جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الأمن السيبراني

Lecture: (3)

Authentication technologies

Subject: authentication and access control

second Stage

Lecturer: Asst. Lecturer. Suha Alhussieny

User Authentication

A user authentication policy is a process in which you verify that someone who is attempting to access services and applications is who they claim to be. This can be accomplished through a variety of authentication methods, such as entering a password into your laptop or phone or a PIN number into the ATM.

What are the different authentication protocols?

Network authentication protocols are used to help securely transfer identity credentials for authentication between the subject (user or device) and the authentication server. There are several different authentication protocols for network access control, including:

1. **Kerberos**
2. **Extensible Authentication Protocol (EAP)**
3. **IEEE 802.1X**
4. **Remote Authentication Dial-In User Service (RADIUS)**
5. **Terminal Access Controller Access-Control System (TACACS)**

Message Authentication

Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that data received are exactly as sent by (i.e., contain no modification, insertion, deletion, or replay) and that the purported identity of the sender is valid.

One of the most fascinating and complex areas of cryptography is that of message authentication and the related area of digital signatures. There are two main components of message authentication: -

1. **Message integrity:** - Ensure that the message has not been changed or altered during transmission, any alteration in the message will be detected by the receiver.

2. **Message authenticity**: - confirms that the message indeed originated from the claimed sender and not from an impersonator.

Message Authentication Functions

Any message authentication or digital signature mechanism has two levels of functionality. At the lower level, there must be some sort of function that produces an authenticator**:** a value to be used to authenticate a message. This lower-level function is then used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

The types of functions that may be used to produce an authenticator. These may be grouped into three classes.

1. **Hash function**: A function that maps a message of any length into a fixed length hash value, which serves as the authenticator.
2. **Message encryption:** The cipher text of the entire message serves as its authenticator
3. **Message Authentication Code (MAC):** A function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

Message Encryption

Message encryption by itself can provide a measure of authentication. The analysis differs for symmetric and public-key encryption schemes.

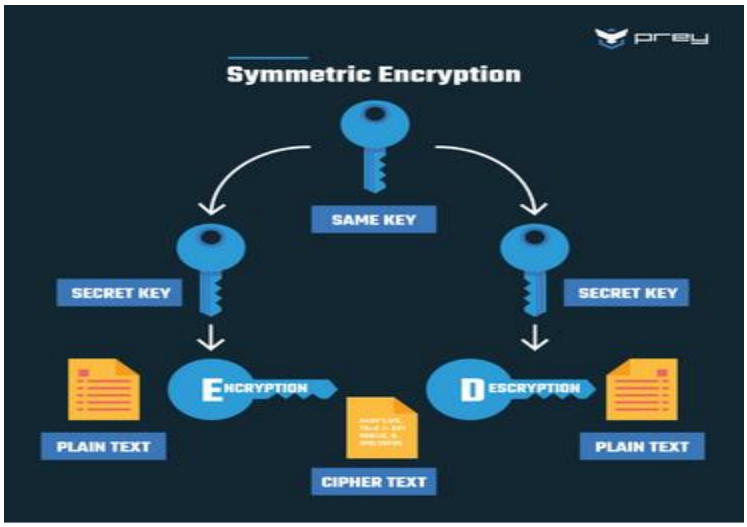
SYMMETRIC ENCRYPTION

Symmetric encryption is an encryption method that uses a single key to encrypt and decrypt data. Though generally less secure than asymmetric encryption, it's often considered more efficient because it requires less processing power.

Encryption is the process of transforming readable plaintext into unreadable ciphertext to mask sensitive data from unauthorized users.

Consider the straightforward use of symmetric encryption Figure **(12.1a).** A message transmitted from source A to destination B is encrypted using a secret key shared by A and B. If no other party knows the key, then confidentiality is provided: No other party can recover the plaintext of the message.

In addition, B is assured that the message was generated by A. Why? The message must have come from A, because A is the only other party that possesses K and therefore the only other party with the information necessary to construct ciphertext that can be decrypted with K. Furthermore, if M is recovered, B knows that none of the bits of M have been altered, because an opponent that does not know K would not know how to alter bits in the ciphertext to produce the desired changes in the plaintext.



So we may say that symmetric encryption provides authentication as well as confidentiality. However, this flat statement needs to be qualified. Consider exactly what is happening at B. Given a decryption function D and a secret key K, the destination will accept any input X and produce output Y = D(K, X) . If X is

the ciphertext of a legitimate message M produced by the corresponding encryption function, then Y is some plaintext message M. Otherwise, Y will likely be a meaningless sequence of bits. There may need to be some automated means of determining at B whether Y is legitimate plaintext and therefore must have come from A.

