Asymmetric Cryptography

Learning Objective

After studying this chapter, you should be able to:

- Understanding the main basics, advantages and disadvantages of Asymmetric cryptography.
- Explain the main uses of public-key cryptosystems.
- Understanding the meaning of a trap door one way function and hard mathematical problem.
- Present an overview of the Diffie-Hellman and RSA algorithms.
- Understanding the meaning of digital certificate, and the PKI.

1.Introduction

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. As we have seen in the previous chapter, symmetric encryption requires the two communicants already share a key, which somehow has been distributed to them. The generation, distribution and storage of the keys are called key management. In open systems, where number of users is big, secret key cryptography does not scale well. For example, if a cryptosystem has *n* users, each user must have $(n-1)$ keys, and the total number of keys is $\frac{n^2-n}{2}$

If $n=1000$, each user must have (999) keys, and the total number of keys in the cryptosystem is 499500.

The second problem is the 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. That is, could a method be developed that would specify, to the satisfaction of all parties, that a digital message had been sent by a particular person?

Before proceeding, we should mention several common misconceptions concerning public-key encryption. One such misconception is that public-key encryption is more secure from cryptanalysis than is symmetric encryption. In fact, the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher. There is nothing in principle about either symmetric or public-key encryption that makes one superior to another from the point of view of resisting cryptanalysis.

A second misconception is that public-key encryption is a general-purpose technique that has made symmetric encryption obsolete. On the contrary, because of the computational overhead of current public-key encryption schemes, there seems no foreseeable likelihood that symmetric encryption will be abandoned.

2.Public Key Cryptography

Public key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman. For this reason, it is sometime called Diffie-Hellman encryption. It is also called asymmetric encryption because it uses two keys instead of one key (symmetric encryption). A public key known to everyone (widely distributed) A private or secret key known only to the recipient of the message.

When Bob wants to send a secure message to Alice, Bob uses Alice's public key to encrypt the message. Alice then uses her private key to decrypt it. Some people have compared public key cryptography to a mailbox. Many people can put mail into the mailbox (in effect, using the public key), but only a postal worker with the appropriate key (corresponding to the private key) can retrieve the mail from the mailbox.

The public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them.

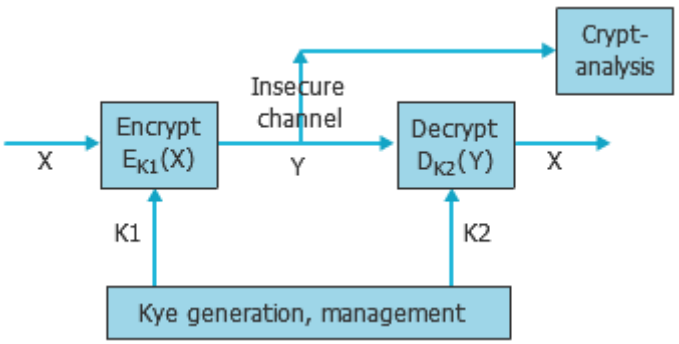
Figure(5.1) shows public key operations, where

X is the plaintext.

Y is the ciphertext.

Two keys K1, and K2. One is secret, other is public

The keys are related mathematically, but the private key cannot be practically derived from the public key. It is virtually impossible to deduce the private key if you know the public key.



Figure(5.1): Public Key Cryptography

Diffie and Hellman lay out the conditions that public key algorithms must fulfill:

1. It is computationally easy for a party B to generate a key pair (public key PU_b, and private key PR_b).
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:
   $C = E(PU_b, M)$.
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:
   $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$.
4. It is computationally infeasible for an adversary, knowing the public key, PU_b, to determine the private key, PR_b.
5. It is computationally infeasible for an adversary, knowing the public key, PU_b, and a ciphertext, C, to recover the original message, M.

3.Hybrid Encryption

Public key encryption and decryption algorithms tend to be incredibly slow relative to symmetric key algorithms. Public key algorithms tend to be about 1000 times slower than AES. In general, encrypting large messages using public key cryptography is not considered practical.

Hybrid encryption (or digital envelop), a mash up of symmetric and asymmetric encryption, provides an elegant solution that preserves the speed of symmetric encryption, while maintaining the security and exchange flexibility of Asymmetric encryption. We also do not end up increasing the length of the text message to be encrypted.

