 32$$

$$= 5 \times 5 \mod 32$$

$$\boxed{X_2 = 25}$$

→ which increases the period of 8. & we would like m to be very Large, so that there is the potential for producing a Long series of distinct random no's.

→ IL proposes 3 tests to be used in evaluating a random no generator.

T_1: The function should be a full period generating function. ie., the function generate all the num b/w 0 & m. before repeating.

T_2: Generate sequence should appear random.

T_3: Function should implement efficiently with 32-bit arithmetic.

For 32 bit arithmetic a convenient Prime value of m is $2^{31}-1$. Thus the generating function becomes.

$$X_{n+1} = (aX_n) \bmod (2^{31}-1)$$

If an opponent knows that the linear Congruential Algm is being used & if parameters are known (eg. $a=7^5$, $c=0$, $m=2^{31}-1$). then once a single no is discovered, all subsequent no's are known.

Suppose that opponent is able to determine values for X_0, X_1, X_2 & X_3 then.

$$X_1 = (aX_0 + c) \bmod m$$

$$X_2 = (aX_1 + c) \bmod m$$

$$X_3 = (aX_2 + c) \bmod m$$, These equations are solved for a, c and m.

Thus, although it is nice to be able to use a good PRNG, it is desirable to make the actual sequenced used nonreproducible, so that knowledge of part of the sequence on the part of an opponent is insufficient to determine future elements of the sequence.

2. BBS

→ Popular Approach to generating secure PRN is known as BBS Generator.

→ It has perhaps the strongest public proof of its Cryptographic strength of any Purpose-built Algorithm.

→ Procedure is follows.

1. Choose 2 Large Prime no's. P & Q, both have a remainder of 3 when divided by 4.

i.e., $P \equiv Q \equiv 3 \pmod 4$.

$P \bmod 4 = Q \bmod 4 = 3$.

For ex:

Prime no's 7 & 11.

$7 \equiv 11 \equiv 3 \pmod 4$, (or)

$7 \bmod 4 = 11 \bmod 4 = 3$

2. Choose random no's S, such that S is relatively Prime to n.

neither P nor Q is factor of S.

Then BBS generates sequence of bits B_i according to the following Algm.

$X_0 = S^2 \bmod n$

for $i = 1$ to ∞

$X_i = (X_{i-1})^2 \bmod n$

$B_i = X_i \bmod 2$

11) Euler's Theorem :-

Theorem :-

For every a & n values

$$a^{\phi(n)} \equiv 1 \pmod{n} .$$

Proof :-

$\phi(n)$ is Euler's Totient function.

For Prime Number n.

$$\boxed{\phi(n) = n-1}$$

eg

$$\phi(37) = 36 .$$

$$\phi(21) = \phi(3) \times \phi(7) = 2 \times 6 = 12 .$$

$\phi(n)$ is +ve integers $< n$, n is reltively Prime to n.

Consider set of integers.

$$R = \{x_1, x_2 \ldots x_{\phi(n)}\} .$$

Each element x_i of R is a unique +ve integer $< n$.

Now mutiply each element by a, mod n.

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n), \ldots$$

$$\ldots (ax_{\phi(n)} \bmod n)\}$$

The set S is a Permutation of R.

1. Because a is relatively prime to n & x_i is relatively prime to n.

ax_i must be relatively prime to n.

Thus all the members of S are integers that are $< n$ & that are relatively prime to n.

2. There are no duplicates in S.

$a x_i \bmod n = a x_j \bmod n$, then $x_i = x_j$.

$\therefore \quad S = R$

$$\prod_{i=1}^{\phi(n)} (a x_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$a^{\phi(n)} \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \left[\prod_{i=1}^{\phi(n)} x_i\right] \bmod n$$

$$\boxed{a^{\phi(n)} \equiv 1 \bmod n}$$

eg Problem-1 :-

Given :-

$a = 3, \quad n = 10$

Soln :-

$\phi(n) = ?$

$\phi(10) = \phi(5) \times \phi(2)$

$= (5-1)(2-1) = 4 \times 1 = 4$

$\boxed{\phi(10) = 4}$

$3^4 \equiv 1 \mod 10$

$81 \equiv 1 \mod 10$.

$81 \bmod 10 = 1$

$1 = 1$

eg Problem - 2 :-

Given :-

$a = 2, \quad n = 11$

Soln :-

$\phi(n) = n - 1$

$\phi(11) = 11 - 1 = 10$

$2^{10} \bmod 11 = 1$

$1024 \bmod 11 = 1$

$1 = 1$

III) CRT → Chinese Remainder Theorem.

→ It is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.

$$M = \prod_{i=1}^{k} m_i .$$

where m_i → Pairwise relatively prime,

i.e., $\gcd(m_i, m_j) = 1$, for $1 \le i, j \le k$ & $i \ne j$

eg Problem:-

Given:- A bag has certain no. of pens,

If you take out 3 pens at a time, 2 pens are left, If you takeout 5 pens at a time, 3 pen is left & If you takeout 7 pens at a time 2 pens are left in the bag.

How many pens in the bag?

