EC-Council



C|CT

Certified | Cybersecurity Technician

Module - 14

Cryptography

This page is intentionally left blank.



Module Objectives

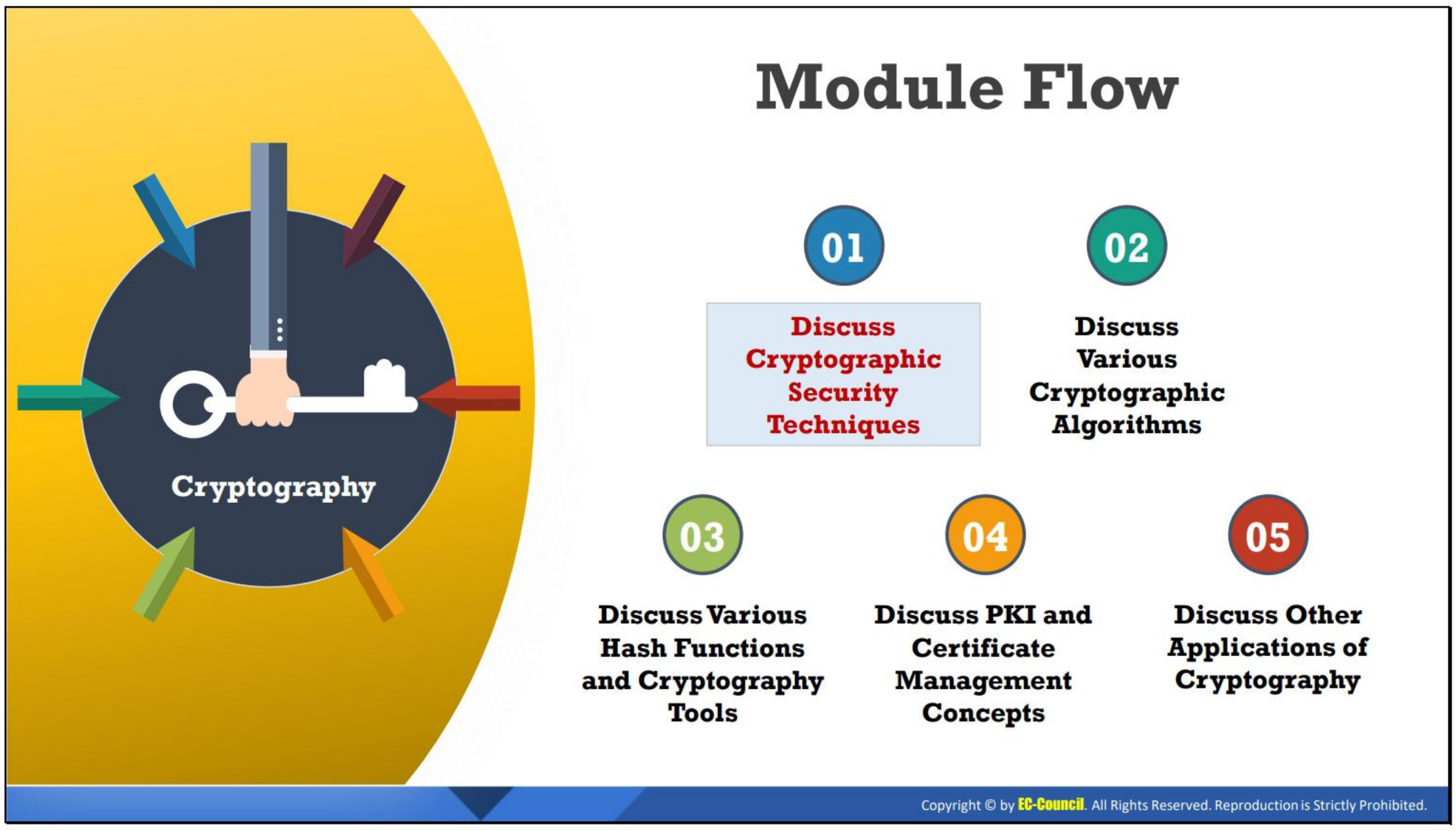
With the increasing adoption of the Internet (World Wide Web) for business and personal communication, securing sensitive information such as credit card details, PINs, bank account numbers, and private messages is becoming increasingly important, albeit more difficult to achieve. Today's information-based organizations extensively use the Internet for e-commerce, market research, customer support, and a variety of other activities. Data security is critical to online business and communication privacy.

Cryptography and cryptographic ("crypto") systems help in securing data against interception and compromise during online transmissions. This module provides a comprehensive understanding of different cryptosystems and algorithms, one-way hash functions, and public-key infrastructures (PKIs). It also covers various tools used to encrypt sensitive data.

At the end of this module, you will be able to do the following:

- Describe cryptographic techniques
- Understand the different encryption algorithms
- Understand the different hashing algorithms
- Use different cryptography tools and hash calculators
- Explain public key infrastructure (PKI) and certificate management concepts
- Understand various applications of cryptography

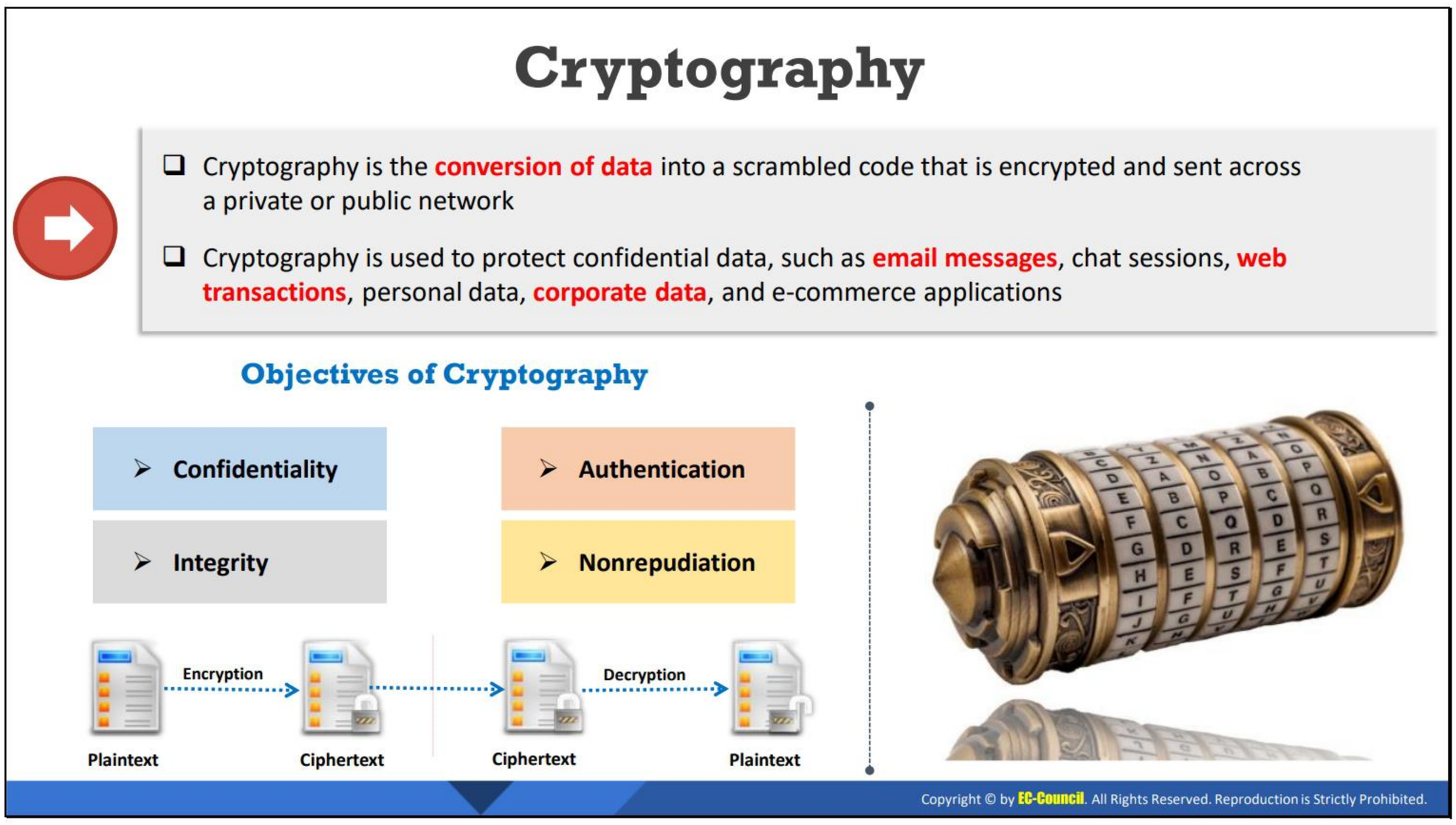
Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Discuss Cryptographic Security Techniques

Cryptography enables one to secure transactions, communications, and other processes performed in the electronic world. This section deals with cryptography and its associated concepts, which will enable you to understand the other topics covered later in this module.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cryptography

Cryptography" comes from the Greek words ***kryptos***, meaning "concealed, hidden, veiled, secret, or mysterious," and ***graphia***, meaning "writing"; thus, cryptography is "the art of secret writing."

Cryptography is the practice of concealing information by converting plaintext (readable format) into ciphertext (unreadable format) using a key or encryption scheme. It is the process of converting data into a scrambled code that is encrypted and sent across a private or public network. Cryptography protects confidential data such as email messages, chat sessions, web transactions, personal data, corporate data, e-commerce applications, and many other types of communication. Encrypted messages can, at times, be decrypted by cryptanalysis (code breaking), even though modern encryption techniques are virtually unbreakable.

Objectives of Cryptography

- **Confidentiality**: Assurance that the information is accessible only to those authorized to access it.
- **Integrity**: Trustworthiness of data or resources in terms of preventing improper and unauthorized changes.
- **Authentication**: Assurance that the communication, document, or data is genuine.
- **Nonrepudiation**: Guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cryptography Process

Plaintext (readable format) is encrypted by means of encryption algorithms such as RSA, DES, and AES, resulting in a ciphertext (unreadable format) that, on reaching the destination, is decrypted into readable plaintext.

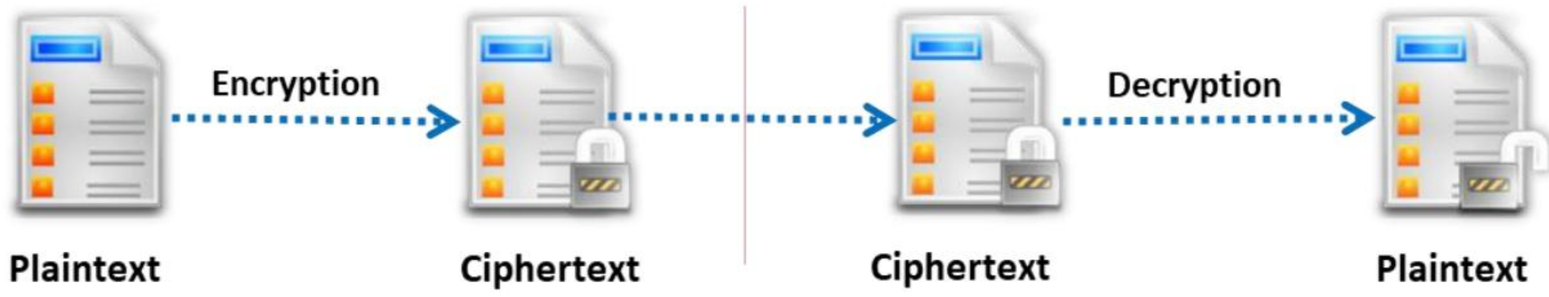
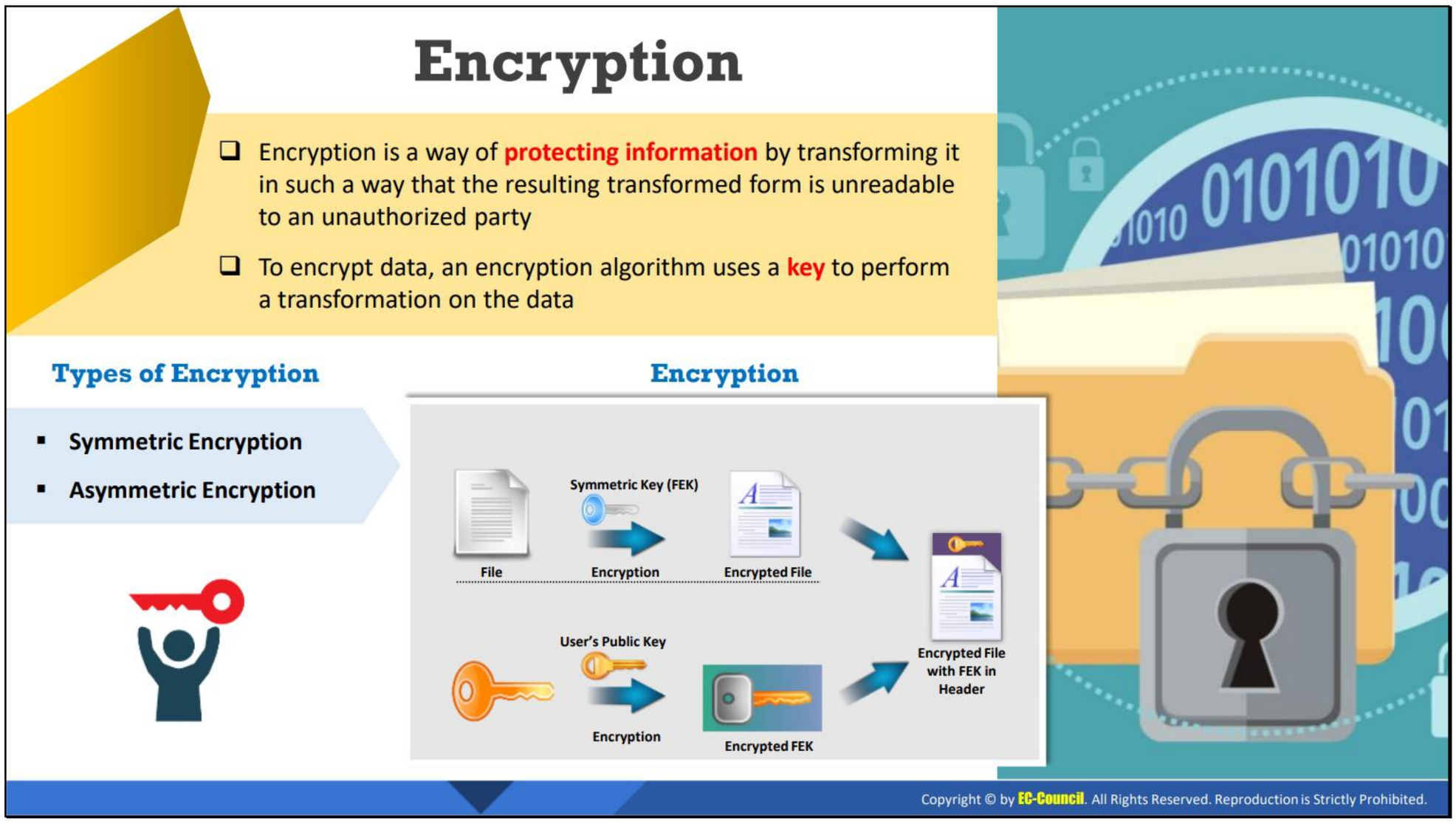


Figure 14.1: Example of Cryptography

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Encryption

Encryption is the practice of concealing information by converting a plain text (readable format) into a cipher text (unreadable format) using a key or an encryption scheme. Encryption guarantees the confidentiality and integrity of the organization's data, at rest or in transit.

The encryption algorithm encrypts the plain text with the help of an encryption key. The encryption process creates a cipher text that needs decrypting with the help of a key. The process of decryption involves the same steps except for the usage of keys in the reverse order.

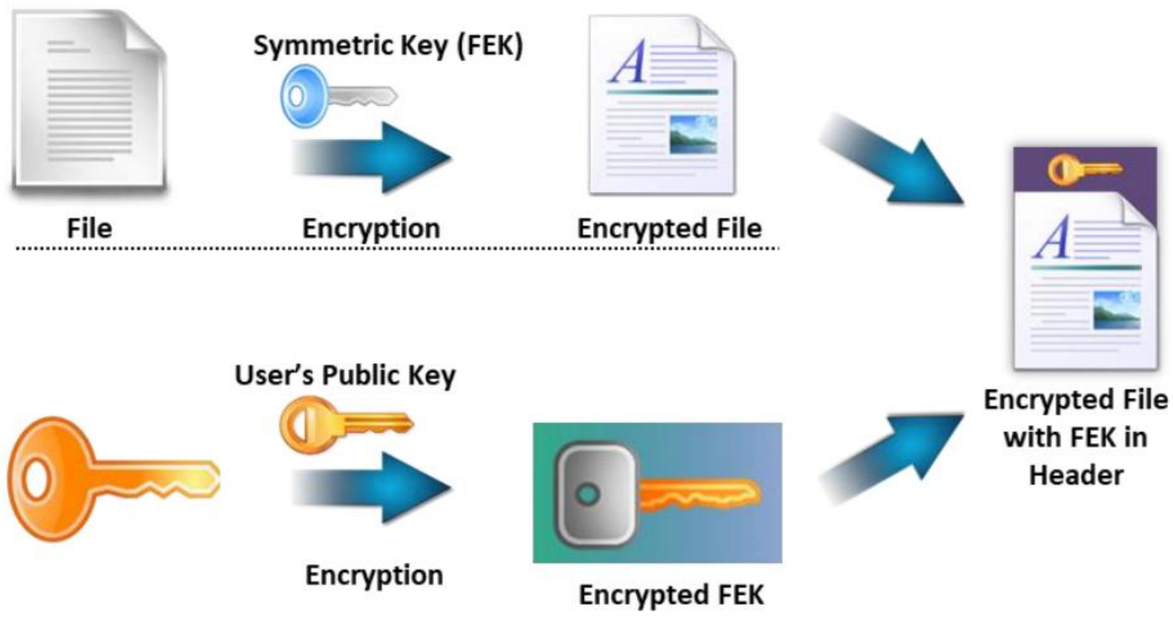


Figure 14.2: Encryption

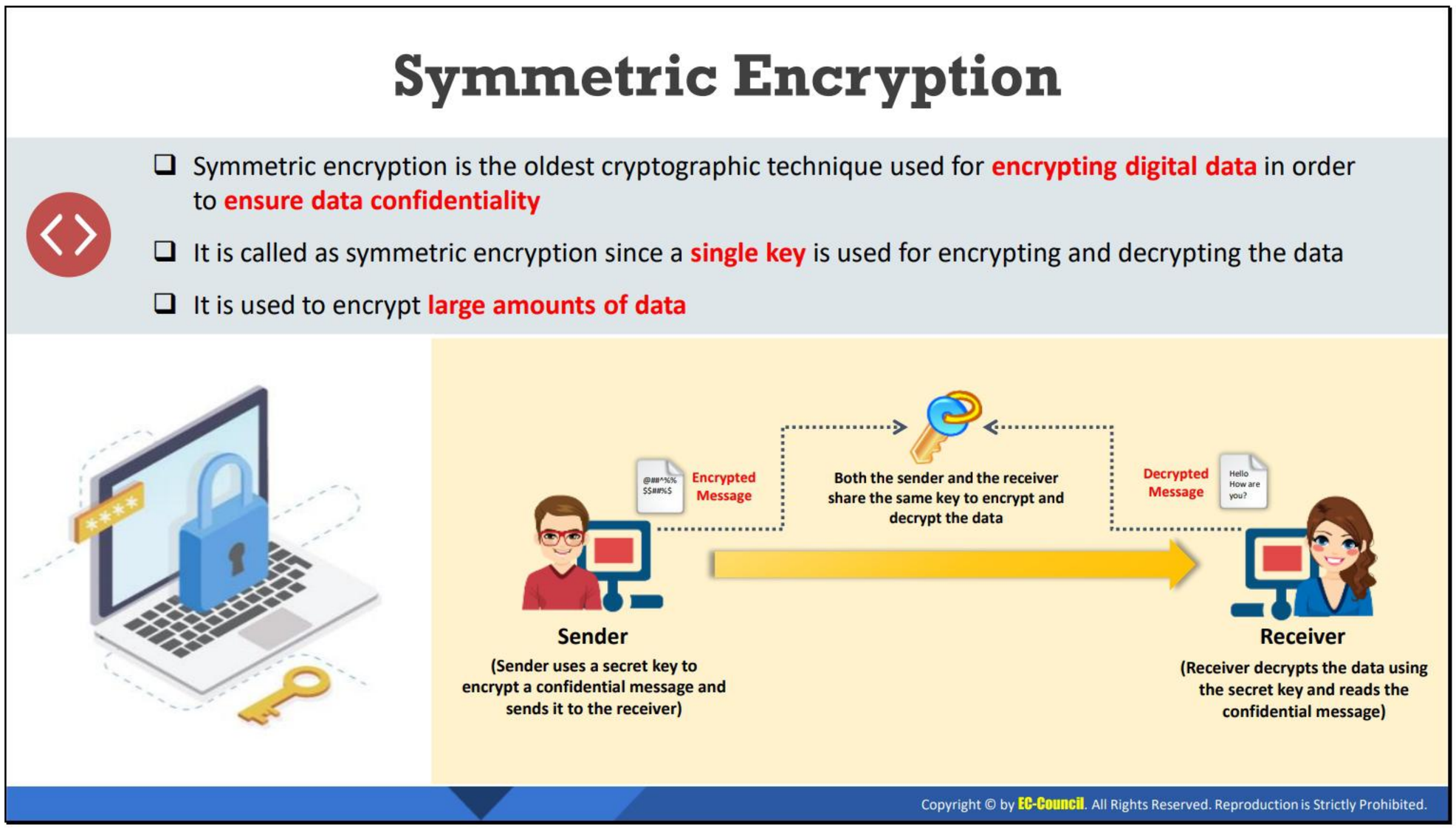
The encryption process is generally applied while transmitting data through a network, mobile phones, wireless transmission, and in Bluetooth devices.

Types of Encryption

There are two types of encryption.

- Symmetric Encryption
- Asymmetric Encryption

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Symmetric Encryption

Symmetric encryption requires that both the sender and the receiver of the message possess the same encryption key. The sender uses a key to encrypt the plain text and sends the resulting cipher text to the recipient, who uses the same key to decrypt the cipher text into plain text. Symmetric encryption is also known as secret key cryptography since it uses only one secret key to encrypt and decrypt the data. This type of cryptography works well when one is communicating with only a few people.

Because the sender and receiver must share the key prior to sending any messages, this technique is of limited use over the Internet in the case where individuals who have not had prior contact frequently require a secure means of communication. The solution to this problem is the public-key cryptography.

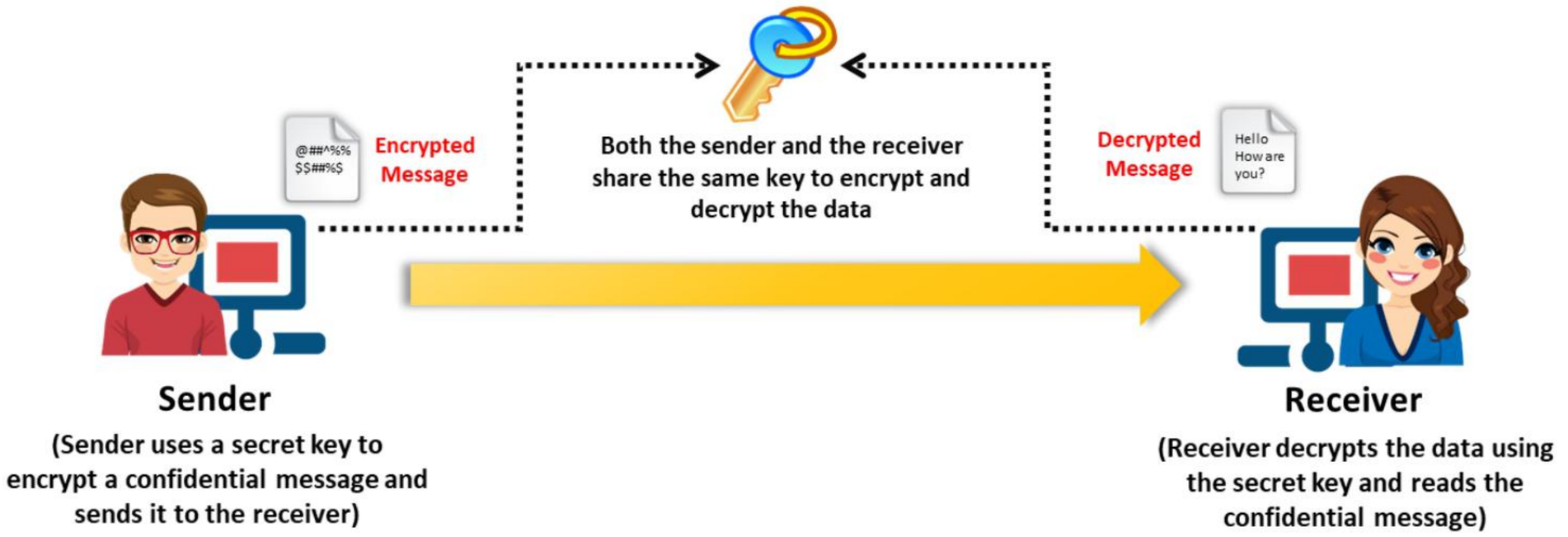


Figure 14.3: Symmetric Encryption

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The symmetric key encryption can use either stream ciphers or block ciphers. Stream ciphers encrypt the bits of a message one at a time, whereas block ciphers encrypt blocks of bits.

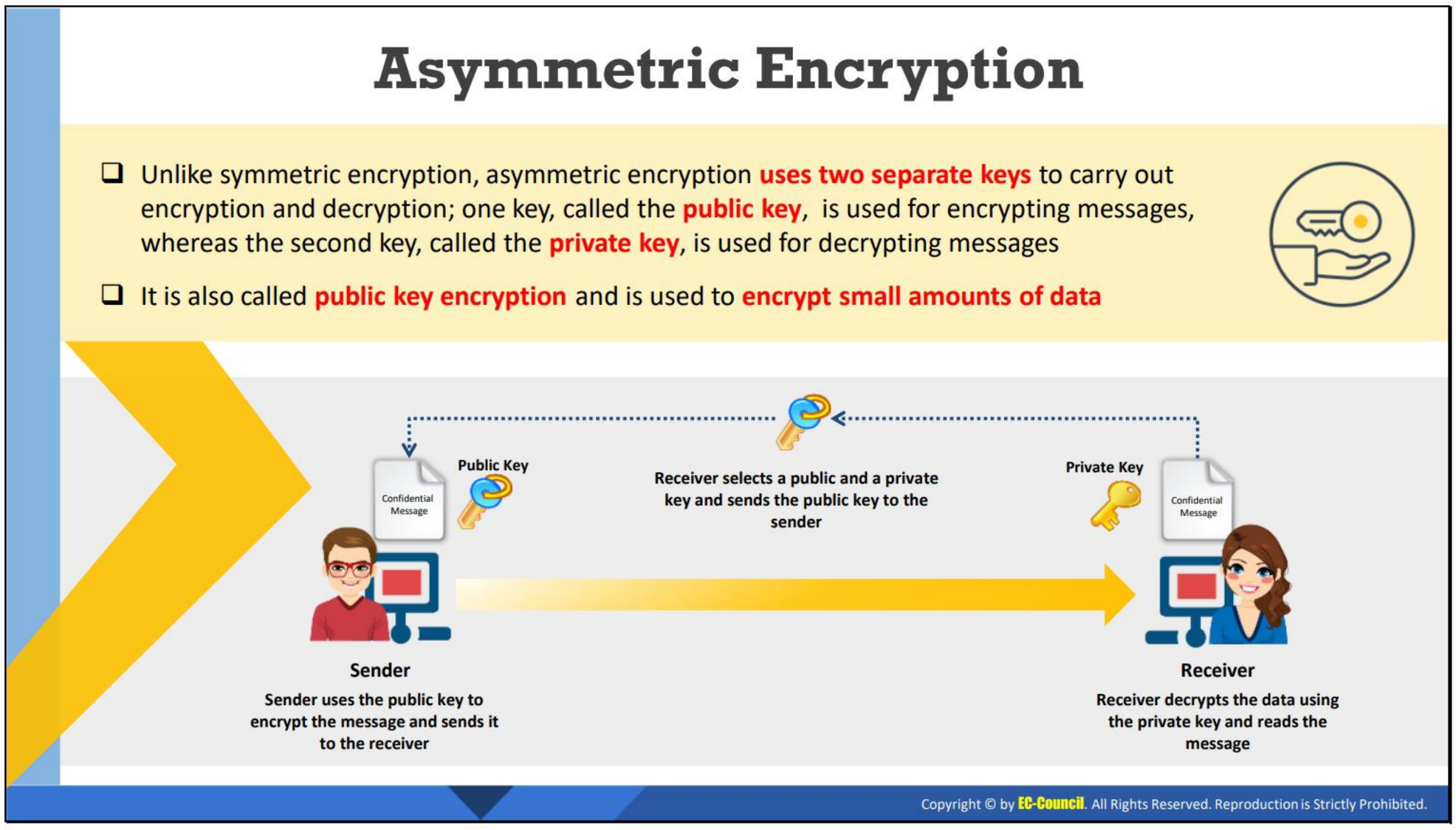
Advantages:

- It is easy to encrypt and decrypt a message
- It is faster than asymmetric encryption
- It is used to encrypt large amounts of data

Disadvantages:

- The communicating parties need to share the key used for transmitting the data
- Unauthorized access to a symmetric key leads to the compromise of data at both ends

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Asymmetric Encryption

Asymmetric encryption was introduced for solving key-management problems. Asymmetric encryption involves a public key and a private key. The public key is publicly available, whereas the sender keeps the private key a secret. It is also called public key encryption and is used to encrypt small amounts of data.

Asymmetric encryption uses the following sequence to send a message:

1. An individual finds the public key of the person they want to contact in a directory.
2. This public key is used for encrypting a message that is sent to the intended recipient.
3. The receiver uses the private key to decrypt the message for reading it.

No one except the holder of the private key can decrypt a message composed with the corresponding public key. This increases the security of the information because all communications involve only public keys; the message sender never transmits or shares the private keys. The sender must link the public keys with the usernames in a secured method to ensure that unauthorized individuals, claiming to be the intended recipient, do not intercept the information. To meet the requirement of authentication, one can use digital signatures.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

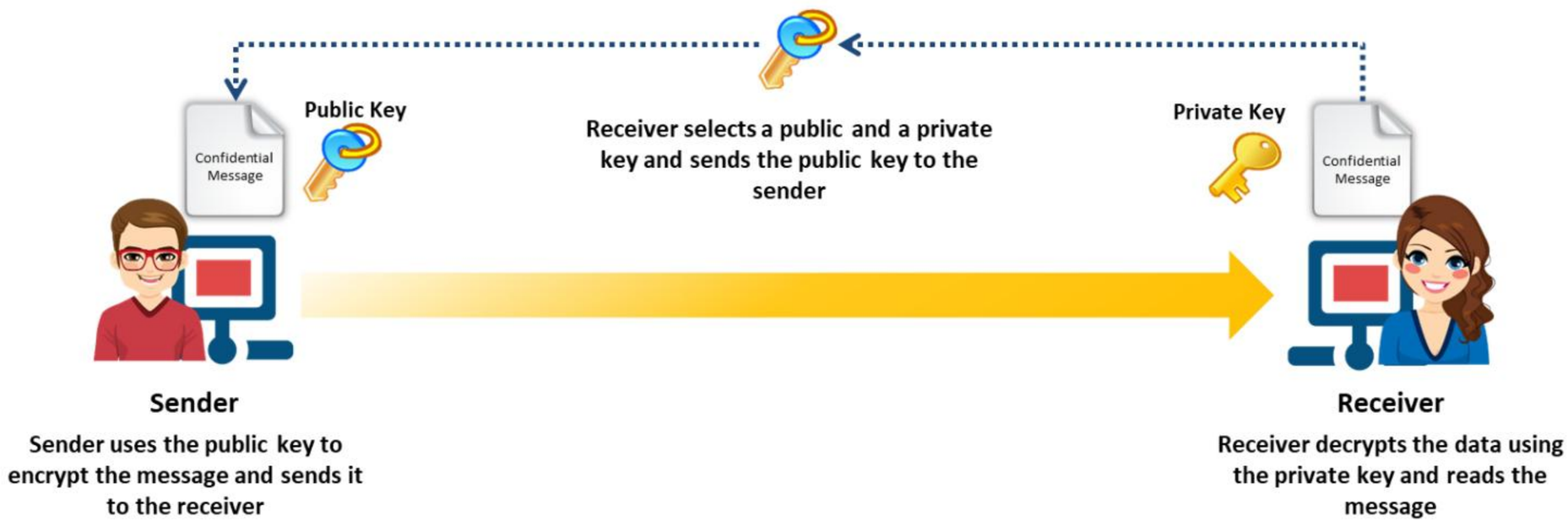


Figure 14.4: Asymmetric Encryption

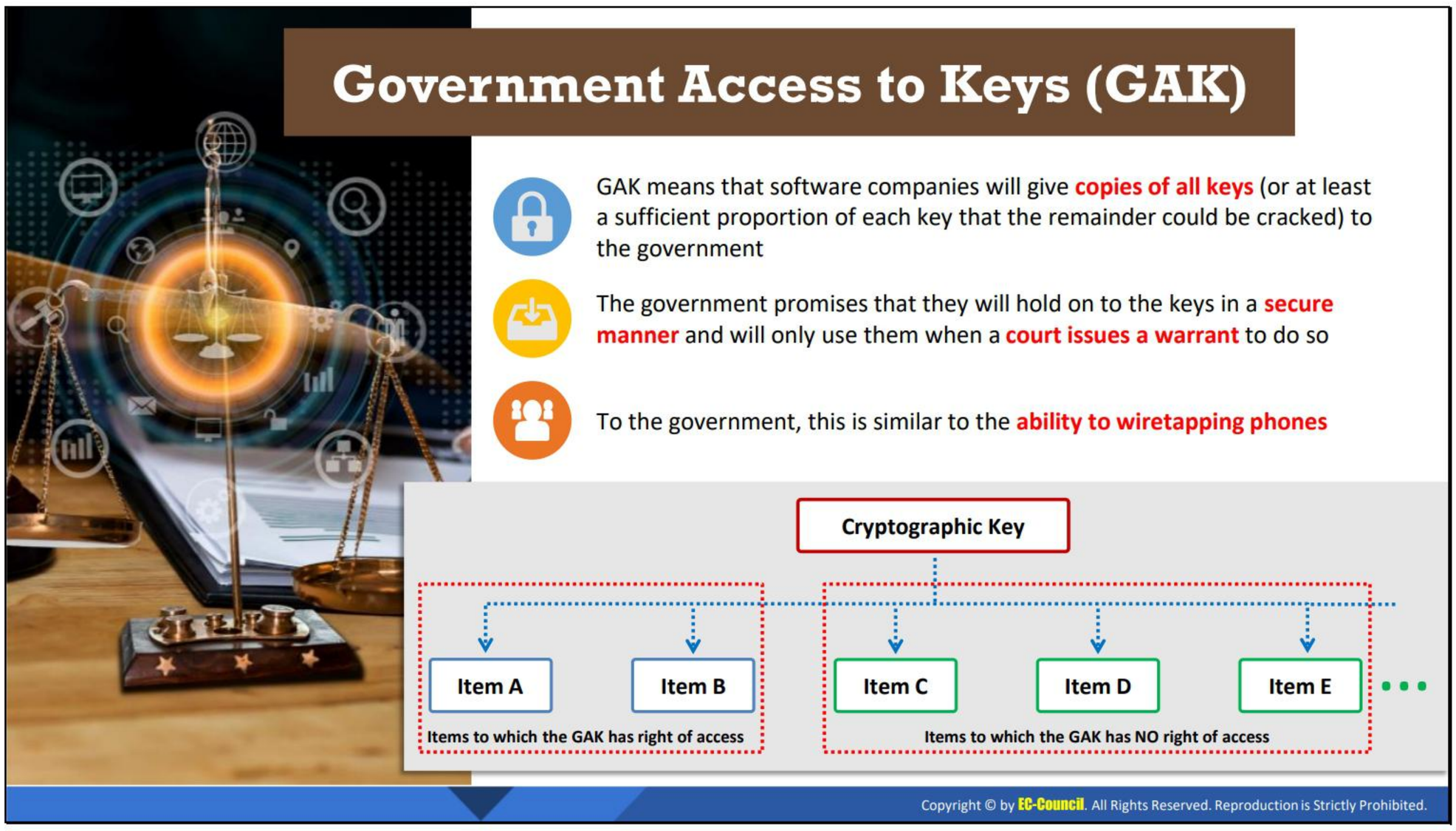
Advantages:

- It is more secure than symmetric encryption.
- There is no need to distribute the keys.

Disadvantages:

- It takes a longer processing time than symmetric encryption since it involves various combinations of secret keys and public keys.
- Various complex algorithms involved in the process of asymmetric encryption also increase the time taken to implement it.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Government Access to Keys (GAK)

Government Access to Keys (GAK) refers to the statutory obligation of individuals and organizations to disclose their cryptographic keys to government agencies. It means that software companies will give copies of all keys (or at least enough of the key such that the remainder can be cracked) to the government. Law enforcement agencies around the world acquire and use these cryptographic keys to monitor suspicious communication and collect evidence of cybercrimes in the interests of national security. The government promises that it will hold on to the keys in a secure manner and only use them when a court issues a warrant to do so. To the government, this issue is similar to the ability to wiretap phones.

Government agencies often use key escrow for uninterrupted access to keys. Key escrow is a key exchange arrangement in which essential cryptographic keys are stored with a third party in escrow. The third party can use or allow others to use the encryption keys under certain predefined circumstances. The third party, with regard to GAK, is generally a government agency that may use the encryption keys to decipher digital evidence under authorization or a warrant from a court of law. However, there is growing concern about the privacy and security of cryptographic keys and information. Government agencies are responsible for protecting these keys. Such agencies generally use a single key to protect other keys, which is not a good idea, as revealing a single key could expose the other keys.

These agencies are not aware of how confidential the information protected by the keys is, which makes it difficult to judge how much protection is required. In cases where seized keys also protect other information that these agencies have no right to access, the consequences of key revelation cannot be determined, because government agencies are not aware of the information that the keys protect. In such cases, the key owner is liable for the consequences of key revelation. Before owners hand over their keys to government agencies, they need to be

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

assured that the government agencies will protect these keys according to a sufficiently strong standard to protect their interests.

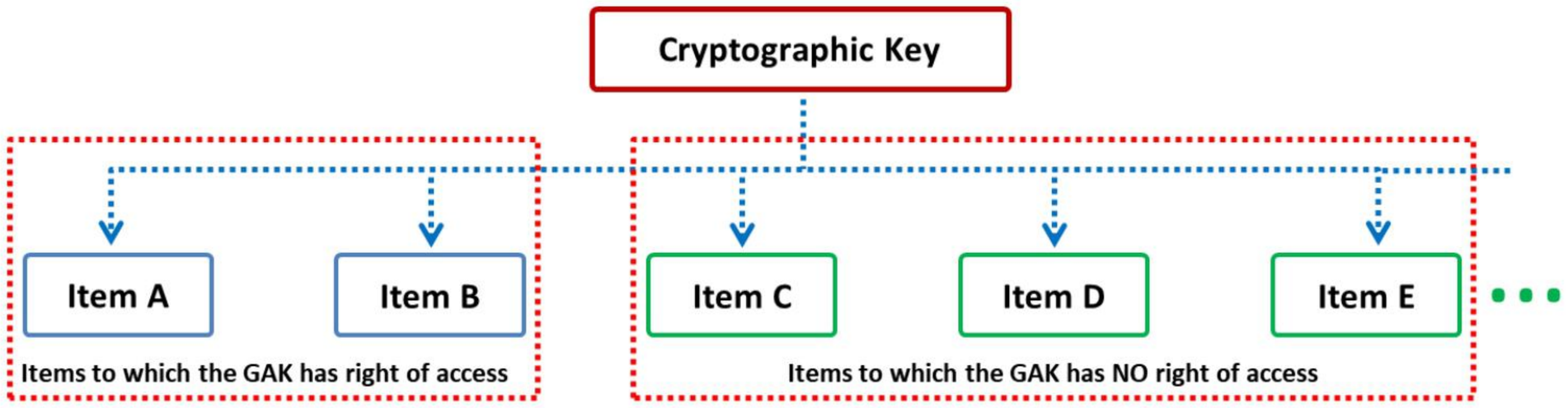
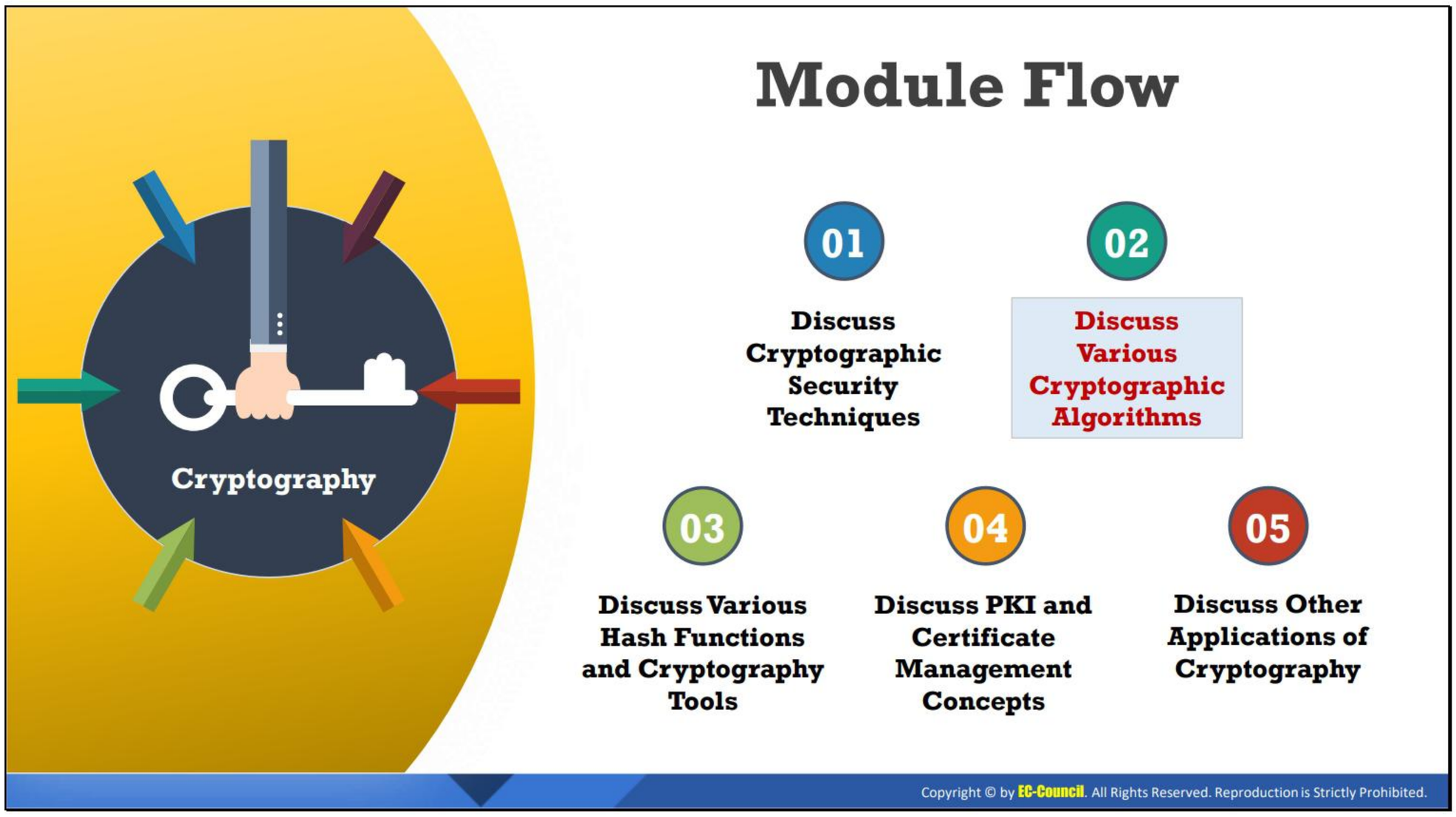


Figure 14.5: Illustration of GAK

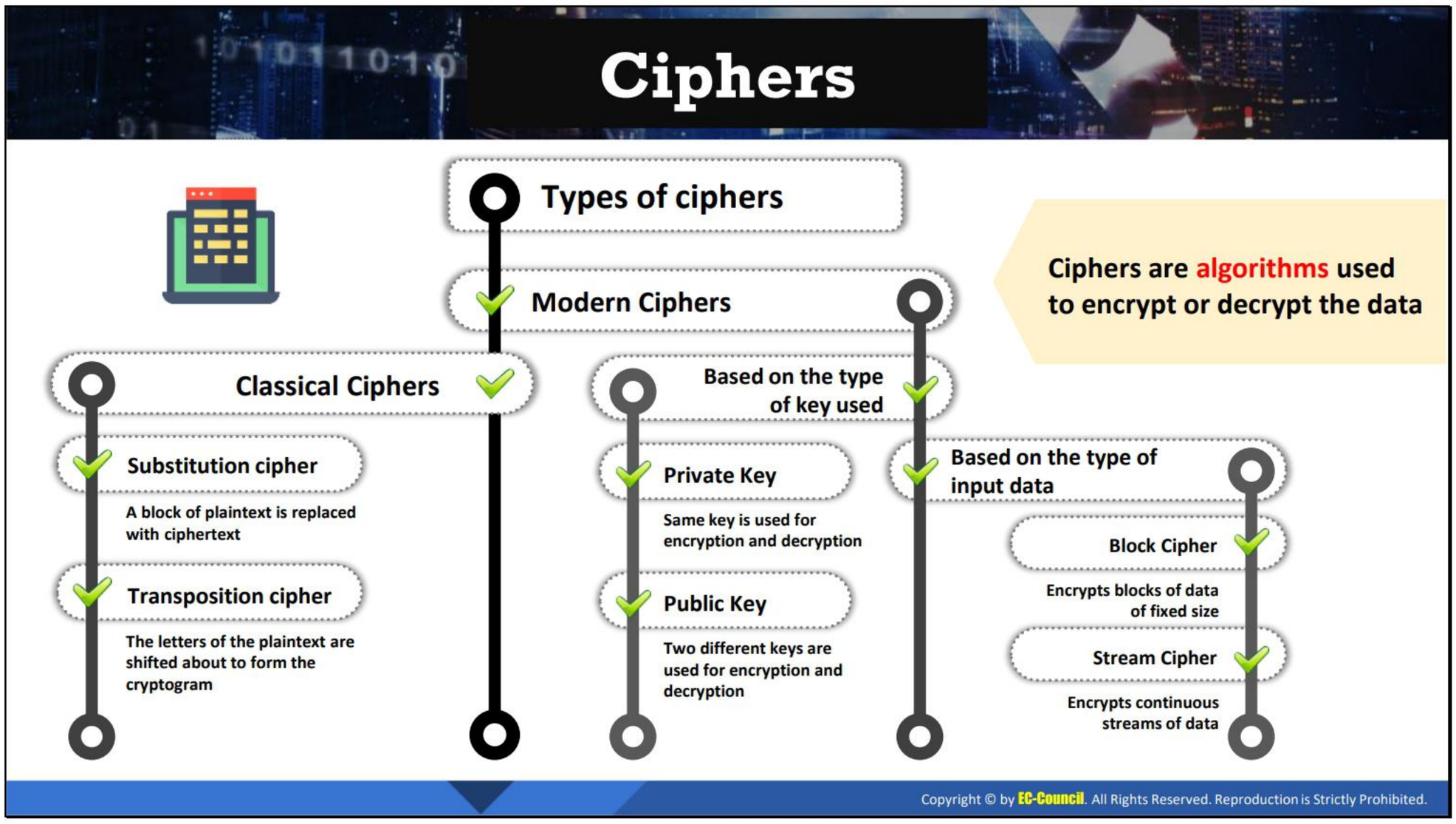
Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Discuss Various Cryptographic Algorithms

Encryption is the process of converting readable plaintext into an unreadable ciphertext using a set of complex algorithms that transform the data into blocks or streams of random alphanumeric characters. This section deals with ciphers and various encryption algorithms such as DES, AES, RC4, RC5, RC6, DSA, RSA, etc.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Ciphers

In cryptography, a cipher is an algorithm (a series of well-defined steps) for performing encryption and decryption. Encipherment is the process of converting plaintext into a cipher or code; the reverse process is called decipherment. A message encrypted using a cipher is rendered unreadable unless its recipient knows the secret key required to decrypt it. Communication technologies (e.g., Internet, cell phones) rely on ciphers to maintain both security and privacy. Cipher algorithms may be open-source (the algorithmic process is in the public domain while the key is selected by a user and is private) or closed-source (the process is developed for use in specific domains, such as the military, and the algorithm itself is not in the public domain). Furthermore, ciphers may be free for public use or licensed.

Types of ciphers

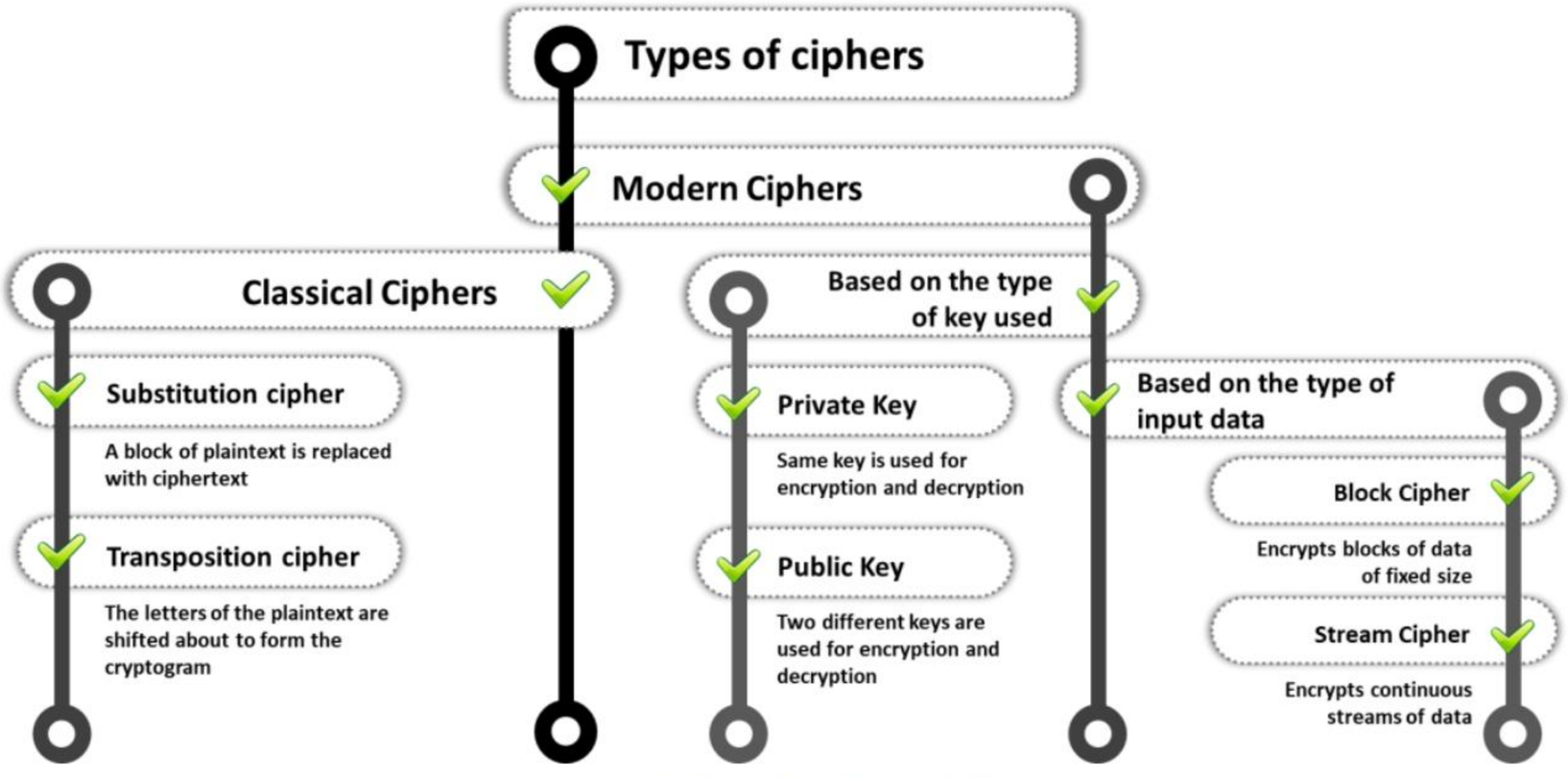


Figure 14.6: Classification of ciphers

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Ciphers are of two main types: classical and modern.

- **Classical Ciphers**

  Classical ciphers are the most basic type of ciphers, which operate on letters of the alphabet (A–Z). These ciphers are generally implemented either by hand or with simple mechanical devices. Because these ciphers are easily deciphered, they are generally unreliable.

  Types of classical ciphers

  - **Substitution cipher**: The user replaces units of plaintext with ciphertext according to a regular system. The units may be single letters, pairs of letters, or combinations of them, and so on. The recipient performs inverse substitution to decipher the text. Examples include the Beale cipher, autokey cipher, Gronsfeld cipher, and Hill cipher.

    For example, "**HELLO WORLD**" can be encrypted as "**PSTER HGFST**" (i.e., H=P, E=S, etc.).

  - **Transposition cipher**: Here, letters in the plaintext are rearranged according to a regular system to produce the ciphertext. For example, "**CRYPTOGRAPHY**" when encrypted becomes "**AOYCRGPTYRHP**." Examples include the rail fence cipher, route cipher, and Myszkowski transposition.

- **Modern Ciphers**

  Modern ciphers are designed to withstand a wide range of attacks. They provide message secrecy, integrity, and authentication of the sender. A user can calculate a modern cipher using a one-way mathematical function that is capable of factoring large prime numbers.

  Types of Modern ciphers

  - **Based on the type of key used**
    - **Symmetric-key algorithms (Private-key cryptography)**: Use the same key for encryption and decryption.
    - **Asymmetric-key algorithms (Public-key cryptography)**: Use two different keys for encryption and decryption.
  - **Based on the type of input data**
    - **Block cipher**: Deterministic algorithms operating on a block (a group of bits) of fixed size with an unvarying transformation specified by a symmetric key. Most modern ciphers are block ciphers. They are widely used to encrypt bulk data. Examples include DES, AES, IDEA, etc. When the block size is less than that used by the cipher, padding is employed to achieve a fixed block size.
    - **Stream cipher**: Symmetric-key ciphers are plaintext digits combined with a key stream (pseudorandom cipher digit stream). Here, the user applies the key to each bit, one at a time. Examples include RC4, SEAL, etc.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Data Encryption Standard (DES)

DES is a standard for data encryption that uses a secret key for both encryption and decryption (symmetric cryptosystem). DES uses a 64-bit secret key, of which 56 bits are generated randomly and the other 8 bits are used for error detection. It uses a data encryption algorithm (DEA), a secret key block cipher employing a 56-bit key operating on 64-bit blocks. DES is the archetypal block cipher—an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length. The design of DES allows users to implement it in hardware and use it for single-user encryption, such as to store files on a hard disk in encrypted form.