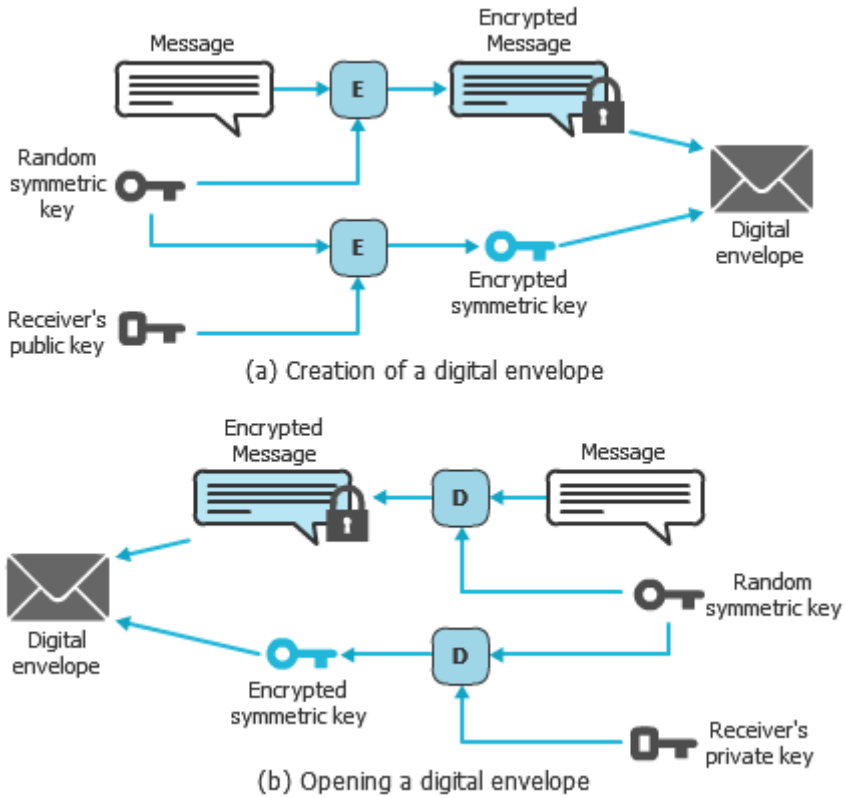
The idea of hybrid encryption is quite simple. Instead of using asymmetric encryption to encrypt the text, we use symmetric encryption to encrypt the message. Then, they maintain the secrecy of the key, and we encrypt the key using asymmetric encryption. The steps of hybrid encryption are figure(5.2):

- Generate a symmetric key. The symmetric key needs to be kept a secret.
- Encrypt the data using the secret symmetric key.

- The person to whom we wish to send a message will share her public key and keep the private key a secret.
- Encrypt the symmetric key using the public key of the receiver.
- Send the encrypted symmetric key to the receiver.
- Send the encrypted message text.
- The receiver decrypts the encrypted symmetric key using her private key and gets the symmetric key needed for decryption.
- The receiver uses the decrypted symmetric key to decrypt the message, getting the original message.

In the hybrid method, the data to be encrypted is not limited by the length of the encryption key. Additionally, the encrypted symmetric key is secure because it is encrypted using the public key of the receiver. Even if the encrypted data and the encrypted key are intercepted by another person, they will not be able to get the original data as long as the private key is maintained as a secret.

Hybrid encryption is used in all forms of internet communication between client and server these days. SSL/TLS, IPsec and SSH are good examples of Hybrid encryption.



Figure(5.2): Hybrid Encryption

4.Cryptography Hash Function

A hash function H is a transformation that takes an input m and returns a fixed size string. When employing in cryptography, the function H must have the following properties:

1. The input m can be of any length.
2. The output has a fixed length.
3. The function $H(m)$ is easy to compute for any given m.
4. $H(m)$ is a one way function i.e. it is computationally infeasible to find m such that $H(m) = h$ for any given h.
5. H is collision free. A weak collision free hash is a hash with the property: if given a message x, it is computationally infeasible to find y not equal to x such that $H(x) = H(y)$. A strong collision free hash is a hash with the property: it is computationally infeasible to find any two different messages x and y such that $H(x) = H(y)$.

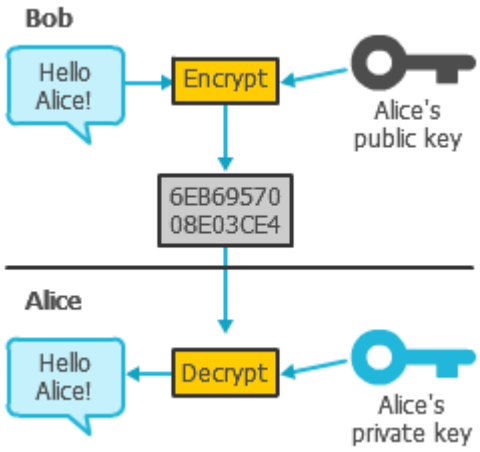
The hash value of a document is called a message digest or digital fingerprint of the document because it represents the original document. The main role of a cryptography hash function is to check the integrity of the document. An integrity checksum is a value that is computed from the data that is being protected. The integrity checksum is stored separately from the protected data. The recipient of the data re-computes the checksum from the data that is received and compares that checksum to the value that was recorded separately by the provider of the data. If the original checksum and the recomputed checksum do not match, then the data has been changed in some way. The other use of the message digest is to calculate the digital signature. The most used hash functions today is the secure hash algorithm (SHA) with different versions (SHA-1, SHA-2, and SHA-3) with different output (160, 224, 256, 384, and 512) bits long.

