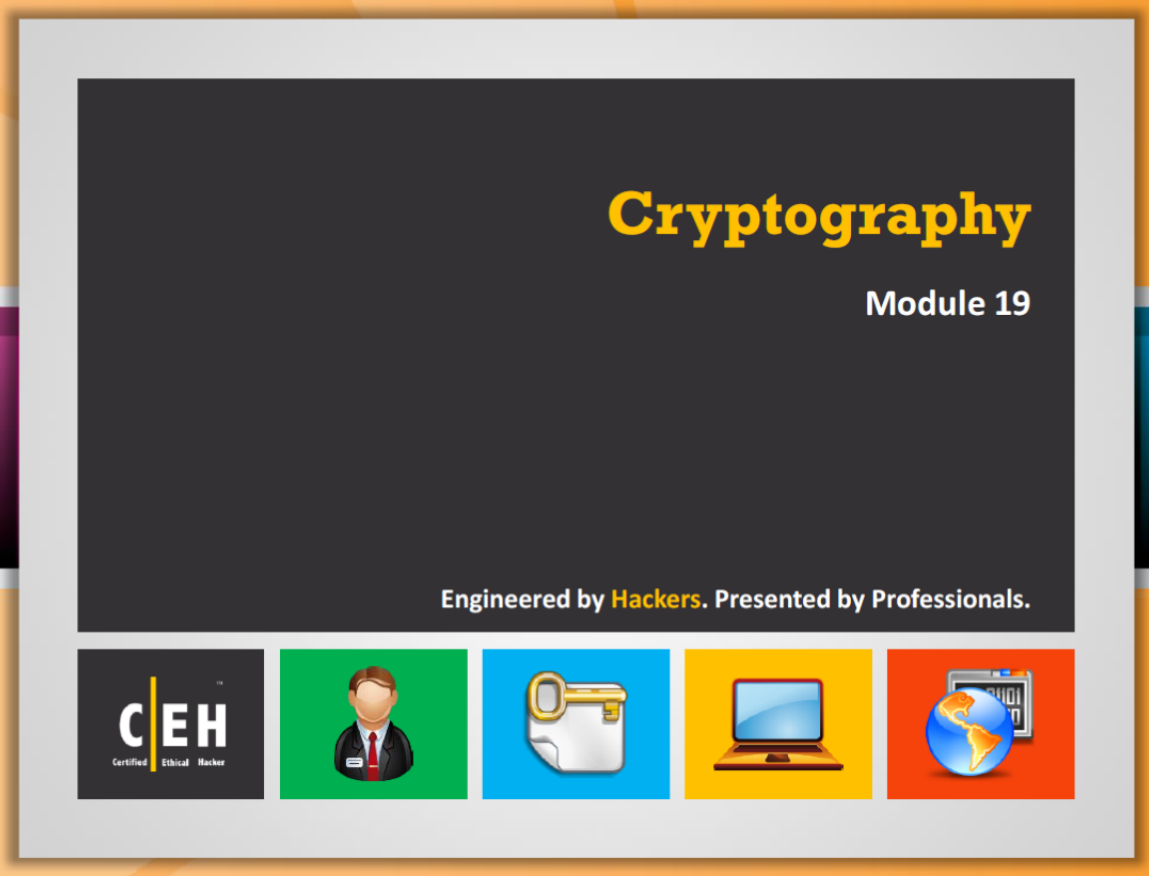
# Cryptography

## Module 19




# Ethical Hacking and Countermeasures v8

## Module 19: Cryptography

## Exam 312-50



## Security News

### Ransom Malware Hits Australia as 30 Businesses Attacked

Source: http://news.techworld.com

The 2012 epidemic of ransom malware appears to have turned even nastier with reports that as many as 30 Australian businesses have now asked police for help coping with attacks in a matter of days.

According to local news, police in the state of **Queensland** have received reports from a dozen businesses while many other are believed to have chosen to keep incidents to themselves.

Businesses affected included those in the medical, entertainment, retail and insurance sectors, the news source said, with several dozen affected in total.

In one recent incident, a business in the Northern Territories reportedly paid an AUD $3,000 (about £2,000) ransom via Western Union to get back access to important financial records, including credit card data and debtor invoices. The attackers demanded the money within seven days or the sum would increase by **AUD $1,000 per week**.

Worryingly, this attack used, to all intents and purposes impossible to crack if the key has not been exposed during the attack.

"A lot of businesses can't afford the interruptions to their trade and will pay straight away," detective superintendent Brian Hay of **Queensland's fraud** and corporate crime group told press.

Ransom malware has become a serious issue during 2012, although its effect on businesses is rarely recorded. Most of the data that has become public has been in the form of police warnings based on attacks against consumers.

Most attacks simply attempt to engineer users into believing their files are encrypted when they are not or make more general threats, often to report victims to national police for non-existent crimes.

The use of industrial-strength encryption is rare although this sort of technique is actually where the form started as long ago in 2006 with a piece of malware called '**Cryzip**.'

In August, the FBI said it had been "**inundated**" with ransom malware reports from consumers, not long after the UK's Police Central e-Crime Unit (PCeU) publicised an identical spate of attacks that had affected over a thousand PCs in the UK.

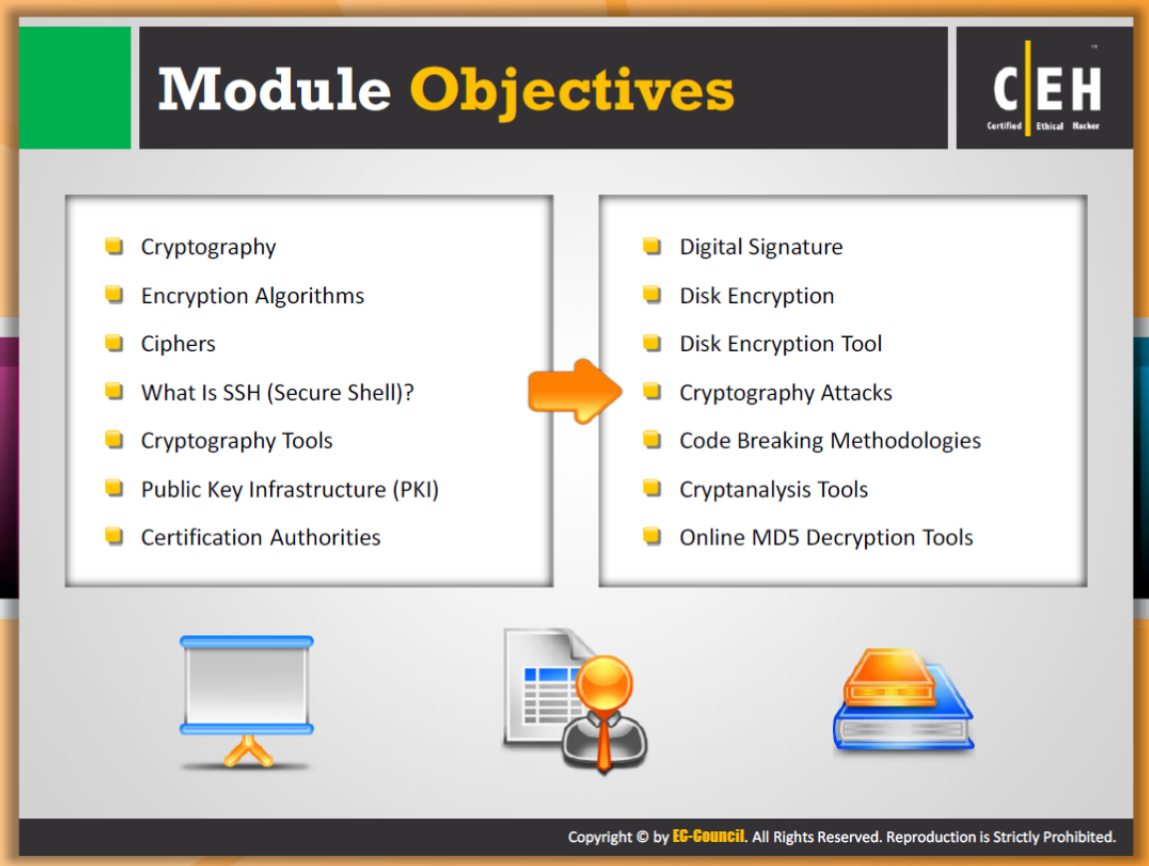
In the past the few security companies that have investigated the issue have pinned the blame on a single cabal of Russian criminals that seem able to operate with impunity. Now the same tactics appear to have spread to gangs in nearby countries such as the **Ukraine** and **Romania**.

The suspicion is that some security vendors say little about the problem because not only is their software unable to stop infections but they can't always unlock the files after the fact either.



*By: John E Dunn*

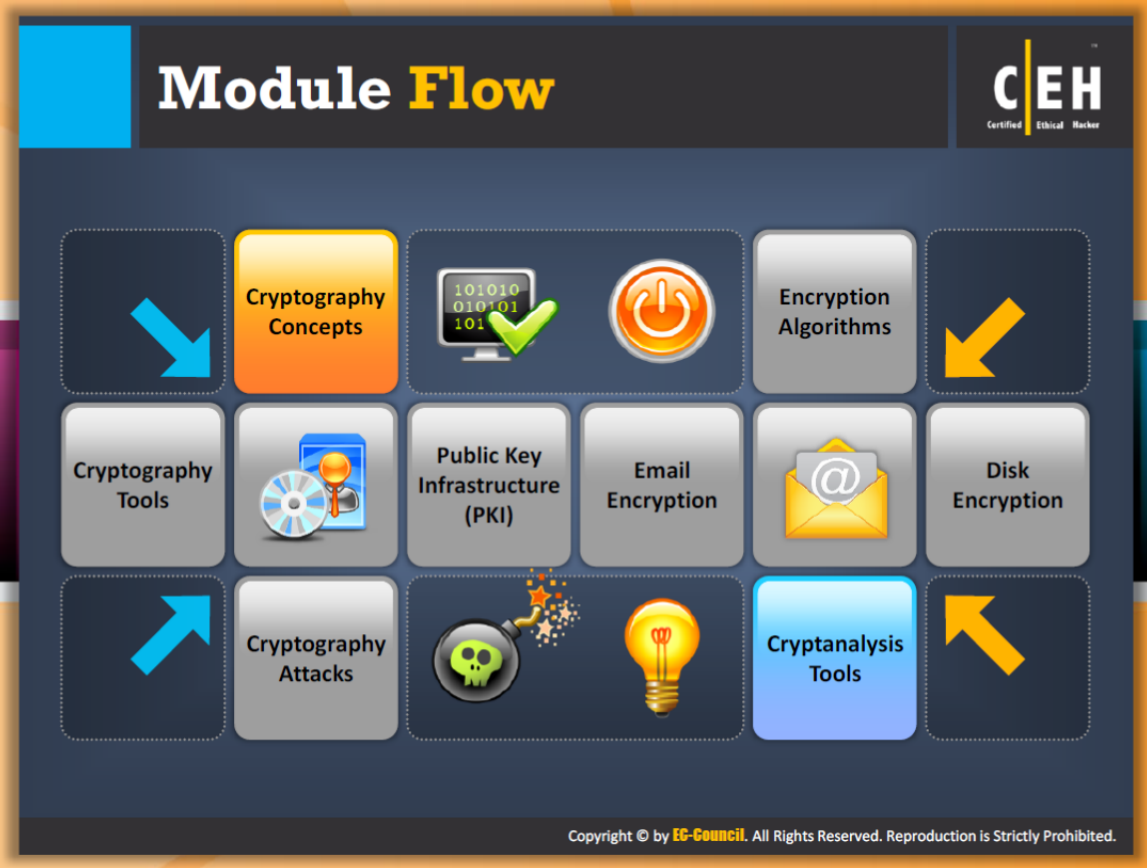
http://news.techworld.com/security/3401328/ransom-malware-hits-australia-as-30-businesses-attacked/




## Module Objectives

Having dealt with various security concerns and countermeasures in the preceding modules, it is obvious that cryptography, as a security measure, is here to stay. This module will familiarize you with:

- Cryptography
- Encryption Algorithms
- Ciphers
- What Is SSH (Secure Shell)?
- Cryptography Tools
- Public Key Infrastructure (PKI)
- Certification Authorities
- Digital Signature
- Disk Encryption
- Disk Encryption Tool
- Cryptography Attacks
- Code Breaking Methodologies
- Cryptanalysis Tools
- Online MD5 Decryption Tools

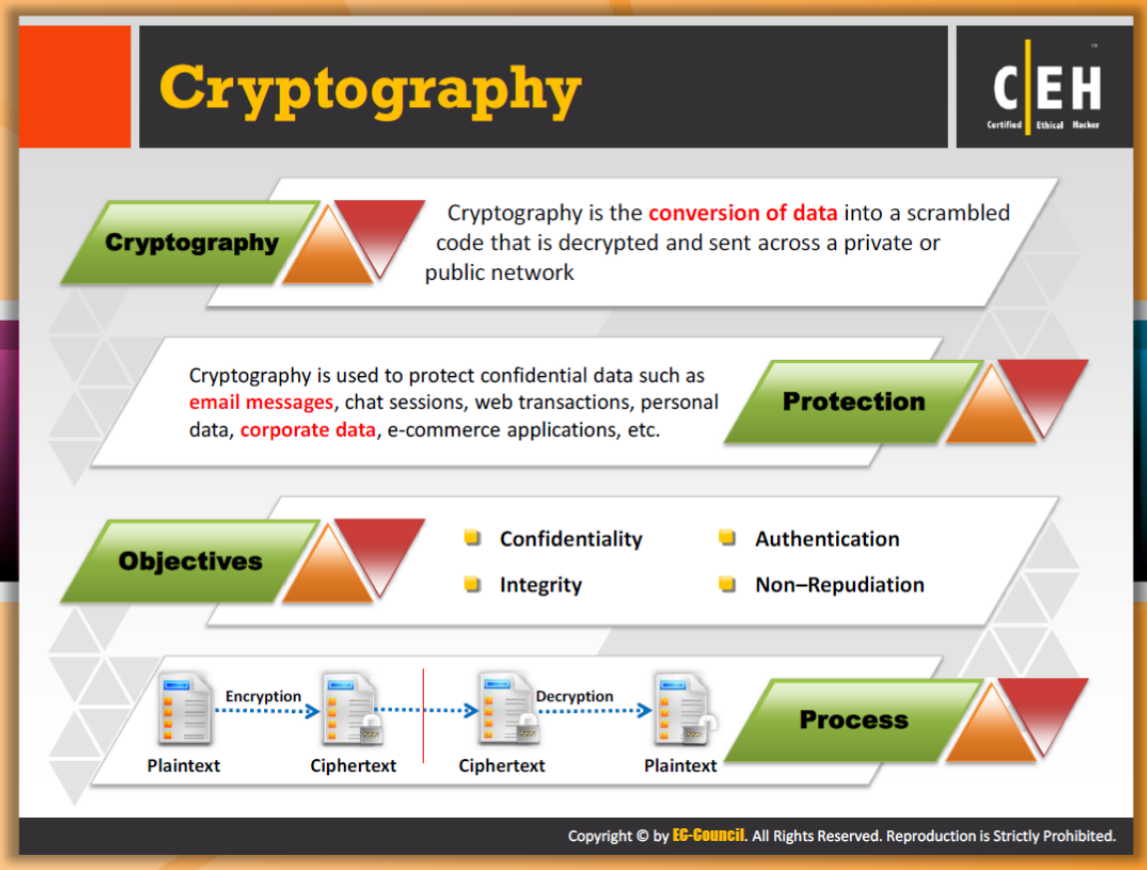



## Module Flow

To understand cryptography security measures, let's begin with **cryptography** and its **associated concepts**.

| | |
|---|---|
| Cryptography Concepts | Encryption Algorithms |
| Cryptography Tools | Public Key Infrastructure (PKI) |
| Email Encryption | Disk Encryption |
| Cryptography Attacks | Cryptanalysis Tools |

This section describes cryptography and the types of cryptography.




## Cryptography

Everyone has secrets, and when it is necessary to transfer that secret information from one person to another, it's very important to protect that information or data during the transfer. Cryptography takes plaintext and transforms it into an unreadable form (ciphertext) for the purpose of maintaining security of the data being transferred. It uses a key to transform it back into readable data when the information reaches its destination. The word crypto is derived from the Greek word kryptos. **Kryptos** was used to depict anything that was **concealed**, hidden, veiled, secret, or mysterious. Graph is derived from graphia, which means writing; hence, cryptography means the art of "the secret writing."

Cryptography is the study of mathematical techniques involved in information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography transforms plaintext messages to ciphertext (encrypted messages) by means of encryption. Modern cryptography techniques are virtually unbreakable, though it is possible to break encrypted messages by means of **cryptanalysis**, also called code breaking. There are four main objectives of cryptography:

## Confidentiality

According to the **International Standards Organization** (ISO), confidentiality is "ensuring that the information/data can be accessed only by those authorized." Confidentiality is the

term used to describe the prevention of revealing information to unauthorized computers or users.

Any breach in confidentiality may lead to both financial and emotional distress. There have been instances of organizations going bankrupt due to a system breach by **rival organizations**. Moreover, personal information in the wrong hands can ruin the lives of system users. Therefore, only authorized users should possess access to information.

## Integrity

Integrity is "ensuring that the information is accurate, complete, reliable, and is in its original form." Valuable information is stored on the computer. Any data corruption/modification can reduce the value of the information. The damage that data corruption/modification can do to an organization is **unfathomable**.

Integrity of the data is affected when an insider (employee) of an organization or an attacker deletes/alters important files or when malware infects the computer.

Although it may be possible to restore the modified data to an extent, it is impossible to restore the value and reliability of the information.

Examples of violating the data integrity include:

- A frustrated employee deleting important files and modifying the payroll system
- **Vandalizing** a website and so on


## Authentication

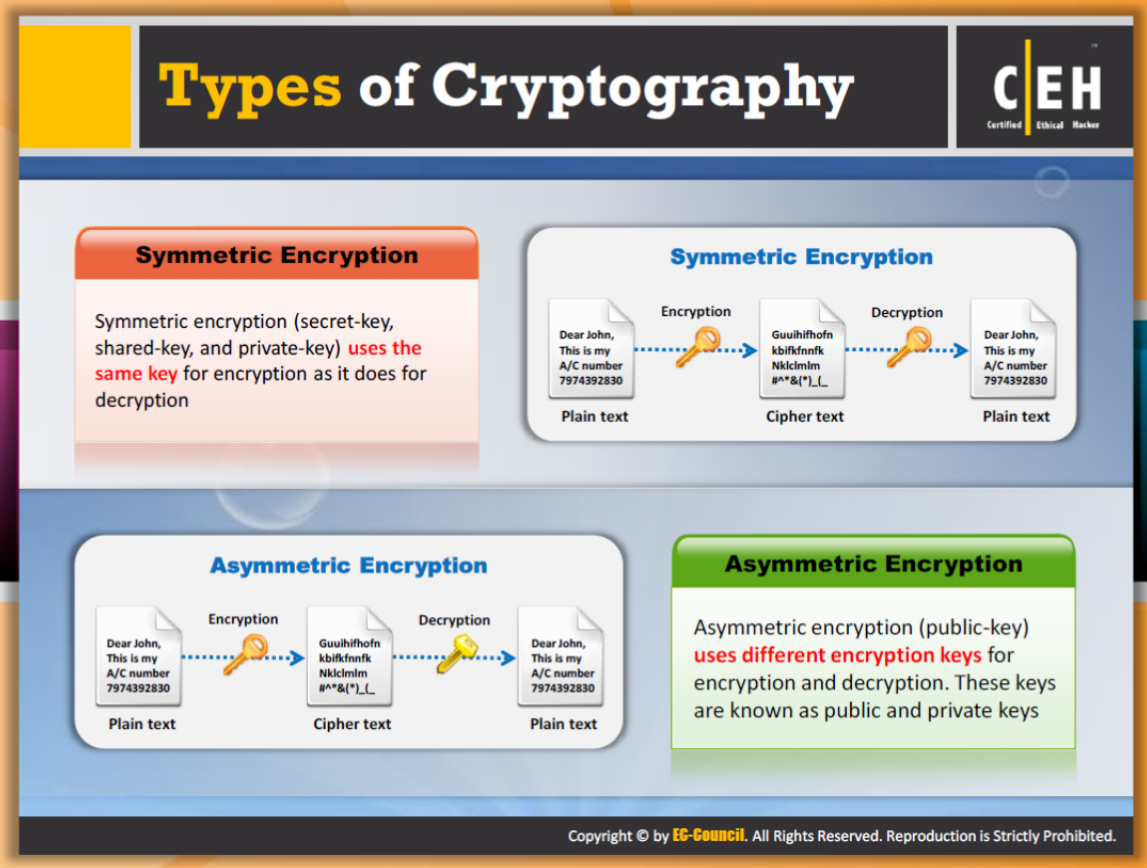
Authenticity is "the identification and assurance of the origin of information." It is important to ensure that the information on the system is authentic and has not been tampered with. It is also important to ensure that the computer users or those who access information are who they claim to be.

## Nonrepudiation

In digital security, **nonrepudiation** is the means to ensure that a message transferred has been sent and received by the persons or parties who actually intended to. Let us assume that party A is sending a message M with the signature S to the party B. Then party A cannot deny the authenticity of its signature S. It can be obtained through the use of:

- **Digital signatures:** A digital signature functions as unique identifier for an individual, like a written signature. It is used to ensure that a message or document is electronically signed by the person.
- **Confirmation services:** It is possible to indicate that messages are received and/or sent by creating digital receipts. These digital receipts are generated by the message transfer agent.


FIGURE 19.1: Illustrating cryptography process




## Types of Cryptography

The following are the two types of cryptography:

- Symmetric encryption (secret key cryptography)
- Asymmetric encryption (public key cryptography)

## Symmetric Encryption

The symmetric encryption method uses the same key for encryption and decryption. As shown in the following figure, the sender uses a key to encrypt the plaintext and sends the ciphertext to the receiver. The receiver **decrypts** the **ciphertext** with the same key that is used for encryption and reads the message in plaintext. As a single secret key is used in this process symmetric encryption is also known as secret key cryptography. This kind of **cryptography** works well when you are communicating with only a few people.

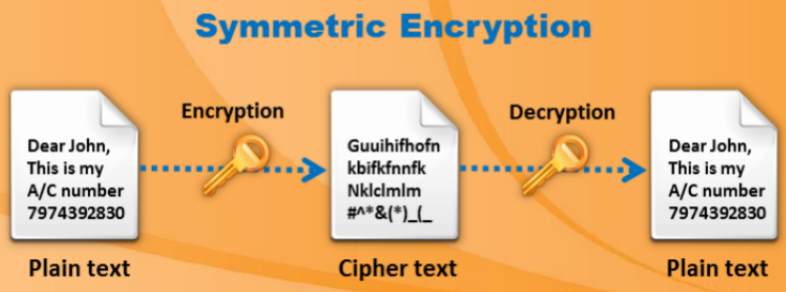



FIGURE 19.2: Symmetric Encryption method

The problem with the secret key is transferring it over the large network or Internet while preventing it from falling into the wrong hands. In this process, anyone who knows the secret key can decrypt the message. This problem can be fixed by **asymmetric encryption**.

## Asymmetric Encryption

Asymmetric cryptography uses different keys for encryption and decryption. In this type of cryptography, an end user on a public or private network has a pair of keys: a public key for encryption and a private key for decryption. Here, a **private key** cannot be derived from the **public key.**

The asymmetric cryptography method has been proven to be secure against attackers. In asymmetric cryptography, the sender encodes the message with the help of a public key and the receiver decodes the message using a random key generated by the **sender's public key**.

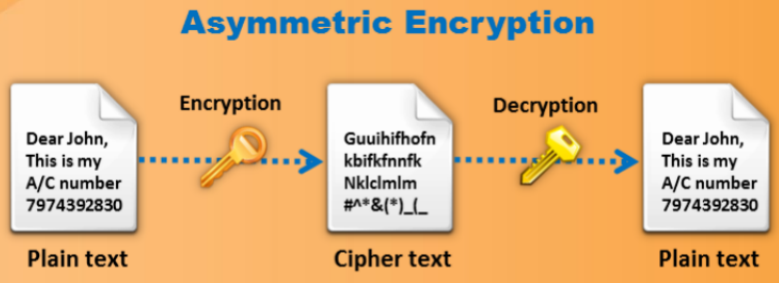
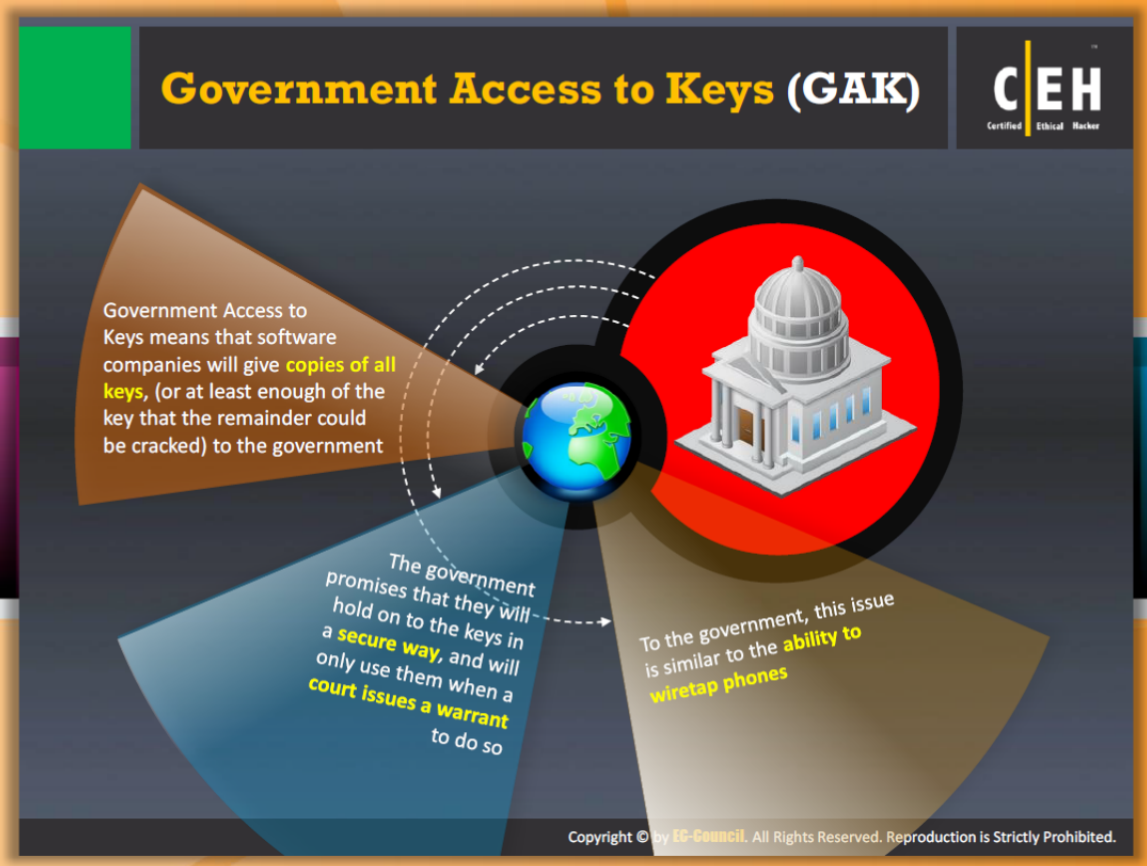



FIGURE 19.3: Asymmetric Encryption method




## Government Access to Keys (GAK)

A key escrow encryption system provides the decrypting capability to certain authorized personnel, under stipulated conditions, and can decrypt the data.

The **data recovery keys** for encrypting and decrypting the data are not similar, but they inform a method to determine the encryption and decryption keys. They include a key escrow (used to refer the safeguard the data keys), key archive, key backup, and data recovery system.

Key recovery systems have gained **prominence** due to the desire of government intelligence and law enforcement agencies to guarantee they have access to the encrypted information without the knowledge or consent of encryption users.

A well-designed cryptosystem provides security by recovering the encrypted data without proper information about the correct key. The maintenance of such **high-security** measures may cause problems to the owner of the encrypted data if the owner loses the key.

The eventual goal of government-driven recovery encryption, as stated in the US Department of Commerce's recent encryption regulations, "Envisions a worldwide key management infrastructure with the use of key escrow and key recovery encryption items."

