Catalog

ELGAMAL CRYPTOGRAPHIC SYSTEM

- In 1984, T. Elgamal announced a public-key scheme based on discrete logarithms, closely related to the Diffie-Hellman technique.
- The ElGamal cryptosystem is used in some form in a number of standards including the digital signature standard (DSS), and the S/MIME e-mail standard
- As with Diffie-Hellman, the global elements of ElGamal are a prime number q and α, which is a primitive root of q.

User A generates a private/public key pair as follows:

1. Generate a random integer X_A, such that $1<X_A<q-1$.

2. Compute $Y^A= \alpha^{XA} \bmod q$.

3. A's private key is X_A; A's pubic key is $\{q,\alpha,Y_A\}$.

Any user B that has access to A's public key can encrypt a message as follows:

1. Represent the message M as an integer in the range 0<=M<=q-1. Longer messages are sent as a sequence of blocks, with each block being an integer less than q.

2. Choose a random integer k such that 1<=k<=q-1.

3. Compute a one-time key $K= (Y_A)^k \bmod q$

4. Encrypt M as the pair of integers (C1,C2)where

$C1= \alpha^k \bmod q$; c2=KM mod q

User A recovers the plaintext as follows:

1. Recover the key by computing .

2. Compute $M=(C_2K^{-1}) \bmod q$

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{XA} \bmod q$
Public key	$PU = \{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{XA} \bmod q$
Plaintext:	$M = (C_2 K^{-1}) \bmod q$

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- Let us demonstrate why the ElGamal scheme works. First, we show how is recovered by the decryption process:

$K = (Y_A)^k \bmod q$ — K is defined during the encryption process
$K = (\alpha^{X_A} \bmod q)^k \bmod q$ — substitute using $Y_A = \alpha^{X_A} \bmod q$
$K = \alpha^{kX_A} \bmod q$ — by the rules of modular arithmetic
$K = (C_1)^{X_A} \bmod q$ — substitute using $C_1 = \alpha^k \bmod q$

Next, using K, we recover the plaintext as

$$C_2 = KM \bmod q$$
$$(C_2 K^{-1}) \bmod q = KMK^{-1} \bmod q = M \bmod q = M$$

1. Bob generates a random integer k.
2. Bob generates a one-time key K using Alice's public-key components YA, q, and k.

3. Bob encrypts k using the public-key component α, yielding C1, C2 provides sufficient information for Alice to recover K.

4. Bob encrypts the plaintext message using K.

5. Alice recovers K from C1 using her private key.

6. Alice uses $K^{-1}$to recover the plaintext message from C2.

Thus, K functions as a one-time key, used to encrypt and decrypt the message.

CONVENTIONAL AND PUBLIC-KEY ENCRYPTION

PUBLIC-KEY CRYPTOSYSTEM: SECRECY

There is some source A that produces a message in plaintext, X = [X1, X2, ...,XM]. The M elements of X are letters in some finite alphabet. The message is intended for destination B. B generates a related pair of keys: a public key, PUb, and a private key, PRb. PRb is known only to B, whereas PUb is publicly available and therefore accessible by A. With the message X and the encryption key PUb as input, A forms the ciphertext

Y = [Y1, Y2, ..., YN]:

Conventional Encryption	Public-Key Encryption
Needed to Work:	Needed to Work:
1. The same algorithm with the same key is used for encryption and decryption	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption
2. The sender and receiver must share the algorithm and the key.	2. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security:	Needed for Security:
1. The key must be kept secret.	1. One of the two keys must be kept secret.
2. It must be impossible or at least impractical to decipher a message if no other information is available.	2. It must be impossible or at least impractical to decipher a message if no other information is available.
3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key

Y = E(PUb, X)

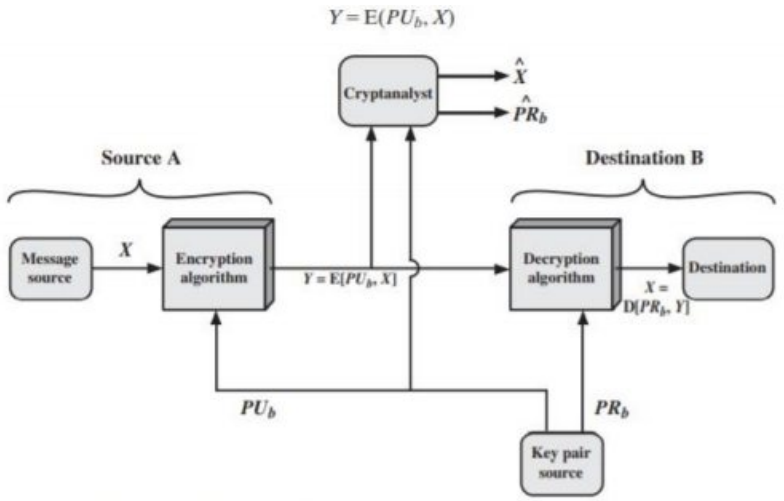

Figure 9.2 Public-Key Cryptosystem: Secrecy

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

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

X = D(PRb, Y) An adversary, observing Y and having access to PUb, but not having access to PRb or 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 X by generating a plaintext estimate Xˆ . Often, however, the adversary is interested in being able to read future messages as well, in which case an attempt is made to recover PRb by generating an estimate PRˆ b.