5.Applications for Public-Key Cryptosystems

Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly. Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function. In broad terms, we can classify the use of public-key cryptosystems into three categories:

Encryption/Decryption Figure(5.3):

Consider Alice whose private key and public key are P_A and U_A respectively. Since U_A is public key all users will be able to get it. For Bob that needs to send the message 'Msg' in a secured way to Alice, he will encrypt the data using Alice's public key to obtain the cipher text 'Ctx'. The encrypted message, cipher text, can only be decrypted using Alice's private key. On receiving the message Alice decrypts it using her private key P_A. Since only Alice knows her private key P_A none other can decrypt the message. It is important that Bob receives the correct public key from Alice, i.e. no middleman must tamper or change the public key to its public key. Digital Certificate helps to deliver the public key in authenticated method. Digital Certificate is explained in the next sections. One of the popular public key encryption algorithms is RSA. RSA encryption is explained in the next sections.



Figure(5.3): Public Key (Encryption)

Digital Signature Figure(5.4):

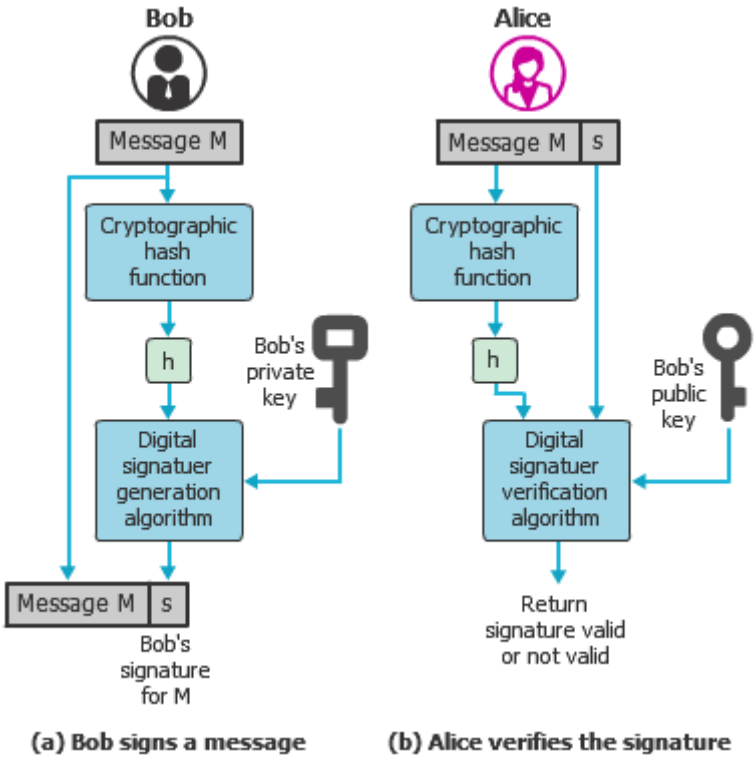
The most important development from the work on public-key cryptography is the digital signature. The digital signature provides a set of security capabilities that would be difficult to implement in any other way.

The main goals of digital signatures are:

- A signature should be proof of authenticity. Its existence on a document should be able to convince people that the person whose signature appears on the document signed the document.
- A signature should be impossible to forge. The person who signed the document should not be able to claim that the signature is not theirs (support for non-repudiation).
- After the document is signed, it should be impossible to alter the document without detection (support for data integrity). The signature is intrinsically linked to the document that is being signed.
- It should be impossible to transplant the signature to another document (the signature is not reusable). Again, the digital signature is intrinsically linked to the document that is being signed.
- It is important to emphasize that a digital signature alone does not provide confidentiality since it does not prevent disclosure of information. If confidentiality is required, we should add other mean such as encryption.

Figure(5.4) is a generic model of the process of constructing and using digital signatures. All of the digital signature schemes have this structure. Suppose that Bob wants to send a message to Alice. Although it is not important that the message be kept secret, he wants Alice to be certain that the message is indeed from him. For this purpose, Bob uses a secure hash function, such as SHA-512, to generate a hash value for the message. That hash value, together with Bob's private key serves as input to a digital signature generation algorithm, which produces a short block that functions as a digital signature. Bob sends the message with the signature attached. When Alice receives the message plus signature, she (1) calculates a hash value for the message, (2) provides the hash value and Bob's public key as inputs to a digital signature verification algorithm. If the algorithm returns the result that the signature is valid, Alice is assured that the message must have been signed by Bob. No one

else has Bob's private key and therefore no one else could have created a signature that could be verified for this message with Bob's public key. In addition, it is impossible to alter the message without access to Bob's private key, so the message is authenticated both in terms of source and in terms of data integrity. Examples of Digital Signature algorithms are RSA, digital signature Algorithm (DSA) and Elliptic curve digital signature algorithm (ECDSA).

