

Private-Key Cryptography

- ⌘ traditional **private/secret/single key** cryptography uses **one** key
- ⌘ shared by both sender and receiver
- ⌘ if this key is disclosed communications are compromised
- ⌘ also is **symmetric**, parties are equal
- ⌘ hence does not protect sender from receiver forging a message & claiming is sent by sender

Public-Key Cryptography

- ⌘ probably most significant advance in the 3000 year history of cryptography
- ⌘ uses **two** keys – a public & a private key
- ⌘ **asymmetric** since parties are **not** equal
- ⌘ uses clever application of number theoretic concepts to function
- ⌘ complements **rather than** replaces private key cryptosystem

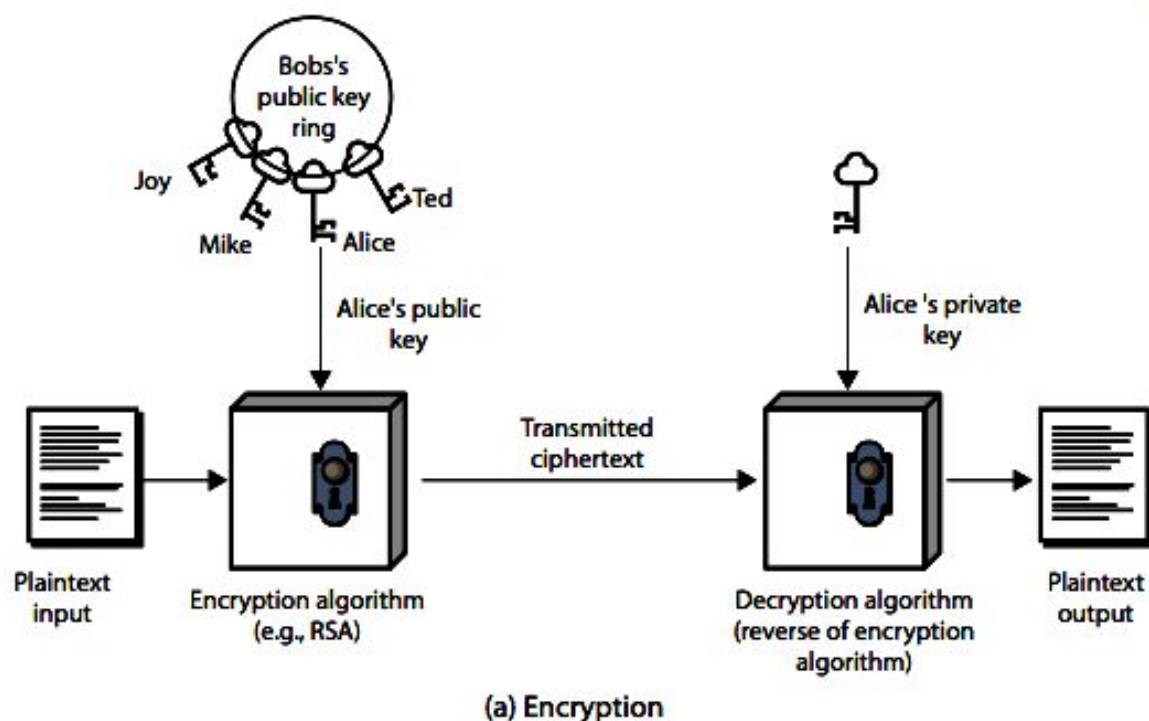
Why Public-Key Cryptography?