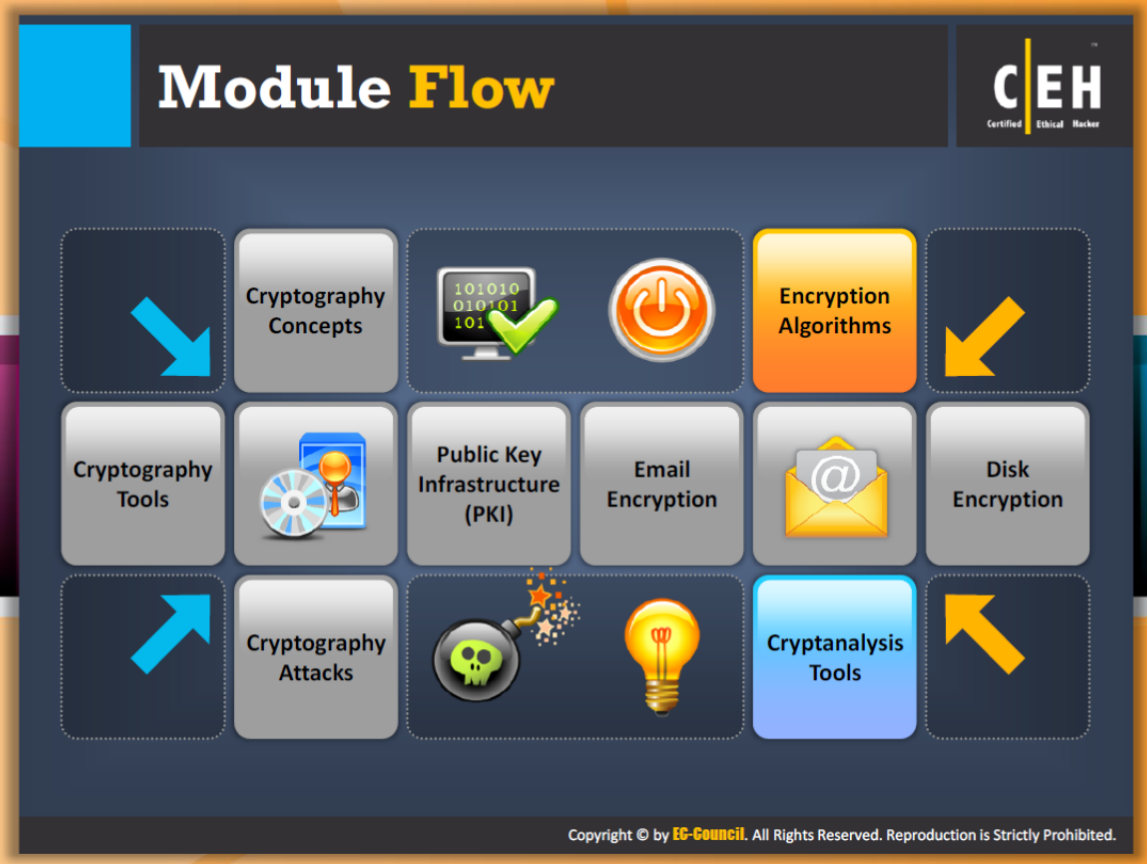
The Clipper Chip is a **hardware-based** cryptographic device used to secure private communications by simultaneously authorizing government agents to obtain the keys upon giving it, vaguely termed "**legal authorization**."

The keys are split between two government escrow agencies. This helps the government in accessing private communication channels. A device called **Clipper** is used to encrypt voice communications and a similar device called Capstone is used to encrypt the data.

The National Security Agency (NSA) is a secret US military intelligence agency responsible for capturing foreign government communications, and cracking the codes of protected transmissions that are developed with an algorithm known as **Skipjack**.

The Skipjack algorithm uses **80-bit keys**. Crypt analyzing requires searching through all keys, which makes it sixteen million times as hard to break as DES.

From the user's viewpoint, any key escrow system diminishes security. It puts the potential for access to the user's communications in the hands of **escrow** agencies, whose intentions, policies, security capabilities, and future cannot be known.

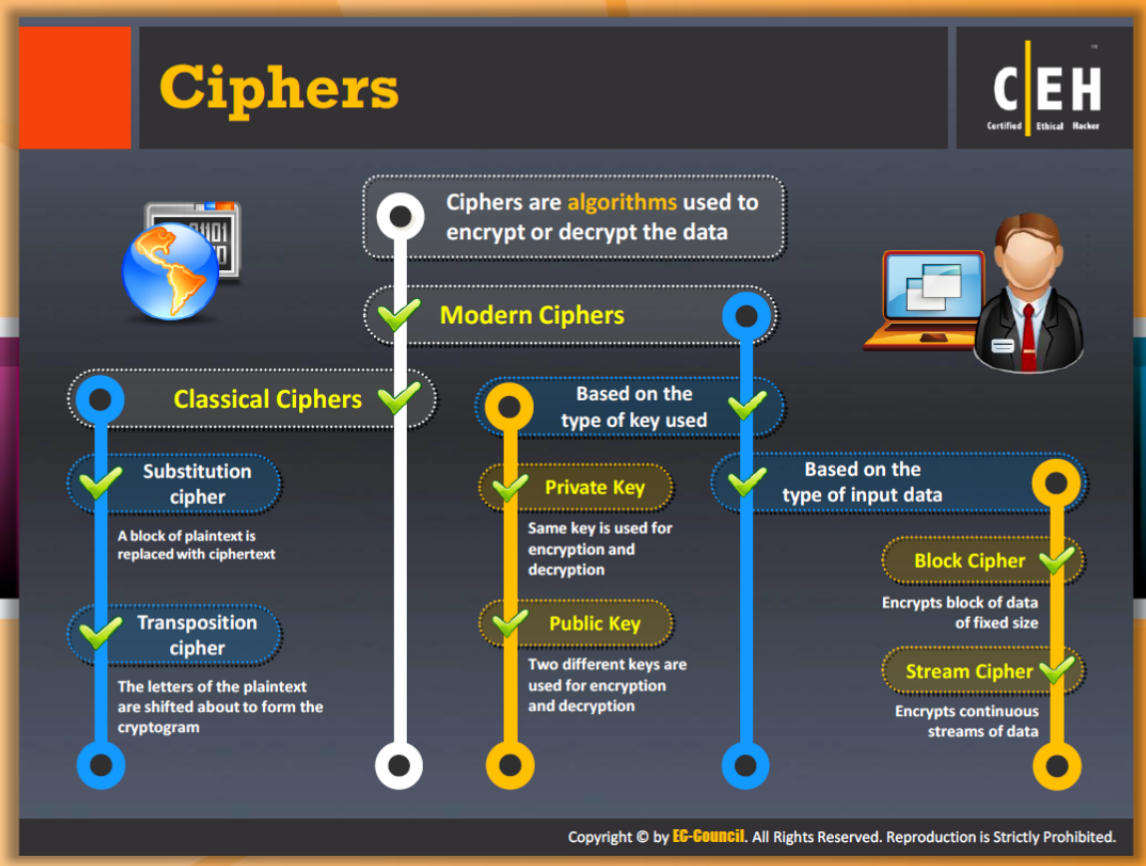



## Module Flow

So far, we have discussed cryptography and the concepts associated with it. Now we will discuss encryption key concepts of cryptography. There are many mechanisms, i.e, encryption algorithms, that allow you to encrypt the plaintext.

| | |
|---|---|
| **Cryptography Concepts** | **Encryption Algorithms** |
| **Cryptography Tools** | **Public Key Infrastructure (PKI)** |
| **Email Encryption** | **Disk Encryption** |
| **Cryptography Attacks** | **Cryptanalysis Tools** |

This section describes ciphers and various encryption algorithms such as **AES, DES, RC4, RC5, RC6, DSA, RSA, MD5, and SSH**.




## Ciphers

Cryptography refers to secret writing and a cipher is nothing more than an algorithm used for both **encryption** as well as **decryption**. The traditional method of encoding and decoding used to be in a different format, which provided numbering for each letter of the alphabet and used to encode the given message. If the attacker also knew the **numbering system**, he or she could decode it.

In cryptography, the cipher algorithm used for encoding is known as enciphering and decoding is known as deciphering.

**Example:**

a b c d e f g h...z are given in codes of numerical numbers, such as 1 2 3 4 5...26.

The message can be encoded based on this example and can be decoded as well. In a cipher, the message appears as plaintext but has been encoded through a key. Based on the requirements the key could be a symbol or some other form of text. If the message is highly confidential, then the key is restricted to the sender and recipient, but in some cases in open domains, some keys are shared without affecting the main data.

There are various types of ciphers:

## Classical Ciphers

Classical ciphers are the most basic type of ciphers that operate on **alphabet letters**, such as A-Z. These are usually implemented either by hand or with **simple mechanical devices**. These are not very reliable. There are two types of classical ciphers:

- **Substitution cipher**: The units of plaintext are replaced with ciphertext. It replaces bits, characters, or blocks of characters with different bits, characters, or blocks.
- **Transposition cipher**: The letters of the plaintext are shifted to form the **cryptogram**. The ciphertext is a permutation of the plaintext.

## Modern Ciphers

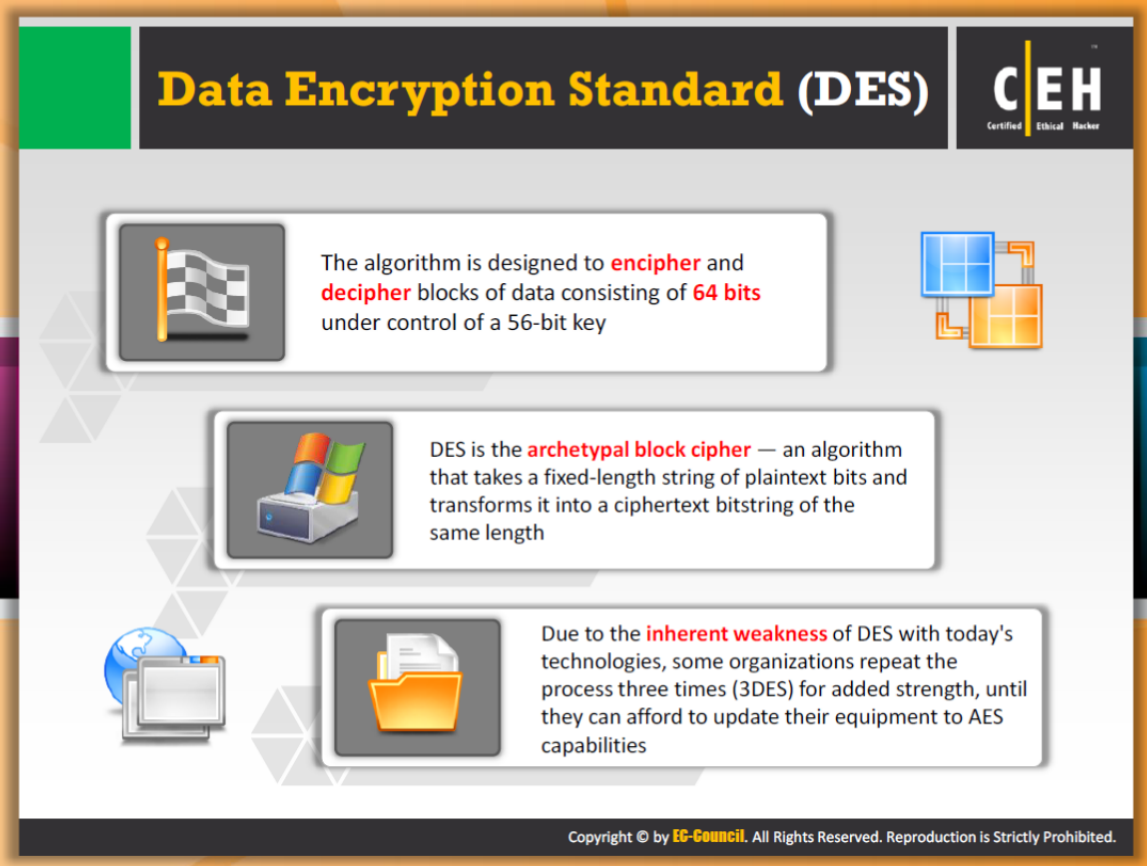
Modern ciphers are designed to withstand a **wide range of attacks**. Modern ciphers provide message secrecy, integrity, and authentication of the sender. The modern ciphers are calculated with the help of a one-way mathematical function that is capable of **factoring large prime numbers**. Modern ciphers are again classified into two categories based on the type of key and the input data. They are:

**Based on the type of key used**

- **Private-key cryptography** (symmetric key algorithm): The same key is used for encryption and decryption.
- **Public-key cryptography** (asymmetric key algorithm): Two different keys are used for encryption and decryption.

**Based on the type of input data**

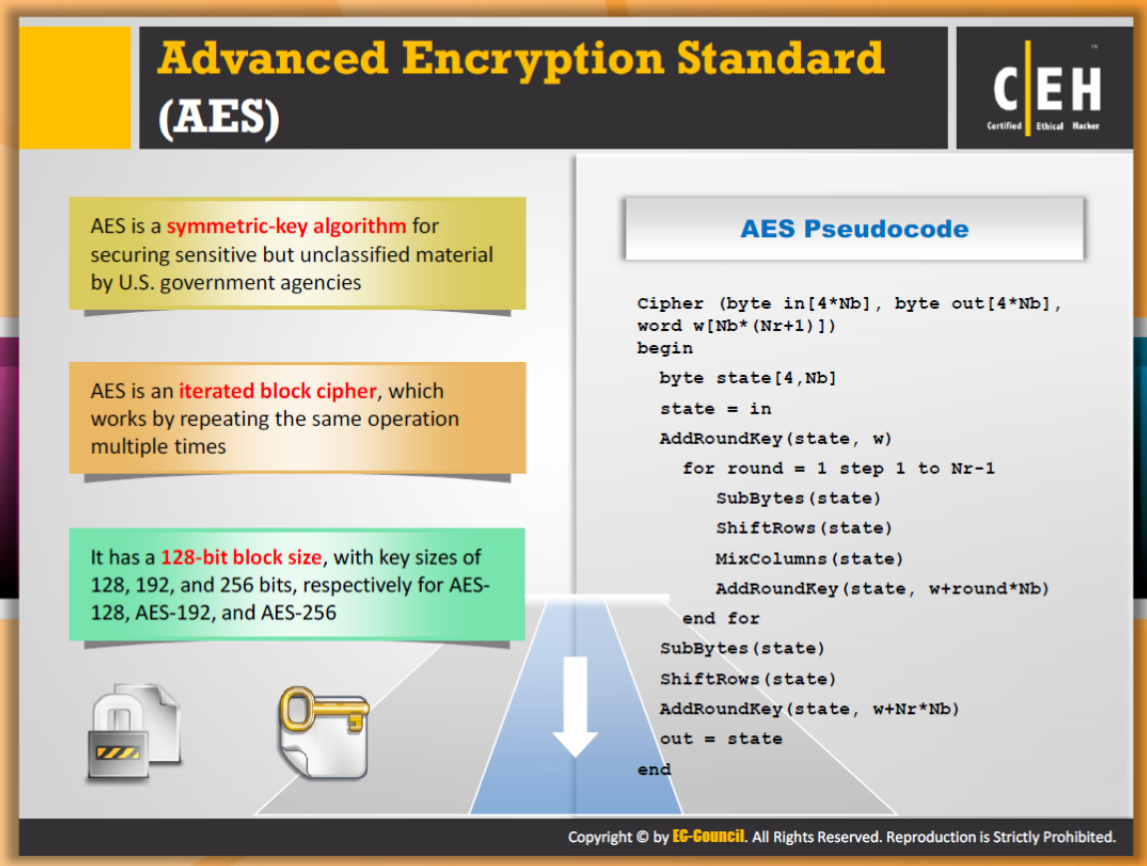
- **Block ciphers:** Refer to an algorithm operating on block (group of bits) of fixed size with an unvarying transformation specified by a symmetric key.
- **Stream ciphers:** Refer to symmetric key ciphers. This is obtained by combining the plaintext digits with a key stream (pseudorandom cipher digit stream).




## Data Encryption Standard (DES)

DES is the name of the Federal information Processing Standard (FIPS) 46-3 that describes the data encryption algorithm (DEA). It is a **symmetric cryptosystem designed** for **implementation** in hardware and used for single-user encryption, such as to store files on a hard disk in encrypted form.

**DES** gives **72 quadrillion** or more possible encryption keys and choses a random key for each message to be encrypted. Though DES is considered to be strong encryption, at present, triple DES is used by many organizations. **Triple DES** applies three keys successively.




# Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a **National Institute of Standards and Technology specification** for the encryption of electronic data. It can be used to encrypt digital information such as telecommunications, financial, and government data. AES consists of a symmetric-key algorithm, i.e., both encryption and decryption are performed using the same key.

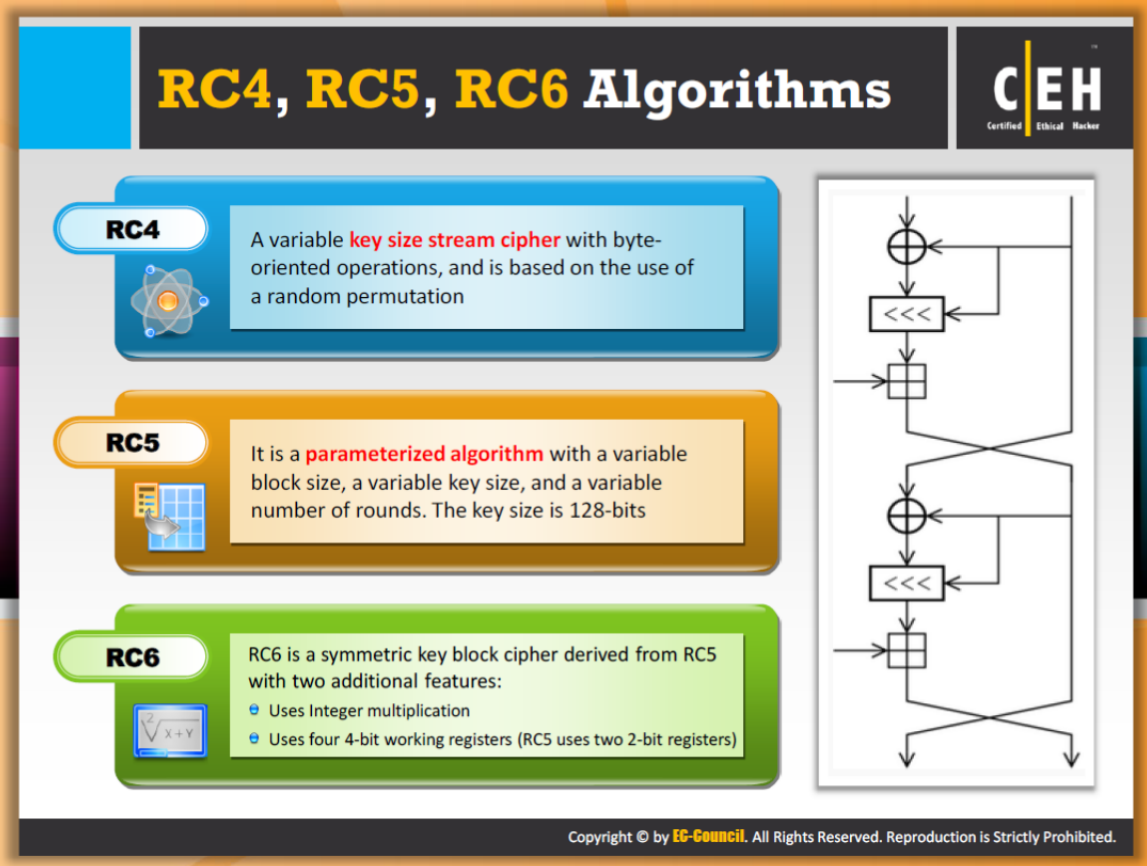
It is an iterated block cipher that works by repeating the defined steps multiple times. This has a 128-bit block size, with key sizes of 128, 192, and 256 bits, respectively, for AES-128, AES-192, and AES-256.

### AES Pseudo code

Initially, the cipher input is copied into the internal state and then an initial round key is added. The state is transformed by iterating a round function in a number of cycles. Based on the block size and key length, the number of cycles may vary. Once rounding is completed, the final state is copied into the **cipher output**. `Cipher (byte in [4*Nb], byte out [4*Nb], word w[Nb*(Nr+1)])`

```
begin
  byte state[4, Nb]
  state = in
```

```
  AddRoundKey (state, w)
    for round = 1 step 1 to Nr-1
       SubBytes(state)
       ShiftRows(state)
       MixColumns(state)
       AddRoundKey(state, w+round*Nb)
    end for
  SubBytes(state)
  ShiftRows(state)
  AddRoundKey(state, w+Nr*Nb)
  out = state
end
```




## RC4, RC5, and RC6 Algorithms

The **encryption algorithms** developed by RSA Security are:

### RC4

RC4 is a stream cipher for RSA Security, which Rivest designed. It is a variable key-size stream cipher with byte-oriented operations and is based on the use of a **random permutation**. According to some analysis, the period of the cipher is likely to be greater than 10100. For each output byte, eight to sixteen system operations are used, which means the cipher can run fast in software. Independent analysts have had a careful and critical look at the algorithm, and it is considered secure. Products like RSA **SecurPC** use this algorithm for file encryption. Rc4 is also used for safe communications like traffic encryption, which secures websites and from secure websites with **SSL protocol**.

### RC5

RC5 is a **block cipher** known for its **simplicity**. Ronald Rivest designed it. This algorithm has a variable block size and key size and a variable number of rounds. The choices for the block-size are 32 bits, 64 bits, and 128 bits. The iterations range from 0 to 255; whereas the key sizes have a range from 0 to 2040 bits. It has three routines: key expansion, encryption, and decryption.

### RC6

It is a block cipher that is based on RC5. Like in RC5, the block size, the key size, and the number of rounds are variable in the RC6 algorithm. The key-size ranges from 0 bits to 2040. In addition to RC5, RC6 has two more features, which are the addition of integer multiplication and the usage of four **4-bit** working registers as an alternative to RC5's two 2-bit registers.

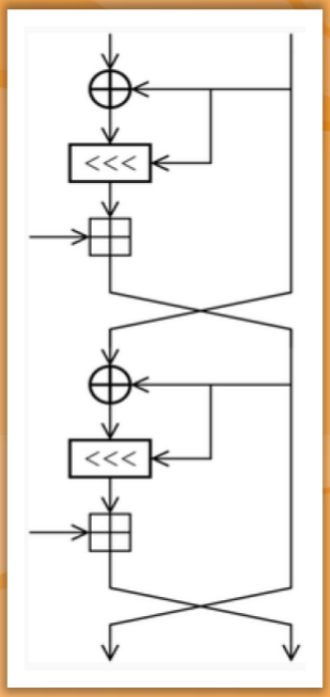
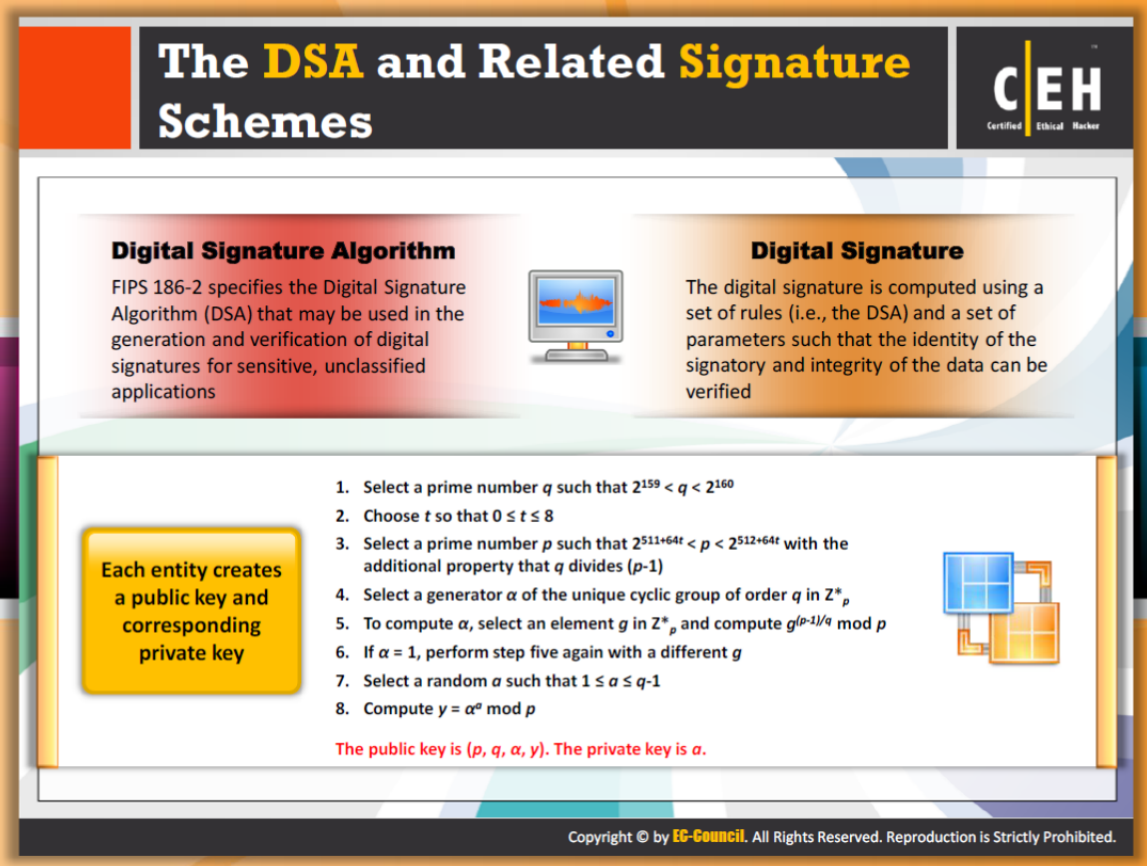


FIGURE 19.4: RC5 block cipher




## The DSA and Related Signature Schemes

A digital signature is a mathematical scheme used for the authentication of a digital message. Digital Signature Algorithm (DSA) is intended for its use in the U.S. Federal Information Processing Standard (FIPS 186) called the **Digital Signature Standard** (DSS). DSA was actually proposed by the National Institute of Standards and Technology (NIST) in August 1991. NIST made the U.S. Patent 5,231,668 that covers DSA available worldwide freely. It is the first digital signature scheme recognized by any government.

A digital signature algorithm includes a signature generation process and a **signature verification process.**

**Signature Generation Process:** The private key is used to know who has signed it.

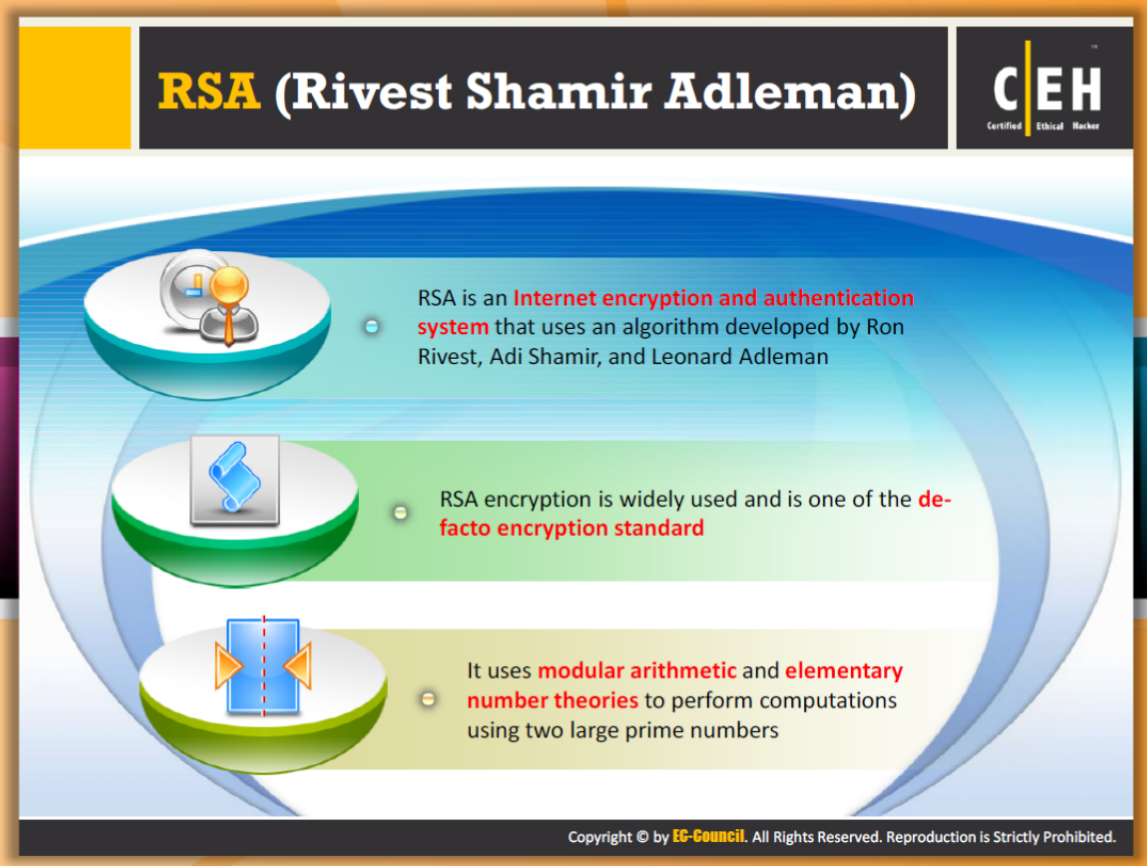
**Signature Verification Process:** The public key is used to verify whether the given digital signature is genuine or not.