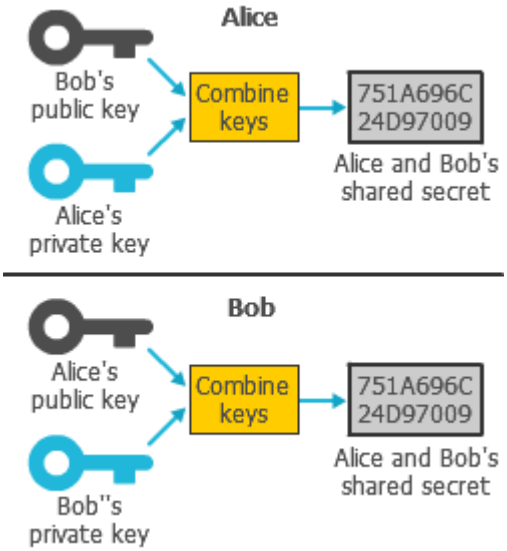


Figure(5.4): Public Key (Digital Signature)

Key Agreement Figure(5.5):

Key agreement (or key exchange) is a method in which the entity communicating in the network establishes a shared secret between them without exchanging any secret data. In this method, the entities that need to establish shared secret between them exchange their public keys. Both the entities on receiving the other entity's public key performs key generation operation using its private key to obtain the shared secret. The key generation algorithm 'Generate_Key' will be such that the generated keys at Alice and Bob will be the same. Since it is practically impossible to obtain private key from the public key any middleman, having access only to the public keys will never be able to obtain the shared secret K. Examples of key agreement algorithms are Diffie-Hellman, RSA and ECDH.

During the key exchange process the public keys may pass through different intermediate points. Any middleman can thus tamper or change the public keys to its public key. Therefore for establishing shared secret it is important that Alice receives the correct public key from Bob and vice versa. Digital Certificate helps to deliver the public key in authenticated method.



Figure(5.5): Public Key (Key Agreement)

6.A Trap-Door One Way Function

A trap-door one-way function is a function that is easy to calculate in one direction and infeasible to calculate in the other direction unless certain additional information is known. With the additional information the inverse can be calculated in polynomial time. We can summarize as follows: A trapdoor one-way function is a family of invertible functions, f_k such that:

$Y = f_k(X)$ easy, if k and X are known

$X = f_k^{-1}(Y)$ easy, if k and Y are known

$X = f_k^{-1}(Y)$ infeasible, if Y is known but k is not known

Thus, the development of a practical public-key scheme depends on discovery of a suitable trap-door one-way function.

We will study two examples of trap-door one-way function, which are factorization of big integer and the discrete logarithm problem.

Factorization of Big Integer: It is easy to take two prime numbers and multiply them together. If they are fairly small we can do this in our heads, on a piece of paper, or on a calculator. As they get bigger and bigger it is fairly easy to write a computer program to compute the product. Multiplication runs in polynomial time. For example, suppose p and q are two big prime integer. Multiplication of two primes $(n = p.q)$ is easy. For a large number with large prime factors, factoring is a hard problem, i.e factorizing n to deduce p and q. Thus, multiplication of two prime numbers is believed to be a one-way function. We say believed because nobody has been able to prove that it is hard to factorise.

The Discrete Logarithm Problem: The process of exponentiation just means raising numbers to a power. Raising a to the power b, normally denoted a^b just means multiplying a by itself b times. In other words: $a^b = a \times a \times a \times \ldots \times a^o$

Modular exponentiation means computing a^b modulo some other number n. We tend to write this as $a^b \bmod n$. Modular exponentiation is “easy”. However, given a, b, and $a^b \bmod n$ (when n is a large prime), calculating b is regarded by mathematicians as a hard problem. This difficult problem is often referred to as the

discrete logarithm problem. In other words, given a number a and a prime number n, the function $f(b) = a^b \bmod n$ is believed to be a one-way function.

7.Diffie-Hellman Algorithm

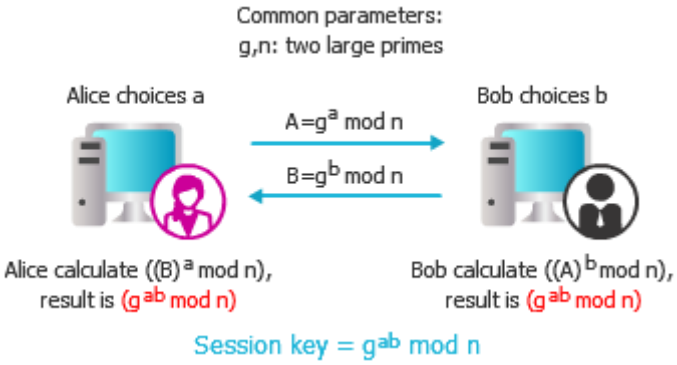
The Diffie–Hellman (DH) key exchange technique was first defined in their seminal paper in 1976. DH key exchange is a method of exchanging public (i.e. non-secret) information to obtain a shared secret. DH is not an encryption algorithm. DH is used in several network protocols and commercial products. With DH, keys can be generated as needed (on the fly) and they can be discarded at the end of the conversation.