DES provides 72 quadrillion or more possible encryption keys and chooses a random key for the encryption of each message. Because of the inherent weakness of DES vis-à-vis today's technologies, some organizations use triple DES (3DES), in which they repeat the process three times for added strength until they can afford to update their equipment to AES capabilities.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

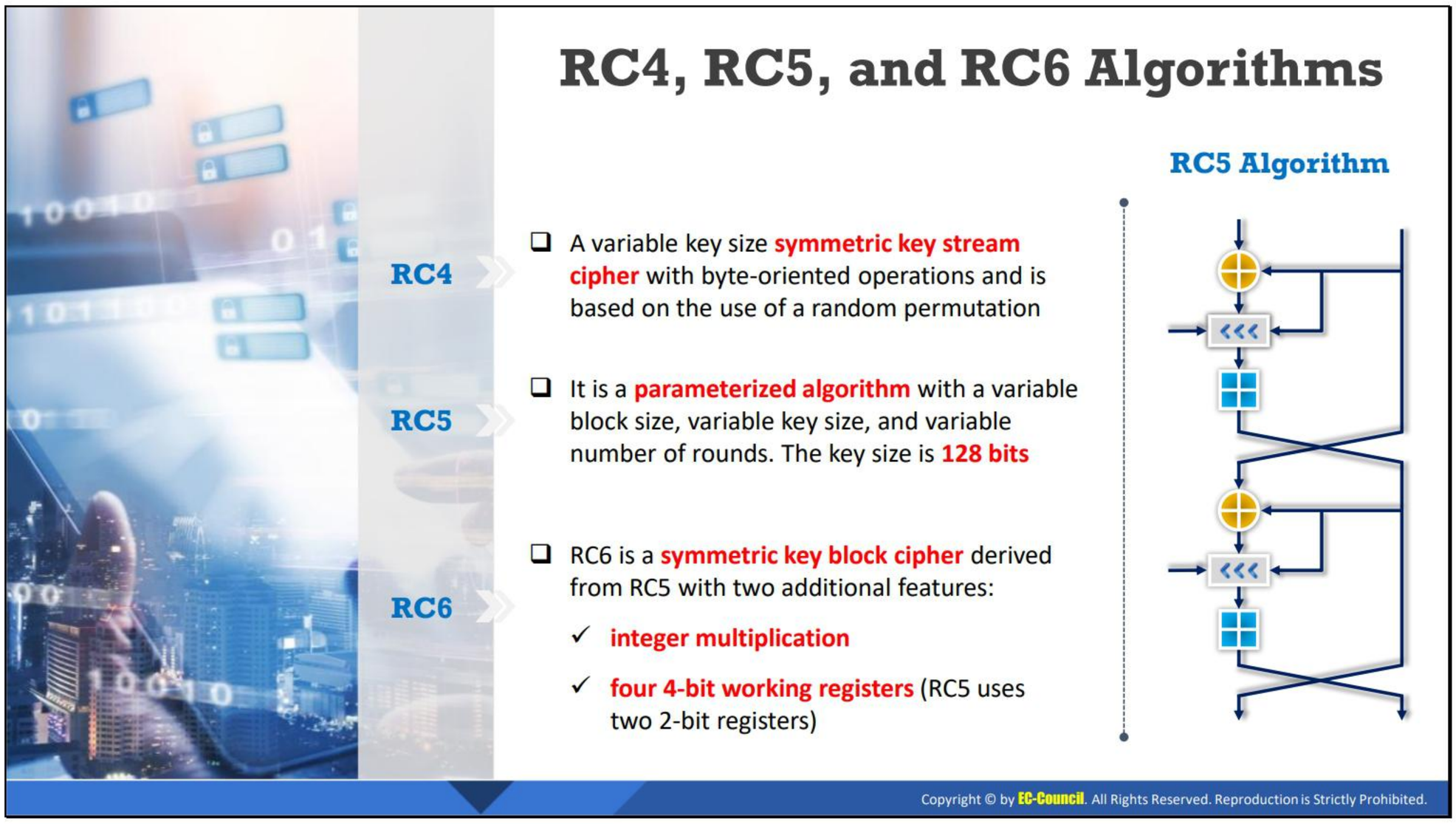


Advanced Encryption Standard (AES)

The advanced encryption standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data. It also helps to encrypt digital information such as telecommunications, financial, and government data. US government agencies have been using it to secure sensitive but unclassified material.

An AES consists of a symmetric-key algorithm in which the encryption as well as decryption is performed using the same key. It is an iterated block cipher that works by repeating the defined steps multiple times. It has a 128-bit block size, having key sizes of 128, 192, and 256 bits respectively for AES-128, AES-192, and AES-256. The design of AES makes its use efficient in both software and hardware. It works at multiple network layers simultaneously.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



RC4, RC5, and RC6 Algorithms

Symmetric encryption algorithms developed by RSA Security are discussed below.

- **RC4**

  RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation. According to some analyses, the period of the cipher is likely to be greater than 10,100. Each output byte uses 8 to 16 system operations; thus, the cipher can run fast when used in software. RC4 enables safe communications such as for traffic encryption (which secures websites) and for websites that use the SSL protocol.

- **RC5**

  RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security). The algorithm is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds. The block sizes can be 32, 64, or 128 bits. The range of the rounds can vary from 0 to 255, and the size of the key can vary from 0 to 2,040 bits. This built-in variability can offer flexibility at all levels of security. The routines used in RC5 are key expansion, encryption, and decryption.

  In the key expansion routine, the secret key that a user provides is expanded to fill the key table (the size of which depends on the number of rounds). RC5 uses a key table for both encryption and decryption. The encryption routine has three fundamental operations: integer addition, bitwise XOR, and variable rotation. The intensive use of data-dependent rotation and the combination of different operations make RC5 a secure encryption algorithm.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

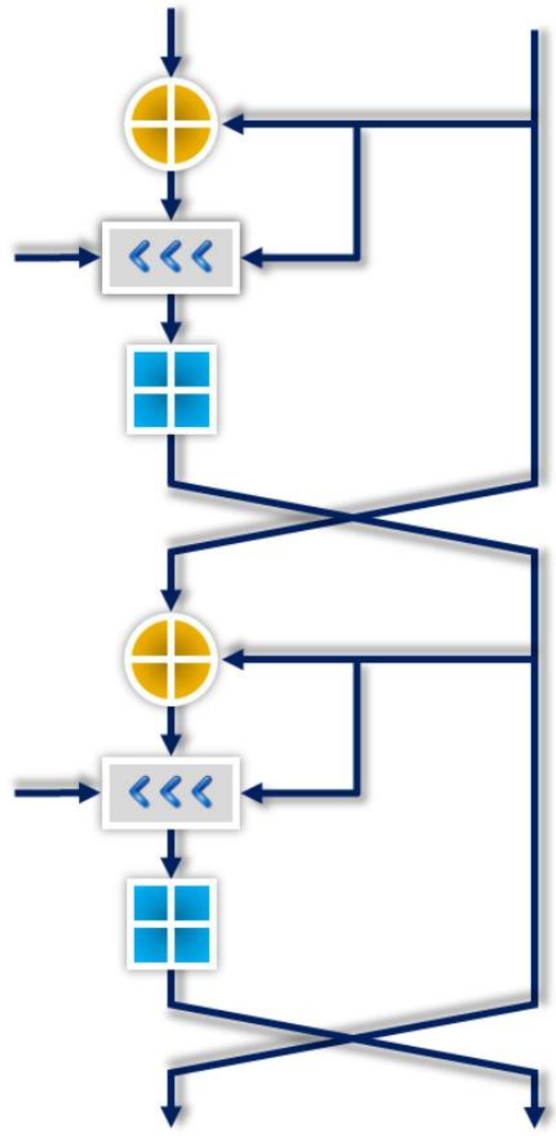
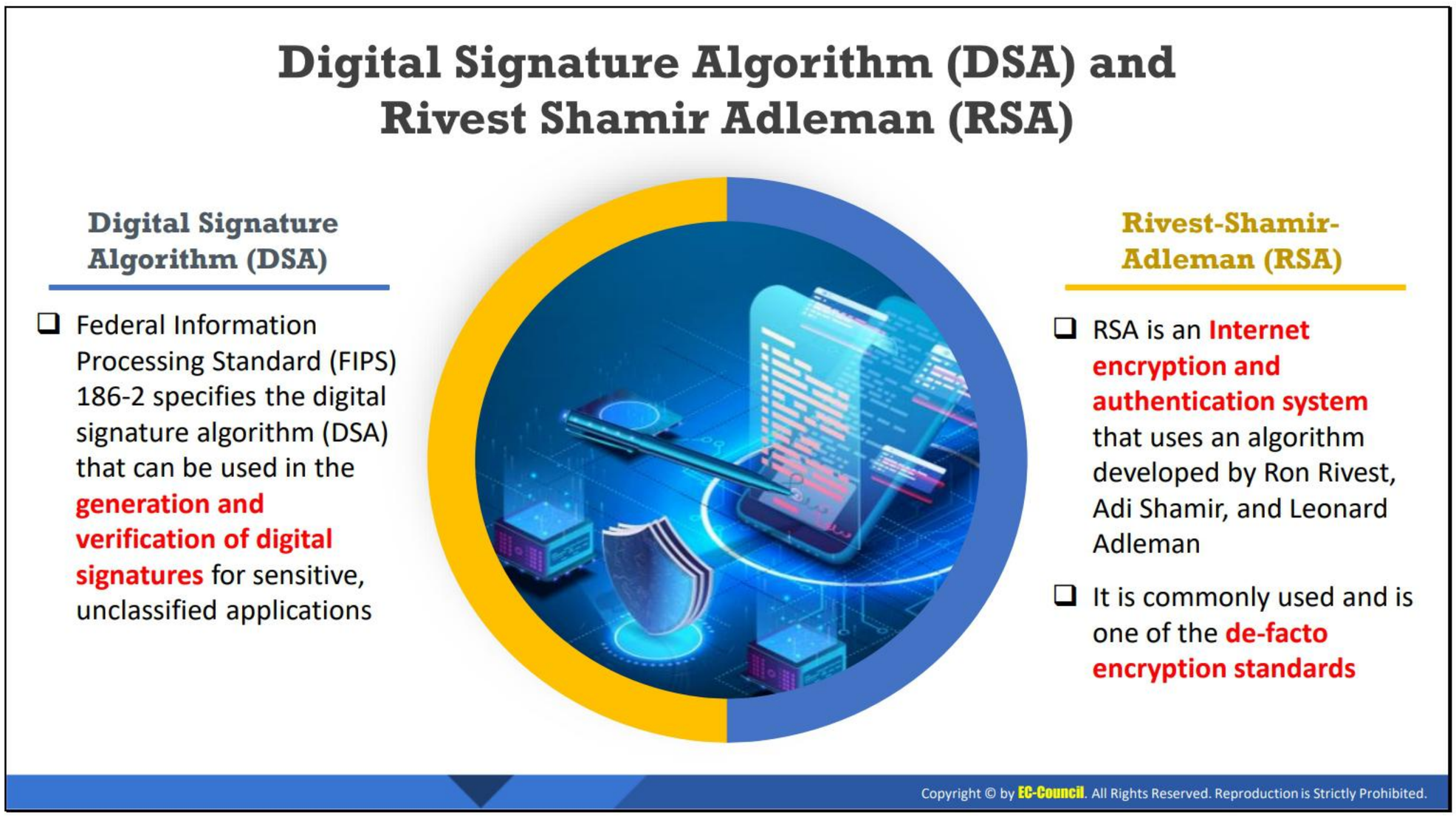


Figure 14.7: Block diagram of the RC5 algorithm

- **RC6**

  RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds. Two features that differentiate RC6 from RC5 are integer multiplication (which is used to increase the diffusion, achieved in fewer rounds with increased speed of the cipher) and the use of four 4-bit working registers rather than two 2-bit registers. RC6 uses four 4-bit registers instead of two 2-bit registers because the block size of the AES is 128 bits.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Digital Signature Algorithm (DSA)

The digital signature algorithm (DSA) is a Federal Information Processing Standard (FIPS) for digital signatures. The NIST proposed the DSA for using it in the digital signature standard (DSS), adopted as a FIPS 186. The DSA helps in the generation and verification of digital signatures in sensitive and unclassified applications. It creates a 320-bit digital signature with a 512–1024 bit security.

A digital signature is a mathematical scheme used for the authenticating digital messages. Computation of a digital signature uses a set of rules (i.e., the DSA) and a set of parameters, using which a user can verify the identity of the signatory and the integrity of the data.

Processes involved in DSA:

- **Signature generation process**: A private key is used to sign the digital message.
- **Signature verification process**: A public key is used to verify whether the given digital signature is genuine.

DSA is a public-key cryptosystem as it involves the use of both private and public keys.

Benefits of DSA:

- Less chances of forgery than in the case of a written signature
- Quick and easy method for business transactions
- Fake currency problem can be drastically reduced

Rivest Shamir Adleman (RSA)

Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication. RSA uses a modular arithmetic and elementary number

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

theory for performing computations using two large prime numbers. The RSA system is widely used in a variety of products, platforms, and industries and is one of the de-facto encryption standards. Companies such as Microsoft, Apple, Sun, and Novell incorporate the RSA algorithm in their operating systems. RSA can also be found on hardware secured telephones, ethernet network cards, and smart cards.

The following sequence is an example of how cryptography uses the RSA algorithm in a practical exchange:

- The sender encrypts a message using a randomly chosen DES symmetric key. DES is a relatively insecure symmetric-key system that uses 64-bit encryption (56 bits for key-length, 8 bits for cyclic redundancy check) to encrypt data.
- The sender then looks up the recipient's public key and uses it to encrypt the DES key using the RSA system.
- The sender transmits an RSA digital envelope, consisting of a DES-encrypted message and an RSA-encrypted DES key, to the recipient.
- The recipient decrypts the DES key using his/her RSA private key and uses it to decrypt the message itself.

This system combines the high speed of DES with the key management convenience of the RSA system.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Other Encryption Techniques and Technologies

Elliptic Curve Cryptography

- ECC is a modern public-key cryptography developed to **avoid larger cryptographic key usage**
- The cryptosystem in ECC depends on **number theory** and **mathematical elliptic curves** (algebraic structure)
- ECC was proposed as a **replacement for the RSA algorithm** to reduce the usage of key size

Quantum Cryptography

- Quantum cryptography is based on quantum mechanics, such as **quantum key distribution** (QKD)
- Data is encrypted by a **sequence of photons** with a spinning trait while travelling from one end to another
- Attackers can eavesdrop but cannot manipulate the data because the photons are transferred through arbitrary filters

Homomorphic Encryption

- Homomorphic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated
- Encryption and decryption are done by the **same key holder**
- It enables a user to encrypt the confidential data and out-source it to the **enterprise via cloud services** to process the given data

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Other Encryption Techniques and Technologies (Cont'd)

Hardware-Based Encryption

Hardware-based encryption **uses computer hardware** for assisting or replacing the software when the data encryption process is underway

These devices are also capable of **storing encryption keys** and other **sensitive information** in secured areas of RAM or other nonvolatile storage devices

Types of hardware encryption devices

TPM	Trusted platform module (TPM) is a **crypto-processor** or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations
HSM	Hardware security module (HSM) is an additional external security device that is used in a system for **crypto-processing** and can be used for managing, generating, and securely storing cryptographic keys
USB Encryption	USB encryption is an additional feature for USB storage devices that offers **onboard encryption** services
Hard Drive Encryption	Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of **encryption options**

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Other Encryption Techniques and Technologies (Cont'd)

Post-quantum Cryptography

- Post-quantum cryptography is an advanced cryptographic algorithm designed to **protect security systems** from attacks initiated on both conventional and quantum computers
- It is intended to provide **secure wide-range communication**, secure secret-key processing, public-key-based signatures, and public-key-based encryption



Lightweight Cryptography

- Most current cryptographic algorithms are suitable for servers and desktops, but lightweight cryptographic algorithms are aimed at **low-complexity applications** such as RFID tags, sensor-based applications, and other IoT-based applications
- The main objective of developing lightweight cryptography is to use **less power** and **less resources** without compromising security

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Other Encryption Techniques and Technologies

- **Elliptic Curve Cryptography (ECC)**

  ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage. The asymmetric cryptosystem depends on number theory and mathematical elliptic curves (algebraic structure) to generate short, quick, and robust cryptographic keys. RSA is an incumbent public-key algorithm, but its key size is large. The speed of the encryption always depends on the key size: a smaller key length allows faster encryption. To minimize the key size, elliptic curve cryptography has been proposed as a replacement for the RSA algorithm.

  The operational key sizes of both algorithms to achieve similar goals are listed below:

ECC Key size	RSA Key Size
160	1024
224	2048
256	3072
384	7680
512	15360

Table 14.1: Comparison of ECC and RSA key size

While RSA uses a key size of 1024 to encrypt the data, ECC provides equal security with a comparatively smaller key size of 160. For high-level computing, RSA uses a key size of 7680 to implement security, whereas ECC can provide the same level of security with a key size of 384.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Quantum Cryptography**

  As the world is increasingly adopting online information sharing, cryptosystems are witnessing a sharp increase in security attacks. Since mathematical encryption uses binary digits (0 and 1), it can be easily eavesdropped on or manipulated using various techniques. Hence, quantum cryptography has been introduced to protect data from theft midway (e.g., MITM attacks). This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.

  In quantum cryptography, the data are encrypted by a sequence of photons that have a spinning trait while traveling from one end to another end. These photons keep changing their shapes during their course through filters: vertical, horizontal, forward slash, and backslash. Here, vertical and backslash spins imply "ones," while horizontal and forward slash spins imply "zeros."

  - Horizontal (–): 0
  - Vertical (|): 1
  - Backslash (/): 1
  - Forward slash (\): 0

  Attackers can eavesdrop on but cannot manipulate the data because the photons are transferred through arbitrary filters. To breach this mechanism, attackers have to know the exact shape of the photons; if they fail to choose the right transmission, the photon polarization is distorted and the receiver detects an error indicating the eavesdrop.

- **Homomorphic Encryption**

  Homomorphic encryption differs from conventional encryption mechanisms, where math operations are performed to encrypt the plaintext. Homographic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated. In this technique, encryption and decryption are performed by the same key holder. The homomorphic mechanism enables the user/sender to encrypt the confidential data and out-source it to an enterprise via cloud services to process the given data.

  How homomorphic encryption differs from other encryption mechanisms:

  In private key encryption:

  - Only keyholders can generate and decrypt ciphertexts using similar keys.

  In public key encryption:

  - Only the public keyholder generates the ciphertext and the secret keyholder decrypts the ciphertext.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

In homomorphic encryption:

- The keyholder can generate the ciphertext and anyone can alter the ciphertext, but only the keyholder again can decrypt the data.

The reason for using this cryptography is that an untrusted entity can manipulate the data. Hence, this mechanism allows the sender himself/herself to encrypt and decrypt the data, allowing anyone to perform mathematical operations on the ciphertext with respect to the rules applied by the sender.

- **Hardware-Based Encryption**

Hardware-based encryption is a technique that uses computer hardware for assisting or replacing the software when the data encryption process is being performed. Devices that offer encryption techniques can be considered as hardware-based encryption devices. In the implementation of hardware-based encryption, the cryptography technique workload is transferred to the hardware processors, making the system resources free for performing other functions. These devices can also store encryption keys and other sensitive information in secured areas of RAM or other nonvolatile storage devices such as flash memory.

Hardware encryption devices reduce instruction sets, where only the authorized code can be executed. These devices do not support third-party software, thereby preventing the execution of any malicious programs. Hardware encryption offers many advantages over software encryption, as it can perform rapid processing of algorithm. It provides tamper-resistant key storage and avoids unauthorized code. Some hardware-based encryption devices are wireless access points, Nitrokey, credit card terminals, and network bulk encryptors.

Types of hardware encryption devices

- **TPM**

  Trusted Platform Module (TPM) is a crypto-processor or a chip that is present in the motherboard. It can securely store the encryption keys and perform many cryptographic operations. TPM offers various features such as authenticating platform integrity, providing full disk encryption capabilities, performing password storage, and providing software license protection.

- **HSM**

  A hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing, and it can be used for managing, generating, and securely storing cryptographic keys. HSM offers enhanced encryption computation that is useful for symmetric keys longer than 256 bits. High-performance HSM devices are connected to the network using TCP/IP. Some HSM devices include SafeNet Luna Network HSM, nSheild, Cloud HSM, and Cryptosec Dekaton.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **USB Encryption**

  USB encryption is an additional feature for USB storage devices, which offers onboard encryption services. Encrypted USB devices need an on-device credential system or software- or hardware-based credentials from a computer. USB encryption provides protection against malware distribution over USB and helps in preventing data loss and data leakage. Some hardware USB-encrypted devices include Crypto USB, Kingston Ironkey D300S, and diskAshur Pro 500GB.

- **Hard Drive Encryption**

  Hard drive encryption is a technology whereby the data stored in the hardware can be encrypted using a wide range of encryption options. Hard drive encryption devices cannot use an on-device keyboard or fingerprint reader; instead, they need a TPM or an HSM. These devices can be installed as an internal drive on a computer. Some hard drive encryption devices include military-grade 256-bit AES Hardware Encryption and DiskCypher AES Sata Hard Drive Encryption.

- **Post-quantum Cryptography**

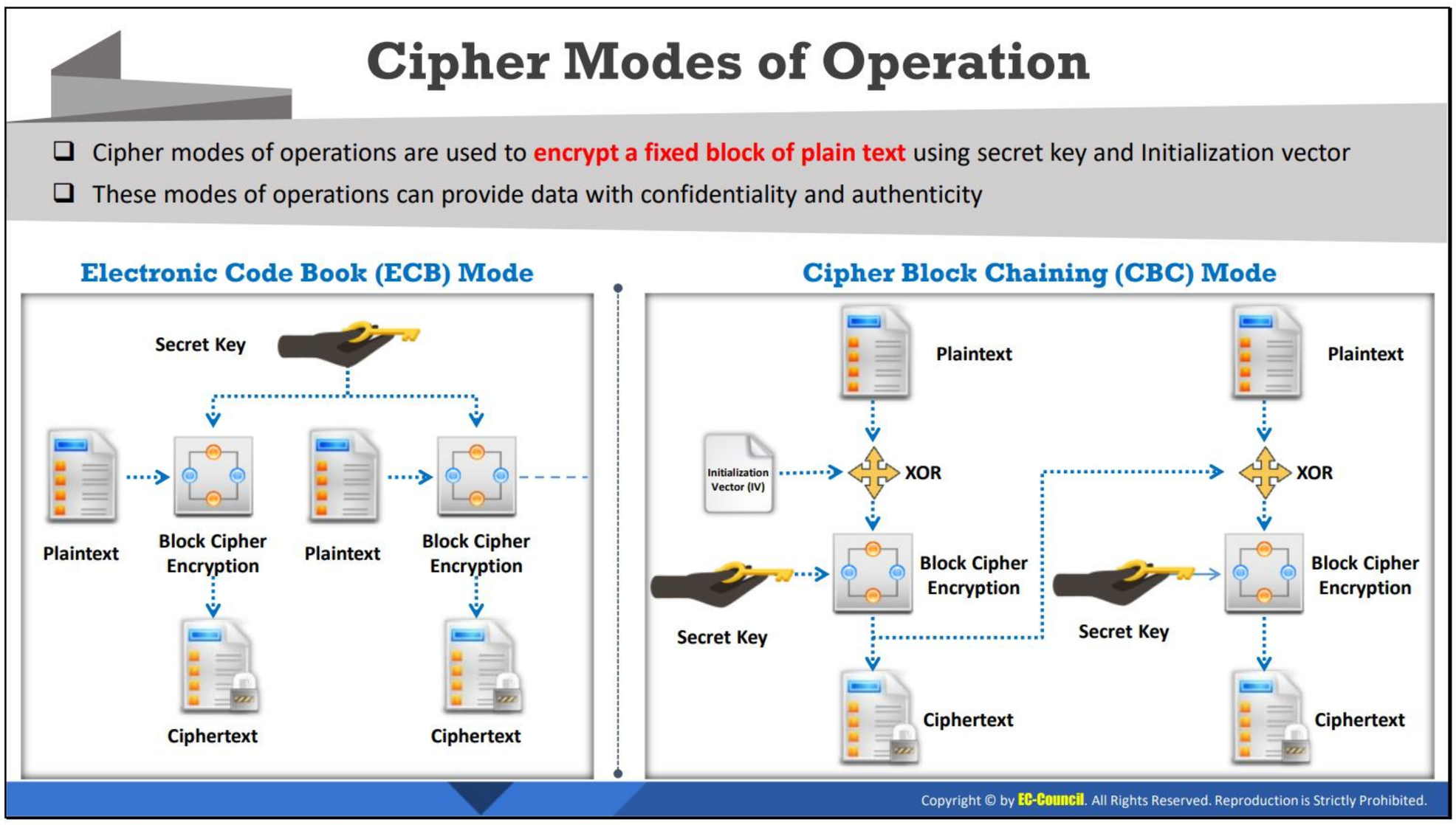
  Post-quantum cryptography is also known as quantum-resistant and quantum-proof cryptography, as it is an advanced cryptographic algorithm (mostly public-key based) designed to protect security systems from attacks initiated on both conventional and quantum computers. It can also work in conjunction with underlying communication protocols and operating networks. Moreover, post-quantum cryptography can serve as a stand-alone encryption algorithm that replaces current vulnerable cryptosystems complying with standard security policies.

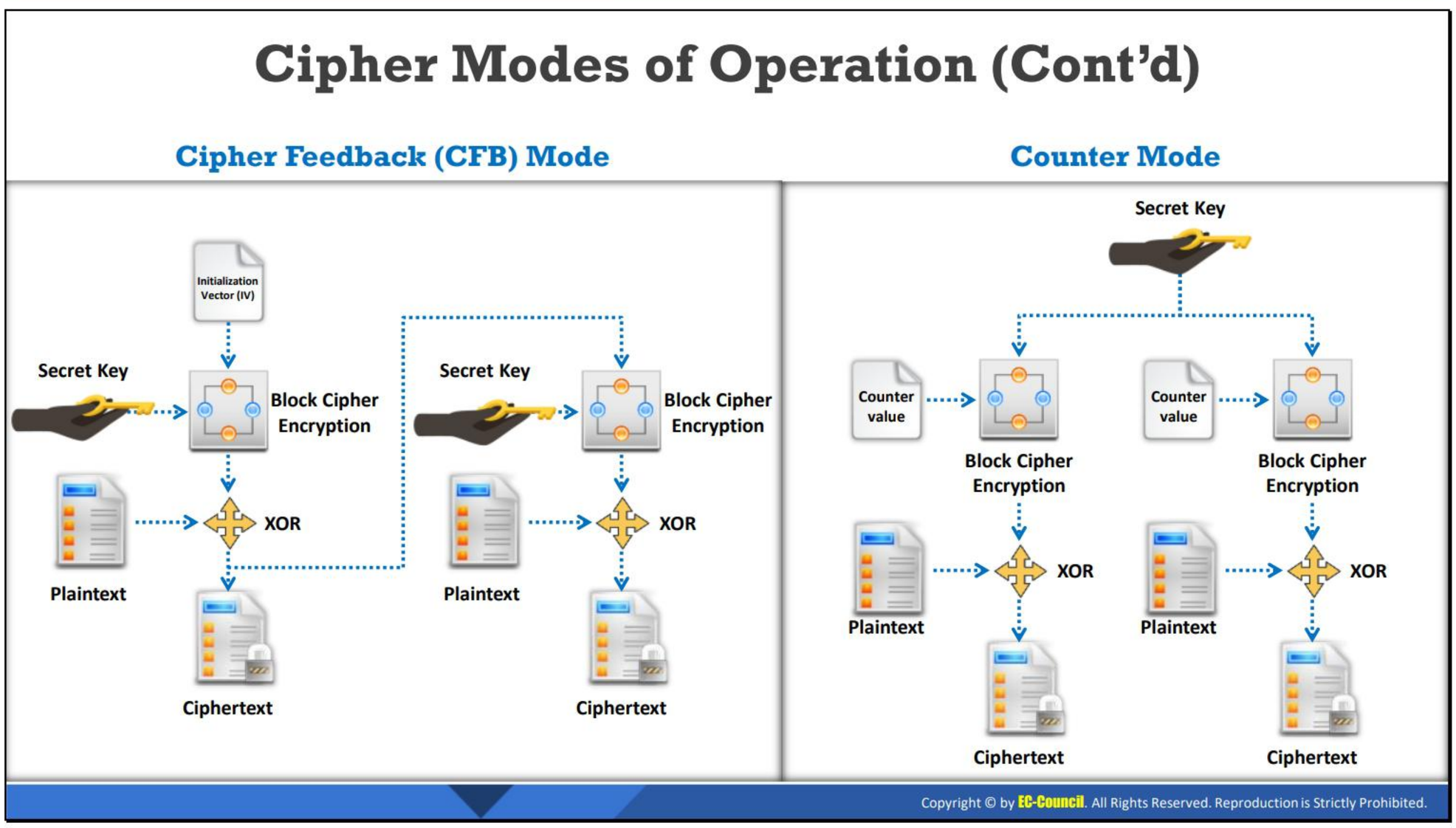
  Post-quantum cryptography is intended to provide secure wide-range communication, secure secret-key processing, public-key-based signatures, and public-key-based encryption for high-end activities such as secure e-voting. The cryptography comprises several low-cost, secure systems and other systems that are usually employed for online communications. Post-quantum cryptography is aimed at preparedness for the age of quantum computing by updating specific algorithms and standards.

- **Lightweight Cryptography**

  A major challenge in current cryptography techniques is its usage in low-powered devices. Researchers are attempting to develop a compact algorithm that is quantum-safe and can be operated effectively on low-powered devices. Most current cryptographic algorithms are suitable for servers and desktops, but lightweight cryptographic algorithms are aimed at low-complexity applications such as RFID tags, sensor-based applications, and other IoT-based applications. The main objective of developing lightweight cryptography is to use less power and less resources without compromising security.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.





Cipher Modes of Operation

Cipher modes of operation, also known as block cipher modes of operation, are used to encrypt a fixed block of plaintext using a secret key and, in some modes, an initialization vector. These modes of operation can ensure the confidentiality and authenticity of data. The client and server exchange an encrypted symmetric key securely to facilitate encryption and decryption. Discussed below are the four block cipher modes of operation that explain how source-side encryption and destination-side decryption work.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Electronic Code Book (ECB) Mode**

  The ECB mode is a straightforward process of encryption and decryption that requires plaintext, a secret key, and a block cipher encryption algorithm. The plaintext is divided into a fixed length of blocks, which is equal to the size of the secret key. In the first stage, the encryption starts by taking the first block of the plaintext, and the secret key is taken as input to the block cipher encryption algorithm; the output is the first block of ciphertext. The process is repeated for all the plaintext blocks.

  On the destination side, decryption is performed in the same manner as generation of the first block of ciphertext. The secret key is taken as input to the block cipher decryption algorithm, which outputs the first block of plaintext. This process is repeated for all the ciphertext blocks. However, this mode has a flaw: if the equally partitioned blocks of plaintext contain the same data, then the output cipher blocks also contain the same ciphertext, providing analysts a chance to predict the plaintext.

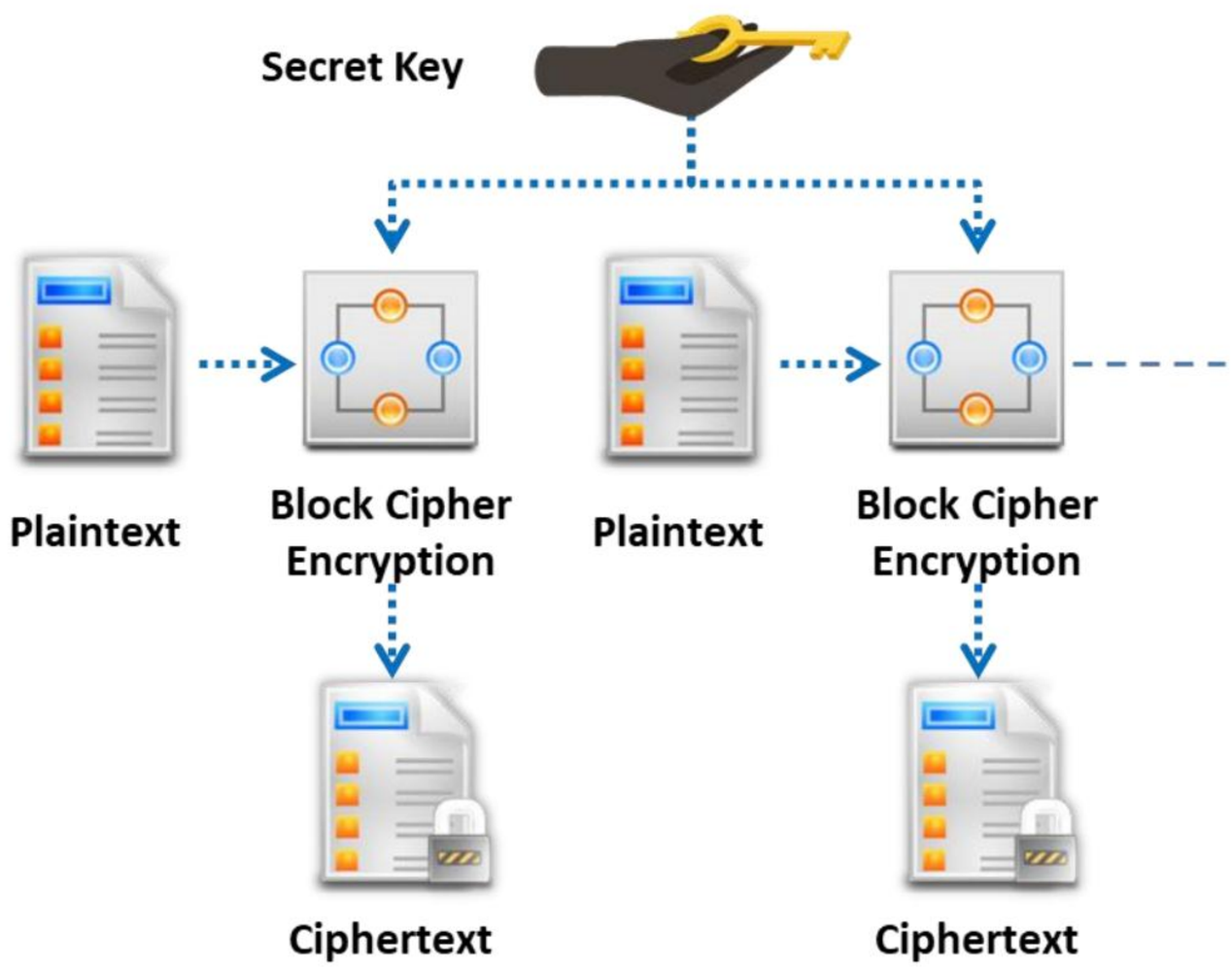


Figure 14.8: Electronic Code Book (ECB) mode encryption

- **Cipher Block Chaining (CBC) Mode**

  The CBC mode is an improvement over ECB that rectifies most of the security flaws in ECB. In the CBC mode, the process of encryption requires an initialization vector and a secret key. First, the plaintext is divided into blocks of the same size. The first block is XOR with the initialization vector (IV), and the resultant is sent as input to the block cipher encryption algorithm, along with the secret key. The output is the first block of ciphertext. This cipher block is used to perform XOR with the next plaintext block; the chain process continues till the last block of plaintext.

  On the destination side, the first block of ciphertext and secret key is sent to the block cipher decryption algorithm, and the result is XOR with the same IV. The output is the first block of plaintext. For the next cipher blocks, in the place of the IV, the previously

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

used cipher block is input to perform XOR; this process continues for the remaining cipher blocks. However, this mode also has a problem: if one generated ciphertext block has an error, it propagates to the subsequent cipher blocks.

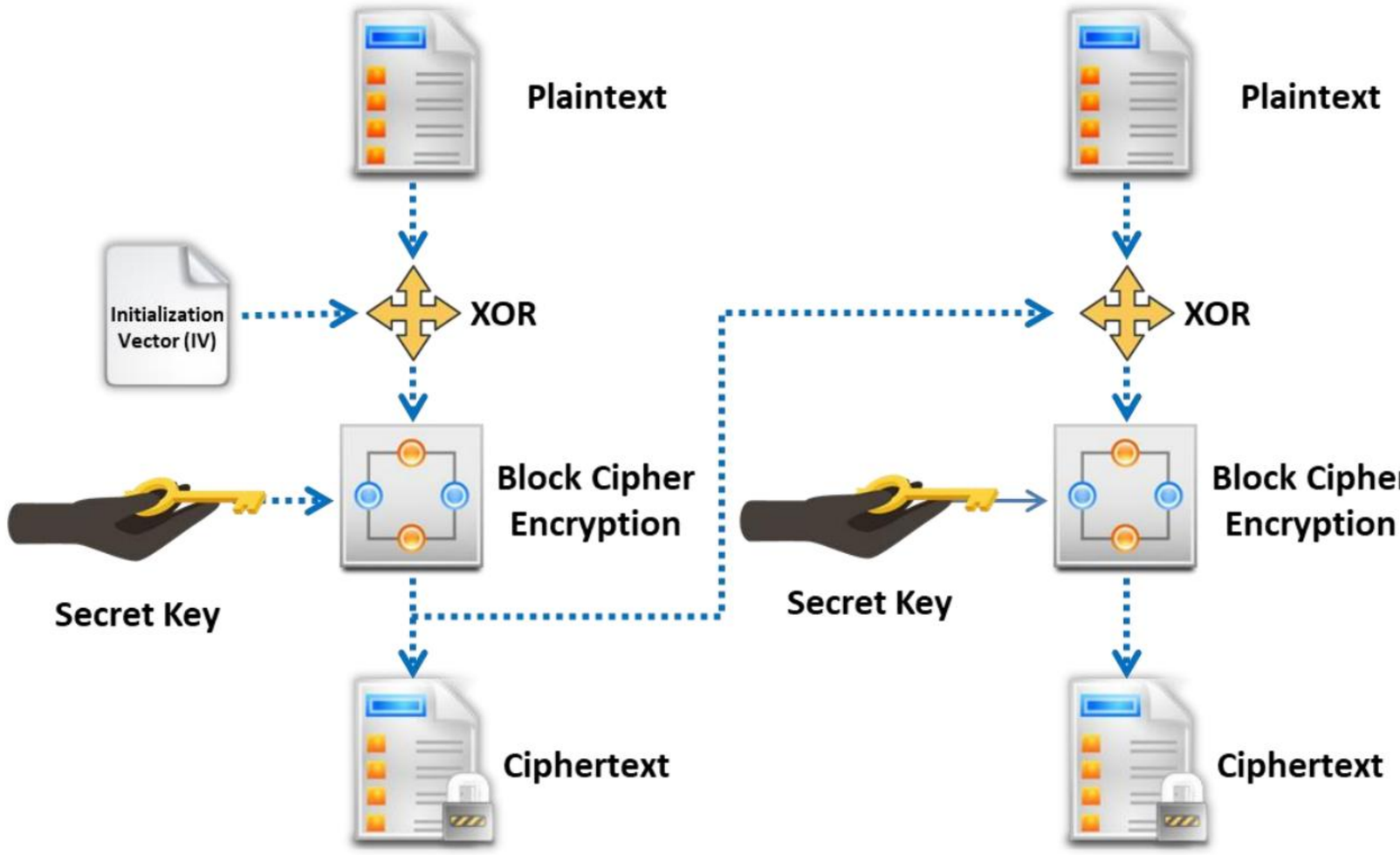


Figure 14.9: Cipher Block Chaining (CBC) mode encryption

- **Cipher Feedback (CFB) Mode**

  In the CFB mode, previously generated ciphertext is used as feedback for the encryption algorithm to encrypt the next plaintext block to ciphertext. First, the initialization vector (IV) is stored in a shift register and sent to the encryption algorithm, along with a secret key. From the result of that encryption, the first S bits are selected, and the XOR operation is performed with a plaintext block of size S. The resultant output is the ciphertext block. For the next encryption block, the previous cipher block is given as the input to the shift register; it shifts S bits to the left, and the process is continued till the end of the plaintext.

  On the destination side, the decryption process is the same till the XOR operation. The XOR operation is performed for the first S bits from the result of the encryption algorithm and the first cipher block, and the output is the first block of plaintext. For the subsequent blocks, the previously used cipher block is taken as the input for the shift register, and the process continues till the last cipher block. The advantage of this mode is that it makes cryptanalysis difficult as it has some data loss because of the use of shift registers.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

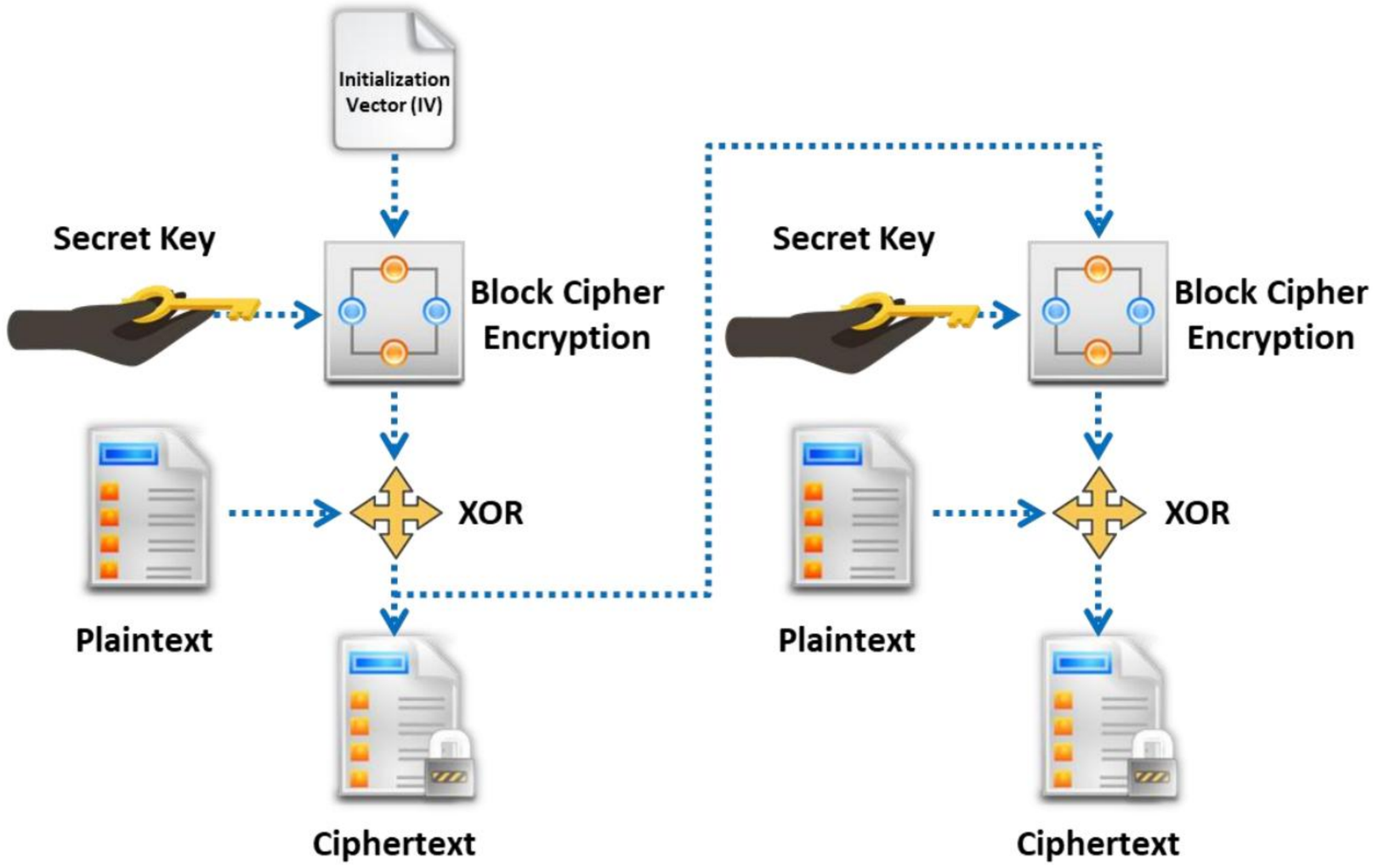


Figure 14.10: Cipher Feedback (CFB) mode encryption

- **Counter Mode**

  The counter mode is a block cipher mode of operation that uses a counter value in the encryption and decryption process. A counter value is initiated and sent as the input to the block cipher encryption algorithm with a secret key, and the result is subjected to the XOR operation with a block of plaintext. The output is the ciphertext block. This process is performed in a sequential manner to encrypt all the other plaintext blocks.

  On the destination side, this mode uses the same counter values and secret keys. The same encryption algorithm is used to encrypt the counter value and secret key, the result is subjected to the XOR operation with the obtained ciphertext block, and the output contains plaintext.

  The counter mode eliminates the problem of error propagation because it does not use previously generated ciphertext in encryption or decryption. The counter mode requires synchronized counter values on both the source and destination sides.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

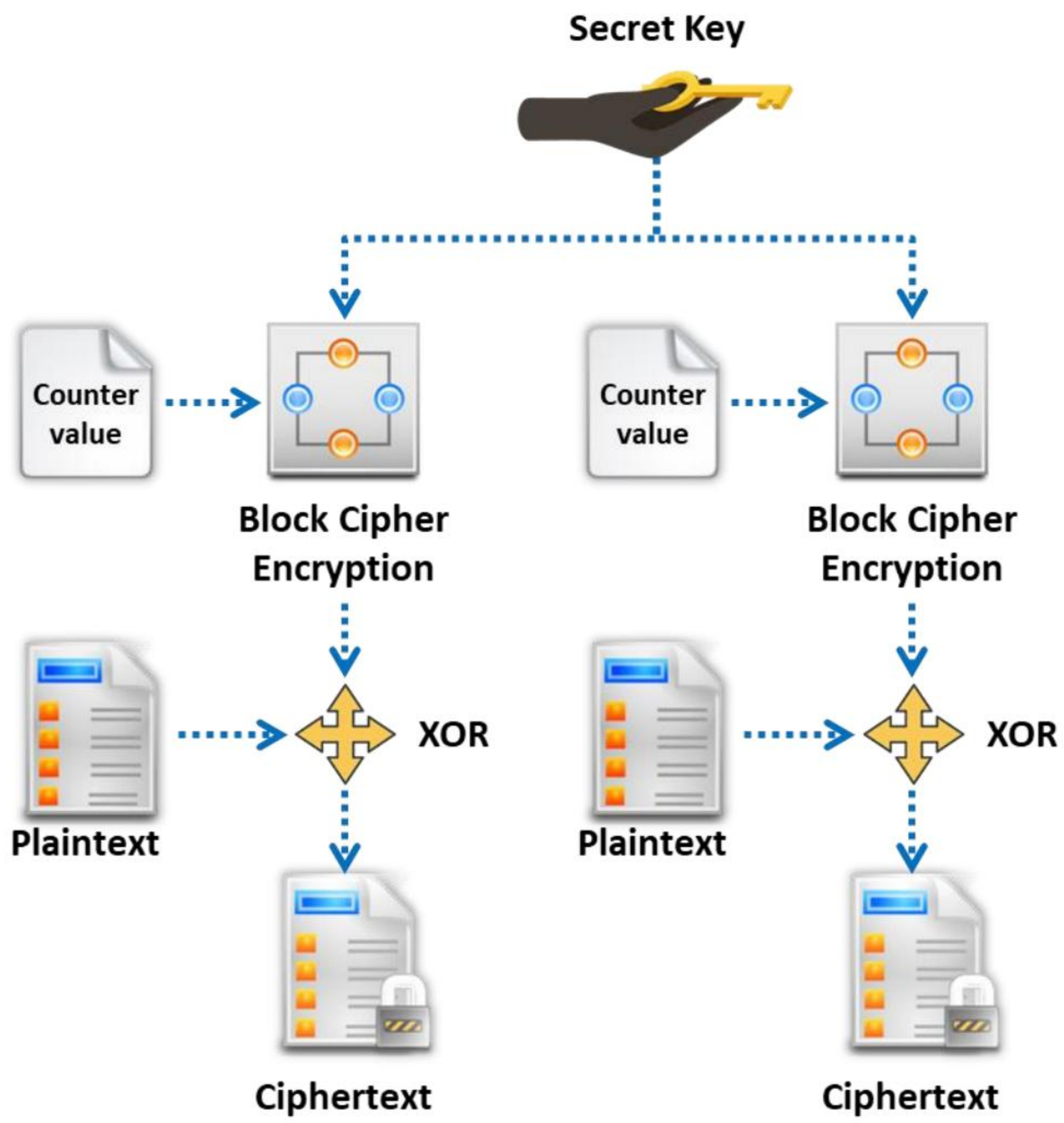
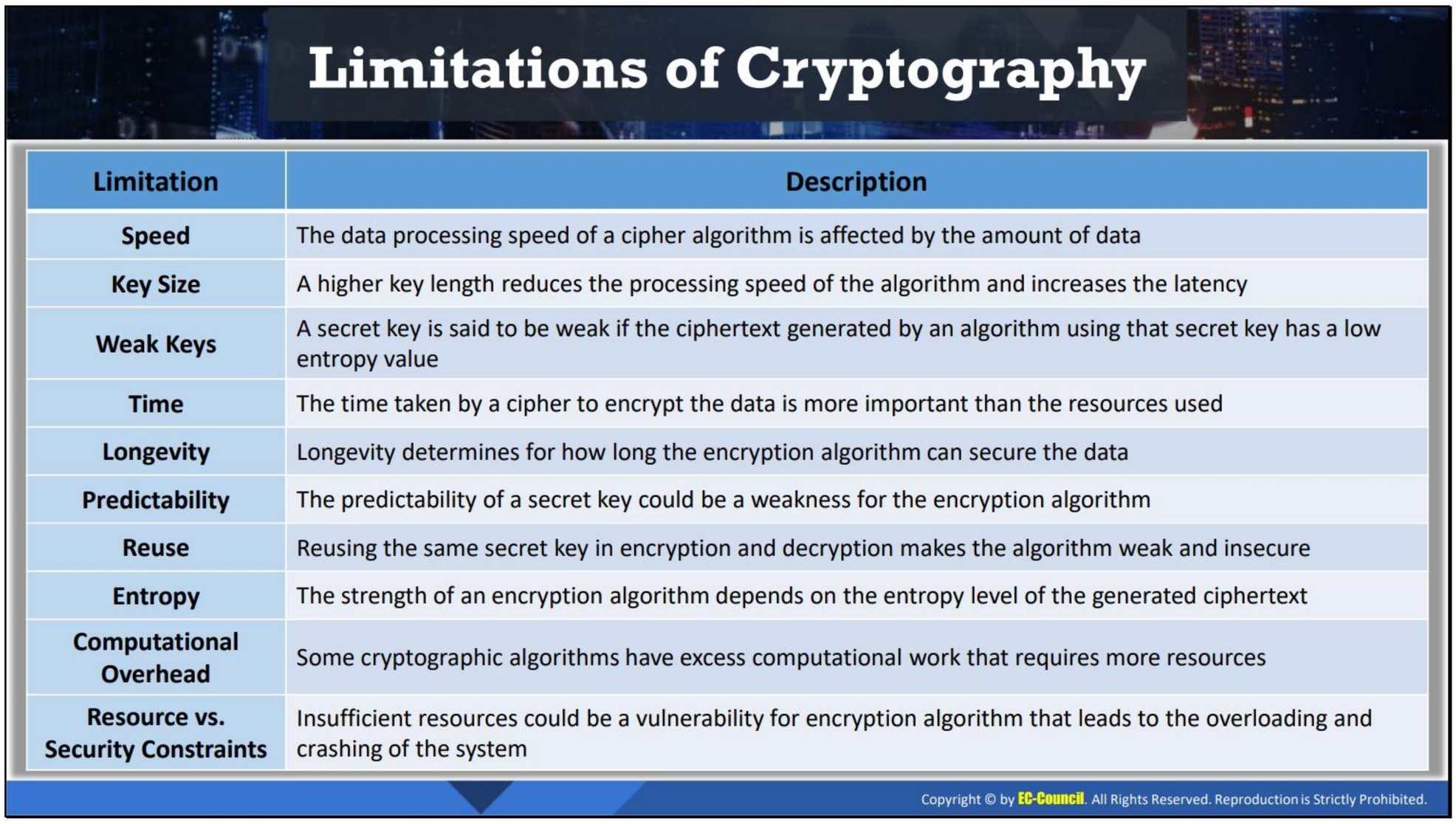


Figure 14.11: Counter mode encryption

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Limitations of Cryptography

The performance of a cipher differs among different algorithms and depends on the environment it is running in. The main factors that affect the performance of a cipher are listed below.

- **Speed**

  Speed determines the amount of data that a cipher algorithm can process in a second. For any symmetric or asymmetric cipher, the data processing speed of a cipher algorithm is affected by the amount of data.

- **Key Size**

  The security of an encryption algorithm depends on the size of the secret key used for encryption. The size of the secret key impacts the performance of a cipher; a higher key length reduces the processing speed of the algorithm and increases the latency.

- **Weak Keys**

  A secret key is said to be weak if the ciphertext generated by an algorithm using that secret key has a low entropy value. Some encryption algorithms are already declared as insecure because of the use of weak secret keys in the encryption process.

- **Time**

  In some cases, the time taken by a cipher to encrypt the data is more important than the resources used. Therefore, the performance of an algorithm is determined by the time taken to complete encryption.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Longevity**

  Longevity is a term used in cryptography to determine for how long the encryption algorithm can secure the data. Cryptanalysts may identify vulnerabilities in a cipher, but this does not mean that the cipher will become vulnerable immediately. It might take a long time to crack the cipher, and the data is safe till then.

- **Predictability**

  The predictability of a secret key could be a weakness for the encryption algorithm. The cryptography algorithm is said to be strong if it has strong randomness while generating a secret key.

- **Reuse**

  Reusing the same secret key in encryption and decryption makes the algorithm weak and insecure. Reusing increases the probability of predicting the key and decreases the entropy value of ciphertext.

- **Entropy**

  Entropy is a term used to measure the disorder level of text. The strength of an encryption algorithm depends on the entropy level of the generated ciphertext. If a single character in the ciphertext resembles the corresponding plaintext, then the algorithm is weak.

