Public key cryptosystem

Asymmetric Keys: Two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Public Key Certificate: A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the corresponding private key.

Public Key (Asymmetric) Cryptographic Algorithm: A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible. The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption. The first problem is that of **key distribution.** The second problem that Diffie pondered, and one that was apparently unrelated to the first, was that of **digital signatures.** If the use of cryptography was to become widespread, not just in military situations but for commercial and private purposes, then electronic messages and documents would need the equivalent of signatures used in paper documents.

Public-Key Cryptosystems

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the important characteristic.

• It is computationally infeasible to determine the decryption key given only knowledge of 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 used for decryption.

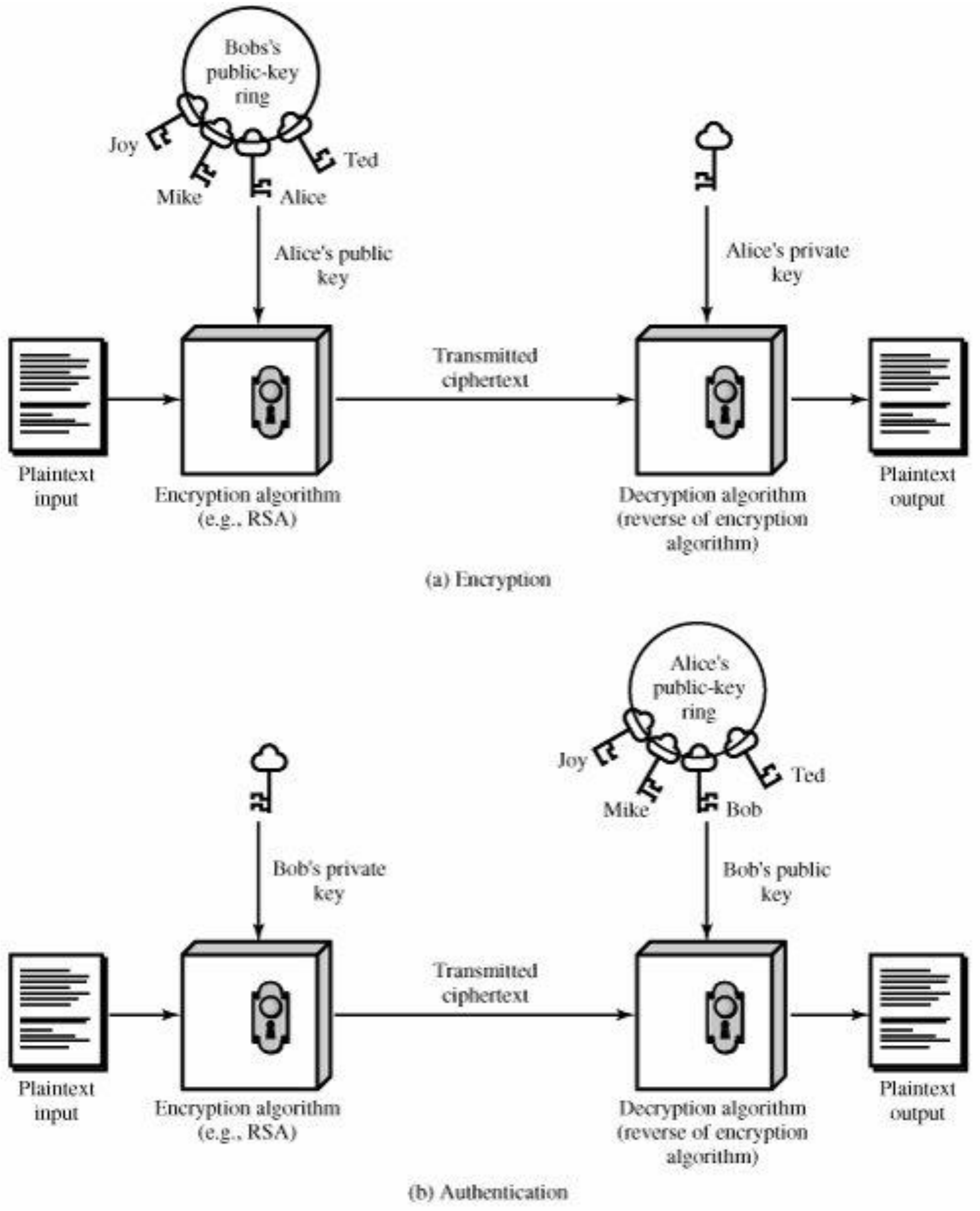

Figure (12) Public-Key Cryptography

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of 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. As Figure (12).a suggests, 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 message using Alice's public key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user's private key remains protected and secret, incoming communication is secure. At any time, a system can change its private key and publish the companion public key to replace its old public key. Table 4 summarizes some of the important aspects of symmetric and public-key encryption.