As to the popularity of online shopping grows, e-payment systems and various other electronic payment modes rely on various systems like DSA.

**Benefits of DSA:**

- Less chances of forgery as it is in the case of **written signature**.
- Quick and easy method of business transactions.
- Fake currency problem can be drastically reduced.

DSA, with its uses and benefits, may bring revolutionary changes in the future.



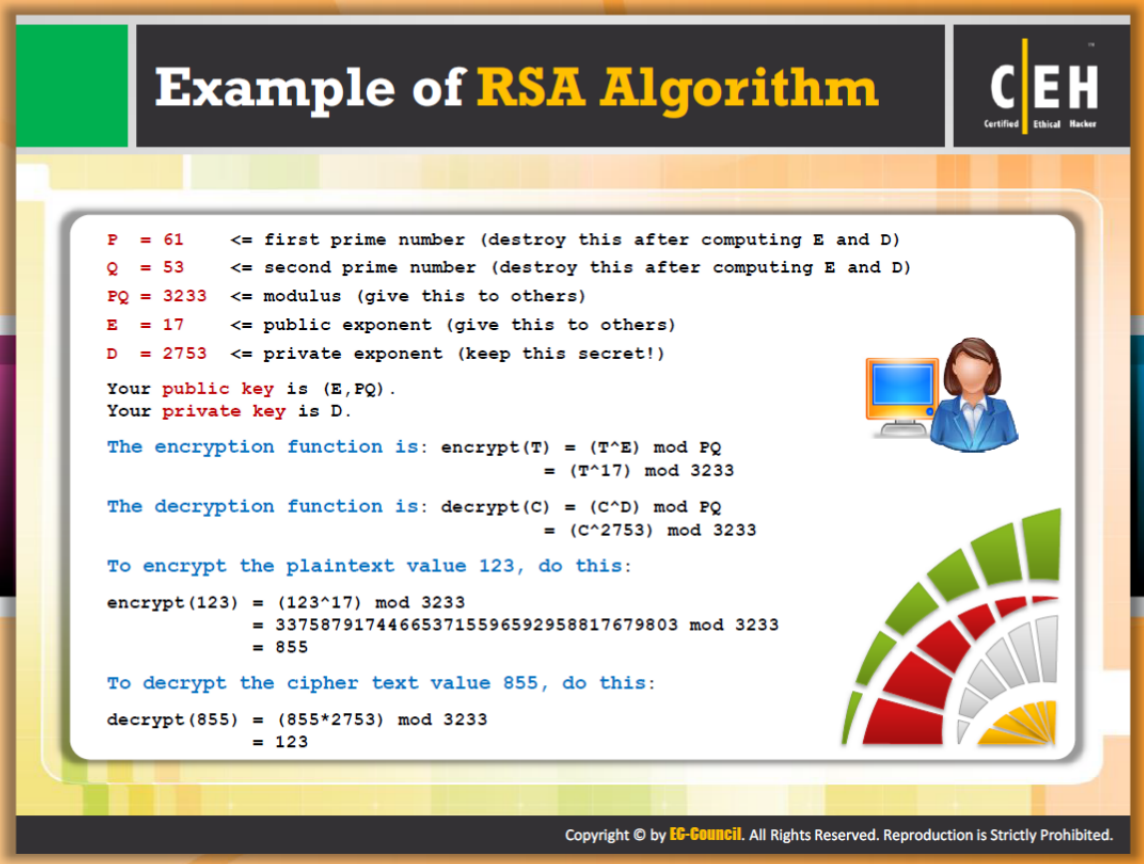

## RSA (Rivest Shamir Adleman)

RSA is a public-key cryptosystem. It uses modular arithmetic and elementary number theories to perform computations using two large prime numbers. **RSA encryption** is widely used and is the de-facto encryption standard.

Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public key cryptosystem for encryption and authentication. It is usually used with a secret key cryptosystem, like DES. The RSA system is widely used in a variety of products, platforms, and industries. Many operating systems like Microsoft, Apple, Sun, and Novell build the RSA algorithms into the existing versions. It can also be found on hardware secured telephones, on Ethernet network cards, and on smart cards. Consider that Alice uses the RSA technique to send Bob a message. If **Alice desires to communicate with Bob**, she encrypts the message using a randomly chosen DES key and sends it to Bob. Then she will look up Bob's public key and use it to encrypt the DES key. The RSA digital envelope, which is sent to Bob by Alice, consists of a DES-encrypted message and RSA-encrypted DES key. When Bob receives the digital envelope, he will decrypt the DES key with his private key, and then use the DES key to decrypt the message itself. This system combines the **high speed of DES with the key management convenience of the RSA system**.

The working of RSA is as follows: Two large prime numbers are taken (say "a" and "b"), and their product is determined (c = ab, where "c" is called the modulus). A number "e" is chosen such that it is less than "c" and relatively prime to (a-1)(b-1), which means that "e" and **(a-1)(b-**

**1)** have no common factors except 1. Apart from this, another number "f" is chosen such that (ef - 1) is divisible by (a-1)(b-1). The values "e" and "f" are called the public and private exponents, respectively. The public key is the pair (c, e); the private key is the pair (c, f). It is considered to be difficult to obtain the private key "f" from the public key (c, e). However, if someone can factor "c" into "a" and "b", then he or she can decipher the private key "f". The security of the RSA system is based on the assumption that such factoring is difficult to carry out, and therefore, the cryptographic technique is safe.




## Example of RSA Algorithm

RSA retains its security through the apparent difficulty in **factoring large composites**. Yet there is a possibility of discovering the polynomial time factoring algorithm using the advance number theory. There are three factors that can **aggravate** the path towards compromising RSA security. The advances include factoring technique, computing power, and decrease in the expenditure of the hardware. The working of RSA as explained before is illustrated in the following example. For P = 61 and Q = 53, PQ = 3233. Taking a public exponent, E = 17, and a private exponent, D = 2753, it can be encrypted into plain text 123 as shown as follows:

P = 61 <= first prime number (destroy this after computing E and D)

Q = 53 <= second prime number (destroy this after computing E and D)

PQ = 3233 <= modulus (give this to others)

E = 17 <= public exponent (give this to others)

D = 2753 <= private exponent (keep this secret!)

Your public key is (E,PQ).

Your private key is D.

The encryption function is: encrypt(T) = (T^E) mod PQ

= (T^17) mod 3233

The decryption function is: decrypt(C) = (C^D) mod PQ

= (C^2753) mod 3233

To encrypt the plaintext value 123, do this:

encrypt(123) = (123^17) mod 3233

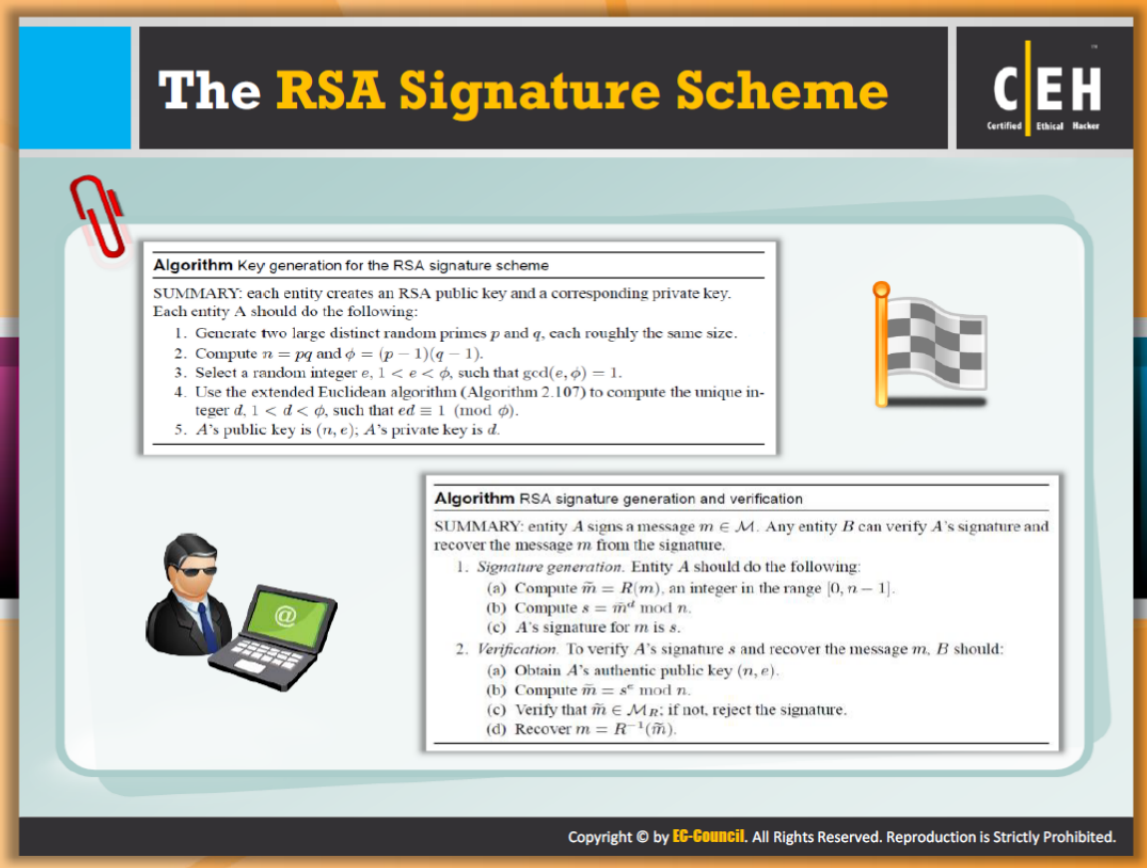
= 337587917446653715596592958817679803 mod 3233

= 855

To decrypt the cipher text value 855, do this:

decrypt(855) = (855*2753) mod 3233

= 123




## The RSA Signature Scheme

RSA is used for both public key encryption and for a digital signature (to sign a message). The RSA signature scheme is the first technique used to generate **digital signatures**. It is a deterministic digital signature scheme that provides message recovery from the signature itself. It is the most practical and versatile technique available.

RSA involves both a public key and a private key. The public key, as the name indicates, means any person can use it for **encrypting messages**. The messages that are encrypted with the public key can only be decrypted with the help of the private key.

Consider that John encrypts his document M using his private key $S_A$, thereby creating a signature $S_{john}(M)$. John sends M along with the signature $S_{john}(M)$ to Alice. Alice decrypts the document using Alice's public key, thereby verifying **John's signature**.

### RSA key generation

The procedure for RSA key generation is common for all the RSA-based signature schemes. To generate an RSA key pair, i.e., both an **RSA public key** and corresponding private key, each entity A should do the following:

- Select two large distinct primes' p and q arbitrarily, each of roughly the same bit length
- Compute n=pq and $\phi$=(p-1)(q-1)

- Choose a random integer e, $1<e<\phi$ such that get(e, $\phi$)=1
- Use the extended **Euclidean algorithm** in order to compute the unique integer d, $1<d<\phi$ such that $ed \equiv 1 \pmod{\phi}$
- The public key of A is (n, e) and private key is d

Destroy p and q at the end of the key generation

The RSA signature is generated and verified in the following way.

**Signature generation**

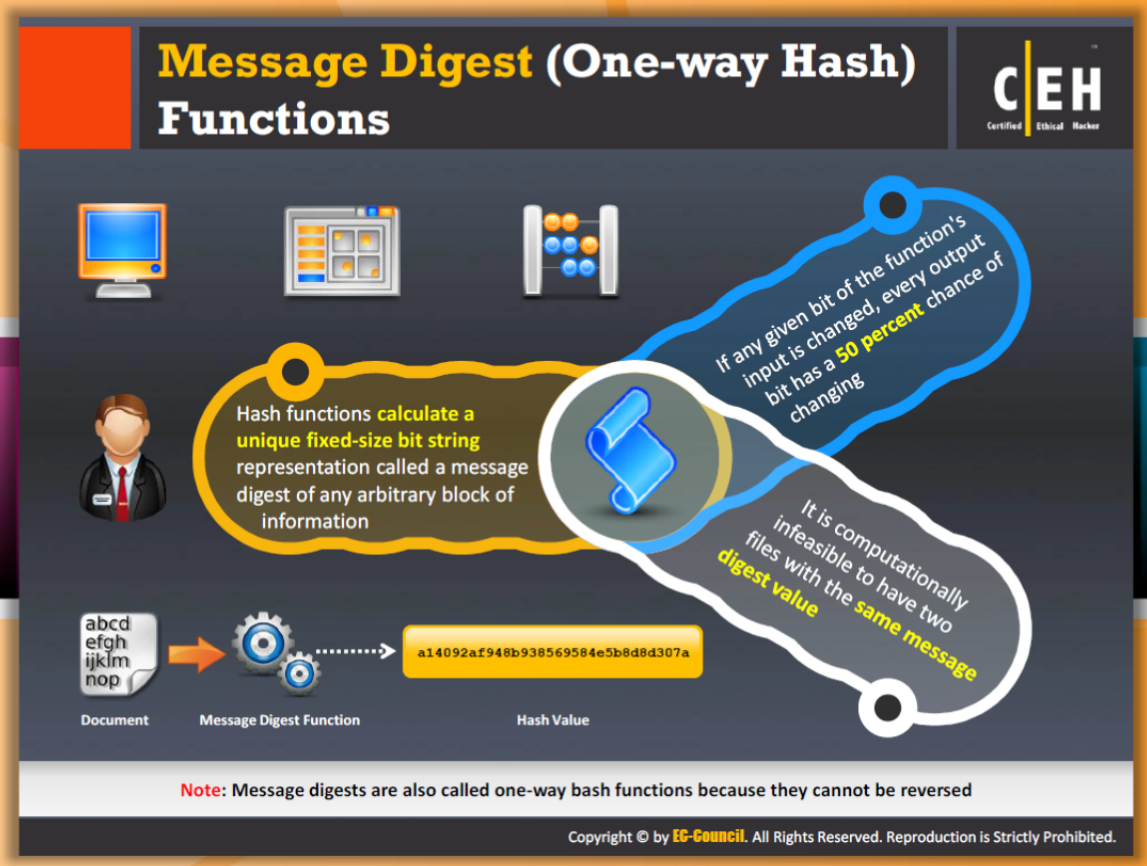
In order to sign a message m, A does the following:

- Compute m* = R(m) an integer in [0, n–1]
- Compute $s = m^{*d} \bmod n$
- A's signature for m is s

**Signature verification**

In order to verify A's signature s and recover message m, B should do the following:

- Obtain A's authentic public key (e, n)
- Compute $m^* = s^e \bmod n$
- Verify that m* is in $M_R$; if not, reject the signature
- Recover $m = R^{-1}(m^*)$




## Message Digest (One-way Hash) Functions

Message digest functions distill the information contained in a file (small or large) into a single large number, typically between **128- and 256-bits in length**. Message digest functions calculate a unique fixed-size bit string representation called hash value of any arbitrary block of information. The best message digest functions combine these mathematical properties. Every bit of the message digest function is influenced by every bit of the function's input. If any given bit of the function's input is changed, every output bit has a **50 percent** chance of changing. Given an input file and its corresponding message digest, it should be infeasible to find another file with the same message digest value.

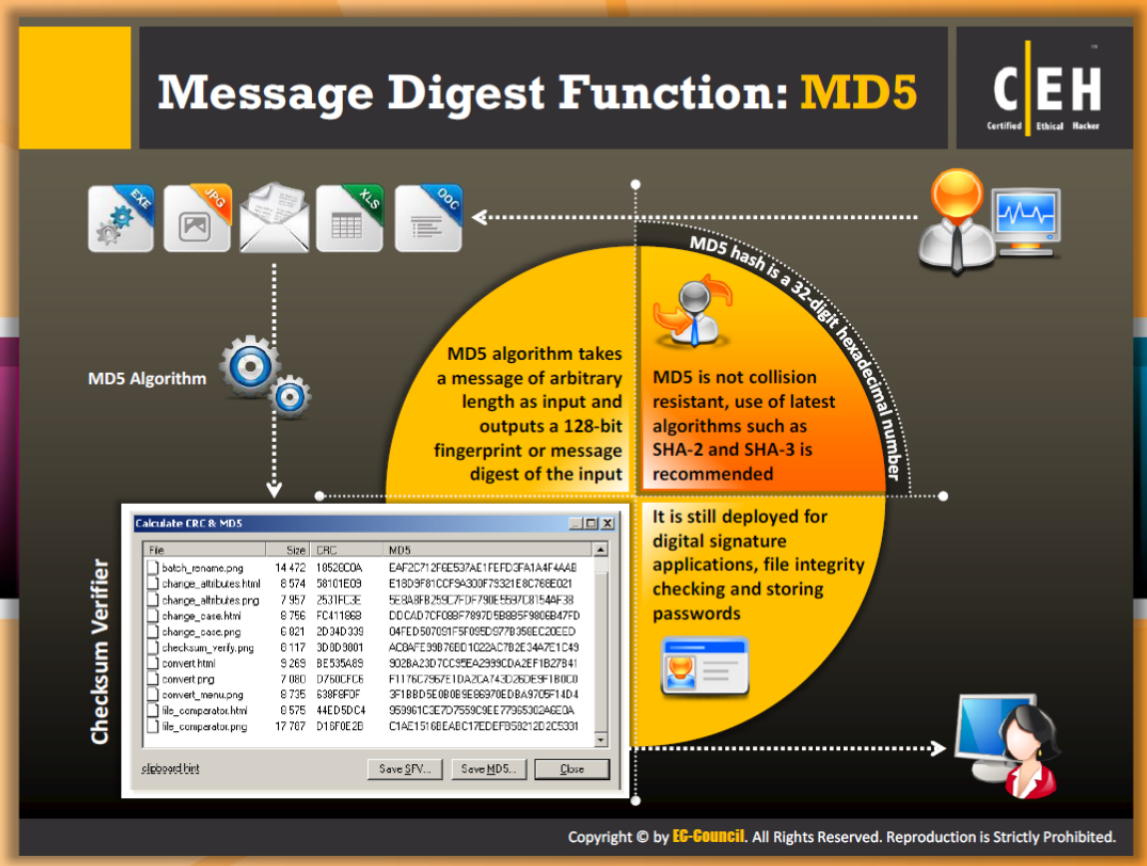
Message digests are also called one-way bash functions because they produce values that are difficult to invert, resistant to attack, mostly unique, and widely distributed.

**Message digest functions:**

- HMAC
- MD2
- MD4
- MD5
- SHA

- SHA-1


FIGURE 19.5: SHA1 a Message digest function




# Message Digest Function: MD5

H is a hash function that is a transformation that accepts a variable of any size as an input, m, and returns a string of a certain size. This is called the hash value h. i.e. h=H (m). The fundamental requirements for the **cryptographic hash** functions are:

- Input of any length
- Output of a fixed length

And H (x), can be easily computed for any value of x and it must be one-way (i.e., it cannot be inverted and it has an infeasible computation for the given input) and collision free. H is considered to be a weak collision free hash function if the given message x is **infeasible** to find a message y, so that H (x) =H (y). It is a collision free hash function if it is infeasible to find any two messages x and y such that H (x) =H (y).

The main role of a cryptographic hash function is to provide digital signatures. Hash functions are relatively faster than digital signature algorithms; hence, its characteristic feature is to calculate the signature of the document's hash value, which is smaller than the document. In addition, a digest can be used publicly without mentioning the contents of the document and the source of the document.

MD2, MD4, and MD5 algorithms that **Rivest** developed are **message-digest algorithms** that are used in digital signature applications, where the document is compressed securely before being

signed with the private key. The algorithms mentioned here can be of variable length but with the resultant message digest of 128-bit.

The structures of all three algorithms appear to be similar, though the design of MD2 is reasonably different from MD4 and MD5. MD2 was designed for the 8-bit machines, whereas the MD4 and MD5 were designed for the 32-bit machines. The message is added with extra bits to make sure that the length of the bits is divisible by 512. A **64-bit binary message** is added to the message.

Development of attacks on versions of MD4 has progressed rapidly and **Dobbertin** showed how collisions for the full version of MD4 could be found in under a minute on a typical PC. MD5 is relatively secure but is slower than MD4. This algorithm has four different rounds, which are designed with slight differences than that of MD4, but both the message-digest size and padding requirements remain the same.

## Brute Force of MD5

The effectiveness of the hash function can be defined by checking the output produced when an arbitrary input message is randomized. There are two types of **brute-force attacks** for one-way hash function: Normal brute force and birthday attack.

Examples of a few message digests are:

- `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`
- `echo "There is CHF1500 in the blue box." | md5sum`
  `86c524497a99824897ccf2cd74ede50f`

The same text always produces the same MD5 code.

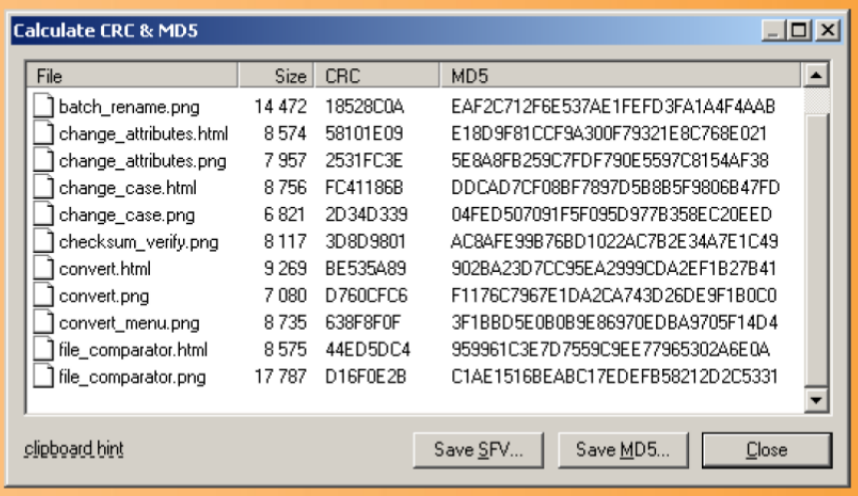
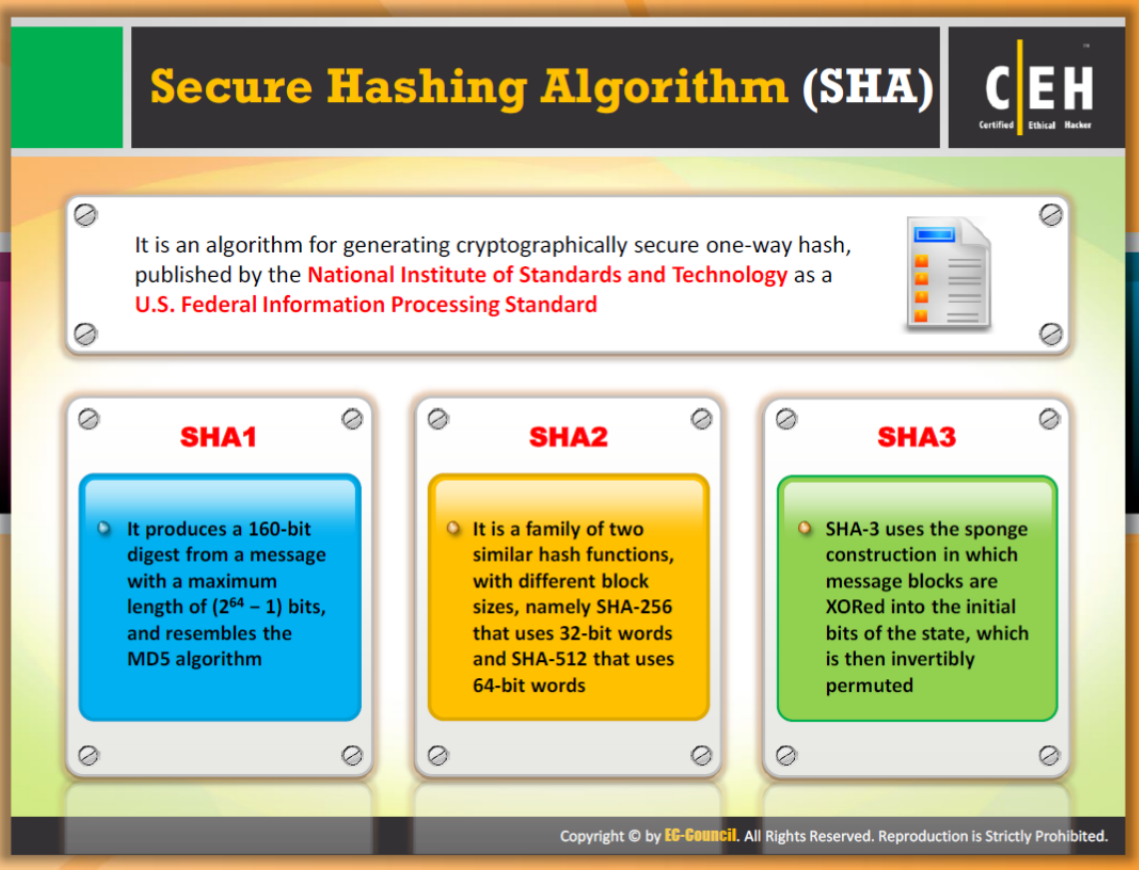



FIGURE 19.6: Checksum verifier




## Secure Hashing Algorithm (SHA)

The Secure Hash Algorithm (SHA), specified in the **Secure Hash Standard (SHS),** was developed by NIST, and published as a federal information-processing standard (FIPS PUB 180). It is an algorithm for generating a cryptographically secure one-way hash. SHA is part of the Capstone Project. Capstone is the U.S. government's long-term project to develop a set of standards for publicly available cryptography, as authorized by the Computer Security Act of 1987. The basic organizations that are responsible for Capstone are NIST and the NSA. SHA is similar to the MD4 message-digest algorithm family of hash functions, which was developed by Rivest.

The algorithm accepts a message of **264 bits** in length and a **160-bit** message output digest is produced, that is designed to complicate the searching of the text, which is similar to the given hash. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

The following are the cryptographic hash functions designed by the **National Security Agency (NSA):**

## SHA1

SHA1 produces a 160-bit digest from a message with a maximum length of (264 – 1) bits, and resembles the MD5 algorithm.

## SHA2

SHA2 is a family of two similar hash functions, with different block sizes, namely SHA-256 that uses 32-bit words and SHA-512 that uses 64-bit words.

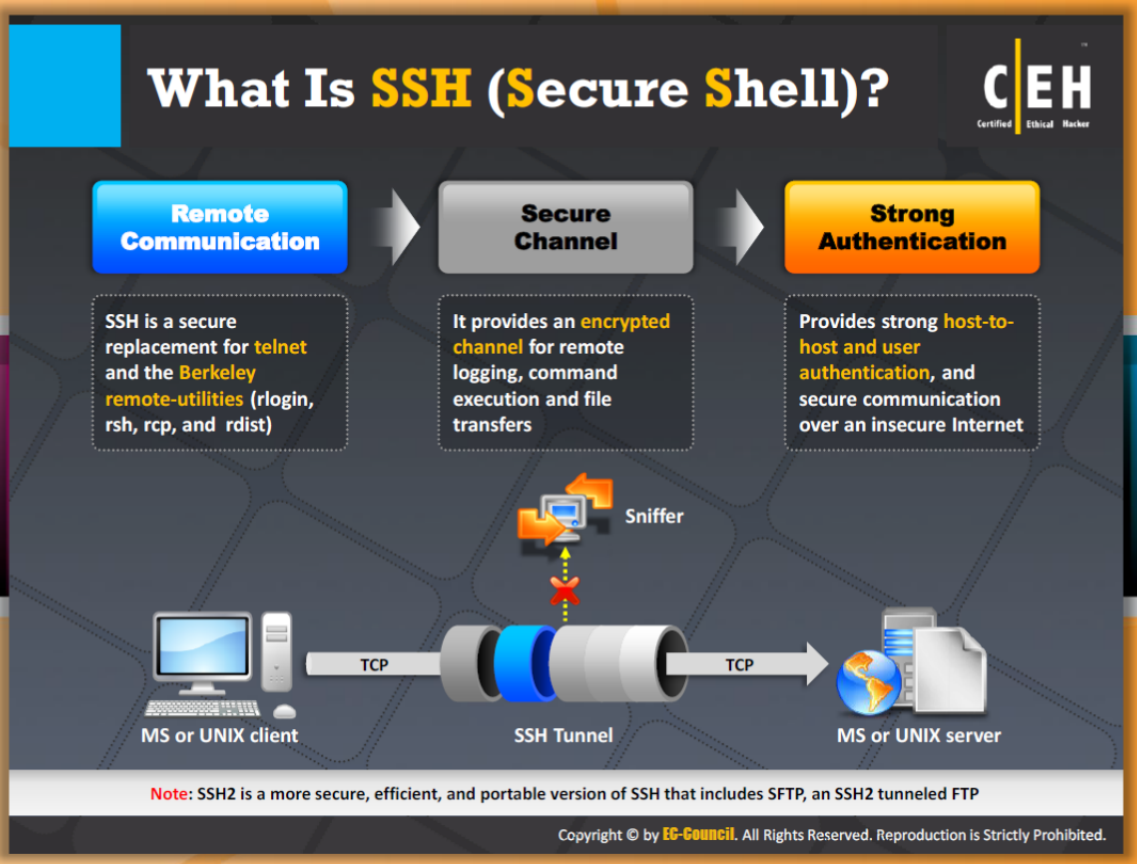
## SHA3

SHA3 is a future hash function standard still in development, chosen in a public review process from non-government designers.

