UNIT III

PUBLIC KEY CRYPTOGRAPHY MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY

TOPICS: PUBLIC KEY CRYPTOGRAPHY MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY: Primes – Primality Testing – Factorization – Euler's totient function, Fermat's and Euler's Theorem - Chinese Remainder Theorem – Exponentiation and logarithm ASYMMETRIC KEY CIPHERS: RSA cryptosystem – Key distribution – Key management – Diffie Hellman key exchange - ElGamal cryptosystem – Elliptic curve arithmetic-Elliptic curvecryptography.

PRIME NUMBERS

Prime numbers

Prime numbers have divisors of 1 and its number itself.

Prime factorisation

To compute **GCD** of any two numbers in prime factorization approach we need tofind **prime factors** of the two numbers.

Fermat Theorem or Fermat's little theorem

If **a** belongs to integer , P is a prime number that does not divide **a** then a^P congruent a (mod P)ie.,

$a^p \equiv a \pmod P$

In special case

$a^{P-1} \equiv 1 \pmod P$ if GCD(a,P)=1 . where a and p are coprime.

It is mainly used to solve modular exponentiation.Eg.

Compute the value of $2^{10} \bmod 11$

$2^{10} \equiv 1 \pmod{11}$

Eg. Compute the value of $2^{340} \bmod 11$

$(2^{340})=(2^{10})^{34} \bmod 11$

$=1^{34} \bmod 11 \Rightarrow 1$

//Proof.

Take division algorithm

a = p.q + r where can be 0<= r <= p-1

let g.c.d (a,p) ie coprime

a is not divisible by p hence 1<= r <=p-1

fact says that if **a** leaves remainder r where 1<= r<= p-1 on dividing by **p** then ka,1<=k<= p-1 also leaves remainders from 1 to p-1.

Means
If a,2a,3a,…(p-1)a surely gives remainders 1,2,3,…(p-1)
So if multiply
a*2a*3a*…*(p-1)a ≡ 1.2.3…(p-1) (mod p)hence a^{p-1}.(p-1)! ≡ (p-1)! (mod p)
which returns
a^{p-1} ≡ 1 (mod p) [as mod p cannot divide (p-1)!]
hence proved. //

Eg.6^{10} mod 11
Sol. 6^{11-1} mod 11
= 1 [as per theorem]
Eg. 5^{15} mod 13
=(5^{2} mod 13) * (5^{13} mod 13)
=(25 mod 13) * 5
=(12 * 5) mod 13
=60 mod 13
=8.

Euler's theorem

If **n** and **a** are coprime positive intergers
then $a^{phi(n)}$ ≡1 (mod)n
In this theorem phi(n) = n-1.
n is prime number and phi(n) is Euler's phi function.
Euler's phi function is also called Euler's totient function and hence named asEuler's totient theorem or Euler's theorem.

Euler's phi function or Euler's totient function(

Euler's phi function phi (n) returns the numbers of integers from 1 to n,that are relativelyprime to n.
The phi function is computed phi(n) using various methods. They are

1.If n is a prime number then phi (n) = n-12.If
n is a composite number then
2.1 Find the prime factors of that number and compute the phi function value asused in step 1.otherwise
2.2.Find prime powers (p^{n})of the given number n. For computing the phi value of prime powers we have to use the formula
($p^{a} - p^{a-1}$).

Eg. Compute Eulers's totient function for the values 3,81.
phi(3)=3-1=2
2. phi(8)= 2^{3}
$= 2^{3} - 2^{3-1}$ (since $p^{a} - p^{a-1}$) $= 2^{3} - 2^{2} = 8-4=4$

Primality Testing Methods

Primality testing method is a method to find and to prove whether the given numberis prime number.

1. Naive Algorithm:

Naïve Algorithm is used to divide the given input number P by all the integersstarting from 2 to root of P — 1
If any one of them is a divisor, then the input number P is not a prime. Otherwise, itis considered as a prime number

Algorithm:

1. Pick any integer P that is greater than 2
2. Try to divide P by all integers starting from 2 to the square root of P
3. If P is divisible by any one of these integer, we can conclude that P is a composite
4. Else P is a prime number

Example:
Find the primality test for the number 100 using naïve algorithm.1.
P=100
2. 2,3,4,5,6,7,8,9
3. Case 1: 100/2 = 50(composite)

Therefore, 100 is not a prime number.

2. Fermat's Primality Test:

If P is a prime and **P** does not divide **a,** which is a natural number then $a^{P-1} \equiv 1 \pmod{P}$

Example:
Check whether the given number 12 is prime number or not using Fermat'stheorem
Given P = 12

To check whether 12 is prime number or not, we have to check $a^{P-1} \equiv 1 \pmod{P}$
$a^{12-1} \equiv 1 \pmod{12}$
$a^{11} \equiv 1 \pmod{12}$
Where 1 <= a < 12
We have to calculate $a^{11} \bmod 12$

If it is equal to 1, then it is called prime number. Otherwise, it is called composite number.

Consider, $a = 5^{11}$
$\equiv 1 \pmod{12}$
(i.e) $5^{11} \bmod 12 = 5$
It is not equal to 1
Therefore it is not a prime number

3. Miller – Rabin Primality Test

Function Miller-Rabin (x)
$x-1 = (2^w)y$ //x is the input number for primality test
// y is an odd number
select **a** randomly in the range [2, (x-1)]
$Z = a^y \bmod x$
if Z congruent 1 (mod x) then return primefor
i = 1 to w — 1
{
If Z congruent -1 (mod x) then return prime
$Z = Z^2 \bmod x$
}
return composite

Example:
Find the primality for 7x = 7
As per algorithm, $x - 1 = 7 - 1 = 6 = 2^1 \times 3$
x = 7, w = 1, y = 3
$Z = a^y \bmod 7$
a = 2 (randomly), where [1<=a<=x-1]
$Z = 2^3 \bmod 7 = 1$
Value of Z = 1, 7 is concluded as prime number

Chinese Remainder Theorem

States that when the moduli of a system of linear congruencies are pairwise prime, thereis a unique solution of the system modulo, the product of the moduli.

$x \equiv a \pmod{m}$.

Chinese mathematician Sun Tsu Suan-Ching asking the following problem:
—There are certain things whose number is unknown. When divided by 3, the remainder is 2; when divided by 5, the remainder is 3; and when divided by 7, theremainder is 2. What will be the number of things?
(Otherwise) Mangos are divided into groups consisting of 3 mangos in each group remaining is 2. If the mangos are divided into groups consisting of 5 mangos in eachgroup remaining 3. If mangos are divided into groups consisting of 7 mangos in each groupremaining 2.Totally how many mangos are available?

- $x \equiv a1 \bmod(m1)$
- $x \equiv a2 \bmod(m2)$
- $x \equiv a3 \bmod(m3)$

$$x = \Sigma(a_i M_i y_i) = (a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3) \bmod M$$

Let M1, M2, …, Mn be (pairwise) relatively prime numbers. Then the system:Step1: Calculate M

- M = m1*m2*m3 … mn.