Table 4. Conventional and Public-Key Encryption

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. **2.** The sender and receiver must share the algorithm and the key.	**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 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. **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.	**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. **3.** Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

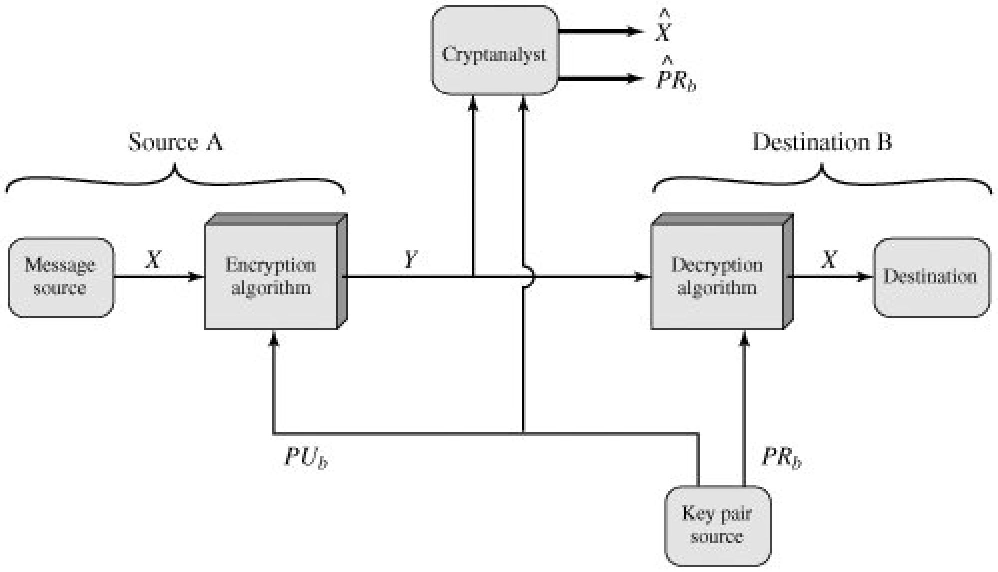

Figure (13) Public-Key Cryptosystem: Secrecy

With the message *X* and the encryption key *PUb* as input, A forms the ciphertext *Y* = [*Y*1, *Y*2,..., *YN*]: *Y* = E(*PUb*, *X*)

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 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.

We mentioned earlier that either of the two related keys can be used for encryption, with the other being used for decryption. This enables a rather different cryptographic scheme to be implemented. Whereas the scheme illustrated in Figure (13) provides confidentiality, Figures (12).b and 14 show the use of public-key encryption to provide authentication:

Y = E(*PRa*, *X*)

Y = E(*PUa*, *Y*)