Comparison of SHA functions (SHA0, SHA1 & SHA2)

| Algorithm and variant | | Output size (bits) | Internal hash sum (bits) | Size of block (bits) | Maximum size of message (bits) | Size of word (bits) | Rounds | Operations | Collision found |
|---|---|---|---|---|---|---|---|---|---|
| SHA-0 | | 160 | 160 | 512 | $2^{64}$-1 | 32 | 80 | +,and, or ,xor, rot | Yes |
| SHA-1 | | 160 | 160 | 512 | $2^{64}$-1 | 32 | 80 | +,and, or, xor, rot | Theoretical attacks ($2^{51}$) |
| SHA-2 | SHA-256/224 | 256/224 | 256 | 512 | $2^{64}$-1 | 32 | 64 | +, and, or, xor, shr,rot | None |
| | SHA-512/384 | 512/384 | 512 | 1024 | $2^{128}$-1 | 128 | 80 | +,and, or, xor, shr, rot | None |

TABLE 19.1: Comparison between SHA-0, SHA-1 & SHA-2 functions




## What Is SSH (Secure Shell)?

Secure Shell is a program that is used to log onto another computer over the network, to transfer files from one computer to another. It offers good authentication and a secure communication channel over insecure media. It might be used as a replacement for telnet, login, rsh, and rcp. In SSH2, **sftp** is a replacement for **ftp**. In addition, SSH offers secure connections and secure transferring of TCP connections. SSH1 and SSH2 are completely different protocols. SSH1 encrypts the user's server and hosts keys to authenticate where SSH2 only uses host keys, which are different packets of keys. SSH2 is more secure than SSH1. It should be noted that the **SSH1** and **SSH2** protocols are in fact different and not compatible with each other. SSH2 is more secure and has an improved performance than SSH1 and is also more portable than SSH1.

The SSH1 protocol is not being developed anymore, as SSH2 is the standard. Some of the main features of SSH1 are as follows:

- SSH1 is more vulnerable to attacks due to the presence of structural weaknesses
- It is an issue of the man-in-the-middle attack
- It is supported by many platforms
- It supports hosts authentication

- It supports varied authentication
- Performance of SSH2 is better than SSH1

**SSH communications security** maintains SSH1 and SSH2 protocols.

It authenticates with the help of one or more of the following:

- Password (the /etc/passwd or /etc/shadow in UNIX)
- User public-key (RSA or DSA, depending on the release)
- Kerberos (for SSH1)
- Host-based (.rhosts or /etc/hosts. equiv in SSH1 or public key in SSH2)

**Secure Shell protects against**:

- A remote host sending out packets that pretend to come from another trusted host (IP spoofing). SSH protects against a spoofer on the local network, who can pretend to be the user's router to the outside.
- A host pretending that an IP packet comes from another **trusted host (IP source routing)**.
- An attacker forging domain name server records (DNS spoofing).
- Capturing of passwords and other data by the intermediate hosts.
- Exploitation of data by the people who control the intermediate hosts.
- Attacking by listening to X authentication data and spoofing connections to the X11 server.

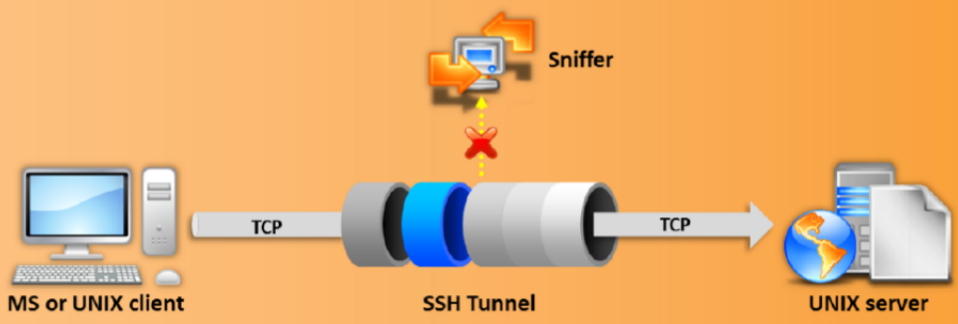
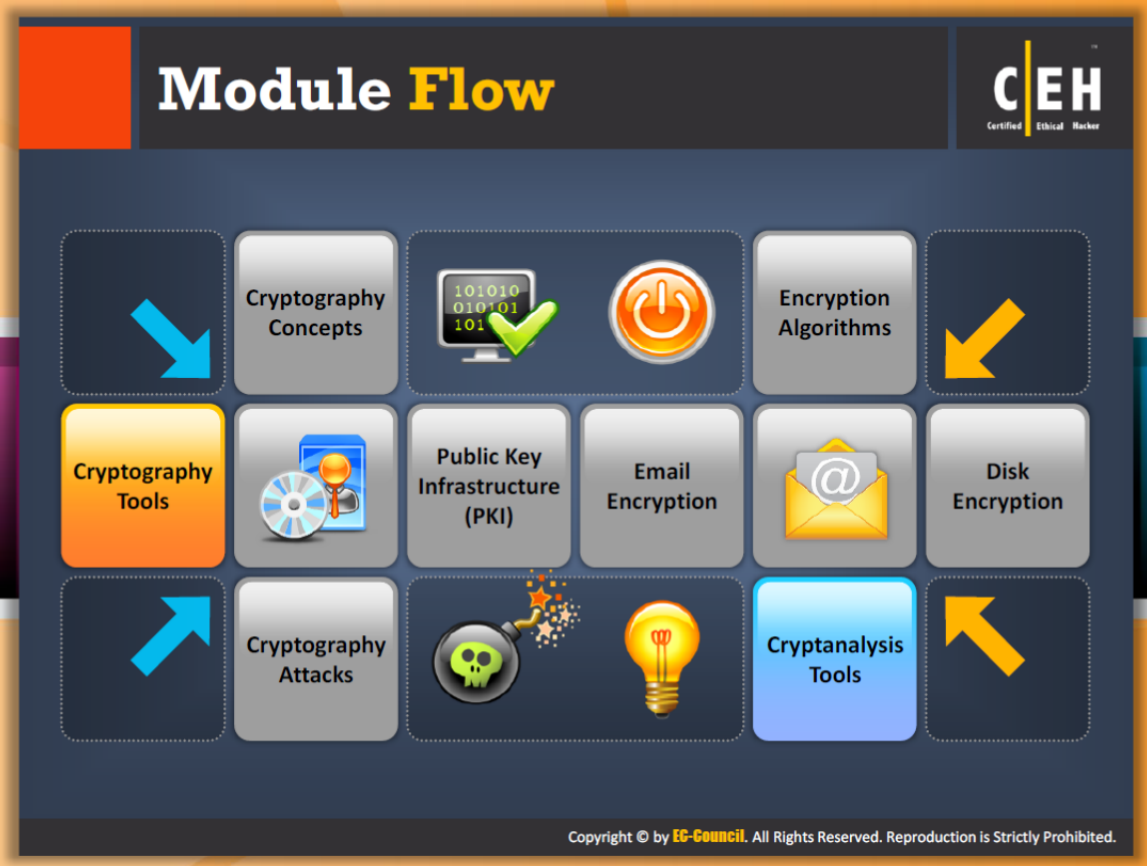



FIGURE 19.7: Secure shell tunneling

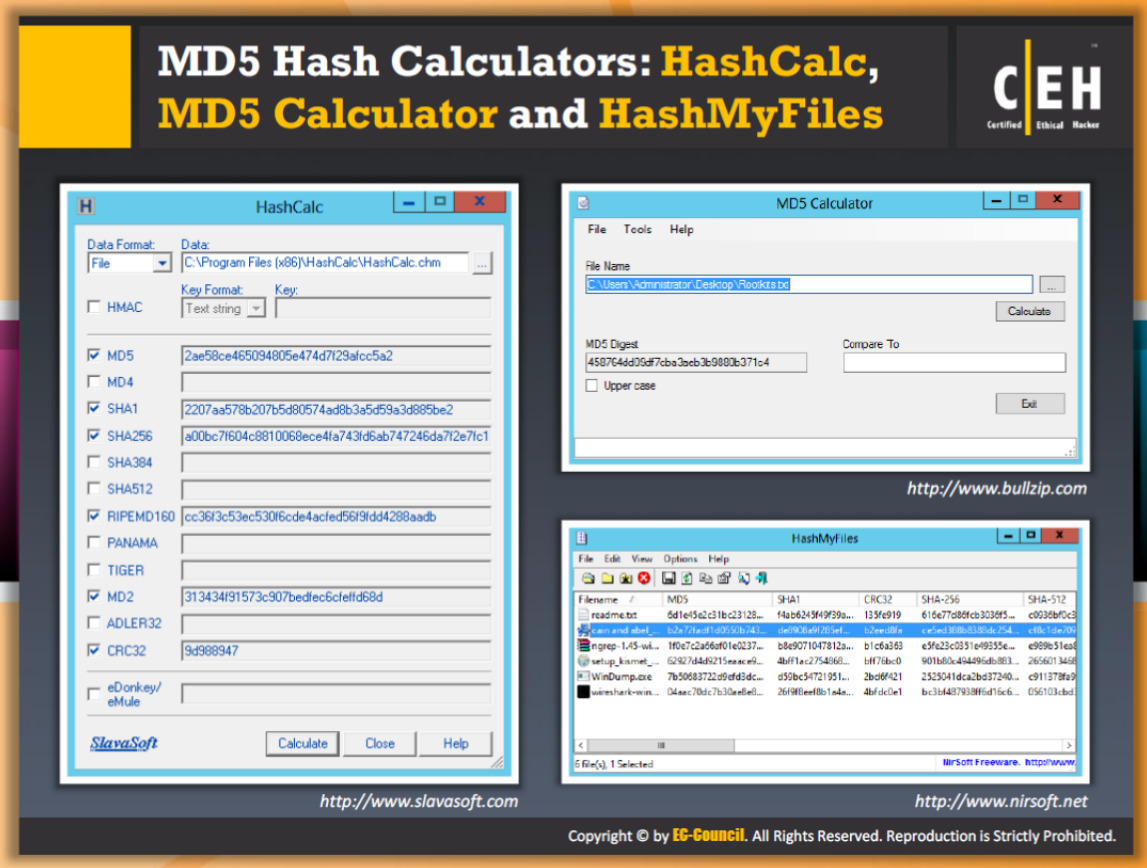



## Module Flow

So far, we have discussed cryptography concepts and various encryption algorithms. Now it is time to discuss how cryptography is usually performed. There are many cryptographic tools readily available in the market that can help you to secure your data..

| | |
|---|---|
| Cryptography Concepts | Encryption Algorithms |
| Cryptography Tools | Public Key Infrastructure (PKI) |
| Email Encryption | Disk Encryption |
| Cryptography Attacks | Cryptanalysis Tools |

This section lists and describes various cryptographic tools.




# MD5 Hash Calculators: HashCalc, MD5 Calculator, and HashMyFiles

Hashing is one form of cryptography in which a message digest function is used to convert plainttext into its equivalent hash value. This message digest function uses different hash algorithms to **convert plaintext** into hash values. Many MD5 hash calculators are readily available in the market. Examples of **MD5** hash calculators include:

## HashCalc

Source: http://www.slavasoft.com

The HashCalc utility allows you to compute message digests, checksums, and HMACs for files, as well as for text and hex strings. It allows you to calculate hash values using different types of hashing algorithms such **as MD2, MD4, MD5, SHA-1, SHA-2 (256, 384, 512), RIPEMD-160, PANAMA, TIGER, ADLER32, and CRC32**. You just need to select the file and hash algorithm for calculating the hash value of a particular file.

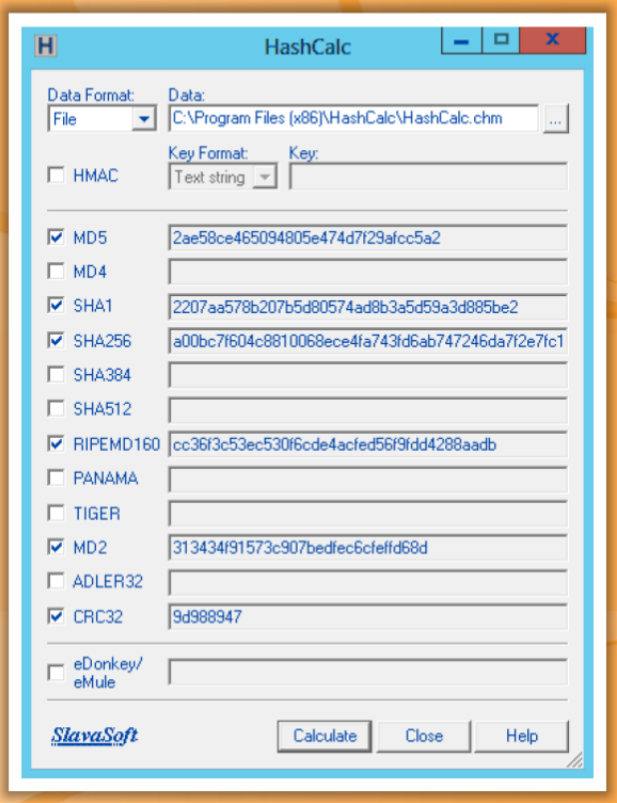



FIGURE 19.8: HashCalc screenshot

## MD5 Calculator

Source: http://www.bullzip.com

MD5 Calculator allows you to calculate the MD5 hash value of the selected file. The **MD5 Digest field** of the utility contains the calculated hash value. You just need to select a file of which the hash value needs to be calculated. You can also compare two hash values with this **tool**.

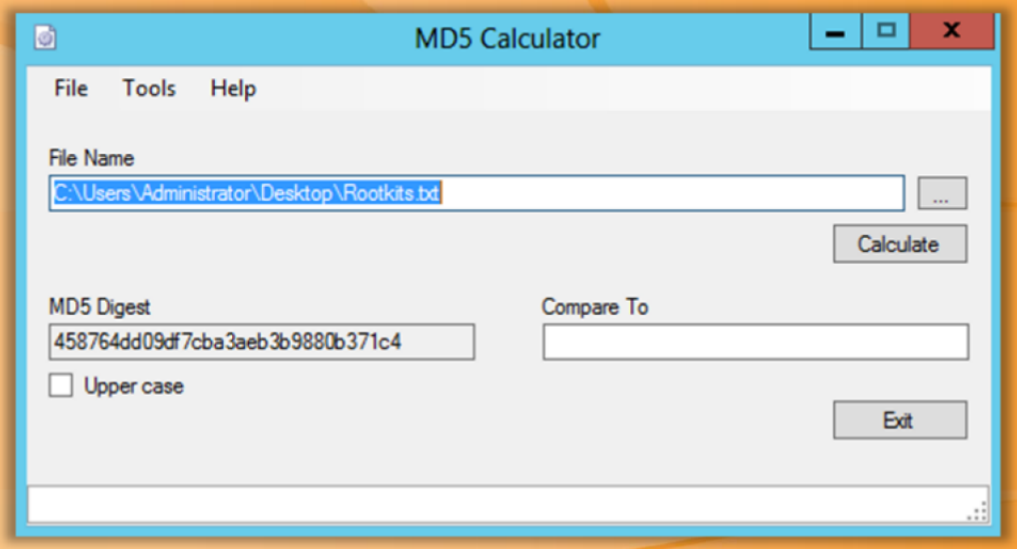



FIGURE 19.9: MD5 Calculator calculating MD5 hash value

## HashMyFiles

Source: http://www.nirsoft.net

The HashMyFiles utility allows you to calculate the MD5 and SHA1 hashes of one or more files. You can copy the **MD5/SHA1** hashes list into the clipboard, or save it into a text/html/xml file. It can also be launched from the context menu of Windows Explorer, and display the MD5/SHA1 hashes of the selected file or folder.

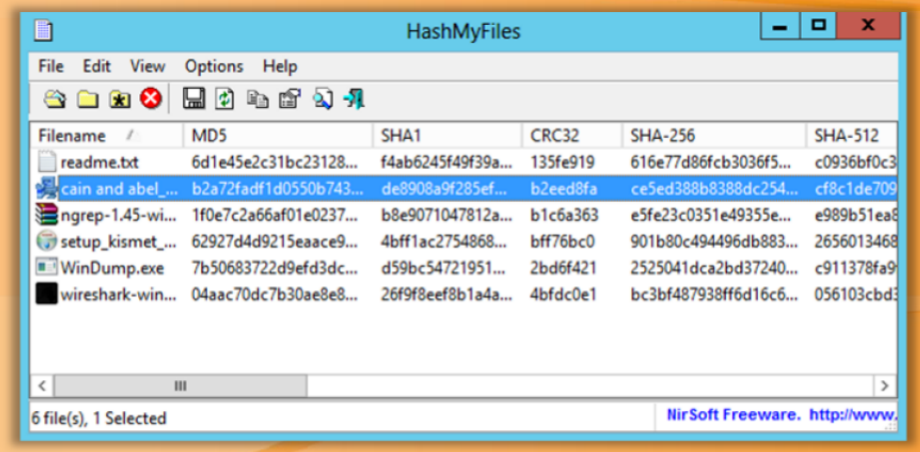
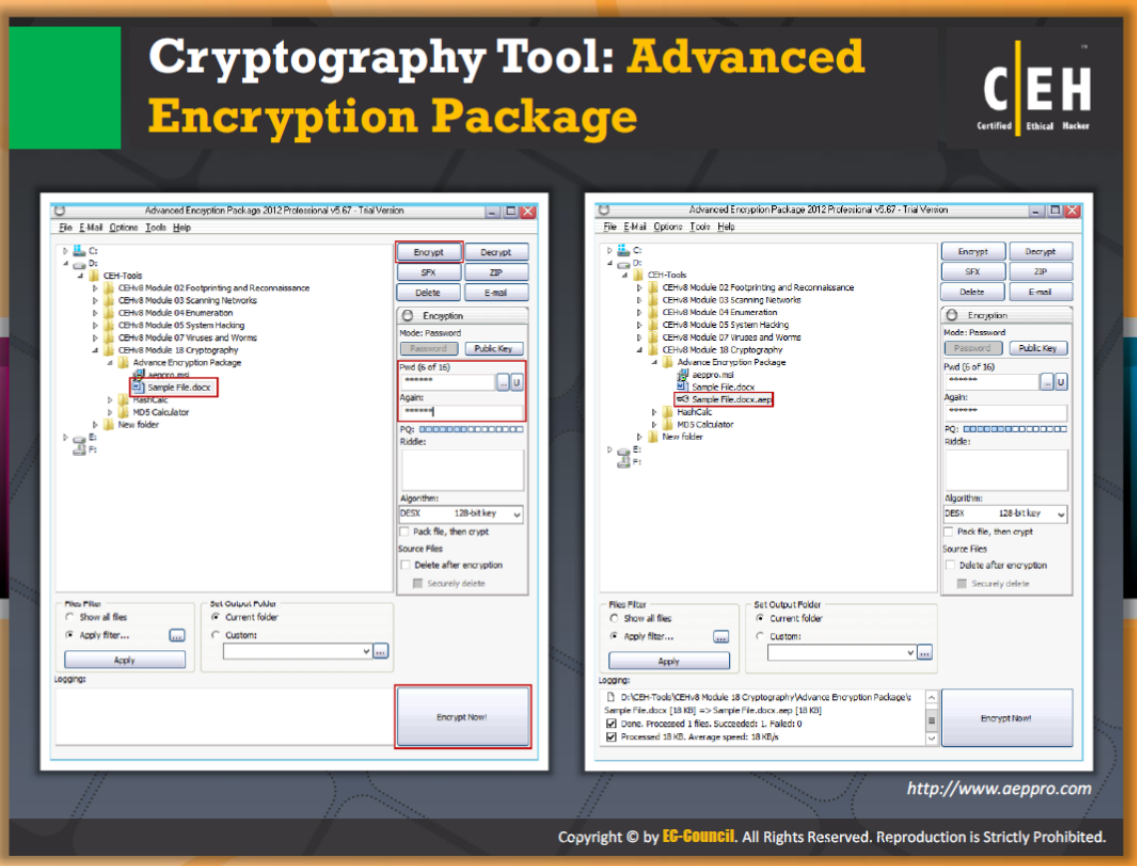



FIGURE 19.10: HashMyFiles screenshot




# Cryptography Tool: Advanced Encryption Package

Source: http://www.aeppro.com

Advanced Encryption Package is file encryption software that helps you maintain the privacy of your information by allowing you to password-protect files. It is able to perform encryption, decryption, and self-decrypting file creation, file Delete/Wipe, Zip management, encryption key management, and file emailing.

**Its feature includes:**

- **Strong and proven algorithms** are used to protect your sensitive documents
- It can encrypt files as well as text
- Performs secure file deletion
- Ability to create **encrypted self-extracting file** to send it as **email** attachment

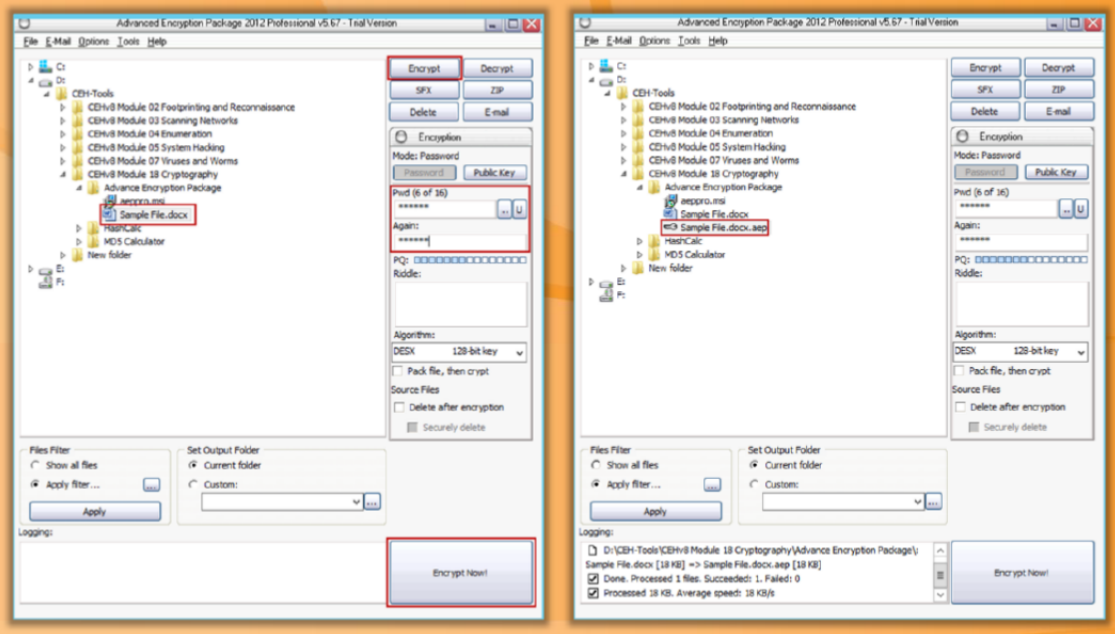
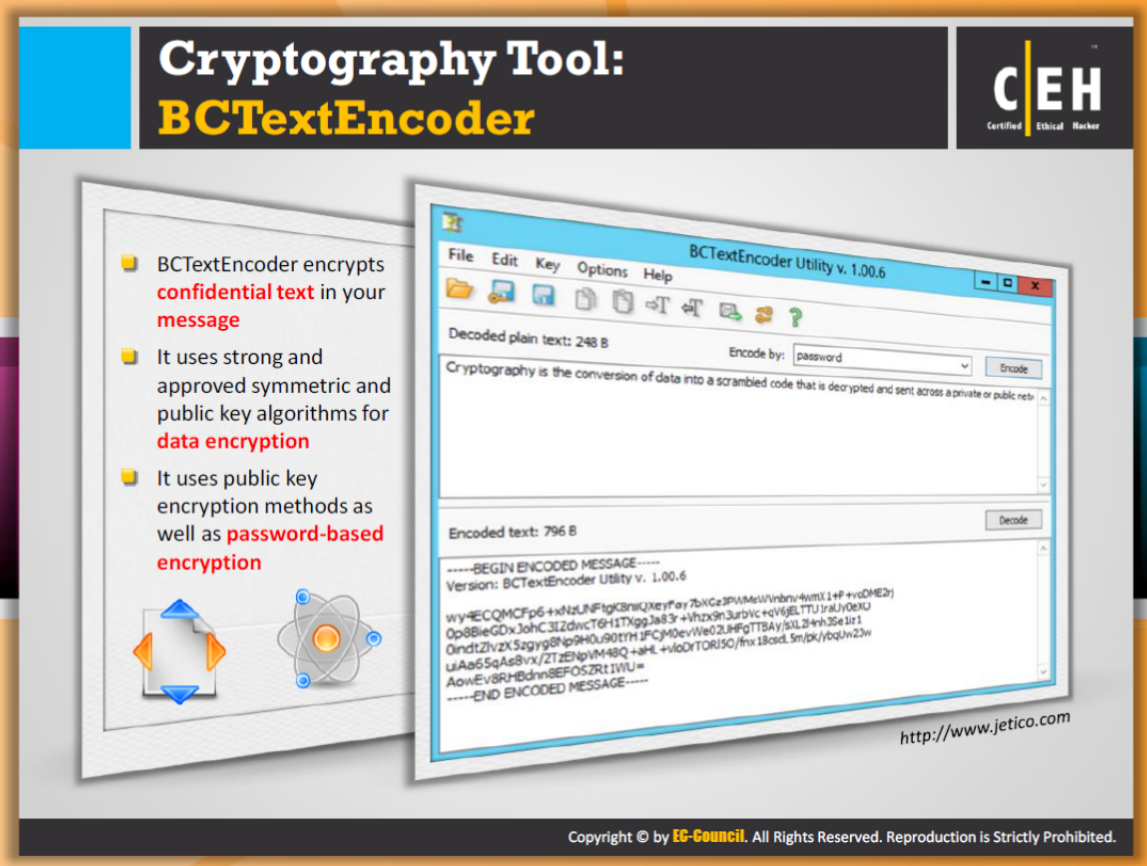



FIGURE 19.11: Advanced Encryption Package protecting files using passwords




# Cryptography Tool: BCTextEncoder

Source: http://www.jetico.com

BCTextEncoder allows you to **encrypt** and **decrypt** the **confidential messages** for secure email or chat communications. It uses public key encryption methods as well as password-based encryption and strong and approved symmetric and public key algorithms for **data encryption**. You simply need to choose the text you want to encrypt and specify the password and then click the button to encode it.