Step 2: Calculate M_k =M/m_k Step
3: Find Inverse of M_K (ie)y_k

Find the X using CRT

- x ≡ 2 mod(3)
- x ≡ 3 mod(5)
- x ≡ 2 mod(7)
- $a_1 = 2$, $a_2 = 3$, $a_3 = 2$; $m_1 = 3$, $m_2 = 5$, $m_3 = 7$;
  i. M = m_1 x m_2 x m_3 = 105.
  ii. For each equation calculate

M_k =M/m_k (ie) $M_1 = M / m_1 = 105 / 3 = 35$
$M_2 = M / m_2 = 105 / 5 = 21$
$M_3 = M / m_3 = 105 / 7 = 15$

iii. inverse of M_k **(ie)** y_k
inverse of M_1 **(ie)** $y_1 = 35^{-1} \bmod (3) = 35^{3-2} \bmod (3) = 2$ [since Fermat's inverse theorem or easy inverse method like 35 x ? mod 3=1 (ie) 2]

y_2 =1 ; y_3 =1
X = $\Sigma(a_i M_i y_i) = (a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3) \bmod$
MX =[(2 x 35 x 2) + (21 x 3 x 1) + (2 x 15 x 1)] mod 105
= (140 + 63 + 30) mod 105
=233 mod 105
X =23

Exponentiation

- Exponentiation is a type of operation where two elements are used in which one element is considered as a base element and another as an exponential element.
- For example, b is an example of exponential operation where x is a base elementand y is an exponential element.
- When y is a positive integer, exponentiation is performed in a similar way to repeatedmultiplication is performed.
- Modular exponentiation is a type of exponentiation in which a modulo division operation is performed after performing an exponentiation operation.
- For example, **(x^y mod n),** where n is an integer number.
- The exponentiation is an important concept discussed in many cryptographic algorithms such as RSA, Diffie-Hellman, Elgamal, etc.,

Example:1
11^7 mod 13
11^2 mod 13 = 121 mod 13 = 4
11^4 mod 13 = (11^2 mod 13 x 11^2 mod 13) mod 13
= (4 x 4) mod 13
= 16 mod 13
= 3
11^7 mod 13 = (11^4 mod 13 x 11^2 mod 13 x 11^1 mod 13) mod 13
= (3 x 4 x 11) mod 13

= (132) mod 13
= 2

Find the result of 2^{90} mod 13.

Solution:

Step 1: Split x and y into smaller parts using exponential rules as shown below: 2^{90} mod 13 = 2^{50} x 2^{40}
Step 2: Calculate mod n for each part
2^{50} mod 13 = 1125899906842624 mod 13=4

2^{40} mod 13 = 1099511627776 mod 13 = 3

Step 3: Use modular multiplication properties to combine these 2 parts, we have

2^{90} mod 13 = (2^{50} x 2^{40}) mod 13

= (2^{50} mod 13 x 2^{40} mod 13) mod 13

= (4 x 3)mod 13 = (12) mod 13 = 12

Logarithms or Indices

- Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups. If G is a multiplicative cyclic group and g is a generator of G, then fromthe definition of cyclic groups, we know every element h in G can be written as g^x for some x. The discrete logarithm to the base g of h in the group G is defined to be x .
- For example, if the group is Z_5^*, and the generator is 2, then the discrete logarithm of 1 is 4 because $2^4 \equiv 1 \bmod 5$.
- Input: p - prime number, a- primitive root of p, b - a residue mod p.
- Goal: Find k such that $a^k = b \pmod p$. (In other words, find the position of y in the large list of $\{a, a^2, \ldots, a^{q-1}\}$.
- 14 is a primitive root of 19.
- For example $L_{14}(5) = 10 \bmod 19$, because $14^{10} = 5 \pmod{19}$.
- the inverse problem to exponentiation is to find the **discrete logarithm** of a number modulo p
- that is to find x where $a^x = b \bmod p$
- written as $x=\log_a b \bmod p$ or $x=\text{ind}_{a,p}(b)$
- if a is a primitive root then always exists

ASYMMETRIC KEY

CIPHERSPUBLIC KEY CRYPTOGRAPHY:

Principles of public key cryptosystems

The concept of public key cryptography evolved from an attempt to attack two of themostdifficult problems associated with symmetric encryption. Key distribution under symmetric key encryption requires either

(1) Two communicantsalready share a key, which someone has been distributed to them

(2) The use of a key distribution center.

- Digital signatures.

Characteristics of Public key cryptosystems

Public key algorithms rely on one key for encryption and a different but related key for decryption.These algorithms have the following important characteristics:

- It is computationally infeasible to determine the decryption key given only theknowledgeof the cryptographic algorithm and the encryption key.

In addition, some algorithms, such as RSA, also exhibit the following characteristic:

- Either of the two related keys can be used for encryption, with the other usedfor decryption.

INGREDIANTS OF PUBLIC KEY CRYPTOGRAPHY

1. Plaintext: This is the readable message or data that is fed into the algorithm as input.
2. Encryption algorithm: The encryption algorithm performs various transformations on theplaintext.
3. **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
4. Ciphertext: This is the scrambled message produced as output. It depends on the plaintextand the key. For a given message, two different keys will produce two different cipher texts.
5. Decryption algorithm: This algorithm accepts the ciphertext and the matching key andproduces the original plaintext.

Encryption:

The essential steps are the following:

1. Each user generates a pair of keys to be used for encryption and decryptionof messages.
2. Each user places one of the two keys in a public register or other accessible file.This is the public key. The companion key is kept private.
3. If A wishes to send a confidential message to B, A encrypts the message usingB"s public key.
4. When B receives the message, it decrypts using its private key.

With this approach(Fig), all participants have access to public keys and private keys are generatedlocally by each participant and therefore, need not be distributed.

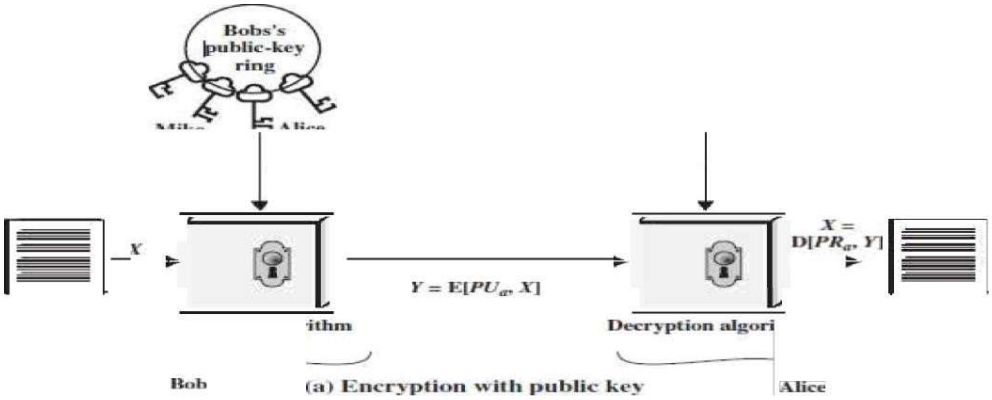


Fig . Public Key Cryptography For Authentication

Let the plaintext be X=[X1, X2. X3, . .,X] where m is the number of letters in some finite alphabets. Suppose A wishes to send a message to B.