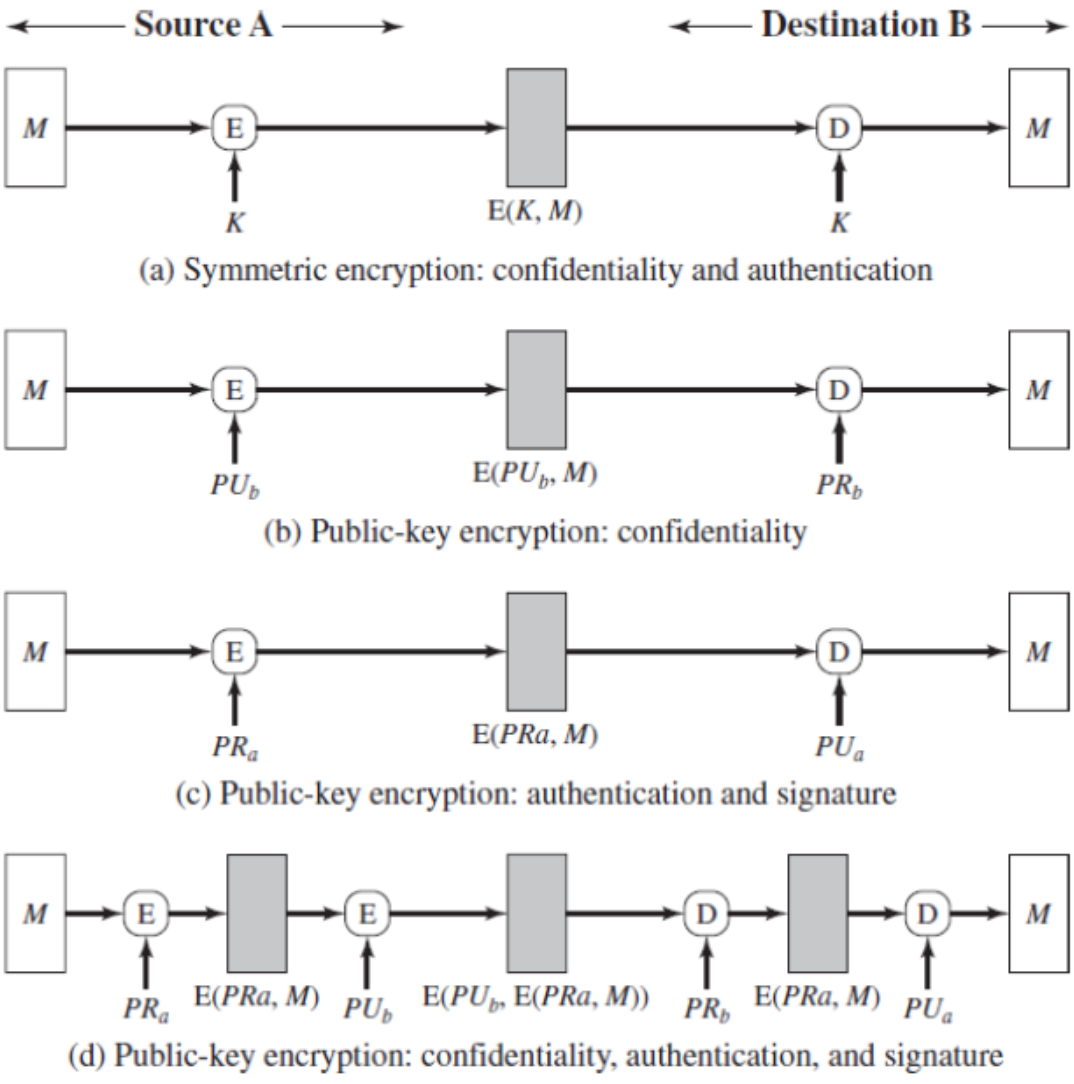


Figure (12.1) Basic Uses of Message Encryption

Symmetric encryption, also known as **symmetric key cryptography** or **secret-key encryption**, is one of 2 main methods of encryption alongside asymmetric encryption. Symmetric encryption works by creating a single shared key to encrypt and decrypt sensitive data. The main advantage of symmetric encryption is that it's generally simple and efficient in securing data.

However, symmetric encryption is often considered less secure than asymmetric encryption, largely because it relies on secure key exchange and meticulous key management. Anyone who intercepts or obtains the symmetric key can access the data.

For this reason, organizations and messaging apps increasingly rely on a hybrid encryption method that uses asymmetric encryption for secure key distribution and symmetric encryption for subsequent data exchanges.

Symmetric encryption involves two main types of symmetric ciphers: **block ciphers** and **stream ciphers**.

- **Block ciphers**, such as Advanced Encryption Standard (AES), encrypt data in fixed-size blocks.
- **Stream ciphers**, like RC4, encrypt data one bit or byte at a time, making them suitable for real-time data processing.

Users frequently choose block ciphers to ensure data integrity and security for large amounts of data. They choose stream ciphers to encrypt smaller, continuous data streams efficiently, such as real-time communications.