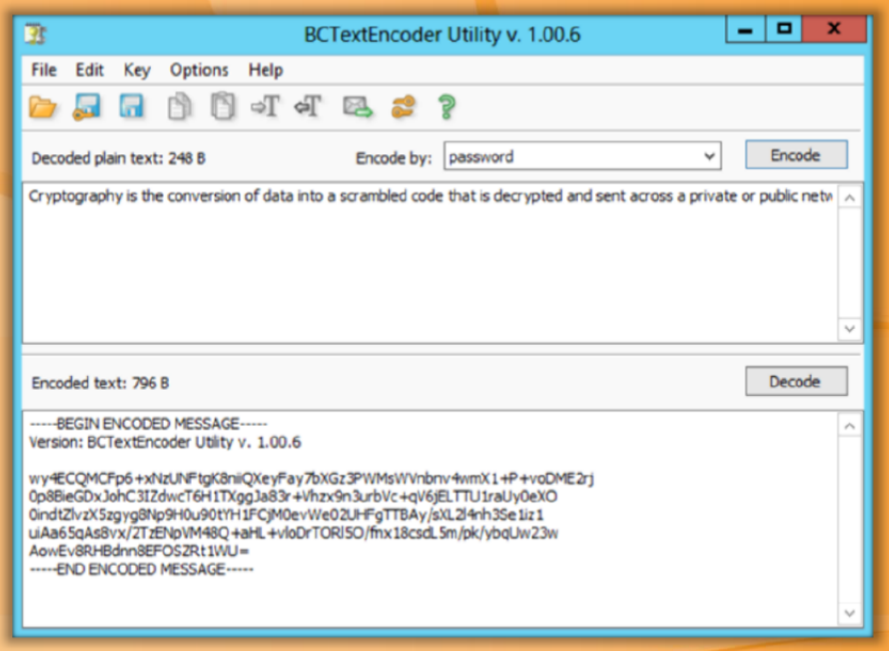
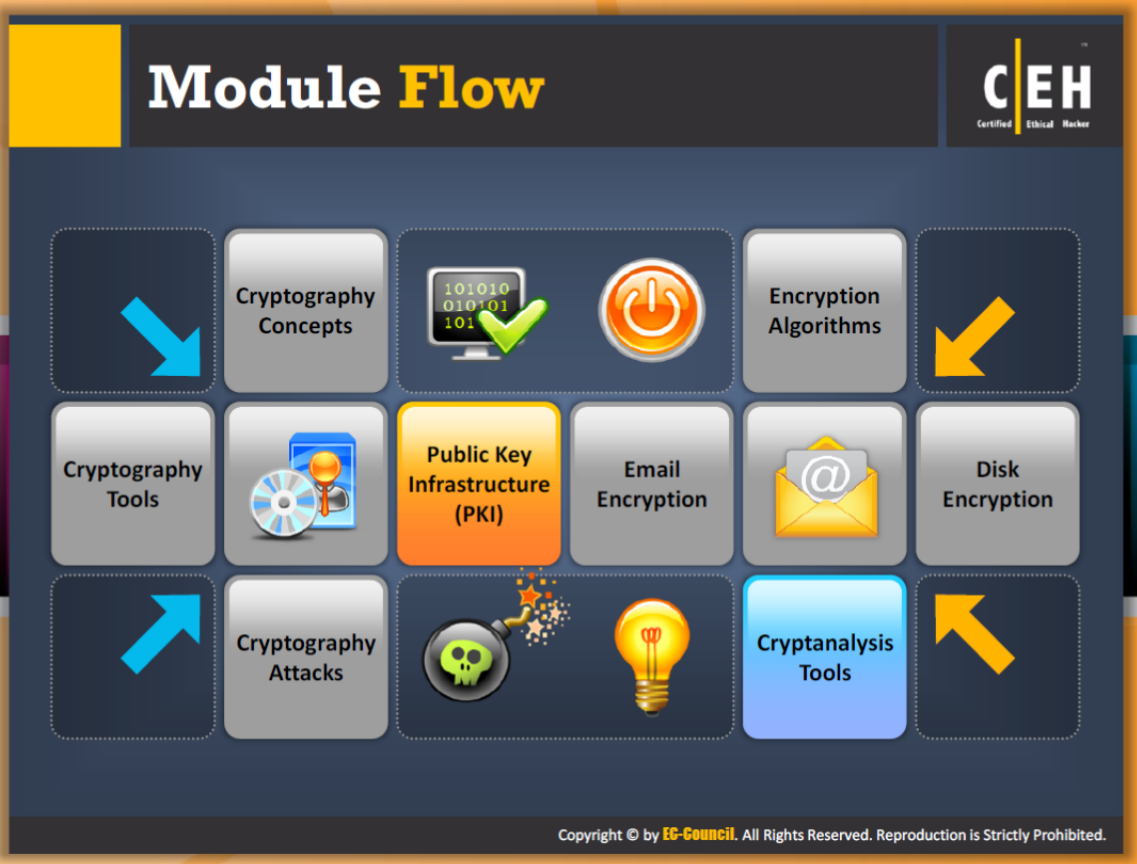



FIGURE 19.12: BCTextEncoder encrypting and decrypting confidential messages


## Cryptography Tools

There are various cryptographic tools that you can use for encrypting and decrypting your information, files, etc. These tools implement different types of available **encryption algorithms**:

- CommuniCrypt File Encryption Tools available at http://www.communicrypt.com
- Steganos LockNote available at https://www.steganos.com
- AxCrypt available at http://www.axantum.com
- AutoKrypt available at http://www.hiteksoftware.com
- CryptoForge available at http://www.cryptoforge.com
- NCrypt XL available at http://www.littlelite.net
- Ccrypt available at http://ccrypt.sourceforge.net
- WinAES available at http://fatlyz.com
- EncryptOnClick available at http://www.2brightsparks.com
- GNU Privacy Guard available at http://www.gnupg.org

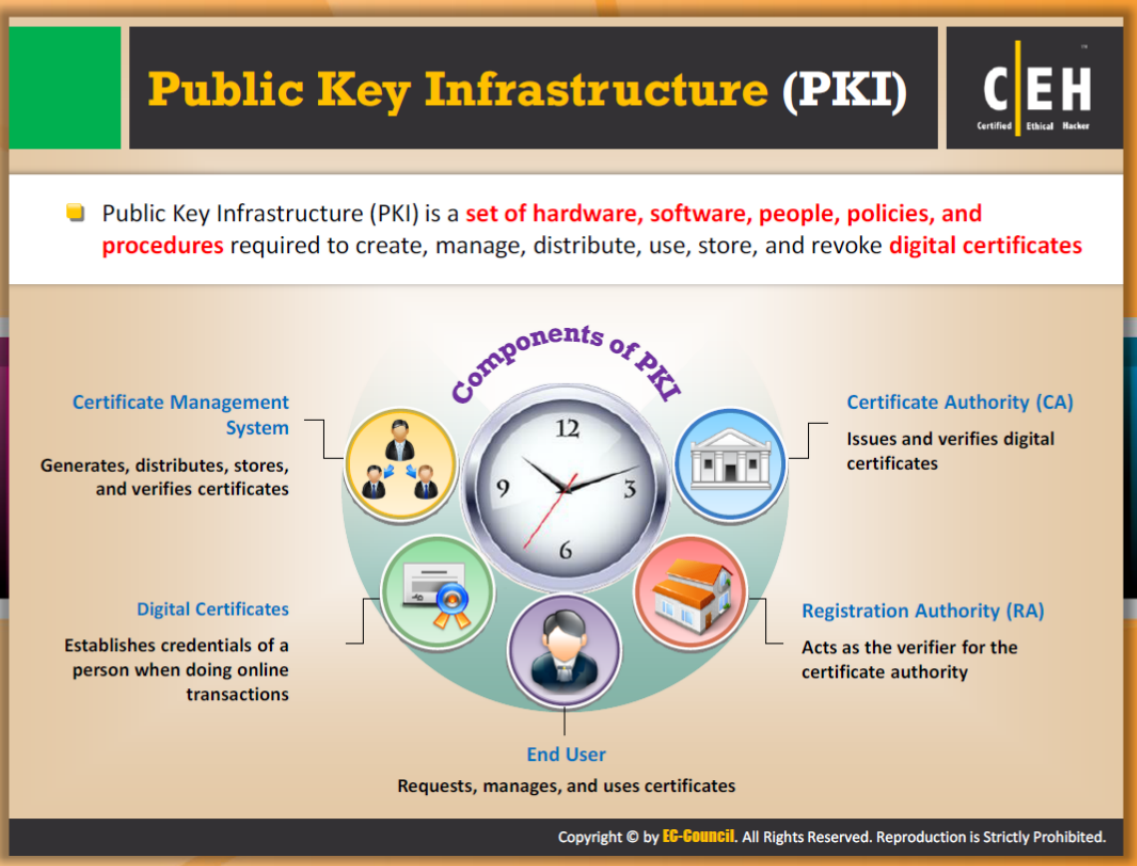



## Module Flow

So far, we have discussed cryptography, various **encryption algorithms**, and the use of encryption algorithms in cryptography. In addition to the cryptographic security mechanisms discussed so far, there is one more infrastructure intended to **exchange data** and **money** over the Internet securely: PKI (Public Key Infrastructure).

| | |
|---|---|
| **Cryptography Concepts** | **Encryption Algorithms** |
| **Cryptography Tools** | **Public Key Infrastructure (PKI)** |
| **Email Encryption** | **Disk Encryption** |
| **Cryptography Attacks** | **Cryptanalysis Tools** |

This section provides information about Public Key Infrastructure (PKI) and the role of each components of **PKI** in the **security public** key encryption. Let's start with what is Public Key Infrastructure (PKI)?

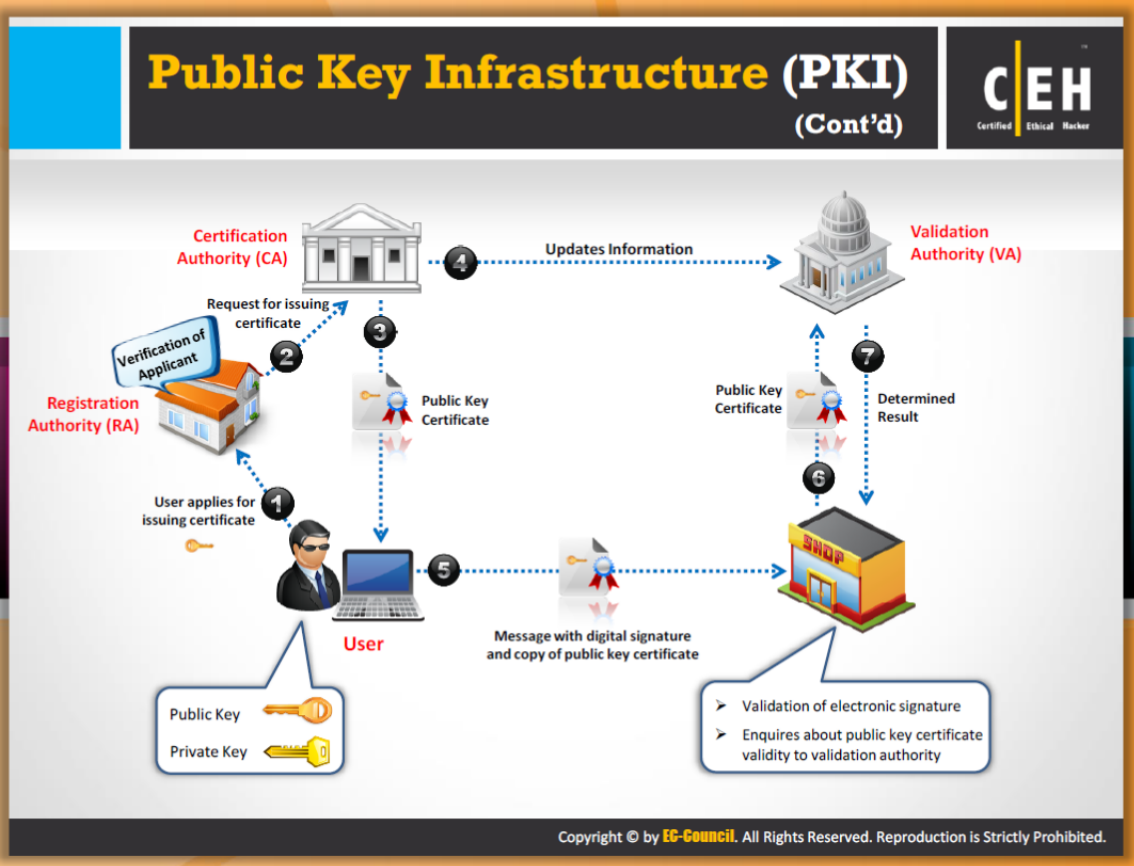



# Public Key Infrastructure (PKI)

Public Key Infrastructure (PKI) is a **security architecture** developed to increase the confidentiality of information being exchanged over the insecure Internet. It includes hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates. In **cryptography**, the PKI helps to bind public keys with corresponding user identities by means of a certificate authority (CA). The following are the components of PKI:

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

Cryptographic keys can be delivered securely between users by PKI.




## Public Key Infrastructure (PKI) (Cont'd)

The public key cryptosystem uses a pair of a public key and a private key to assure secure communication over the Internet. In public key cryptosystem authentication, it is important to connect the correct person and the public key. This is accomplished with the help of Public Key Infrastructure (PKI). **Asymmetric** (public key) **cryptography** is the foundation technology of PKI, when sender and receiver agreed upon a secret communication using public key encryption with a digital signature.

The figure that follows shows how a message gets digitally signed by the organization involved in authentication and certification by means of PKI. In public key cryptosystems, the correspondence between a public key and the private key is taken care by the certification authority (CA), i.e., based on the public key the **CA determines the owner of the respective private key**. Initially, the user requests the certification authority for binding his or her public key; a certification authority digitally signs it and issues a public key certificate to the user. It binds the user's identity with the user's public key. In between the user and the CA, there exists an organization, the Registration Authority (RA). The job of the RA is to verify the identity of the user requesting the certificate face-to-face. There exists another authority in PKI, i.e., the validation authority (VA). The job of the VA is to check whether the certificate was issued by **trustworthy** a CA or not, i.e., is it valid or not. The sender and receiver can then exchange a secret message using public key cryptography.

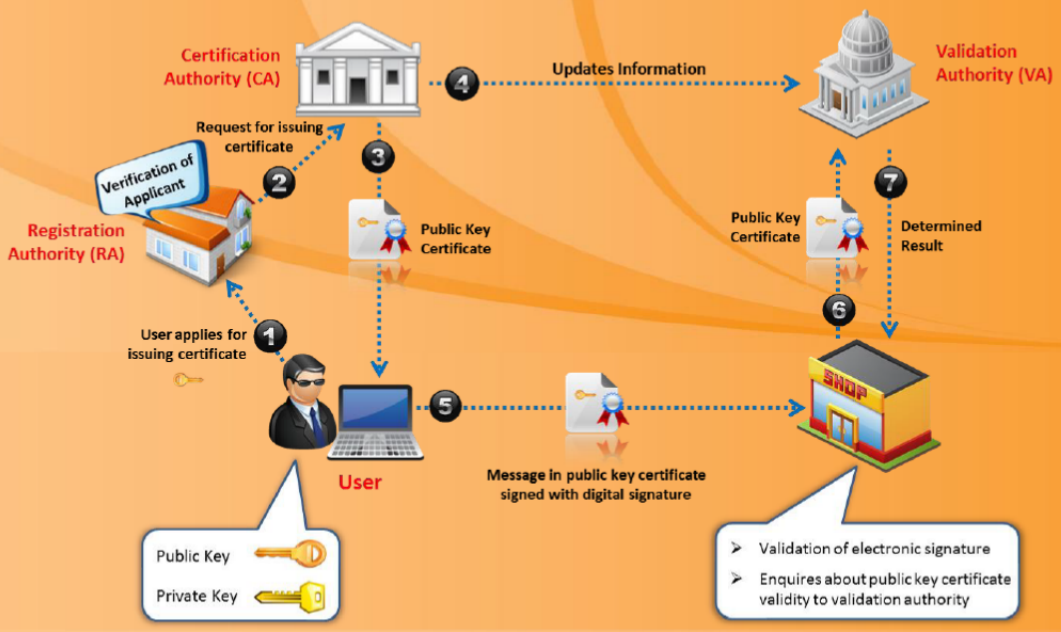
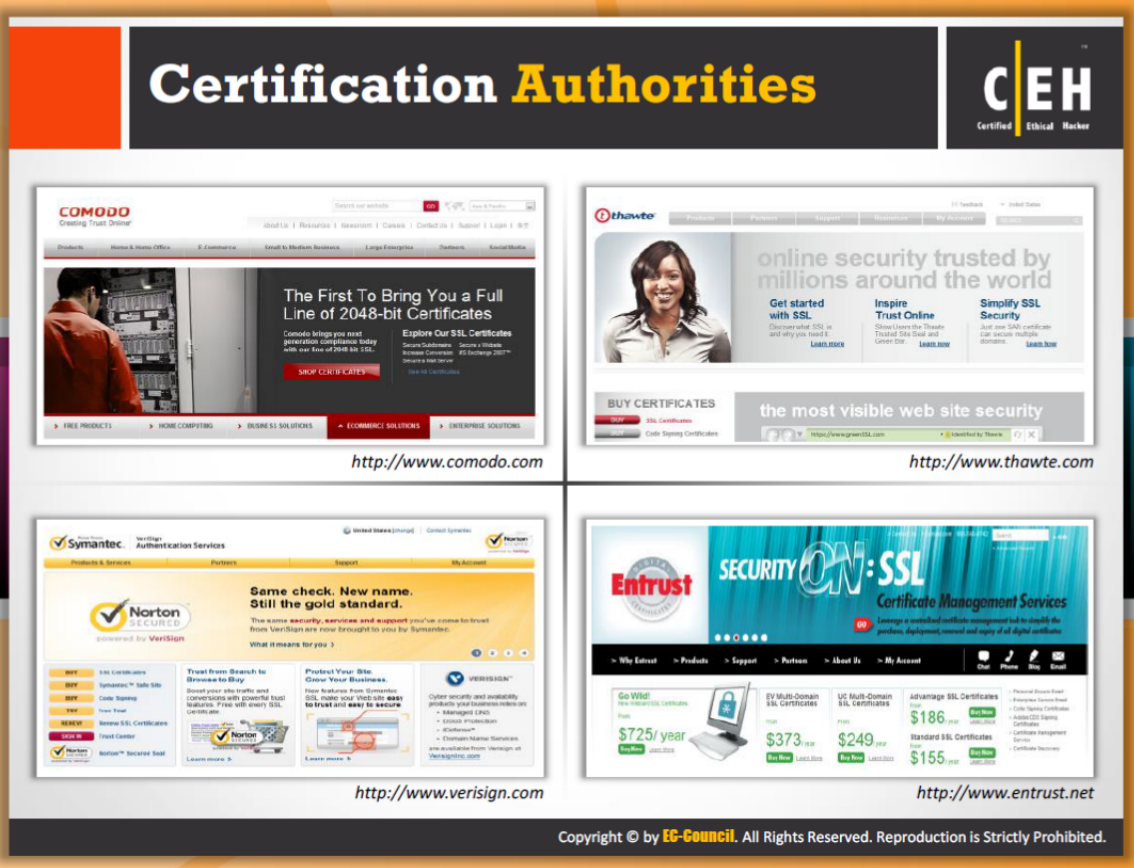



FIGURE 19.13: Public Key Infrastructure (PKI)




## Certification Authorities

Certification authorities are the entities that issue digital certificates. The following are some of the **certificate authorities**:

### Comodo

Source: http://www.comodo.com

Comodo offers a complete range of PKI digital certificates with strong SSL encryption available. It ensures standards of confidentiality, system reliability, and pertinent business practices as judged through qualified independent audits. The PKI (Public Key Infrastructure) management solutions offered by Comodo include **Comodo Certificate Manager** and **Comodo EPKI Manager**.

**Available Digital Certificates:**

- Extended validation (EV)-SSL
- Multi-domain EV SSL
- Wildcard SSL
- Unified communications (UC)
- Intel Pro Series

- General purpose SSL
- Secure Email - S/MIME
- Client authentication
- Code signing

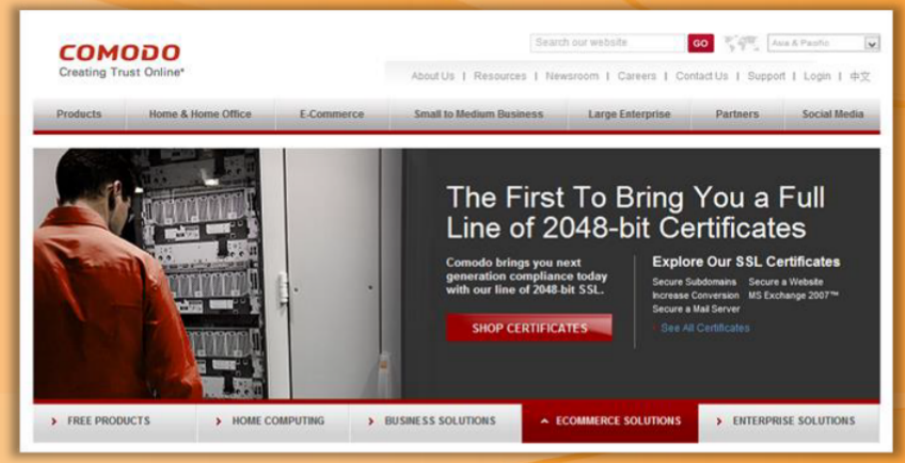



FIGURE 19.14: Comodo screenshot

## thwate

Source: http://www.thawte.com

thawte is a Certification Authority. thwate offers SSL and code signing digital certificates to secure servers, provides data encryption, authenticates users, protects privacy, and assures online identifies through stringent authentication and verification processes. The **SSL certificates** offered by thwate include Wildcard SSL Certificates, SAN /UC Certificates, SGC SuperCerts, and Extended Validation SSL Certificates.

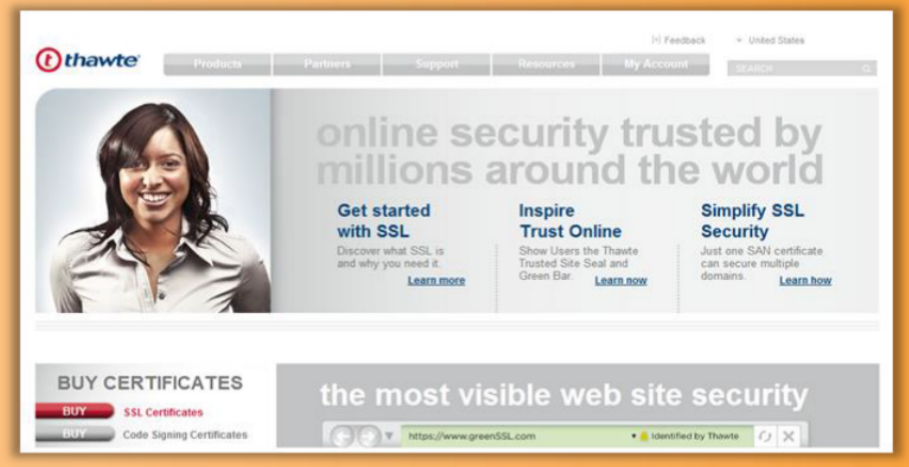



FIGURE 19.15: thawte screenshot

## Verisign

Source: http://www.verisign.com

VeriSign Authentication Services, now part of Symantec Corp. (NASDAQ: SYMC), provides solutions that allow companies and consumers to **engage in communications** and **commerce** online with confidence.

**SSL Certificates:**

- Secure Site Pro with EV
- Secure Site with EV
- Secure Site Pro
- Secure Site
- Managed PKI for SSL
- SSL for the Enterprise
- SSL Partner Programs
- Symantec Certificate Intelligence Center

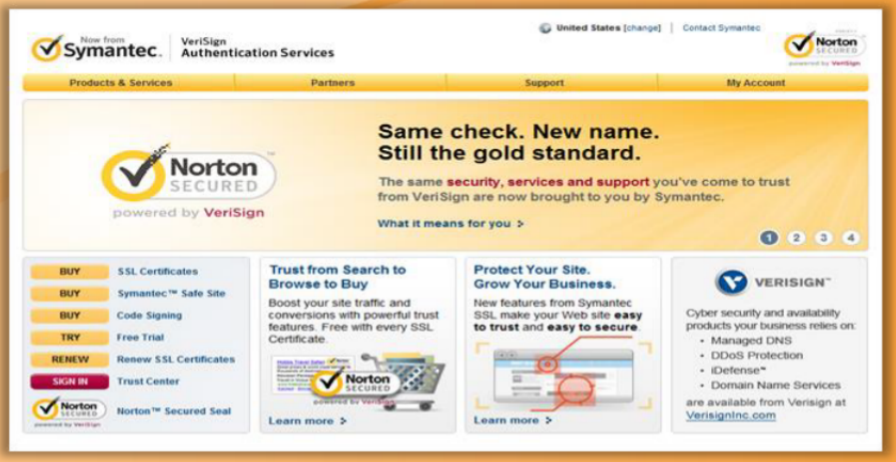


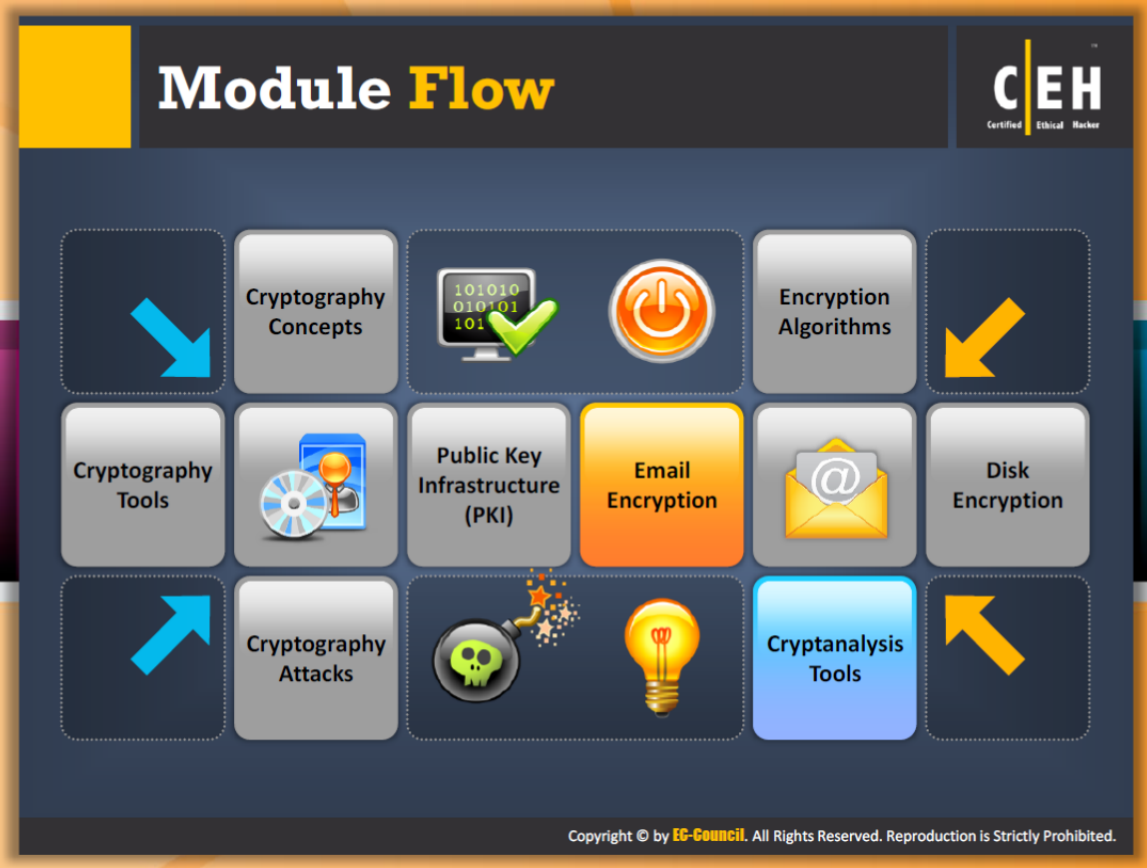

FIGURE 19.16: Verisign screenshot

## Entrust

Source: http://www.entrust.net

Entrust provides identity-based security solutions that empower enterprises, consumers, citizens, and the web. Entrust's solutions include strong authentication, fraud detection, digital certificates, SSL, and PKI. Entrust can deploy appropriate security solutions to help protect digital identities and information at multiple points to address **ever-evolving threats.**


FIGURE 19.17: Entrust screenshot

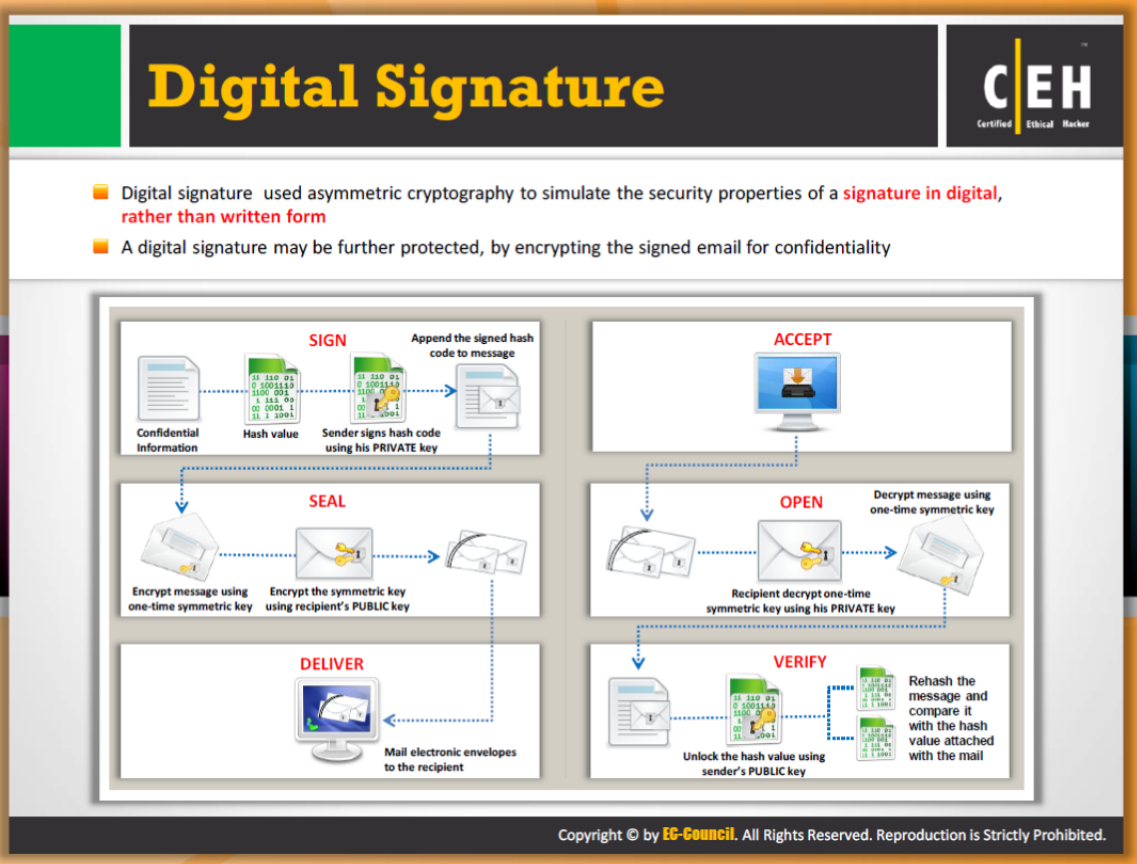



## Module Flow

At present, most businesses use email as the major source of communication as it is simple and easy to communicate or share information. These emails may contain **sensitive information** about their projects, updates, etc. If this information falls into the wrong hands, then the organizations may face **huge losses**. This risk can be avoided by encrypting the email messages. Email encryption is the means to transfer the plaintext message into an unreadable form.

| | |
|---|---|
| **Cryptography Concepts** | **Encryption Algorithms** |
| **Cryptography Tools** | **Public Key Infrastructure (PKI)** |
| **Email Encryption** | **Disk Encryption** |
| **Cryptography Attacks** | **Cryptanalysis Tools** |

This section focuses on various email security mechanisms such as **digital signatures, SSL, and TLS.**




## Digital Signature

A digital signature is a cryptographic means of authentication. Public key cryptography, which uses an **asymmetric key algorithm**, is used for creating the digital signature. The two types of keys in public key cryptography are the private key (which is known only to the signer and used to create the digital signature) and the public key (which is more widely known and is used by a relying party to verify the digital signature). A **hash function** is a process, or an algorithm, that is used in creating and verifying a digital signature. This algorithm creates a digital representation of a message, which is also known as a "**fingerprint**." This fingerprint is of a "hash value" of a standard length, which is much smaller than the message, but is unique to it. If any change is made to the message, it will automatically produce a different hash result; it is not possible to derive the original message from the hash value in case of a secure hash function, which is also known as a one-way hash function.