B generates a pair of keys: a public key KUb and a private key KRb KRb is known only toB, whereas KUb is publicly available and therefore accessible by A.

With the message X and encryption key KUb as input, A forms the cipher text Y=[Y1, Y2, Y3›

i.e., Y=EKUb(X)

The receiver can decrypt it using the private key KRb i.e., X=D KRb(Y)

The other approach (using sender—s private key for encryption and sender—s public key for decryption) will provide authentication which is illustrated in the following diagram(Fig 2.26).

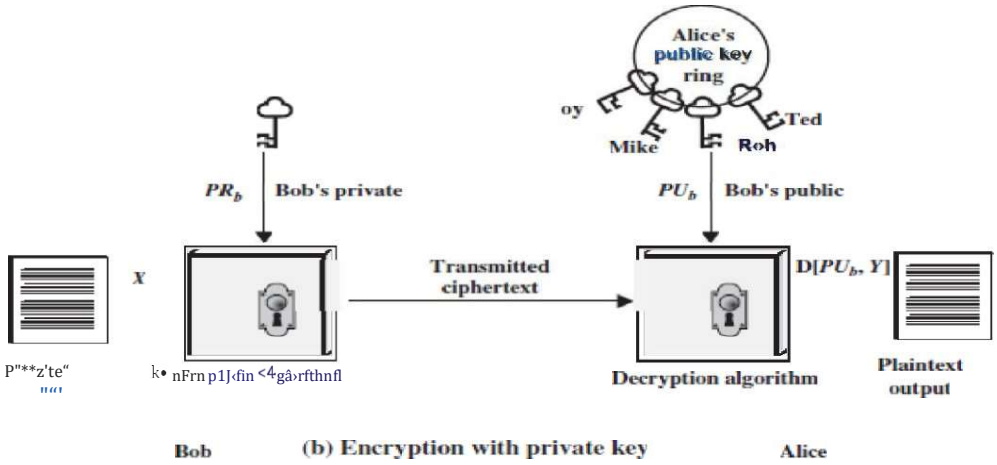


Fig .Private Key Cryptography For Authentication

The encrypted message serves as a digital signature. It is important to emphasize that the encryption process just described does not provide confidentiality.

Public Key Cryptography for Security

There is some source A that produces a message in plaintext, *X* —— [X 1, +2, . ,X]. Theelements of X are letters in some finite alphabet.

The message is intended for destination B. B generates a related pair of keys: a publickey, *PUT,* and a private key, *PRb* PRtis known only to B, whereas PL/his publicly available andtherefore accessible by A. With the message and the encryption key PL/,as input, A forms the ciphertext *Y* =[Y 1. *Yo. . ,YN]*

$Y = E(PU_B, X)$

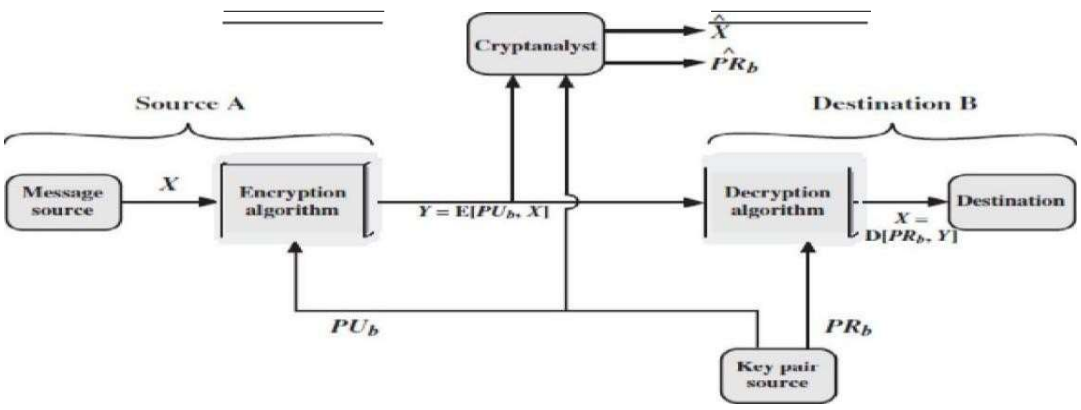


Figure.PubIic-Key Cryptosystem: Secrecy

The intended receiver, in possession of the matching private key, is able to invertthe transformation:

$$X\text{—— D (PRd›}$$

An adversary, observing *Y* and having access to *PUT,* but not having access to *PRbor X,* must attempt to recover X and/or *PRb* It is assumed that the adversary does have knowledge of the encryption (E) and decryption (D) algorithms.

If the adversary is interested only in this particular message, then the focus of effort is to recover A by generating a plaintext estimate X *. Often, however, the adversary is interested in being able to read future messages as well, in which case an attempt is made to recover PRtby generating an estimate *PRb.*

$$Y = E(PR_{a1})$$
$$X = D(PU_{a1}$$
$$Y)$$

In this case, A prepares a message to B and encrypts it using A‘s private key before transmitting it. B can decrypt the message using A‘s public key. Because the message was encrypted using A‘s private key, only A could have prepared the message.

Therefore, the entire encrypted message serves as **a digital signature.** In addition, it is impossible to alter the message without access to A‘s private key, so the message is authenticated both in terms of source and in terms of data integrity.

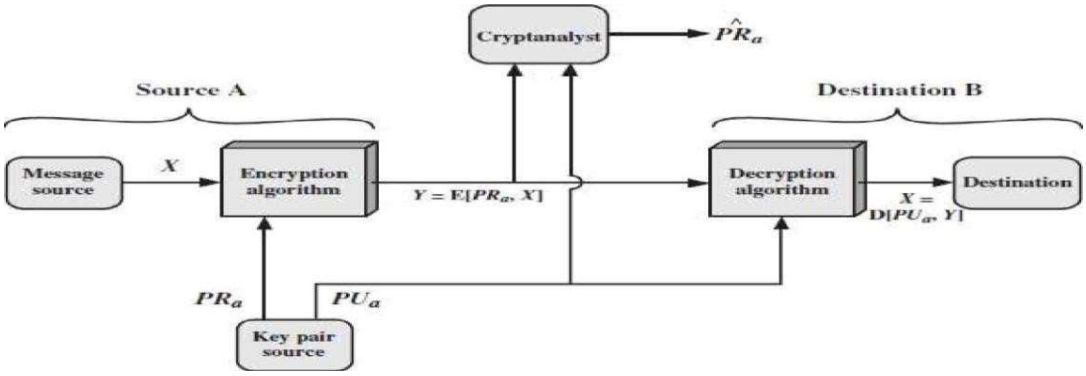


Figure .Public-Key Cryptosystem: Authentication

It is important to emphasize that the encryption process depicted in above Figures does not provide confidentiality. That is, the message being sent is safe from alteration but not from eaves dropping. This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear. Even in the case of complete encryption, as shown in Figure13, there is no protection of confidentiality because any observer can decrypt the message by using the sender‘s public key.