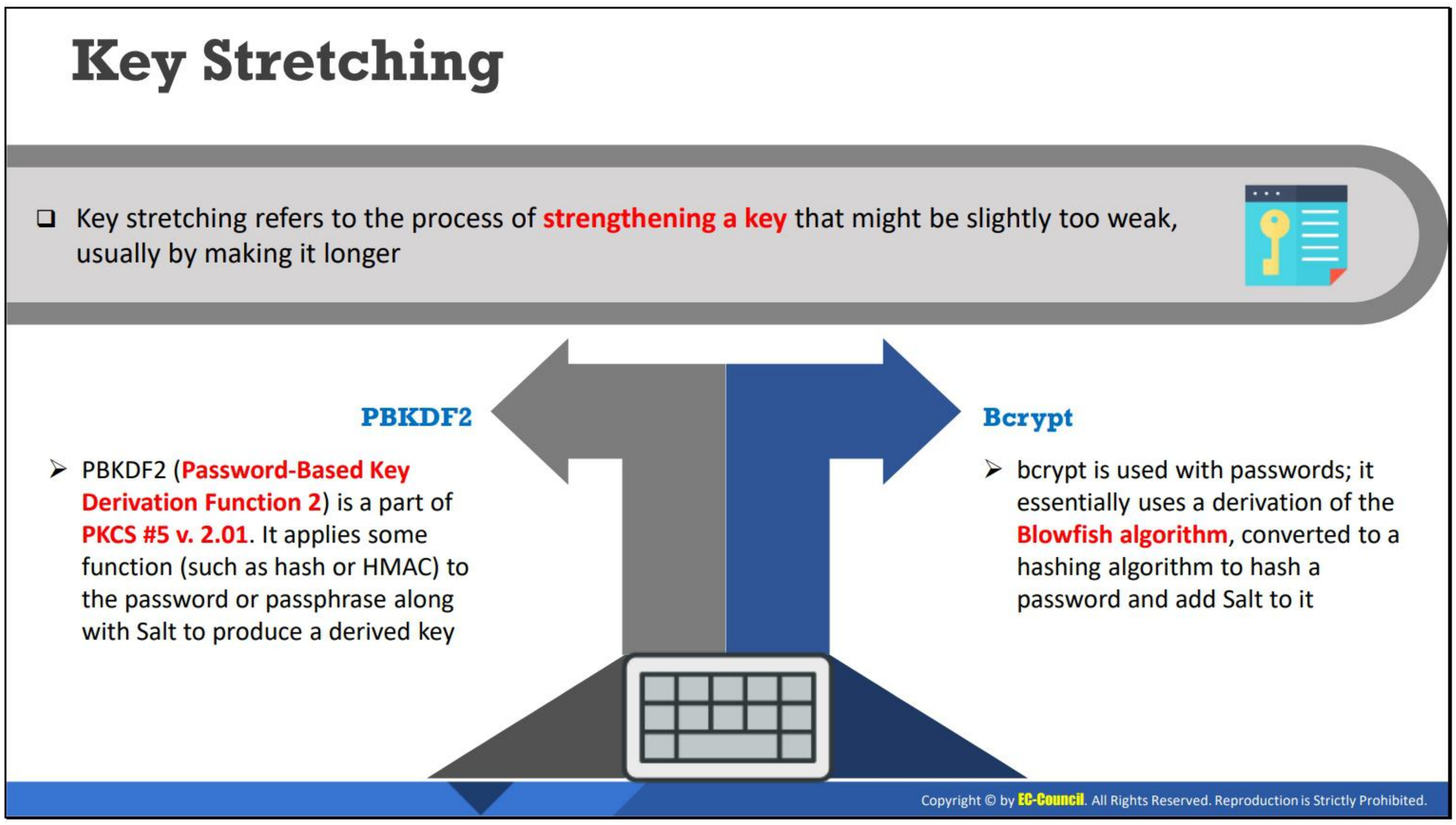
- **Computational Overhead**

  Some cryptographic algorithms have excess computational work that requires more resources such as CPU power and memory space, which is known as computational overhead. Such algorithms are not suitable for a limited-resource environment.

- **Resource vs. Security Constraints**

  The encryption algorithm is secure only when using a lengthy secret key for processing the data. However, a longer secret key requires more resources such as CPU power and storage space. Insufficient resources could be a vulnerability that leads to the overloading and crashing of the system. The resources should balance the requirements of the server for processing.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Key Stretching

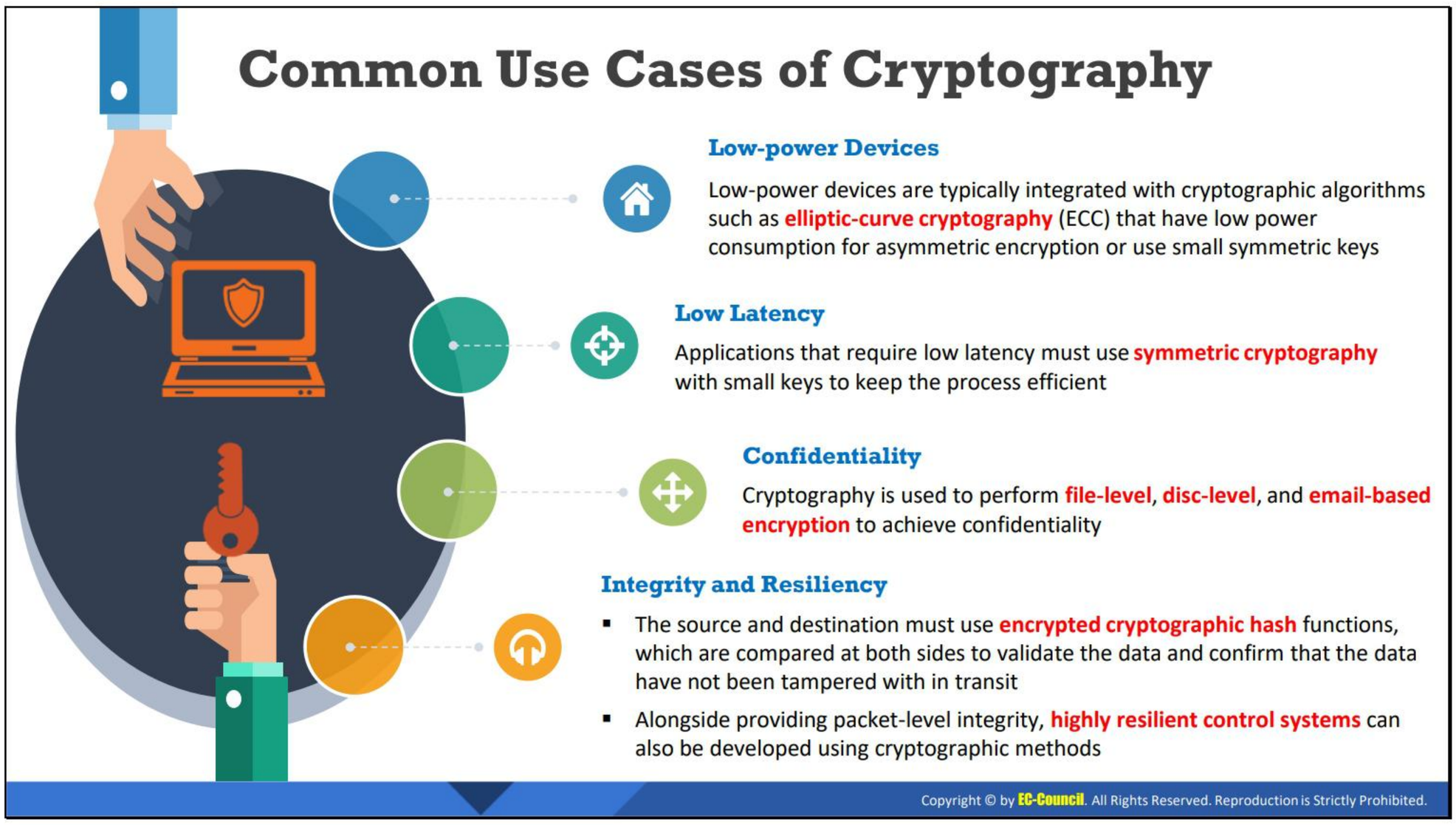
Key stretching refers to processes used to make a weak key stronger, usually by making it longer. This technique helps in defending against brute-force attacks. In general, passwords or passphrases generated by end users are weak and predictable. Hence, key stretching helps security professionals/users to prevent such attacks by strengthening their passwords.

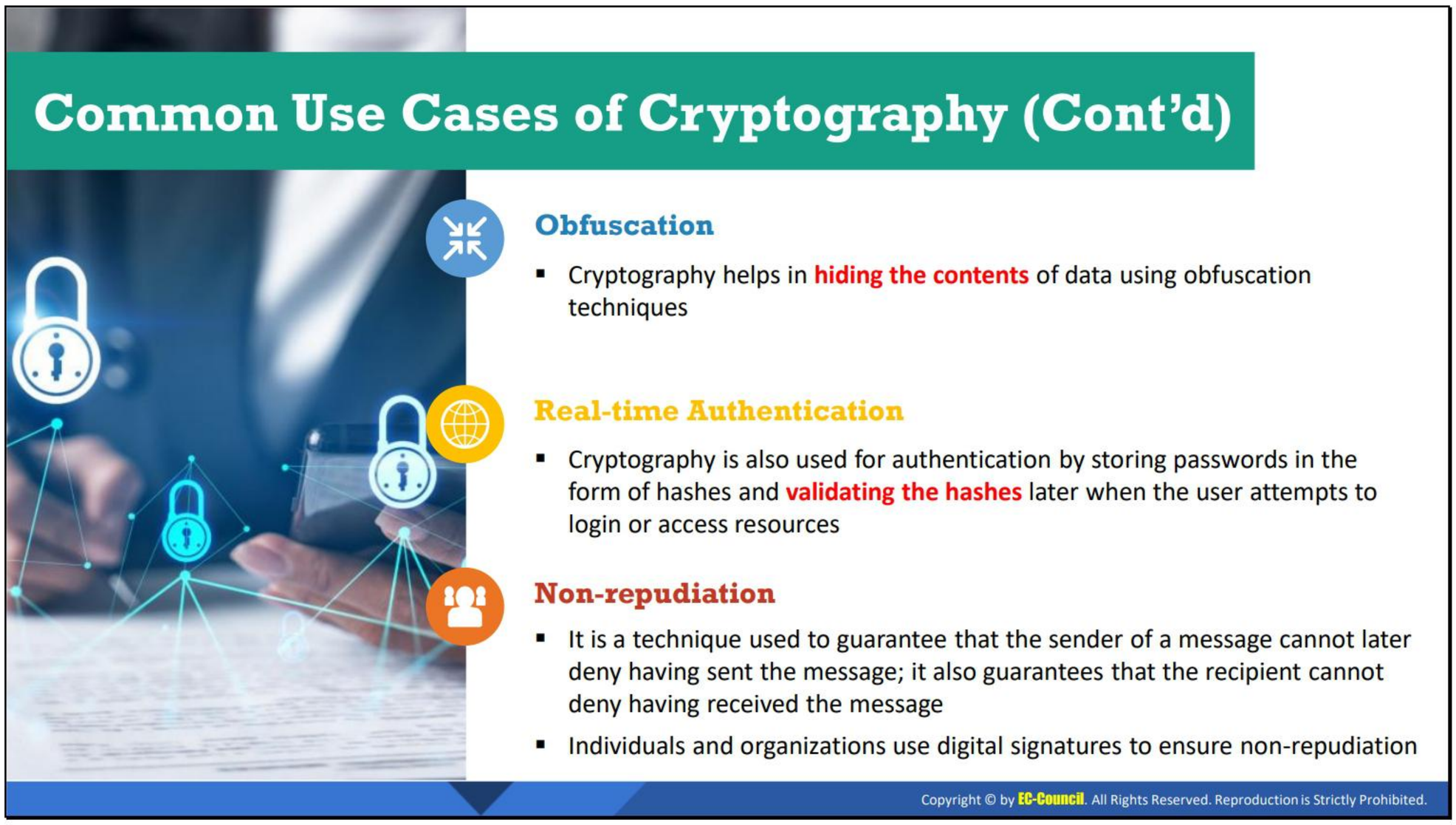
In the key stretching technique, the initial key is given as input to an algorithm that generates an enhanced key. The key must be sufficiently resistant to brute-force attacks. It is very difficult for the attackers to predict the enhanced key as they need to try every possible combination of the key or likely combinations of the enhanced key.

There are many functions and libraries that perform key stretching as part of their working:

- Password-Based Key Derivation Function 2 (PBKDF2) is part of PKCS #5 v. 2.01. It applies some function (such as a hash or HMAC) to the password or passphrase along with Salt to produce a derived key.
- bcrypt is used with passwords, and it essentially uses a variant of the Blowfish algorithm, converted to a hashing algorithm, to hash a password and add Salt to it.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.





Common Use Cases of Cryptography

In this Internet era, cryptographic algorithms are used everywhere from simple communication to cryptocurrency mining operations. They are used in a variety of industries to protect sensitive data from unauthorized access.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

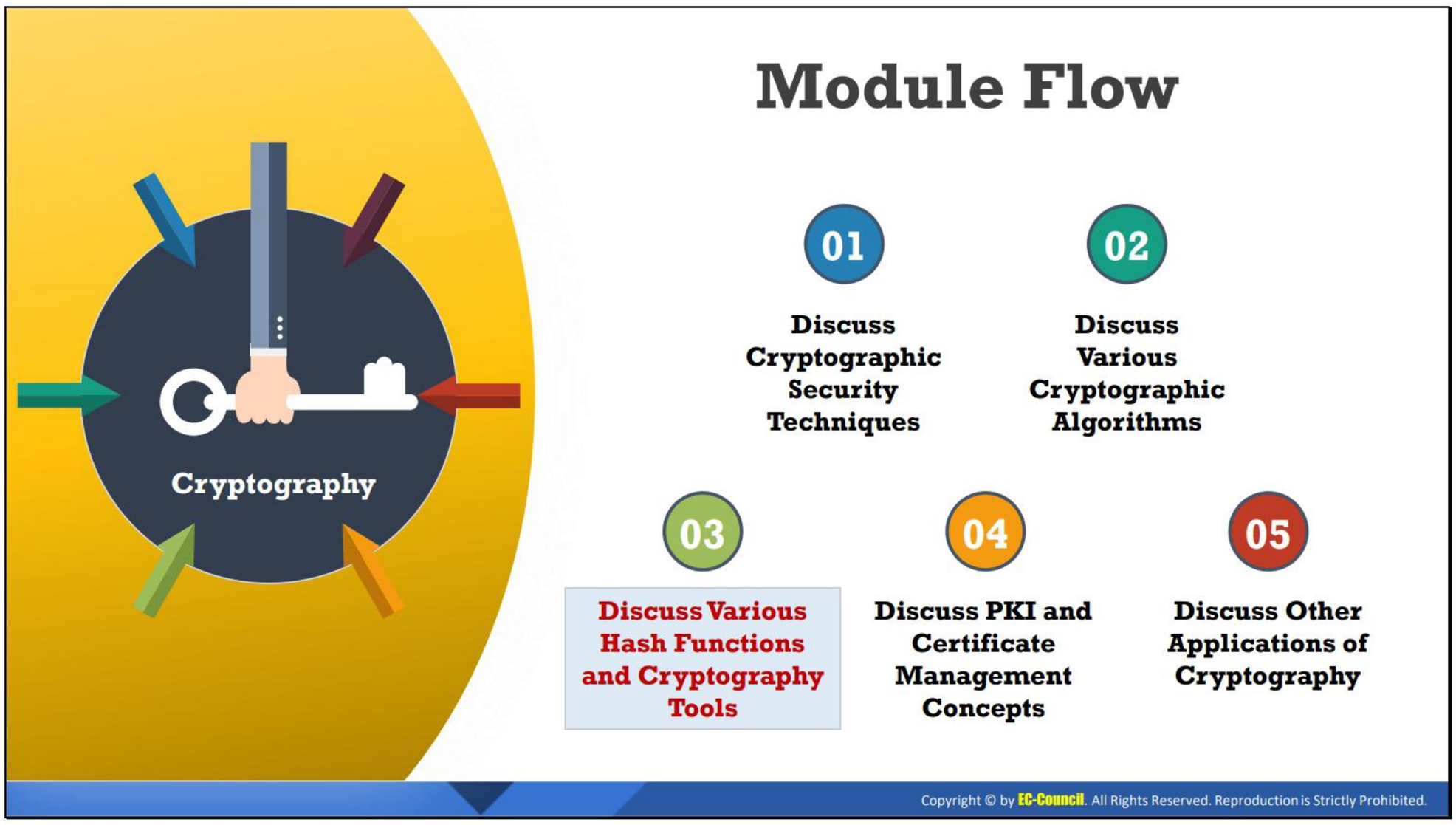
The following are some of the most common use cases of cryptography.

- **Low-power devices**: One of the major concerns with modern mobile technology is its battery backup. In addition to providing security for day-to-day activities, the battery should be available throughout the day. Therefore, mobile devices are typically integrated with cryptographic algorithms such as elliptic-curve cryptography (ECC) that have low power consumption for asymmetric encryption or use small symmetric keys.
- **Low latency**: For an application or software that needs low latency, an encryption algorithm with low computational resource utilization is required. It is necessary to encrypt and decrypt the data rapidly to achieve low latency. For this purpose, security professionals use symmetric cryptography with small keys to keep the process efficient.
- **Confidentiality**: One of the important use cases of cryptography is to ensure that information or data are confidential. Cryptography is used to ensure the confidentiality of private and hidden conversations; even if the ciphertext is stolen, the eavesdropper cannot understand or alter what they have stolen. Cryptography is used to perform file-level, disc-level, and email-based encryption to achieve confidentiality.
- **Integrity and resiliency**: Occasionally, the data transmitted by a sender might be tampered with during transition, and the data may not be appropriately delivered to the recipient, causing integrity failure. To avoid such problems, the source and destination must use encrypted cryptographic hash functions, which are compared at both sides to validate the data and confirm that the data have not been tampered with in transit. In this manner, both parties validate the contents through cryptographic hashes. Alongside providing packet-level integrity, highly resilient control systems can also be developed using cryptographic methods. Resilient control systems may include several parts of hosts, data units, and complex algorithms, which protect other parts of the system if one part is compromised.
- **Obfuscation**: Maintaining integrity and resiliency is also a challenging task in the cyber space. An attacker who obtained valid credentials can tamper with data during transmission. In such cases, cryptography helps in hiding the data using obfuscation techniques. With such techniques, programmers can make content modification difficult so that attackers cannot understand the content. However, obfuscation can also be used for malicious purposes such as hiding malware programs inside content, which can be executed during decryption for subsequently infecting the system without the user's knowledge.
- **Real-time authentication**: Cryptography is also used for authentication by storing passwords in the form of hashes and validating the hashes later when the user attempts to login or access resources. Cryptography adds a salt value to randomly stored password hashes, making identical passwords appear unique; this helps in protecting passwords in storage.
- **Non-repudiation:** Non-repudiation is a technique used to guarantee that the sender of a message cannot later deny having sent the message; it also guarantees that the recipient cannot deny having received the message. Individuals and organizations use

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

digital signatures to ensure non-repudiation. Through cryptographic digital signatures, both non-repudiation and integrity of data can be achieved, as any unauthorized modifications to the data can be detected because the signatures are validated at end points. Non-repudiation ensures that the sender cannot refuse to send the packet and that the recipient can decrypt the packet but not encrypt the packet to masquerade as the sender.

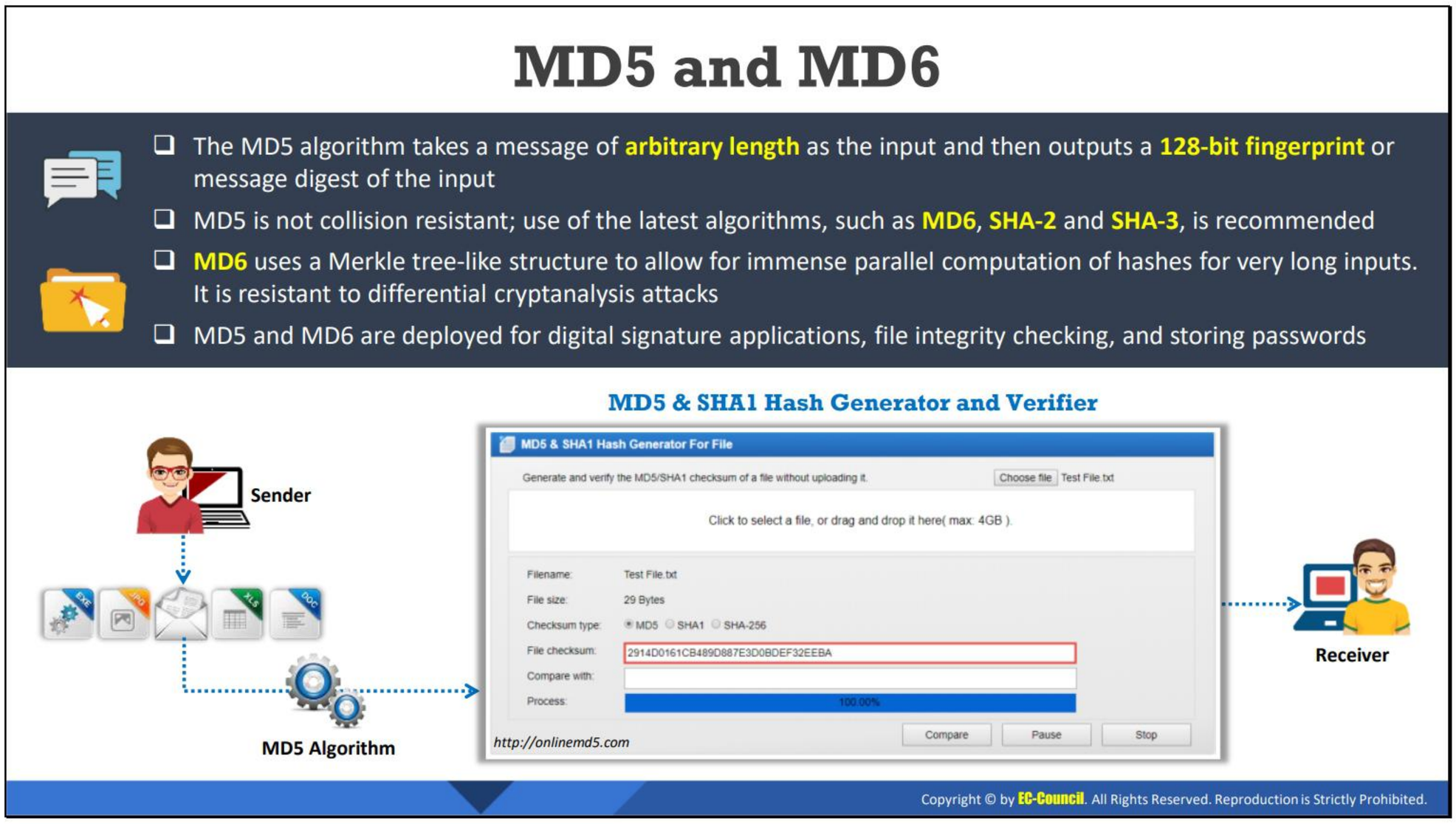
Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Discuss Various Hash Functions and Cryptography Tools

This section deals with various hash functions such as MD5, MD6, SHA, etc. and cryptography tools that you can use to encrypt sensitive data to protect it from unauthorized access by any party other than the person for whom it is intended.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



MD5 and MD6

MD2, MD4, MD5, and MD6 are **message digest algorithms** used in digital signature applications to compress a document securely before the system signs it with a private key. The algorithms can be of variable length, but the resulting message digest always has a size of 128 bits.

The structures of all three algorithms (MD2, MD4, and MD5) appear similar, although the design of MD2 is reasonably different from that of MD4 and MD5. MD2 supports 8-bit machines, while MD4 and MD5 support 32-bit machines. The algorithm pads the message with extra bits to ensure that the number of bits is divisible by 512. The extra bits may include a 64-bit binary message.

Attacks on versions of MD4 have become increasingly successful. Research has shown how an attacker launches collision attacks on the full version of MD4 within a minute on a typical PC. MD5 is slightly more secure but is slower than MD4. However, both the message digest size and the padding requirements remain the same.

MD5 is a widely used cryptographic hash function that takes a message of arbitrary length as input and outputs a 128-bit (16-byte) fingerprint or message digest of the input. MD5 can be used in a wide variety of cryptographic applications and is useful for digital signature applications, file integrity checking, and storing passwords. However, MD5 is not collision resistant; therefore, it is better to use the latest algorithms, such as MD6, SHA-2, and SHA-3.

MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks.

To calculate the effectiveness of hash functions, check the output produced when the algorithm randomizes an arbitrary input message.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The following are examples of minimally different message digests:

- echo “There is CHF1500 in the blue bo” | md5sum
  e41a323bdf20eadafd3f0e4f72055d36
- echo “There is CHF1500 in the blue box” | md5sum
  7a0da864a41fd0200ae0ae97afd3279d
- echo “There is CHF1500 in the blue box.” | md5sum
  2db1ff7a70245309e9f2165c6c34999d

Even minimally different texts produce radically different MD5 codes.



Figure 14.12: Verifying MD5 Hash

Note: Message digests are also called as one-way hash functions because they cannot be reversed.

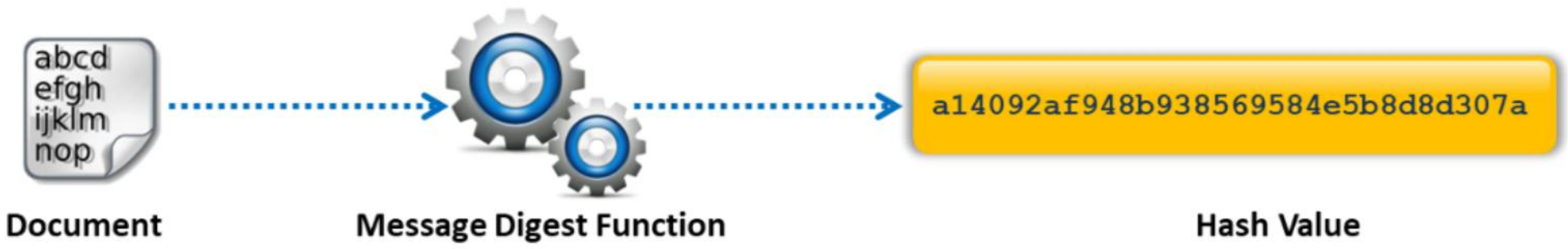
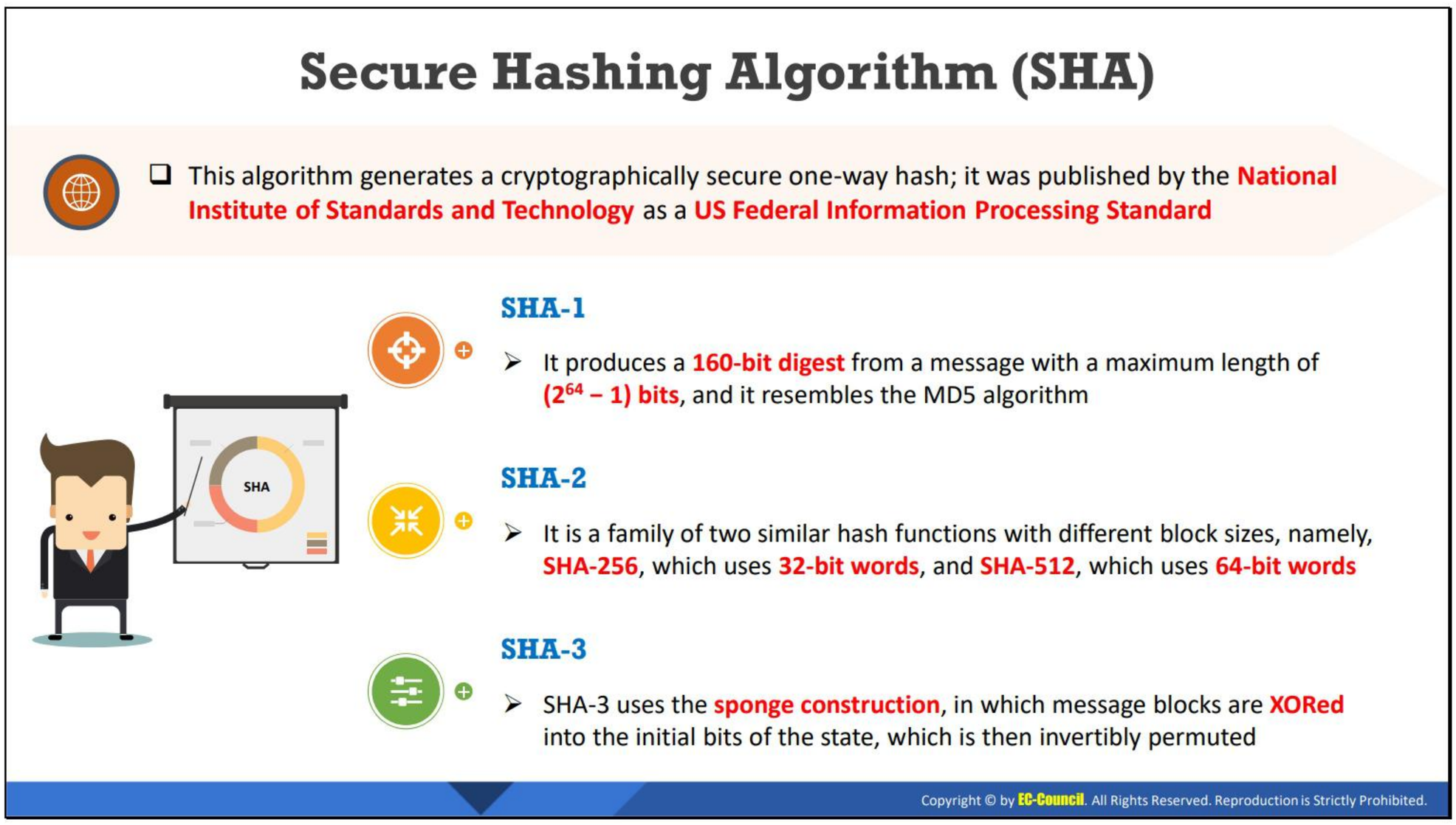


Figure 14.13: Working of Message Digests

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Secure Hashing Algorithm (SHA)

The NIST has developed the Secure Hash Algorithm (SHA), specified in the **Secure Hash Standard** (**SHS**) and published as a federal information-processing standard (FIPS PUB 180). It generates a cryptographically secure one-way hash. Rivest developed the SHA, which is similar to the message digest algorithm family of hash functions. It is slightly slower than MD5, but its larger message digest makes it more secure against brute-force collision and inversion attacks.

SHA encryption is a series of five different cryptographic functions, and it currently has three generations: SHA-1, SHA-2, and SHA-3.

- **SHA-0**: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name SHA, which was withdrawn from trade due to an undisclosed "**significant flaw**" in it. It was replaced with a slightly revised version, namely SHA-1.
- **SHA-1**: It is a 160-bit hash function that resembles the former MD5 algorithm developed by Ron Rivest. It produces a 160-bit digest from a message with a maximum length of (2^{64} – 1) bits. It was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm (DSA). It is most commonly used in security protocols such as PGP, TLS, SSH, and SSL. As of 2010, SHA-1 is no longer approved for cryptographic use because of its cryptographic weaknesses.
- **SHA-2**: SHA2 is a family of two similar hash functions with different block sizes, namely SHA-256, which uses 32-bit words, and SHA-512, which uses 64-bit words. The truncated versions of each standard are SHA-224 and SHA-384.
- **SHA-3**: SHA-3 uses sponge construction in which message blocks are XORed into the initial bits of the state, which the algorithm then invertibly permutes. It supports the same hash lengths as SHA-2 but differs in its internal structure considerably from the rest of the SHA family.

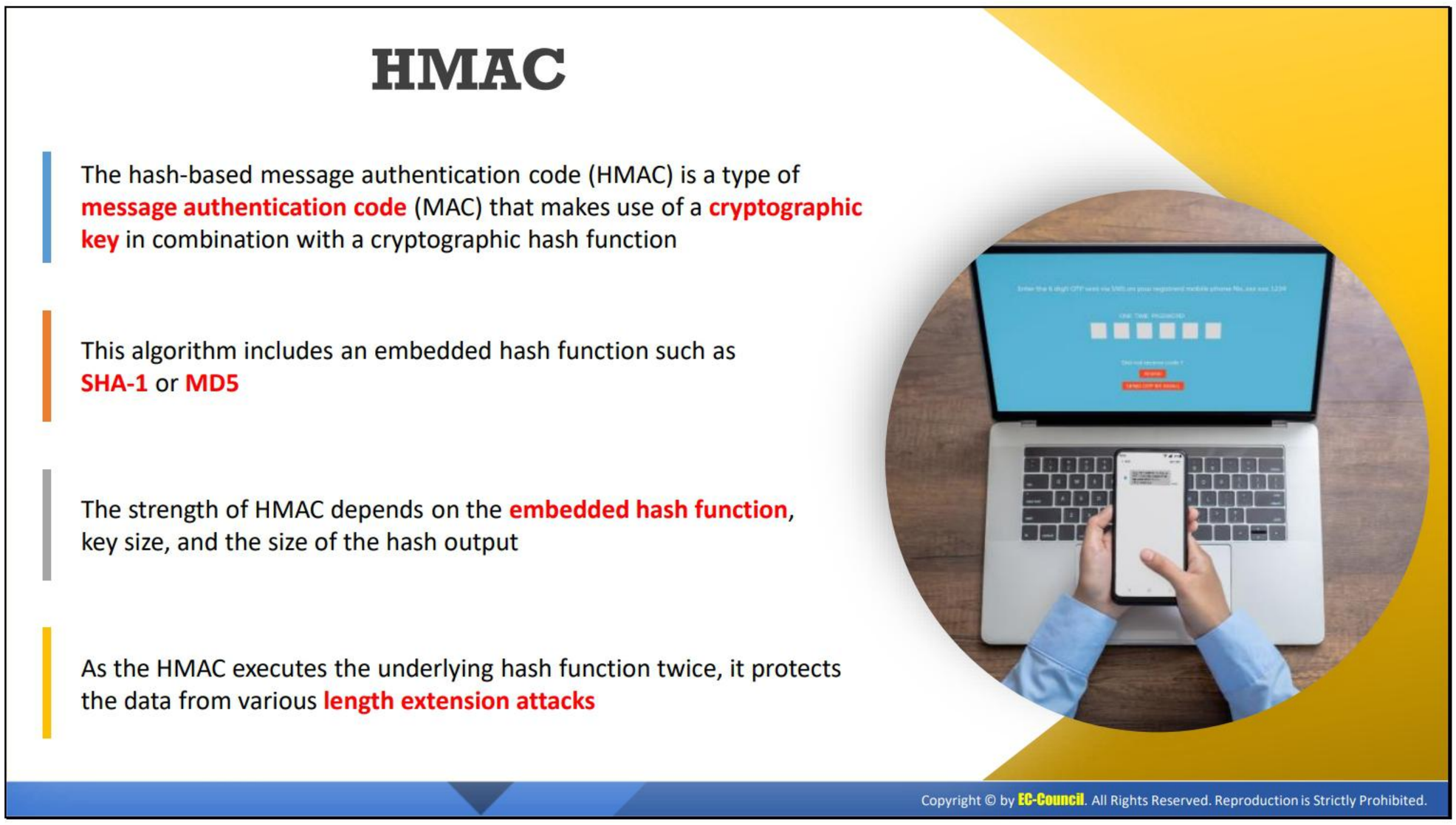
Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Comparison of SHA functions (SHA-0, SHA-1, SHA-2, and SHA-3).

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block Size (bits)	Maximum message size (bits)	Rounds	Operations	Security (bits)
MD5 (as reference)		128	128 (4*32)	512	2^{64}-1	64	Add mod 2^{32}, and, or, xor, rot	<=18 (collisions found)
SHA-0		160	160 (5*32)	512	2^{64}-1	80	Add mod 2^{32}, and, or, xor, rot	<34 (collisions found)
SHA-1		160	160 (5*32)	512	2^{64}-1	80	Add mod 2^{32}, and, or, xor, rot	<63 (collisions found)
SHA-2	SHA-224 SHA-256	224 256	256 (8*32)	512	2^{64}-1	64	Add mod 2^{32}, and, or, xor, shr, rot	112 128
	SHA-384 SHA-512 SHA-512/224 SHA-512/256	384 512 224 256	512 (8*64)	1024	2^{128}-1	80	Add mod 2^{64}, and, or, xor, shr, rot.	192 256 112 128
SHA-3	SHA3-224 SHA3-256 SHA3-384 SHA3-512 SHAKE128 sHAKE256	224 256 384 512 d(arbitrary) d(arbitrary)	1600 (5*5*64)	1152 1088 832 576 1344 1088	∞	24	And, xor, not, rot	112 128 192 256 Min(d/2,128) Min(d/2,256)

Table 14.2: Comparison between SHA functions

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



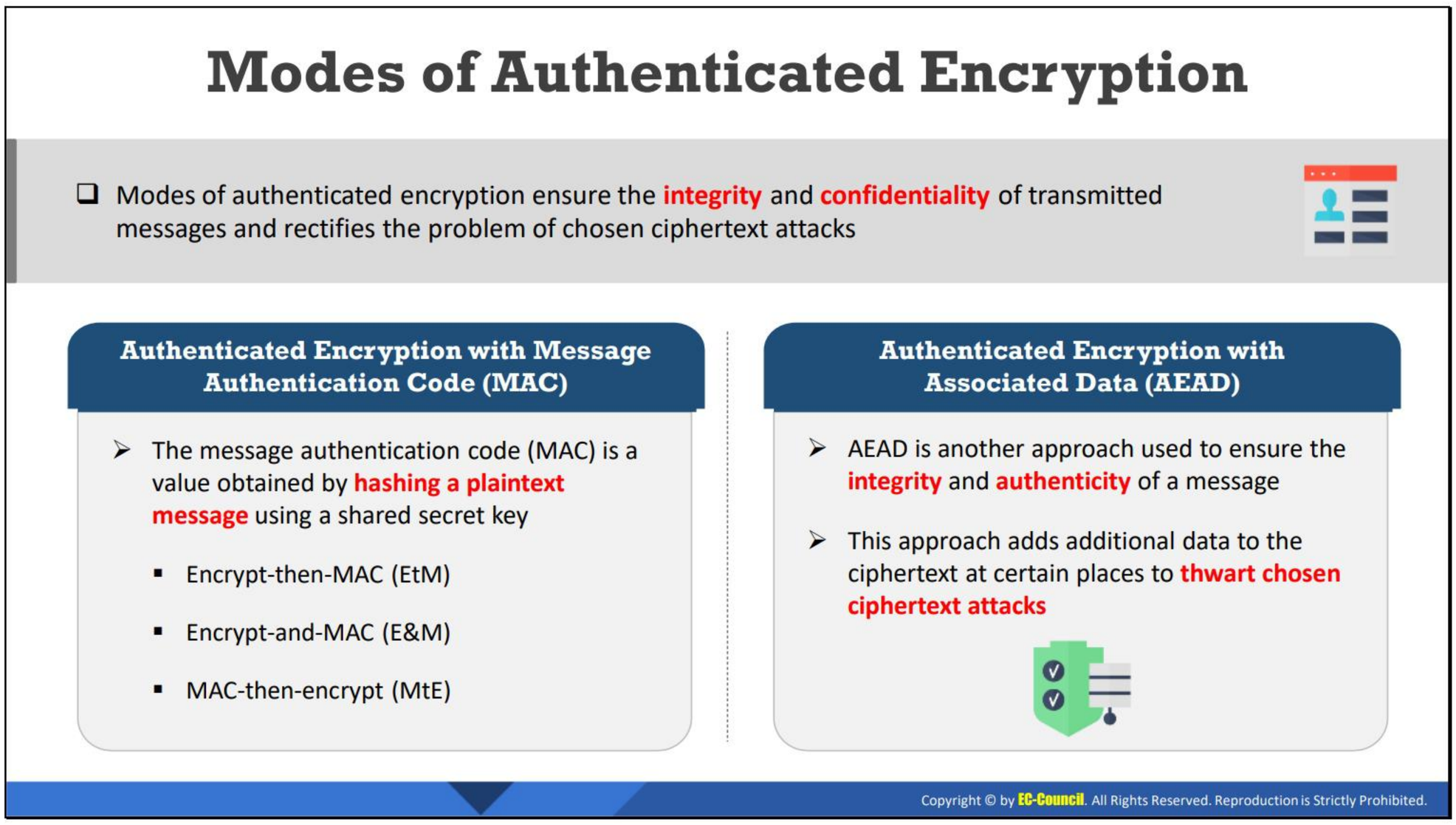
HMAC

Hash-based message authentication code (HMAC) is a type of message authentication code (MAC) that uses a cryptographic key along with a cryptographic hash function. It is widely used to verify the integrity of data and authentication of a message. This algorithm includes an embedded hash function such as SHA-1 or MD5. The strength of HMAC depends on the embedded hash function, key s1ize, and size of the hash output.

HMAC includes two stages for computing the hash. The input key is processed to produce two keys, namely the inner key and the outer key. The first stage of the algorithm inputs the inner key and message to produce an internal hash. The second stage of the algorithm inputs the output from the first stage and outer key and produces the final HMAC code.

As HMAC executes the underlying hash function twice, it offers protection against various length extension attacks. The size of the key and the output depends on the embedded hash function, e.g., 128 or 160 bits in the case of MD5 or SHA-1, respectively.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Modes of Authenticated Encryption

Authenticated encryption (AE) modes of operation provide integrity and confidentiality for a transmitted message. In any cipher mode of operation, encryption/decryption is possible only with the shared secret key, preventing man-in-the-middle (MITM) attacks. However, the attacker can perform a chosen ciphertext attack to crack the encryption schema. The AE schema rectifies the problem of chosen cipher attacks. In AE modes, the ciphertext is combined with a message authentication code (MAC). Therefore, choosing a part in the ciphertext is not possible, and the AE scheme rejects improper ciphertexts while decrypting.

Authenticated encryption with Message Authentication Code (MAC)

A MAC is a value obtained by hashing a plaintext message using a shared secret key. It provides integrity for the message, and the receiver can verify the message using the hash value attached to it. The following are the three different ways of using a MAC while encrypting a message.

- **Encrypt-then-MAC (EtM)**

  In this approach, the plaintext is first encrypted using a secret key. For the obtained ciphertext, a hash value called message authentication code (MAC) is generated. The MAC is attached to the ciphertext and transmitted. This approach provides higher security for the transmitted message than other AE approaches.

- **Encrypt-and-MAC (E&M)**

  In the E&M approach, a MAC is first generated for the plaintext, following which the plaintext is encrypted using a secret key. Finally, both the ciphertext and MAC are combined and transmitted.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

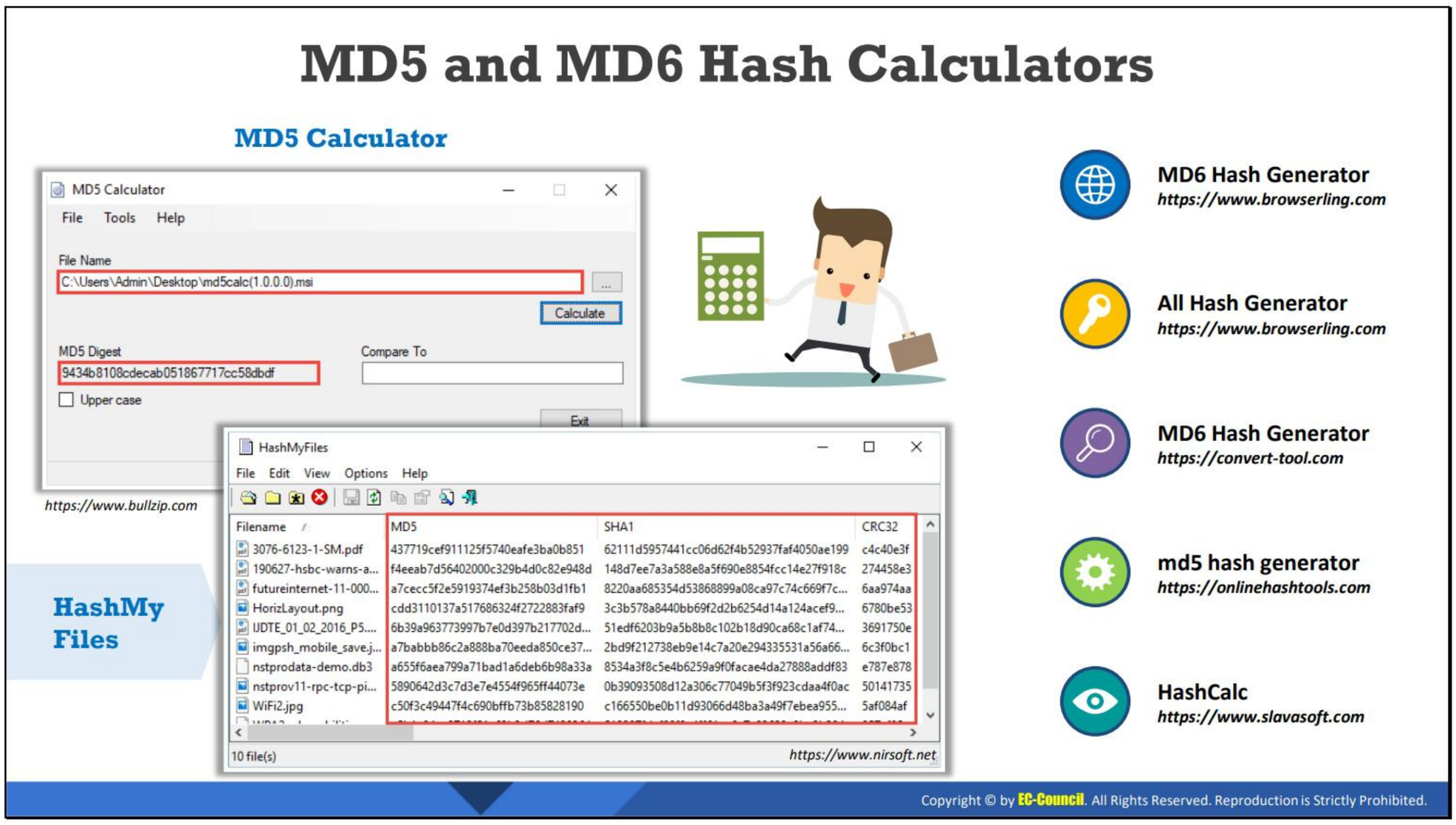
- **MAC-then-encrypt (MtE)**

  In the MtE approach, a MAC is first generated for the plaintext using the hash function, and the MAC is combined with the plaintext. The combination of the plaintext and MAC is encrypted with a secret key to produce ciphertext. The ciphertext contains the encrypted MAC.

Authenticated Encryption with Associated Data (AEAD)

AEAD is another approach used to ensure the integrity and authenticity of a message that contains both encrypted and unencrypted data. This approach adds additional data to the ciphertext at certain places to thwart chosen ciphertext attacks. The message header is kept unencrypted so that the receiver can verify the source of the message, and the payload is encrypted to ensure confidentiality.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



MD5 and MD6 Hash Calculators

MD5 and MD6 hash calculators that use different hash algorithms to convert plaintext into its equivalent hash value are discussed below.

- **MD5 Calculator**

  Source: *https://www.bullzip.com*

  MD5 Calculator is a simple application that calculates the MD5 hash of a given file. It can be used with large files (e.g., several gigabytes in size). It features a progress counter and a text field from which the final MD5 hash can be easily copied to the clipboard. MD5 Calculator can be used to check the integrity of a file.

  It allows you to calculate the MD5 hash value of the selected file. Right-click the file and choose "**MD5 Calculator**;" the program will calculate the MD5 hash. The MD5 Digest field contains the calculated value. To compare this MD5 digest with another, one can paste the other value into the Compare To field. Obviously, an equal to sign ("=") appears between the two values if they are equal; otherwise, the less than ("<") or greater than (">") sign will tell you that the values are different.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

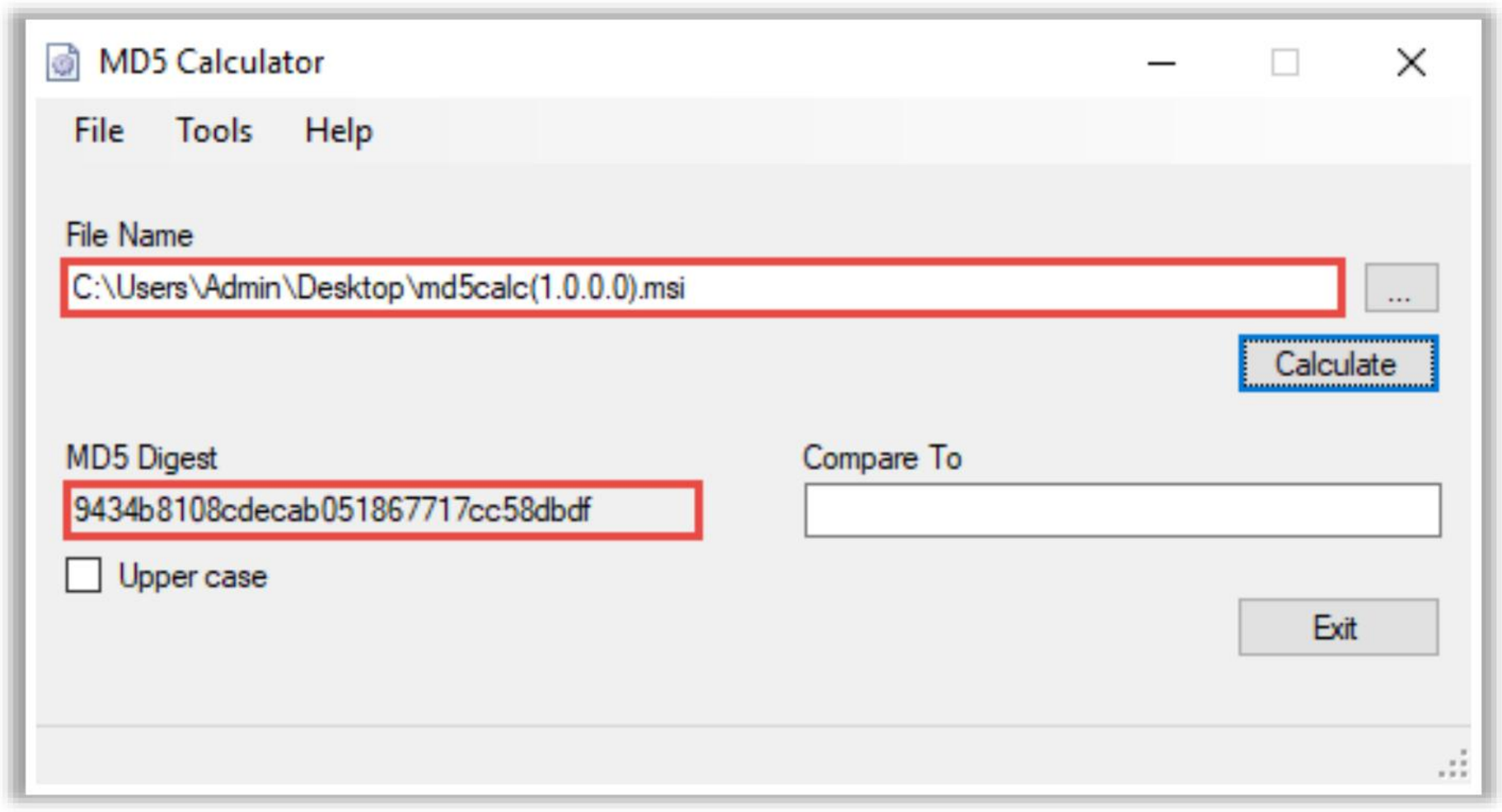


Figure 14.14: Screenshot of MD5 Calculator

- **HashMyFiles**

  Source: *https://www.nirsoft.net*

  HashMyFiles is a utility that allows you to calculate the MD5 and SHA1 hashes of one or more files in the system. It allows you to copy the MD5/SHA1 hash list to the clipboard or save it in a text/html/xml file. You can launch HashMyFiles from the context menu of Windows Explorer and display the MD5/SHA1 hashes of the selected files or folders.

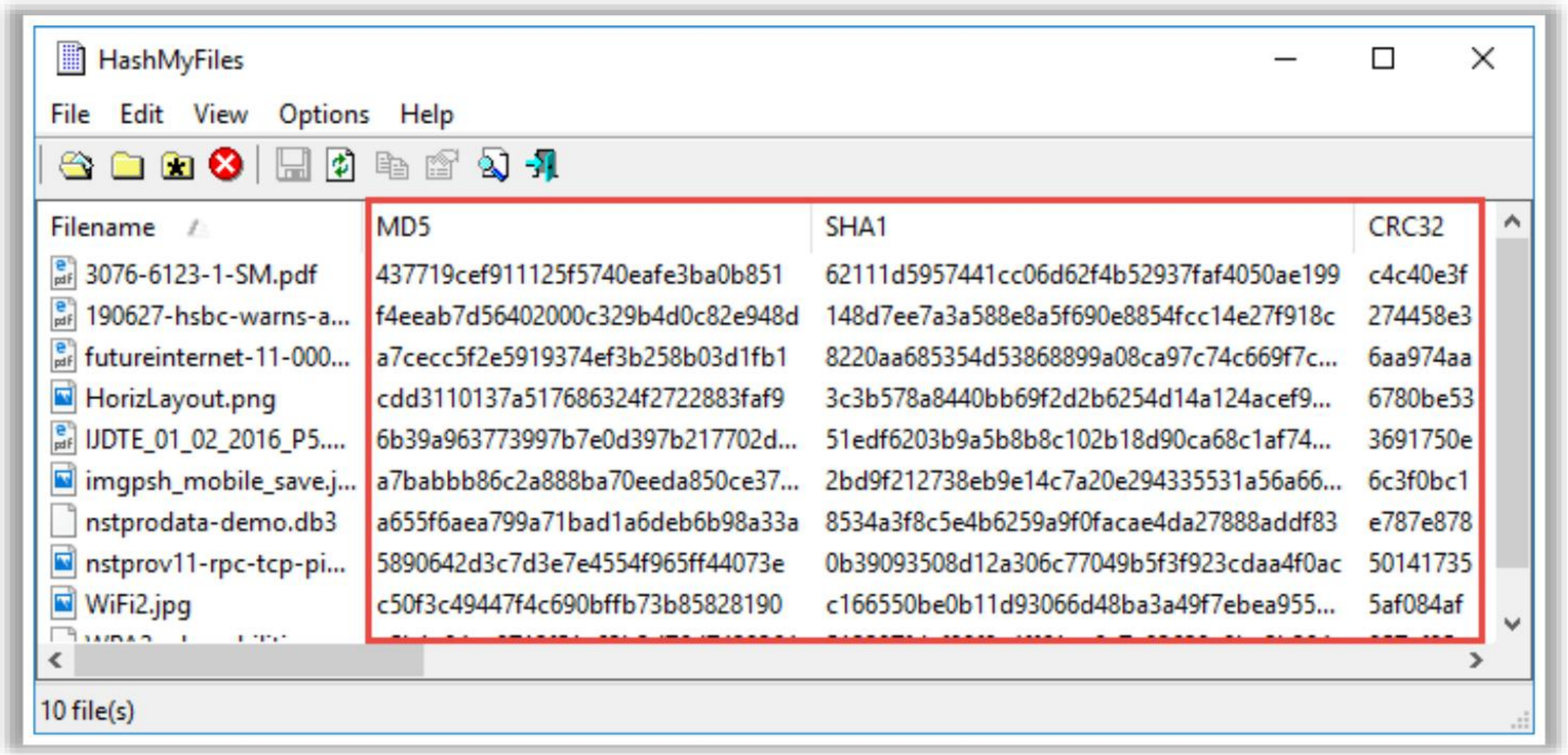


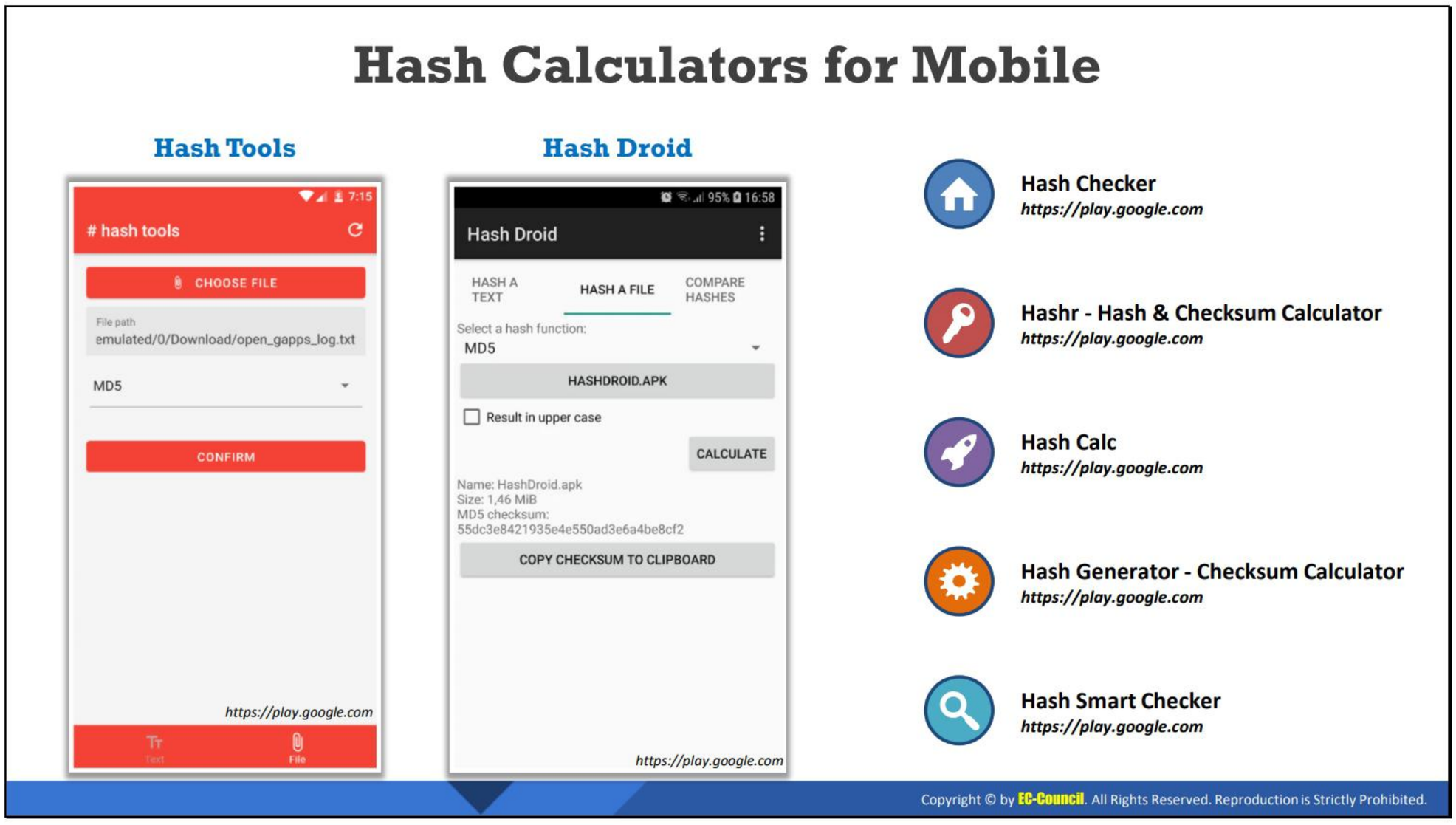
Figure 14.15: Screenshot of HashMyFiles

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some additional MD5 and MD6 hash calculators are as follows:

- MD6 Hash Generator (*https://www.browserling.com*)
- All Hash Generator (*https://www.browserling.com*)
- MD6 Hash Generator (*https://convert-tool.com*)
- md5 hash generator (*https://onlinehashtools.com*)
- HashCalc (*https://www.slavasoft.com*)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Hash Calculators for Mobile

Some hash calculators for mobile devices are discussed below.