The hash result of the original message and the hash function that is used to create the digital signature are required to verify the digital signature. With the help of the public key and the new result, the verifier checks:

- If the digital signature is created with the **related private key**. If the new hash result is the same as the original hash result, which was converted into a digital signature during the **signing process**.

To correlate the key pair with the respective signer, the certification authority presents a certificate that is an electronic record of the public as the subject of the certificate, and confirms the identity of the signer as the related private key owner. The future signer is called the subscriber. The main function of a certificate is to bind a pair of public and private keys to a particular subscriber. The recipient of the certificate relies on a digital signature created by the subscriber named in the certificate. The public key listed can be used to verify that the private key is used to create the related digital signature.

The certification authority digitally signs the certificate to assure the authenticity of both the public key and the subscriber's identity. The authority's digital signature on the certificate can be verified with the help of the public key of the certification authority recorded in another certificate, which belongs to another **certification's authority**. This certificate can be authenticated with the help of another public key recorded in another certificate and so on.

The repository can be made to publish the certificate; the public key and its identity are available for verification of the certificate. The retrieval and verification of the digital signature is made with the help of an online database called repositories, which holds the certificates and other information. The certification authority may suspend or revoke the certificate.

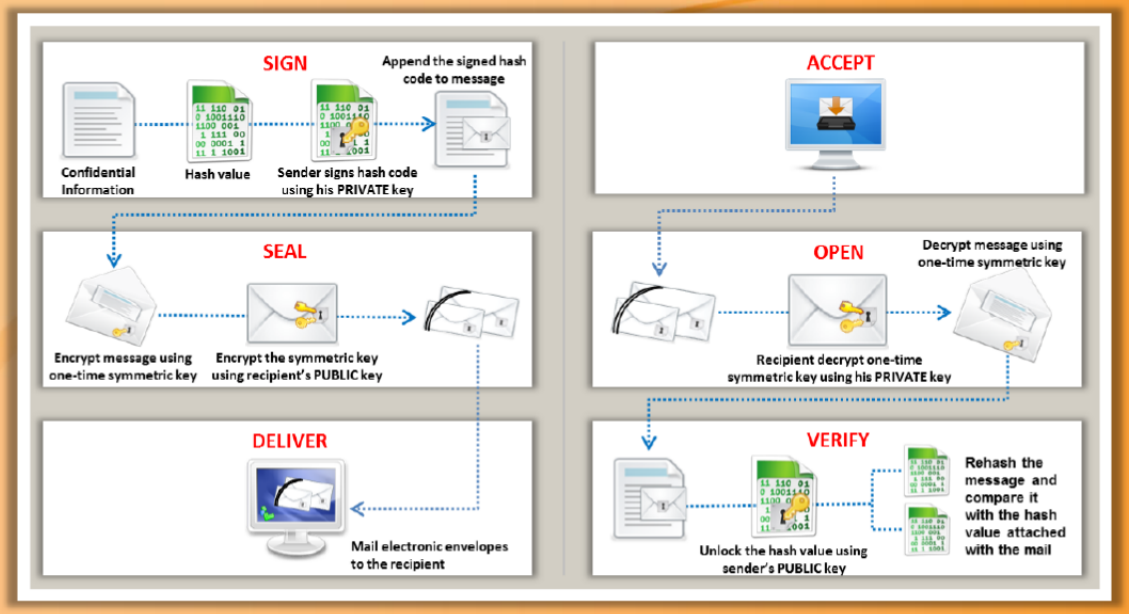
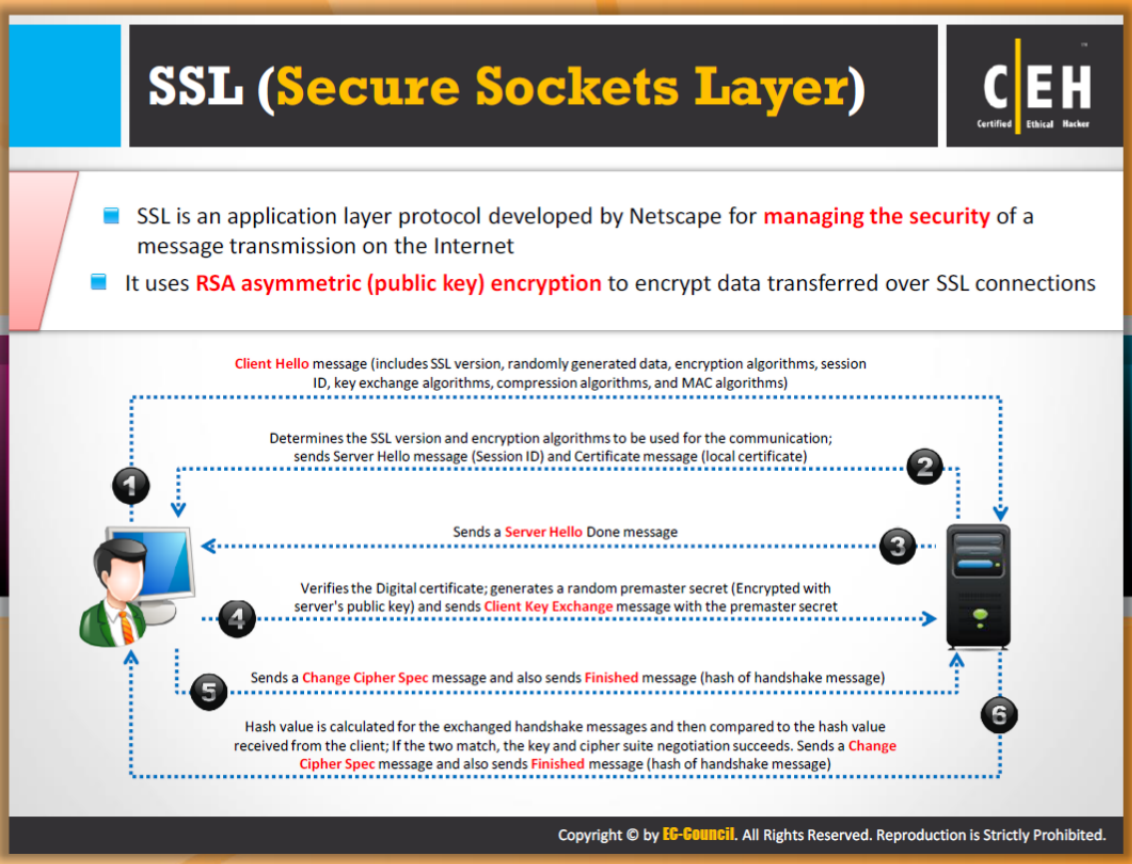



FIGURE 19.18: Digital signatures




## SSL (Secure Sockets Layer)

SSL is **acronym** for Secured Sockets Layer, developed by Netscape. It is a protocol for sending private documents over the Internet. It works with the help of the private key to encrypt data that is transferred over an SSL connection. The main motive behind designing the SSL protocol is to provide privacy between two communicating applications, such as a client and a server. Moreover, the protocol is designed to authenticate the server and the client; SSL requires a **reliable transport protocol** such as **TCP** for **data transmission** and **reception**.

Any application-layer protocol that is higher than **SSL**, such as **HTTP, FTP, and TELNET**, can be layered on top of SSL transparently. The SSL acts as an arbitrator between the encryption algorithm and session key, and also verifies the destination server before the transmission and reception of data. The complete data of the application protocol is encrypted, to ensure security. It also offers channel security which has three basic properties:

- It has a private channel, where the messages are encrypted after the simple handshake that defines the secret key.
- The channel is authenticated. The server endpoints are always authenticated but the client endpoints are optionally **authenticated**.
- The channel is reliable. The transmission has an integrity check.

An SSL session is responsible for the **SSL handshake protocol** to organize the states of the server and clients, thus ensuring the consistency of the protocol state machines (the states are not exactly parallel).

There are two different types of states: operating and pending. In addition to the two states, two additional states are also maintained; the read and write states. When the server or client obtains the cipher spec message, the message is copied into a current read state from the pending read state. In a similar way, when the data is transmitted from the server or client, it transmits a changed cipher spec message, and copies the message into the write current state from the pending write state. After the completion of the **handshake arbitration**, the server and client exchange the changed spec message and the communication is based on the newly agreed upon cipher spec. An SSL may include many secure connections, and it might have multiple concurrent sessions. The elements included in **session** state are as follows:

## Session Identifier

**Session identifier** is a random sequence of bytes transmitted by the server to identify an active or **presumable session** state:

- Peer Certificate – X509.v3[X509] is the certificate of the peer and may be null.
- Compression Method – Is the algorithm used to compress data prior to encryption.
- Cipher Spec – Enumerates the **bulk data encryption** and **MAC algorithms**. It also defines cryptographic attributes like the size of the hash.
- Master Secret – Is the **48-byte** secret shared between the client and server.
- Is Resumable – A flag specifies whether a new session can be started.

The elements of the connection state are as follows:

- Server and client random – Is the sequences of bytes, which are selected by the server and the client for every connection.
- Server write MAC secret – Is the secret used in MAC operations on data written by the server.
- Client write MAC secret – Is the secret used in MAC operations on data written by the client.
- Server write key – Is the huge cipher key for data encrypted by the server and decrypted by the client.
- Client write key – Is the cipher key for data encrypted by the client and decrypted by the server.
- Initialization vectors – In CBC (Cipher Block Chain) mode when the block cipher is used, an initialization vector is managed for every key. It is started by the SSL handshake protocol and is used to make the first cipher text. The last **cipher text block** of every text is used with the subsequent record.

- Sequence numbers – Every party maintains a different and unique sequence of numbers for the transmission and reception of messages for every connection. The appropriate sequence is set to **zero** depending on the party that sends and receives cipher spec.

## SSL Handshake Protocol Flow

The SSL handshake protocol works on top of the SSL record layer. These processes that are executed in the three handshake protocol are summarized as follows:

- The client sends a **hello message** to the server and the server must respond to the hello message with a hello message, or else the connection will fail due to the **occurrence of a fatal error**. The attributes that are established due to the server and client hello are: protocol version, session ID, cipher suite, and compression method.
- After the connection is established, the server sends a certificate to the client for authentication. In addition, a **server-key exchange message** might be sent. If the server is authenticated, the client may be requested for the certificate, if that is appropriate to the **cipher suite** selected.
- The server sends a hello done message, to inform that the handshake phase is complete and waits for the client's response.
- If the client receives a certificate request message, the client must respond to the message by sending a certificate message or "no certificate" alert. The client-key exchange message is sent and the content of the message depends on the public-key algorithm between the server hello and client hello. If the certificate sent by the client has signing ability, a **digitally signed certificate** verifies the message, and is transmitted.
- The client transmits the changed cipher spec message and copies the pending cipher spec into the current cipher spec. The client sends a message to initiate the completion of the message under the new algorithm, keys, and secrets. In response the server replies by sending its own changed cipher spec message, transfers the pending cipher spec to the current cipher spec, and initiates the completion of the message under the new cipher spec. This is the point of completion of the handshake and the server starts to exchange the application layer data.

The message of the previous session or the **replica** of an existing session is as follows:

The client initiates the communication by sending a hello message with the session I of the session that is to be resumed. The server checks its cache to look for the match of the session ID; if it finds a match it re-establishes the session under the specified session state with same session ID. This is the point where both the server and the client exchange the changed spec messages and proceed directly to the finished messages. After re-establishment, the server and the client exchange the data at the application layer. If the session I is not found, the server creates a **new session ID**, and the SSL client and server carry out a complete handshake.

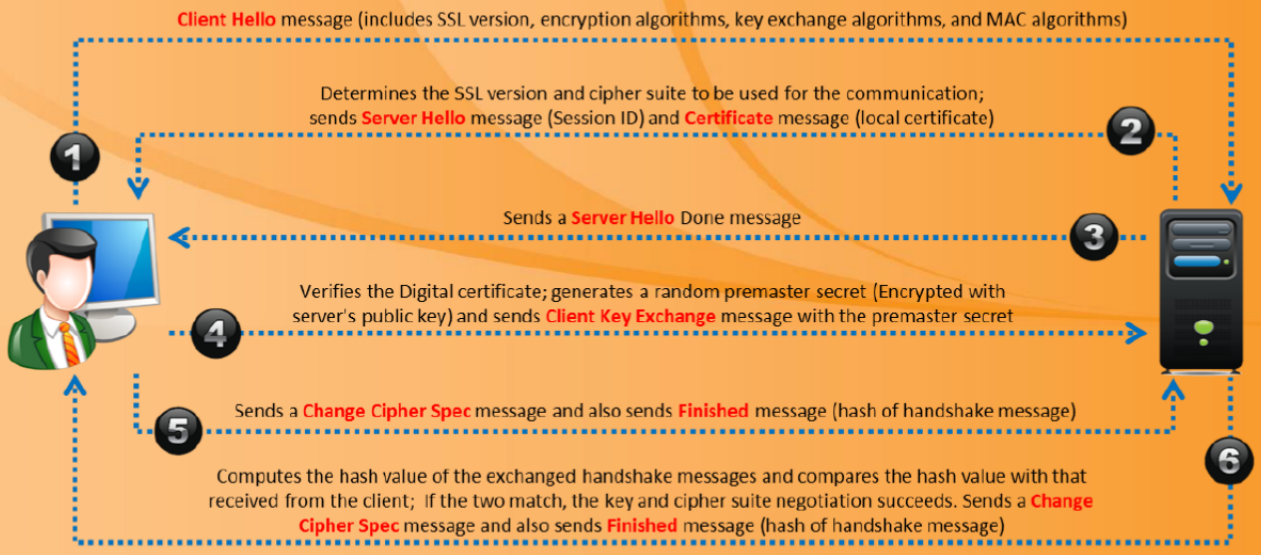
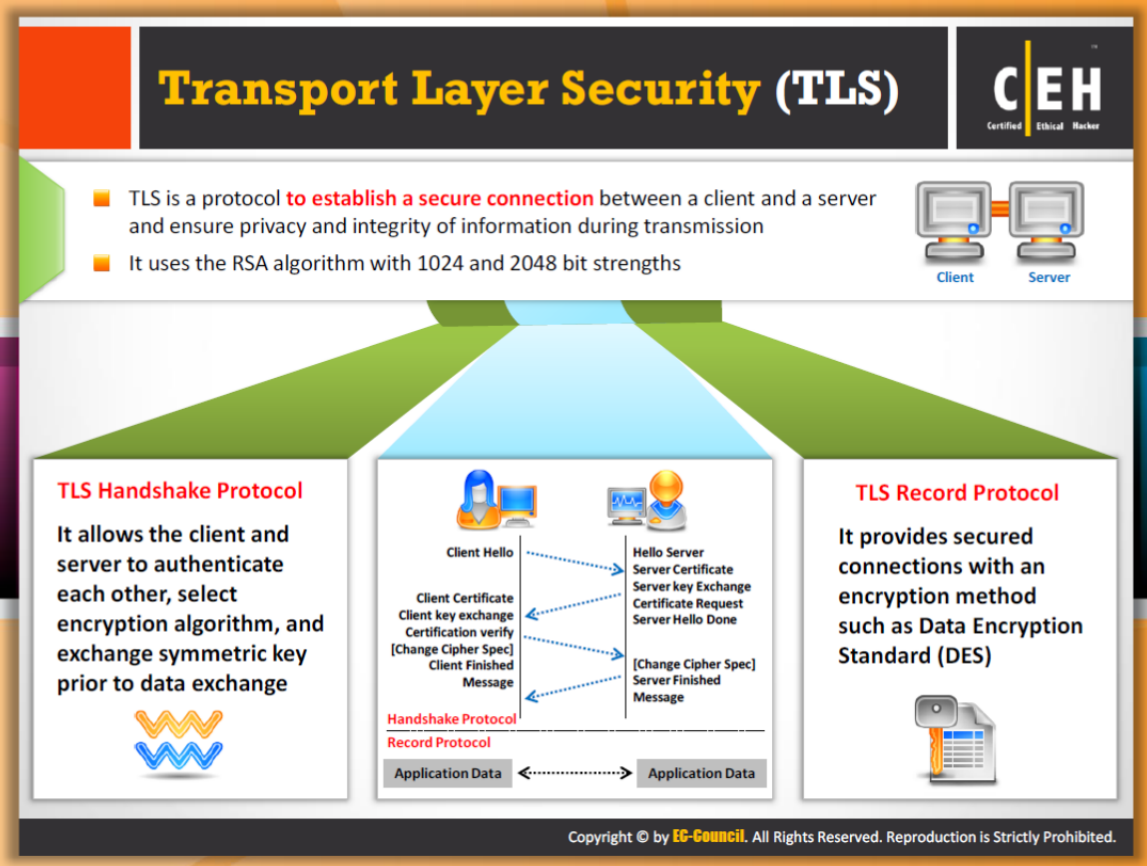



FIGURE 19.19: Depicting SSL Handshake Protocol Flow




## Transport Layer Security (TLS)

TLS is a protocol to establish a secure connection between a client and a server and ensure privacy and integrity of information during transmission. It is a cryptographic protocol intended to provide information security over the Internet. The **TLS encrypts** the network connection segments at the application layer for the transport layer. It uses asymmetric cryptography for key exchange, symmetric encryption for confidentiality, and message authentication codes for message integrity. With the help of TLS, you can reduce some of the risks such as tampering, message forgery mail communications, and eavesdropping during transmission of electronic mails or information.

TLS protocol consists of two layers:

- TLS record protocol
- TLS handshake protocol

## TLS Record Protocol

The TLS record protocol provides secure communications. It is intended for encryption, authentication, and compression (optional) of packets. Once the **handshake**

**process** is done, then record layer functions can be called at any time whenever there is a need to send or receive data. It is responsible for securing application data and also verifying its integrity and origin of the data. **TLS Record Protocol** manages the following:

- Dividing and reassembling messages
- Compressing and decompressing blocks (optional)
- Applying MAC (Message Authentication Code) and verifying incoming messages based on MAC
- Encrypting and decrypting messages

The outgoing **encrypted data** from the record protocol is sent to TCP layer for transport.

## TLS Handshake Protocol

The TLS handshake protocol is responsible for peers to agree upon **security parameters** for the record layer, authentication. This also negotiates a session consisting of session identifier, peer certificate, compression method, cipher spec, master secret, and information about resuming a connection. The figure that follows shows the process of client-authenticated TLS handshake:

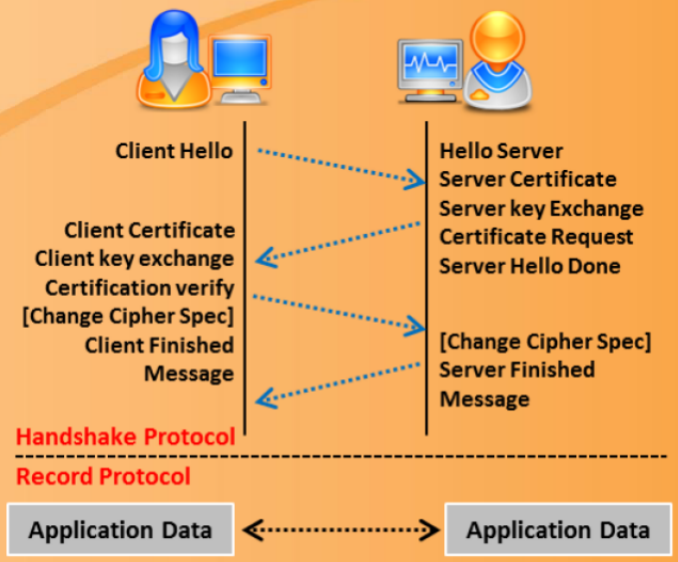
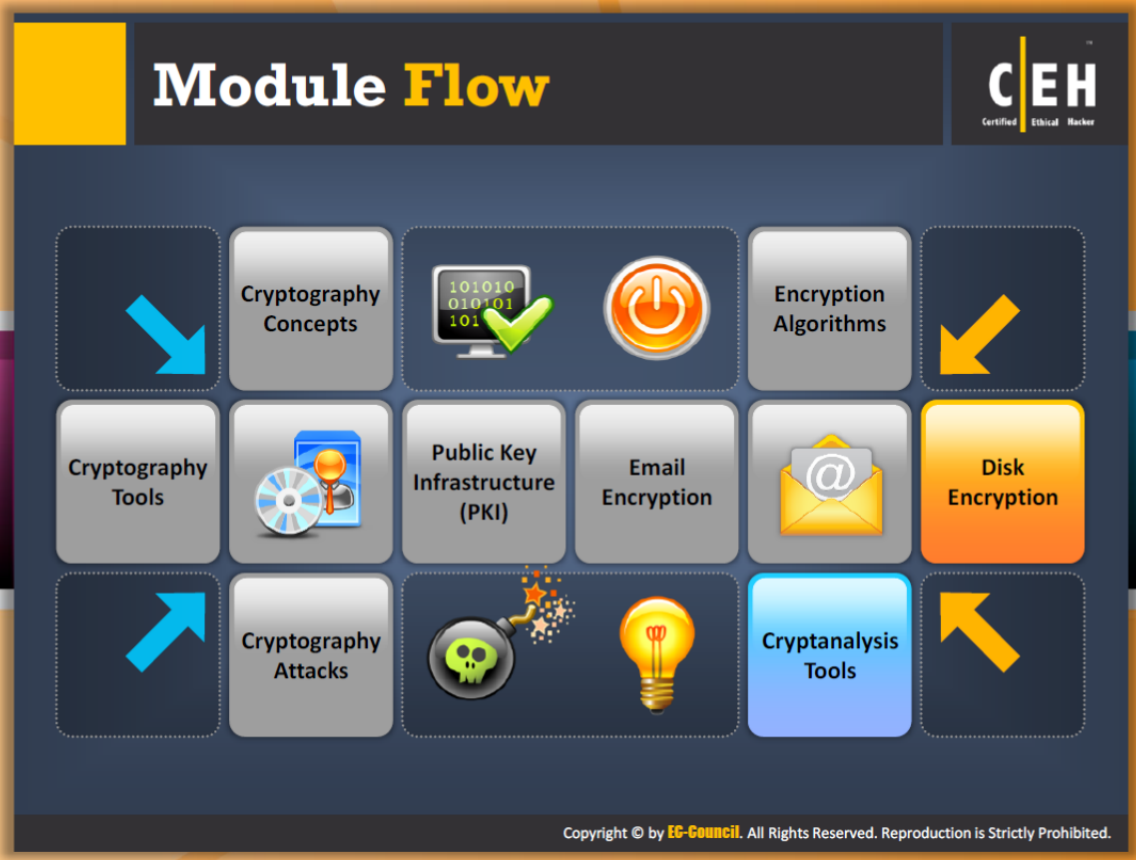



FIGURE 19.20: Showing the client-authenticated TLS handshake process

A handshake protocol exchanges a series of message in between a client and a server for a secure connection. Initially, the client sends a "**hello**" to the server. The server, in response to the client, sends "hello." During this period, the security capabilities including protocol version, compression method, cipher suite, session ID, and initial random number have been established. Then the server may send a certificate and key exchange and requests a certificate. Now, the server signals the end of the hello message. In response to the certificate request by the server, the client sends the certificate and key exchange. The client then sends certificate verification. Both the client and server exchange their **cipher suite** and finish the handshake protocol.

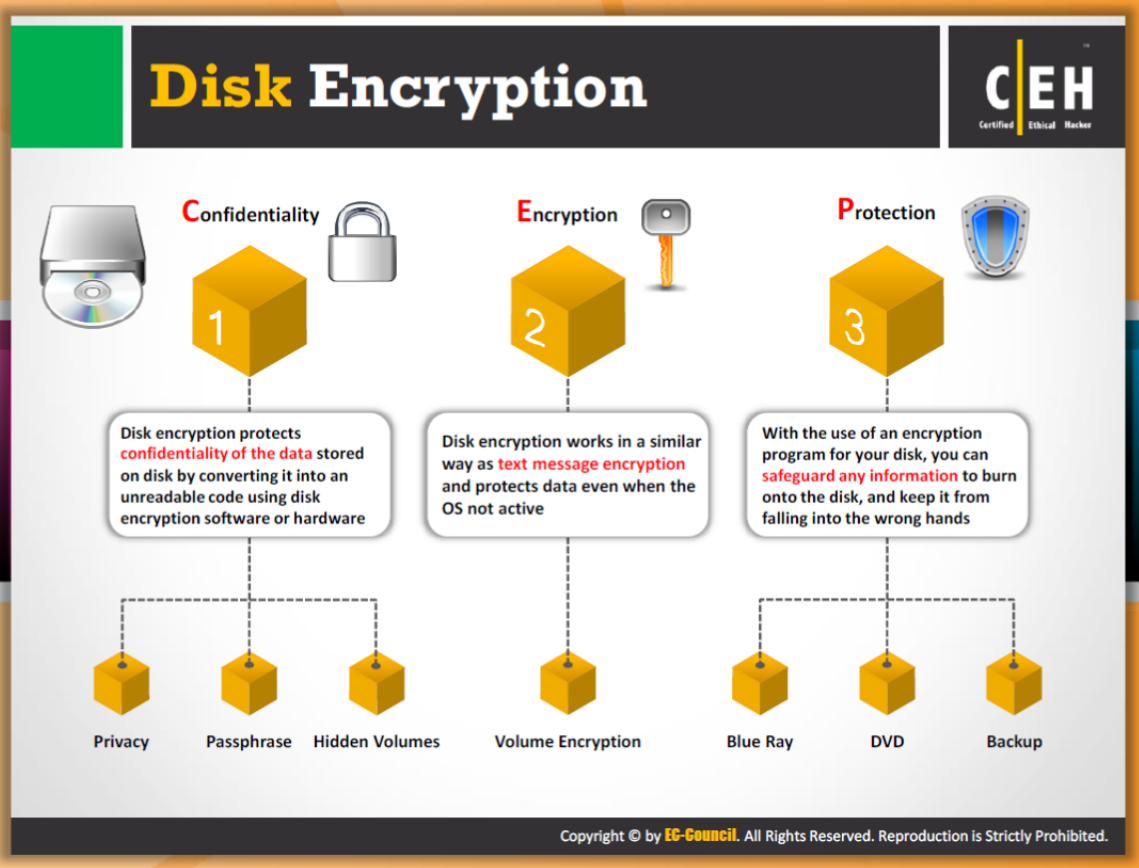



## Module Flow

So far, we have discussed cryptography, the need for cryptography, **cryptographic encryption algorithms**, cryptography tools, PKI, and email encryption. In addition to all these encryption methods, there is one more encryption method: disk encryption.

| | |
|---|---|
| **Cryptography Concepts** | **Encryption Algorithms** |
| **Cryptography Tools** | **Public Key Infrastructure (PKI)** |
| **Email Encryption** | **Disk Encryption** |
| **Cryptography Attacks** | **Cryptanalysis Tools** |

This section describes disk encryption and disk encryption tools.