- ⌘ developed to address two key issues:
 - ⌘ **key distribution** – how to have secure communications in general without having to trust a KDC with your key
 - ⌘ **digital signatures** – how to verify a message comes intact from the claimed sender
- ⌘ public invention due to Whitfield Diffie & Martin Hellman at Stanford University in 1976/1977
 - ⌘ known earlier in classified community

Public-Key Cryptography

- ⌘ **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
 - ⌘ a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
 - ⌘ a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign (create) signatures**
- ⌘ is **asymmetric** because
 - ⌘ those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

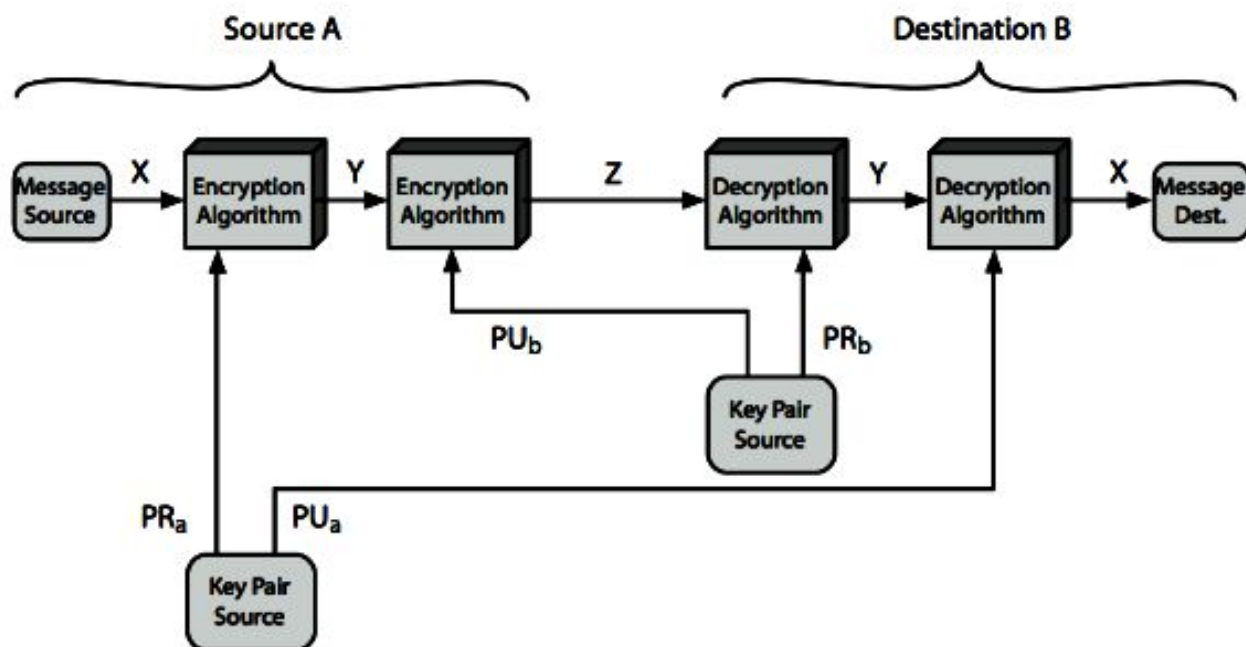
Public-Key Cryptography



Public-Key Characteristics

- ❑ Public-Key algorithms rely on two keys where:
 - ❑ it is computationally infeasible to find decryption key knowing only algorithm & encryption key
 - ❑ it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
 - ❑ either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

Public-Key Cryptosystems



Public-Key Applications

- ❑ can classify uses into 3 categories:
 - ❑ **encryption/decryption** (provide secrecy)
 - ❑ **digital signatures** (provide authentication)
 - ❑ **key exchange** (of session keys)
- ❑ some algorithms are suitable for all uses, others are specific to one

- ⊠ can classify uses into 3 categories:
 - ⊠ **encryption/decryption** (provide secrecy)
 - ⊠ **digital signatures** (provide authentication)
 - ⊠ **key exchange** (of session keys)
- ⊠ some algorithms are suitable for all uses, others are specific to one

Security of Public Key Schemes

- ⊠ like private key schemes brute force **exhaustive search** attack is always theoretically possible
- ⊠ but keys used are too large (>512bits)
- ⊠ security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalyse) problems
- ⊠ more generally the **hard** problem is known, but is made hard enough to be impractical to break
- ⊠ requires the use of **very large numbers**
- ⊠ hence is **slow** compared to private key schemes

Diffie-Hellman Key Exchange/Agreement Algorithm

- ❑ Diffie Hellman (DH) key exchange algorithm is a method for securely exchanging cryptographic keys over a public communications channel.
- ❑ Keys are not actually exchanged – they are jointly derived. It is named after their inventors Whitfield Diffie and Martin Hellman (1976)
- ❑ The beauty of this algorithm is that two parties, who wants to communicate securely can agree on a symmetric key.