Authentication and Secrecy

It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme (Figure 2.29):

Ciphertext Z = EKUb[EKRa (X)]

Plaintext X = EKUa[EKRb (Y)]

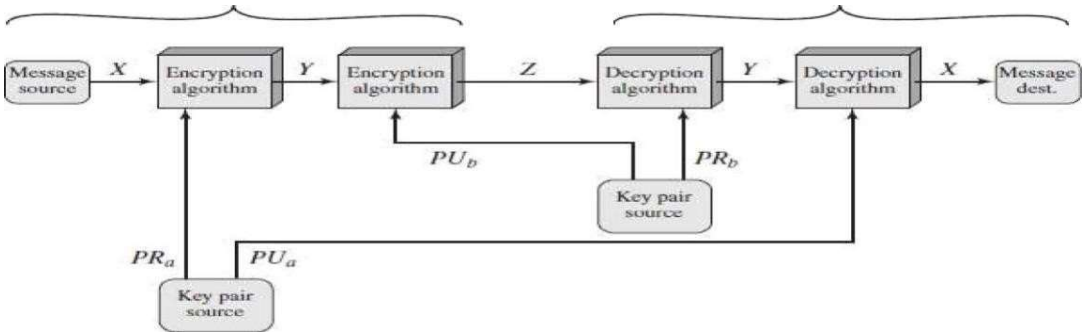


Figure .PubIic-Key Cryptosystem: Authentication and **Secrecy**

Initially, the message is encrypted using the sender—s private key. This provides the digital signature. Next, we encrypt again, using the receiver—s public key. The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus confidentiality is provided.

Applications for Public-Key Cryptosystems

We can classify the use of public-key cryptosystems into three categories

1. Encryption /decryption: The sender encrypts a message with the recipient‘s public key.

2. Digital signature: The sender —signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
3. Key exchange: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

Requirements for public key cryptography

- It is computationally easy for a party B to generate a pair [KUb ,KRb]
- It is computationally easy for a sender A, knowing the public key and the message to beencrypted M, to generate the corresponding ciphertext: C=EKUb(M).
- It is computationally easy for the receiver B to decrypt the resulting ciphertext usingthe private key to recover the original message:

  M = DKRb (C) = DKRb [E KUb (M)]

- It is computationally infeasible for an opponent, knowing the public key KUb, todetermine the private key KRb
- It is computationally infeasible for an opponent, knowing the public key KUb, and aciphertext C, to recover the original message M.
- The encryption and decryption functions can be applied in either order:

  M = EKUb [D KRb (M) = DKUb [E KRb (M)]

Public-Key Cryptanalysis

Attack Type 1 :

The public-key encryption scheme is vulnerable to a brute-force attack; therefore use large key.The tradeoff is that makes use of some sort of invertible mathematical function.

Therefore choose key size such that the brute force attack is not possible, at the same time should not be too slow for general use.

Attack type 2:

Attack is of other types (i.e.) given the algorithm and the public key deduce private key. This method has not been successful till date.

Attack Type 3:

A probable-message attack. When a confidential message is to be transmitted using DES,the attacker will find all 2^{56} possible keys using the public key and discover the encrypted
key by matching the generated cipher text and the actual cipher. This attack can be avoided byappending some random bits to the message.

RSA ALGORITHM

It was developed by Rivest, Shamir and Adleman. This algorithm makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n.

The RSA scheme is a cipher in which the plaintext and cipher text are integers between 0 and n - 1 for some n. A typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than

2 1024

That is, the block size must be less than or equal to Iog 2 (n); in practice, the block size is k-b its, where $2^k < n < {}^{k}{}_{2}{}^{*}{}^{l}$. Encryption and decryption are of the following form, for some plaintext

block M and ciphertext block C:

$$c = M^{e} \bmod n \qquad e\ d \qquad e$$

$$M = C \bmod n = (M \bmod n = * \bmod n$$

Both the sender and receiver know the value of n. the sender knows the value of e andonly the receiver knows the value of d. thus, this is a public key encryption algorithm with a public key of KU = {e, n) and a private key of KR = {d, n}. For this algorithm to be satisfactory for public key encryption, the following requirements must be met:

1. It is possible to find values of e, d, n such that $M^{ed} = M \bmod n$ for all M < n.
2. It is relatively easy to calculate M^e and C^d for all values of M < n.
3. It is infeasible to determine d given e and n.

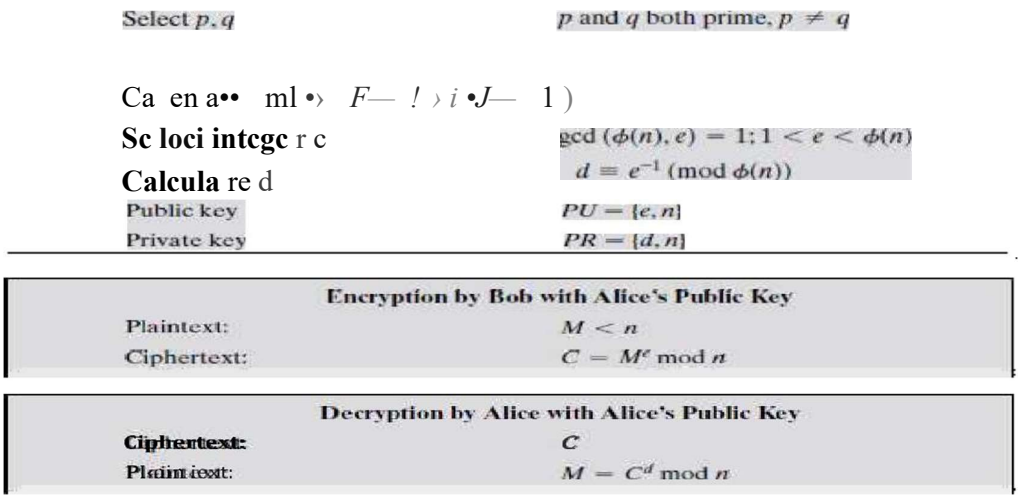


Fig . The RSA Algorithm

Let us focus on the first requirement. We need to find the relationship of the form:

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

Given two prime numbers p and q and two integers, n and m, such that n=pq and 0<m<n, and arbitrary integer k, the following relationship holds

$$k6(n)+1 \quad k(p-1)(q-1)+1 \quad m \bmod n$$

where 6(n) —— Euler totient function, which is the number of positive integers less than nand relatively prime to n. we can achieve the desired relationship, if

$$ed = k6(n)+1$$

This is equivalent to saying:

$$ed - 1_1 \bmod 6(n)$$

d = e- mod 6(n)

That is, e and d are multiplicative inverses mod 6(n). According to the rule of modular arithmetic,this is true only if d (and therefore e) is relatively prime to 6(n). Equivalently, gcd(6(n), d) = 1.