## Disk Encryption

Disk encryption is the process of securing data by transferring it into unreadable code that cannot be deciphered by unauthorized persons. You can use **disk encryption software** or hardware to encrypt every bit of information that is written on the disk.

**Disk encryption** works similar to text message encryption. With the use of an encryption program for the user's disk, the user can safeguard any, and all, information burned onto the disk and save it from falling into **wrong hands**.

A computer disk is a round plate onto which data is recorded and/or burned. If the user needs to store information on a disk, and keep it safe, it is recommended that an encryption program be used. Encryption software, for disks, scrambles the information burned on the disk into an illegible code. It is only after the disk information is **decrypted**, that it can be read and/or used.

Encryption for disks is useful when the user needs to send sensitive information through the mail. For instance, the user needs to mail his or her friend a disk, but cannot take the risk of it being stolen and the information is being compromised. In this case, the user could simply encrypt the information on the disk and then rest assured, even if the **disk** is lost or stolen, the information on it would not be compromised.

In addition, disk encryption can also be useful in protecting the real-time exchange of information from being compromised. When the exchange of information is made in an

encrypted form, the chances of the information being compromised are minimized. The only way the attacker can access the information is by decrypting the message, which can only be done via the **authentication process**.

Furthermore, the **encryption software** installed on one's system ensures the security of the system. Thus, it is recommended to install encryption software on systems that hold valuable information and/or are exposed to unlimited data transfer in order to protect the data and information from compromise.


# Disk Encryption Tool: TrueCrypt

Source: http://www.truecrypt.org

TrueCrypt is software that allows you to establish and maintain an encrypted volume (data storage device). No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. The entire **file system** is encrypted (e.g., file names, folder names, contents of every file, free space, meta data, etc).

**Main Features:**

- Creates a virtual encrypted disk within a file and mounts it as a real disk
- Encrypts an entire partition or storage device such as USB flash drive or hard drive
- Encrypts a partition or drive where Windows is installed (pre-boot authentication)
- Encryption can be hardware-accelerated on modern processors
- Provides plausible deniability, in case an adversary forces you to reveal the password
- Hidden volume (**steganography**) and hidden operating system

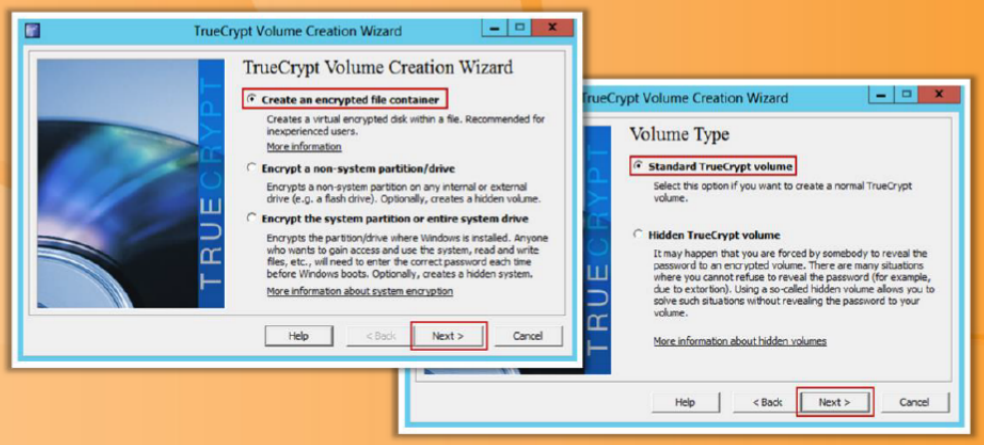



FIGURE 19.21: TrueCrypt Screenshot


# Disk Encryption Tool: GiliSoft Full Disk Encryption

Source: http://www.gilisoft.com

GiliSoft Full Disk Encryption allows you to encrypt all disk partitions, including the system partition. Through password protecting a disk, disk partition, or operating system launch, the program disables any **unauthorized** reading/writing activity on your disk or PC and restricts access and launch of specific disks and files. It provides automatic security for all information on endpoint hard drives, including user data, operating system files, and temporary and erased files. For maximum data protection, **multi-factor pre-boot** authentication ensures user identity, while encryption prevents data loss from theft.

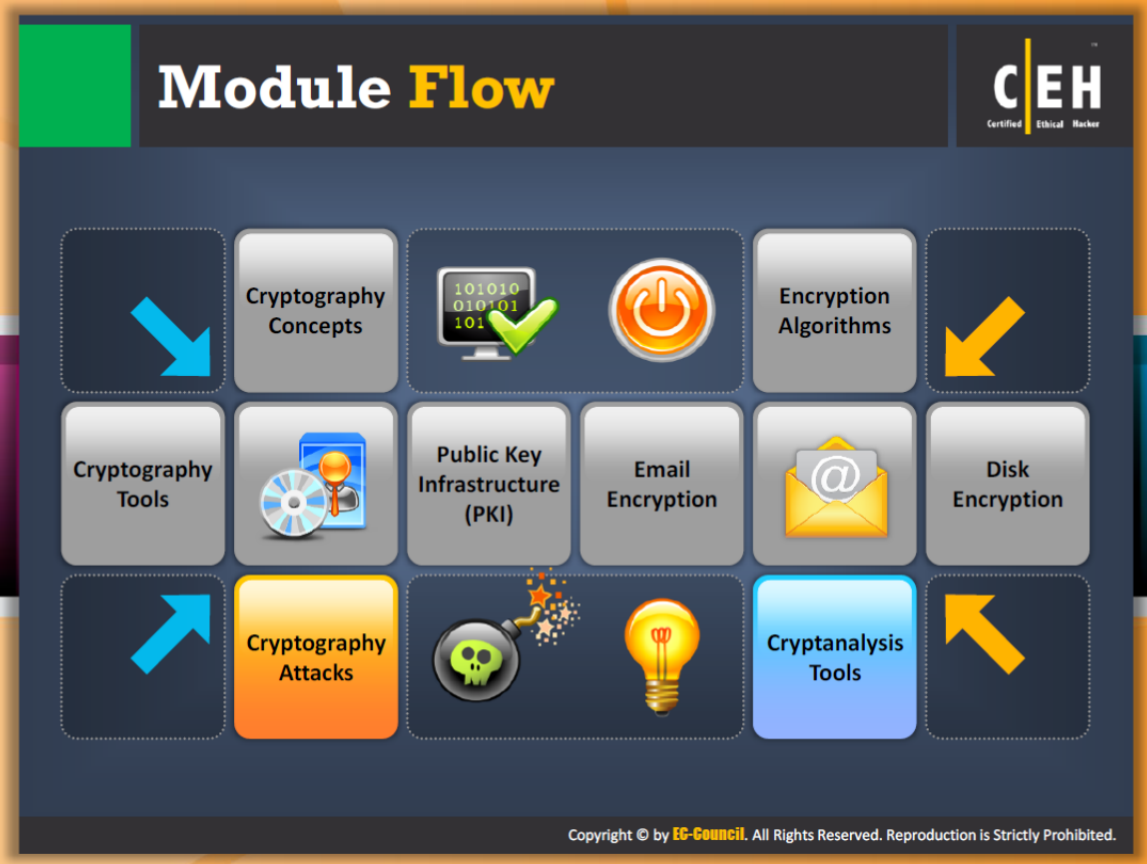

FIGURE 19.22: GiliSoft Full Disk Encryption screenshot


## Disk Encryption Tools

In addition to TrueCrypt and GiliSoft Full Disk Encryption, there are many other disk encryption tools that allow you to fully encrypt all data. A list of **disk encryption tools** is mentioned below as follows. All these tools have a common goal, i.e., encrypting a disk partition. But environment or purpose may change. If one tool is intended to create a virtual encrypted disk of the **target disk partition**, then the other may be intended to encrypt data on Pocket PCs running Windows Mobile and so on:

- DriveCrypt available at http://www.securstar.com
- ShareCrypt available at http://www.securstar.com
- PocketCrypt available at http://www.securstar.com
- Rohos Disk Encryption available at http://www.rohos.com
- R-Crypto available at http://www.r-tt.com
- SafeBit Disk Encryption available at http://www.safebit.net
- DiskCryptor available at http://diskcryptor.net
- alertsec available at http://www.alertsec.com
- Symantec Drive Encryption available at http://www.symantec.com

- DriveCrypt Plus Pack available at http://www.securstar.com

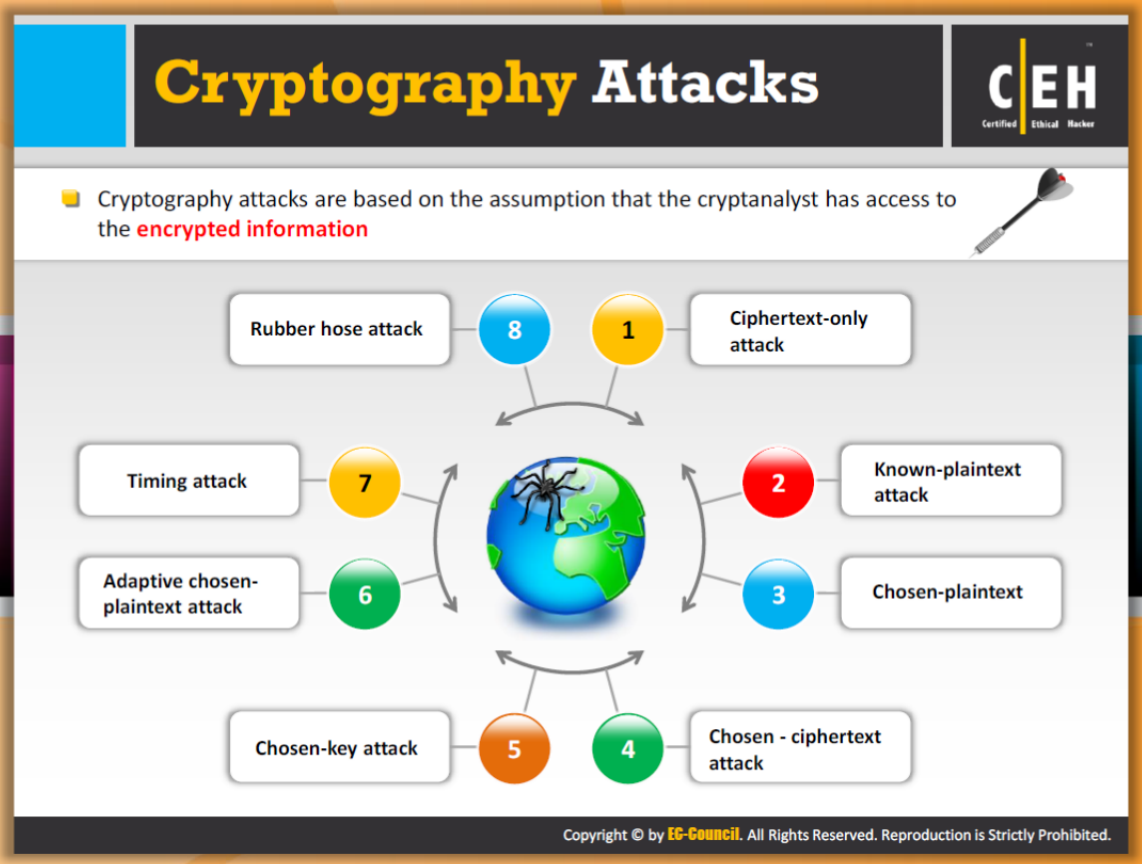



## Module Flow

So far, we have discussed cryptography concepts, various cryptography mechanisms, and encryption algorithms. Now it's time to discuss how cryptography systems can be exploited by an external user.

| | |
|---|---|
| **Cryptography Concepts** | **Encryption Algorithms** |
| **Cryptography Tools** | **Public Key Infrastructure (PKI)** |
| **Email Encryption** | **Disk Encryption** |
| **Cryptography Attacks** | **Cryptanalysis Tools** |

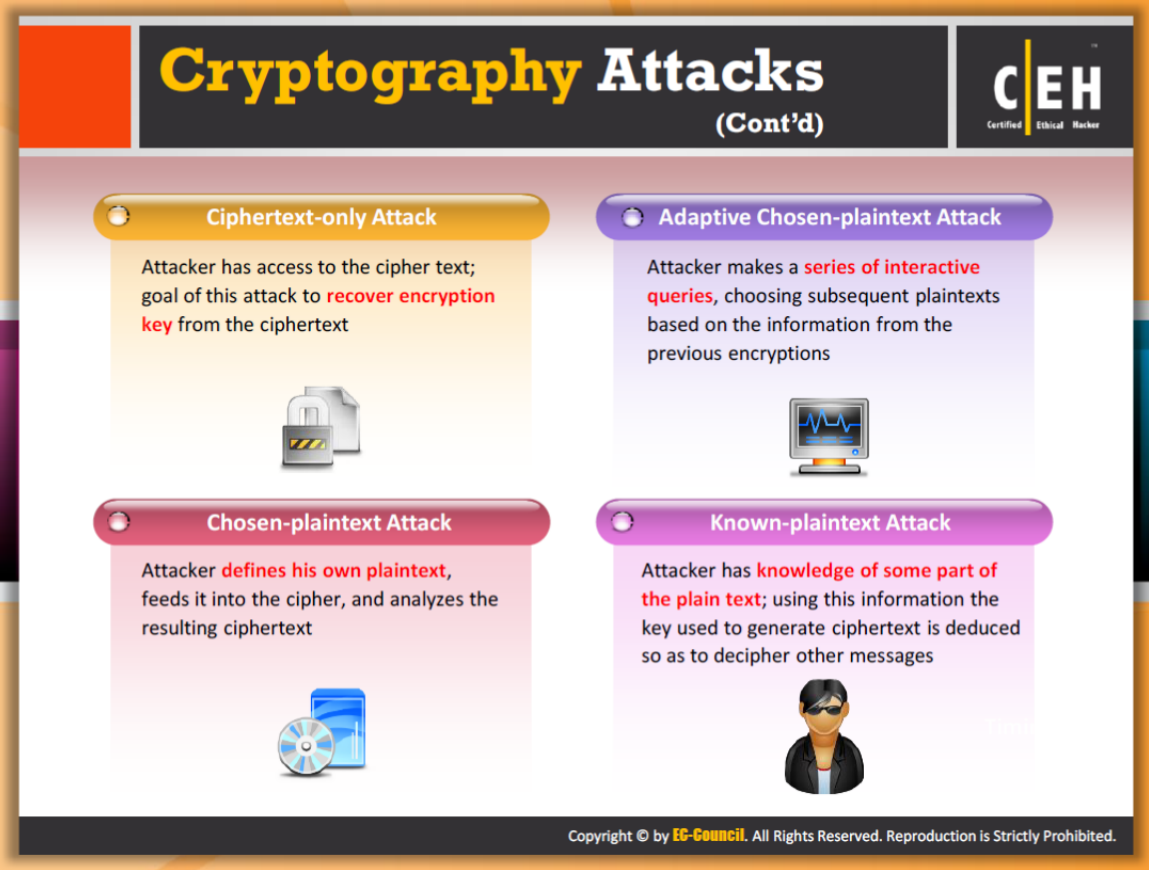
This section focuses on various types of cryptography attacks, code breaking methodologies, and other kinds of attacks that exploit cryptography systems.




## Cryptography Attacks

Cryptographic attacks are the means by which the attacker **decrypts** the ciphertet (breaks the ciphertext) without the knowledge of the key. In these attacks, the attacker subverts the cyrptographic system's security by exploiting the loopholes in code, cipher, cryptographic protocol or key management scheme. **Cryptography attacks** are based on the assumption that the cryptanalyst has knowledge of the information encrypted. Attackers have found various attacks for defeating the cryptosystem and they are categorized into eight types:

- Ciphertext only attack
- Known-plaintext attack
- Chosen-plaintext
- Chosen-ciphertext attack
- Chosen key attack
- Adaptive chosen-plaintext attack
- Timing attack
- Rubber hose attack




## Cryptography Attacks (Cont'd)

Attackers gain access to the content of the encrypted message through **cryptanalysis by defeating the cryptographic security algorithms**, even without the knowledge of encryption details. Though the algorithms are strong and are resistant to all attacks, the demands of practical cryptosystem easily introduce vulnerabilities. These vulnerabilities are the sources of various cryptography attacks. As discussed previously, there are eight types of cryptography attacks. All these attacks try either to retrieve the key or expose the plaintext. These attacks are distinguished based on the information available to the cryptanalyst to mount an attack. The main goal of attackers in all the cases is to decrypt the new pieces of **encrypted** message without additional information.

## Ciphertext only attack

A ciphertext only attack is one of the basic types of active attacks because it is very easy for the attacker to get ciphertext by **sniffing** the traffic of any individual. In this type of attack, the attacker will have access only to ciphertexts of several messages, all of which were encrypted using the same encryption algorithm. Finding the key used for encryption is the main objective of the attacker as it allows the attacker to decode all the messages encrypted with the respective key.

## Adaptive chosen-plaintext attack

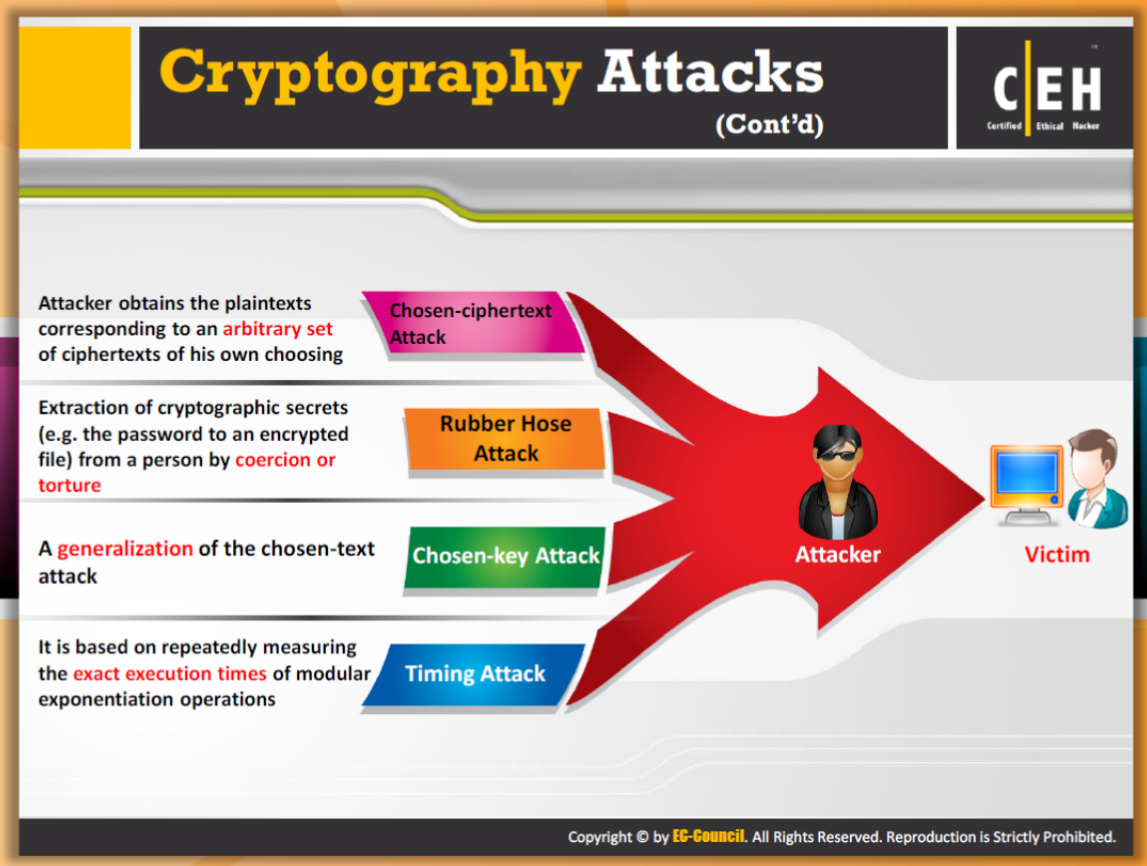
An adaptive chosen-ciphertext is the **collaborative version** of the **chosen-plaintext** attack. In this type of attack, the attacker choses further ciphertexts based on prior results. Here the cryptanalyst not only chooses the plaintext that is encrypted but can also modify his or her choice based on the results of the previous encryption.

## Chosen-ciphertext attack

In a chosen-ciphertext attack, the attacker chooses some part of ciphertext to be decrypted and tries to find out the corresponding decrypted plaintext. This is usually done with the help of a decryption oracle (a machine that decoded the text without disclosing the key). Basically, this type of attack is applicable to **public-key cryptosystems**. This attack is harder to perform when compared to other attacks, and the attacker needs to have complete control of system containing cryptosystem in order to carry out this attack.

## Rubber hose attack

In a rubber hose attack, the attacker extracts the secret key from the user by **threatening**, **blackmailing**, or torturing him or her until the key is handed over.




# Cryptography Attacks (Cont'd)

## Chosen-plaintext

This is more powerful than a plaintext attack. In this type of attacker, the attacker not only has access to the **ciphertext** and associated plaintext for several messages, but also chooses the plaintext that is encrypted, and obtains the resulting ciphertext.

## Known-plaintext attack

In a known-plaintext attack, the attacker has access to the ciphertext of one or more messages as well as access to the respective plaintext. With the help of both these items, the **cryptographic** key can easily extracted. The attacker can recover the remaining encrypted, zipped files with the help of the extracted key.

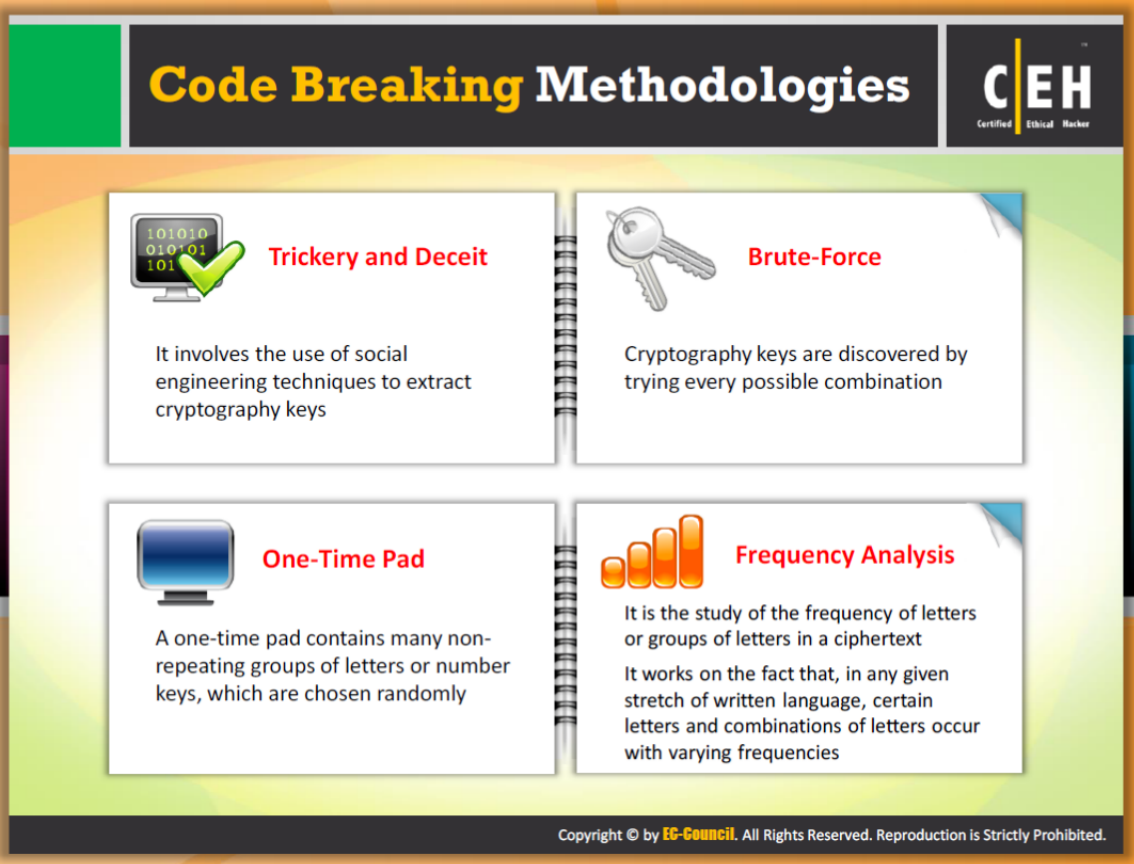
In general, most people start their messages with the same type of beginning notes such as greetings and close with the same type of ending such as specific salutations, contact information, name, etc. Attackers can use this as an advantage to launch known-plaintext attacks. Here the attacker has some plaintext (i.e., the data that are the same on each message) and can capture an encrypted message, and therefore capture the **ciphertext**. Once the few parts of the message re discovered, the remaining can easily be accomplished with the help of reverse engineering, frequency analysis, or **brute force** attempts.

## Chosen key attack

A chosen key attack is a generalization of the **chosen-text attack**. In this attack, the attacker has some knowledge about the relationship between the different keys, but cannot choose the key.

## Timing Attack

A timing attack also is known as a side channel attack. In this type of attack, the attacker tries to compromise a cryptosystem by analyzing the time taken to execute **cryptographic algorithms**.




## Code Breaking Methodologies

The strength of an encryption algorithm is measured, in large part by cryptanalysts, by using various code breaking techniques. The various **code-breaking** techniques that are available are:

- Brute-Force
- Frequency Analysis
- Trickery and Deceit
- One-Time Pad

## Brute-Force

Code-breakers, or cryptanalysts, want to recover the plaintext of a message without knowing the required key in advance. They may first try to recover the key, or go after the message itself. One of the familiar ways of the **cryptanalytic** technique is brute-force attack or an exhaustive search, (where the keys are guessed by trying every possible combination).

The efficiency of the brute-force depends on the hardware configuration. Usage of faster processors means testing more keys per second. Michael Weiner, put forth a brute-force attack

on the DES with the help of specially designed computers with **cryptographers** sounding the old standard's death knell.

Moreover, the combination of advanced factoring and the faster computers used in the recent attacks on **RSA-129**, makes algorithms appear weak. The NSA that has top computing power is the center of the brute-force attack.

## Frequency Analysis

Frequency analysis of the letters makes the brute-force method not a suitable method for attacking the cipher. For example the letter "e" is the common word in the English language and the letter "k" appears commonly in the ciphertext, it can be concluded reasonably that k=e, and so on.

Encrypted source codes are more exposed to the attacks because few words like "**#define**," "struct," "else," and "return" are repeated frequently. Frequency analysis was first used by papal courts in the Middle Age, which built frequency tables for Latin and Italian words. Sophisticated cryptosystems are required to maintain the security of the messages.

## Trickery and Deceit

There has always been a need for a high level of mathematical and cryptographic skills, but trickery and deceit have a long history in **code-breaking** as well the value of the encrypted data must be below the cost entitled to break the algorithm. In the modern world, computers are faster and cheaper, therefore it would be better to check the limits of these two parameters.

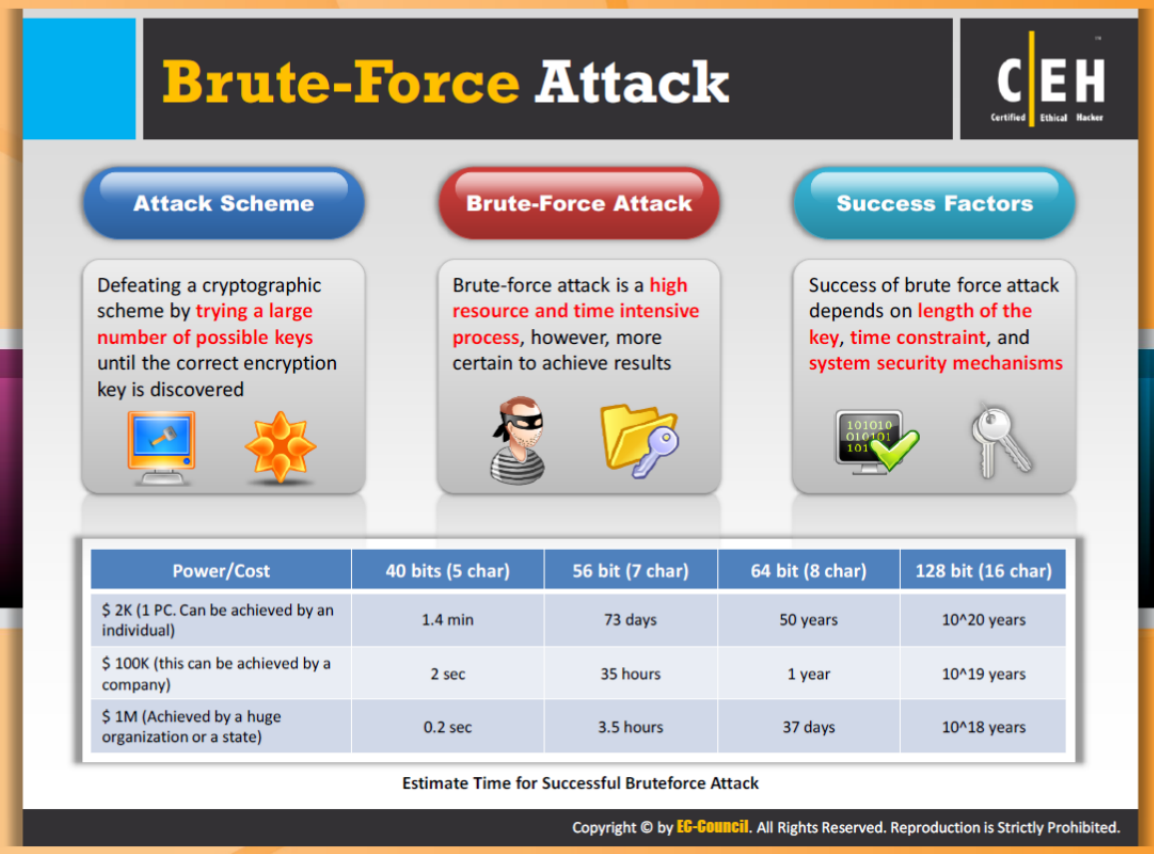
## One-time Pad

It is considered that any cipher can be cracked if sufficient time and resources are provided. But there is an exception called a one-time pad, which is considered to be unbreakable even after infinite resources are provided.