Algorithm Steps

- ❑ Firstly, Ram and Shyam agree on two large prime number, n & g . These two integer need not to be kept secret. Ram and shyam can use an insecure channel to agree on them.
- ❑ Ram chooses another large random number x , and Calculate A such that
$$A = g^x \text{ mod } n.$$
- ❑ Ram sends the number to Shyam.
- ❑ Shyam independently chooses another large random integer and calculate B such that
$$B = g^y \text{ mod } n.$$
- ❑ Shyam sends the number B to Ram.
- ❑ **A now compute the secret key $K1$:**
$$K1 = B^x \text{ mod } n$$
- ❑ **B now Compute the secret key $K2$:**
$$K2 = A^y \text{ mod } n$$

Example

- Let $n=11$ and $g=7$

- Compute $A = g^x \bmod N$

Assume $x=3$. Then,

$$A = 7^3 \bmod 11 = 343 \bmod 11 = 2$$

- Ram Sends 2 to Shyam.

- Compute $B = g^y \bmod n$

- Assume $y=6$. Then,

$$B = 7^6 \bmod 11 = 117649 \bmod 11 = 4$$

- Compute Key $K1 = B^x \bmod n$

$$K1 = 4^3 \bmod 11 = 64 \bmod 11 = 9$$

- Compute Key $k2 = A^y \bmod n$

$$K2 = 2^6 \bmod 11 = 64 \bmod 11 = 9$$

Advantages

- ⊠ The sender and receiver don't need any prior knowledge of each other.
- ⊠ Once the keys are exchanged, the communication of data can be done through an insecure channel.
- ⊠ The sharing of the secret key is safe.

14

Disadvantages

- ⊠ The algorithm can not be used for any asymmetric key exchange.
- ⊠ Similarly, can not be used for signing digital signatures.
- ⊠ Since it doesn't authenticate any party in the transmission, the Diffie-Hellman key exchange is susceptible to a man-in-the-middle attack.

The RSA Algorithm

- ⊠ RSA Algorithm Based on the idea that factorization of integers into their prime factors is hard.

- ★ Compute $n = p \cdot q$, where p and q are distinct prime numbers.

- ⊠ RSA Algorithm Proposed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978

- ⊠ .

- ⊠ RSA algorithm is an asymmetric cryptography algorithm which means, there should be two keys involve while communicating, i.e., public key and private key.

- ⊠ Public Key same for all users in the Network.

- ⊠ Private key is the separate key or secret key for decryption.

RSA Algorithm

- ☒ Chooses two primes p, q and compute $n = p \cdot q$
- ☒ Compute $\phi(n) = (p-1)(q-1)$
- ☒ Chooses e with $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$, (e is prime)
- ☒ Solves $d \times e \bmod \phi(n) = 1$
- ☒ Public Key (e, n) , Private Key (d, n)
- ☒ **Encrypt Message M as $C = M^e \bmod n$**
- ☒ **Decrypt Message $M = C^d \bmod n$**

18

RSA Example

- ☒ Choose Two Prime Number $p=3$ and $q=11$
- ☒ Compute $n = p \times q = 3 \times 11 = 33$
- ☒ Compute $\phi(n) = (p-1) \times (q-1) = 2 \times 10 = 20$
- ☒ Choose e that is 7

Because e is not part of $\phi(n) = 20 = 5 \times 2 \times 2$

e should not multiply by 5 and 2 and should not divide by 20.