- **Hash Tools**

  Source: *https://play.google.com*

  Hash Tools is a utility for calculating a hash from a given text or decrypting a hash to its original text. In this application, the available hash functions are MD5, SHA-1, SHA-256, SHA-384, and SHA-512.

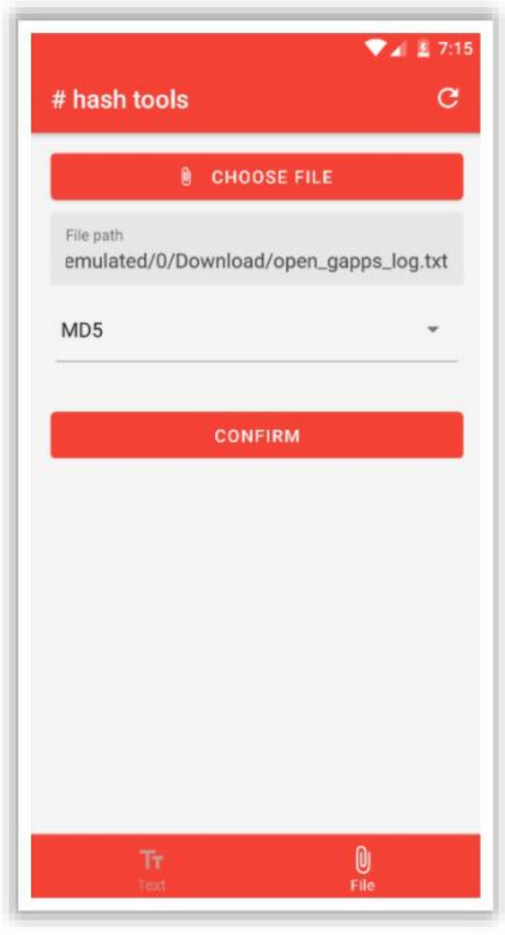


Figure 14.16: Screenshot of Hash Tools

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Hash Droid**

  Source: *https://play.google.com*

  The Hash Droid utility helps to calculate a hash from a given text or a file stored on the device. In this application, the available hash functions are Adler-32, CRC-32, Haval-128, MD2, MD4, MD5, RIPEMD-128, RIPEMD-160, SHA-1, SHA-256, SHA-384, SHA-512, Tiger, and Whirlpool.

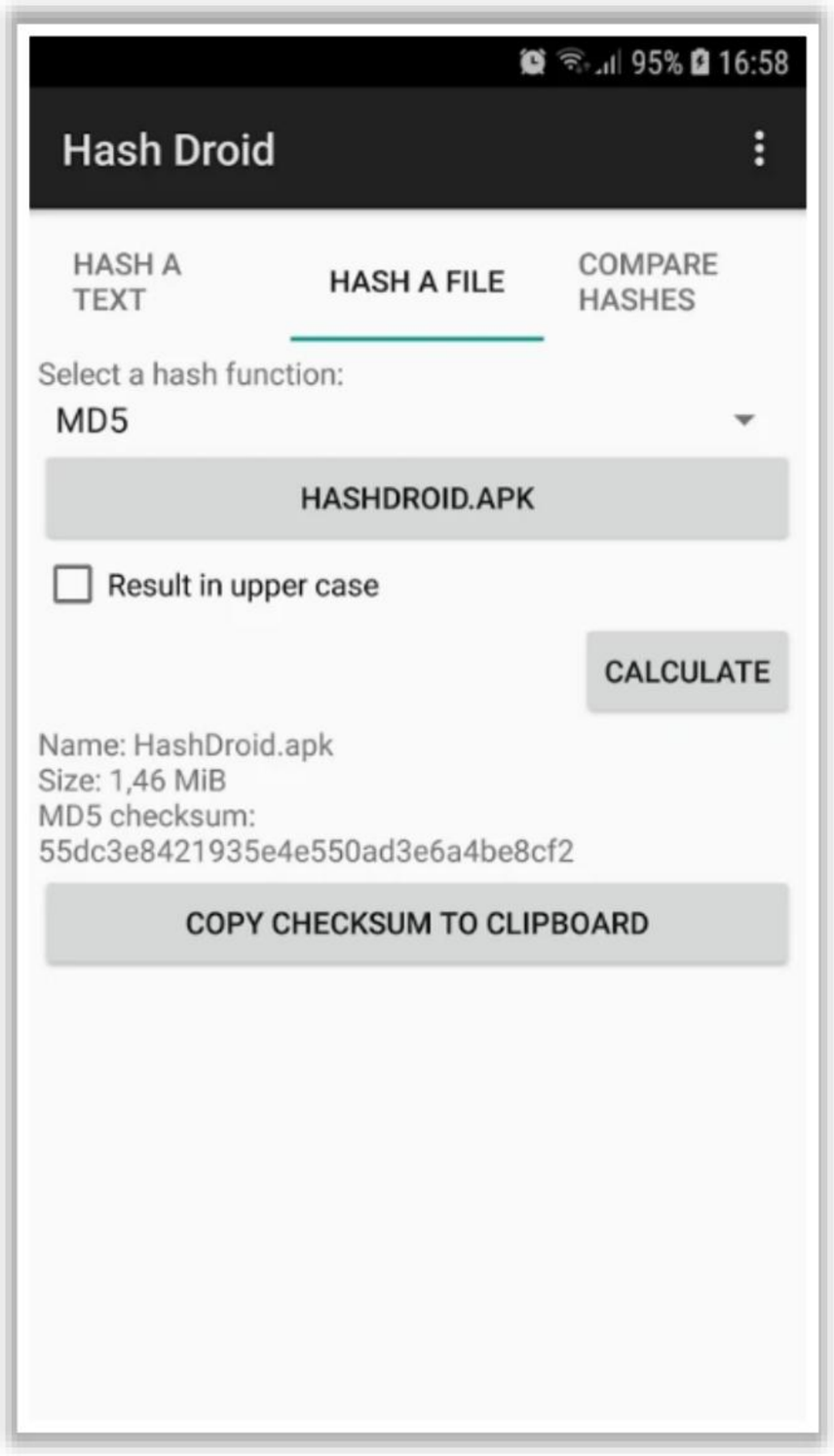
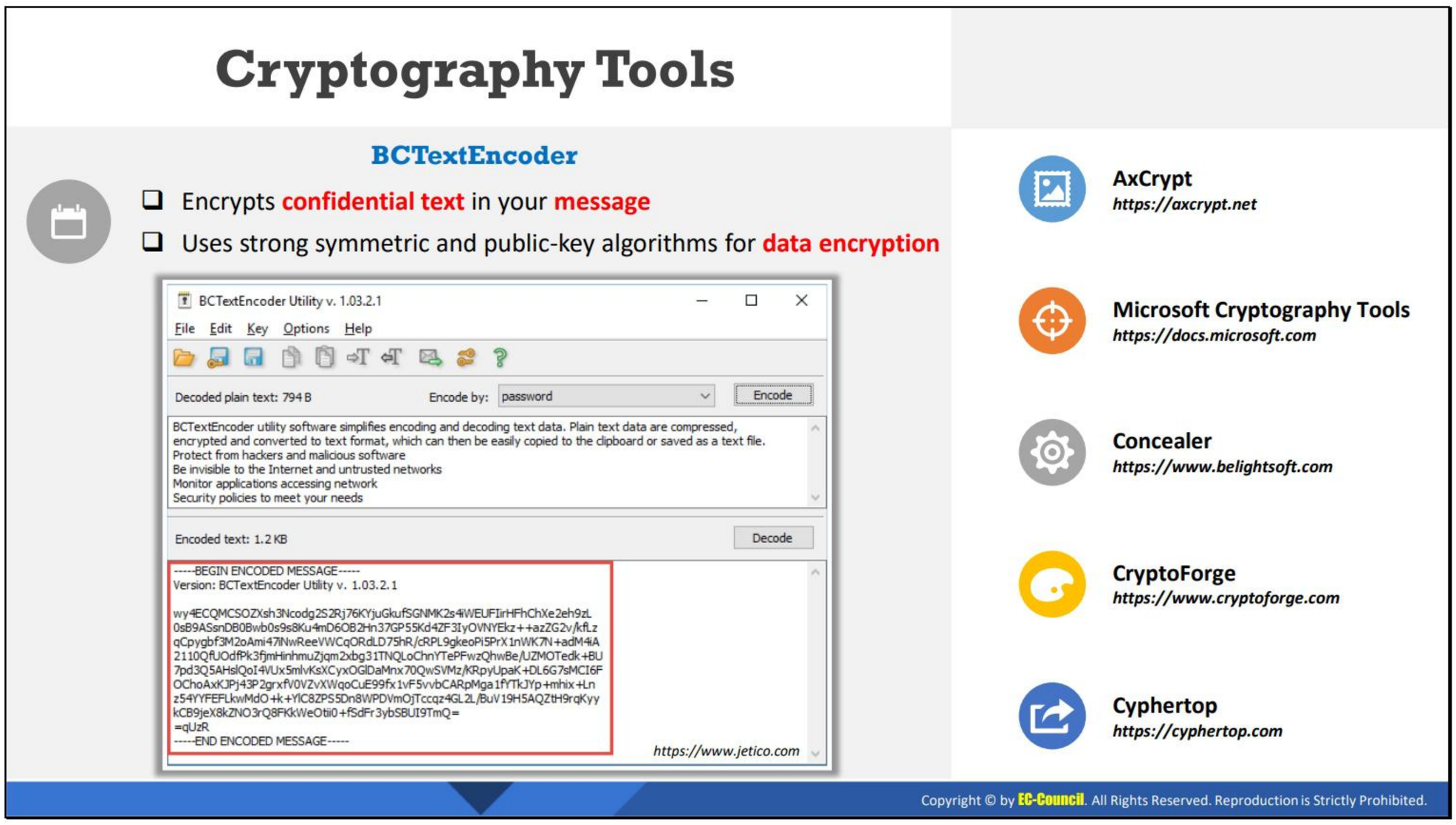


Figure 14.17: Screenshot of Hash Droid

Some additional MD5 hash calculators are as follows:

- Hash Checker (*https://play.google.com*)
- Hashr - Hash & Checksum Calculator (*https://play.google.com*)
- Hash Calc (*https://play.google.com*)
- Hash Generator - Checksum Calculator (*https://play.google.com*)
- Hash Smart Checker (*https://play.google.com*)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cryptography Tools

You can use various cryptographic tools for encrypting and decrypting your information, files, etc. These tools implement different types of encryption algorithms.

- **BCTextEncoder**

  Source: *https://www.jetico.com*

  The BCTextEncoder utility simplifies the encoding and decoding of text data. It compresses, encrypts, and converts plaintext data into text format, which the user can then copy to the clipboard or save as a text file. It uses public key encryption methods as well as password-based encryption. Furthermore, it uses strong and approved symmetric and public-key algorithms for data encryption.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

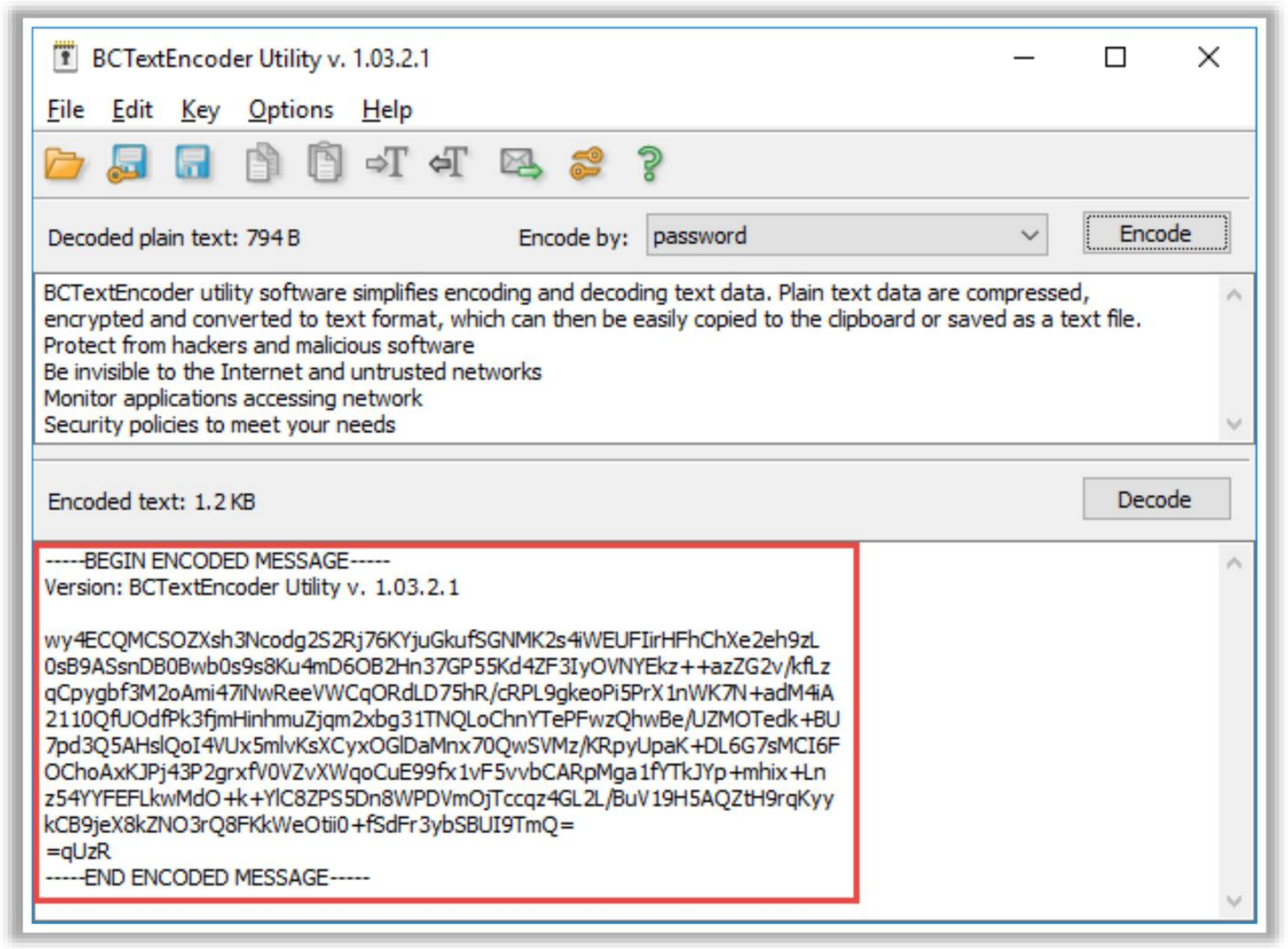
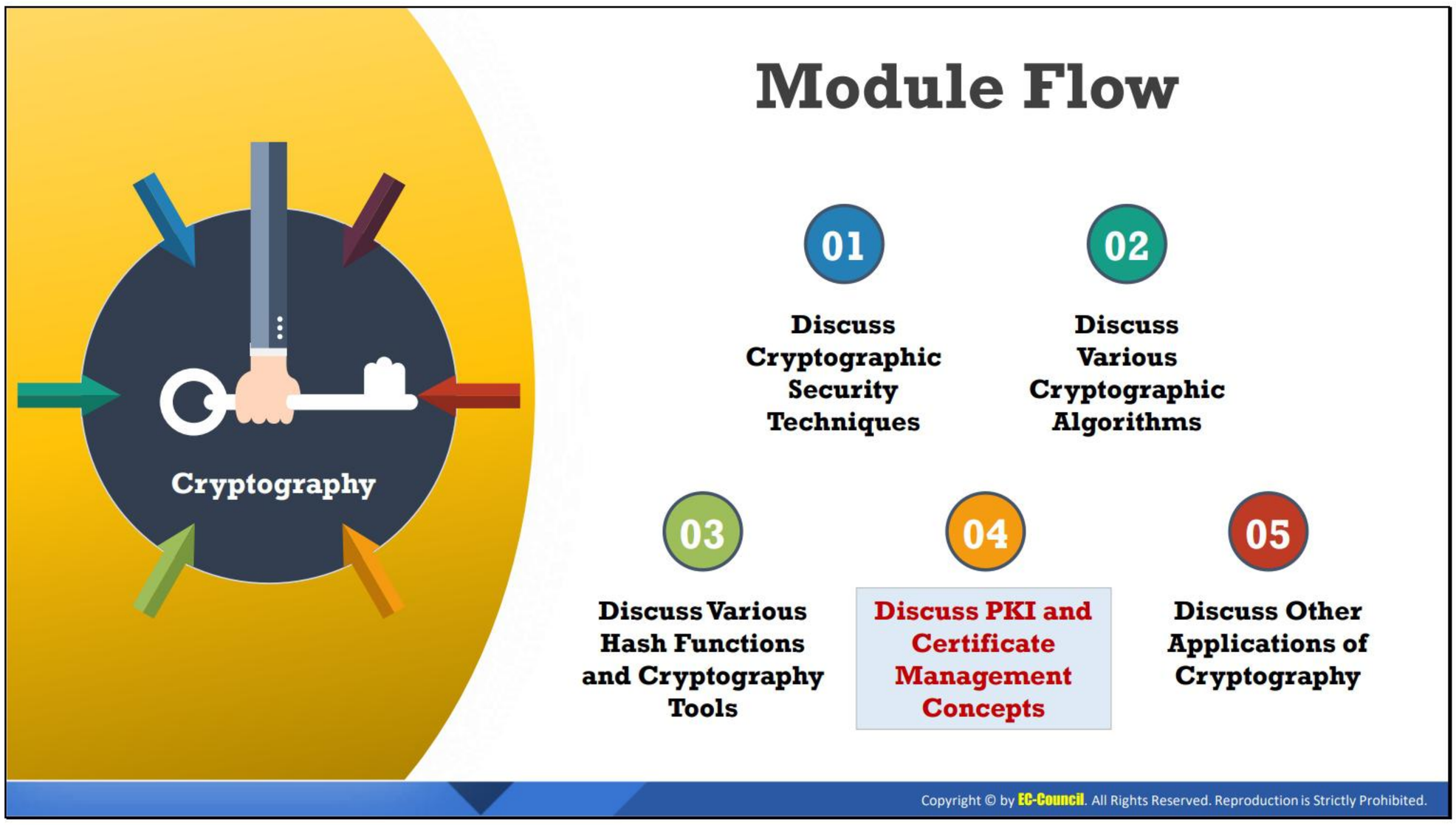


Figure 14.18: Screenshot of BCTextEncoder

Some additional cryptography tools are as follows:

- AxCrypt (*https://axcrypt.net*)
- Microsoft Cryptography Tools (*https://docs.microsoft.com*)
- Concealer (*https://www.belightsoft.com*)
- CryptoForge (*https://www.cryptoforge.com*)
- Cyphertop (*https://cyphertop.com*)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



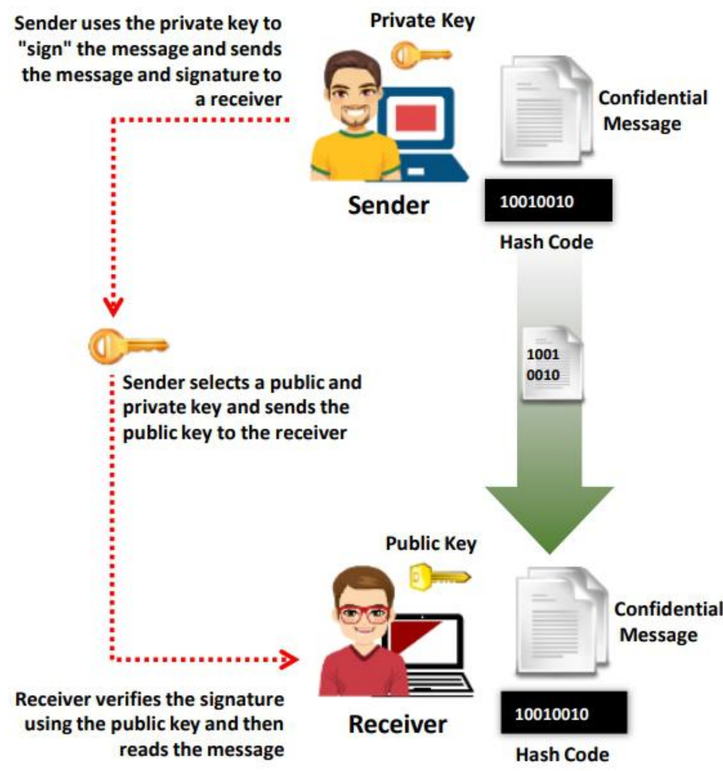
Discuss PKI and Certificate Management Concepts

This section deals with public key infrastructure (PKI) and the role of each component of PKI, and certification authorities.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Digital Signature

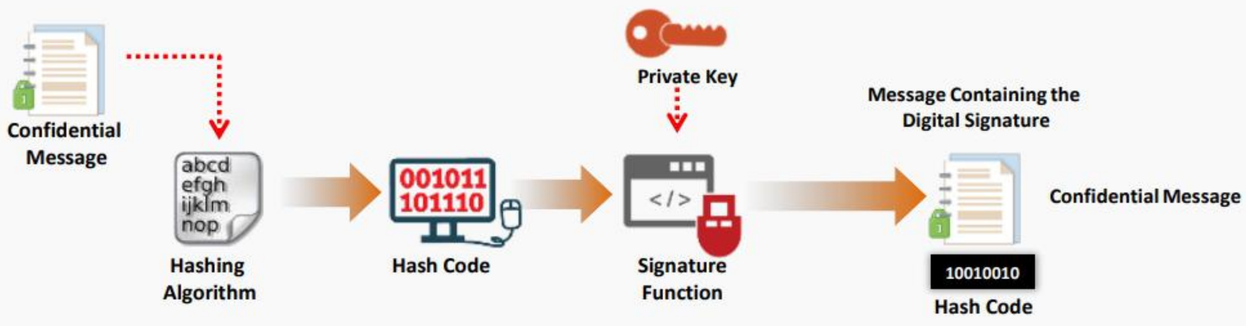
- Digital signatures use the **asymmetric key algorithms** to provide **data integrity**
- A specific signature function is added to the asymmetric algorithm at the sender's side to **digitally sign the message** and a specific **verification function** is added to verify the signature to ensure message integrity at the recipient side
- The asymmetric algorithms that support these two functions are called **digital signature algorithms**
- Digitally signing messages **slows the performance of during verification**; the hash value of the message is used instead of the message itself for better performance
- A **digital signature** is created using the hash code of the message, the **private key** of the sender, and the signature function
- It is then verified using the hash code of the message, the **public key** of sender, and the verification function



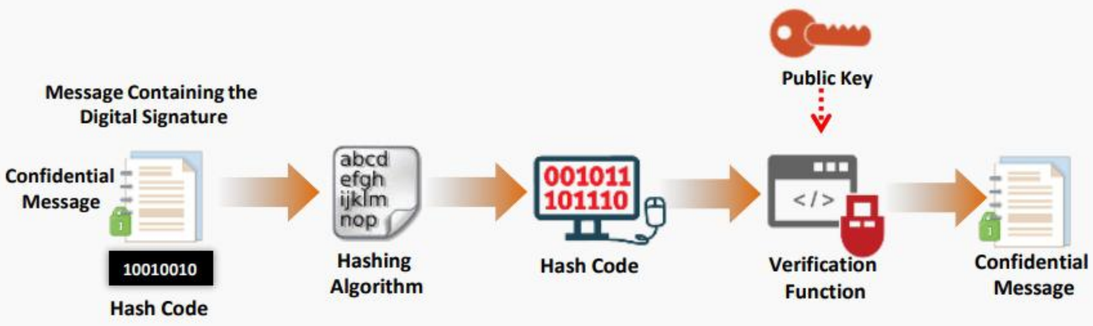
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Digital Signature (Cont'd)

Creating a digital signature at the sender side



Verifying a digital signature at the recipient side



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Digital Signature

A digital signature is a cryptographic means of authentication. Public-key cryptography uses asymmetric encryption and helps the user to create a digital signature.

A specific signature function is added to the asymmetric algorithm at the sender's side to digitally sign the message and a specific verification function is added to verify the signature to

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

ensure message integrity at the recipient side. The asymmetric algorithms that support these two functions are called digital signature algorithms.

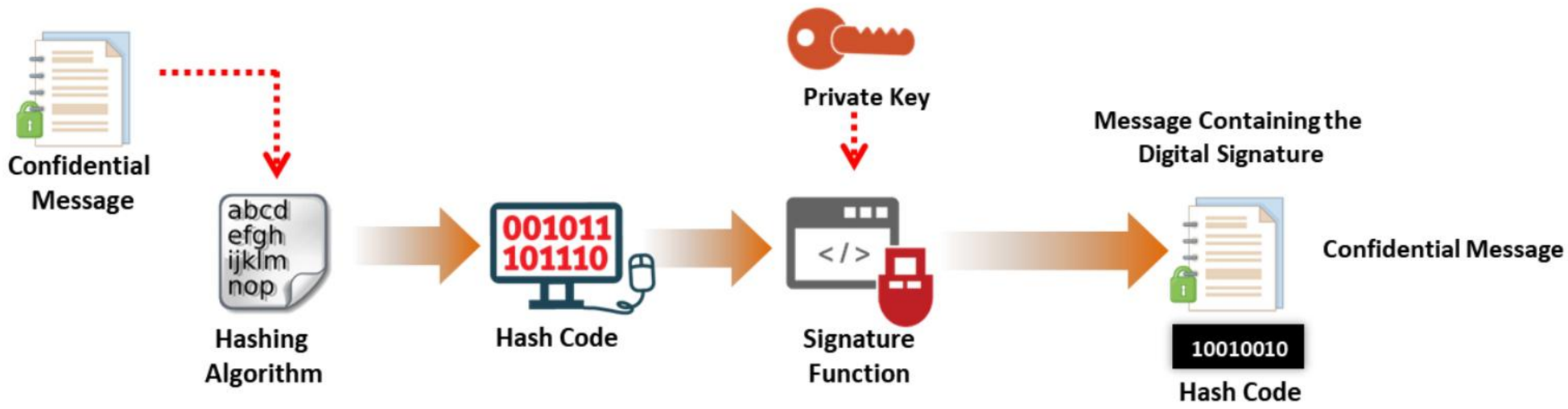


Figure 14.19: Creating a digital signature at the sender side

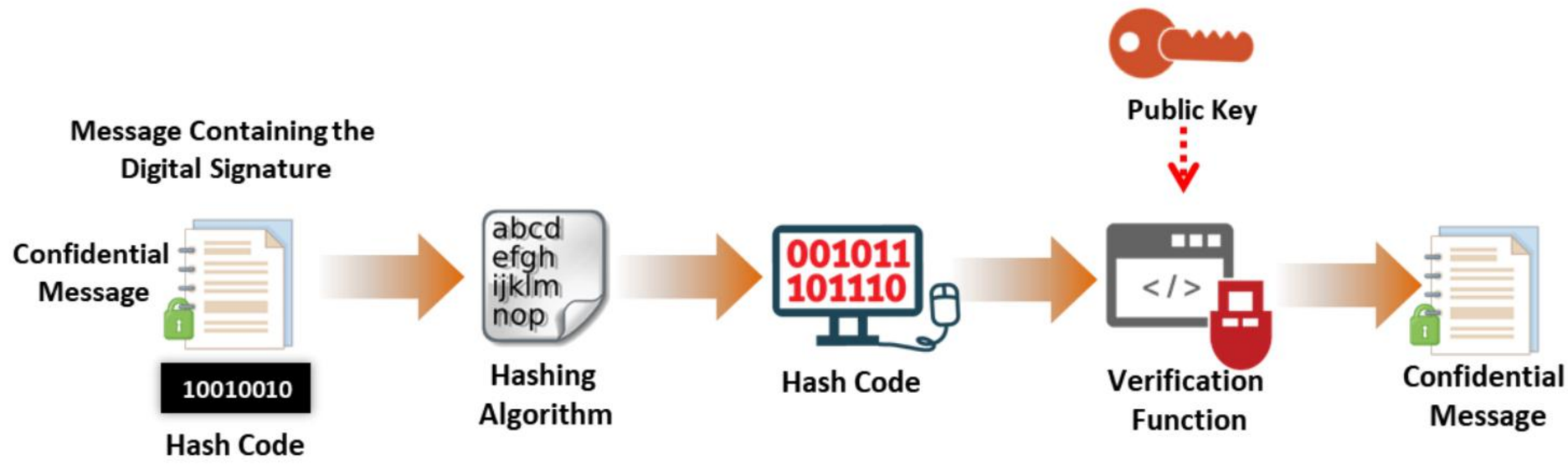


Figure 14.20: Verifying a digital signature at the recipient side

A hash function is an algorithm which helps users to create and verify digital signatures. This algorithm creates a digital representation, also known as a message fingerprint. This fingerprint has a hash value that is much smaller than the message, but one that is unique. If an attacker changes the message, the hash function will automatically produce a different hash value.

In order to verify a digital signature, one requires the hash value of the original message and the hash function used for creating the digital signature. With the help of the public key and the new result, the verifier checks whether the digital signature was created with the related private key and whether the new hash value is the same as the original or not. Digitally signing messages slows the performance of during verification; the hash value of the message is used instead of the message itself for better performance.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

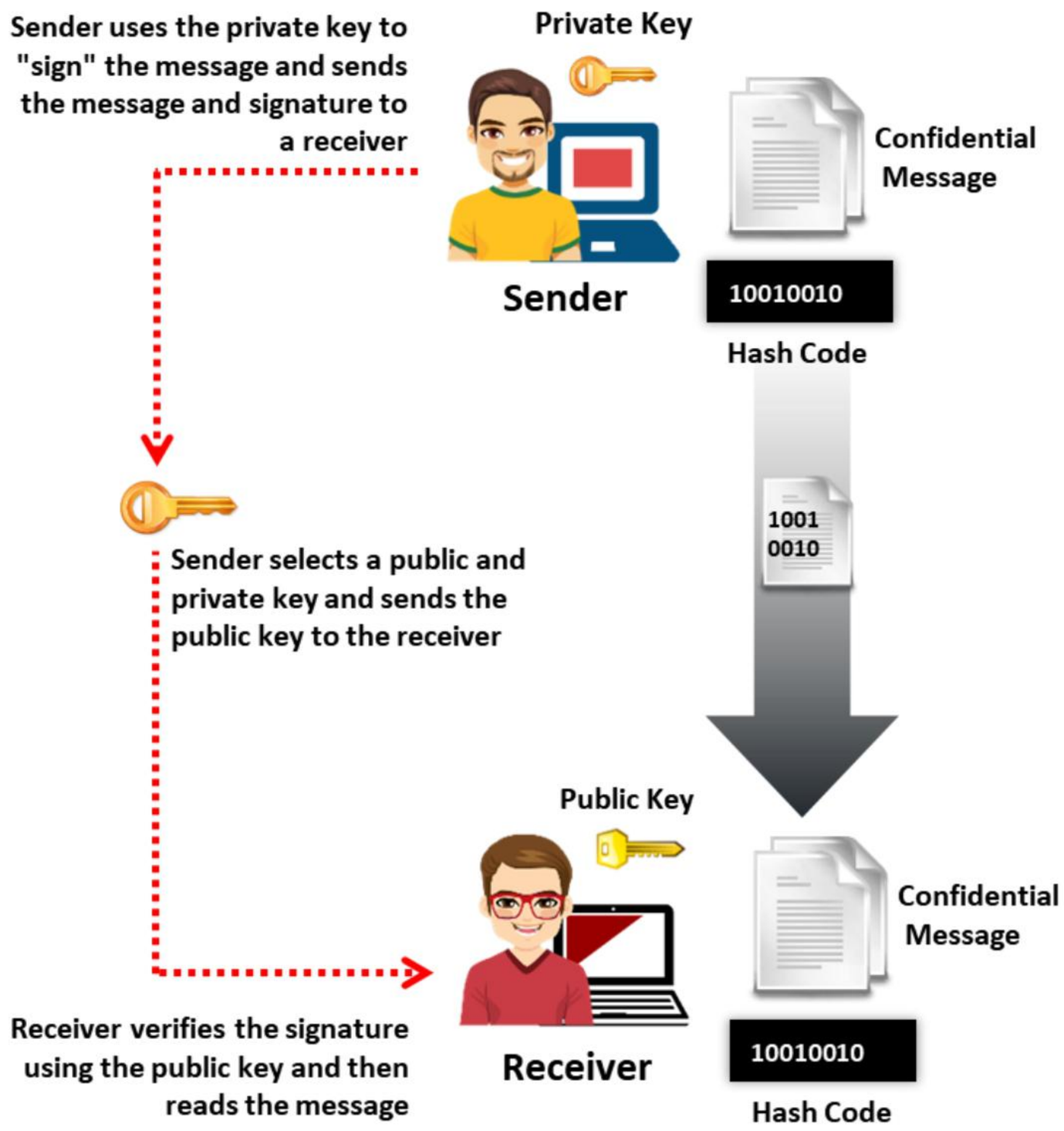
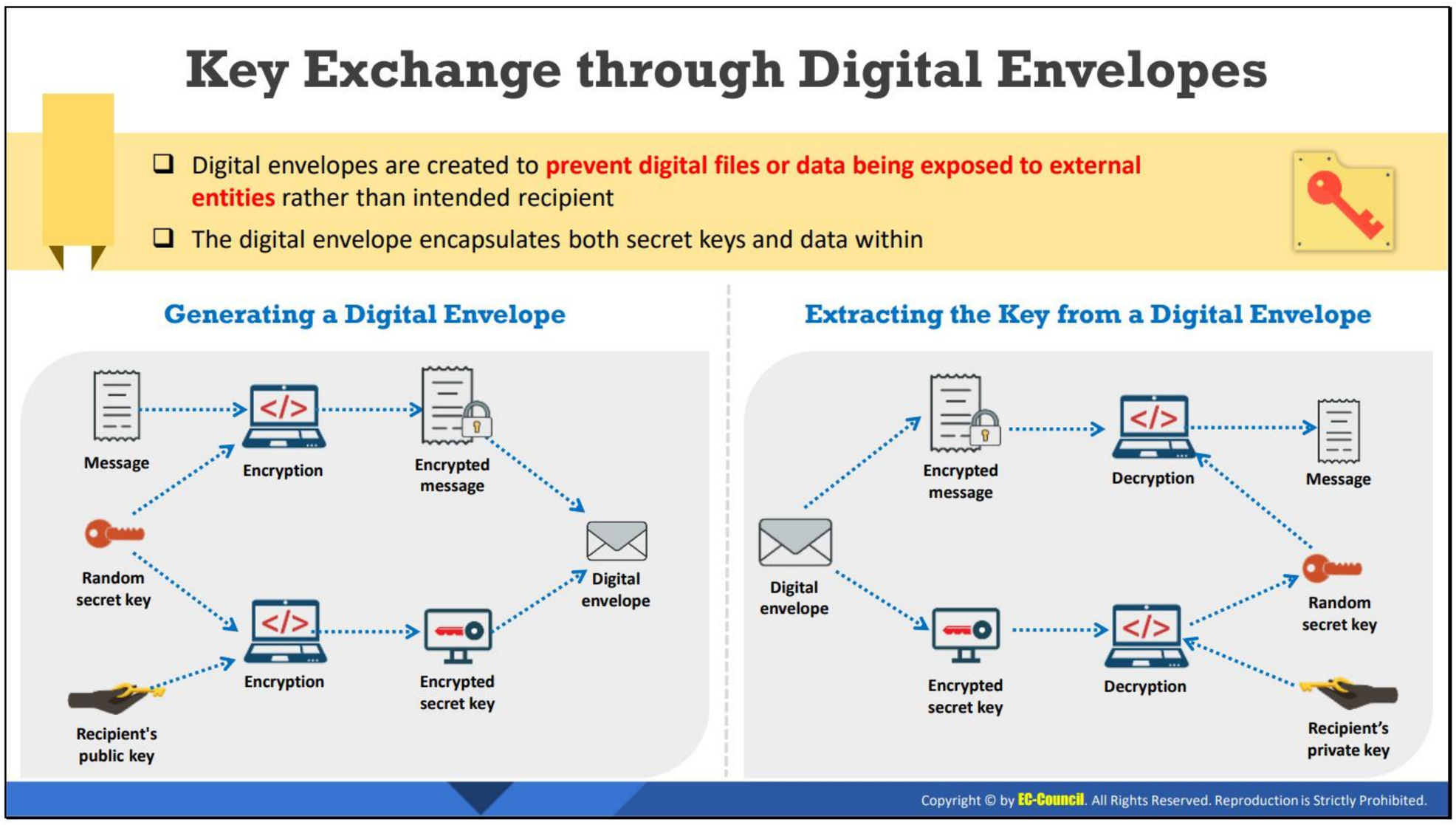


Figure 14.21: Working of Digital Signature

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Key Exchange through Digital Envelopes

Digital envelopes are created to prevent the exposure of digital files or data to external entities instead of the intended recipient. The symmetric key algorithm is generally used for encrypting/decrypting large amounts of data, but sharing the secret keys is a complex process. The asymmetric key algorithm can make this procedure easy, but it can encrypt/decrypt only a limited amount of data. To overcome these restrictions and improve security, the digital envelope system is used, which encapsulates both secret keys and the data within.

To generate a digital envelope, the sender must obtain the public key belonging to the recipient. The public key can be obtained based on the PKI implementation of the organization.

The process of key exchange through digital envelopes is as follows. Client A generates a new symmetric key (known as randomly generated secret key) and uses it to encrypt a message. Client A uses Client B's public key to encrypt this symmetric key. The encrypted message and secret key are both placed under a single packet known as a digital envelope and transferred to Client B.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

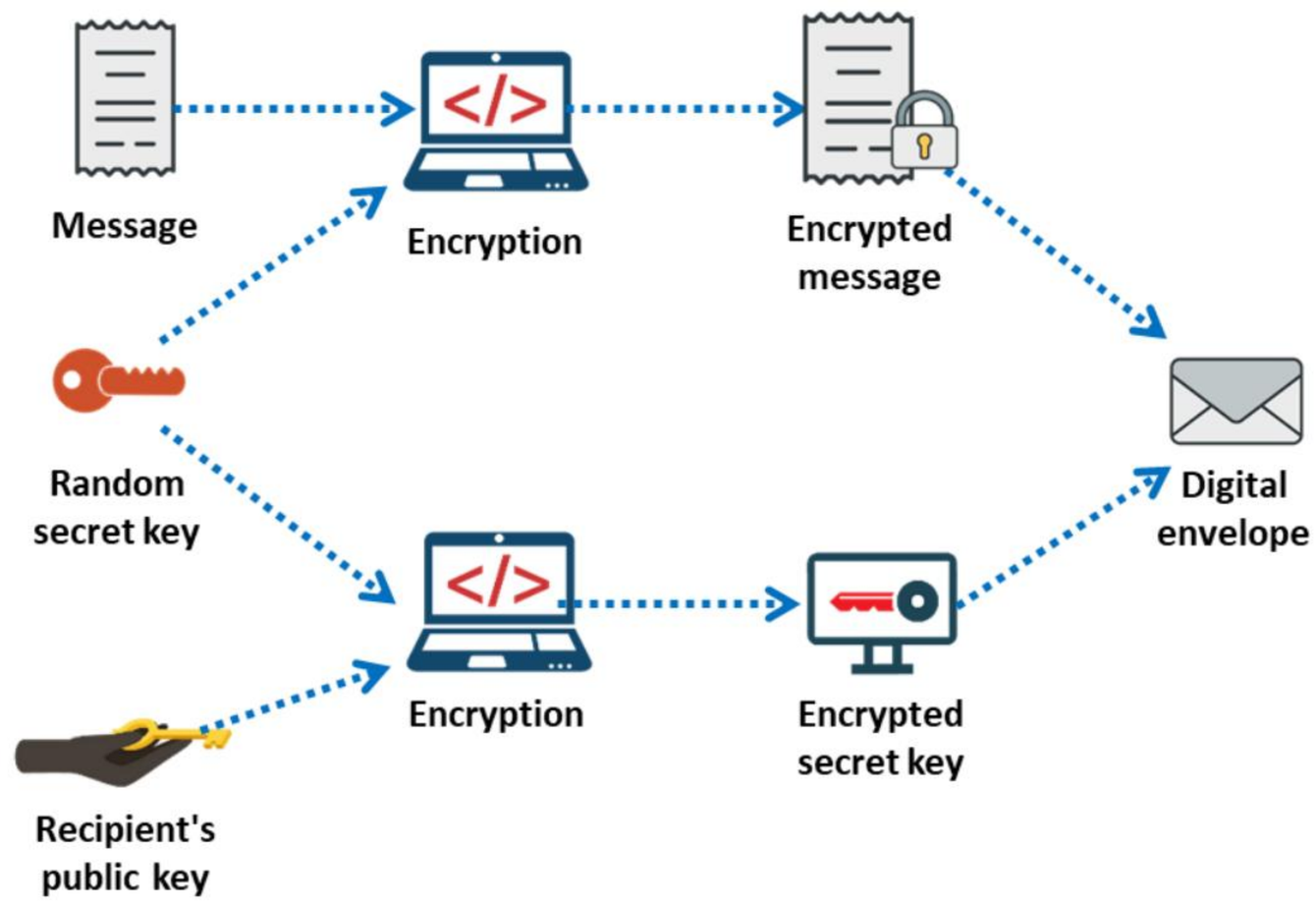


Figure 14.22: Generating a digital envelope

On the receiving end, Client B uses their private key to decrypt the symmetric key. Using the symmetric key, Client B then decrypts the original message. The digital envelope facilitates the secure exchange of the symmetric key between both parties through public key cryptography, which is also known as the hybrid encryption model.

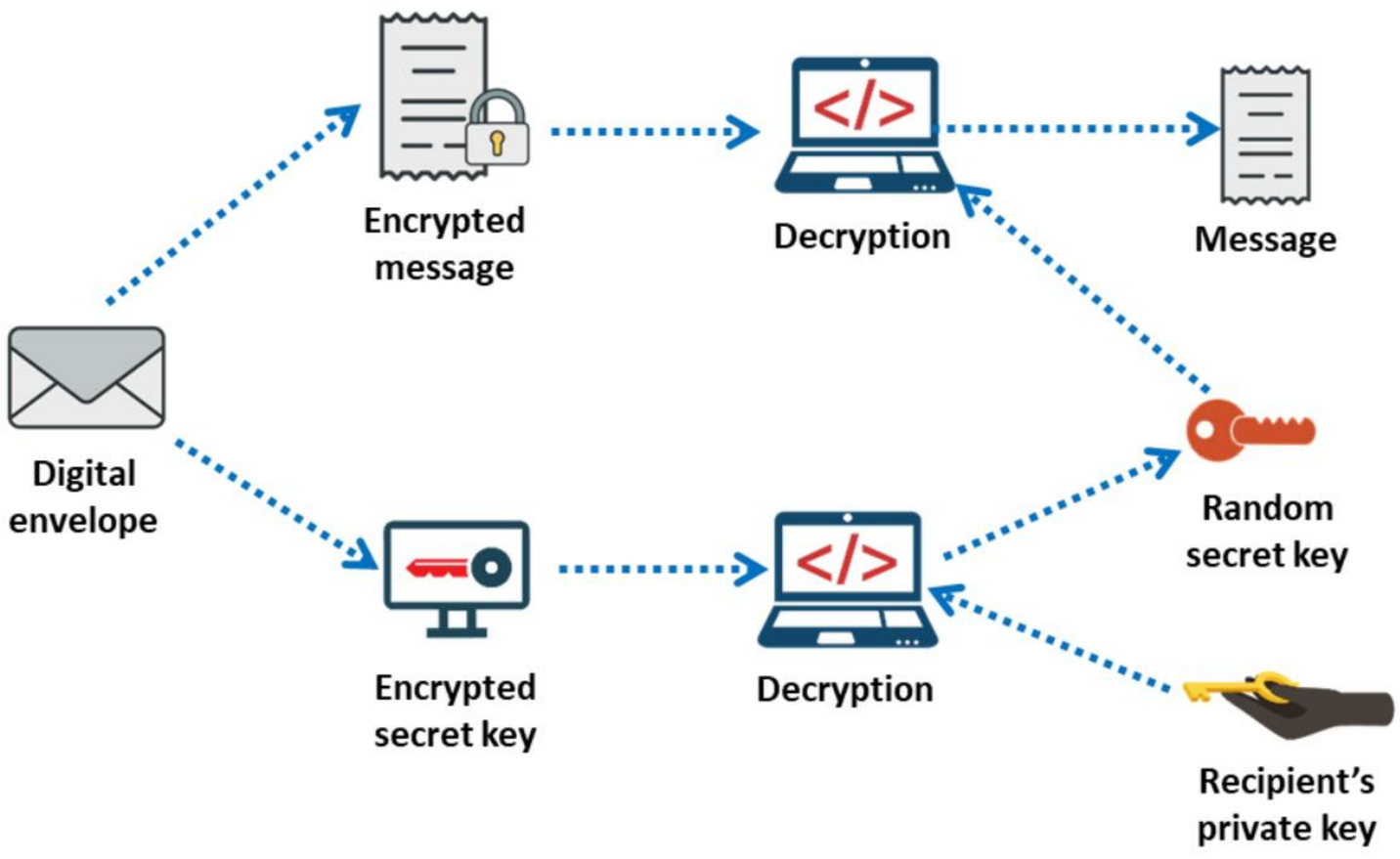
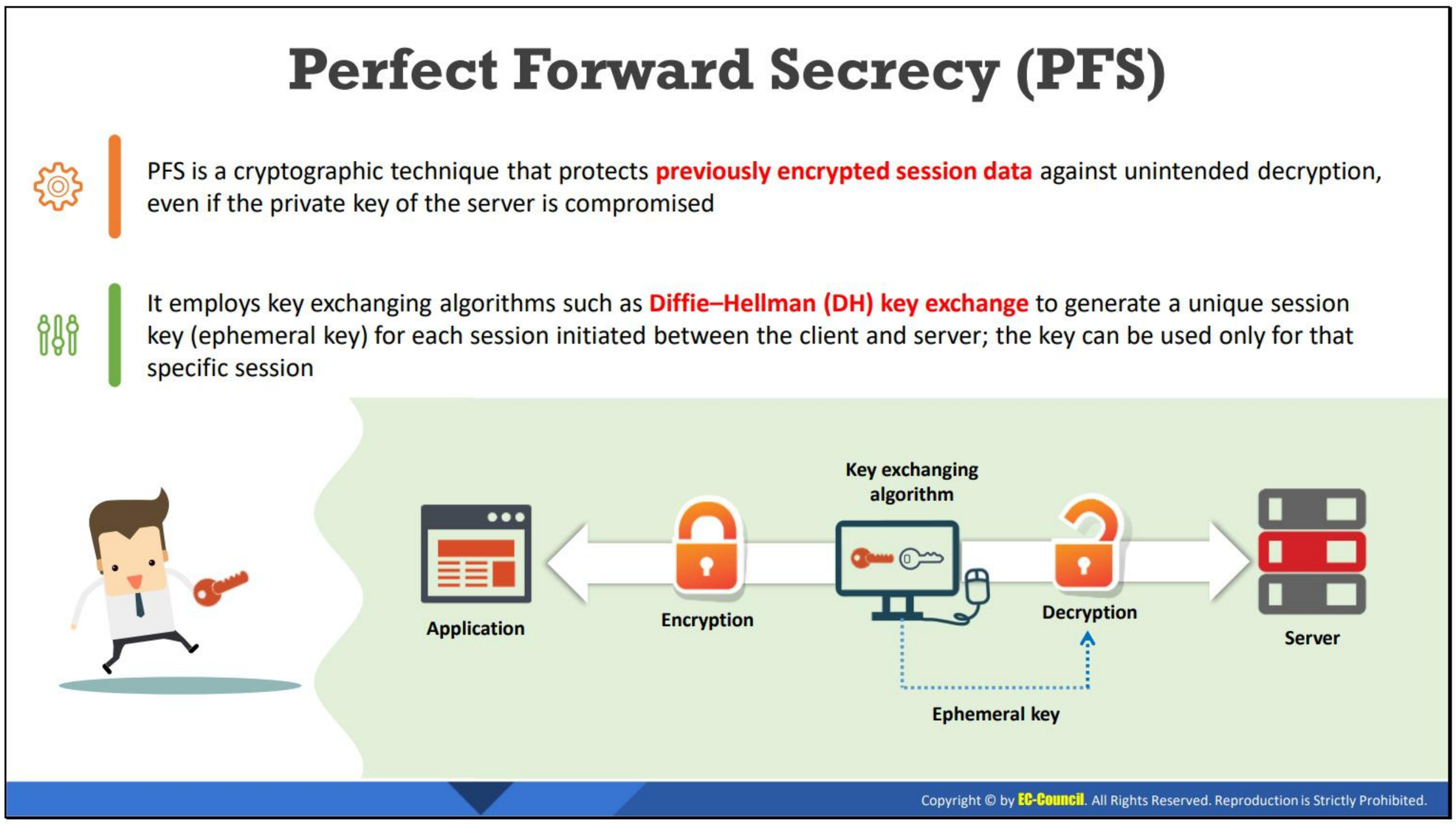


Figure 14.23: Extracting the key from a digital envelope

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Perfect Forward Secrecy (PFS)

In a digital envelope system, both the client and server exchanges secret keys using the RSA key pair of the server. If an attacker can compromise the private key of the server, then the confidential session data can be decrypted easily. To overcome risks associated with server-side RSA key exchange, perfect forward secrecy (PFS) is used. PFS is a cryptographic technique that protects previously encrypted session data against unintended decryption, even if the private key of the server is compromised. It employs key exchanging algorithms such as Diffie–Hellman (DH) key exchange to generate a unique session key (ephemeral key) for each session initiated between the client and server; the key can be used only for that specific session. In this manner, if the most recent key from the session is compromised, the rest of the data remains safe. Only the data protected by that particular key is susceptible to the attack. Web applications, messaging apps, and online voice calling applications use PFS, which changes the secret key in each conversation or each time the web application is refreshed.