DH key exchange has the following important properties:

- The DH algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.
- The resulting shared secret cannot be computed by either of the parties without the cooperation of the other.
- A third party observing all the messages transmitted during DH key exchange cannot deduce the resulting shared secret at the end of the protocol.

Here is how DH works figure(5.6), allowing Alice and Bob to establish a secret message key. Assume that n is some prime number and g is a base number:

- Alice generates a secret key, a
- Bob generates a secret key, b
- Alice computes a public key $A = g^a \bmod n$
- Bob computes a public key $B = g^b \bmod n$
- Bob and Alice exchange their public keys
- Alice now computes the message key, K, as $K = B^a \bmod n$
- Bob now computes the message key, K, as $K = A^b \bmod n$
- Both Bob and Alice end up with the same key: $K = g^{a \times b} \bmod n$



Figure(5.6): Diffie-Hellman Key Exchange

Example:

In practice, very large numbers are used (several hundred DIGITS each), but here is an example using small numbers:

$n = 11$, $g = 5$, $a = 2$, $b = 3$.

- Alice computes her public key $A = 5^2 \bmod 11 = 3$
- Bob computes his public key $B = 5^3 \bmod 11 = 4$
- Bob and Alice exchange their public keys
- Alice computes the message key, K, as $K = 4^2 \bmod 11 = 5$
- Bob computes the message key, K, as $K = 3^3 \bmod 11 = 5$
- Both end up with the same message key, namely:

  $K = 5^{2\times 3} \bmod 11 = 15{,}625 \bmod 11 = 5$

Man-in-the-Middle Attack:

The protocol depicted in Figure(5.6) is insecure against a man-in-the-middle attack. Suppose Alice and 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 d_1 and d_2 and then computing the corresponding public keys D_1 and D_2.
2. Alice transmits A to Bob.
3. Darth intercepts A and transmits D_1 to Bob. Darth also calculates.
   - $K_2 = (A)_2^d \bmod n$

4. Bob receives D_1 and calculates $K_1 = \left(D_1\right)^b \bmod n$.
5. Bob transmits B to Alice.
6. Darth intercepts B and transmits D_2 to Alice. Darth calculates.
   - $K_1 = \left(B\right)_1^d \bmod n$
7. Alice receives D_2 and calculates $K_2 = \left(D_2\right)^a \bmod n$.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K_1 and Alice and Darth share secret key K_2. All future communication between Bob and Alice is compromised. The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.

8.The RSA Algorithm

It is named after the three researchers Ron Rivest, Adi Shamir and Len Adleman who first published it. The RSA public key encryption algorithm was the first practical implementation of public key encryption discovered. It remains the most widely accepted and implemented general-purpose approach to public-key encryption.

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 $2^i < n \le 2^{i+1}$. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C.

- $C = M^e \bmod n$
- $M = C^d \bmod n = \left(M^e\right)^d \bmod n = M^{e.d} \bmod n$

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

For the first requirement, we need to find a relationship of the form

- $M^{e.d} \bmod n = M$

The preceding relationship holds if e and d are multiplicative inverses modulo $\Phi(n)$, where $\Phi(n)$ is the Euler totient function.

For p, q prime, $\Phi(p.q) = (p-1)(q-1)$. The relationship between e and d can be expressed as $e.d \bmod \Phi(n) = 1$. This is equivalent to saying $e.d = 1 \bmod \Phi(n)$

And $d = e^{-1} \bmod \Phi(n)$. That is, e and d are multiplicative inverses $\bmod \Phi(n)$. Note that, according to the rules of modular arithmetic, this is true only if d (and therefore e) is relatively prime to $\Phi(n)$.

8.1. RSA Operations

- Let n be the product of two large primes p and q and $\Phi(n) = (p-1)(q-1)$. By "large" we typically mean at least 1024 bits.
- Select a special number e such that e is greater than 1 and less than $\Phi(n)$. The precise mathematical property that e must have is that there must be no numbers that divide neatly into e and into $\Phi(n)$ except for 1. In other words e and $\Phi(n)$ are relatively prime, i.e the gcd $(e, f(n)) = 1$.
- Publish the pair of numbers (n,e), as the public key.
- Compute the private key d from p, q and e. As we have seen, the private key d is computed to be the unique inverse of e modulo $\Phi(n)$. Written mathematically: $e.d = 1 \bmod \Phi(n)$. The Euclidean Algorithm is the process that you need to follow in order to compute d.