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

$$d \times 7 \bmod 20 = 1$$

Choose $d=3$

$$3 \times 7 \bmod 20 = 1$$

OR

$$d = 1 + k(\phi(n))/e$$

$$= 1 + 1 \times 20/7 = 3$$

Where $K=0$ to $\phi(n)$

Condition is satisfied.

Public Key = (7,33)

Private Key = (3,33)

☒ Encrypt Message M as $C = M^e \bmod n$

Assume $M=31$ ($M < n$)

$$C = 31^7 \bmod 33 = 4$$

☒ Decrypt Message $M = C^d \bmod n$

$$M = 4^3 \bmod 33 = 31$$

For encryption and decryption process user uses the two different key.

Advantages of RSA

RSA algorithm is hard to crack since it involves factorization of prime numbers which are difficult to factorize.

Moreover, RSA algorithm uses the public key to encrypt data and the key is known to everyone, therefore, it is easy to share the public key.

Disadvantages of RSA

RSA algorithm can be very slow in cases where large data needs to be encrypted by the same computer. It requires a third party to verify the reliability of public keys. Data transferred through RSA algorithm could be compromised through middlemen who might temper with the public key system.

In conclusion, both the symmetric encryption technique and the asymmetric encryption technique are important in encryption of sensitive data.

Rabin Cryptosystem

- ✘ **Rabin Cryptosystem** is an public-key cryptosystem invented by Michael Rabin.
- ✘ Published in 1979.
- ✘ It uses asymmetric key encryption for communicating between two parties and encrypting the message.
- ✘ The security of Rabin cryptosystem is related to the difficulty of factorization.

Algorithm

Key generation

- ✘ Generate two very large prime numbers, p and q , which satisfies the condition

$$p \neq q \rightarrow p \equiv q \equiv 3 \pmod{4}$$

For example:

$$p=139 \text{ and } q=191$$

- ✘ Calculate the value of n

$$n = p \cdot q$$

Publish n as public key and save p and q as private key

Encryption

- ☒ Get the public key n .
- ☒ Convert the message to ASCII value. Then convert it to binary and extend the binary value with itself, and change the binary value back to decimal m .
- ☒ Encrypt with the formula:
$$C = m^2 \bmod n$$
- ☒ Send C to recipient.

Decryption

- ☒ Accept C from sender.
- ☒ Specify a and b with Extended Euclidean GCD such that, $a.p + b.q = 1$
- ☒ Compute r and s using following formula:
$$r = C^{(p+1)/4} \bmod p$$
$$s = C^{(q+1)/4} \bmod q$$
- ☒ Now, calculate $M1, M2, M3$ & $M4$ using following formula:
$$M1 = (a.p.s + b.q.r) \bmod n$$
$$M2 = n - M1$$
$$M3 = (a.p.s - b.q.r) \bmod n$$
$$M4 = n - M3$$
- ☒ Determine in which the left and right half are same. Keep that binary's one half and convert it to decimal m . Get the ASCII character for the decimal value m . The resultant character gives the correct message sent by sender.

Example of Rabin Cryptosystem

Key Generation:

- ☒ Taking two prime numbers
- ☒ $p=7$ & $q=11$
- ☒ $n=7*11=77$
- ☒ p & q are private key and n is the public key.

Encryption:

- ☒ Assume value of $M=20$
- ☒ Then $C = M^2 \bmod n$
- ☒ $= 20*20 \bmod 77 = 15$

Decryption:

- ☒ Specify a and b with Extended Euclidean GCD such that, $a.p + b.q = 1$

Choose $a=-3, b=2$

$$-3*7+2*11=1$$

- ☒ Compute r and s using following formula:

$$r = C^{(p+1)/4} \bmod p$$

$$r = 15^2 \bmod 7 = 1$$

$$s = C^{(q+1)/4} \bmod q$$

$$s = 15^3 \bmod 11 = 9$$

- ☒ Now, calculate m_1, m_2, m_3, m_4 using following formula:

$$m_1 = (a.p.s + b.q.r) \bmod n$$

$$m_1 = 64$$

$$m_2 = n - m_1 = 13$$

$$m_3 = (a.p.r - b.q.s) \bmod n$$

$$m_3 = 20$$

$$m_4 = n - m_3 = 57$$

So Exact Value is m_3 .