Soln:

$x \equiv 2 \mod 3 \rightarrow ①$ $\quad x \bmod 3 = 2$

$x \equiv 3 \mod 5 \rightarrow ②$ $\quad x \bmod 5 = 3$

$x \equiv 2 \mod 7 \rightarrow ③$ $\quad x \bmod 7 = 2$.

Equation ① & ②.

$3 \times 5 \bmod 7 = 2$

$15 \bmod 7 = 2$.

$1 \times 15 \mod 7 = 1 \times$

$2 \times 15 \mod 7 = 2$ ✓

$\downarrow 30$

equation ② & ③ .

$5 \times 7 \mod 3 = 2$.

$35 \mod 3 = 2$.

$1 \times 35 \mod 3 = 2$ ✓

$\downarrow 35$

equation ③ & ①

$3 * 7 \mod 5 = 3$.

$21 \mod 5 = 3$.

$1 \times 21 \mod 5 = 1 \times$

$2 \times 21 \mod 5 = 2$ ✓

42

$3 \times 21 \mod 5 = 63 \mod 5 = 3$ ✓

$X = 30 + 35 + 63 = 125$

$\text{mod} = 3 \times 5 \times 7 = 105$ (-)

23 ✓

$X \equiv 2 \pmod 3 \equiv 3 \pmod 5 \equiv 2 \pmod 7$

Answer:

$X = 23 + 105\ k$

IV) Public Key Encryption Scheme → It has six ingredients.

1. Plain Text.
2. Encryption Algm
3. Public key
4. Private key.
5. Ciphen Text
6. Decryption Algm.

Essential steps:-

1) Each user generates a pair of keys to be used for the Encryption & Decryption of messages.

2) Each User places one of the 2 keys on the Public reg → This is Public key → companion Key kept Secret (private).

Each user maintains a collection of public keys obtained from others

3) If bob wishes to send a confidential message to Alice, Bob Encrypts the mssg using Alice Public key.

4) When Alice receives the msg, She decrypts it using her private key. No other receipient can decrypt the msg. becz only Alice knows its Pri. key.

Principles of Public Key Cryptography

→ Public key Cryptography follows a Asymmetric key Encryption Algorithm.

→ 2 problems in Symmetric key Encryption Algorithm.

1. key distribution.
2. digital Signature → Authentication.

→ Cryptography → wide spread Usage.

1. Military
2. Commercial &
3. Private.

→ Diffie-Hellman solved this problems

Public key Cryptosystems:-

→ Asymmetric Encryption Algm.

→ It has following Char.

1. Computationally infeasible to determine the decryption, key given by Knowledge of Cryptographic Algm & Encryption key.

In addition some Algms such as RSA following Char.