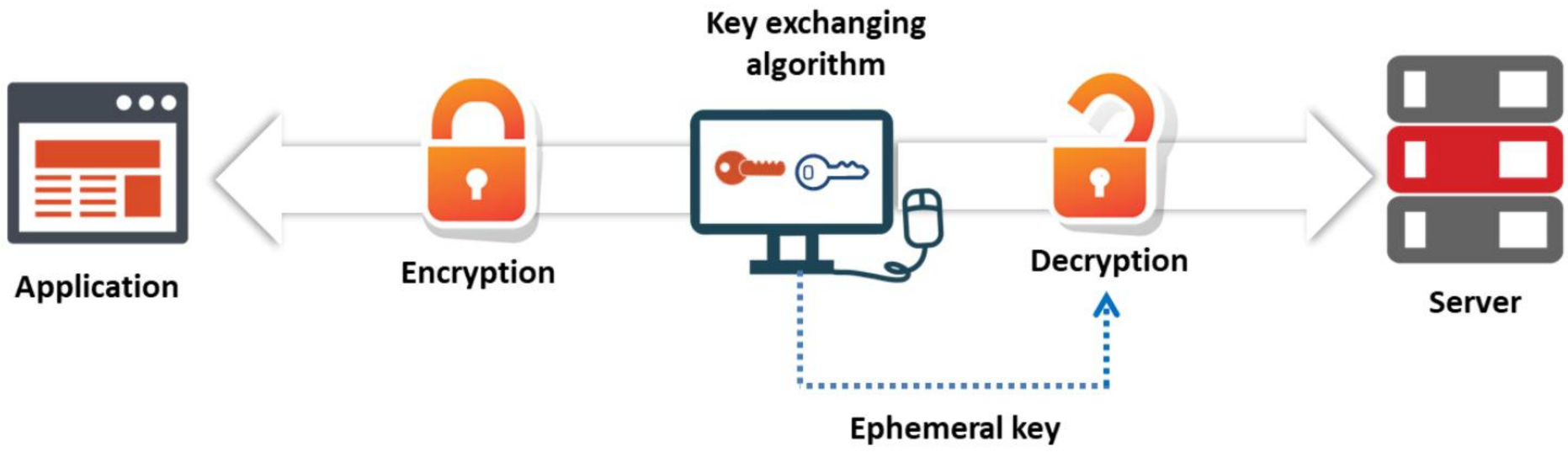
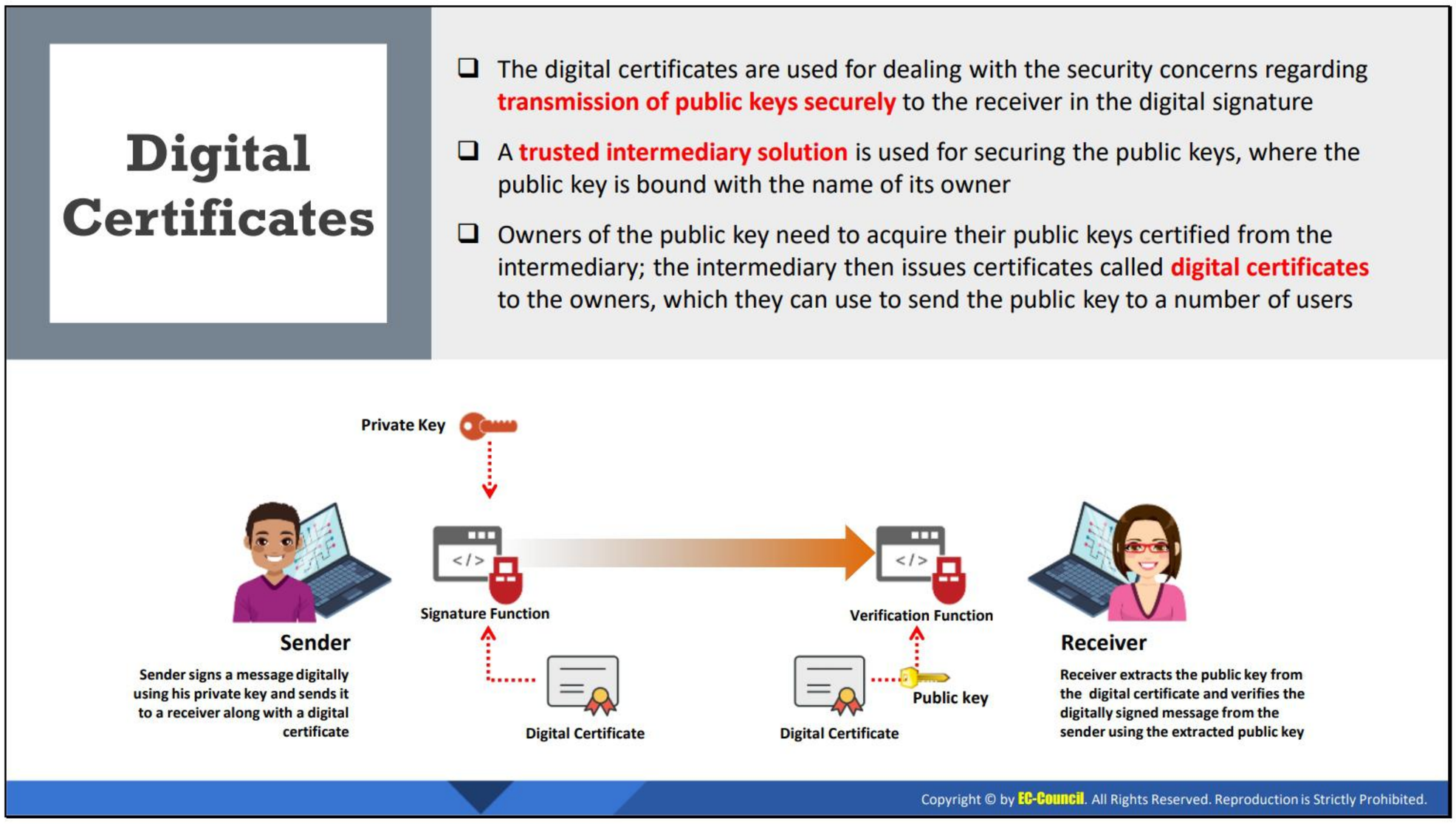


Figure 14.24: Perfect forward secrecy (PFS)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Digital Certificates

Digital certificates allow a secure exchange of information between a sender and a receiver. This enables the use of a public key by the sender to the receiver. A trusted intermediary solution is used for securing the public keys, where the public key is bound with the name of its owner. Owners of the public key need to acquire their public keys certified from the intermediary; the intermediary then issues certificates called digital certificates to the owners, which they can use to send the public key to a number of users.

The sender applies for a digital certificate from the certificate authority (CA). Along with the encrypted message and the public key, the CA provides other identity validating information. The receiver accepts the encrypted message and uses the CA's public key to decode the digital certificate. This allows the receiver to identify the digital signature and obtain the sender's public key and other identification details.

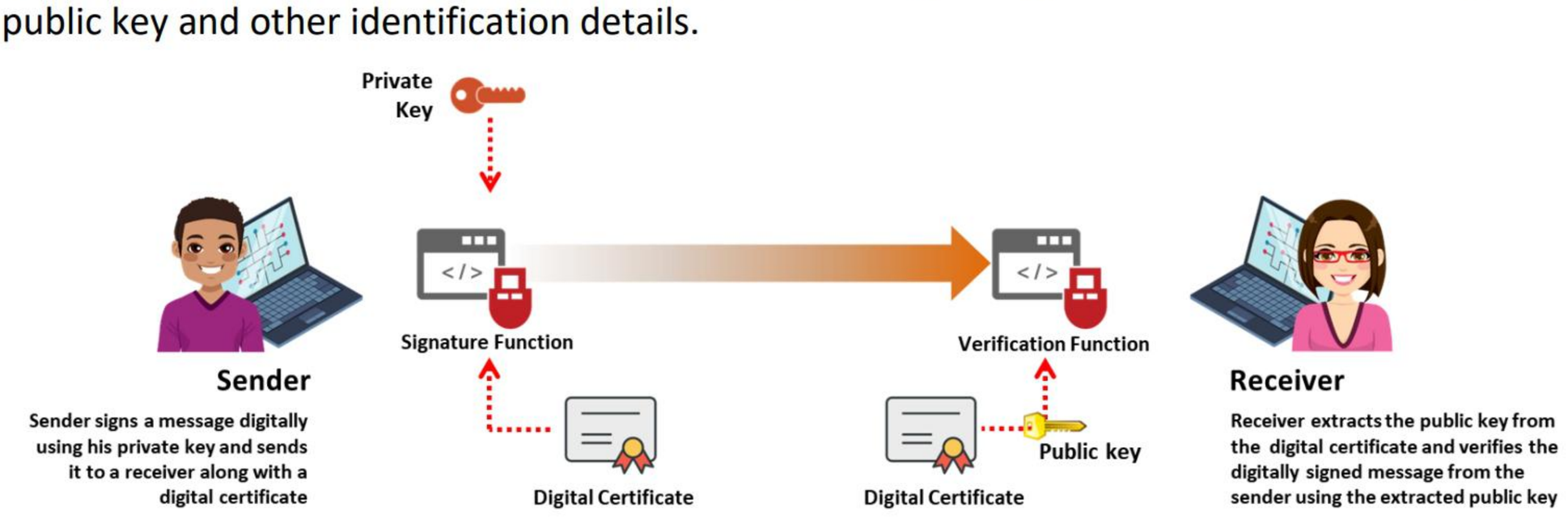


Figure 14.25: Working of digital certificates

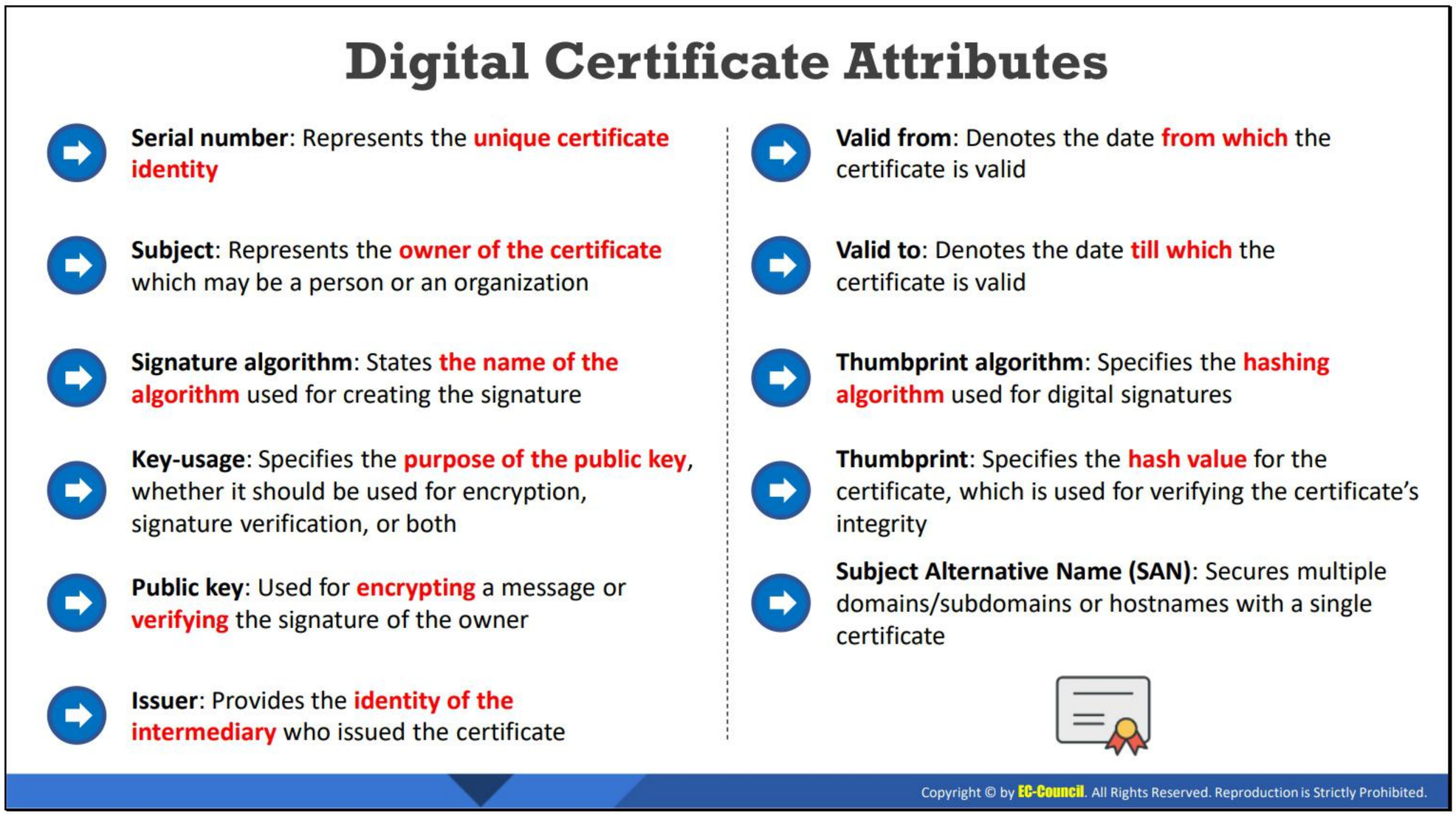
A digital certificate can hold information such as the name of the sender who applied for the certificate, expiration date, and a copy of the sender's public key digital signature of the CA. The

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

receivers who receive the digital certificate can check the validity of the certificate using the signature attached from the approved authorities using the private key of the authority. Each OS and web browser carries authorized certificates from the CA which enables easy validation. The main aim of implementing a digital certificate is to ensure nonrepudiation.

Most of the secure sockets layer (SSL)/ transport layer security (TLS) protocols use certificates in order to prevent attackers from changing or modifying the data. Digital certificates are used in email servers and code signing.

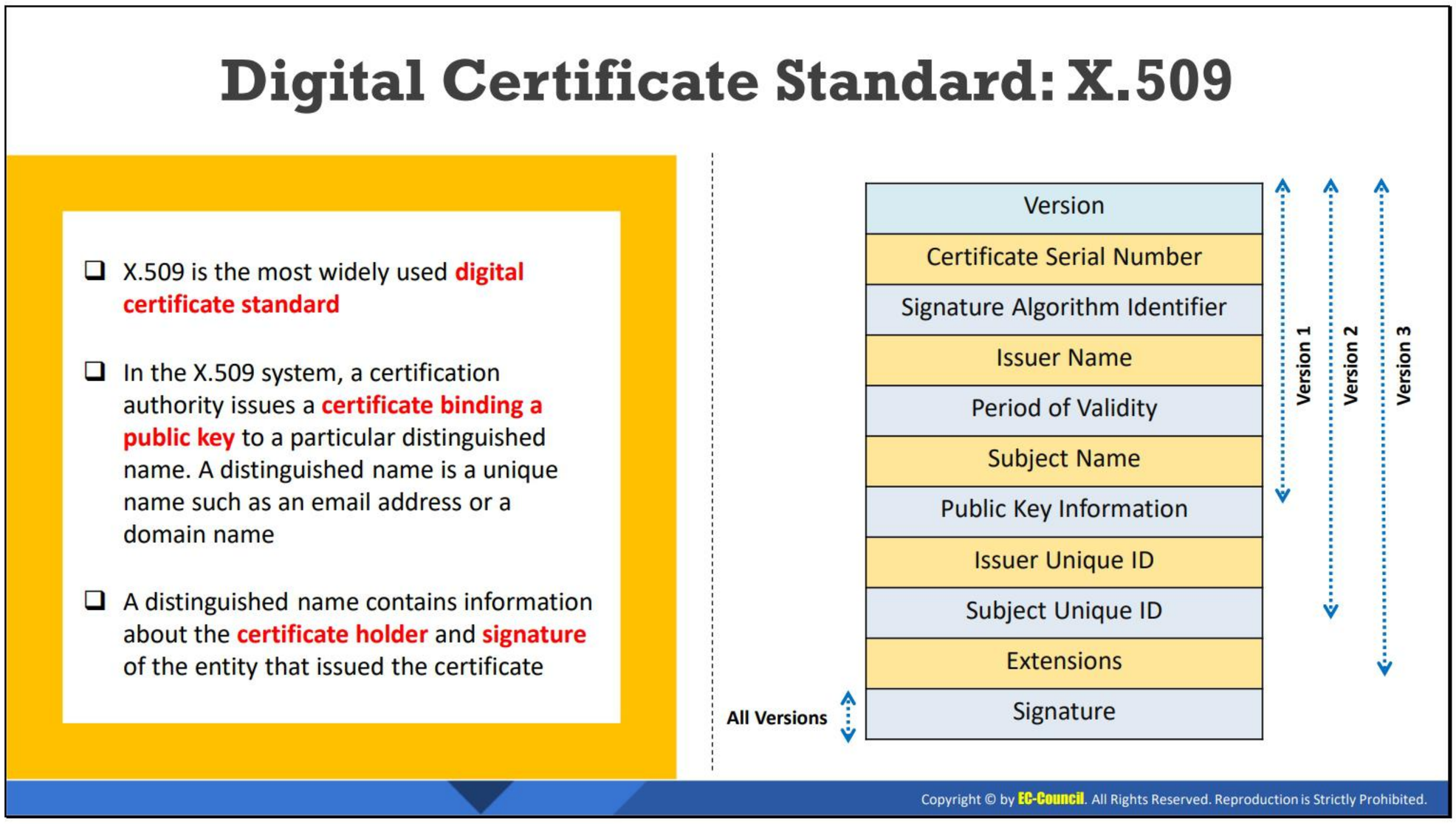
Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Digital Certificate Attributes

- **Serial number**: Represents the unique certificate identity.
- **Subject**: Represents the owner of the certificate which may be a person or an organization.
- **Signature algorithm**: States the name of the algorithm used for creating the signature.
- **Key-usage**: Specifies the purpose of the public key, whether it should be used for encryption, signature verification, or both.
- **Public key**: Used for encrypting a message or verifying the signature of the owner.
- **Issuer**: Provides the identity of the intermediary who issued the certificate.
- **Valid from**: Denotes the date from which the certificate is valid.
- **Valid to**: Denotes the date till which the certificate is valid.
- **Thumbprint algorithm**: Specifies the hashing algorithm used for digital signatures.
- **Thumbprint**: Specifies the hash value for the certificate, which is used for verifying the certificate's integrity.
- **Subject Alternative Name (SAN):** SAN is also known as a multi-domain SSL certificate. It can secure multiple domains/subdomains or hostnames with a single certificate. It can also secure websites, intranet, email servers, etc., without dealing with individual certificates.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Digital Certificate Standard: X.509

X.509 is the most widely used digital certificate standard. In the X.509 system, a certificate authority (CA) issues a certificate binding a public key to a particular distinguished name. A distinguished name is a unique name such as an email address or a domain name. It contains information about the certificate holder and signature of the entity that issued the certificate.

X.509 is a standard that defines the structure of a digital certificate. The data fields that should be included in an SSL certificate are defined by this standard. These certificate formats are defined by Abstract Syntax Notation One (ASN.1), which is an ISO format used to accomplish interoperability among platforms. Certificate files have distinct extensions depending on the format and encoding used. X.509 is a widely used digital certificate structure, and version 3 of this standard is currently in use.

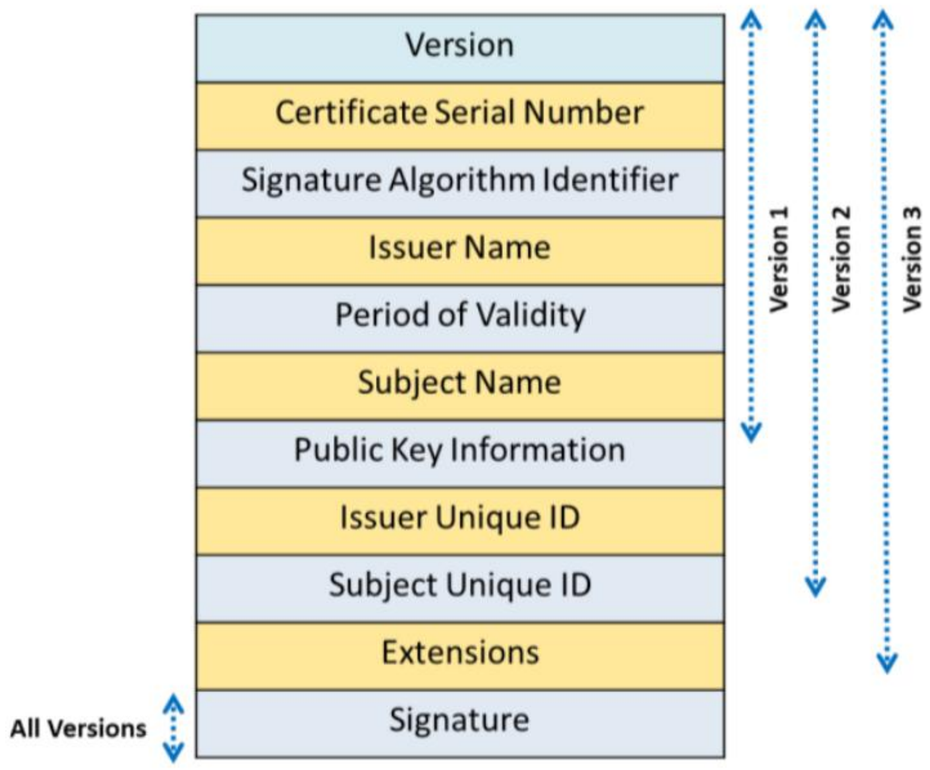


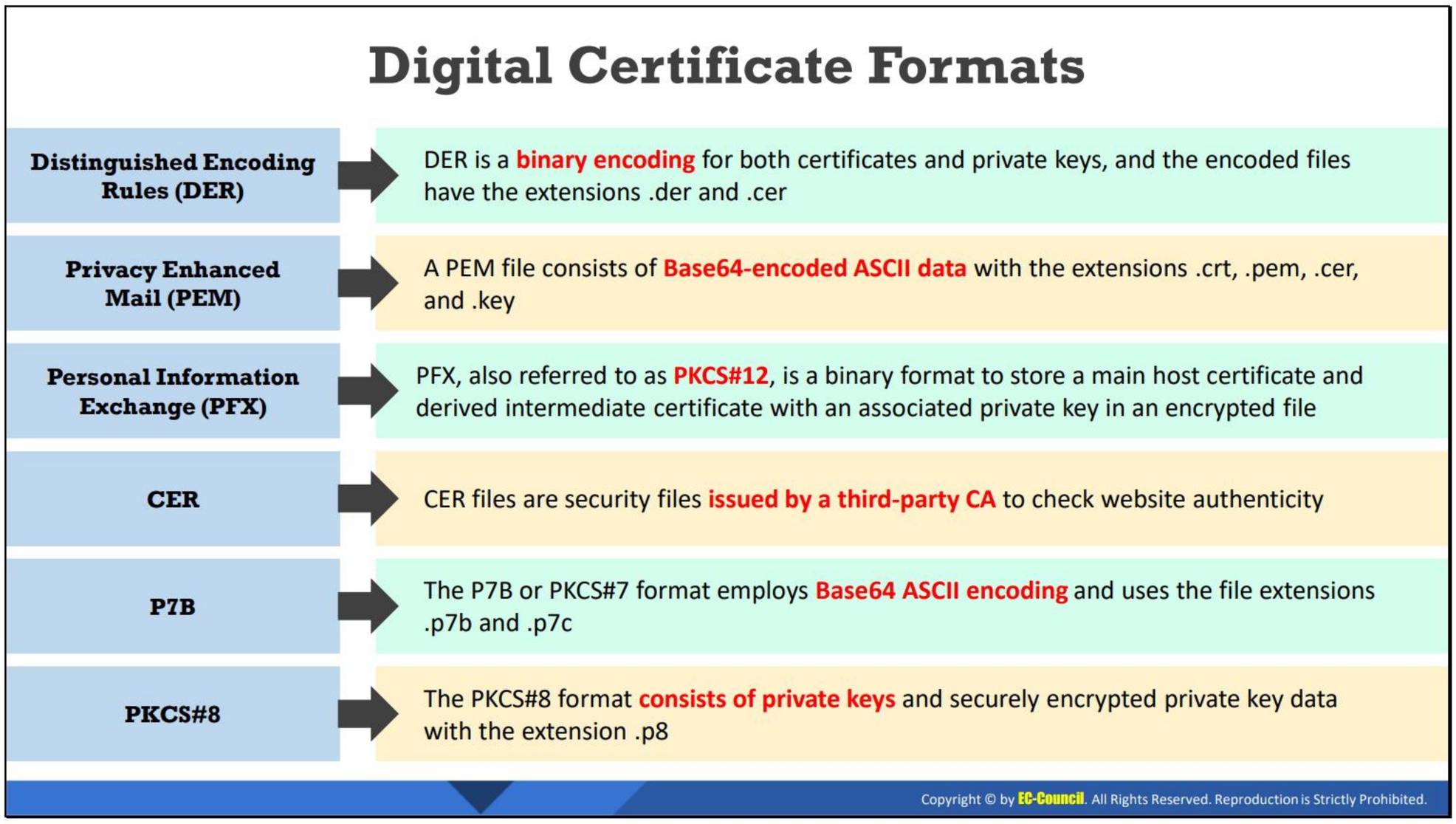
Figure 14.26: X.509 digital certificate format

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The following are the basic fields included in the format.

- **Version**: This field specifies the version number of the certificate; its value can be 1, 2, or 3.
- **Certificate serial number**: It is a distinct positive number assigned for each certificate and is assigned by the issuer to identify the certificate.
- **Signature algorithm identifier**: It indicates the algorithm that the issuer uses for signing the certificate.
- **Issuer name**: It indicates the X.500 distinguished name of the trusted third party that signed and issued the certificate.
- **Period of validity**: This field indicates the dates from and till which the certificate is valid.
- **Subject name**: It is the name of the entity that owns the certificate. It can be CA, RA, a person, or a company.
- **Public key information**: This field contains the public key of the subject and the corresponding algorithm identifier.
- **Issuer unique ID**: It is the unique identifier used to facilitate the reuse of the issuer name over time.
- **Subject unique ID**: It is the unique identifier used to facilitate the reuse of the subject name over time.
- **Extension**: It is present in version 3 certificates and consists of an extension identifier, criticality flag, and extension value. The extension identifier specifies the format of the extension value and criticality flag, which indicates the importance level of the extension.
- **Signature**: It contains the issuer's digital signature, which is used for verifying the authenticity of the digital certificate.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Digital Certificate Formats

A digital certificate format comprises details related to the owner and authority of the certificate. Digital certificate files have distinct extensions depending on the format and encoding used.

Extension: Extensions are present in version 3 certificates and consist of an extension identifier, criticality flag, and extension value, all of which are conventional file extensions and not standard. Extensions often do not provide information about the type of encoding used or type of content in the certificate file. It can be viewed only by opening a text editor.

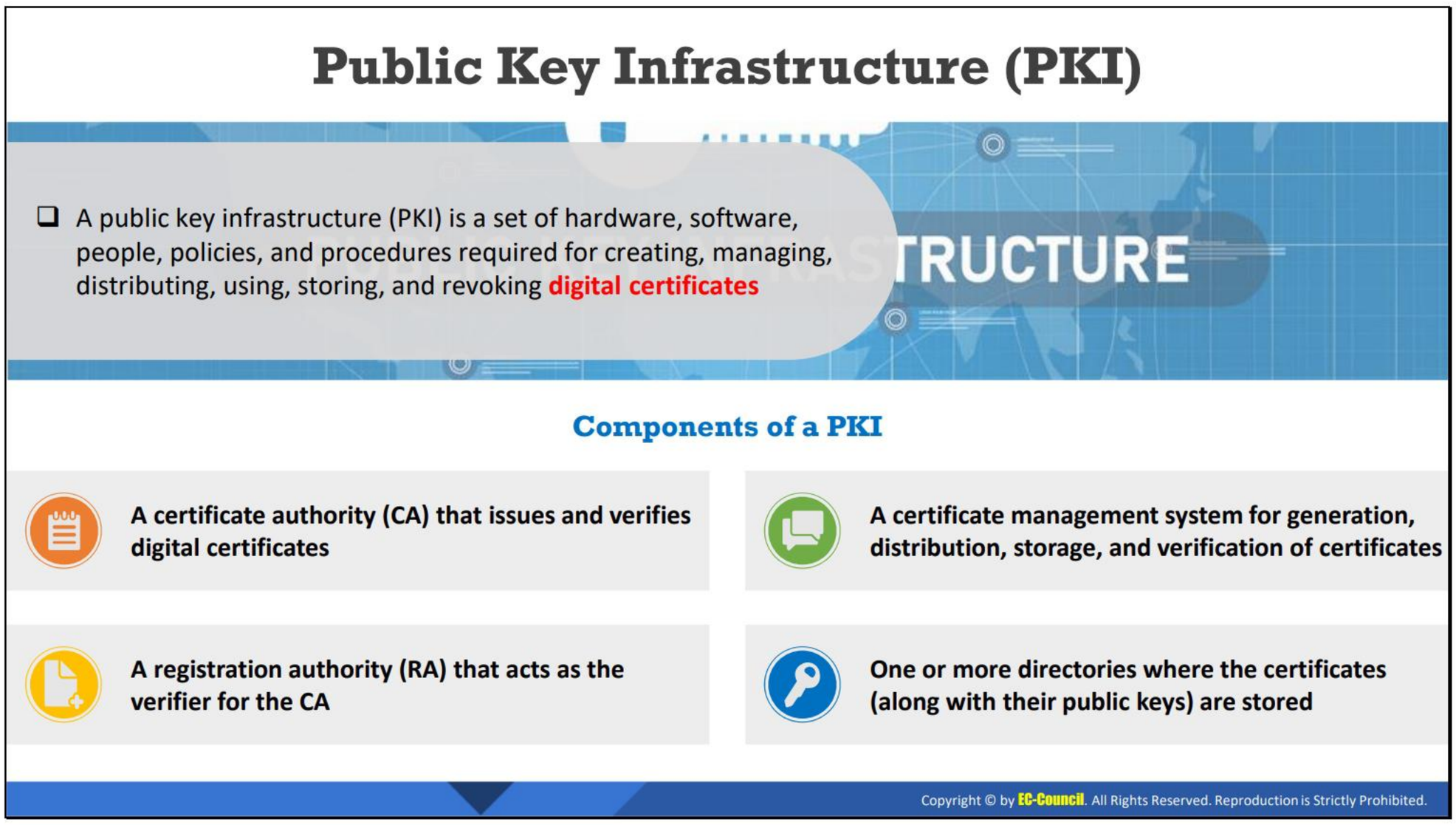
The following are some of the types of certificate formats:

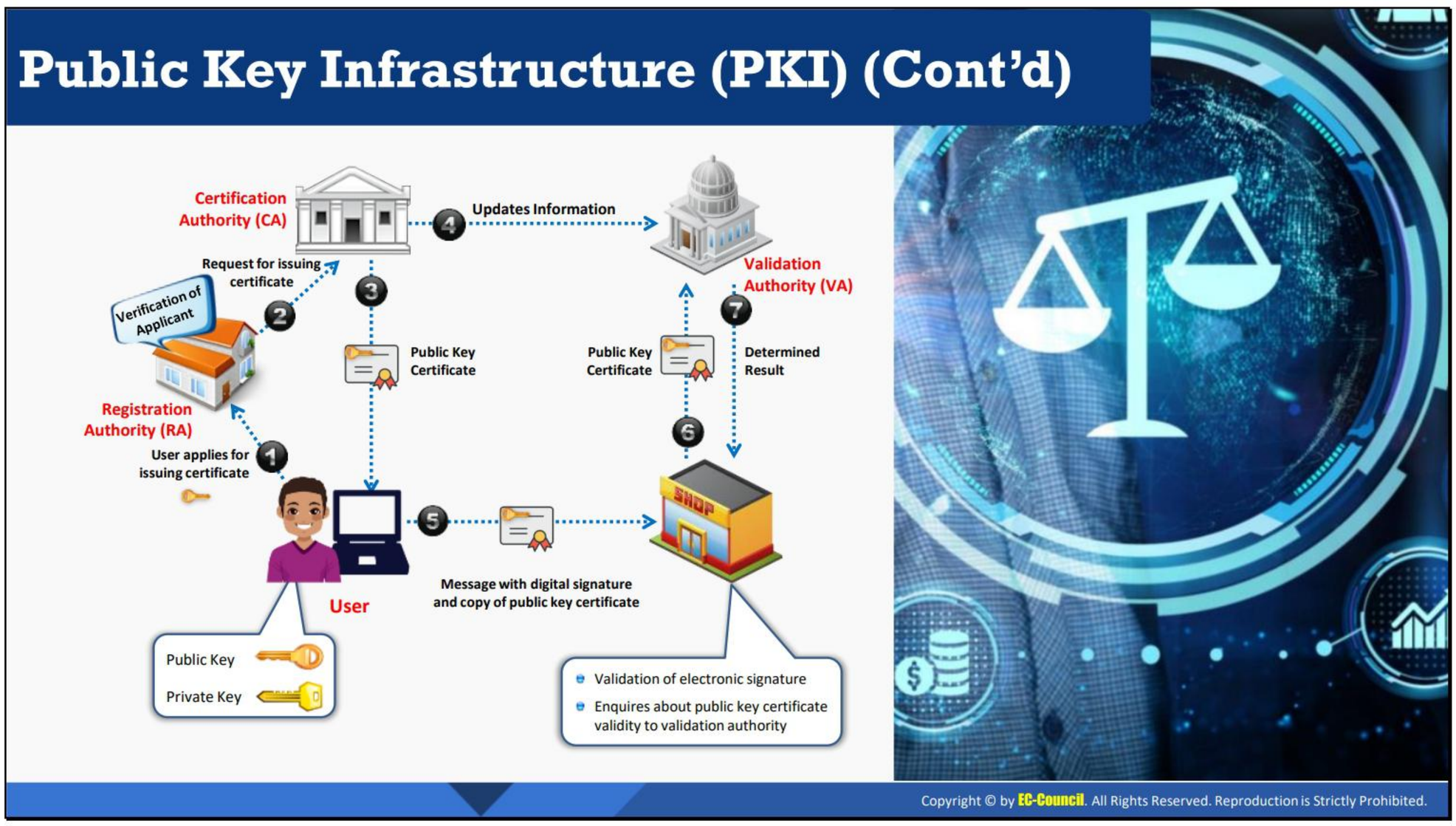
- **Distinguished Encoding Rules (DER)**: DER is a binary encoding for both certificates and private keys, and the encoded files have the extensions .der and .cer. A DER file can be read with a text editor. These files are commonly seen in Java platforms.
- **Privacy Enhanced Mail (PEM):** It is a common file format used by CAs to provide certificates. A PEM file consists of Base64-encoded ASCII data with the extensions .crt, .pem, .cer, and .key. The ASCII format data contains a plaintext header and footer such as the lines "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----." These files are commonly seen in Linux/Unix platforms.
- **Personal Information Exchange (PFX):** PFX, also referred to as PKCS#12, is a binary format used to store a main host certificate and derived intermediate certificate with an associated private key in an encrypted file with the extensions .pfx and .p12. PFX follows Public Key Cryptography Standards (PKCS). In Windows platforms, it commonly uses the extension .pfx, and in iOS platforms, it uses .p12. It can be employed to import/export digital certificates along with private keys.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **CER:** CER files are security files issued by a third-party CA to check website authenticity. These files have the extension .cer. CER certificates are configured on a web server to set the validity of a specific website hosted on the server. These files can be seen by clicking on the lock symbol at the browser's edge while visiting a secure website. When a user enters the URL to a website hosted on the same server, the browser verifies the validity of the website using the certificate attached to it.
- **P7B:** P7B or PKCS#7 is a format that employs Base64 ASCII encoding and uses the file extensions .p7b and .p7c. It comprises only main certificates and intermediate (chain) certificates but not the private key. It contains header and footer statements such as "--BEGIN PKCS--" and "--END PKCS7--". It is commonly seen in Microsoft Windows and Java Tomcat.
- **PKCS#8:** This format consists of private keys and securely encrypted private key data with the extension .p8. Data are stored in the Base64 encoded form, generally using the DER or PEM structure, and then encrypted. The header of the PKCS8 syntax is "-----BEGIN PRIVATE KEY-----", and the footer is END PRIVATE KEY-----". If a private key is encrypted, then the header is "-----BEGIN ENCRYPTED PRIVATE KEY-----", and the footer is "-----END ENCRYPTED PRIVATE KEY-----". PKCS#8 is the recognized syntax to store private key information.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.





Public Key Infrastructure (PKI)

A public key infrastructure (PKI) is a security architecture developed for increasing the confidentiality of the information exchanged over the Internet. It includes hardware, software, people, policies, and procedures required for creating, managing, distributing, using, storing, and revoking digital certificates. In cryptography, a PKI helps to bind the public keys with the corresponding user identities by means of a CA.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

PKI is a comprehensive system that allows the use of public-key encryption and digital signature services across a wide variety of applications. PKI authentication depends on digital certificates (also known as public-key certificates) that CAs sign and provide. A digital certificate is a digitally signed statement with a public key and the subject's (i.e., a user, company, or system) name on it.

The components of a PKI include,

- A certificate authority (CA) that issues and verifies digital certificates
- A registration authority (RA) that acts as the verifier for the CA.
- A certificate management system for generation, distribution, storage, and verification of certificates.
- One or more directories where the certificates (along with their public keys) are stored.

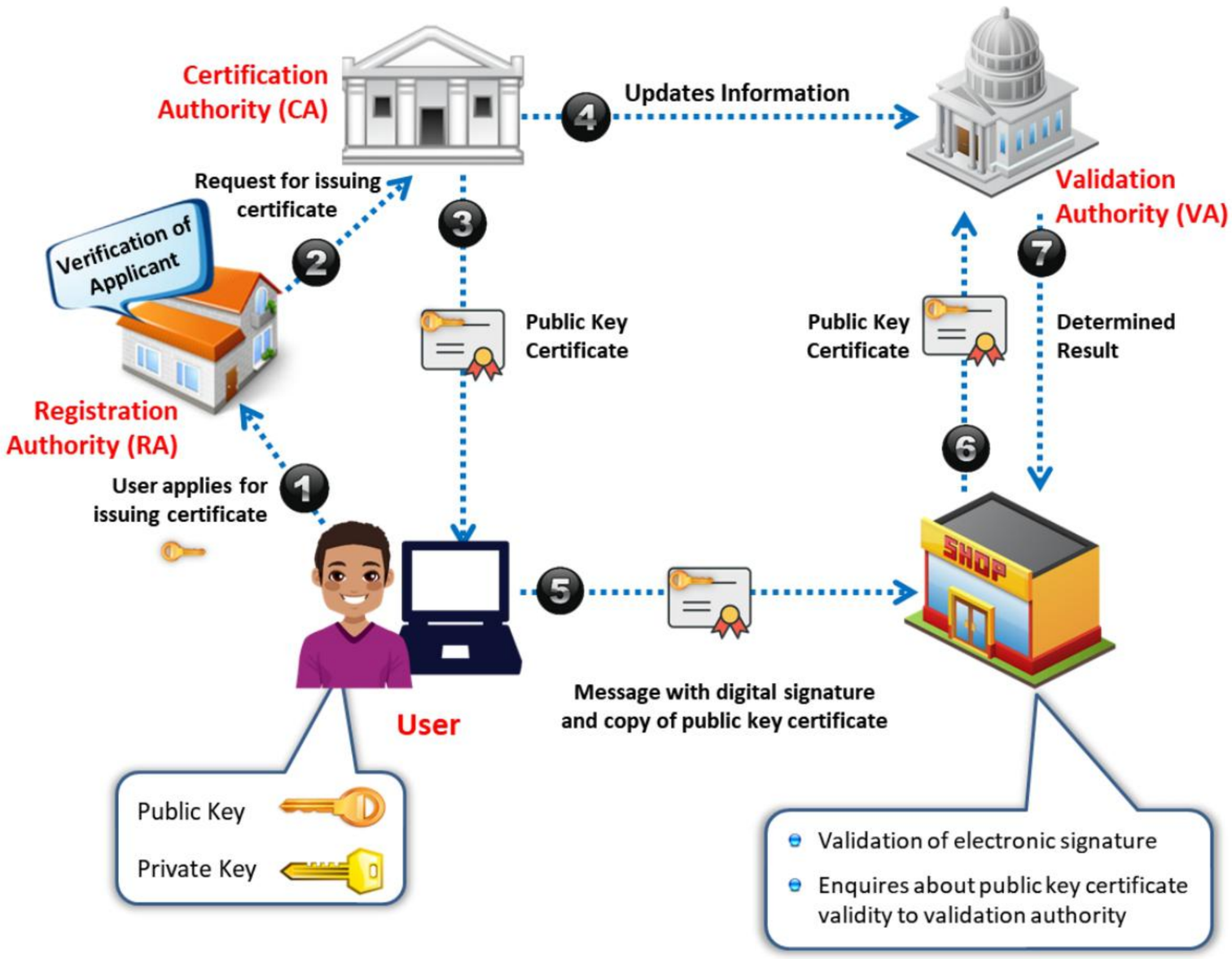


Figure 14.27: PKI Components

PKI is widely recognized as a best practice for ensuring digital verification for electronic transactions. These are the most effective methods for providing verification during electronic transactions. The digital signatures supported by PKI include the following:

- With whom you are dealing (identification)
- Who is authorized to access what information (entitlements)
- A verifiable record of the transaction (verification)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Uses of PKI

PKI does not serve only as a business function; it provides the foundation for other security services. The primary use of PKI is to allow the distribution and use of public keys and certificates with security. The security mechanisms that are based on PKI include email, chip card application, value exchange with e-commerce, home banking, and electronic postal systems. PKI enables basic security services for a variety of systems as listed below:

- It uses the SSL, internet protocol security (IPsec), and the hypertext transfer protocol secure (HTTPS) protocols for communication security.
- It uses the secure/multipurpose internet mail extensions (S/MIME) and pretty good privacy (PGP) protocols for email security.
- It uses the secure electronic transaction (SET) protocol for value exchange.

The following are the key benefits of PKI:

- It reduces the transactional processing expenses.
- It reduces risk.
- It improves the efficiency and performance of systems and networks.
- It reduces the difficulty of security systems with binary symmetrical methods.

Certificate Authority (CA) and Registration Authority (RA)

The primary role of CA is to digitally sign and publish the public key bound to a given user. It is an entity trusted by one or more users to manage certificates. Often, a registration authority (RA) is used for handling verification prior to the issue of certificates to reduce the burden of a CA. An RA acts as a proxy between the user and the CA. The RA receives a request, authenticates it, and forwards it to the CA.

An RA is a type of licensed certificate distributor or intermediary between users and the CA, and it performs CA activities such as accepting and processing certificate requests and checking identity and revoking credentials, if necessary. This authority can generate keys on behalf of users and can also be used for key backup and recovery.

Intermediate CA

Another way of issuing a certificate is using intermediary CAs that allow the hierarchical distribution of certificates, where the main CA represented as a root CA. Under the root CA, there may be many intermediate authorities (subordinate CAs), among which the load of certificate creation is distributed, and these intermediate authorities provide facilities for organizations in different regions to create and handle their own certificates. Creating intermediate CAs can also make the certificate revocation process easy. If an attacker managed to gain access to the key used on an intermediate CA, it is possible to revoke only the certificates under that specific CA.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

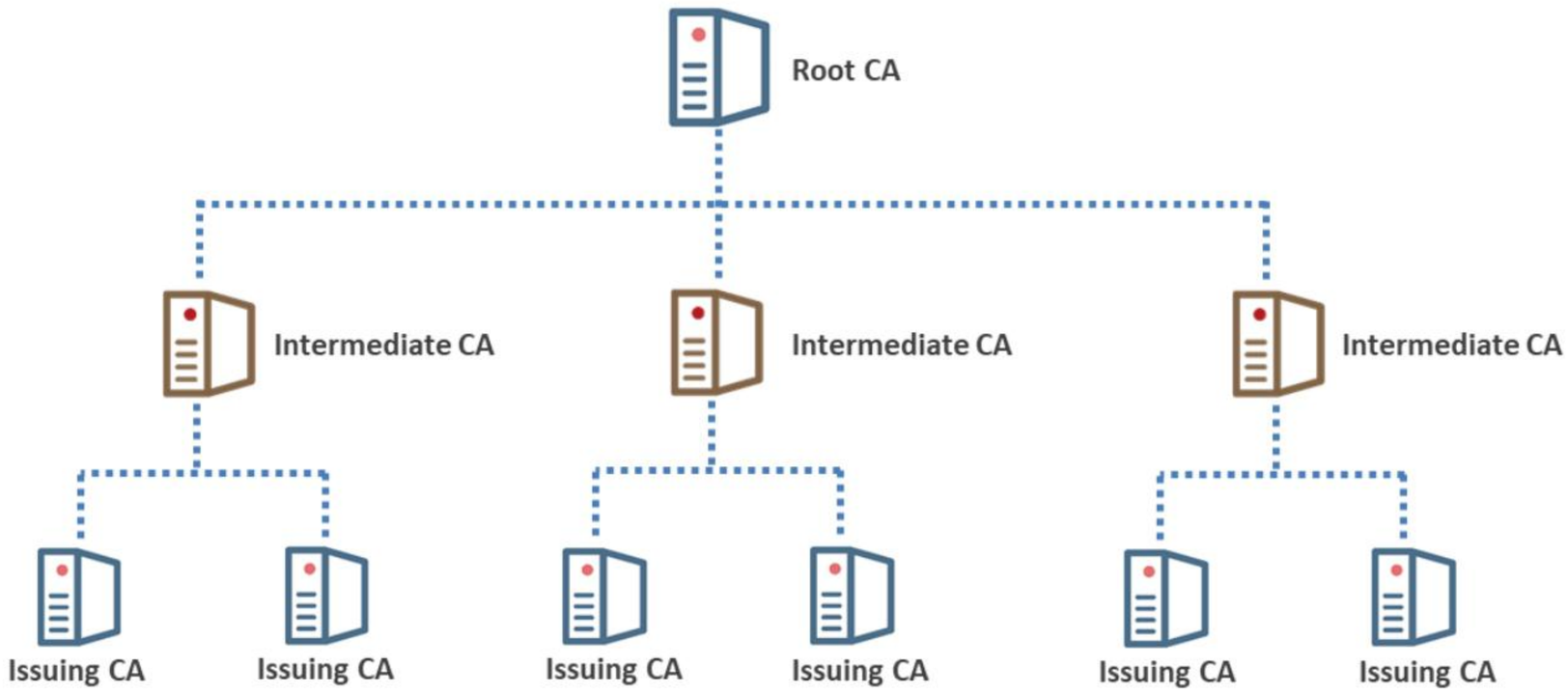
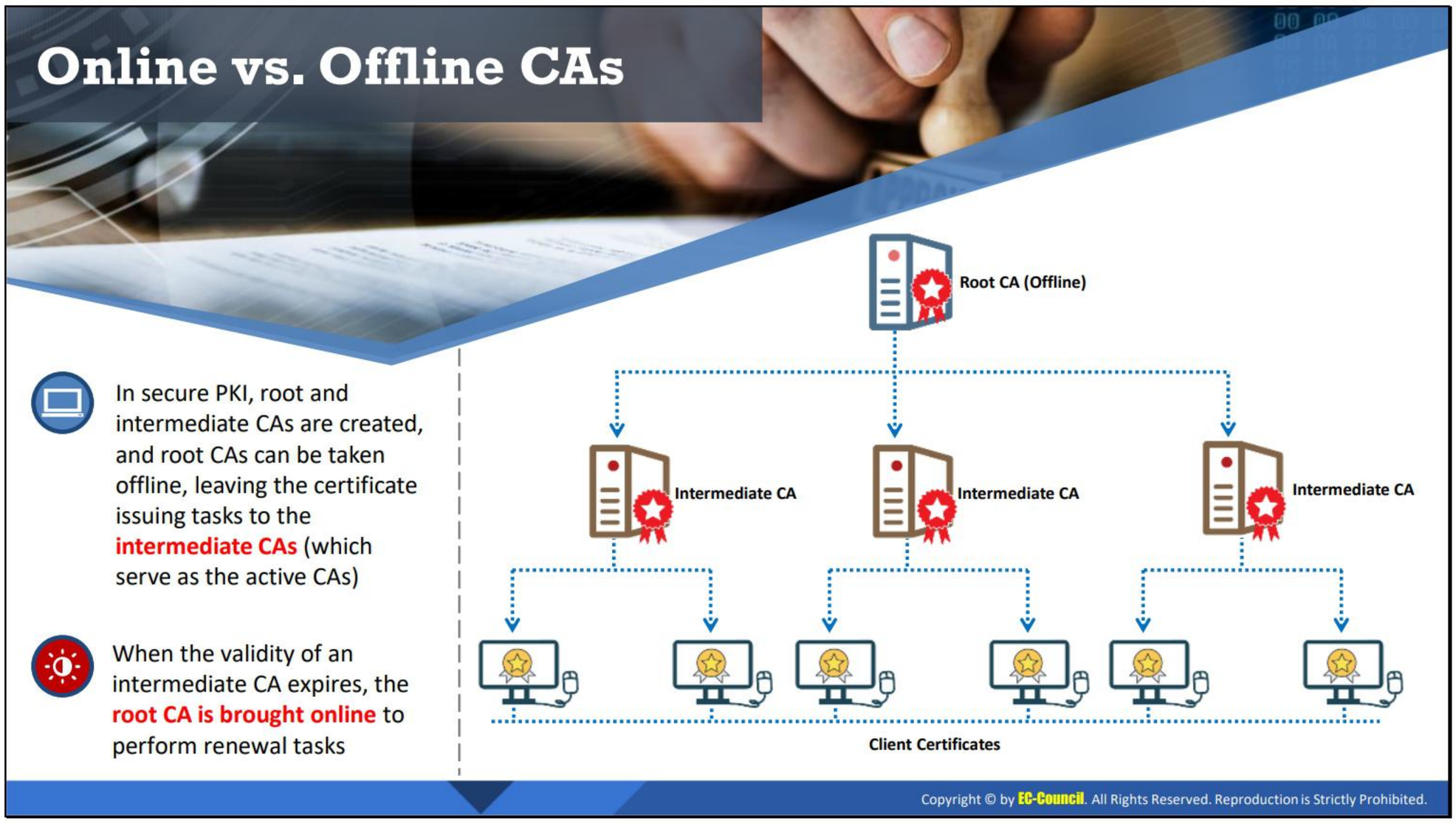


Figure 14.28: CA hierarchy

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Online vs. Offline CAs

The PKI model runs based on trust, which is usually established by the certificate authority (CA). When a CA is compromised, users will not trust any of the certificates published by that particular CA. This is the reason why root and subordinate (intermediate) CAs are created, where root CAs can be taken offline, leaving certificate issuing tasks to the intermediate CAs (serving as active CAs). This process can protect the master or root CA from being compromised. If an attacker compromises the intermediate CA, only the certificates issued by that specific subordinate/intermediate CA are impacted. When the validity of an intermediate CA expires, the root CA shall be brought online to perform renewal tasks. In this case, the root CA is not required to be a part of the trust domain; it only needs to serve in a workgroup.

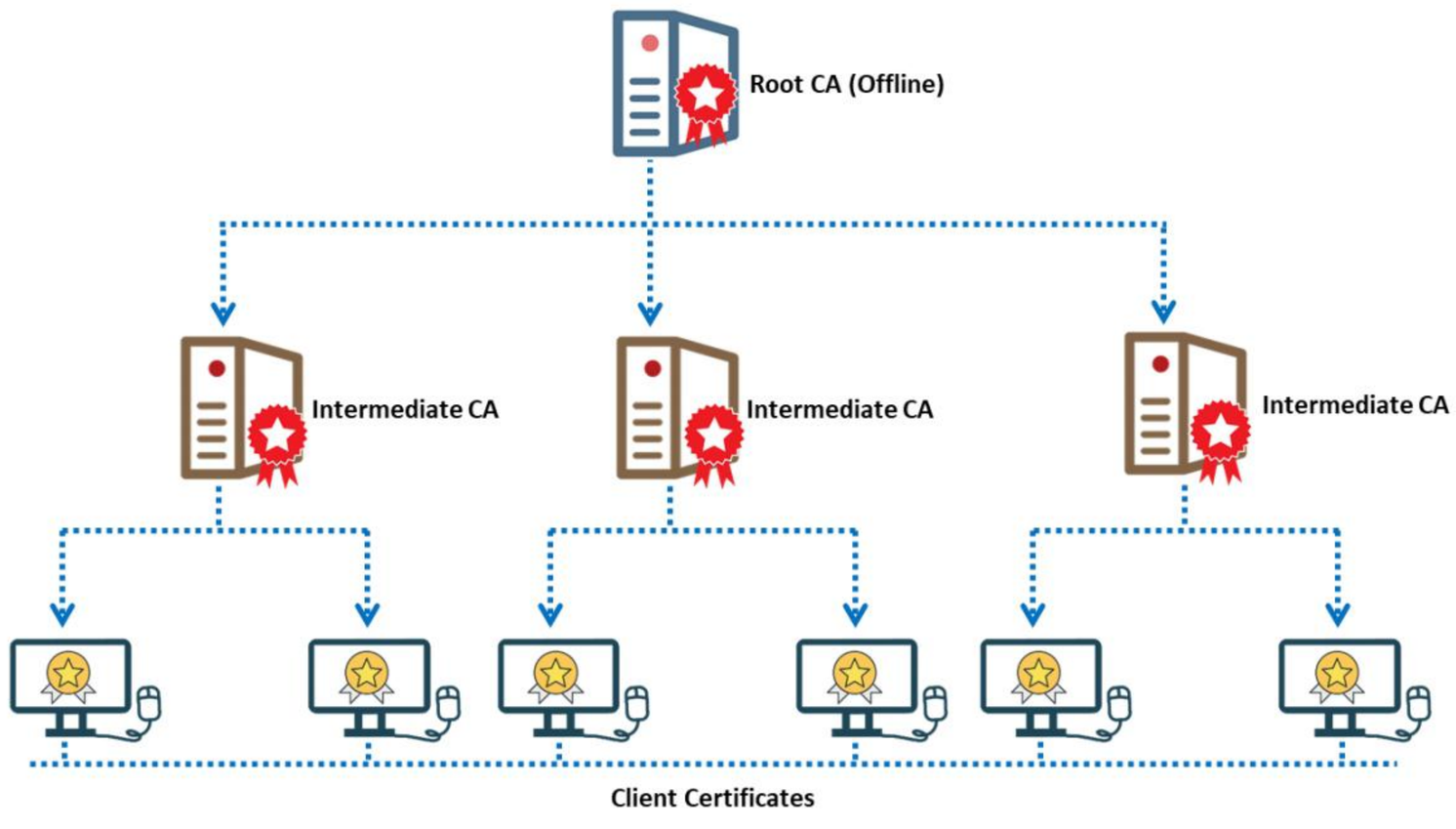
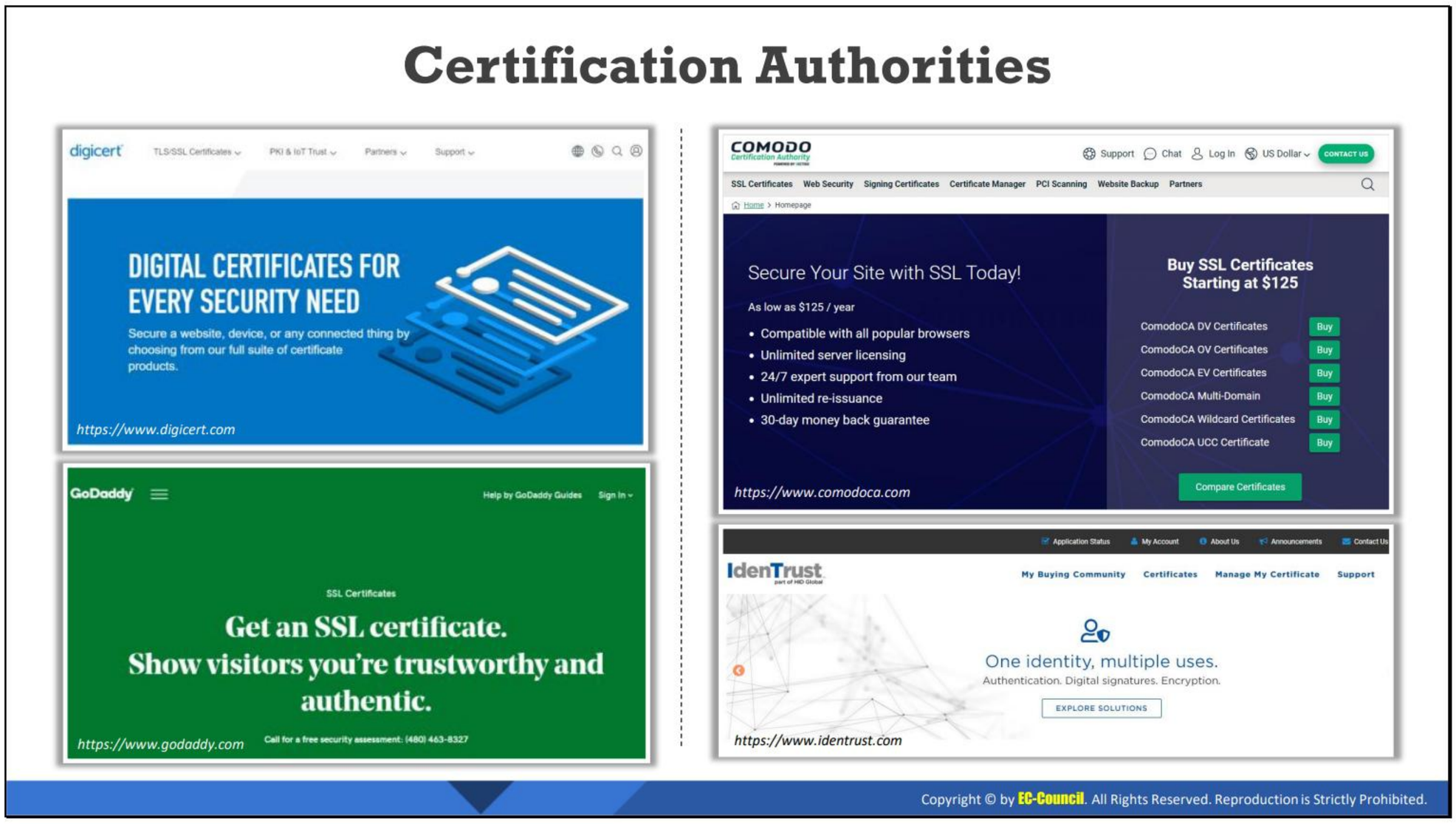


Figure 14.29: Online vs. offline CAs

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Certification Authorities

Certification authorities (CAs) are trusted entities that issue digital certificates. The digital certificate certifies the possession of the public key by the subject (user, company, or system) specified in the certificate. This aids others to trust signatures or statements made by the private key that is associated with the certified public key.