We are now ready to state the RSA scheme. The ingredients are the following:

p, q, two prime numbers	(private, chosen)
n —— *pq*	(public, calculated)
e, with gcd(6 *(n),* e) = 1; 1 <e < 6 *(n)*	(public, chosen)
d-Ke-1 (mod 6 (n))	(private,

calculated)The steps involved in RSA algorithm for generating the

key are

- Select two prime numbers, p = 17 and q = 11.

Calculate n = p*q = 17*11 = 187
Calculate 6(n) = (p-1)(q-1) = 16*10 =
160.
Select e such that e is relatively prime to 6(n) = 160 and less than 6(n); we choose

Determine *d* such that *de* K 1 (mod 160) and *d* < 160.The correct value is *d* —— 23, because 23 • 7 = 161 = (1 • 160) + 1; *d* can be calculated using the extended Euclid‘s algorithm

The resulting keys are public key *PU* —— {7, 187} and private key *PR* —— {23, 187}.

The example shows the use of these keys for a plaintext input of *M*—— 88. For encryption, we need to calculate C = 887 mod 187.

Exploiting the properties of modular arithmetic, we can do this as follows.

$88^7 \bmod 187 = [(88^4 \bmod 187) \bullet (88^2 \bmod 187) \text{x} (88^1 \bmod 187)] \bmod 187$
$88^1 \bmod 187 = 88$
$88^2 \bmod 187 = 7744 \bmod 187 = 77$
$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$
$88^7 \bmod 187 = (88 \bullet 77 \bullet 132) \bmod 187 = 894{,}432 \bmod 187 = 11$

For decryption, we calculate *M* —— 1123 mod 187:

$11^{23} \bmod 187 = [(11^1 \bmod 187) \bullet (11^2 \bmod 187) \bullet (11^4 \bmod 187) \bullet (11\text{‘} \bmod 187) \bullet (11 \bmod 187)] \bmod 187$
$11 \bmod 187 = 11$
$\bmod 187 = 121$
$11 \bmod 187 = 14{,}641 \bmod 187 = 55$
$11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33$
$11^{23} \bmod 187 = (11 \bullet 121 \bullet 55 \bullet 33 \bullet 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$

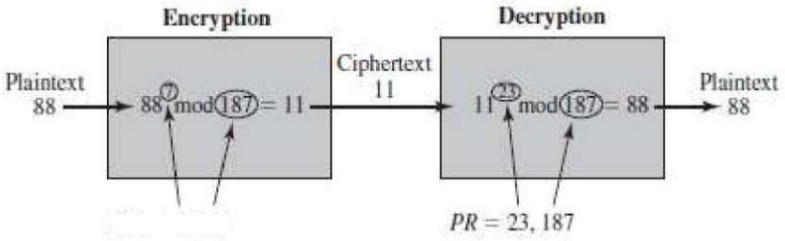


Fig . Example of RSA Algorithm

Security of RSA:

There are three approaches to attack the RSA:

1. Brute force: This involves trying all possible private keys.
2. Mathematical attacks: There are several approaches, all equivalent in effort to factoring the product of two primes.
3. Timing attacks: These depend on the running time of the decryption algorithm.

Type 1 RSA Attack: Defense to Brute Force attack:

Use large key space (i.e) large number of bits in e and d the better secured but problems are,

1. Increases computing power
2. Factoring Problem

Type 2 RSA Attack: Mathematical Attack:

Mathematical approach takes 3 forms:

- Factor n = p*q, hence find 6(n) and then d.
- Determine 6(n) directly without determining p and q and find d. d=e- (mode (n))
- Find d directly, without first determination 6(n).

Type 3 RSA Attack: Timing attacks:

This attack is learning for 2 reasons

1. Comes completely from unexpected direction
2. Cipher text only attack

Attack:

If the system does lastly the modular multiplication in majority of cases but takes longertime in few cases. The average is also longer.

The attack is done bit by bit

Start with left most bit

b,,

Suppose first j bits are known.

For a given cipher text the attacker completes the j iteration.

If the bit is set then d<- (d * a) mod n.

Methods to overcome Timing attacks:

1. Constant exponentiation time: All exponentiations take the same amount of time beforereturning a result. This is a simple fix but does degrade performance.
2. Random delay: Better performance could be achieved by adding a random delay to theexponentiation algorithm to confuse the timing attack.
3. Blinding: Multiply the cipher text by a random number before performing exponentiation. This process prevents the attacker from knowing what cipher text bits are being processedinside the computer and therefore prevents the bit-by-bit analysis essential to the timing attack.

KEY MANAGEMENT

There are two uses of public key cryptography regarding the issues of key distribution. They are

1. Distribution of public keys
2. Use of public key encryption to distribute secret keys

Distribution of Public Keys

Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:

a) Public announcement
b) Publicly available directory
c) Public-key authority
d) Public-key certificates

(a) Public Announcement of Public Keys

In public-key encryption the public key is public. Thus, if there is some broadly accepted public-key algorithm, such as RSA, any participant can send his or her public key toany other participant or broadcast the key to the community at large as shown in Figure 2.32.

Figure. Uncontrolled Public-Key Distribution

Disadvantage:

Anyone can forge such a public announcement. That is, some user could pretend tobe user A and send a public key to another participant or broadcast such a public key.

Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A and can use the forged keys for authentication.

(b) Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization as shown in Figure 2.33. Such a scheme would include the following elements:

1. The authority maintains a directory with a (name, public key} entry for each participant.
2. Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.
3. A participant may replace the existing key with a new one at any time, due to either the key has been used for a large amount of data, or the corresponding private key has been compromised in some way.
4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory

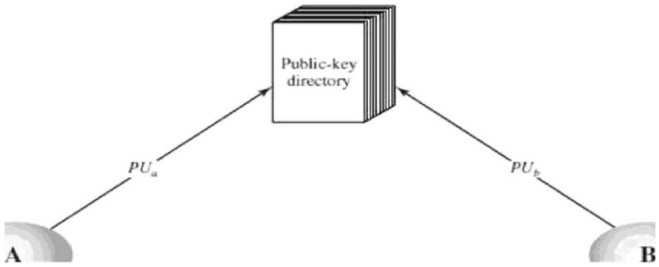


Figure Public-Key Publications

Vulnerabilities:

6 Tamper the records of public key directories.

6 If an adversary succeeds in obtaining or computing the private key of the directory authority, the adversary could authoritatively pass out counterfeit public keys and impersonate any participant and eavesdrop on messages sent to any participant.

(c) Public-Key Authority

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory. A typical scenario is illustrated in Figure 2.34.