1. Either of the 2 related keys can be used in E with other used in D.

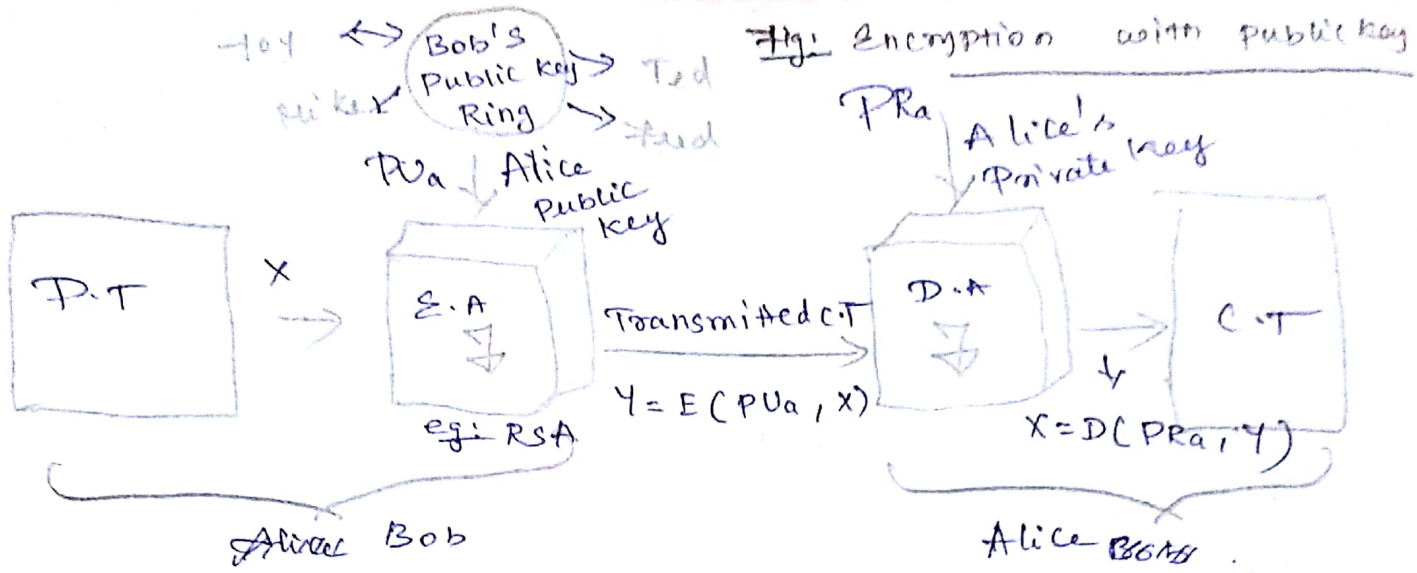
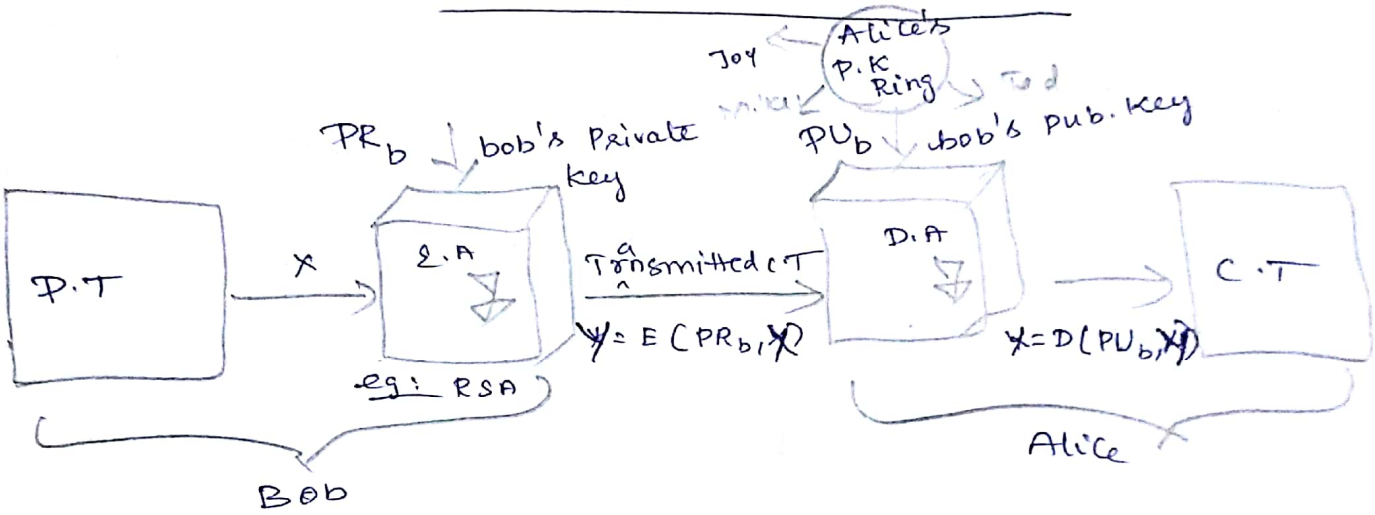


Fig: Encryption with public key



$X \rightarrow$ P.T $PU_a, PU_b \rightarrow$ Pub. keys

$Y \rightarrow$ C.T $PR_a, PR_b \rightarrow$ Pr. keys.

Fig: Public key cryptography

Differences

Conventional Encryption	Public key Encryption.
Work:-	
1. Same Algm with same key used for En & De	one Algm is used for En & De. with pair of keys, one for En & another for De.
2. The sender & Receiver must share the Algm & the key	The sender & receiver must each have one of matched pair of key

Security:

1. Key must be kept secret
2. It must be impossible to decipher a msg If no other Intm is available.
3. Knowledge of the Algm + samples of C.T must be insufficient to determine the key.

1. one of 2 key must be secret
It must be impossible to decipher a msg if no other intm is available.
Knowledge of the Algm + one of the keys + samples of C.T must be insufficient to determine the other key.

$P.T \rightarrow X = [X_1, X_2 \ldots X_M]$ → M elements of X

$C.T \rightarrow Y = [Y_1, Y_2 \ldots Y_N]$.