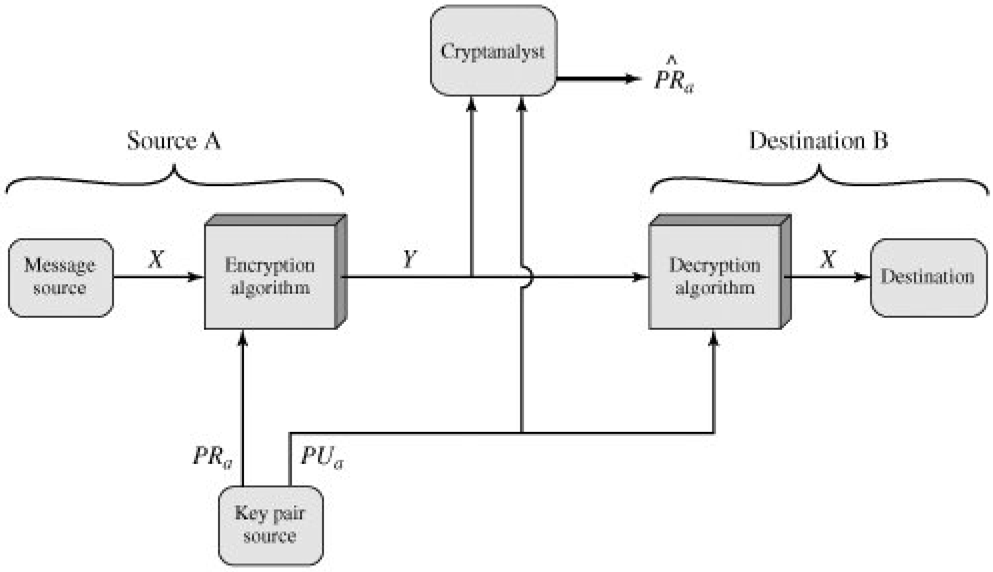

Figure (14) Public-Key Cryptosystem: Authentication

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.

In the preceding scheme, the entire message is encrypted, which, although validating both author and contents, requires a great deal of storage. Each document must be kept in plaintext to be used for practical purposes. A copy also must be stored in ciphertext so that the origin and contents can be verified in case of a dispute. A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document. Such a block, called an authenticator, must have the property that it is infeasible to change the document without changing the authenticator. If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing.

It is important to emphasize that the encryption process depicted in Figures (12).b and 14 does not provide confidentiality. That is, the message being sent is safe from alteration but not from eavesdropping. 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

Figure (14), there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key.

It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme (Figure (15)):

$Z = \mathrm{E}(PUb, \mathrm{E}(PRa, X))$
$X = \mathrm{D}(PUa, \mathrm{E}(PRb, Z))$

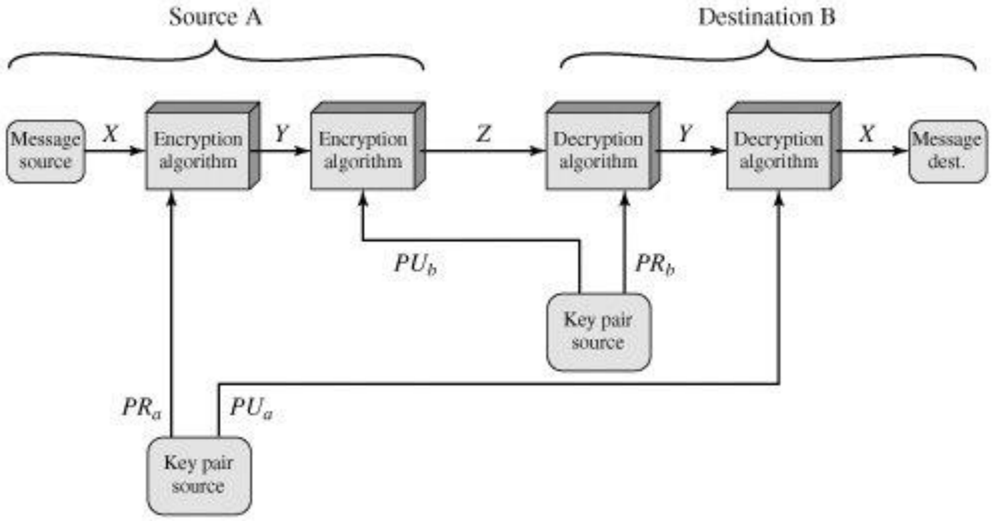

Figure (15) Public-Key Cryptosystem: Authentication and Secrecy

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

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.

Some algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications.

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:

$C = \mathrm{E}(PUb, M)$

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

$M = \mathrm{D}(PRb, C) = \mathrm{D}[PRb, \mathrm{E}(PUb, M)]$

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:

$M = \mathrm{D}[PUb, \mathrm{E}(PRb, M)] = \mathrm{D}[PRb, \mathrm{E}(PUb, M)]$

Before elaborating on why the requirements are so formidable, let us first recast them. The requirements boil down to the need for a trap-door one-way function. A *one-way function* is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy whereas the calculation of the inverse is infeasible:

$Y = \mathrm{f}(X)$ easy

$X = \mathrm{f}^{-1}(Y)$ infeasible