As before, the scenario assumes that a central authority maintains a dynamic directory of public keys of all participants. In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key. The following steps (matched by number to Figure 2.34) occur:

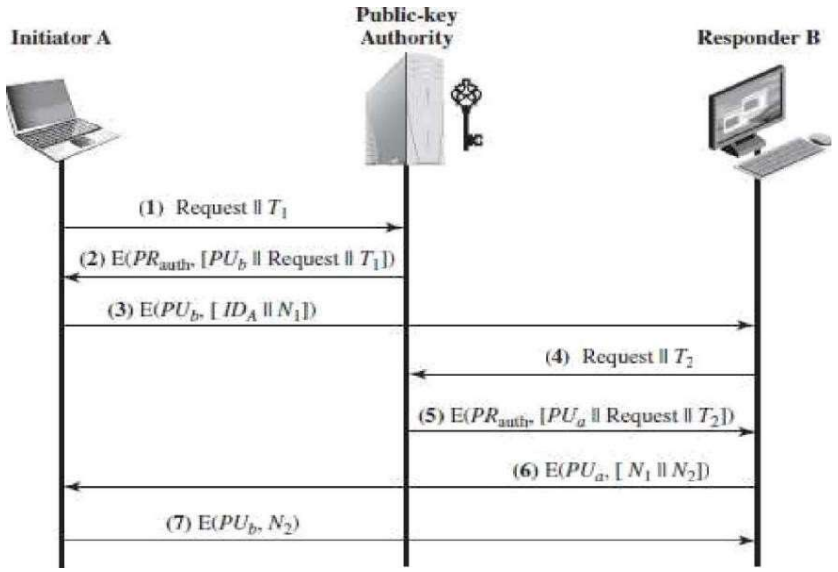


Figure .PubIic-Key Distribution Scenario

r

1. A sends a time stamped message to the public-key authority containing a request for the current public key of B.

2. The authority responds with a message that is encrypted using the authority‘s private key, *PRauthT* hus, A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:

- B‘s public key, PL/,which A can use to encrypt messages destined for B
- The original request, to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority
- The original timestamp, so A can determine that this is not an old message from the authority containing a key other than B‘s current public key

3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A *(IDA)* and a nonce (/\/1), which is used to identify this transaction uniquely.

4,5 B retrieves A‘s public key from the authority in the same manner as A retrieved B‘s public key.

6.B sends a message to A encrypted with K aand containing A‘s nonce (/V1) as well as a new nonce generated by B (/V2) Because only B could have decrypted message (3), the presence of /V1 in message (6) assures A that the correspondent is B.

7. A returns 2. encrypted using B‘s public key, to assure B that its correspondent is A.

Thus, a total of seven messages are required. However, the initial four messages need be used only infrequently because both A and B can save the other‘s public key for future use, a technique known as caching.

Disadvantages:

- Bottle neck at the authority.

(d) Public-Key Certificates

The scenario of Figure 2.35 is attractive, yet it has some **drawbacks.** The public-key authority could be somewhat of a bottleneck in the system, for a user must appeal to the authority for a public key for every other user that it wishes to contact. As before, the directoryof names and public keys maintained by the authority is vulnerable to tampering.

An alternative approach is to use **certificates** that can be used by participants to exchange keys without contacting a public-key authority.

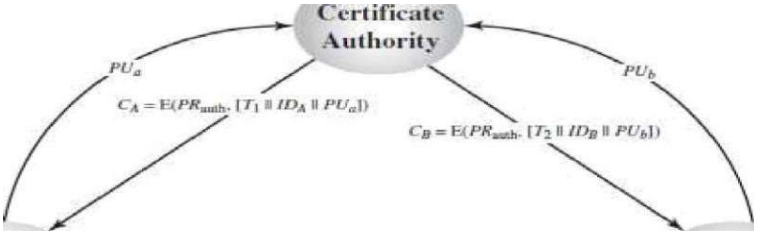
A certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.

A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate. Anyone needed this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature.

1. Any participant can read a certificate to determine the name and public key of the certificate‘s owner.

2. Any participant can verify that the certificate originated from the certificate authority and is notcounterfeit.
3. Only the certificate authority can create and update certificates.
These requirements are satisfied by the original proposal in. Denning added the followingadditional requirement:
4. Any participant can verify the currency of the certificate.

A certificate scheme is illustrated in Figure. Each participant applies to the certificate authority, supplying a public key and requesting a certificate.



(a) Obtaining certificates from CA

Public-Key distribution of Secret Keys using public key cryptography:

- Use previous methods to obtain public-key
- Can use for secrecy or authentication
- Public-key algorithms are slow so usually want to use private-key encryption to protectmessage contents, Hence need a session key

a) Simple
b) Secret key distribution with confidentiality and authentication
c) Hybrid
d) Diffie Hell man key exchange

(a) SimpIe Secret Key Distribution:

1. A generates a public/private key pair (KUa,KRa) and transmits a message to B consisting ofKUa and an identifier of A,IDA.
2. B generatesa secret key , and transmits it to A, encrypted with A‘s public key.
3. A computes DKRa [EKUa [s]] to recover the secret key. Because only A can decrypt themessage, only A and B will know the identity of Ks -
4. A discards KUaand KRa and B discards KUa•

Advantages:

- No keys exist before the start of the communication no key exist after the completion ofcommunication
- Secure from eaves dropping

Disadvatages:

- Replay attack
- Meet in the middle attack
- A generates a public/private key pair *{PUa, PRaj* and transmits a message intended forB consisting of PL/aand an identifier of A, *IDA.*
- D intercepts the message, creates its own public/private key pair *\PUd, PRO* andtransmits *PUs* 0 0 *IDA* to B.
- B generates a secret key, *Ks,* and transmits E(PL/s, *Ks).*
- D intercepts the message and learns *Ks* by computing D(PRd, E(PL/d, *Ks)).*
- D transmits E(PL/a, *Ks)* to A.

(b) Secret Key Distribution with Confidentiality and Authentication:

1. A uses B‘s public key to encrypt a message to B containing an identifier of A (ID A) anda nonce (N1),which is used to identify this transaction uniquely.
2. B sends a message to A encrypted with KUa and containing A‘s decrypted message (1) ,the presence of N1 in message (2) assures A that correspondent is B.
3. A returns Ne. encrypted using B‘s public key, to assurer B that its correspondent is A.

4. A select a secret key s find sends M = EKUb[EKRa[Ks]] TO B. Encryption of this message with B‘s public key ensures that only B can read it.;encryption with A‘s private key ensures that only A could have sent it.
5. Computes DxUa[DKRb[M]]to recover thesecret key.

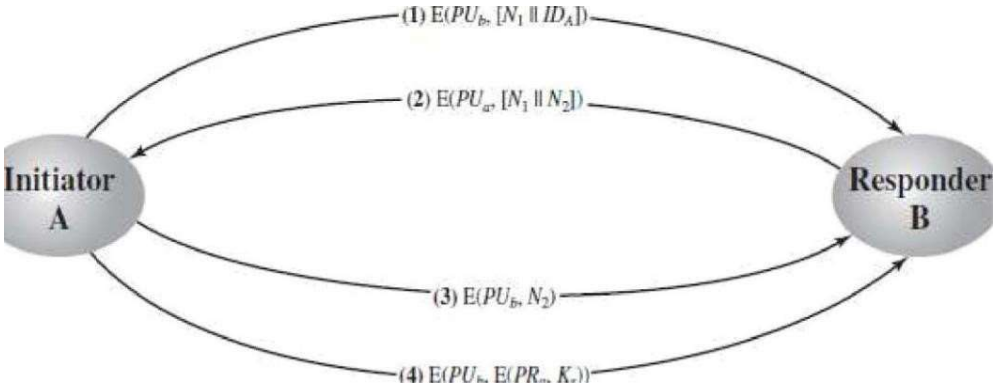


Figure . Public Key Distribution of secret Keys

Advantages:

Scheme ensures both confidentiality and authentication in the exchange of a secret key.