$Enc \rightarrow Y = E(PU_b, X)$

$De \rightarrow X = D(PR_b, Y)$

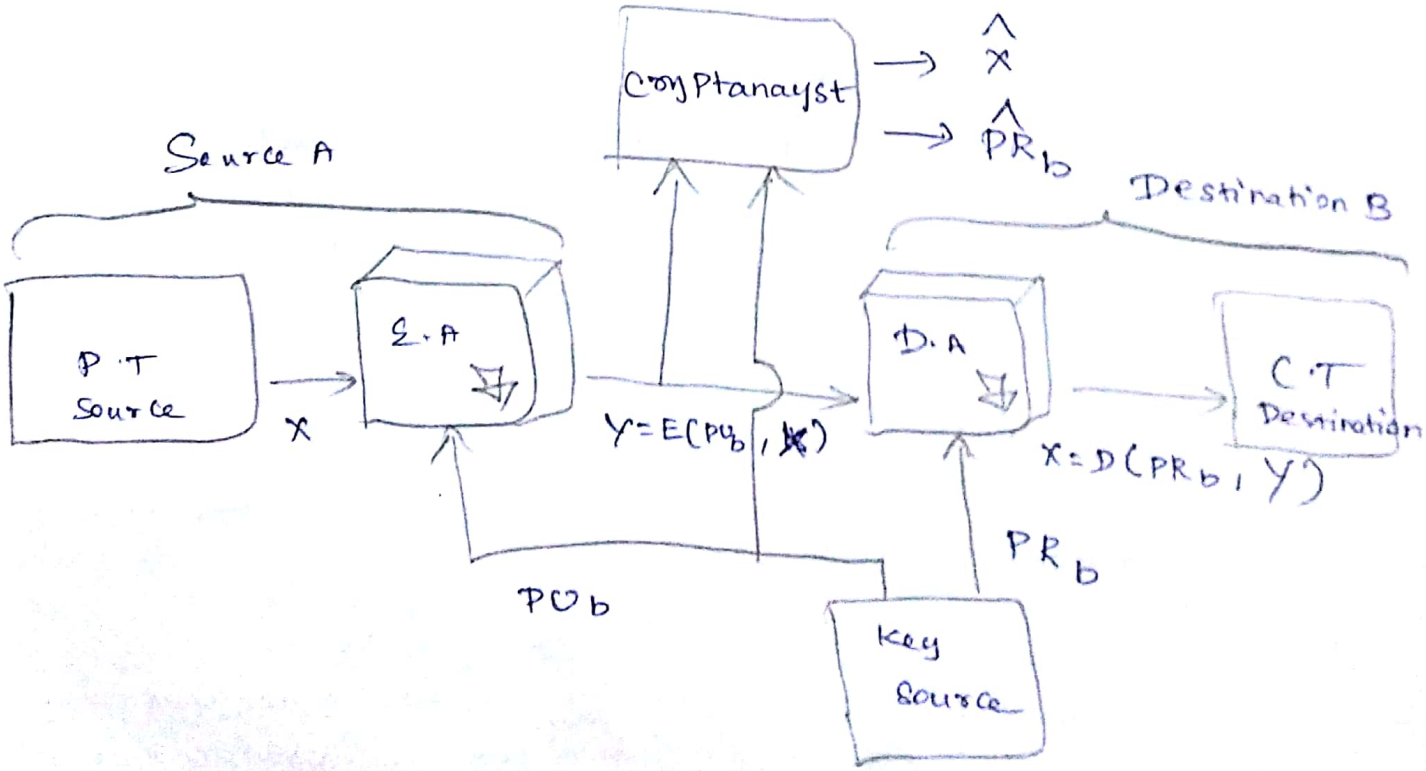


Fig: Public Key cryptosystem: Secrecy.