Example:

In practice, very large numbers are used (several hundred DIGITS each), but here is an example using small numbers:

Step 1: Let $p = 47$ and $q = 59$. Thus $n = 47 \times 59 = 2773$

Step 2: Select $e = 17$

Step 3: Publish $(n, e) = (2773, 17)$

Step 4: $\Phi(n) = (p-1) \times (q-1)$

- $\ldots\ldots\ldots = 46 \times 58 = 2668$
- Use the Euclidean Algorithm to compute the modular inverse of 17 mod 2668. The result is $d = 157$
- << Check: $17 \times 157 = 2669 = 1 \pmod{2668}$ >>

Public key is $(2773, 17)$

Private key is 157

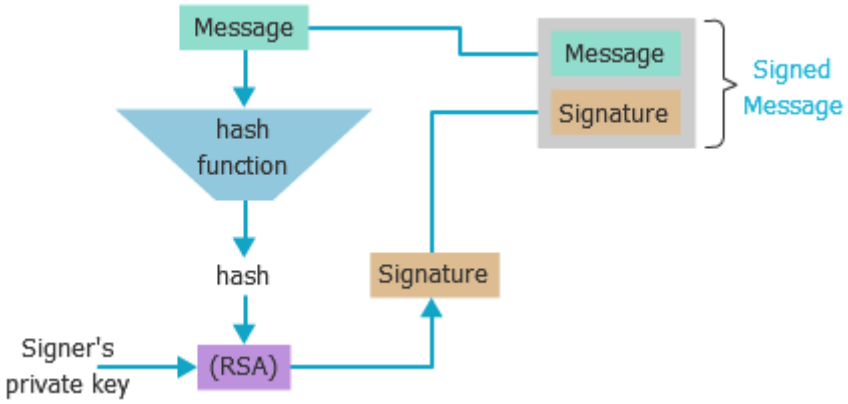
Suppose the plaintext block represented as a number: $M = 31$

Encryption using Public Key: $C = 31^{17} \pmod{2773} = 587$

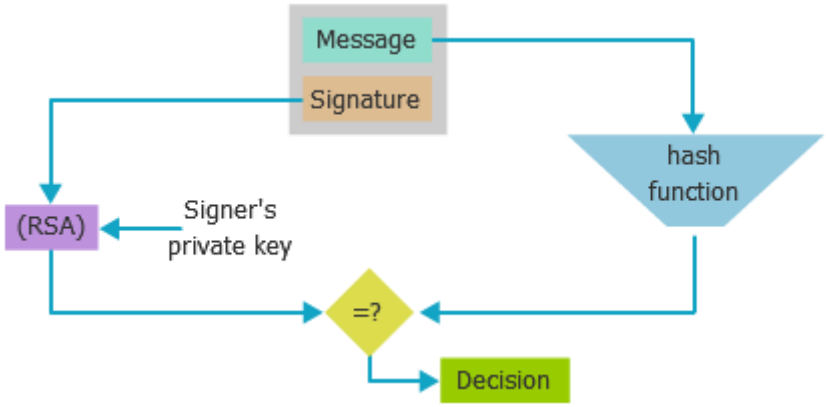
Decryption using Private Key: $M = 587^{157} \pmod{2773} = 31$

RSA Signature:

RSA signature is similar to RSA encryption with inverse roles of keys, i.e., for signing, the sender's private key is used to encrypt the hashed code. The verifier decrypts the signature with the sender's public key and then compares the result with the message's hash code.



Figure(5.7): RSA signature



Figure(5.8): RSA Signature Verification

8.2.Security of RSA

We will look at two different strategies for trying to "break" RSA:

1. Trying to decrypt a ciphertext without knowledge of the private key: The encryption process in RSA involves computing the function $C = M^e \bmod n$, which is regarded as being easy. An attacker who observes this ciphertext c, and has knowledge of e and n, needs to try to work out what m is. i.e., find m such that $m^e = c \bmod n$. In other words, find the e^{th} root of c mod n. computing m from c, e and n is regarded as a hard problem and known as RSA problem.
2. Trying to determine the private key: If the attacker knows the public key of a user (e,n), what would he need to do in order to obtain the corresponding private key? He needs to find d such that $e.d \bmod \Phi(n) = 1$ i.e., needs to know p and q. In other words, he must factor n (problem of prime factorization).

   Recommended size of n: RSA Lab. claims that 1024-bit keys are likely to become crackable sometime between 2006 and 2010 and that 2048-bit keys are sufficient until 2030. An RSA key length of 3072 bits should be used if security is required beyond 2030.

9.Public Key Certificate

The point of public-key encryption is that 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 to any other participant. Although this approach is convenient, it has a major weakness. Anyone can forge such a public announcement. That is, some user could pretend to be Bob and send a public key to another participant. The forger is able to read all encrypted messages intended for Bob and can use the forged keys for authentication. The solution to this problem is the public-key certificate.