(c)A Hybrid Scheme

Public-key scheme isused to distribute the master keys. The following rationale is providedfor using thisthree-level approach:

1. Performance:

The public key encryption is used occasionally to update the master key between usesand KDC

When the distribution of session keys is done by public key encryption the performancedegrades because of high computation needed by P.K.E.

2. **Backward compatibility:** The hybrid scheme is easily overlaid on an existingKDCscheme with minimal disruption or software changes.

The addition of a public-key layer provides a secure, efficient means of distributing master keys.

DIFFIE HELLMAN KEY EXCHANGE

The purpose of the algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages. The Diffie-Hellman algorithm depends forits effectiveness on the difficulty of computing discrete logarithms.

First, we define a primitive root of a prime number p as one whose power generate all the integers from 1 to (p-1) i.e., if _a‘ is a primitive root of a prime number p, then the numbers

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

are distinct and consists of integers from 1 to (p-1) in some permutation.

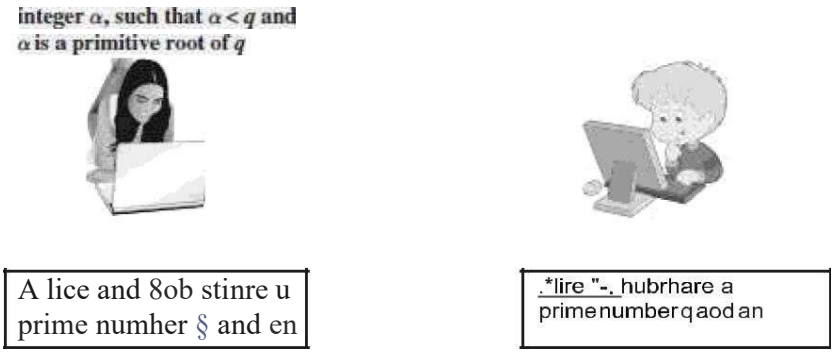
For any integer _b‘ and a primitive root _a‘ of a prime number _p‘, we can find a unique exponent _i‘ such that

$$b = a^i \bmod p \text{ where } 0 \le i \le (p-1)$$

The exponent _i‘ is referred to as discrete logarithm.

The Algorithm

Figure 2.37 summarizes the Diffie-Hellman key exchange algorithm.There are publicly known numbers: a prime number _q‘ and an integer a that is primitive root of q. suppose users A and B wish to exchange a key. User A selects a random integer XA‹ q and computes YA = a XA mod q.



integer o. szsch that a < g' azid

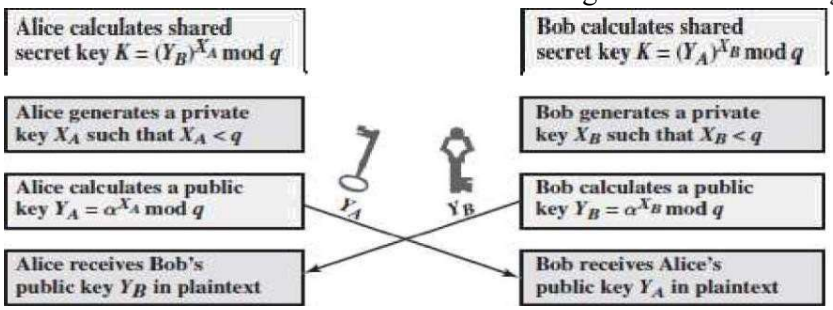


Fig.Diffie Hellman Key Exchange

Similarly, user B independently selects a random integer XB< q and computes YB a XB mod q. Each side keeps the X value private and makes the Y value available publicly to the other side.

User A computes the key as

$$K = (YB) \bmod q \text{ and}$$

User B computes the key as

$$K = (YA)^{XB} \bmod q$$

These two calculations produce identical results.

$$K = (YB) \bmod q$$

$$= (a^{XB} \bmod q) \bmod q$$

$$= (a^{XB})^{mod\ q}$$

$$\text{—} (a)^{mod\ q}$$

$$= (a^{X_B} \bmod q) \bmod q$$

= (Y A) mod q

Global Public Elements
q prime number α a<q and a is the primitive root of q
User A Key generation
Select XA Xy< q Calculate public Yr YA mod q
User B Key generation
Select XB XB‹ q Calculate public YB YB= ^{XB} mod q
Generation of secret key by User A
K = (YB) mod q
Generation of secret key by User B
K = (YA)^{XB} mod q

The result is that two sides have exchanged a secret key. The security of the algorithm lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms.

Key Exchange Protocols

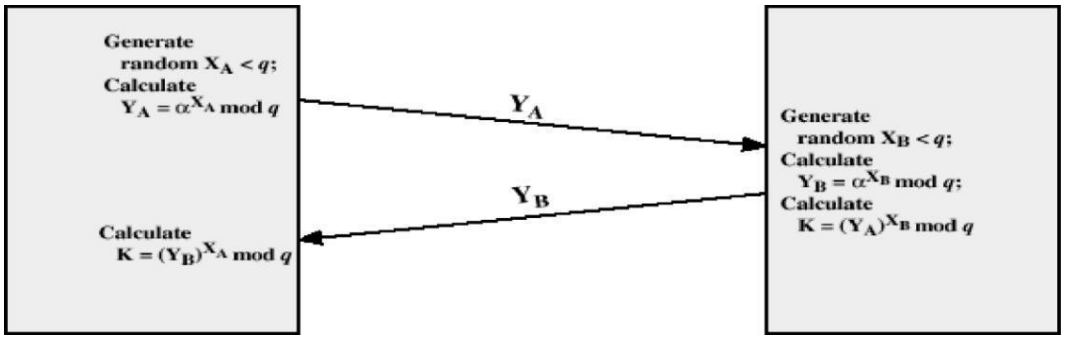


Figure .Diffe-Hellman Key Exchange

The protocol depicted in figure 2.38 is insecure against a man-in-the-middle attack. Suppose Aliceand Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows:

1. Darth prepares for the attack by generating two random private keys XD1 and XD2 and thencomputing the corresponding public keys YD1 and
Yo2-2. Alice transmits YA to Bob.
3. Darth intercepts YA and transmits YD1 to Bob. Darth also calculates = (YA) ^{X}D mod q.
4. Bob receives YD1 and calculates K1 = (YD1) ^{X} B mod q.
5. Bob transmits XA to Alice.
6. Darth intercepts XA and transmits YD2 to Alice. Darth calculates 1 " (YB) ^{X}D1 mod q.
7. Alice receives Yo2 and calculates 2' (YD2) ^{X} mod q.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K1 and Alice and Darth share secret key All future communication between Bob and Alice is compromised in the following way:

1. Alice sends an encrypted message M: E(2. M).

2. Darth intercepts the encrypted message and decrypts it, to recover M.
3. Darth sends Bob E(K1. M) or E(K1. M‘), where M‘ is any message

Example:

Key exchange is based on the use of the prime number *q* —— 353 and a primitive root of 353, inthis case a = 3. A and B select secret keys *EA*—— 97 and *XB*—— 233, respectively.