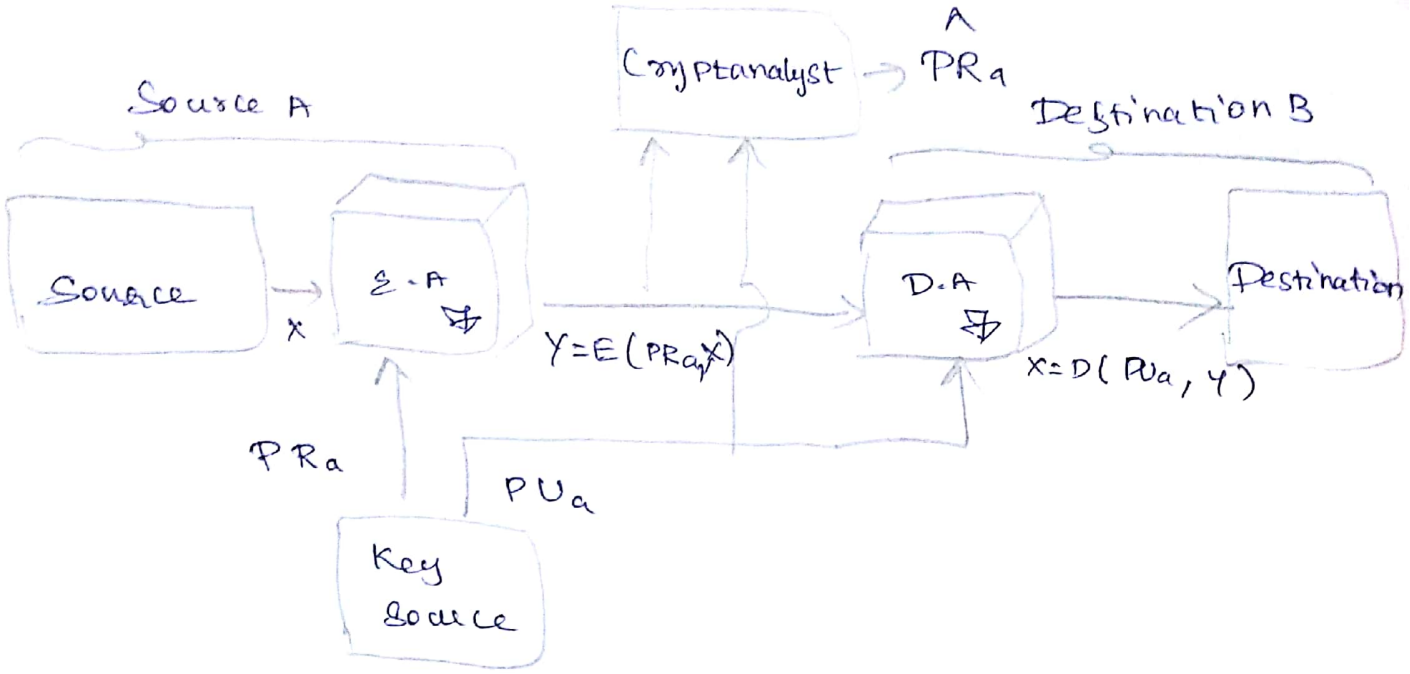


Fig: Public Key Cryptosystem : Authentication

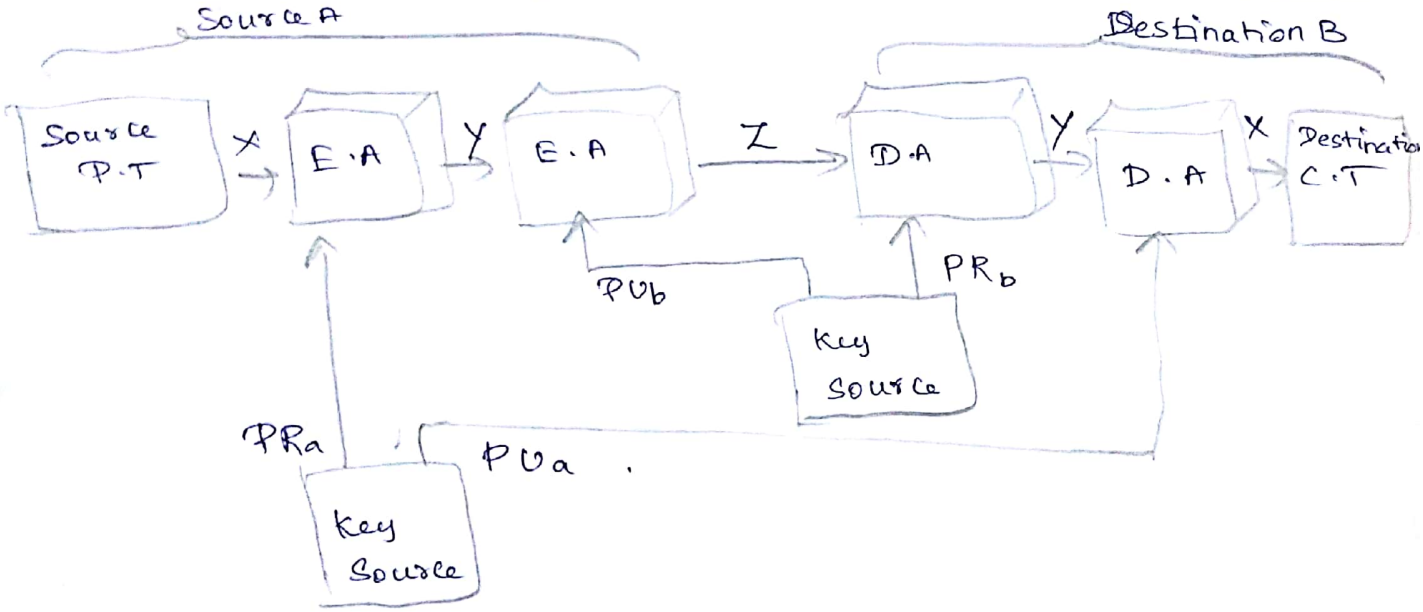


Fig: Public - key - Cryptosystem : Authentication & Secrecy

$$Z = E(PU_b, E(PR_a, X))$$

$$X = D(PU_a, D(PR_b, Z))$$

Public key cryptosystem :→ classify the use

1. Encryption / Decryption
2. Digital Signature
3. Key Exchange.

V) RSA Algorithm :-

Introduction :-

→ Asymmetric Key Encryption.

→ 2 Keys — Public
Private.

→ This Algm Proposed for Public key Cryptography.

→ RSA → General Purpose Approach to Public key Cryptography.

Ron Rivest, Adi Shamir, Len Adleman.

→ Developed in 1977 & Published in 1978.

→ RSA scheme is a block cipher in which the P.T & C.T are integers from 0 & n-1 for some n.

→ A typical size for n is 1024 bits (or) 309 decimal digits

i.e., n is $< 2^{1024}$.

Description of the Algorithm :-

→ It makes the use of Expression & Exponentials

→ P.T is En in blocks, each block has binary Value

→ i.e., block size $\leq \log_2(n) + 1$

→ block size is i bits where $2^i < n \leq 2^{i+1}$.

→ En & De are the following form for the some P.T block M.

En → $C = M^e \bmod n$

De → $M = C^d \bmod n$.

C → C.T e → Pub. key

M → P.T d → Pri. key.

→ Both Sender & Receiver must know the value of n.

→ Sender knows the value of e.

→ only Receiver knows the value of d.

→ Thus this is Public key Encryption Algm with Public key of $PU = \{e, n\}$

Private key of $PR = \{d, n\}$.

Requirements :— (must be met)

1. It is Possible to find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M < n$.

2. It is relatively easy to calculate $M^e \bmod n$ & $C^d \bmod n$ $\forall M < n$.

3. It is infeasible to determine d given e & n $M^{ed} \bmod n = M$.

e & d → Multiplicative Inverse Modulo of $\phi(n)$

where $\phi(n)$ → Euler's Totient Function.