In essence, a certificate consists of a public key plus a user ID of the key owner, with the whole block signed by a trusted third party. The certificate also includes some information about the third party plus an indication of the period of validity of the certificate. Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution. Mathematically, we can write the certificate as

$$C_A = \left\{ \text{Alice}, K_{u\_a}, DS_{kv}\left(K_{u\_a} \| \text{Alice} \| T\right) \right\}$$

Where:

K_{u_a} is Alice public key

DS_{kv} is the digital signature function by using the CA private key

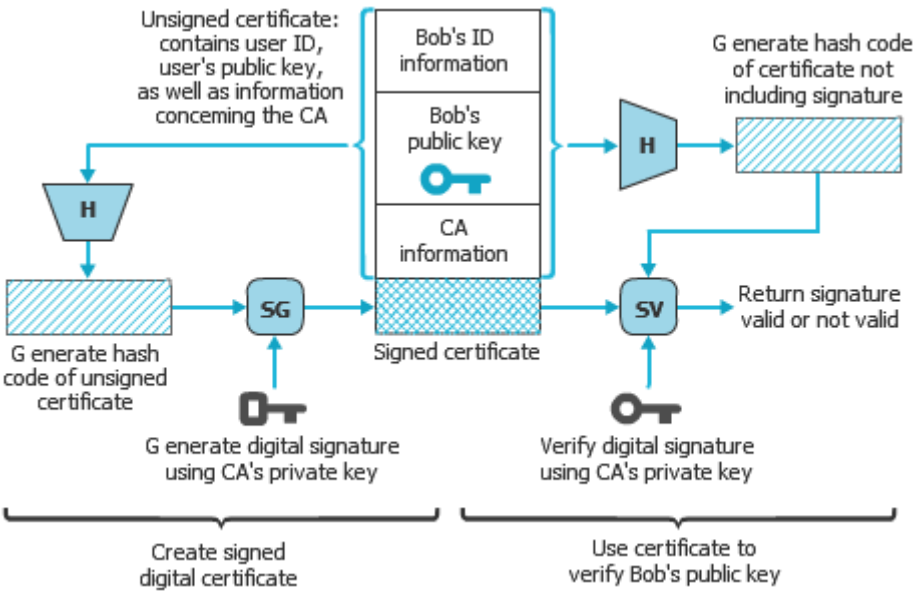
When Bob gets Alice's certificate, if he knows CA's public key, he can validate the certificate, and be certain of Alice's public key. There is still a problem is that: Bob needs CA's public key to validate Alice's certificate. There are many solutions:

- Public Key Infrastructure (PKI)
- Trust-based certificates (PGP)

The key steps to get a certificate can be summarized as follows figure(5.9):

1. User software (client) creates a pair of keys: one public and one private.
2. Client prepares an unsigned certificate that includes the user ID and user's public key.
3. User provides the unsigned certificate to a CA in some secure manner. This might require a face-to-face meeting, the use of registered e-mail, or happen via a Web form with e-mail verification.

4. CA creates a signature as follows:
   a. CA uses a hash function to calculate the hash code of the unsigned certificate.
   b. CA generates digital signature using the CA's private key and a signature generation algorithm.
5. CA attaches the signature to the unsigned certificate to create a signed certificate.
6. CA returns the signed certificate to client.
7. Client may provide the signed certificate to any other user.
8. Any user may verify that the certificate is valid as follows:
   a. User calculates the hash code of certificate (not including signature).
   b. User verifies digital signature using CA's public key and the signature verification algorithm. The algorithm returns a result of either signature valid or invalid.



Figure(5.9): Public Key Certificate

One scheme has become universally accepted for formatting public-key certificates: the X.509 standard. X.509 certificates are used in most network security application. Key certificate fields in X.509v3 (RFC 5280):

- Version
- Serial number (unique)
- Signature algorithm identifier: hash algorithm
- Issuer's name, uniquely identifies issuer
- Interval of validity
- Subject's name, uniquely identifies subject
- Subject's public key
- Signature

10.Public Key Infrastructure (PKI)

A Public Key Infrastructure (PKI) is a set of services and policies that lays the framework for binding a public key to an identity and distributing that binding. In other words, a PKI is a system that provides authentic public keys to applications.

A PKI has three basic processes which are:

1. Certification: Binding identities (or attributes) to a public key by a trusted third party to form a certificate. A trusted third party in a PKI is called Certificate Authority (CA). A CA digitally signs a certificate means it vouches for the correctness of the certificate's contents.
2. Validation: The process of verifying the authenticity of the certificate. This means that the certificate's contents can be believed. This involves:
   a. Verifying the signature of the CA by using the CA's public key
   b. Checking the validity period contained in the certificate itself
   c. Checking the certificate against a Certificate Revocation List (CRL)
3. Certificate Revocation: The process of disavowing a previously issued certificate before its expiration date. This may happen if some of the information contained in the certificate changes. A CRL contains certificate that have been revoked (ie, not valid) by the CA.

10.1. Advantages and Disadvantages of Public Key Cryptography

The primary advantage of public-key cryptography is increased security and convenience: private keys never need to transmitted or revealed to anyone. In a secret-key system, by contrast, the secret keys must be transmitted (either manually or through a communication channel), and there may be a chance that an enemy can discover the secret keys during their transmission.

Another major advantage of public-key systems is that they can provide a method for digital signatures. Authentication via secret-key systems requires the sharing of some secret and sometimes requires trust of a third party as well. As a result, a sender can repudiate a previously authenticated message by claiming that the shared secret was somehow compromised by one of the parties sharing the secret. Public-key authentication, on the other hand, prevents this type of repudiation, each user has sole responsibility for protecting his or her private key. This property of public-key authentication is often called non-repudiation.