Some popular CAs are discussed below:

- **Comodo**

  Source: *https://www.comodoca.com*

  Comodo offers a range of PKI digital certificates with strong SSL encryption (128/256 available) with Server-Gated Cryptography (SGC). It ensures standards of confidentiality, system reliability, and pertinent business practices as judged via qualified independent audits. It offers PKI management solutions such as Comodo Certificate Manager and Comodo EPKI Manager.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

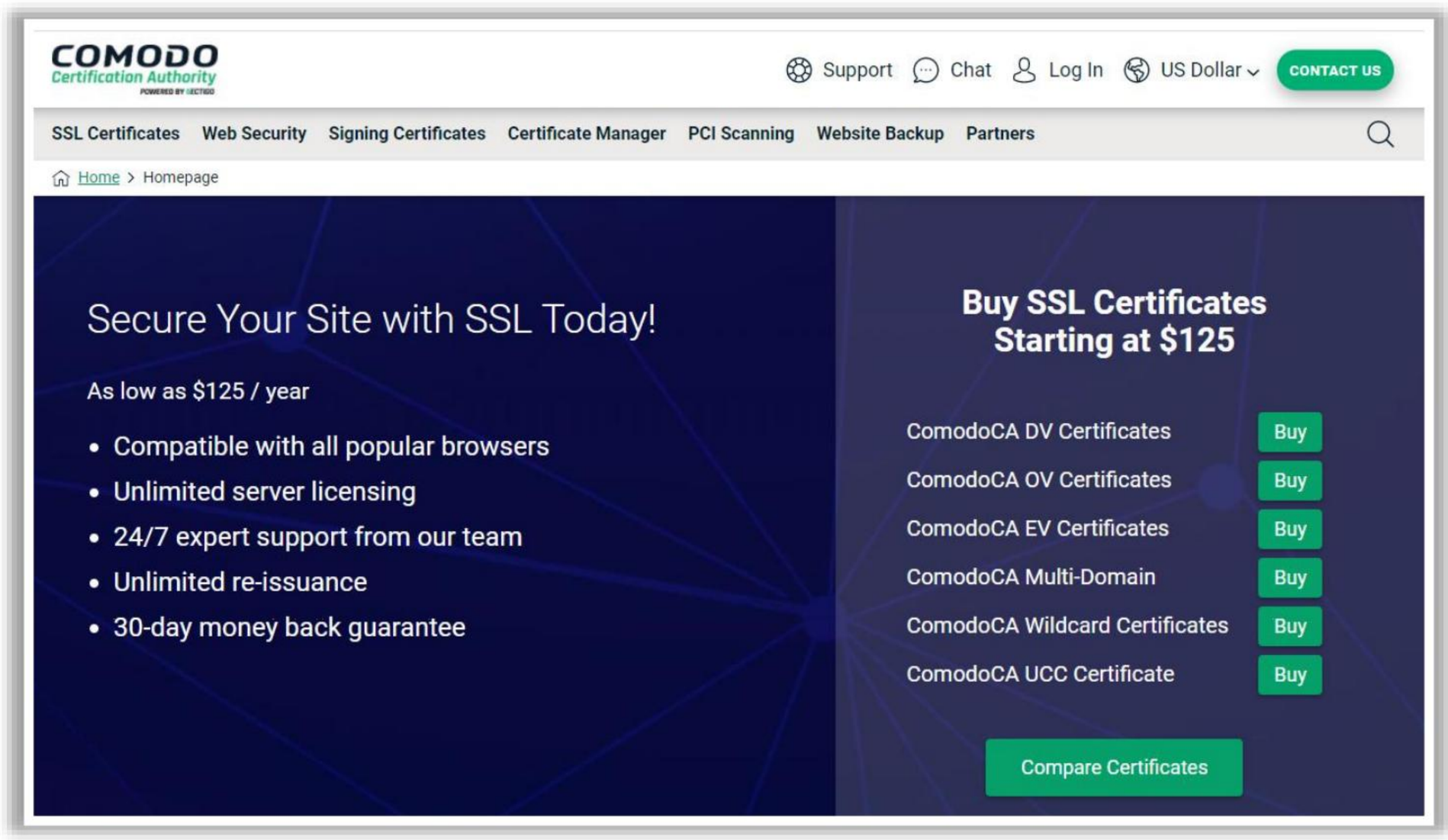


Figure 14.30: Screenshot of Comodo Website

- **IdenTrust**

  Source: *https://www.identrust.com*

  IdenTrust is a trusted third party that provides CA services for many sectors such as banks, corporates, governments, and healthcare. It provides solutions such as digital signing and sealing, compliance with NIST SP 800-171, global identity networks, and managed PKI hosting services.

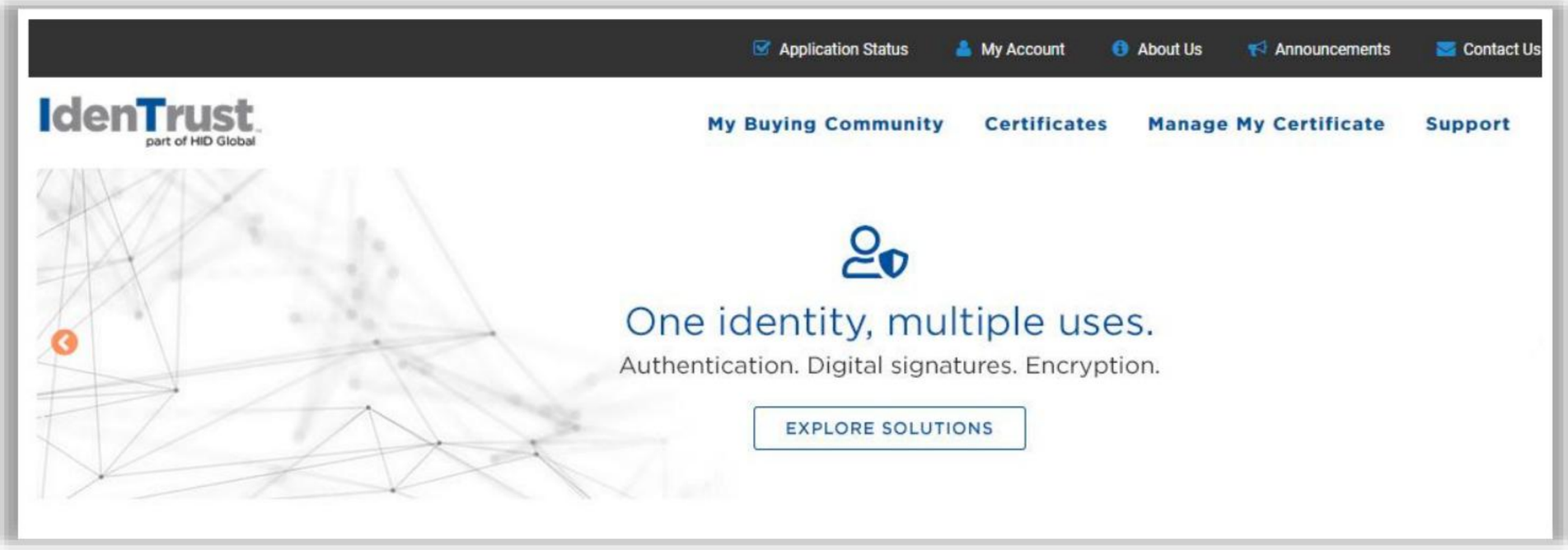


Figure 14.31: Screenshot of IdenTrust Website

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **DigiCert CertCentral**

  Source: *https://www.digicert.com*

  CertCentral simplifies the entire lifecycle by consolidating tasks for issuing, installing, inspecting, remediating, and renewing TLS/SSL certificates. It manages high-volume TLS/SSL certificate issuance for multiple individuals and teams.

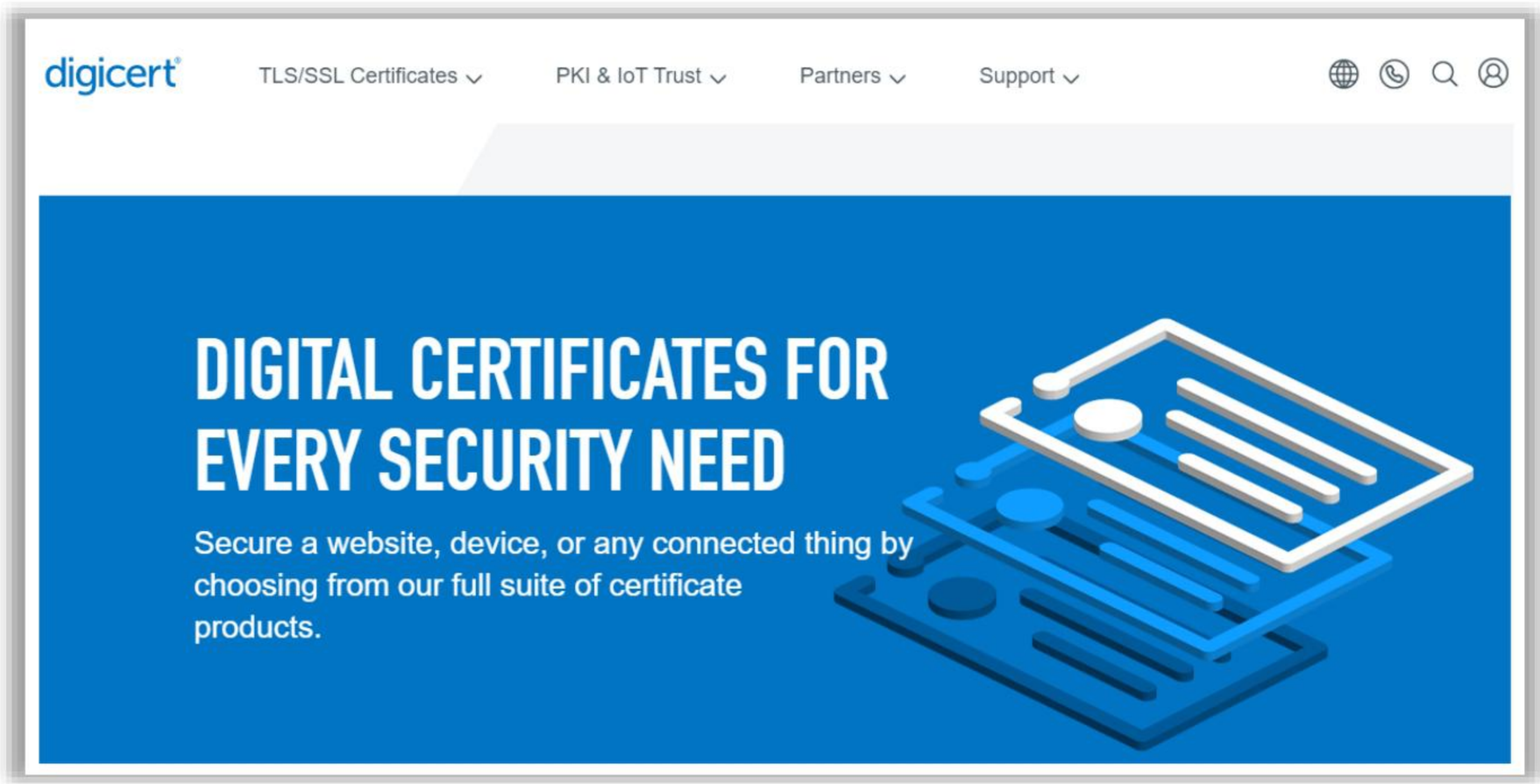


Figure 14.32: Screenshot of DigiCert Website

- **GoDaddy**

  Source: *https://www.godaddy.com*

  GoDaddy SSL Certificates offer a complete range of certificates that comply with CA/Browser Forum guidelines. They provide the SHA-2 hash algorithm and 2048-bit encryption, protection of unlimited servers, etc.



Figure 14.33: Screenshot of GoDaddy Website

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

PKI Trust Models

A trust model is a set of rules or constraints that instruct a client application on how to verify the **authenticity of digital certificates**

Peer-to-Peer trust model

Clients depend on their **local CAs**, which serve as a starting point

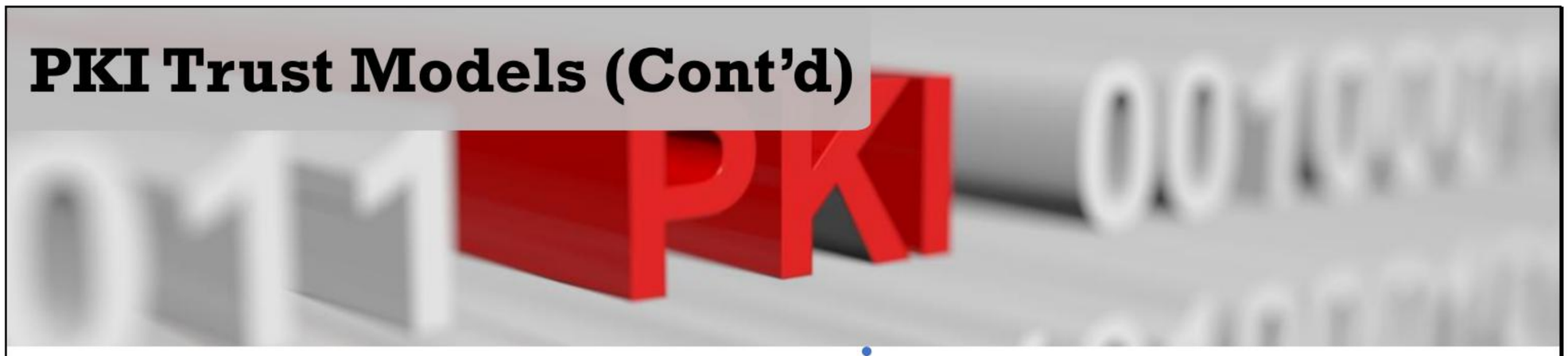
It is useful for small organizations





Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

PKI Trust Models (Cont'd)



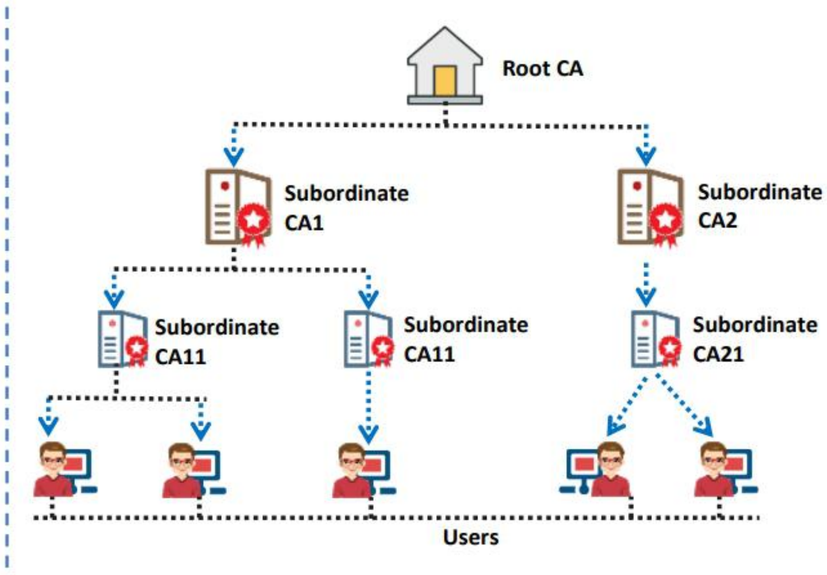
Hierarchical Trust Model

It is an inverted tree–like structure in which one master CA, called a root, is the **initial point of trust**

The root CA sends all the information to its descendants known as intermediate or **subordinate CAs**

All nodes in this model trust the **master CA** and hold the public key certificate of the master CA

Any problem in the root CA can **impact the overall trust** in the PKI infrastructure



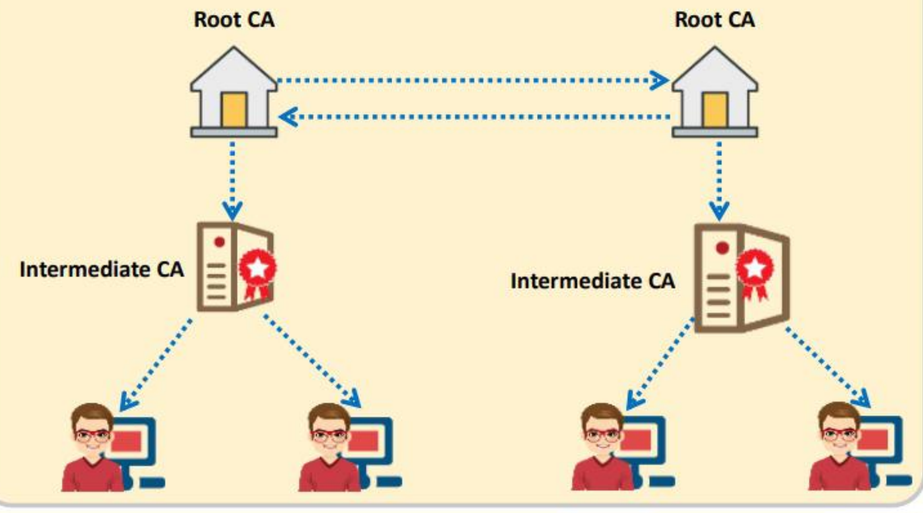
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

PKI Trust Models (Cont'd)

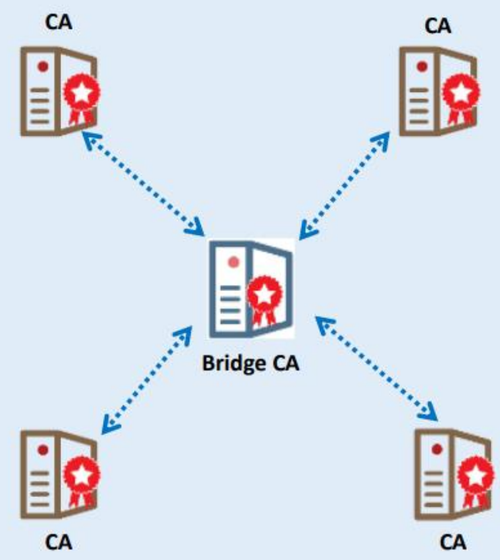
Hybrid Trust Model

- It is a combination of **peer-to-peer** and **hierarchical trust models**
- The root CAs perform peer-to-peer tasks by sharing public keys
- The intermediate CAs **trust only their respective root CAs** and other intermediate CAs within the hybrid environment



Bridge Trust Model

- It reduces the number of **cross-certificates** needed to be associated with PKIs
- In a bridge CA, one CA serves as a **central hub** for connecting all other CAs and acts as a communication medium for them

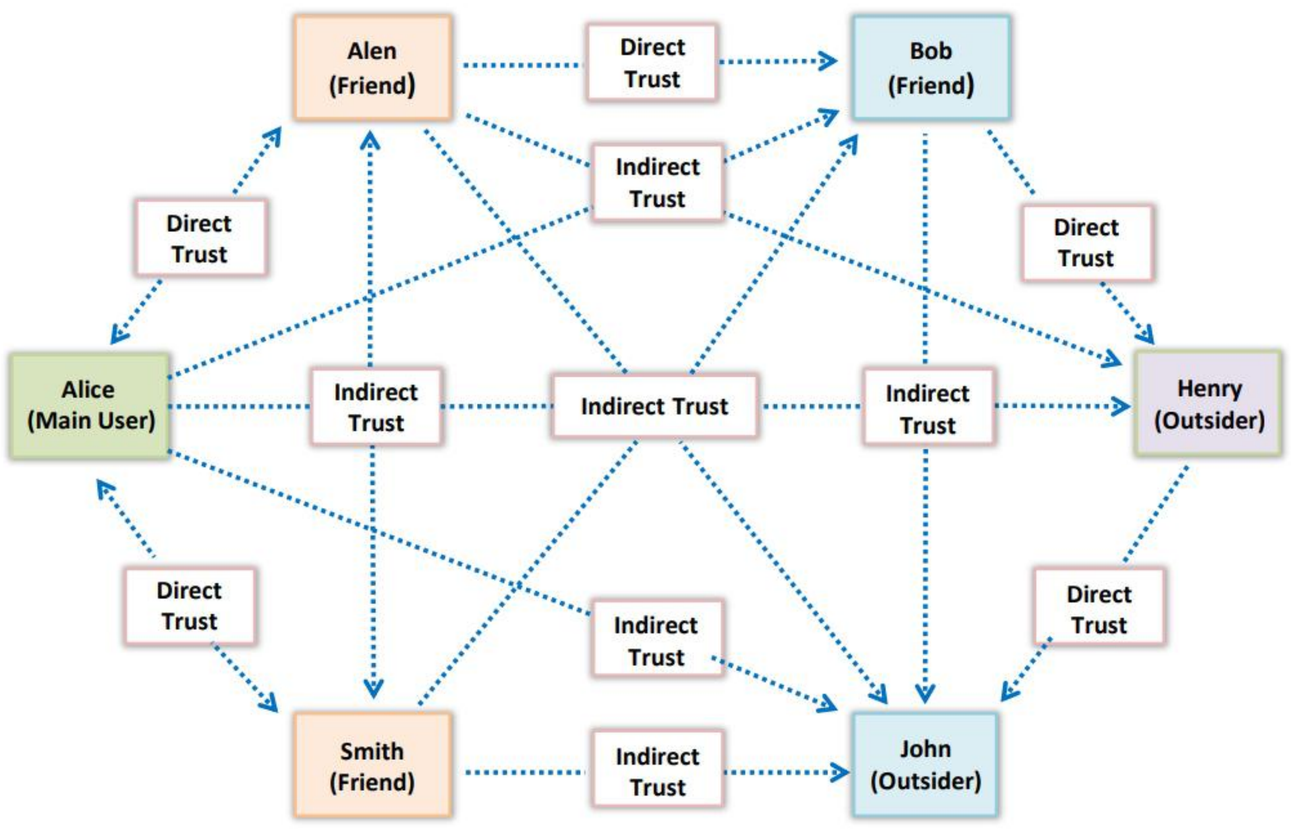


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

PKI Trust Models (Cont'd)

Web of Trust (WOT)

- Web of trust (WoT) is a trust model of **PGP**, **OpenPGP**, and **GnuPG** systems
- Everyone in the network is a certificate authority (CA) and signs for other trusted entities
- WoT is a **network chain** in which individuals act as intermediaries to validate each other's certificates using their signatures
- Every user in the network has a **ring of public keys** to encrypt data, and they introduce many other users whom they trust



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

PKI Trust Models

A trust model is a set of rules or constraints that instruct client applications on how to verify the authenticity of digital certificates. The trust is built from different security policies, services, operations, and protocols that provide interoperability through public-key encryption and certificate management and together provide a certain level of security. In public key infrastructure (PKI), the trust originates from a third party known as the certificate authority (CA). The PKI architecture explains its certificates and the trust relationships among them.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Discussed below are different types of trust models in PKI.

- **Peer-to-peer trust model**: In the peer-to-peer trust model, CAs do not have any intermediate CAs. There is no trust anchor between the CAs involved in the certificate process. In this model, clients usually depend on their local CAs, which serve as a starting point. The two CAs here are separate trust domains; only domain users can validate a domain user using their public key, which creates bi-directional trust. This type of trust model is useful for small organizations.



Figure 14.34: Peer-to-peer trust model

- **Hierarchical trust model**: This trust model is an inverted tree-like structure in which one master CA called a root is the initial point of trust. The root CA sends all the information to its descendants known as intermediate CAs or subordinate CAs, which only trust the information sent by the master CA. The master CA also trusts the intermediate CAs in the hierarchical structure. The leaf nodes are the users. All nodes in this model trust the master CA and hold the public key certificate of the master CA. The interaction between two users to validate themselves using the public key certificate should be performed through the root or master CA. The entire trust can be achieved from the root CA; any problem in the root CA can impact the overall trust in the PKI infrastructure.

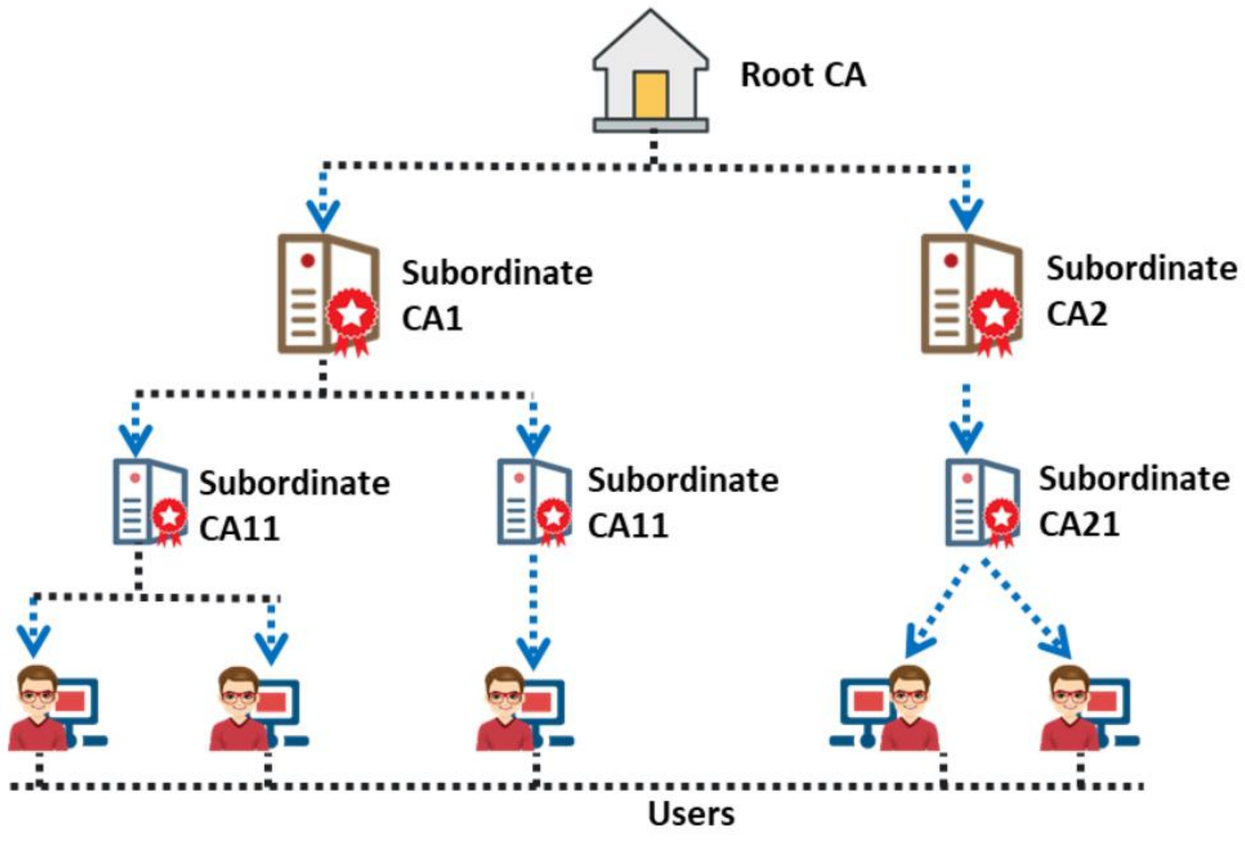


Figure 14.35: Hierarchical trust model

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Hybrid trust model**: This trust model is a combination of peer-to-peer and hierarchical trust models, in which root CAs perform peer-to-peer tasks by sharing public keys. The intermediate CAs trust only the respective root CAs and other intermediate CAs within the hybrid environment. Each root CA forms a separate trust domain with its child classes (nodes). This model is useful for cross-certification, where one domain user can verify another domain user. This is a widely used and easy operable model, in which two organizations or departments need to be combined.

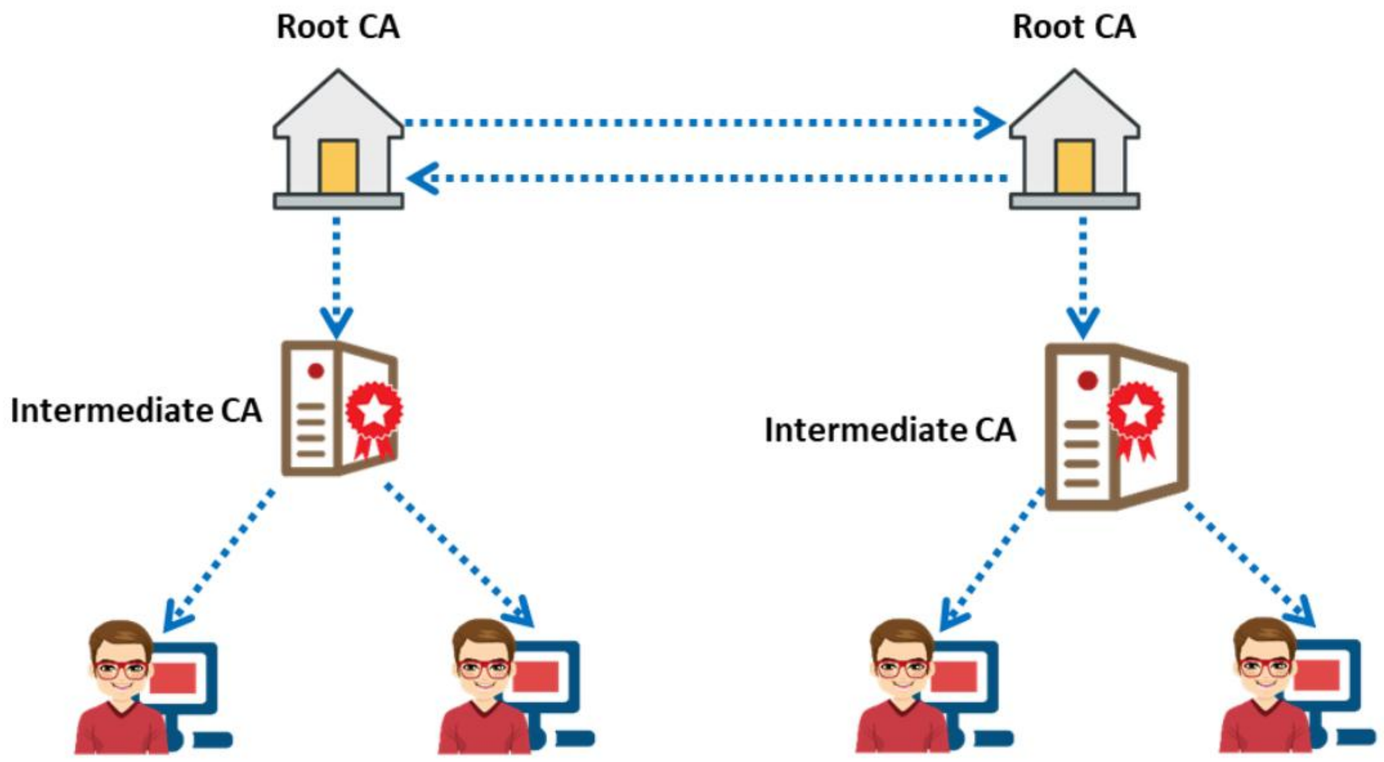


Figure 14.36: Hybrid trust model

- **Bridge trust model**: This model supports PKI apps across organizations and avoids conditions where clients need to hold the information of numerous trust points or enterprises wish to build crosslinks to various other enterprises (or departments). Bridge CAs (BCAs) reduce the number of cross-certificates needed to be associated with PKIs. In a BCA, one CA serves as a central hub for connecting all other CAs and acts as a communication medium for them. Every intermediate CA trusts only the CAs that are connected above and below it, but an additional CA can be constructed without generating extra layers of CAs. The major advantage of this model includes its flexibility and support for smooth interoperability between enterprises or departments.

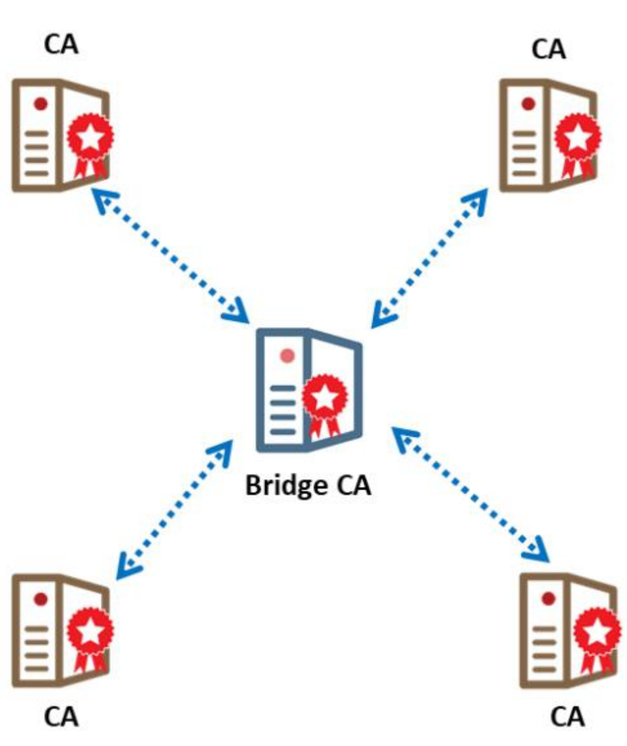


Figure 14.37: Bridge trust model

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Web of trust (WoT)**: WoT is a trust model of PGP-, OpenPGP-, and GnuPG-accessible systems. It is based on the idea of decentralizing key distribution among PGP users. In PKI, only a centralized power such as the CA signs certificates in the network, ensuring authenticity between the public key and its owner. In WoT, everyone in the network is a CA and can sign for other trusted entities. WoT is a network chain in which individuals act as intermediaries to validate each other's certificates using their signatures. These signatures verify the ownership of keys from various trust levels. There are various similar trust levels through direct or indirect references in WoT.

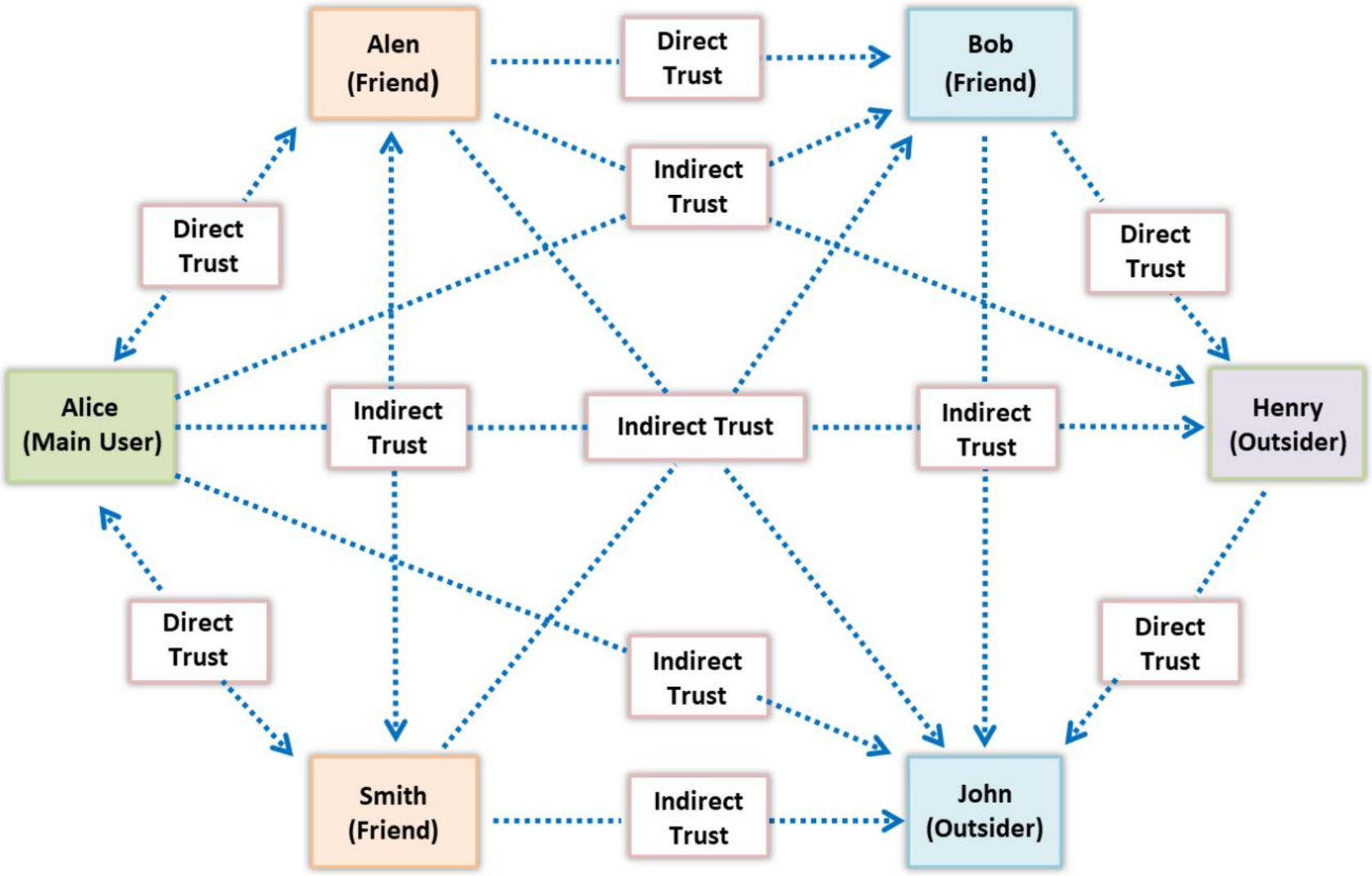
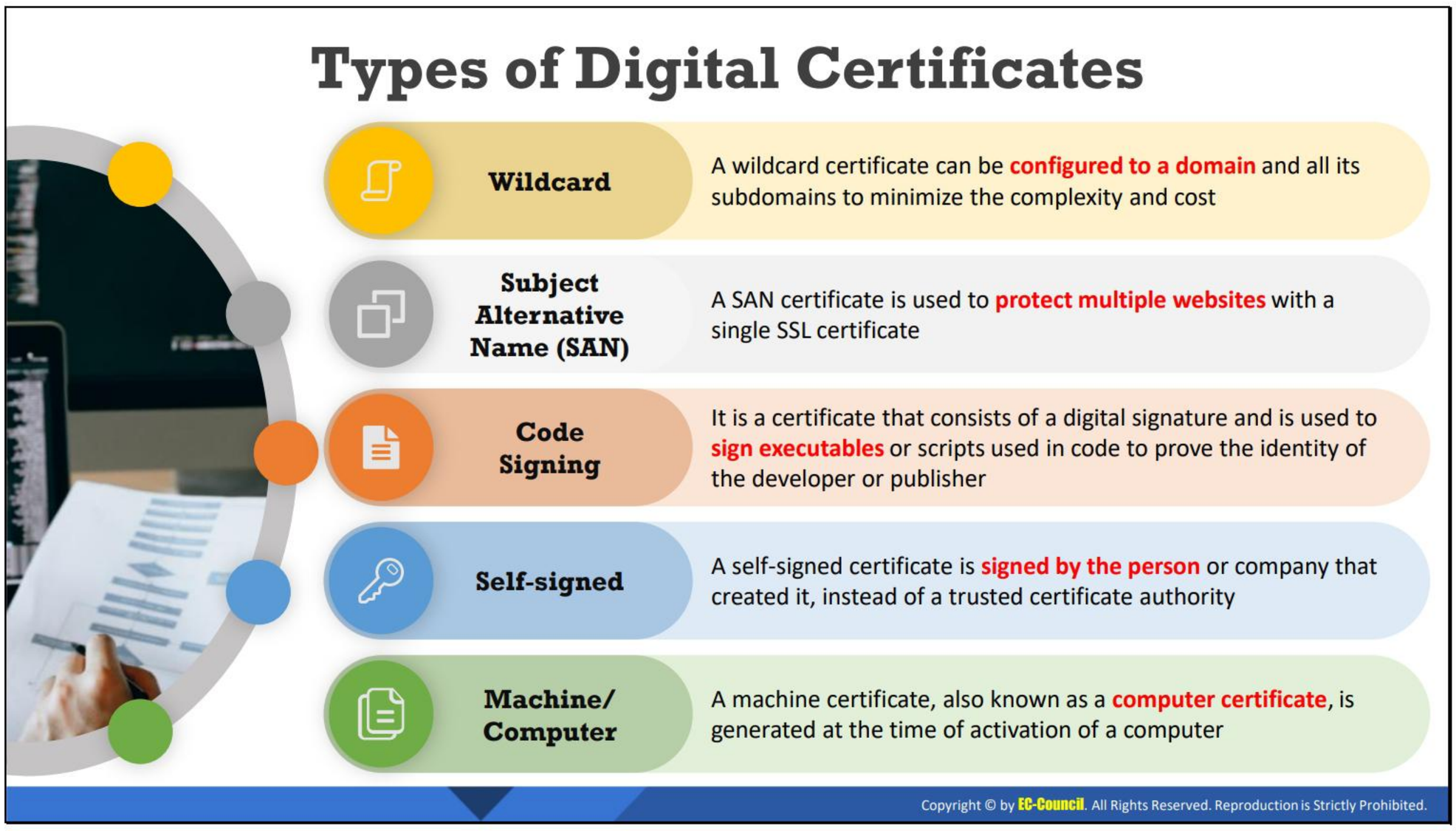
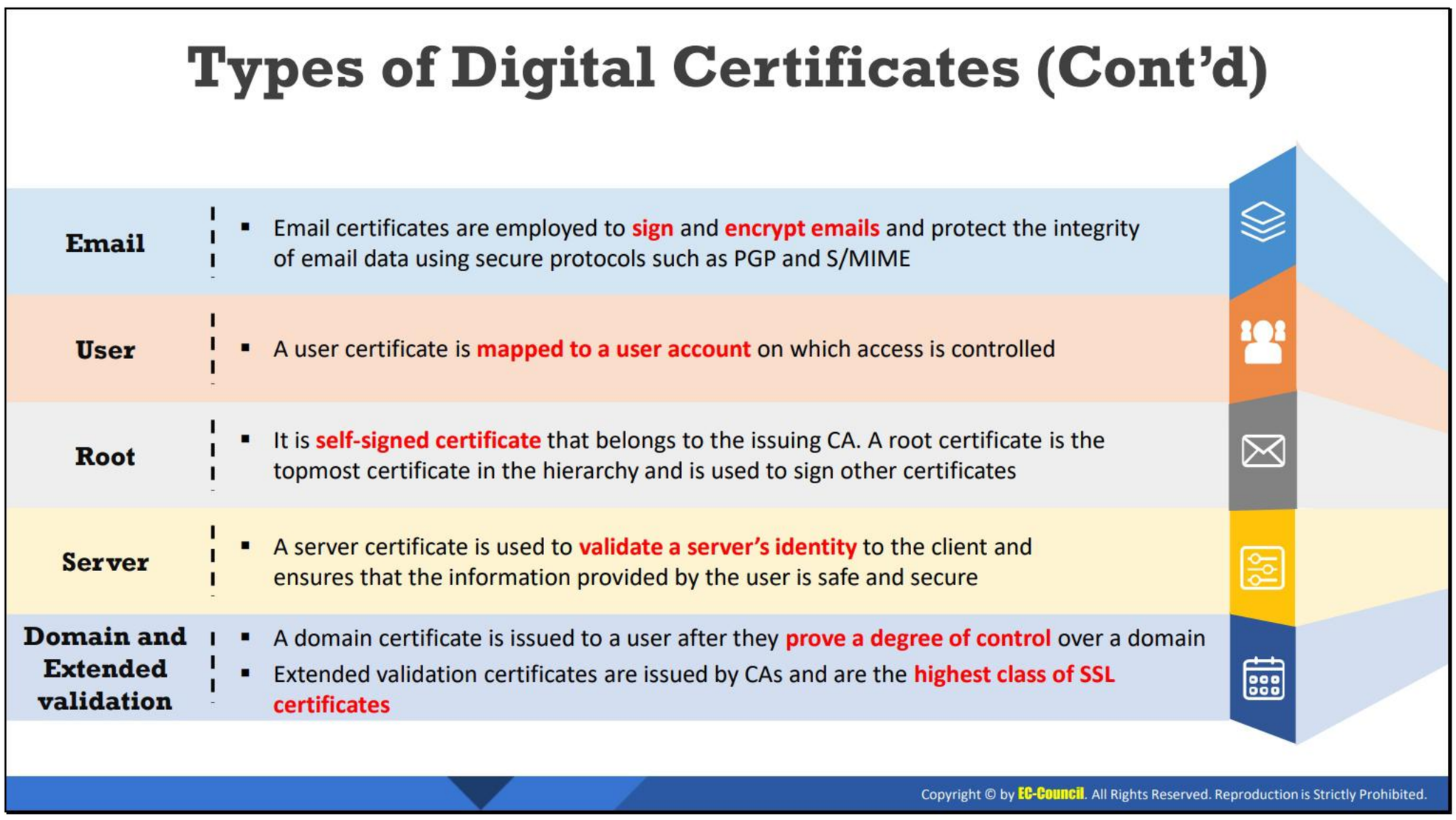


Figure 14.38: Working of WOT

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.





Types of Digital Certificates

A digital certificate allows enterprises to exchange data securely over the Internet with the help of public key infrastructure (PKI). It is used to verify the identity of the sender/receiver of a message and provides a method to encrypt or decrypt messages between the sender and receiver.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

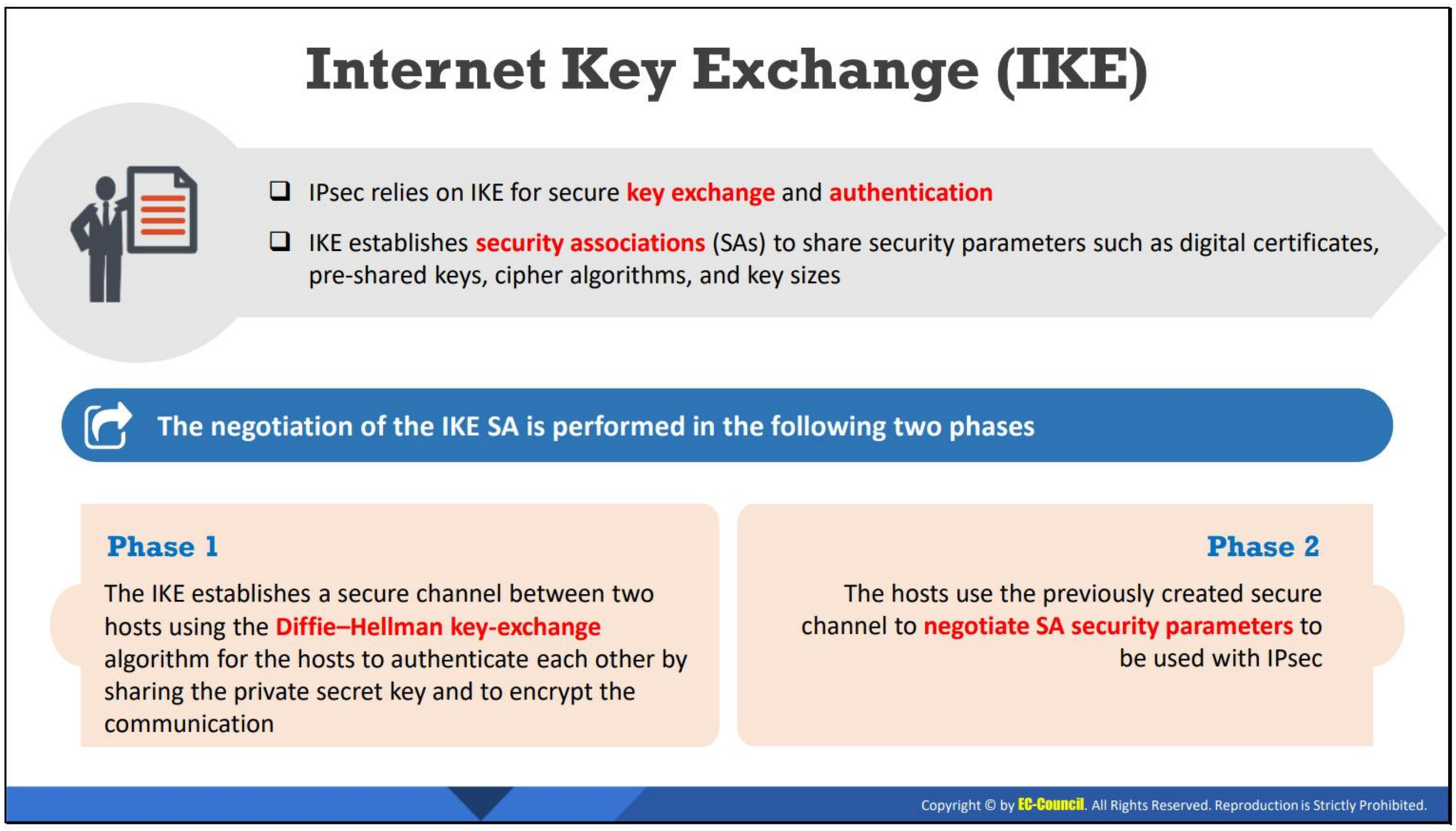
Discussed below are various types of digital certificates.

- **Wildcard**: A wildcard certificate can be configured to a domain and all its subdomains to minimize complexity and cost. It is mostly used in SSL certificates to extend SSL certification to subdomains. It is represented by an asterisk and a period before the domain name. It enables users to secure a large number of subdomains on a single certificate. A one-time purchase of a certificate can cover any number of subdomains that may be included in the future.
- **Subject alternative name (SAN)**: A SAN certificate is used to protect multiple websites with a single SSL certificate. It can also cover domains with distinct top-level domains and various levels of subdomains. It is also referred to as a unified communication certificate, exchange certificate, or multi-domain certificate. Using a SAN certificate, users can secure a primary domain and later add more domains to the subject alternative field of the certificate.
- **Code signing**: It is a certificate that consists of a digital signature and is employed to sign executables or scripts used in code to prove the identity of the developer or publisher and to check if the code has been altered since signing. These certificates ensure that users do not receive a warning during installation. Code signing is used to ensure that the code is trustworthy.
- **Self-signed**: A self-signed certificate is signed by the person or company that created it, instead of a trusted certificate authority. It is easy to create but cannot offer all the features of a CA-signed certificate. Self-signed certificates are suitable for internal sites and are used for encrypting incoming and outgoing data using identical ciphers, as in any other paid SSL certificate.
- **Machine/computer**: A machine certificate, also known as a computer certificate, is generated at the time of activation of a computer. These certificates can be issued to computing devices such as hosts and mobile devices, irrespective of their operation, and to network components such as firewalls and routing devices. A single certificate is installed for every user who has activated the computer into the hierarchy. It is mainly used for client- and server-side authentication.
- **Email**: Email certificates are employed to sign and encrypt emails and protect the integrity of email data using secure protocols such as PGP and S/MIME. If an email is encrypted using an email certificate, it can be decrypted only by the concerned receiver of the email. Email certificates are installed in the email application to ensure reliable communication. These certificates protect email data against common attacks such as email eavesdropping, phishing, and snooping.
- **User:** A user certificate is mapped to a user account on which access is controlled. User certificates are stored in smart cards. When a user wishes to access a system, the smart card is queried. The user unlocks the smart card to present the user certificate on it for authentication. The user certificates employ PKI for validating the user or device. User certificates are used only for authentication and not for encryption. These certificates allow access to only authorized individuals.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Root:** It is self-signed certificate that belongs to the issuing CA. A root certificate is the topmost certificate in the hierarchy and is used to sign other certificates. Certificates approved by a root CA are trusted by most web browsers. For such certificates to function as intended, they need to be installed on web servers where the domain is hosted.
- **Server certificate:** A server certificate is used to validate a server's identity to the client and ensures that the information provided by the user is safe and secure. It performs data encryption during transit to ensure confidentiality. The major problem with asymmetric cryptography is that anybody can implement PKI. When a client blindly trusts the certificates for a visited website, confidential data can be exposed. These certificates can be used to offer different levels of security based on the grade and reputation of the website.
  - **Domain validation:** A domain certificate is issued to a user after they prove some degree of control over a domain. It is necessary for the user to prove website ownership to the CA for its validation. It consists of 256-bit encryption and is compatible with most web browsers. It can be purchased by small or medium-level website owners who wish to encrypt their domain.
  - **Extended validation (EV):** EV certificates are issued by CAs for securing online banking transactions, ecommerce, etc. It is the highest class of SSL certificate. A website protected by EV SSL will show a padlock and an HTTPS prefix to users to assure them that they are on an encrypted website and that their sensitive data are secured. The website owner must go through an identity verification process to receive this certificate.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



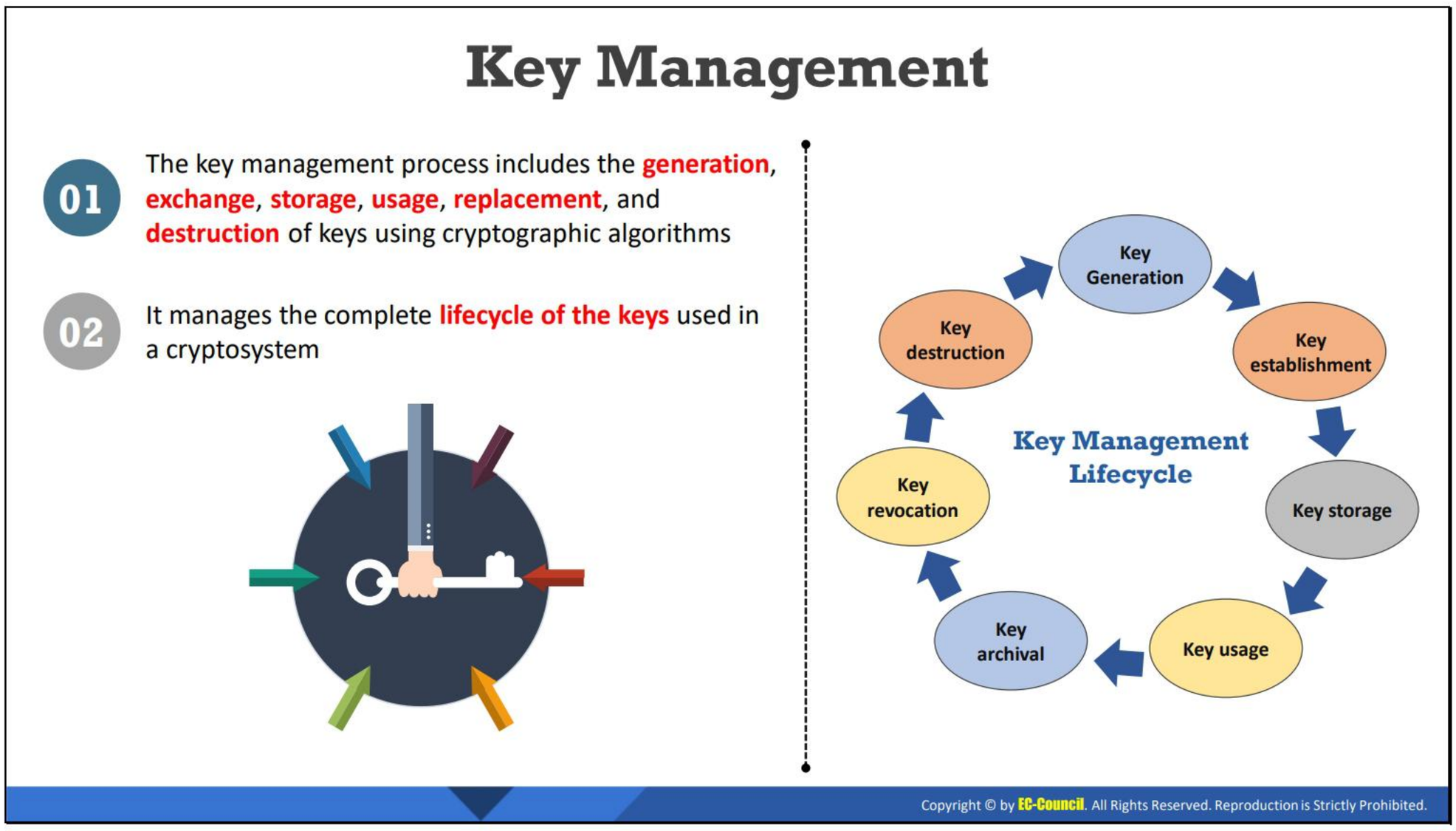
Internet Key Exchange (IKE)

IPsec relies on IKE for secure key exchange and authentication. IKE establishes security associations (SAs) to share security parameters such as digital certificates, pre-shared keys, cipher algorithms, and key sizes. IKE provides protection against man-in-the-middle (MITM) and replay attacks.