→ P & q are Prime.

→ $\phi(pq) = (p-1)(q-1)$

→ the r/p b/w e & d expressed as

$$ed \equiv 1 \mod \phi(n)$$

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

→ rule → Modular Arithmetic → true → only If d is prime to $\phi(n)$.

→ $\gcd(\phi(n), d) = 1$.

RSA scheme → Ingredients :-

p, q → 2 Prime nos p ≠ q.

$n = p \times q$

e, with $\gcd(\phi(n), e) = 1$, $1 < e < \phi(n)$.

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

Resulting keys :-

$Pr.K = \{d, n\}$

$Pu.K = \{e, n\}$.

Encryption :-

$C = M^e \mod n$ & transmits C.

Decryption :-

$M = C^d \mod n$

Algorithm :-

I - Key Generation :-

Select p, q → Large Prime No's $p \neq q$.

Calculate $n = p \times q$

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

Select integer e → Pub. Key.

$\gcd(\phi(n), e) = 1 \quad 1 < e < \phi(n)$.

Calculated d → Pr. Key.

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

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

Public key.

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

Public key

$$PR = \{d, n\}$$

II Encryption by Bob with Alice Pu. Key.

$$C = M^e \bmod n, \quad M < n.$$

III Decryption by Alice with Alice Private key.

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

Ex: Problem :-

Given :-

$P = 17, \quad q = 11, \quad M = 88$

Soln :-

1. Prime no's $P = 17, q = 11$

2. Calculate $n = P \times q = 17 \times 11 = \boxed{187 = n}$

3. Find $\phi(n) = (P-1) \times (q-1) = (17-1)(11-1) = 16 \times 10$

$\boxed{\phi(n) = 160}$

4. Find e.

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

$gcd(160, ?) = 1$

7

$\boxed{e = 7}$

5. Determine d

$d \times e \mod \phi(n) = 1$

$? \times 7 \mod 160 = 1 \Rightarrow 161 \bmod 160 = 1$

23

$\boxed{d = 23}$ $\quad 1 = 1$ ✓

Resulting keys :-

$PU \rightarrow \{7, 187\} \rightarrow \{e, n\}$

$PR \rightarrow \{d, n\} \rightarrow \{23, 187\}$.