A disadvantage of using public-key cryptography for encryption is speed: there are popular secret-key encryption methods that are significantly faster than any currently available public-key encryption method. Nevertheless, public-key cryptography can be used with secret-key cryptography to get the best of both worlds. Such a protocol is called the hybrid encryption. Public key cryptography may be vulnerable to impersonation, however, even if users' private keys are not available. A successful attack on a certification authority will allow an adversary to impersonate whomever the adversary chooses to by using a public-key certificate from the compromised authority to bind a key of the adversary's choice to the name of another user.

In some situations, public-key cryptography is not necessary and secret-key cryptography alone is sufficient. This includes environments where secure secret-key agreement can take place, for example by users meeting in private. It also includes environments where a single authority knows and manages all the keys, e.g., a closed banking system. Since the authority knows everyone's keys already, there is not much advantage for some to be "public" and others "private". Also, public-key cryptography is usually not necessary in a single-user environment. For example, if you want to keep your personal files encrypted, you can do so with any secret-key encryption algorithm using, say, your personal password as the secret key. In general, public-key cryptography is best suited for an open multi-user environment.

Exercises:

True or False

Question	True	False
1. The one-way hash function is important not only in message authentication but also in digital signatures.		
2. SHA is perhaps the most widely used family of hash functions.		
3. Asymmetric encryption utilizes only a public key for encryption and decryption.		
4. Asymmetric encryption can be used for confidentiality but not for authentication.		
5. Plaintext is transformed into ciphertext using two keys and a decryption algorithm.		
6. Public-key encryption is more secure from cryptanalysis than symmetric encryption.		
7. Asymmetric algorithms rely on one key for encryption and a different but related key for decryption.		
8. If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing.		
9. A trap-door one-way function is easy to calculate in one direction and infeasible to calculate in the other direction unless certain additional information is known.		
10. A public-key encryption scheme is not vulnerable to a brute-force attack.		
11. RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and $n-1$ for some n.		

12. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.		
13. The key exchange protocol is vulnerable to a man-in-the-middle attack because it does not authenticate the participants.		
14. Unlike RSA, DSS cannot be used for encryption or key exchange.		
15. Cryptographic hash functions generally execute faster in software than conventional encryption algorithms such as DES.		

Multiple Choice Questions

1. SHA-1 produces a hash value of _____ bits.
   - **A.** 256
   - **B.** 160
   - **C.** 384
   - **D.** 180

2. The _____ scheme has reigned supreme as the most widely accepted and implemented approach to public-key encryption.
   - **A.** SHA-1
   - **B.** HMAC
   - **C.** MD5
   - **D.** RSA

3. A _____ attack involves trying all possible private keys.
   - **A.** mathematical
   - **B.** timing
   - **C.** brute-force
   - **D.** chosen ciphertext

4. _____ was the first published public-key algorithm.
   - **A.** NIST
   - **B.** Diffie-Hellman
   - **C.** RC4
   - **D.** RSA

5. The _____ uses an algorithm that is designed to provide only the digital signature function and cannot be used for encryption or key exchange.

A. ECC

B. RSA

C. DSS

D. XOR

6. Public-key encryption is also known as _____ .

A. digital-key encryption

B. asymmetric encryption

C. one-way time-exchange encryption

D. optimal-key encryption

7. Plaintext is recovered from the ciphertext using the paired key and _____ .

A. a digital signature

B. a recovery encryption

C. a decryption algorithm

D. an encryption algorithm

8. The most widely used public-key cryptosystem is _____ .

A. optimal asymmetric encryption

B. asymmetric encryption

C. RSA

D. DES

9. _____ are 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.

A. Asymmetric keys

B. Key exchanges

C. Symmetric keys

D. Cipher keys

10. The _____ indicates that the subscriber identified in the certificate has sole control and access to the private key.

A. OAEP

B. Public Key Certificate

C. Digital Signature

D. PKI

11. The key used in symmetric encryption is referred to as a _____ key.

A. public

B. secret

C. private

D. decryption

12. The readable message or data that is fed into the algorithm as input is the _____ .

A. ciphertext

B. exchange

C. plaintext

D. encryption

References

1. Stallings, W.: Cryptography and Network Security: Principles and Practice, 7th edn. Prentice Hall (2017).
2. Stallings, W.: Network Security essentials: application and standards, 6th edn. Pearson India Education Services Pvt. LTD (2017).
3. https://www.sans.org/network-security/.

Number of the Question	True	False
1	✓	
2	✓	
3		✓
4		✓
5		✓
6		✓
7	✓	
8	✓	
9	✓	
10		✓
11	✓	
12	✓	
13	✓	
14	✓	
15	✓	

Number of the Question	Answer
1	B
2	D
3	C
4	B
5	C
6	B
7	C
8	C
9	A
10	B
11	B
12	C