Example of symmetric encryption

Imagine Alice wants to send a confidential document to Bob. In this scenario, symmetric encryption would work as follows:

1. Alice and Bob agree on a secret key or use asymmetric encryption for secure key exchange.
2. Alice encrypts the document using the secret key, turning it into unreadable ciphertext.
3. Alice sends the ciphertext to Bob.
4. Upon receiving the encrypted document, Bob uses the same secret key to decrypt it back to its original form, ensuring its confidentiality throughout transmission.

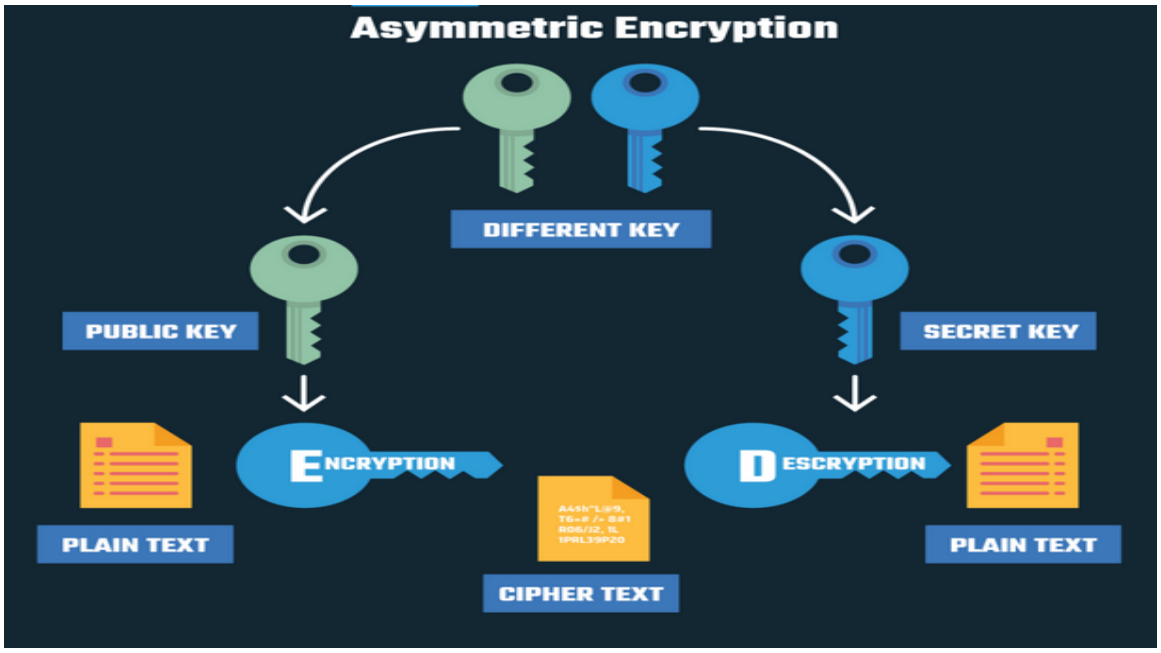
The most well-known symmetric key **algorithms** include:

- Data Encryption Standard (DES) and Triple DES (3DES)
- Advanced Encryption Standard (AES)
- Twofish
- Blowfish

PUBLIC-KEY ENCRYPTION

The straightforward use of public-key encryption **(Figure 12.1b)** provides confidentiality but not authentication. The source (A) uses the public key *PUb* of the destination (B) to encrypt *M*. Because only B has the corresponding private key *PRb*, only B can decrypt the message. This

scheme provides no authentication, because any opponent could also use B's public key to encrypt a message and claim to be A.



To provide authentication, A uses its private key to encrypt the message, and B uses A's public key to decrypt **(Figure 12.1c).**This provides authentication using the same type of reasoning as in the symmetric encryption case: The message must have come from A because A is the only party that possesses *PRa* and therefore the only party with the information necessary to construct ciphertext that can be decrypted with PUa . Again, the same reasoning as before applies: There must be some internal structure to the plaintext so that the receiver can distinguish between well-formed plaintext and random bits.

Assuming there is such structure, then the scheme of Figure **12.1c** does provide authentication. It also provides what is known as digital signature.1

Only A could have constructed the ciphertext because only A possesses *PRa* . Not even B, the recipient, could have constructed the ciphertext. Therefore, if B is in possession of the ciphertext, B has the means to prove that the

message must have come from A. In effect, A has "signed" the message by using its private key to encrypt. Note that this scheme does not provide confidentiality. Anyone in possession of A's public key can decrypt the ciphertext.

To provide both confidentiality and authentication, A can encrypt *M* first using its private key, which provides the digital signature, and then using B's public key, which provides confidentiality **(Figure 12.1d).** 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.