IKE is used in Virtual Private Network (VPN) negotiation, remote network access, etc. to establish a secure tunnel between two hosts. The main benefit of using IKE is that it eliminates the manual specification of security parameters for both hosts. The negotiation of the IKE security association (SA) is performed in the following two phases.

- **Phase 1:** In the first phase, the IKE establishes a secure channel between two hosts. The hosts use the Diffie–Hellman key-exchange algorithm to authenticate each other by sharing the private secret key and to encrypt the communication. Pre-shared keys and RSA signatures are also supported for authentication. The result of this phase is a bidirectional Internet Security Association and Key Management Protocol (ISAKMP) SA channel.
- **Phase 2:** In the second phase of IKE, the hosts use the previously created secure channel to negotiate SA security parameters to be used with IPsec. The result of this phase is two bidirectional secure channels.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Key Management

Key management is the process of managing cryptographic keys in a cryptosystem. The key management process includes the generation, exchange, storage, usage, archival, revocation, and destruction of keys using cryptographic algorithms. The key management process also involves key servers, cryptographic protocols, user procedures, and other relevant protocols. It manages the complete lifecycle of the keys used in the cryptosystem.

The following are the main phases involved in the key management process include.

- **Key generation:** Strong and secure key pairs are generated using standard algorithms.
- **Key establishment or registration**: The generated key should be specific to the user, system, or process with a name and attributes. The key is submitted to the CA to identify the public part of the key pair. The CA verifies the client and registers the key. Once the key is registered, it can be used for its intended purpose.
- **Key storage**: The registered key will be used over the long term. Therefore, it needs to be stored in a protected storage space. It is imperative for client to ensure that the private key is stored securely and no unauthorized access is allowed
- **Key usage**: The key should be used for its intended purpose by the client. The registered key is used to sign and encrypt the data and other keys.
- **Key archival**: It is the backup or recovery mechanism used to obtain lost or stolen private key.
- **Key revocation:** If the private key is compromised, the key pair must be revoked as early as possible so that it cannot be used for new encryption and decryption. If the key has already been used for encryption, it must be retained for some time. The administrator

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

can restore the revoked key to decrypt the previously encrypted information. Key revocation is also performed when an organization closes its business activities, an employee leaves the organization, or the organization changes its domain name.

- **Key destruction** The key is destroyed on the expiry date (it can also be renewed). When it needs to be removed permanently, the decommissioned key should be deleted along with all its associated data and instances.

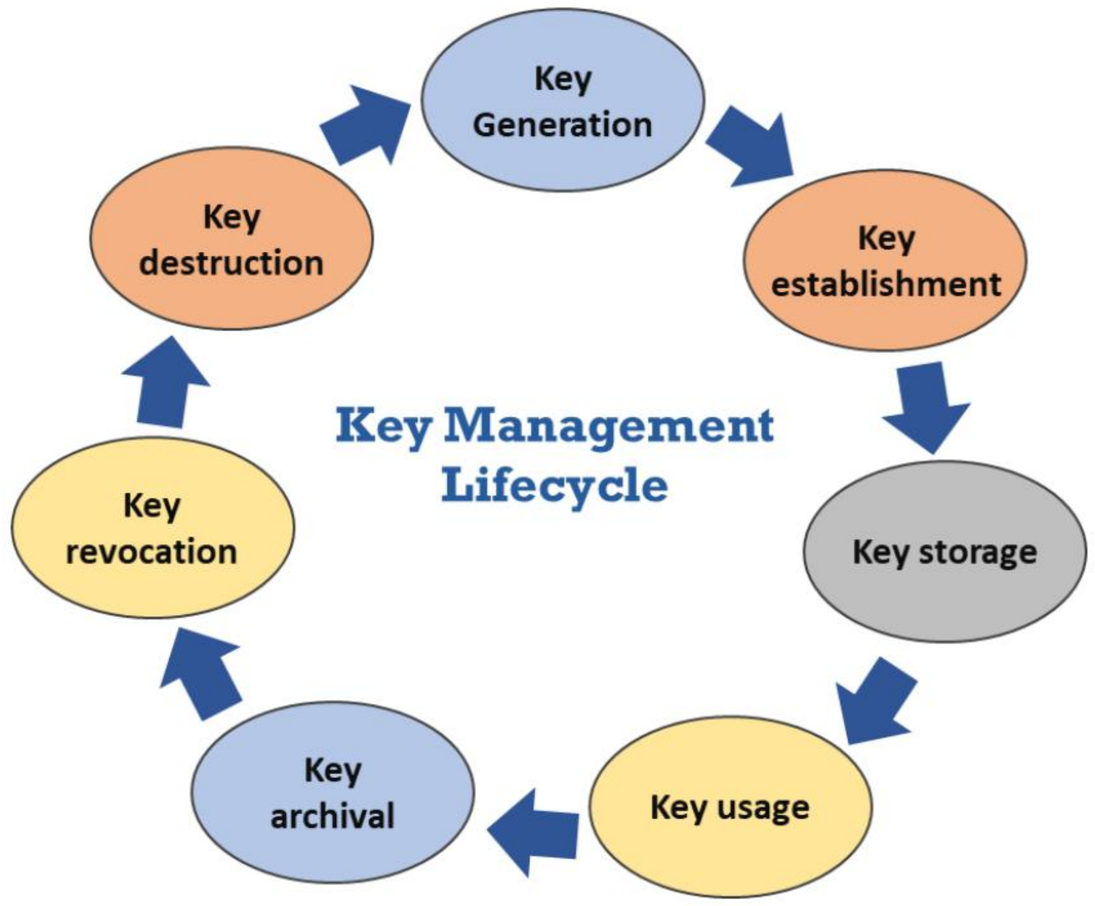


Figure 14.39: Key management lifecycle

The distribution of public keys is also a part of the key management process and can be achieved using the following methods.

- **Public announcement**: In this method, the public key is broadcasted to all users. This is not a secure method because it has high chances of key forging.

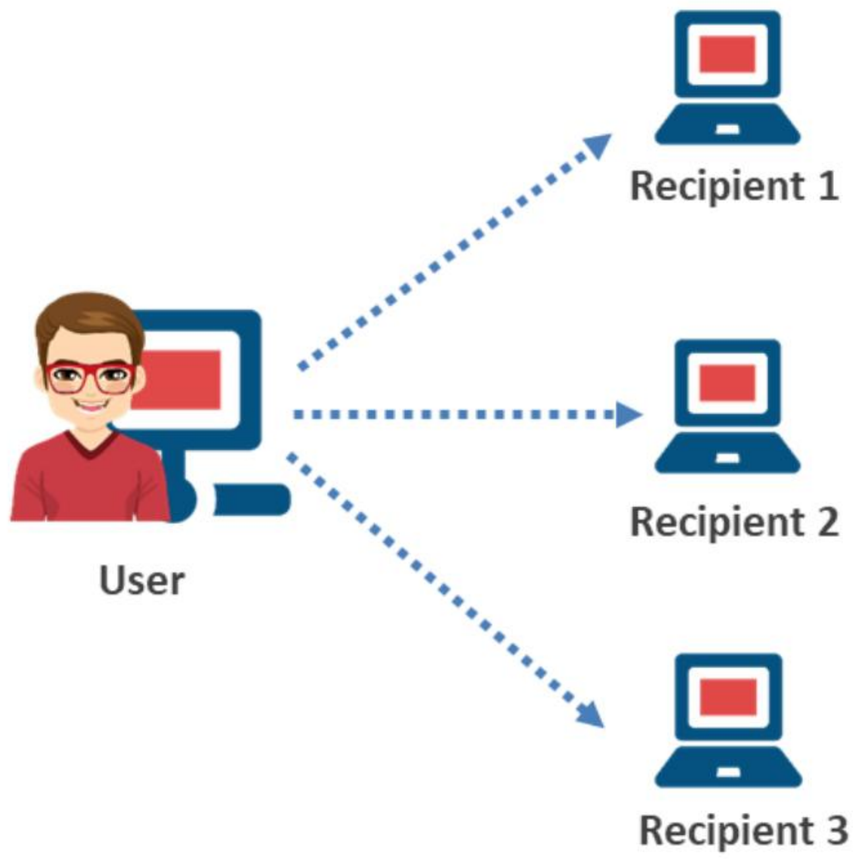


Figure 14.40: Key exchange through public announcement

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Publicly available directory**: In this method, the public key is stored in the public key directory. Every user should register a public key with the directory to establish their user ID and password. Public directories can also be subjected to forgery.

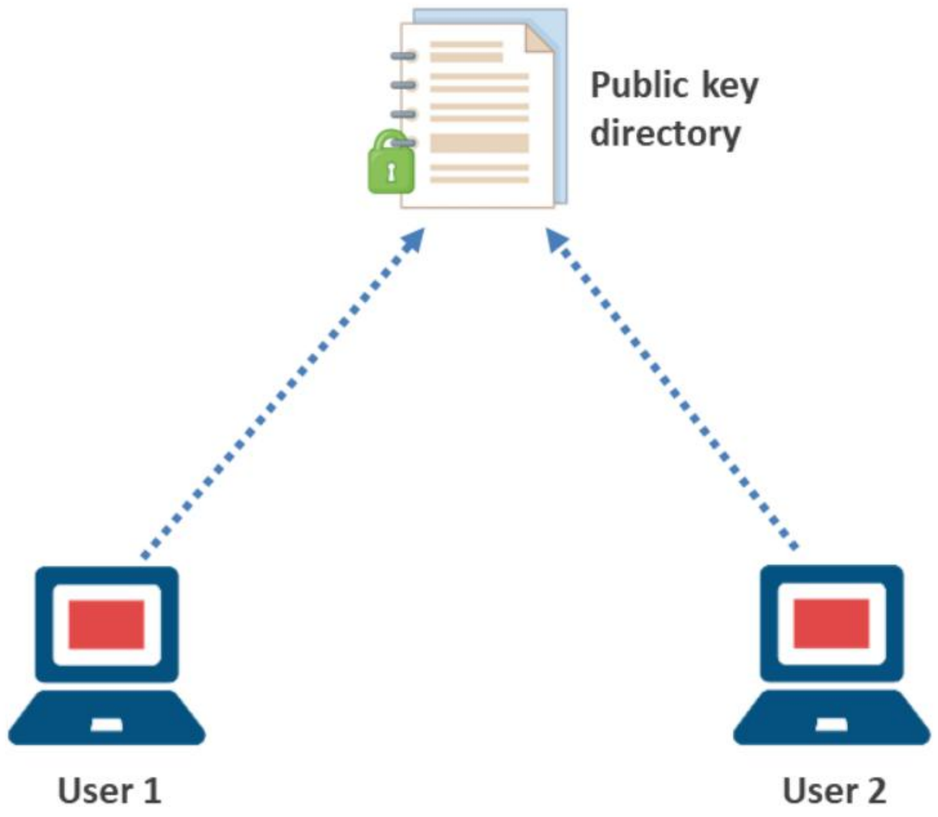


Figure 14.41: Key exchange through a publicly available directory

- **Public key authority**: This method is similar to the method of publicly available directories but enhances security by hardening control over the key distribution from the directories. In this method, the public key has a timestamp and is signed by the authority to protect keys from forgery.
- **Public key certificates**: In some cases, the public keys and directories maintained by the authority can be susceptible to forgery. To overcome this issue, the certification approach has been recommended. A certificate, incorporated with a public key, key owner identifier, and the entire block, is signed by a trusted entity such as a government agency or third party. The certificate also has a validity period and rights to use. The certificate can be verified by any user holding the CA's public key.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Key Escrow

Key escrow is the process of keeping secret keys with a certified third-party that serves as a backup for cryptographic keys. In this mechanism, the private key is split into two parts, each of which is encrypted. The two halves of the key are handed over to a third party that stores the key segments in a secure location. Users can obtain these two copies of keys and combine them to decrypt the data. Using this mechanism, organizations can ensure that their critical keys are safe in the case of a disaster or stolen/lost keys. These keys are often secured with additional encryption and can be accessed by users when required.

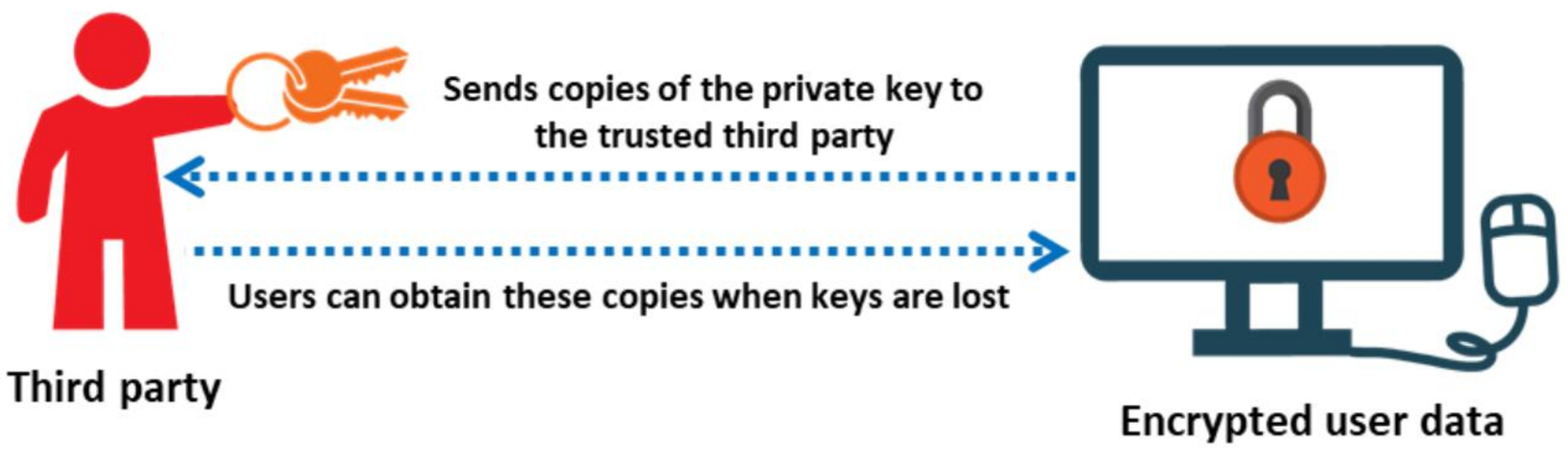


Figure 14.42: Illustration of key escrow

Certificate Chaining

The certificate chain, also referred to as the chain of trust, is established by a set of certificates starting from the server certificates and ending with the root certificates. If the server certificate is legitimate, the signature of the certificate should be discoverable by its root certificate authority (CA). The chain of trust is formed when each certificate is signed by a party that is trusted by the next certified party. By default, users can discover Root CAs, but intermediate and server certificates should be signed by a CA that is not discoverable by the

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

client's browser. In such cases, the root CA should validate and sign the intermediate CA, which then signs the server certificate. When a user attempts to connect with a server that holds a certificate signed by an intermediate CA, the server certificates can be tracked by the root CA through its intermediate CA, which is how the user gains trust. Certificate chaining makes key management and certificate tracking easy by forming a hierarchical structure of certificates in which validating the master or root CA automatically validates the rest of the chain.

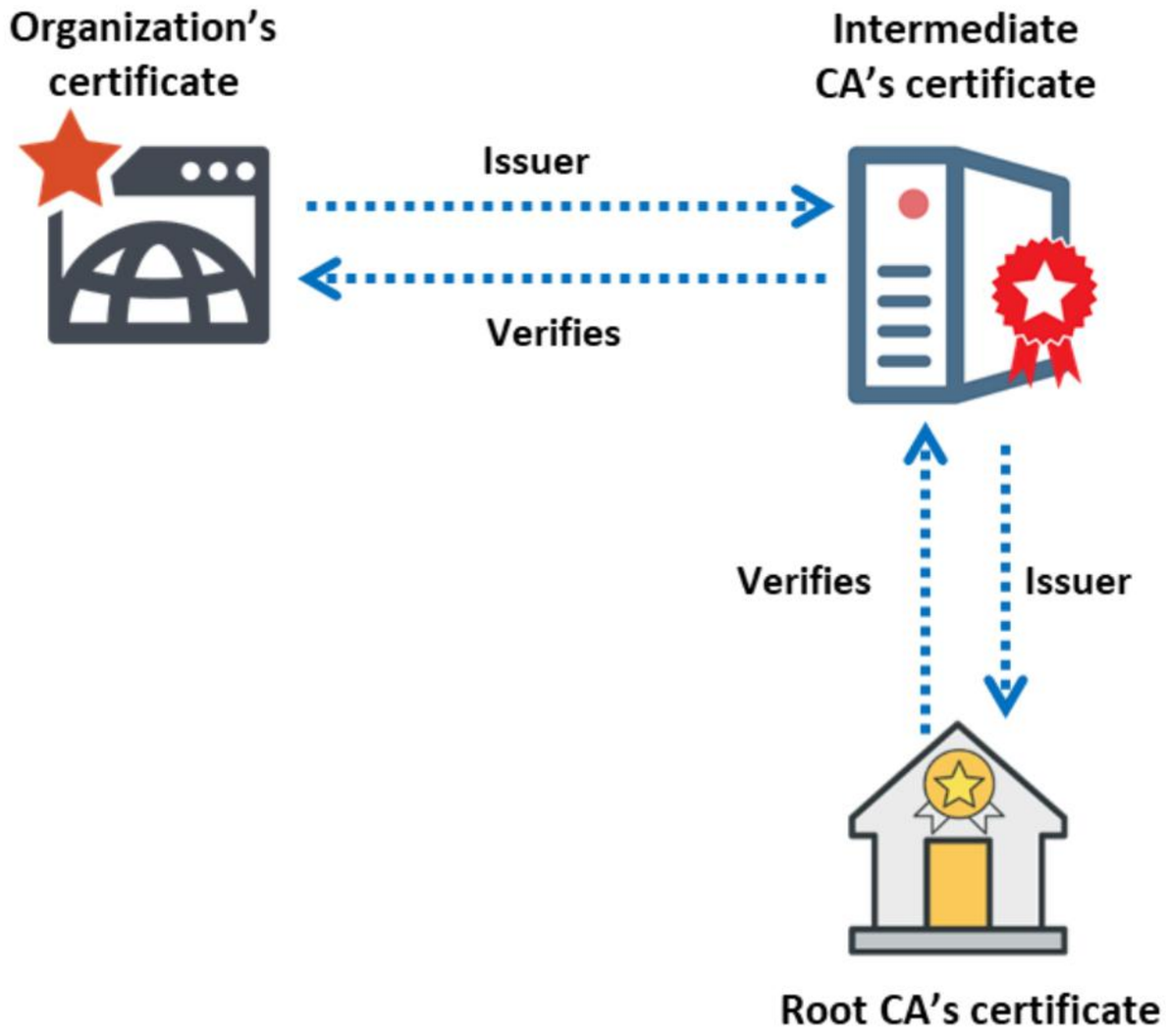
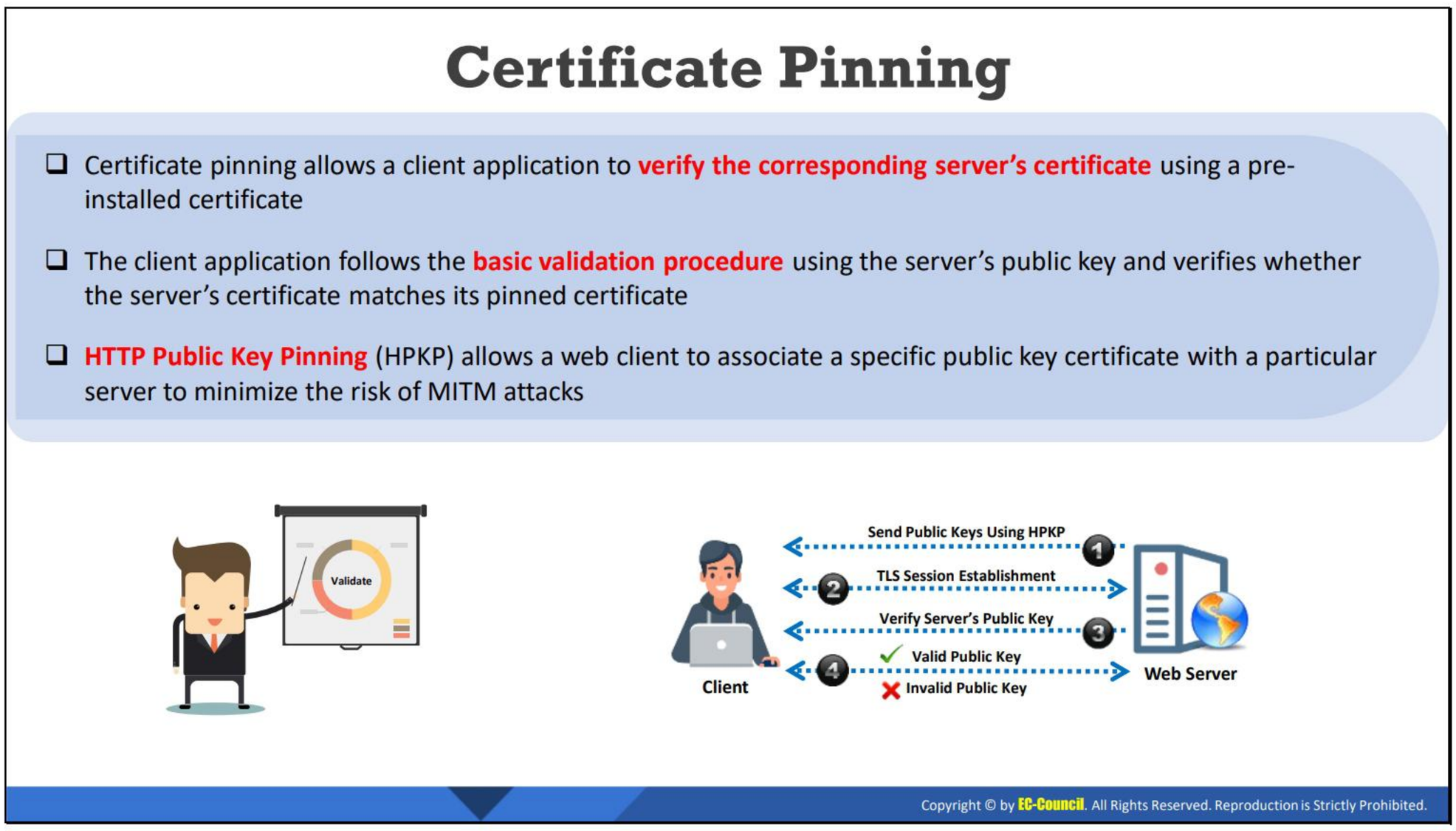


Figure 14.43: Certificate chaining

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Certificate Pinning

Certificate pinning allows a client application to verify the corresponding server's certificate using a pre-installed digital certificate. Upon pinning the server's certificate to the client's application, the client application follows the basic validation procedure using the server's given public key and also verifies whether the server's certificate matches its pinned certificate.

To determine whether the server's certificate matches the pinned certificate, the client must check the entire certificate or its public key. Instead of using a copy of the public key or certificate, the client can use fingerprints that serve the same purpose. Here, a fingerprint can be a hash form of the same public key or certificate. If fingerprints match between the server and client certificates, the connection is established; else, the client's application denies the connection. The certificate integrated inside an application can eventually expire. Upon expiration, the client must update the application to a new version that is embedded with an updated certificate.

How Certificate Pinning Works

- A copy of the certificate integrated in the client's application is used during the SSL/TLS handshake.
- The client SDK checks whether the server certificate's public key matches the application certificate's public key.
- If the pinning process is successful, the public key inside the certificate is used to check the integrity of the server certificate.
- If the pinning process is unsuccessful, all SSL/TLS queries to the server are refused by the client application.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

HTTP Public Key Pinning (HPKP) is an example of certificate pinning. It is a trust on first use (TOFU) technique used in an HTTP header that allows a web client to associate a specific public key certificate with a particular server to minimize the risk of MITM attacks based on fraudulent certificates.

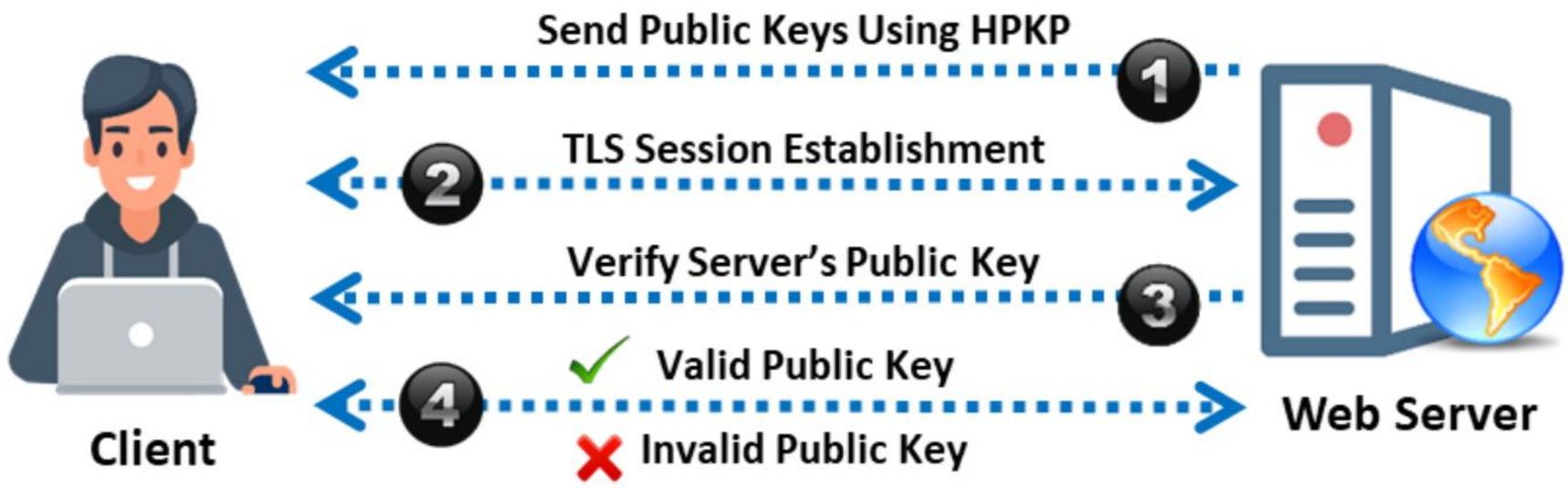
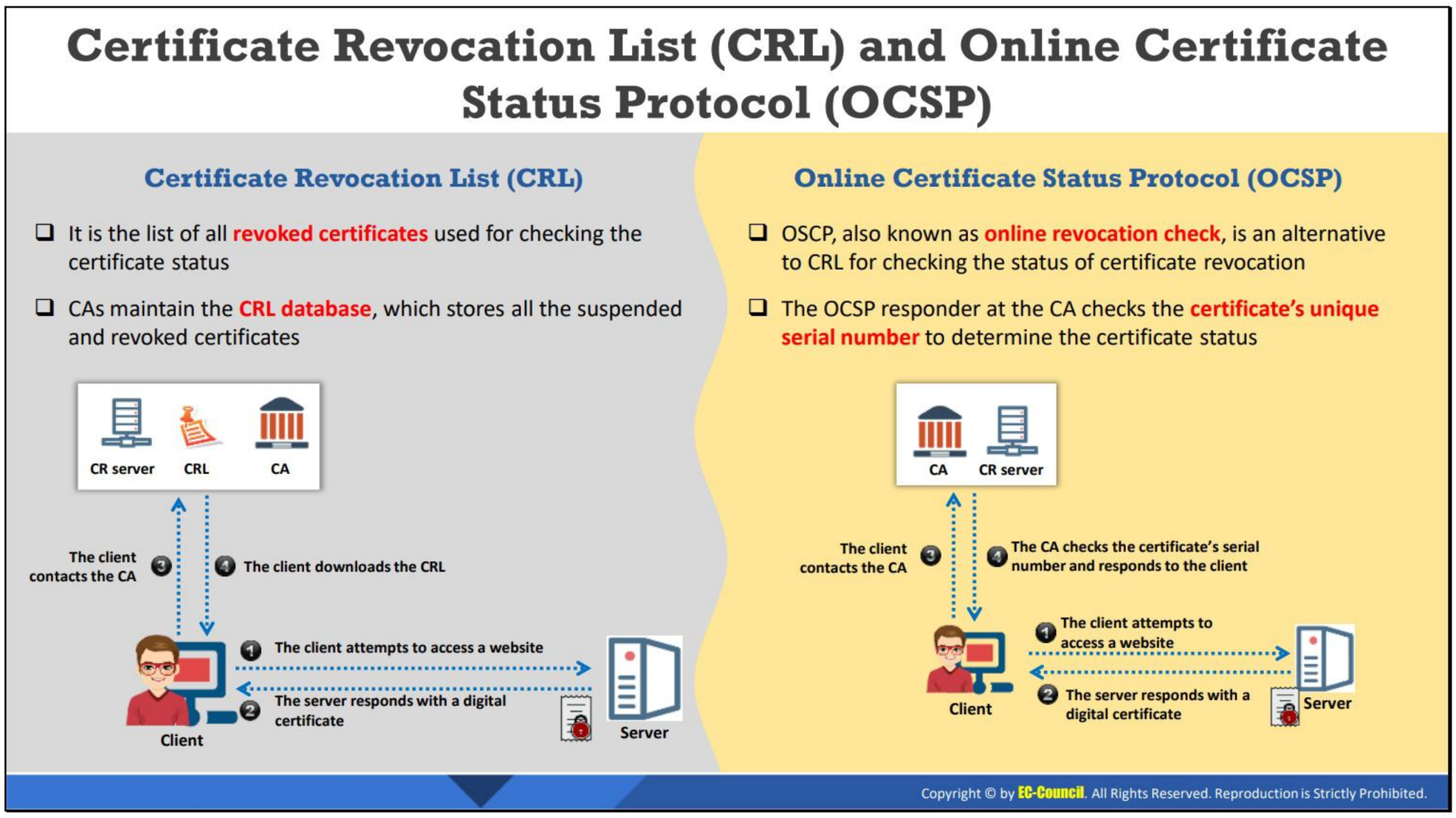


Figure 14.44: Certificate pinning using HPKP

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Certificate Revocation List (CRL)

CRL is the list of all revoked certificates and is used for checking certificate status in a process known as offline revocation check. CAs maintain the CRL database, in which all the suspended and revoked certificates are stored. The client should download the list to check whether the certificate is revoked. It is a time-consuming and complex process because the server must download and check the certificate status from the entire CRL list; Online Certificate Status Protocol (OCSP) was designed to overcome these issues.

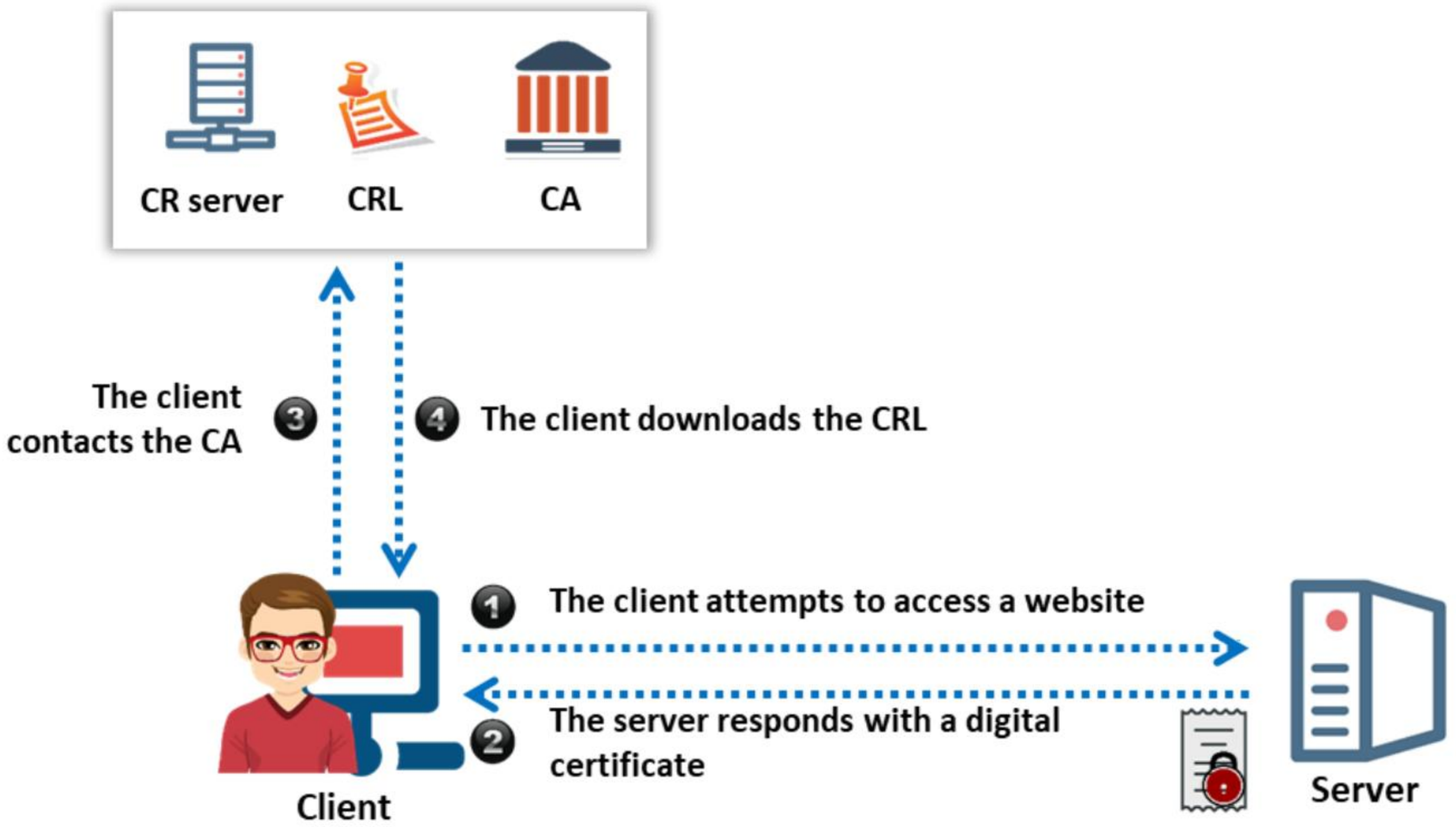


Figure 14.45: Verification of certificate revocation

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Online Certificate Status Protocol (OCSP)

OSCP, also known as online revocation check, is an alternative to CRL for checking the status of certificate revocation. OCSP requests are sent to the CA using HTTP. The OCSP responder at the CA checks the certificate's unique serial number and responds with its status. With this method, instead of going through the entire CRL, the client can check the certificate status using its unique serial number. The main disadvantage of this method is that the OSCP responder placed at the CA requires enormous processing resources to send responses to client requests. Moreover, the responder monitoring the client's requests may have an impact on the client's privacy. OSCP stapling has been introduced to solve these issues.

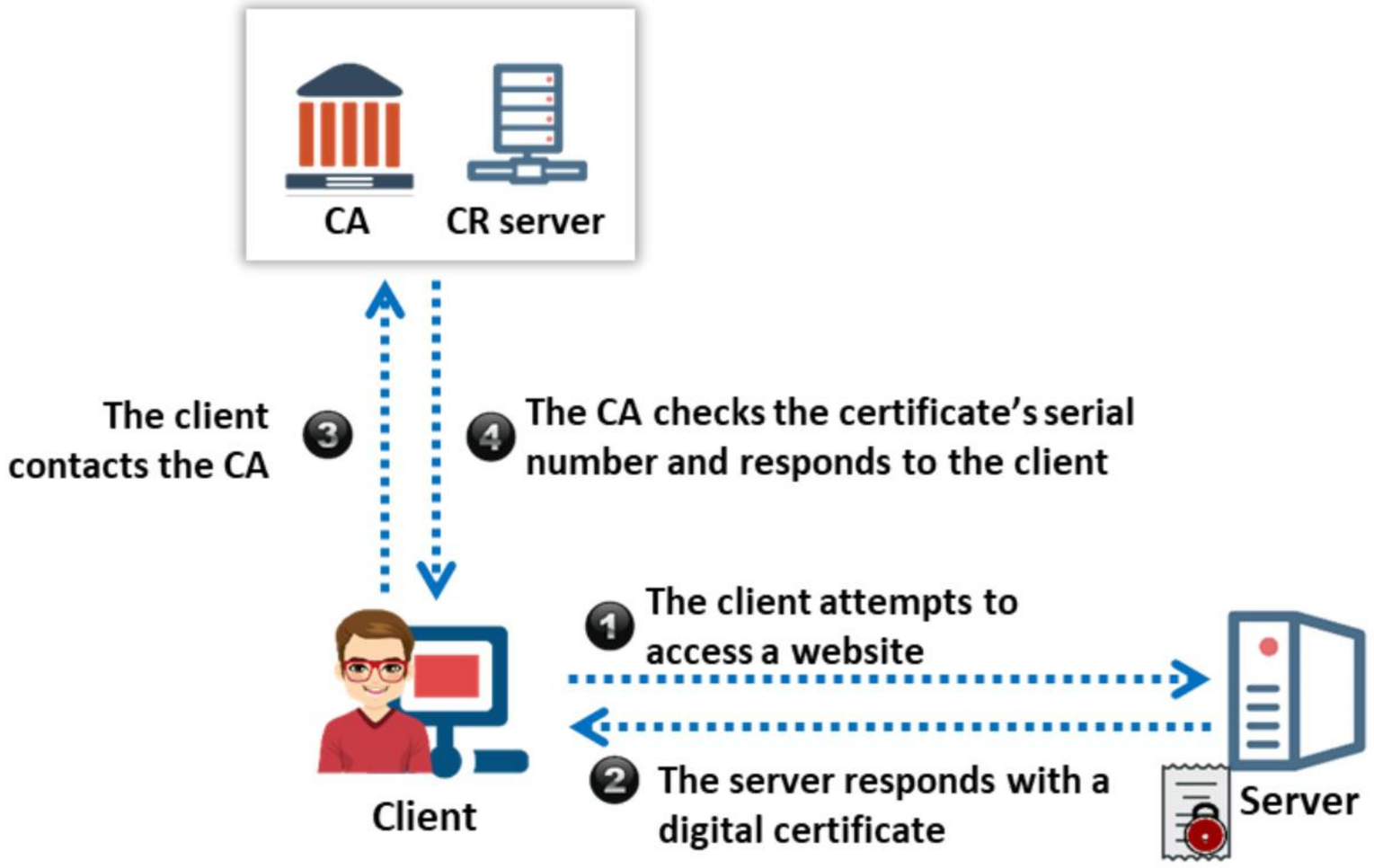
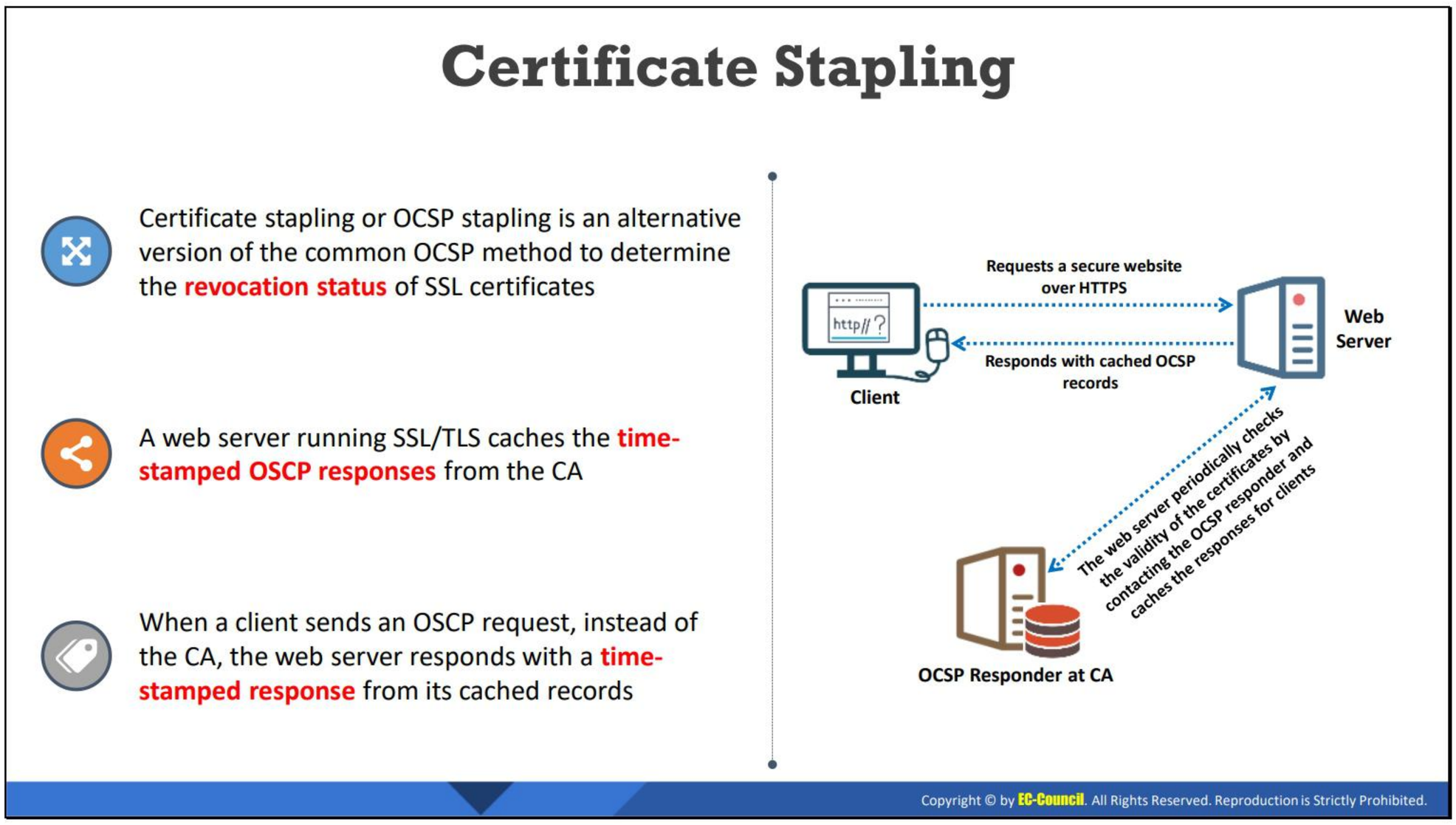


Figure 14.46: Working of OSCP

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Certificate Stapling

Certificate stapling, also known as OCSP stapling, is an alternative version of the common OCSP method to determine the revocation status of SSL certificates. It serves as a bridge between browsers and the CA. It addresses some of the issues associated with the common OCSP checking method, minimizing the communication time between the client and the server or CA. It allows browsers to check certificate validity without frequently contacting the CA through the web server.

How Certificate Stapling Works

- The client's web server sends an OCSP request to the issuer CA server.
- The issuing CA replies with the OCSP status and a timestamp.
- The web server caches the OCSP response and delivers it to the client, instead of making multiple requests on behalf of the client.
- The client validates the signature on the timestamp to check if it is signed by the CA.

In the common OCSP mechanism, the server sends a certificate to the client, and the client verifies it. The browser then sends an OCSP request to the CA, where OCSP responder is located. The OCSP responder sends one of three possible replies: good, revoked, or unknown. In OCSP stapling, the corresponding server performs OCSP operation, stores OCSP records for the client, and provides responses directly to the client when requested. The web server running SSL/TLS caches the time-stamped OSCP responses from the CA. When a client sends an OSCP request, instead of the CA, the web server responds with time-stamped response from its cached records.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

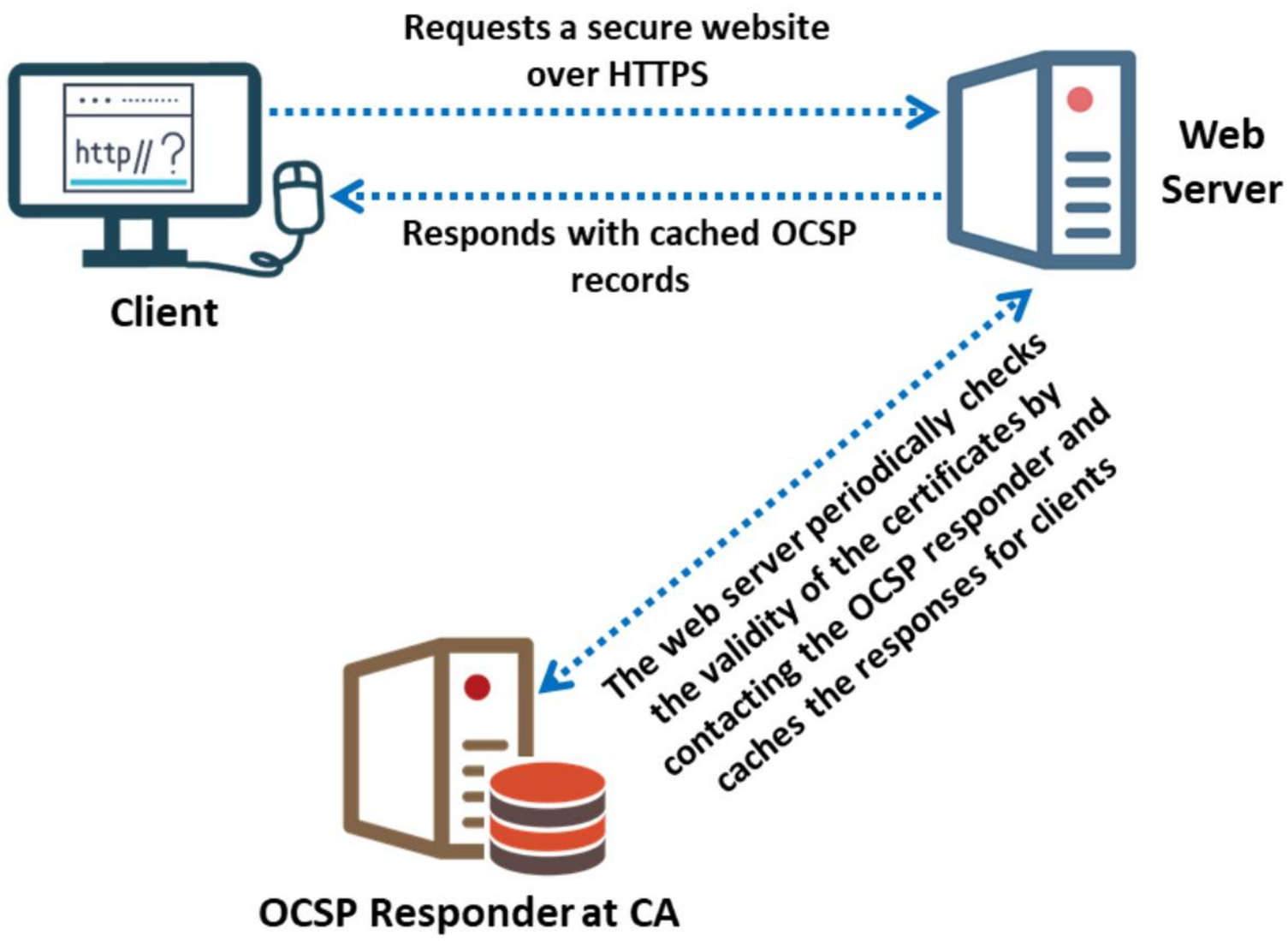
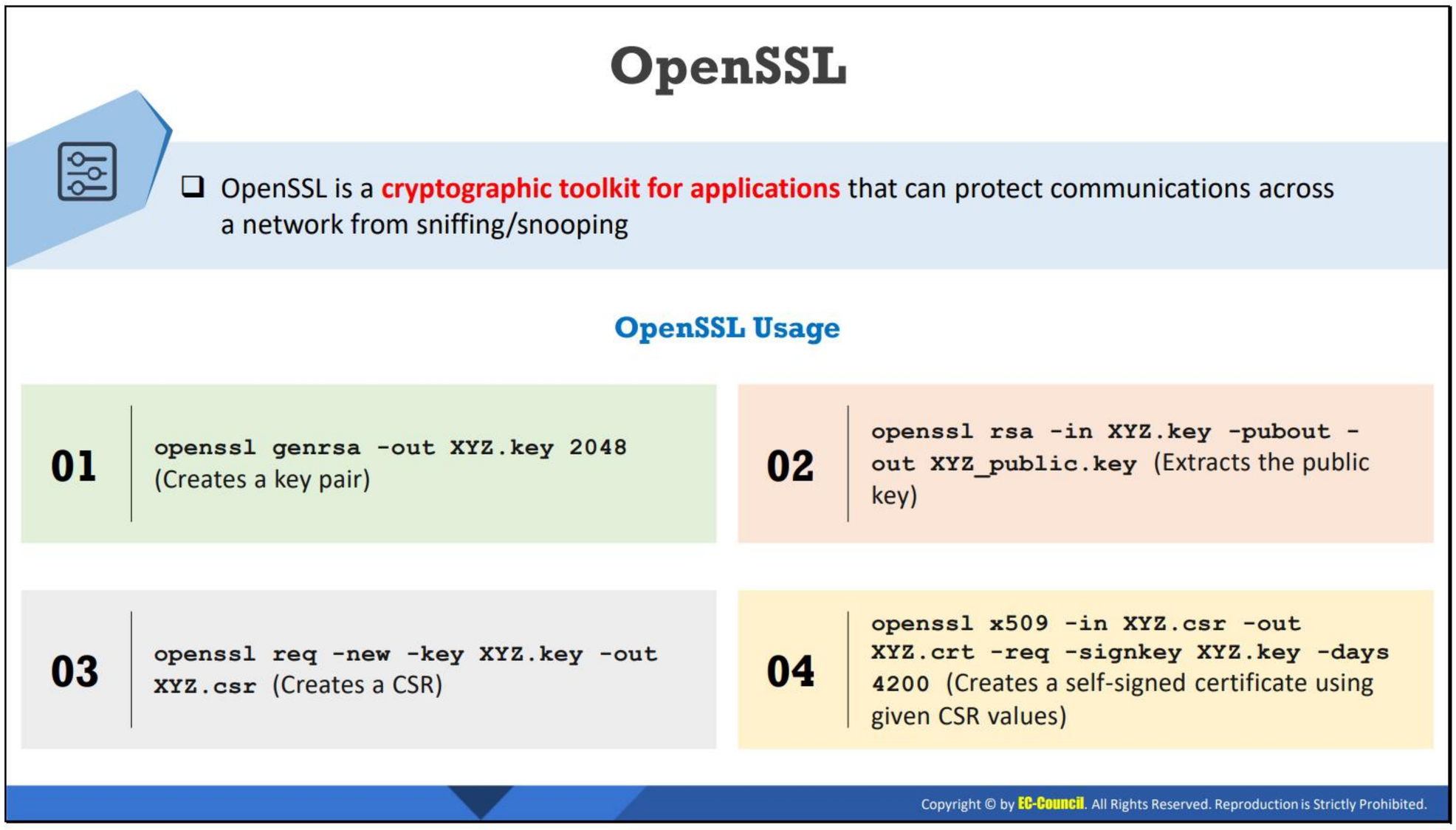


Figure 14.47: Certificate stapling

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



OpenSSL

OpenSSL is a cryptographic library or toolkit for applications that can protect communications across a network from sniffing/snooping and verifies the party at the other end. Most HTTPS sites use this library for secure communication. It comprises a comprehensive implementation of SSL/TSL protocols. OpenSSL offers a rich set of commands, each of which is associated with useful options and arguments. The following are some common operations that can be implemented using OpenSSL.

Creating the Root CA and Certificate Using OpenSSL

- To create a key pair using the RSA algorithm, use the following command:

  ```
  openssl genrsa -out XYZ.key 2048
  ```

- To decode a private key and its associated content, use the following command:

  ```
  openssl rsa -text -in XYZ.key -noout
  ```

- To extract the public key from a key pair, use the following command:

  ```
  openssl rsa -in XYZ.key -pubout -out XYZ_public.key
  ```

Creating Certificate Signing Requests (CSRs)

- To create a CSR, use the following command:

  ```
  openssl req -new -key XYZ.key -out XYZ.csr
  ```

Now, CSR values such as organization name and location can be entered.

- To verify whether the CSR is entered correctly, use the following command:

  ```
  openssl req -text -in XYZ.csr -noout -verify
  ```


Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The “noout” parameter omits the out of the encoded version of the private key.