Encryption :-

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

$$= 88^7 \bmod 187$$

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

$$= (88 \times 77 \times 132) \bmod 187$$

$$= 894,432 \bmod 187$$

$$\boxed{C = 11}$$

$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 7744 \bmod 187 = 77$

$88^4 \bmod 187 = 59969536 \bmod 187 = 132$

Decryption :-

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

$$= 11^{23} \bmod 187$$

$$= [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187) \bmod 187]$$

$11^1 \bmod 187 = 11$

$11^2 \bmod 187 = 121$

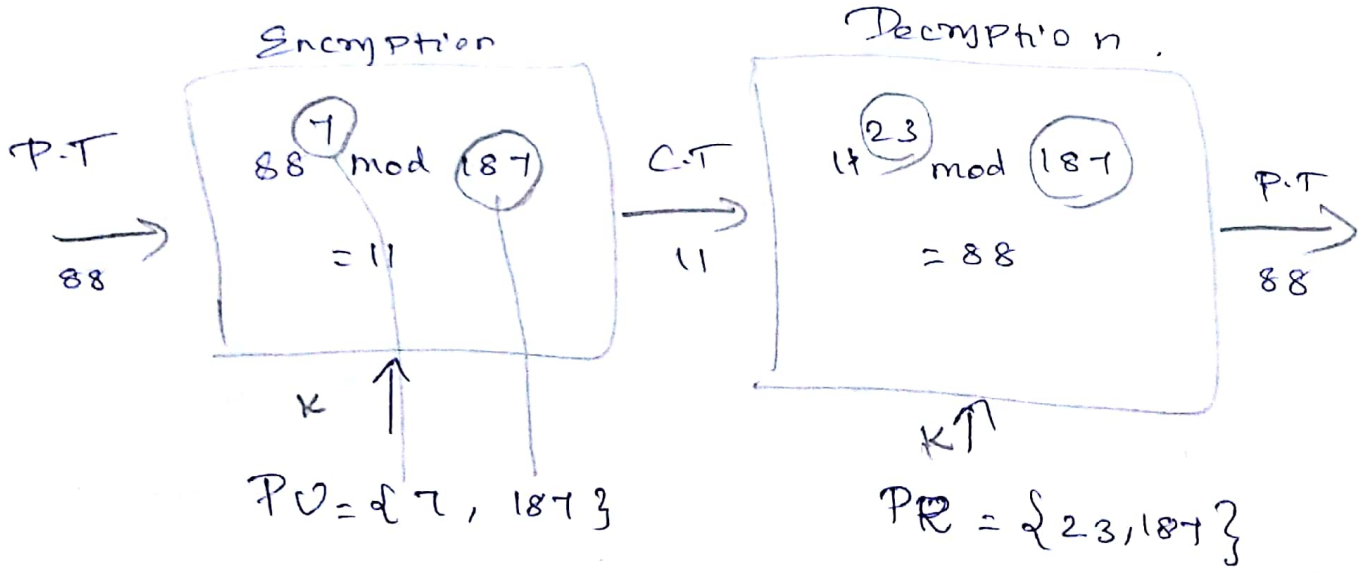
$11^4 \bmod 187 = 14641 \bmod 187 = 55$

$11^8 \bmod 187 = 214358881 \bmod 187 = 33$

$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187$

$= 79720245 \bmod 187$

$M = 88$

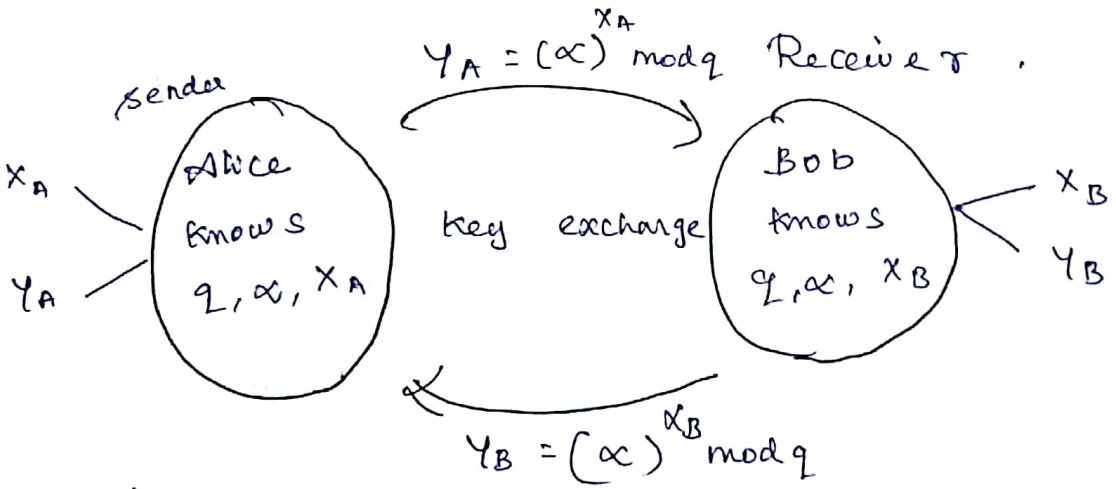


VI) Diffie - Hellman Key Exchange (1976)

→ First Practical Method for establishing a shared Secret key over an unsecured communication channel.

→ It defines Public key Cryptography.

→ It Generally refered to as Diffie - Hellman key Exchange Algorithm.



$\{X_A, X_B\}$ → Private key

$\{Y_A, Y_B\}$ → Public key.

Purpose

To Enable 2 users to securely exchange a key that can then be used for subsequent Encryption of messages.

Diffie Hellman Key Exchange Algorithm.

Global Public Elements.

q Prime Number.

α $\alpha < q$ and α is a Primitive root of q.

User A Key Generation.

Select Private X_A, $X_A < q$.

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, $Y_B = (\alpha)^{X_B} \bmod q$.

[Note :- Key exchange A → B].

Calculation of Secret Key by User A

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

Calculation of Secret Key by User B

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

Ex. Problem – 1

Given:

$q = 353$, $\alpha = 3$.

User A

Select $x_A = 97$

User B

select $x_B = 233$

↓

calculate y_A

$y_A = (\alpha)^{x_A} \bmod q$

$= (3)^{97} \bmod 353$

$y_A = 40$

↓

calculate y_B

$y_B = (\alpha)^{x_B} \bmod q$

$= (3)^{233} \bmod 353$

$y_B = 248$

$y_B = 248$

$y_A = 40$

Determine Secret key

$K = (y_B)^{x_A} \bmod q$

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

$K = 160$

Determine Secret key

$K = (y_A)^{x_B} \bmod q$

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

$K = 160$

∴ Secret key is same on both sides (i.e, communicating parties)

Ex. Problem :- 2

Given

$q = 23$, $\alpha = 5$.

User A	User B
Select private X_A	Select Private X_B
$X_A = 6$	$X_B = 15$.
Calculate Public Y_A	Calculate Public Y_B
$Y_A = (\alpha)^{X_A} \bmod q$	$Y_B = (\alpha)^{X_B} \bmod q$
$= (5)^6 \bmod 23$	$= (5)^{15} \bmod 23$
$Y_A = 8$	$Y_B = 19$
$Y_B = 19$	$Y_A = 8$
Determine secret key	Determine Secret key
$K = (Y_B)^{X_A} \bmod q$	$K = (Y_A)^{X_B} \bmod q$
$= (19)^6 \bmod 23$	$= (8)^{15} \bmod 23$
$K = 2$	$K = 2$

Security Attacks

It is possible attacks can mount in Diffie Hellman key Exchange Algorithm.