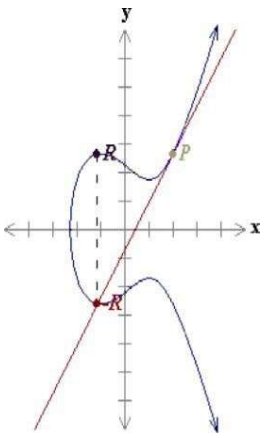
Each computes its public key:
A computes *YA* 3^{97} mod 353 = 40.
B computes *YB*—— 3^{233} mod 353 = 248.
After they exchange public keys, each can compute the common secret key:A computes *K* —— *(Yy)* mod 353 = 248 mod 353 = 160.
B computes *K* —— (Y A) mod 353 = 40 ^{2,3} mod 353 = 160.

ELLIPTIC CURVE ARITHMETIC

Elliptic Curves:
An elliptic Curve is a Cubic equation of the form

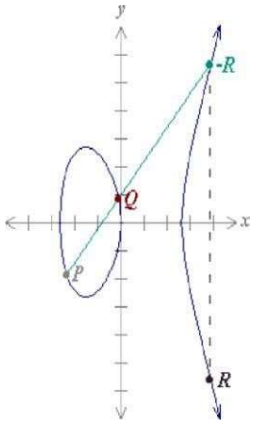
$$Y^2 + axy + by = x^3 + cx^2 + dx + e$$

Where a,b,c,d and e are real numbers



-E(1.11,-2.64)
(.11,2.64)

IP - R - (-1.11, 2.64).



/'(-J.35,-1.86)
g (-0.1, 0.836) -
/t (3.89, 5.62)
J2 (3.89, -5.62)
/' + g =/t = (3.89, -5.62).

$y^2 = x^3 - 3x + 5$

$y^2 = x^3 - 7x$

A special addition operation is defined over elliptic curves and with the inclusion of a point —0" called point at infinity.

If three points are on a line intersecting an elliptic curve, then their sum is equal to thispoint at infinity O (which acts as the identity element for this addition operation)

Elliptic Curves over Galois field:

An elliptic group over the Galois Field Ed(a,b) is obtained by computing x^3+ax+b mod p for 0ñ xñ p. The constants a&b are non-negative integers smaller than the prime number p must satisfy the condition.

$$4a^3 + 27b^2 \bmod p \; 1 \; 0$$

For each value of x, one needs to determine whether or not it is a quadratic residue. If not then the point is not in the elliptic group Ed (a,b)

Addition and multiplication operation over elliptic groups:

Let the points P = (x1. y1) and Q = (X ,Y2) be in the elliptic group Ed (a,b) and O be the point atinfinity.

The rules for addition over the elliptic group Ed (a,b) are:

1. P+O=O+P=P

2. If x2 = x1 and y2 = -y1, that is P = (x1,y1) and Q = (X2,Y2)' (x1.-y1)' -P Then P+Q = O

3. If Q I -P ,then their sum P+Q =(x3.y3) is given by

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

and

$$\lambda = \begin{cases} \frac{y2 \text{ —+ } j \quad p, \quad ,;;}{x_2 - x_1} \\ \frac{3x + a}{2J,} \quad if \quad P = Q \end{cases}$$

Elliptic Curve Encryption:

Elliptic curve cryptography can be used to encrypt the plain text message M, into ciphertext. The plain text message M is encoded into a point PM from the finite set of points in the elliptic group, Ed(a,b).

The first step consists in choosing a generator point, G c Ed(a, b) , such that the smallest valueof n for which nG = o is a very large prime number.

The elliptic group Ed(a,b) and the generator point G are made public.

Each user select a private key,nA< n and compute the public key PA as PA = nAG

To encrypt the message point PM for Bob (B),
Alice (A) choses a random integer k and compute the ciphertext pair of points c

Using Bob‘s public key PB

Pc' [(KG),(PM B)]

After receiving the ciphertext pair of points, Pc. Bob multiplies the firstpoint,(KG) with his privatekey nB and then adds the result to the second point in the ciphertext pair of points (PM + KPB)

(PM + B) — [nB(KG)]= (PM+KnBG)-[nB(KG)]=PM

which is the plaintext point, corresponding to the plaintext message M.

Only Bob knowing the private key nB. can remove nB(KG) from the second point of the ciphertextpair of point, i.e (PM + KGB), and hence retrieve the plaintext information PM

Elliptic curve cryptography

Security of ECC:

1. The cryptographic strength of elliptic curve encryption lies in the difficulty for a crypt analystto determine the secret random number k from KP & P itself.

2. The fastest method to solve this problem (known as elliptic curve logarithm problem is thepollard factorization method).

3. The computational complexity for breaking the elliptic curve cryptosystem, using the pollardmethod is 3.3×10^{10} MIPS years for an elliptic curve key size of only 150bits.

4. For comparison the fastest method to break RSA, using General Number Field Sieve methodto factor the composite integer n in to the two prime p & q requires $2x10^{11}$ MIPS years for a 768 bit RSA key & $3x\ 10^{11}$ MIPS years for a RSA key length 1024

5. If the RSA key length is increased to 2048 bits, the GNES method will need $3x10^{20}$ MIPS years to factor n whereas increasing the elliptic curve key length to only 24 bits will impose a computational complexity of $1.6x10^{2}$ MIPS years.

Analog of Diffie-Hellman Key Exchange:

Key exchange using elliptic curves can be done in the following manner.

First pick a large integer q, which is either a prime number p or

an integer of the form2‖ and elliptic curve parameters a and b. This defines the elliptic group of points Ed(a, b).

Next, pick a base point G = (x1. y1) in Ed(a, b) whose order is a very large value n. The order n of a point G on an elliptic curve is the smallest positive integer n such that nG = O. Eq(a,b) and G are parameters of the cryptosystem known to all participants.

1. A selects an integer nA less than n. This is A's private key. A then generates a public key PAnA x G; the public key is a point in Ed(a, b)•
2. B similarly selects a private key nB and computes a public key PB

3. A generates the secret key K = nAx PB B generates the secret key K = nB x EA.

Global Public elements	
Ed(a,b) an	Elliptic curve with parameters a,b and q, where q is a prime or integer of the form 2‖
G	point on elliptic curve whose order is large value n
User A Key Generation	
Select private	n A nA<n
Calculate public PA	PA nA X G
User B Key Generation	
Select private	nB nB‹n
Calculate public PA	PB = n B X G
Calculation of secret key by User A	
K = nAX PB	
Calculation of secret key by User B	
K = nBX PA	

Figure .ECC Diffie-Hellman Key Exchange