- To create a self-signed certificate, use the following command:

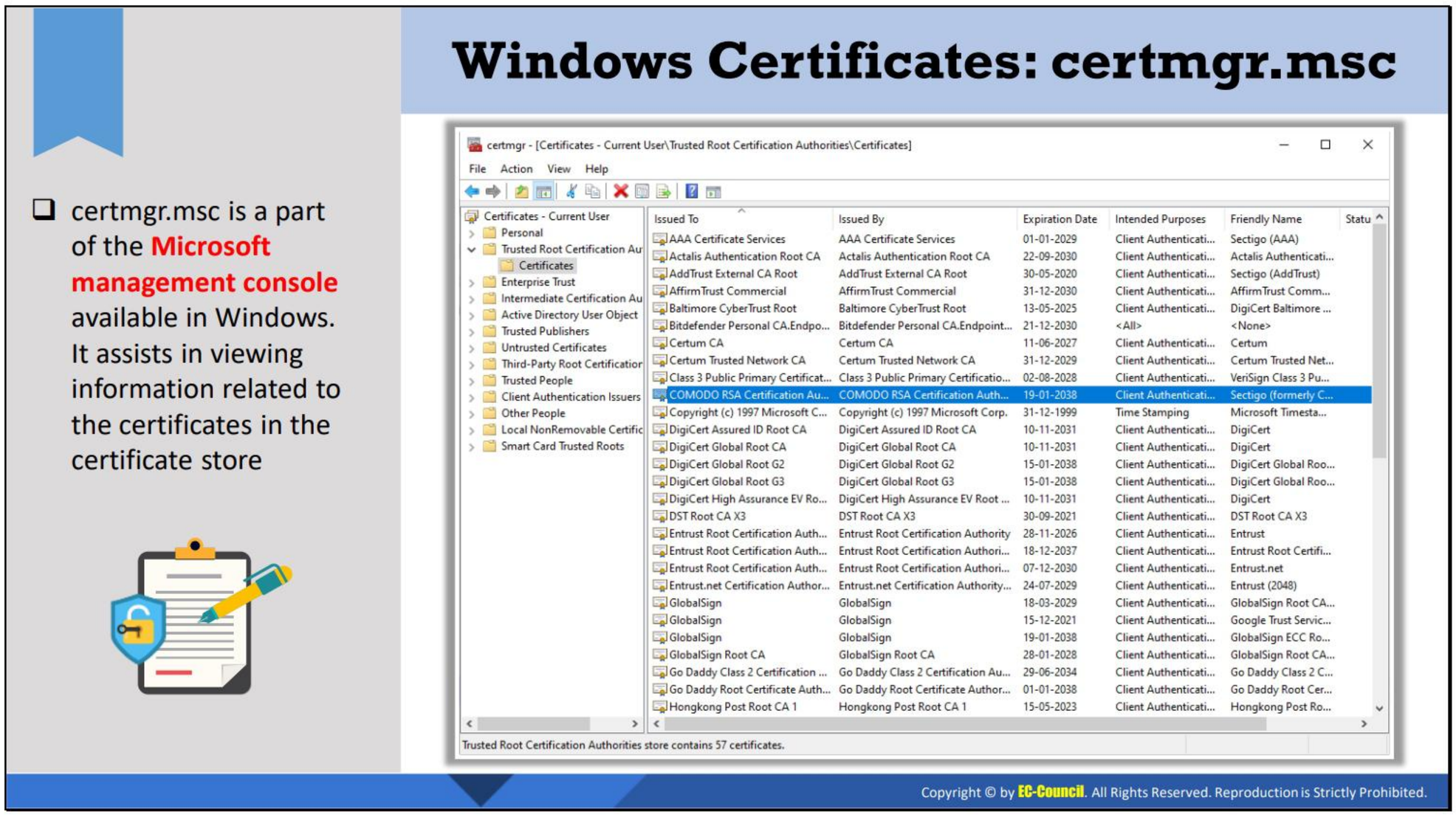
  ```
  openssl x509 -in XYZ.csr -out XYZ.crt -req -signkey XYZ.key -days 4200
  ```

  A validity period of 4200 days is added to the certificate. Administrators can set reminders to receive alerts before certificates expire.

- To review a certificate, use the following command:

  ```
  openssl x509 -in XYZ.crt -text
  ```


Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Windows Certificates: certmgr.msc

certmgr.msc is a part of Microsoft's management console available in Windows 7/8/10 that assists in viewing information related to the certificates in the certificate store. It also helps export/import, alter, remove, and request new certificates.

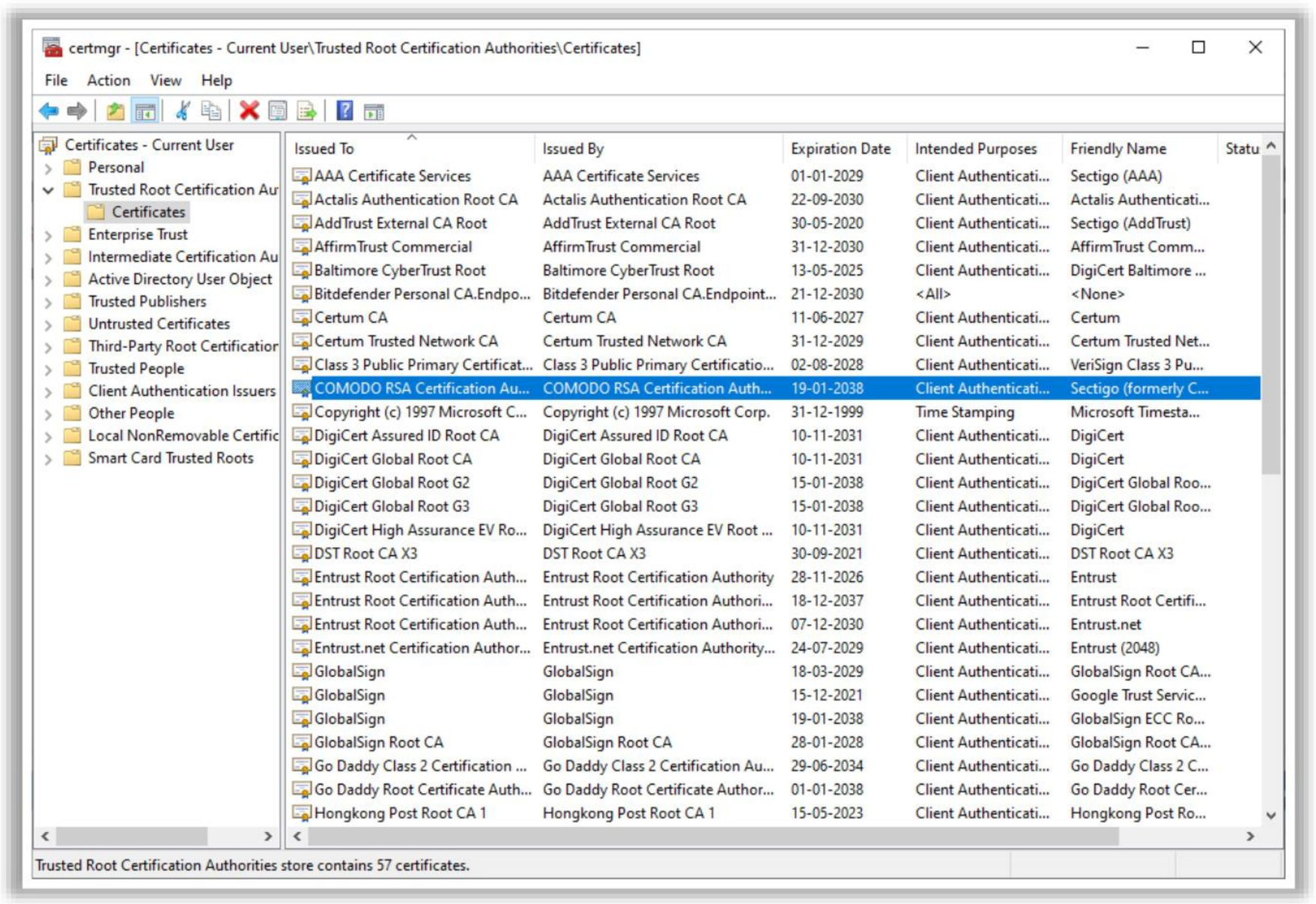
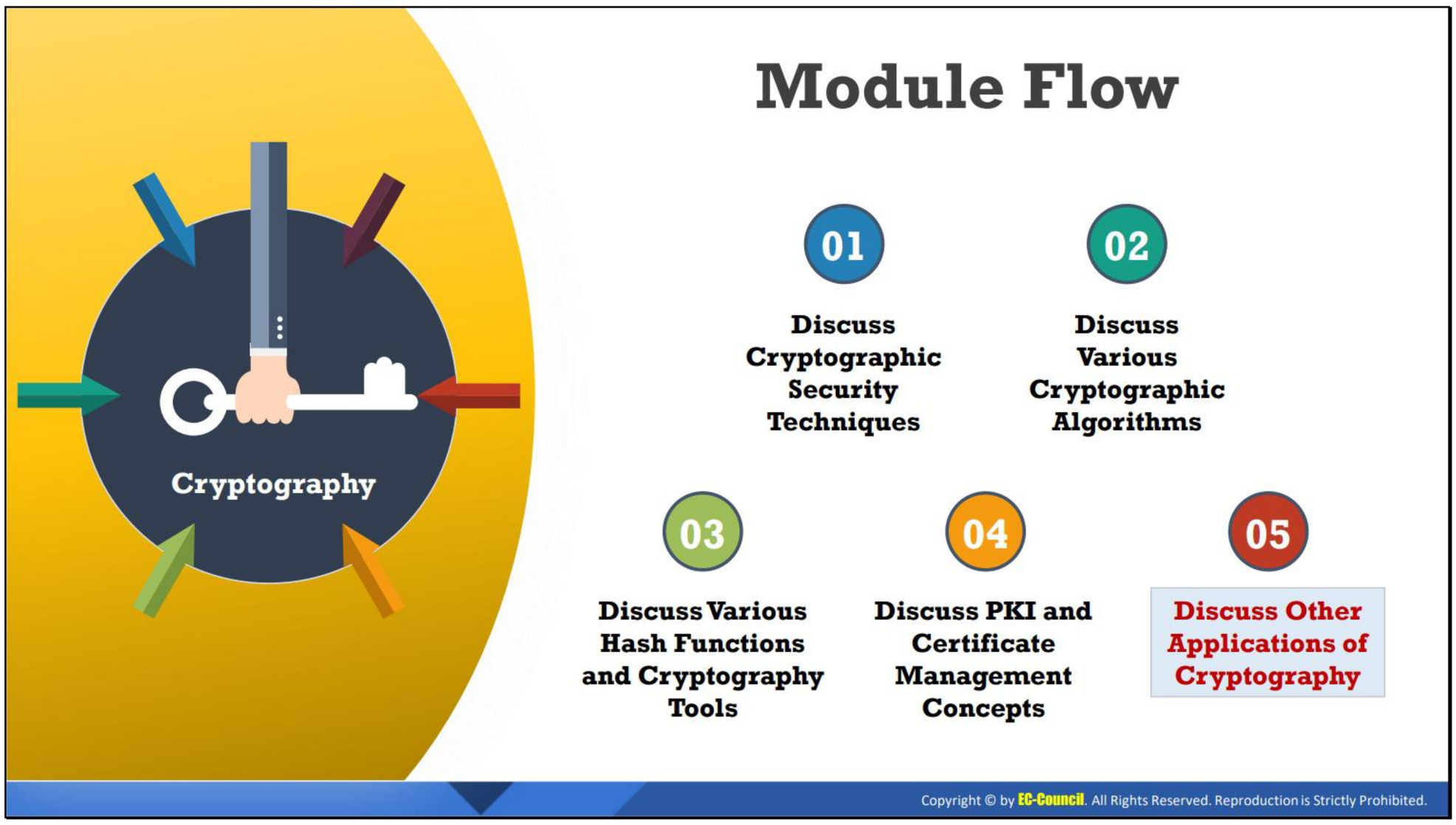


Figure 14.48: Screenshot of certmgr

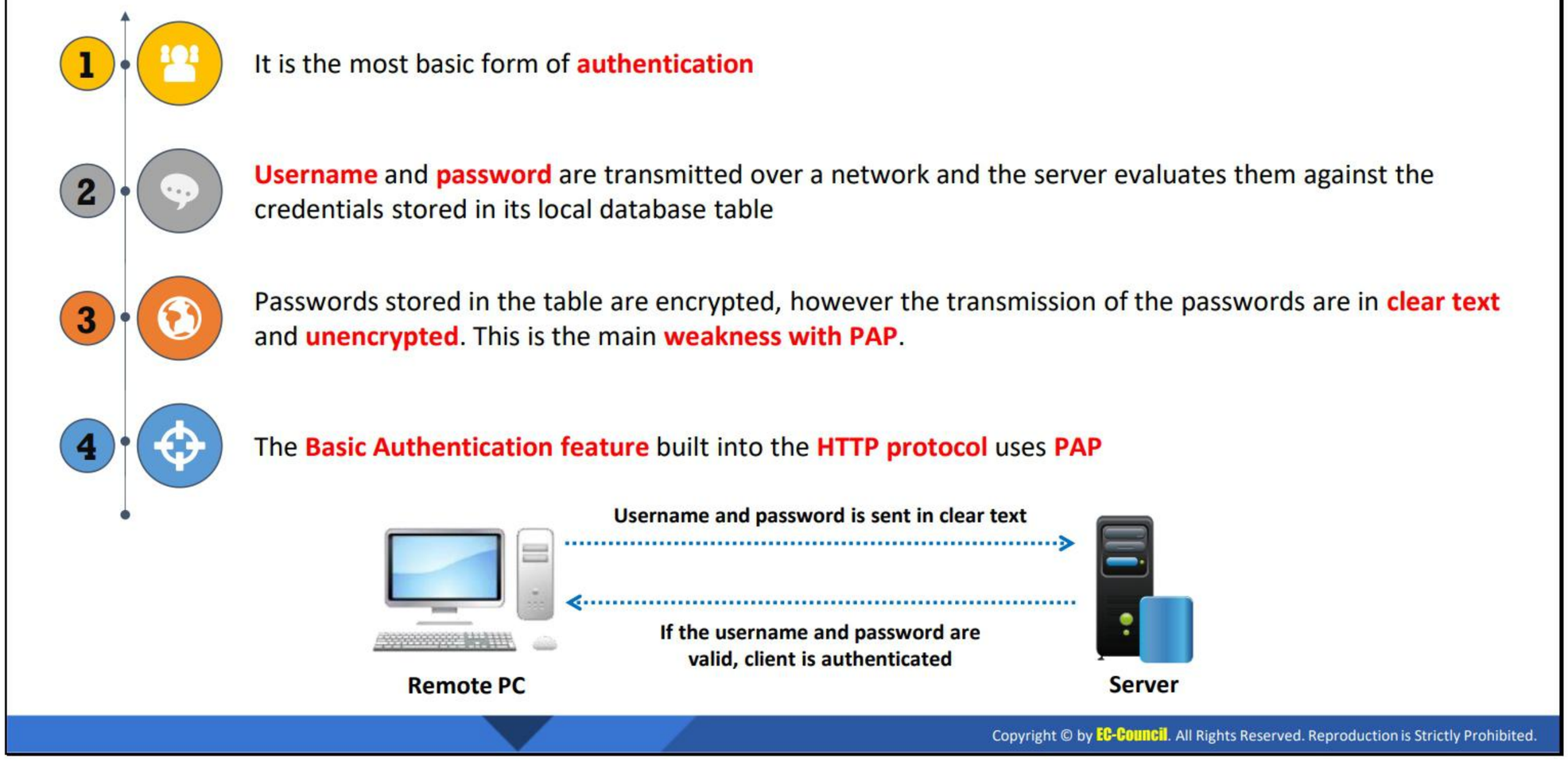
Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Discuss Other Applications of Cryptography

This section discusses other important applications of cryptographic techniques.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Authentication Protocols: Password Authentication Protocol (PAP)

Password Authentication Protocol (PAP) is the simplest form of authentication. With PAP, username and password are transmitted over a network and the server evaluates them against the credentials stored its local database table. Typically, the passwords stored in the table are encrypted; however, the transmission of the passwords are in cleartext and unencrypted. This is the main weakness with PAP. The Basic Authentication feature built into HTTP uses PAP.

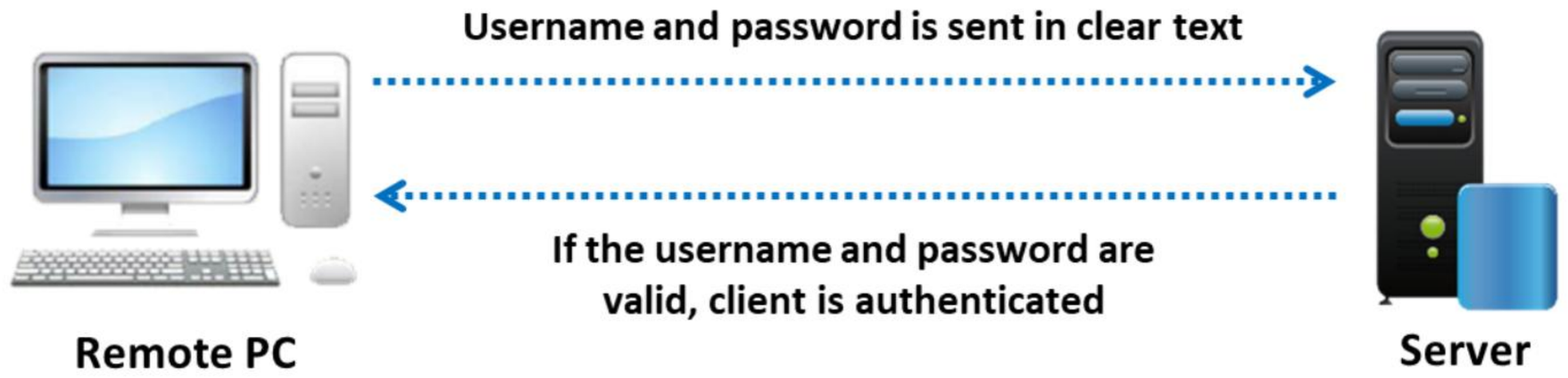
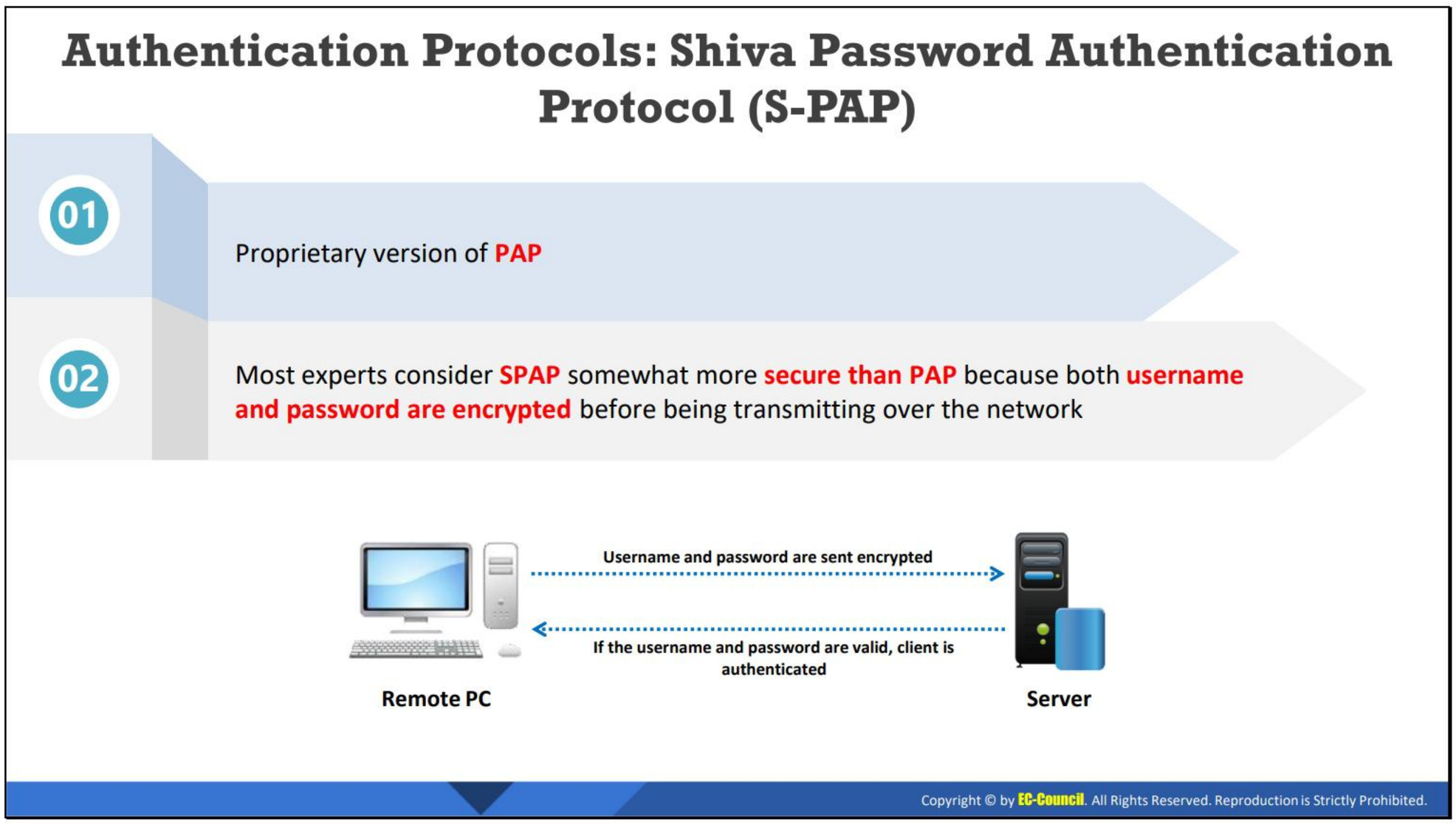


Figure 14.49: Working of PAP

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



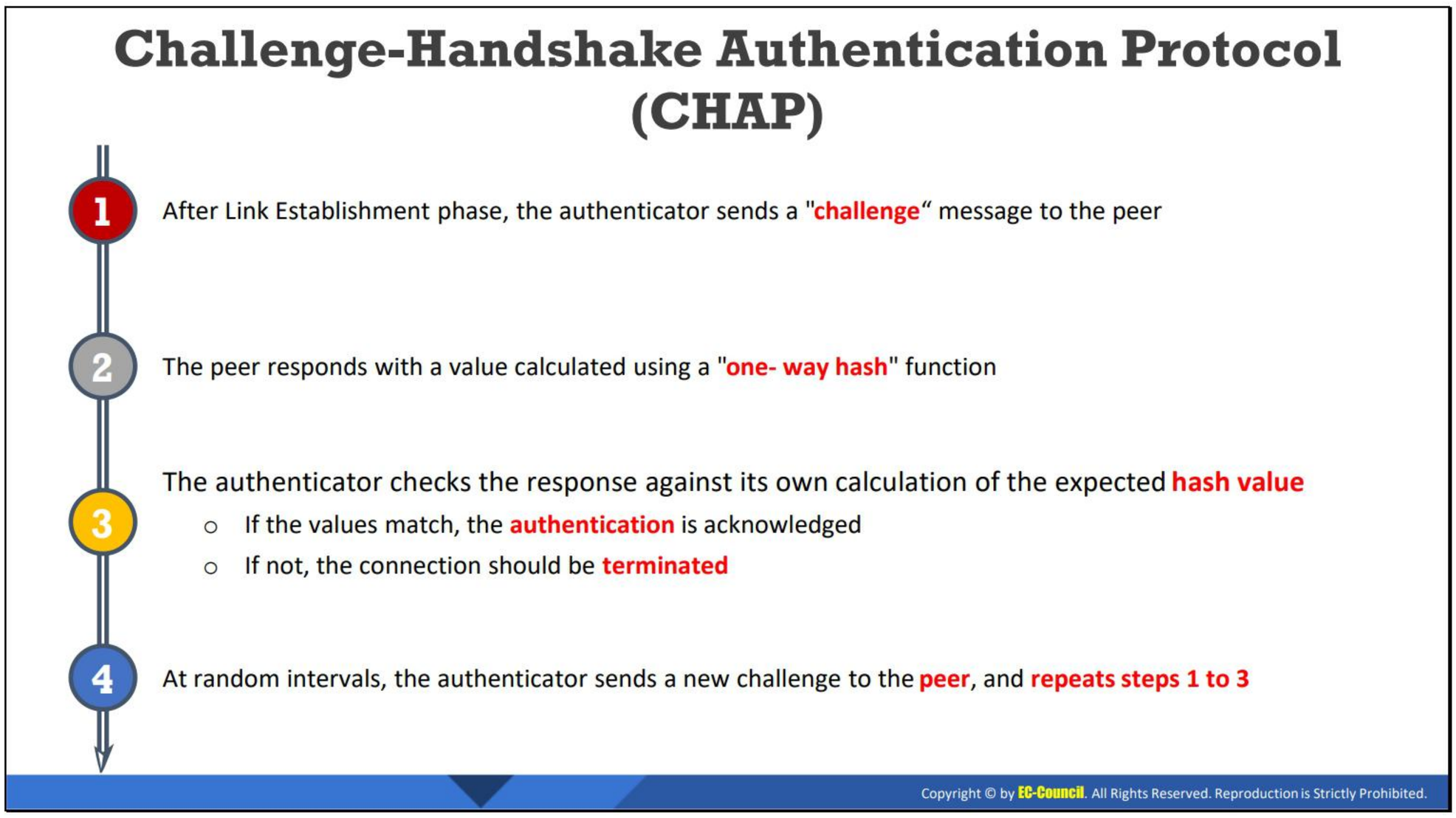
Authentication Protocols: Shiva Password Authentication Protocol (S-PAP)

Shiva Password Authentication Protocol (SPAP) is an extension to PAP that does encrypt the username and password that is sent over the Internet. SPAP is a proprietary version of PAP. Most experts consider SPAP somewhat more secure than PAP. This is because both username and password are encrypted before being transmitting over the network, unlike PAP which sends them in cleartext.



Figure 14.50: Working of S-PAP

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Challenge-Handshake Authentication Protocol (CHAP)

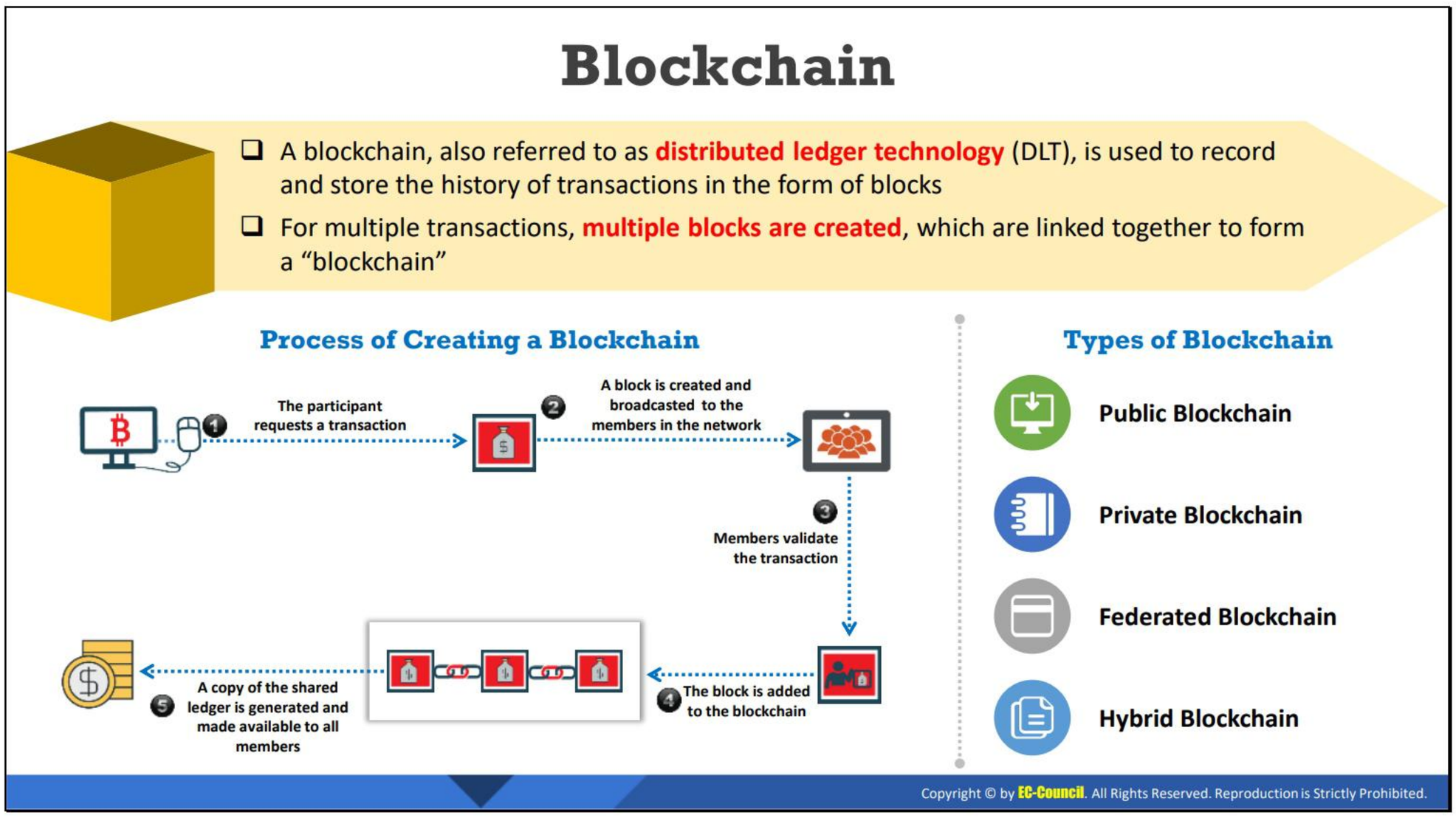
Challenge-Handshake Authentication Protocol (CHAP) calculates a hash soon after the user gets logged in, then it shares that hash with the client system. Periodically, the server will ask the client to provide that hash (this is the challenge part). If the client cannot, then it is clear that the communication has been compromised. MS-CHAP is a Microsoft-specific extension to CHAP. The process for CHAP is as follows:

1. After the link establishment phase is complete, the authenticator sends a “challenge” message to the peer.
2. The peer responds with a value calculated using a hash function.
3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise, the connection is terminated.
4. At random intervals, the authenticator sends a new challenge to the peer, and repeats steps 1 to 3.

Example of CHAP: MS-CHAP v2

Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2) is the successor of MS-CHAP v1 and CHAP developed by Microsoft for handling authentication. MS-CHAP v2 uses Point-to-point Tunneling Protocol version 3 (PPTPv3) for password authentication in RADIUS servers and PPTP-based VPNs; it is further employed as the authentication method in Protected Extensible Authentication Protocol (PEAP). The MS-CHAPv2 implementation includes some new features such as a mutual validation mechanism between peers and a password changing feature for expired passwords in RADIUS servers. Microsoft has recommended implementing or upgrading to MS-CHAP v2 encapsulated with PPTP to protect VPNs running in a nonsecure channel.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Blockchain

A blockchain is a type of distributed ledger technology (DLT) that is used to record and store the history of transactions securely in the form of blocks. Data recorded in blockchains are resistant to unwanted modifications, and account transparency is maintained through cryptographic techniques. For multiple transactions, multiple blocks are created, which are cryptographically linked together to form a "blockchain." This chain of records or blocks are known as ledgers, which are shared in the network to make other participants aware of all the transaction details and the number of bits owned by each member. The members in the network authenticate blocks using their hash values, and the hashes are further validated by crypto miners using complex cryptographic algorithms, following which the blocks are approved to join the blockchain mechanism.

Blockchains are generally implemented using two mechanisms: hash functions (mostly SHA-256) and asymmetric key algorithms. The process of validating blocks is known as "proof of work," for which crypto miners are compensated, and the process of adding blocks to a blockchain after performing "proof of work" is referred to as "crypto mining."

Each block in a blockchain consists of three elements: data (transaction details), hash, and the hash of the previous block. Every time a new block is created using a new hash value, the hash value is shared with the next block. The first block in a blockchain referred to as the genesis and is represented by 0s. Once a block verifies its previous block's hash, it is allowed to join the blockchain.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

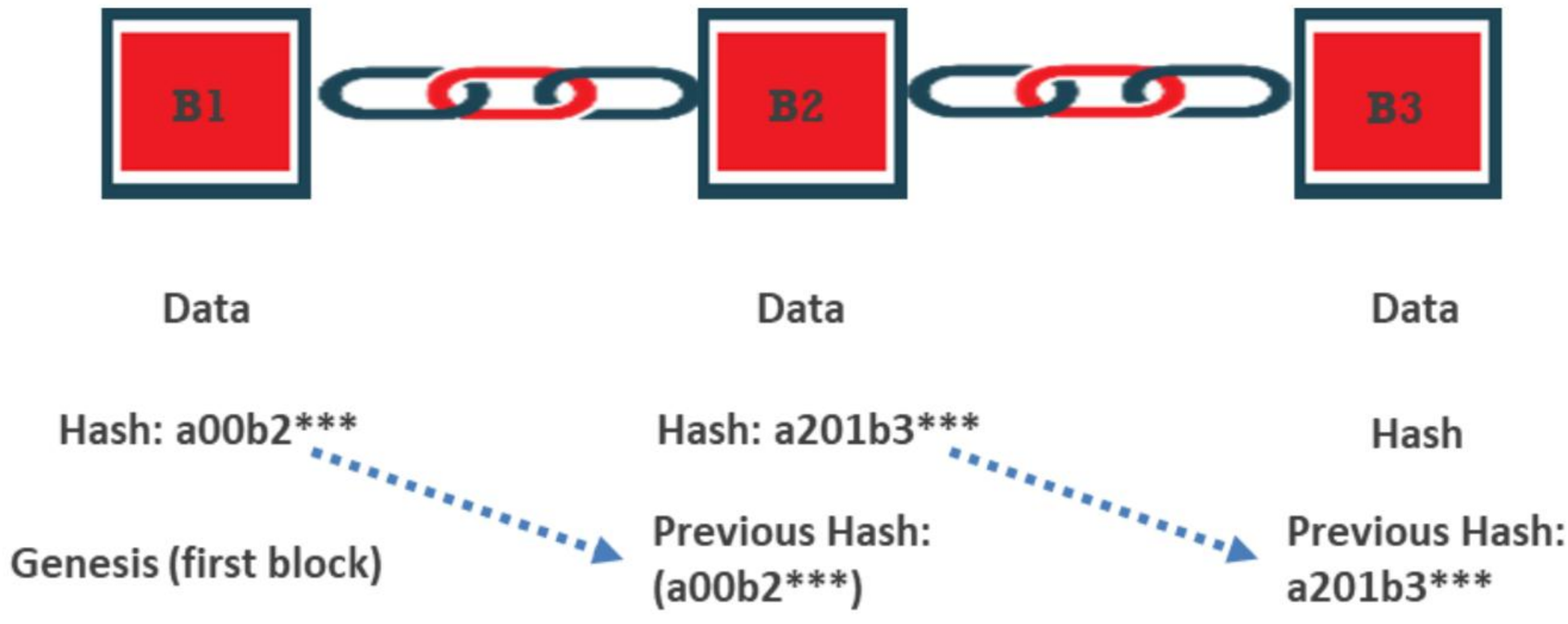


Figure 14.51: Blockchain

If a block is tampered with, the next block in the chain invalidates it because it does not match the previous hash value in the current block. However, a blockchain is not completely protected by merely generating hashes and comparing them with other blocks. Attackers can generate valid hashes for each block using many cryptographic techniques. The "proof of work" mechanism is used, as described above, to mitigate such risks. The security of the blockchain is ensured by both the effective usage of hashes and the "proof of work" by miners (on public ledgers).

Figure below illustrates the process of creating a blockchain.

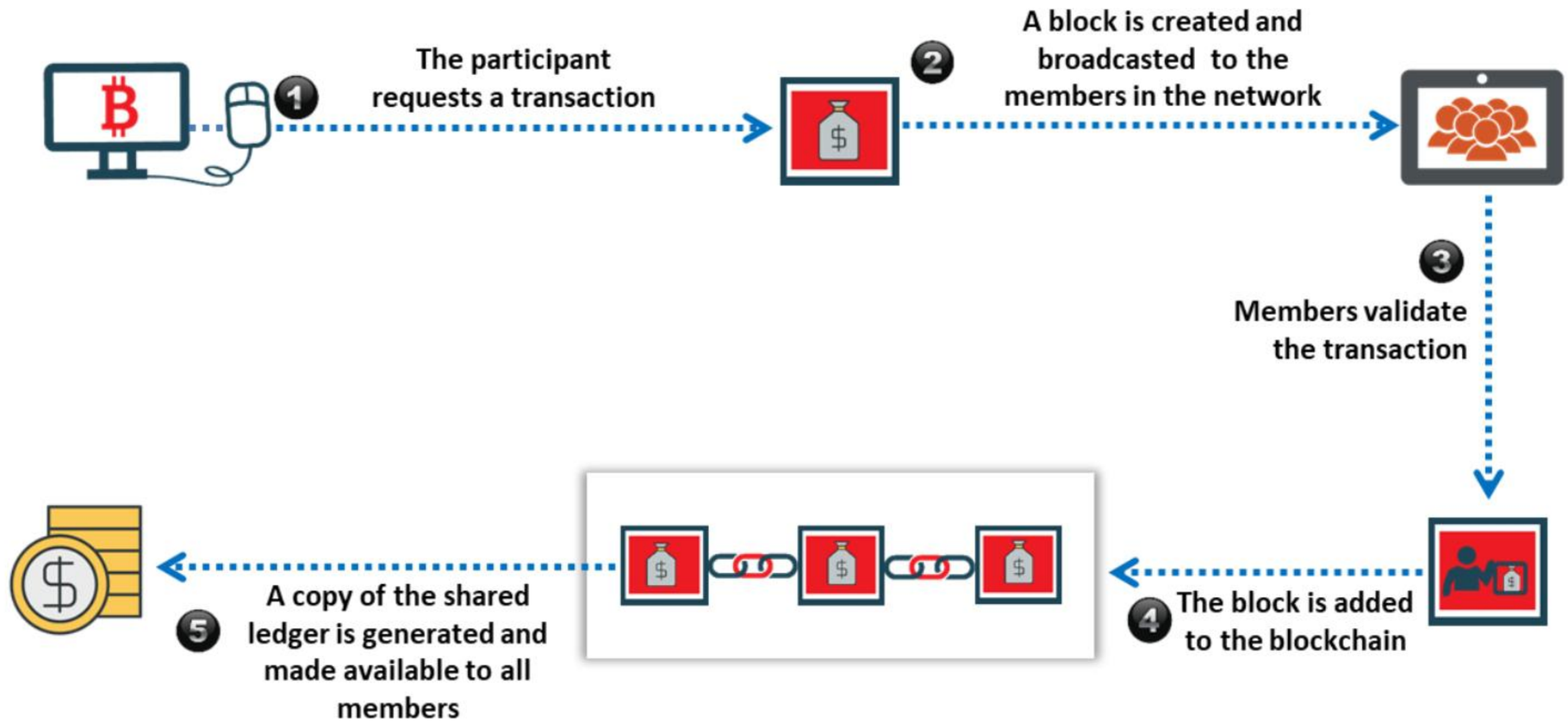


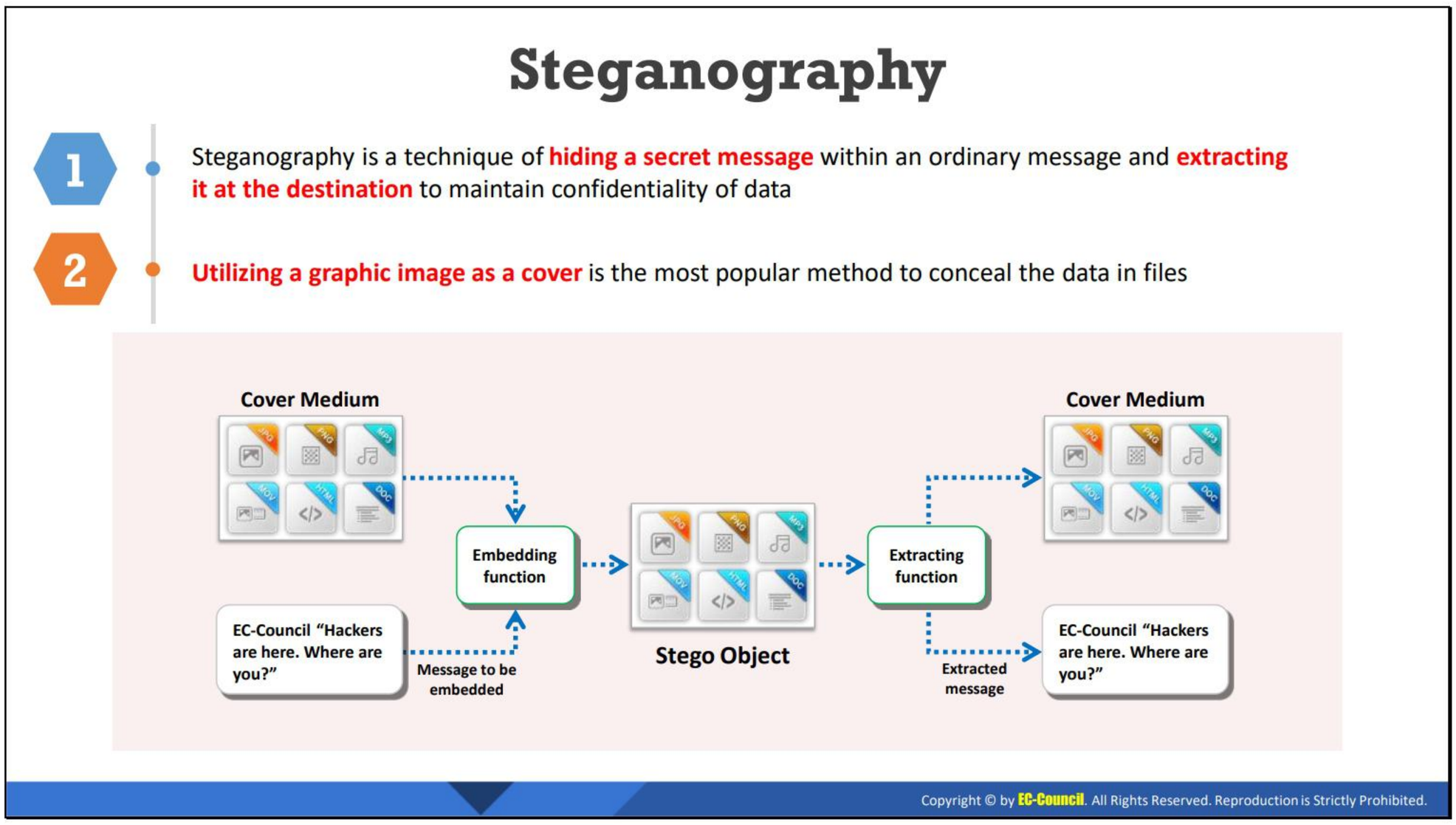
Figure 14.52: Creation of a blockchain

In figure, a block is created by a person involved in a transaction. This block is shared with all the members in the network. Each member validates the block using its hashes, following which the block is added to the blockchain. Thus, each member has the details of the new transaction. Blockchains can be created in four variants, each of which serves a different purpose.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Public ledger or public blockchain:** This type of blockchain has no central authority or administration to manage the blocks or ledgers. It is a decentralized and permission-less network in which anyone can join, create, and share blocks. Once the data on the blockchain have been validated, it is secure from modifications or alterations. Some examples of public blockchains include Bitcoin and Ethereum. Each member in this blockchain can access a copy of other ledgers without any permissions. The following are the key aspects of public ledgers.
  - Everyone in the network can participate in validation.
  - Once a block is created, it cannot be modified or tampered with.
  - Public ledgers can be employed in different sectors such as education and healthcare.
  - Public ledgers are suitable for B2C services.
- **Private ledger or private blockchain:** In this type of blockchain, a supervisor or central authority decides who can join and participate in the blockchain network. In a private ledger, only the members involved in a transaction will have knowledge about the corresponding ledgers. Some examples of private blockchains are Hyperledger and Ripple (XRP). The following are the key aspects of private ledgers.
  - An administrator provides a certain level of access to participants.
  - Organizations can add and delete participants on demand.
  - Private ledgers can be employed in sectors such as defense and banking.
  - Private ledgers are suitable for B2B services.
- **Federated blockchain or consortium blockchain:** It is a partially decentralized blockchain in which a group of individuals or organizations, rather than a single entity as in private blockchains, create and manage separate blockchain networks. Control over the blockchain is provided to a group of predetermined or trusted nodes. Participants in a consortium blockchain are mostly from government organizations or central banks. This type of blockchain is extremely fast and scalable. EWF (Energy) and R3 (banks) are instances of federated blockchains.
- **Hybrid blockchain**: It is a combination of both private and public blockchain. In a hybrid blockchain, only a selected set of records or data from the blockchain can be publicly accessed; the remaining data are kept confidential in a private network. This type of blockchain enables organizations to select which data they wish to make public private. One important example of a hybrid blockchain is the IBM Food Trust.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Steganography

Steganography is a technique of hiding a secret message within an ordinary message and extracting it at the destination to maintain confidentiality of data. It replaces bits of unused data into ordinary files, such as graphics, sound, text, audio, and video with other surreptitious bits. The hidden data can be in the form of plaintext or ciphertext, and sometimes, an image. Utilizing a graphic image as a cover is the most popular method to conceal the data in files. Unlike encryption, the detection of steganography can be challenging. Thus, steganography techniques are widely used for malicious purposes.

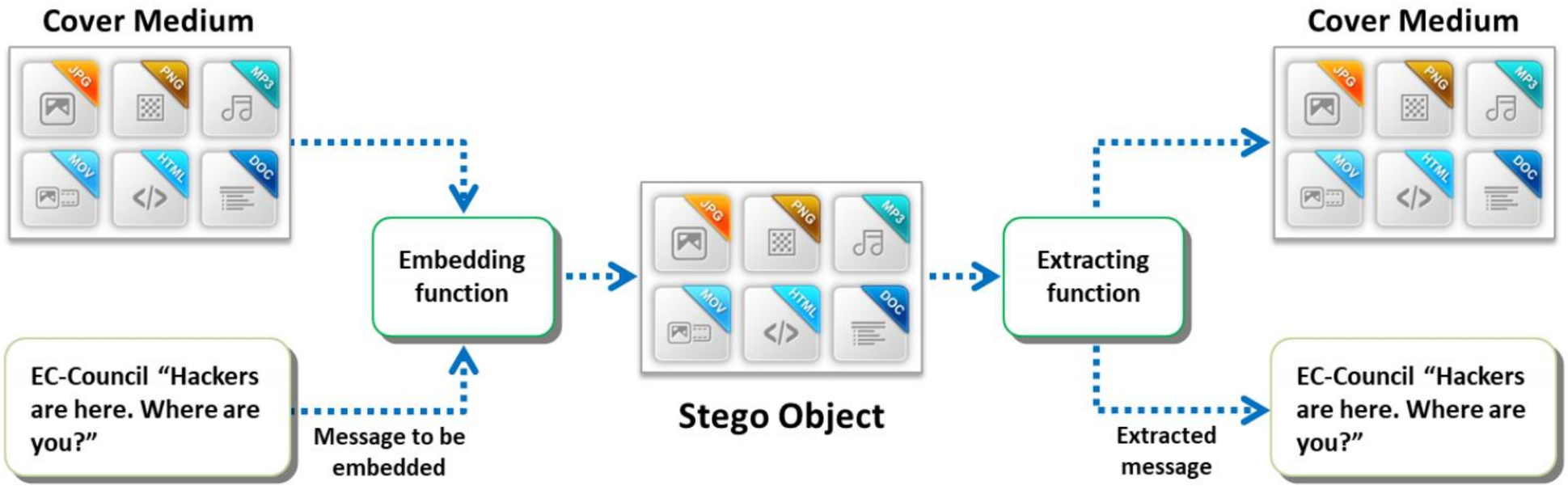
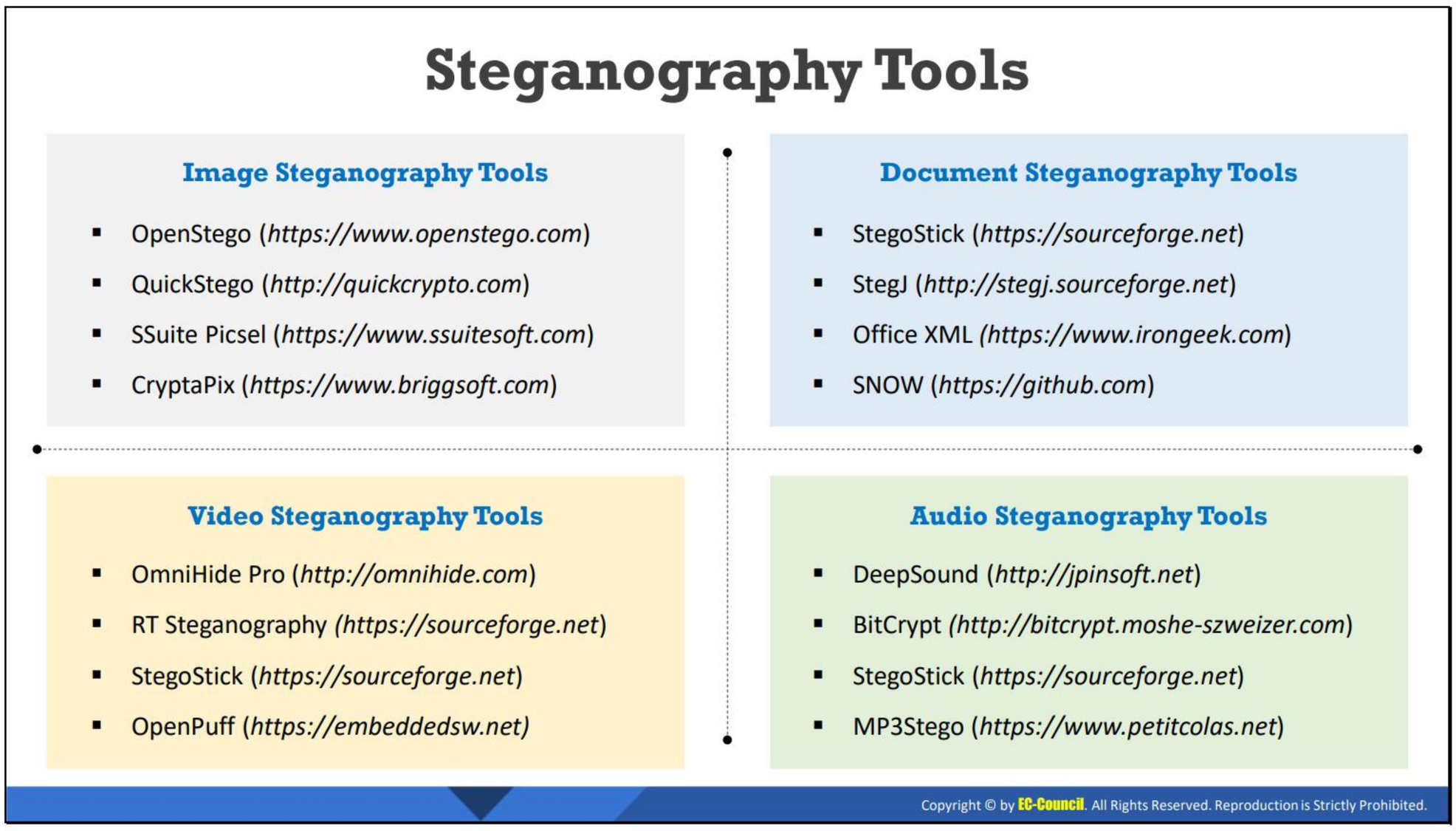


Figure 14.53: Hiding message using steganography

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Steganography Tools

Listed below are various steganography tools:

Image Steganography Tools

- OpenStego (*https://www.openstego.com*)
- QuickStego (*http://quickcrypto.com*)
- SSuite Picsel (*https://www.ssuitesoft.com*)
- CryptaPix (*https://www.briggsoft.com*)

Document Steganography Tools

- StegoStick (*https://sourceforge.net*)
- StegJ (*http://stegj.sourceforge.net*)
- Office XML (*https://www.irongeek.com*)
- SNOW (*http://www.darkside.com.au*)

Video Steganography Tools

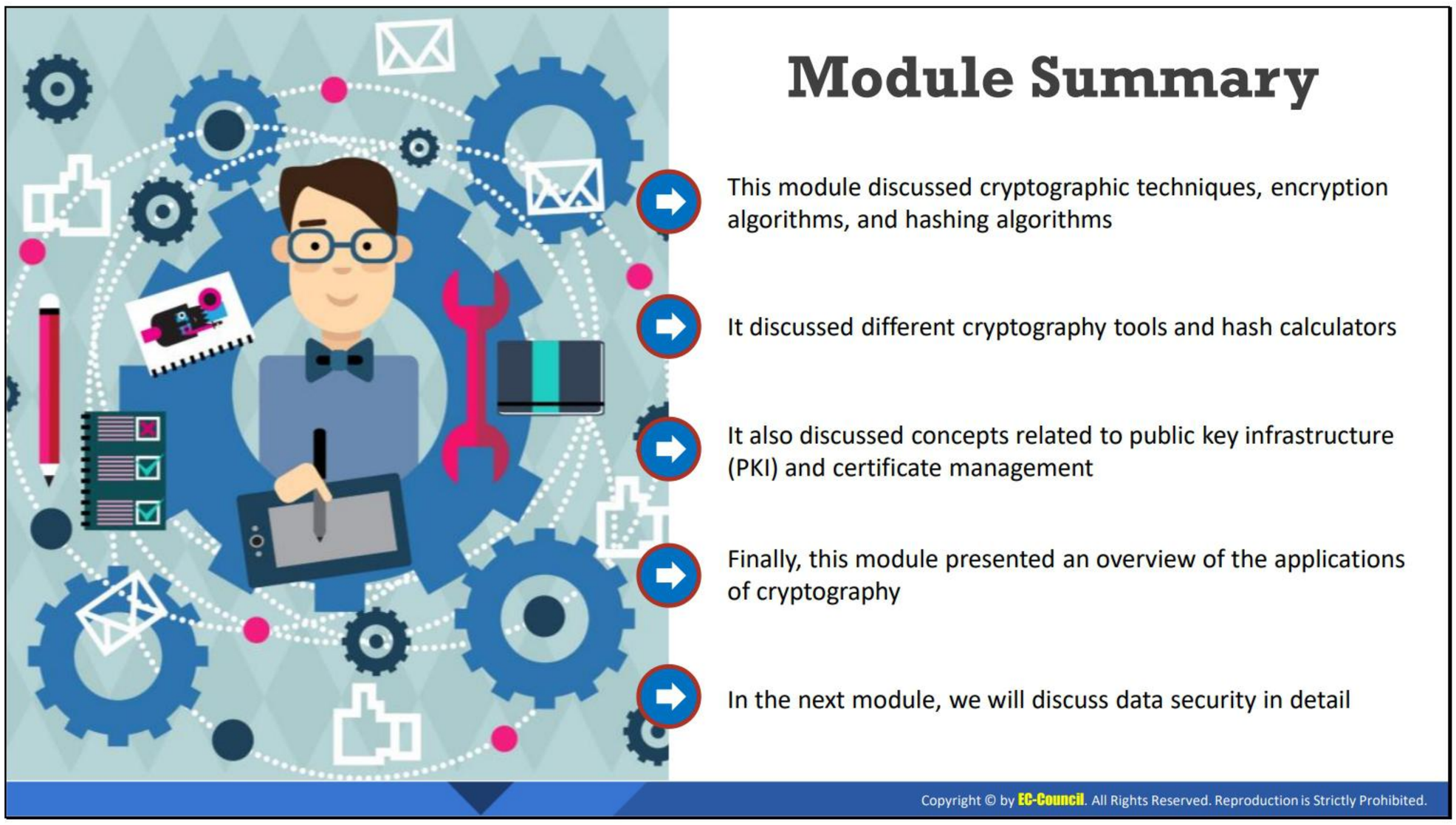
- OmniHide Pro (*http://omnihide.com*)
- RT Steganography (*https://rtstegvideo.sourceforge.net*)
- StegoStick (*https://sourceforge.net*)
- OpenPuff (*https://embeddedsw.net*)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Audio Steganography Tools

- DeepSound (*http://jpinsoft.net*)
- BitCrypt (*http://bitcrypt.moshe-szweizer.com*)
- StegoStick (*https://sourceforge.net*)
- MP3Stego (*https://www.petitcolas.net*)

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Module Summary

This module discussed cryptographic techniques, encryption algorithms, and hashing algorithms. Further, it discussed different cryptography tools and hash calculators. It also discussed concepts related to public key infrastructure (PKI) and certificate management. Finally, this module presented an overview of the applications of cryptography.

In the next module, we will discuss data security in detail.

Certified Cybersecurity Technician Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

This page is intentionally left blank.