1. Brute force Attack
2. Man-in-the-Middle Attack.

Brute Force Attack

→ Attacker (E) → can determine the common key by discovering a solution to the equation. (∵ for eg. problem-1)

$$x_a^a \bmod q = \text{public key}_a$$

$$3^a \bmod 353 = 40 \text{ (or)}$$

$$3^b \bmod 353 = 248.$$

→ The Brute force Approach is to calculate Powers of 3 mod 353, Stopping when the result equals either 40 (or) 248.

→ The desired Answer is

$$3^{94} \bmod 353 = 40$$

Avoid:

with Larger numbers, the Problem becomes impractical.

Key Exchange Protocols.

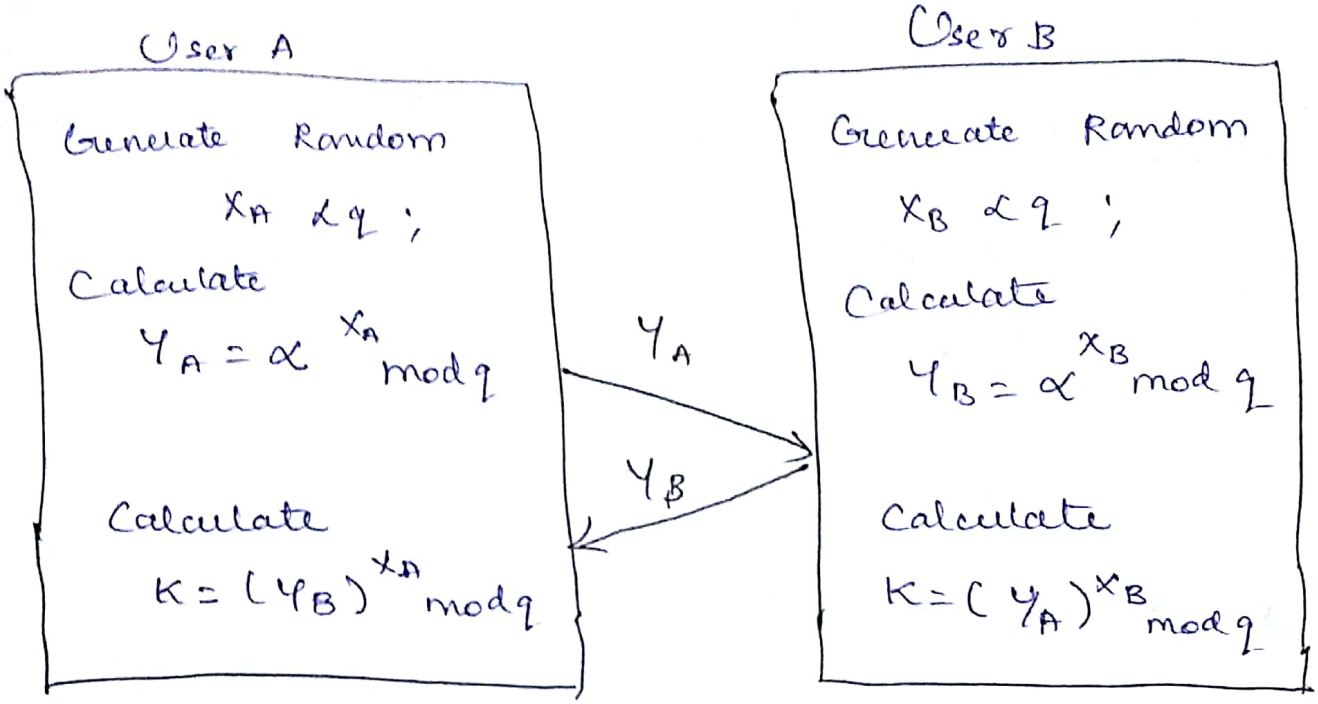


Fig: Diffie Hellman Key Exchang.

Man-in-the-Middle Attack.

→ Key exchange protocol is insecure against a man-in-the-middle Attack.

→ Suppose Alice & bob wish to exchange keys and Oath is adversary.

Steps:—

1) Oath prepares for the Attack by generating 2 random Private key X_{D_1} & X_{D_2} & then computing the corresponding Public keys Y_{D_1} & Y_{D_2}.

2) Alice transmits Y_A to Bob.

3) Darth intercepts X_A & transmit Y_{D1} to Bob. Darth also calculates $K_2 = (Y_A)^{X_{D2}} \bmod q$.

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

5) Bob transmits Y_B to Alice

6) Darth intercepts Y_B & transmits Y_{D2} to Alice. Darth calculates $K = (Y_B)^{X_{D1}} \bmod q$.

7) Alice Receives Y_{D2} & calculates $K_2 = (Y_{D2})^{X_B} \bmod q$

(∴ At this point Bob & Alice think that they share a secret key but instead bob & Darth share secret key K & Alice & Darth share Secret key K_2)

All future comm b/w Bob & Alice is

1) Alice sends encrypted msg $M : E(K_2, M)$.

2) Darth intercepts the encrypted message & decrypt it to receiver M.

3) Darth sends Bob $E(K_1, M)$ (or) $E(K_1, M')$

M' → any message.

Case : 1 → Darth simply wants to eavesdrop on the comm without altering it.

Case : 2 → Darth wants to modify the message going to bob.