Advantage & Disadvantage

- ✘ It has the advantage over the others that the problem on which it banks has proved to be hard as **integer factorization**.
- ✘ It has the disadvantage also, that each output of the Rabin function can be generated by any of four possible inputs. if each output is a ciphertext, extra complexity is required on decryption to identify which of the four possible inputs was the true plaintext.

ElGamal Cryptosystem

- ✘ The **ElGamal encryption system** is an asymmetric key encryption for public-key cryptography which is based on the Diffie–Hellman Key Exchange.
- ✘ Described by Taher Elgamal in 1985.
- ✘ ElGamal encryption is used in the free GNU Privacy Guard software, recent versions of PGP, and other cryptosystems.
- ✘ The Digital Signature Algorithm (DSA) is a variant of the ElGamal signature scheme, which should not be confused with ElGamal encryption.
- ✘ ElGamal cryptosystem, called Elliptic Curve Variant, is based on the Discrete Logarithm Problem.

Algorithm With Example

Key Generation:

- ✘ Select Large Prime Number P (1st Part of Encryption key)

Assume $P=11$

- ✘ Select Decryption Key/Private Key D ($1 < D < P-1$)

Assume $D=3$

- ✘ Select 2nd Part of Encryption Key $E1$ ($1 < E1 < P-1$)

Assume $E1=2$

- ✘ Calculate 3rd part of Encryption Key $E2$

$E2 = E1^D \bmod P = 8$

- ✘ Public Key $(P, E1, E2) = (11, 2, 8)$, Private Key $D=3$

Encryption:

☒ Select any Random Integer R

Assume $R=4$

☒ $C1 = E1^R \bmod P$

$$C1 = 2^4 \bmod 11 = 5$$

☒ $C2 = (PT \times E2^R) \bmod P$

$$C2 = (7 \times 8^4) \bmod 11 = 28672 \bmod 11 = 6$$

(Assume $PT=7$)

☒ $CT = (C1, C2) = (5, 6)$

Where PT =Plain Text, CT = Cipher Text

Decryption:

☒ $PT = (C2 \times (C1)^{-D}) \bmod P$

$$PT = 6 \times (5)^{-3} \bmod 11$$

$$PT = 6 \times 3 \bmod 11$$

$$PT = 7$$

Cipher text = $(C1, C2) = (5, 6)$

$$P = [C2 \times (C1^d)^{-1}] \bmod 11 = [6 \times (5^3)^{-1}] \bmod 11$$

$$(5^3)^{-1} \bmod 11 \\ = 125^{-1} \bmod 11 = 3$$

$$(ab \bmod n = a \bmod n \times b \bmod n)$$

$$\rightarrow (125 \times x) \bmod 11 = 1 \\ x = 3$$

$$P = (6 \times 3) \bmod 11$$

ElGamal Analysis

- ✘ In ElGamal system, each user has a private key D and has **three components** of public key – **prime modulus p , generator $E1$, and public $E2$** .
- ✘ The strength of the ElGamal is based on the difficulty of discrete logarithm problem.
- ✘ The secure key size is generally > 1024 bits. Today even 2048 bits long key are used.
- ✘ On the processing speed front, Elgamal is quite slow, it is used mainly for key authentication protocols.

RSA and ElGamal Schemes – A Comparison

RSA	ElGamal
It is more efficient for encryption.	It is more efficient for decryption.
It is less efficient for decryption.	It is more efficient for decryption.
For a particular security level, lengthy keys are required in RSA.	For the same level of security, very short keys are required.
It is widely accepted and used.	It is new and not very popular in market.