PUBLIC-KEY CRYPTOSYSTEM: AUTHENTICATION

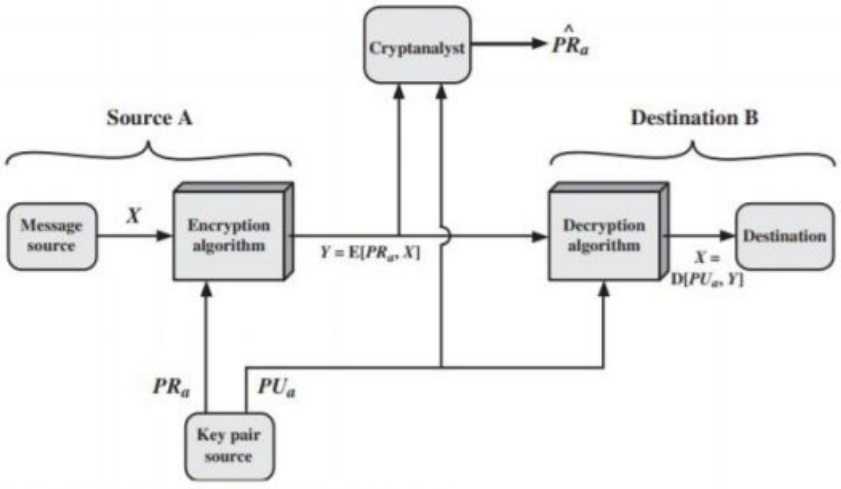

Figure 9.3 Public-Key Cryptosystem: Authentication

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

the scheme illustrated in the above Figure provides confidentiality to provide authentication:

Y = E(PRa, X) X = D(PUa, 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.

PUBLIC-KEY CRYPTOSYSTEM: AUTHENTICATION AND SECRECY

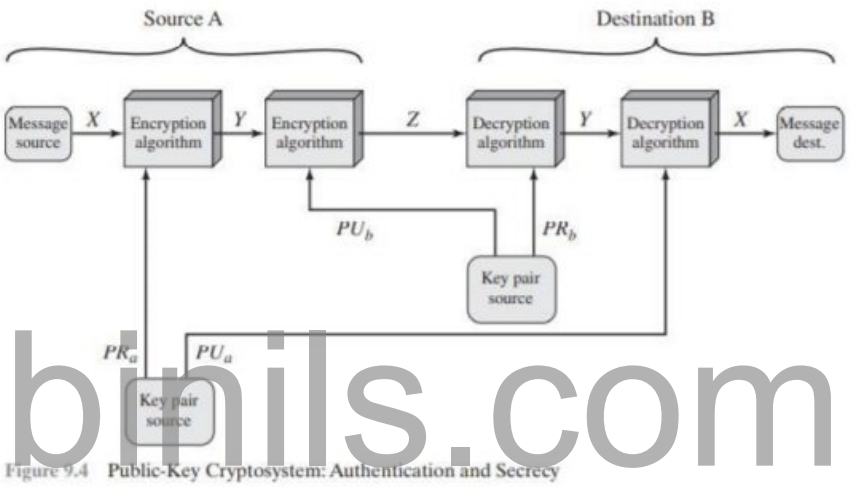

Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

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

Z = E(PUb, E(PRa, X)) X = D(PUa, D(PRb, Z))

In this case, we begin as before by encrypting a message, 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. The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.

APPLICATIONS FOR PUBLIC-KEY CRYPTOSYSTEMS

- We can classify the use of public-key cryptosystems into three categories
  - Encryption /decryption: The sender encrypts a message with the recipient's public key.

- Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
- 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.

APPLICATIONS FOR PUBLIC-KEY CRYPTOSYSTEMS

- Some algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications.
- Table indicates the applications supported by the algorithms.

Table 9.3 Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

REQUIREMENTS FOR PUBLIC-KEY CRYPTOGRAPHY

- 1. It is computationally easy for a party B to generate a pair (public key PUb, private key PRb).
- 2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext: C = E(PUb, M)
- 3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message: M = D(PRb, C) = D[PRb, E(PUb, M)]
- 4. It is computationally infeasible for an adversary, knowing the public key, PUb, to determine the private key, PRb.
- 5. It is computationally infeasible for an adversary, knowing the public key, PUb, and a ciphertext, C, to recover the original message, M. We can add a sixth requirement that, although useful, is not necessary for all public-key applications:
- 6. The two keys can be applied in either order: M = D[PUb, E(PRb, M)] = D[PRb, E(PUb, M)]

PUBLIC-KEY CRYPTANALYSIS

- A public-key encryption scheme is vulnerable to a brute-force attack. The countermeasure is the same: Use large keys

- Another form of attack is to find some way to compute the private key given the public key
- There is a form of attack that is peculiar to public-key systems. This is, in essence, a probable-message attack

THE RSA ALGORITHM

- Developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978.
- The RSA scheme is a block cipher in which the plaintext and ciphertext 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}.

DESCRIPTION OF THE ALGORITHM

- RSA makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n.
- That is, the block size must be less than or equal to $\log_2(n) + 1$; in practice, the block size is i bits, where $2^i\ n \leq 2^{i+1}$.
- Encryption and decryption are of the following form, 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.
- The sender knows the value of e, and only the receiver knows the value of d.
- Thus, this is a public-key encryption algorithm with a public key of PU = {e, n} and a private key of PR = {d, n}.

REQUIREMENTS

- 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 $M^{ed} \bmod n = M$ for all $M < n$.
- 2. It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$.
- 3. It is infeasible to determine d given e and n.
- Need to find a relationship of the form $M^{ed} \bmod n = M$
- The preceding relationship holds if e and d are multiplicative inverses modulo $\varphi(n)$, where $\varphi(n)$ is the Euler totient function.
- for p, q prime, $\varphi(pq) = (p - 1)(q - 1)$.
- The relationship between e and d can be expressed as $ed \bmod \varphi(n) = 1$

THE RSA ALGORITHM

Key Generation Alice

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

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

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

Figure 9.5 The RSA Algorithm

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006