A one-time pad contains many **non-repeating** groups of letters or number keys, which are chosen randomly. These are then pasted together on a pad.

Bob encrypts only one plaintext character with the pad and Alice decrypts each and every character of the ciphertext with the help of the same key characters from an identical pad. After the use, the characters are securely removed from the pad. The major drawback of the one-time **padding** is the length of the pads. The length of key is same as the length of the message, which makes it impossible to encrypt and send large messages.

The Soviet spies commonly used one-time pads during the Cold War. The agent carried the encrypted message to the field, leaving the identical pad at the headquarters. The well-known, one-time padding was used on the communication lines between **Moscow** and **Washington**.




## Brute-force Attack

It is very difficult to crack cryptographic systems as they have no practical weaknesses to exploit. But, it is not impossible. Cryptographic systems use **cryptographic algorithms** to encrypt a message. These cryptographic algorithms use a key to encrypt or decrypt messages. In cyrptography, this key is the important parameter that specifies the transformation of plaintex to ciphertext and vice versa. If you are able to guess or find the key used for decryption then you can decrypt the messages and read it in clear text; **128-bit keys** are commonly used and considered strong. From security perspectives to avoid the key being guessed, the cryptographic systems use randomly generated keys. This makes you put a lot of effort in guessing the key. But you still have a choice to determine the key used for encryption or decryption. Attempt to decrypt the message with all possible keys until you discover the key used for encryption. This method of discovering a key is usually called a brute-force attack. In a brute-force attack, the attacker tries every possible key until the message can be decrypted. But this needs a huge amount of processing power for determining the key used to secure cryptographic communications. For any **non-flawed protocol**, the average time needed to find the key in a brute-force attack depends on the length of the key. If the key length is small, then it will take less time to find the key. If key length is larger, then it will take more time to discover the key. A **brute-force attack** will be successful if and only if enough time is given for discovering the key. However, the time is relative to the length of the key.

The difficulty of a brute-force attack depends on various issues, such as:

- Length of the key
- The numbers of possible values each component of the key can have
- The time it takes to attempt each key
- If there is any mechanism, which locks the attacker out after a certain number of failed attempts

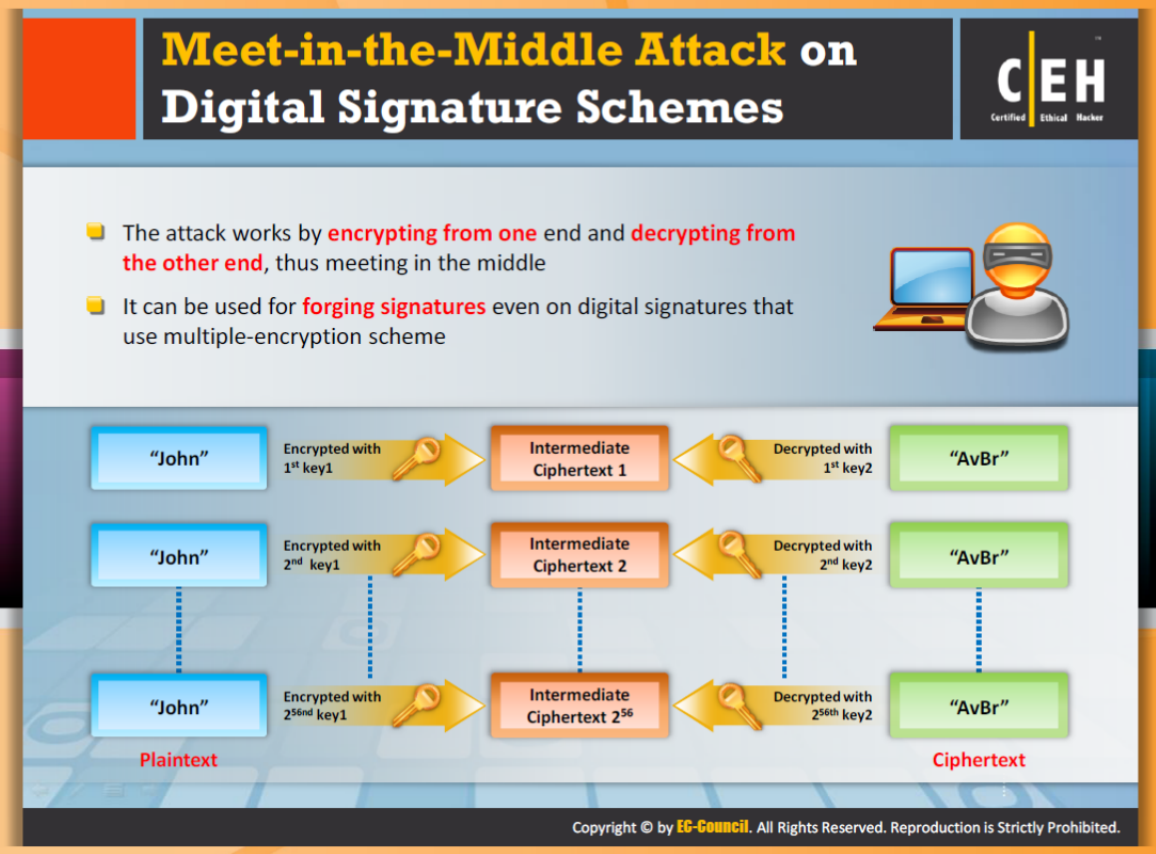
For example, if a system could brute force a **DES 56-bit key** in one second, then for an AES 128-bit key it takes approximately **149 trillion** years to brute force. To perform a brute-force attack, the time is doubled for every additional bit of key length; the reason behind it is that the number of potential keys is doubled.

A brute-force attack is, however, more certain to achieve results.

Estimate Time for Successful Brute – Force Attack

| Power/Cost | 40 bits (5 char) | 56 bit (7 char) | 64 bit (8 char) | 128 bit (16 char) |
|---|---|---|---|---|
| $ 2K (1 PC. Can be achieved by an individual) | 1.4 min | 73 Days | 50 Years | 10^20 Years |
| $ 100K (this can be achieved by a company) | 2 Sec | 35 Hours | 1 Year | 10^19 Years |
| $ 1M (Achieved by a huge organization or a state) | 0.2 Sec | 3.5 Hours | 37 Days | 10^18 Years |

TABLE 19.2: Time estimation for successful Brute-Force Attaack




# Meet-in-the-Middle Attack on Digital Signature Schemes

A meet-in-the-middle attack is the best attack method for cryptographic algorithms using multiple keys for encryption. This attack reduces the number of brute force permutations needed to decode text that has been encrypted by more than one key and is conducted mainly for forging signatures on mixed type digital signatures. A **meet-in-the-middle** attack uses space-time trade-off; it is also known as birthday attack because it exploits the mathematics behind the **birthday paradox**. It takes less time than an exhaustive attack. It is called a meet-in-the-Middle attack because this attack works by encrypting from one end and decrypting from the other end, thus meeting in the middle.

In the meet-in-the-middle attack, the attacker uses a known plaintext message. The attacker has access to both the plaintext as well as the respective encrypted text.

Consider an example where the plain text is "John" and the resulting double DES encrypted message is "AvBr."

In order to recover both the keys, i.e. key1 and key2, that are used for encryption, the attacker performs a brute-force attack on key1 using all $2^{56}$ different Single DES possible keys to encrypt the plaintext of "John" and saves each key and the resulting intermediate ciphertext in a table. The attacker conducts brute force on key2, decrypting "AvBr" up to $2^{56}$ times. The attack is successful, when the second brute-force attack gives the same result as that of the

intermediate ciphertext present in the ciphertext table after **first brute-force attack**. Once the match is found, both keys can be determined and the attack is complete. This attack at most takes $2^{56}$ plus or maximum $2^{57}$ total operations. This enables the attacker to gain access to the data easily when compared with the Double DES.

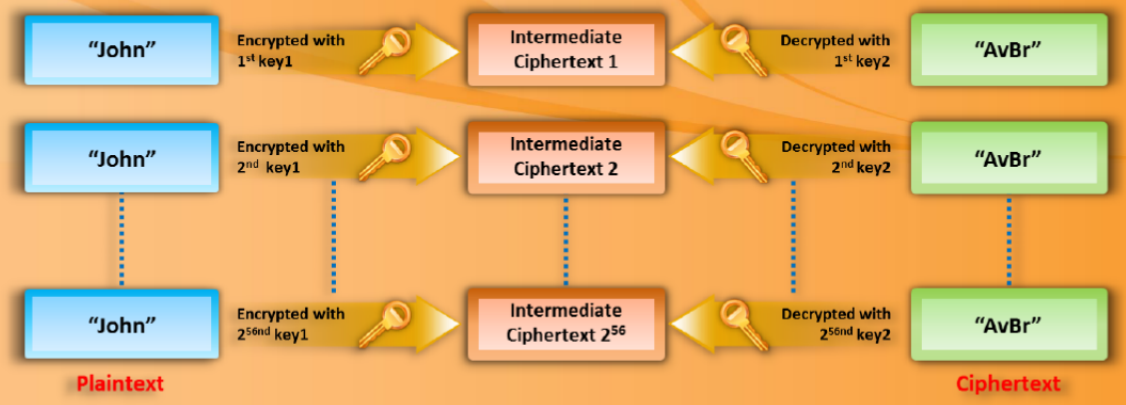
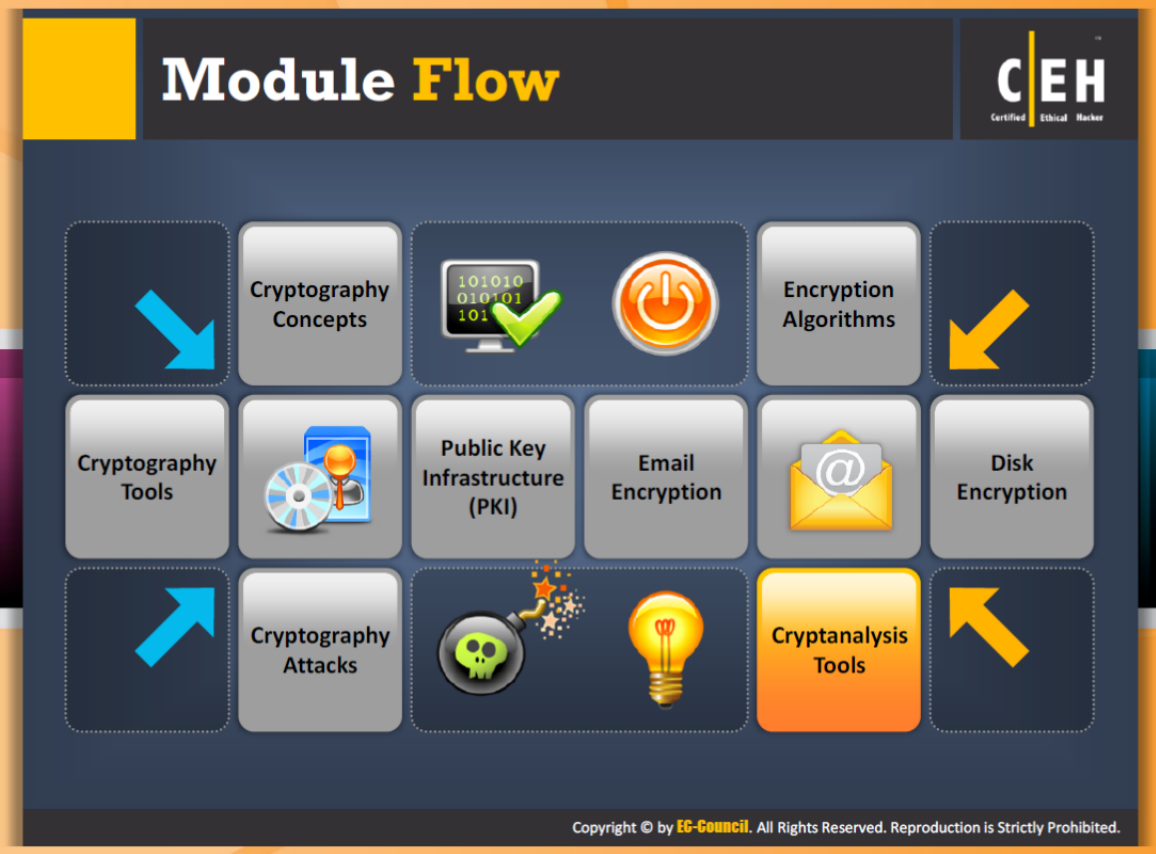



TABLE 19.23: Example illustrating Meet-in-the-middle attack

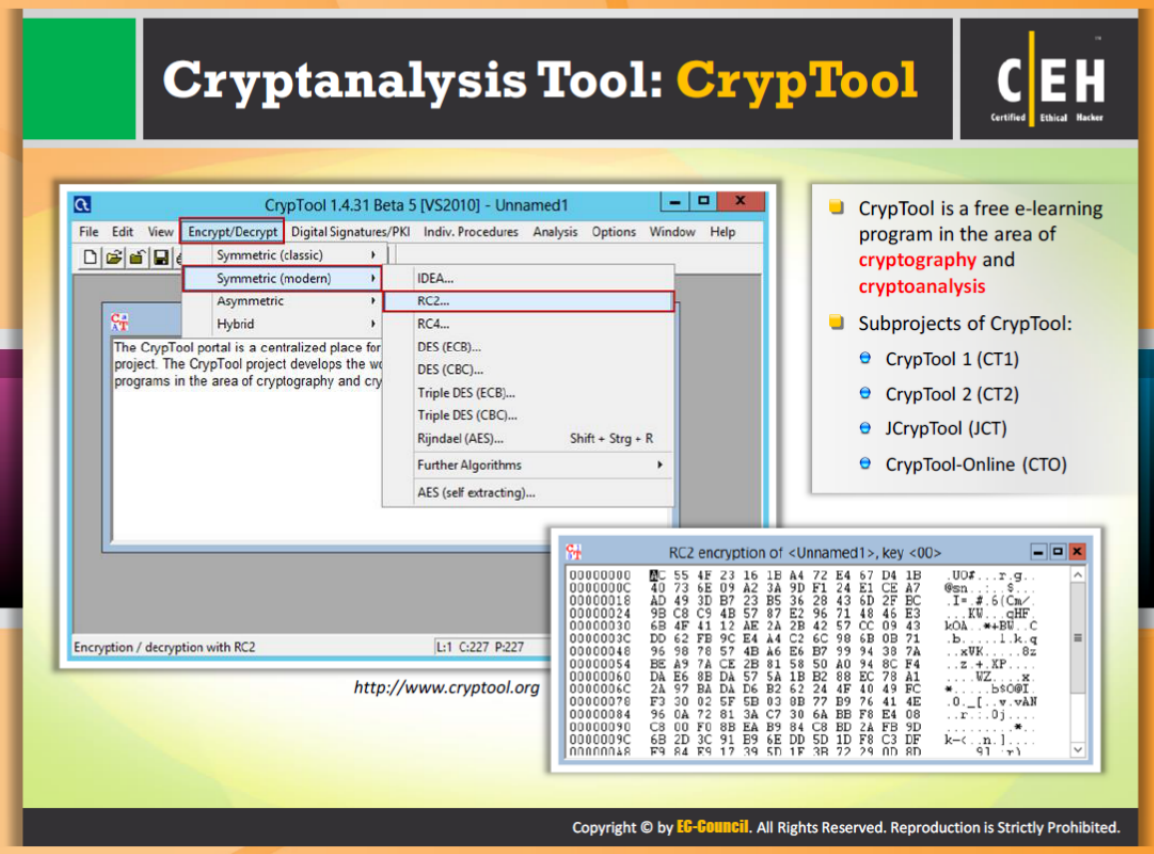



## Module Flow

So far, we have discussed all cryptography concepts, various cryptographic encryption algorithms, tools that help in cryptography, email and disk encryption, and how cryptographic mechanisms can be compromised. Now it's time to discuss cryptanalysis tools that help you in breaking old ciphers.

| | |
|---|---|
| Cryptography Concepts | Encryption Algorithms |
| Cryptography Tools | Public Key Infrastructure (PKI) |
| Email Encryption | Disk Encryption |
| Cryptography Attacks | Cryptanalysis Tools |

This section describes and lists cryptanalysis tools.




## Cryptanalysis Tool: CrypTool

Source: http://www.cryptool.org

The CrypTool project develops e-learning programs in the area of cryptography and cryptanalysis. It consists of four different subprojects: They are (CT1, CT2, JCT, CTO) related to the **CrypTool software** in various facets for different purposes.

- CrypTool 1 (CT1) was the first version of CrypTool. It was released in 1998 and allows to experiment with different cryptographic algorithms. CT 1 has two successors.
- CrypTool 2 (CT2) supports visual programming and execution of cascades of cryptographic procedures.
- JCrypTool (JCT) which is platform-independent.
- CrypTool-Online (CTO) was released in spring 2009. This tool allows trying out different algorithms in a browser/smartphone.
- Another subproject is the international crypto cipher challenge "MTC3," offering cryptographic riddles of different levels.

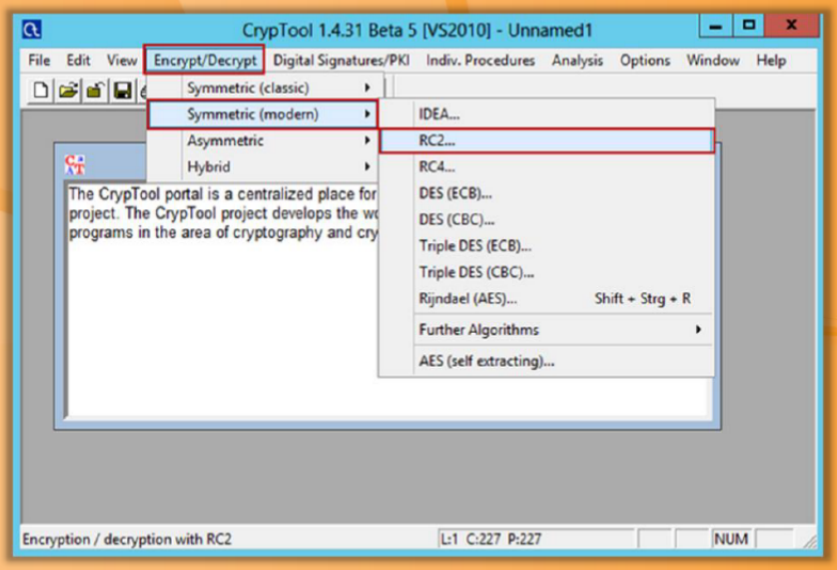



TABLE 19.24: CrypTool Screenshot

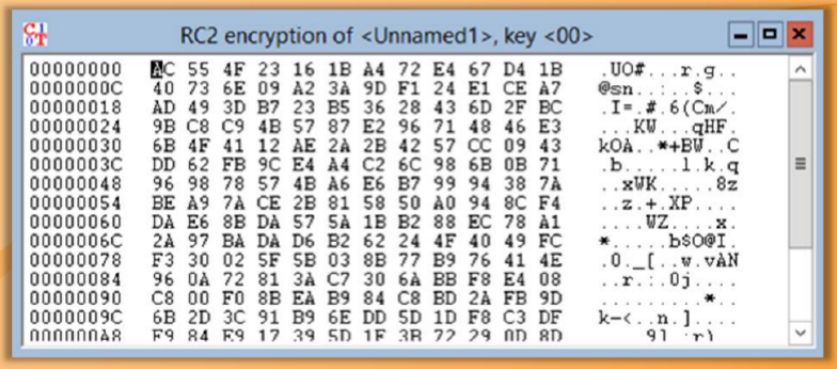
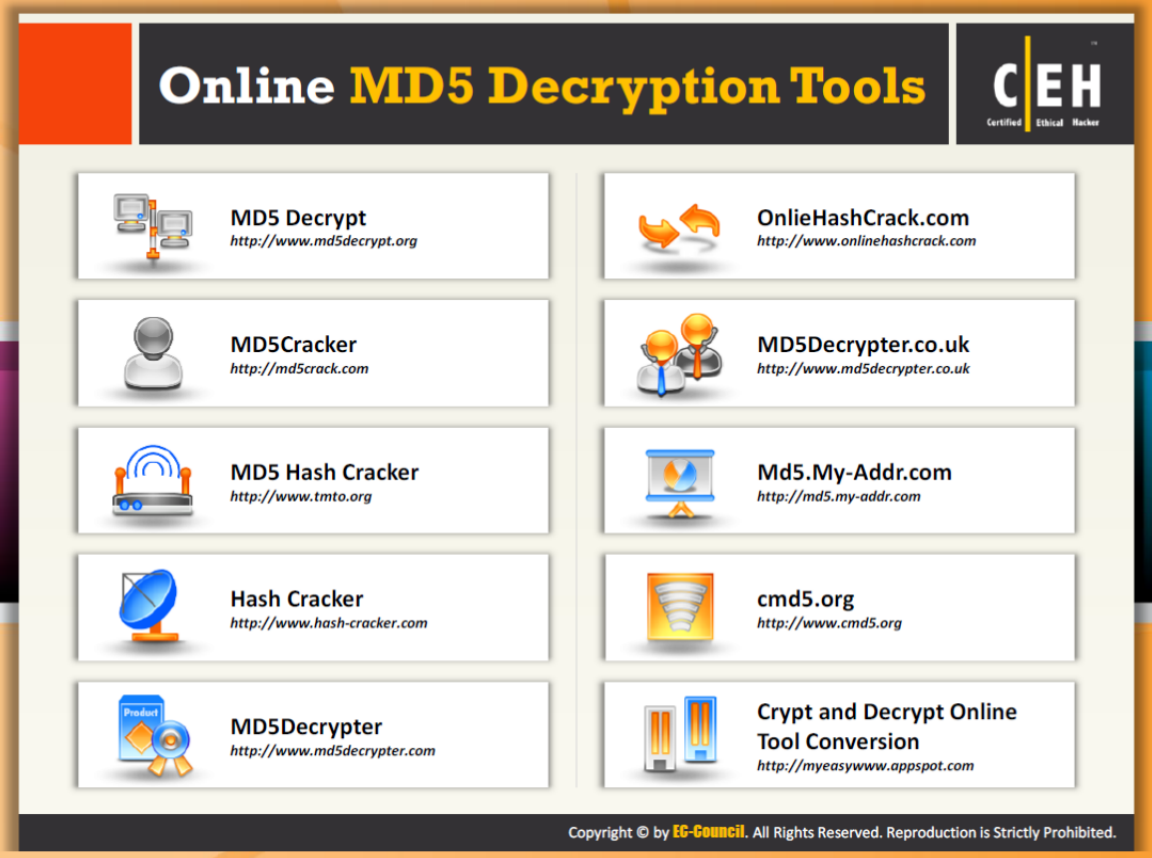



TABLE 19.24: RC2 encrytion Screenshot


## Cryptanalysis Tools

In addition to CrypTool, many tools that allow you to perform cryptanalysis are available:

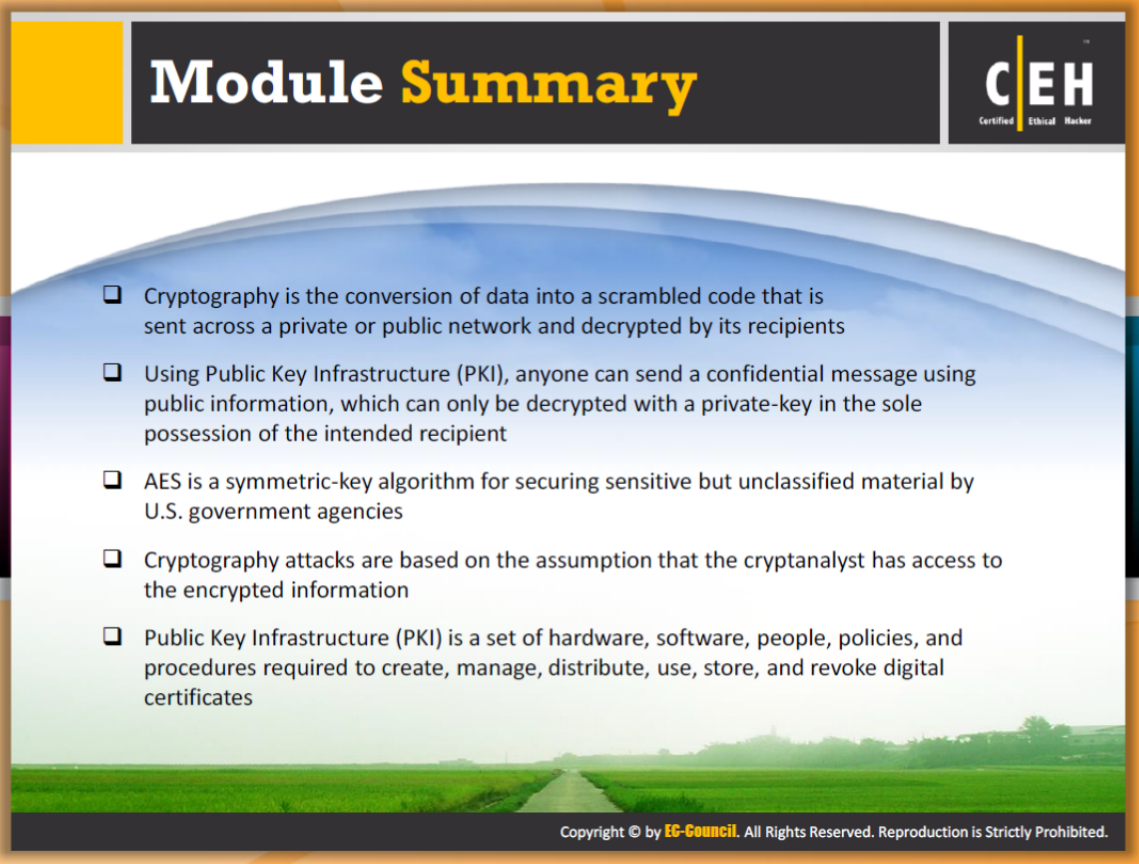
- CryptoBench available at http://www.addario.org
- JCrypTool available at http://www.cryptool.org
- Ganzúa available at http://ganzua.sourceforge.net
- Crank available at http://crank.sourceforge.net
- EverCrack available at http://evercrack.sourceforge.net
- AlphaPeeler available at http://alphapeeler.sourceforge.net
- Draft Crypto Analyzer available at http://www.literatecode.com
- Linear Hull Cryptanalysis of PRESENT available at http://www.ecrypt.eu.org
- Mediggo available at http://code.google.com
- SubCypher available at http://www.esclepiusllc.com




## Online MD5 Decryption Tools

Online MD5 decryption tools allow you to read the encrypted messages in clear text. All you need to do is sumbit the MD5 hash of the message that you want to read to an online MD5 decryptor. It decrypts the MD5 hash value and simply gives you the original message that has been encrypted. These tools eliminate the need for installing **MD5 decryptors**. Many online MD5 decryption tools are readily available:

- MD5 Decrypt available at http://www.md5decrypt.org
- MD5Cracker available at http://md5crack.com
- MD5 Hash Cracker available at http://www.tmto.org
- Hash Cracker available at http://www.hash-cracker.com
- MD5Decrypter available at http://www.md5decrypter.com
- OnlieHashCrack.com available at http://www.onlinehashcrack.com
- MD5Decrypter.co.uk available at http://www.md5decrypter.co.uk
- Md5.My-Addr.com available at http://md5.my-addr.com
- cmd5.org available at http://www.cmd5.org
- Crypt and Decrypt Online Tool Conversion available at http://myeasywww.appspot.com




## Module Summary

- Cryptography is the conversion of data into a scrambled code that is **decrypted** and sent across a private or public network.
- Using Public Key Infrastructure (PKI), anyone can send a confidential message using public information, which can only be decrypted with a private key in the sole possession of the intended recipient.
- RSA encryption is widely used and is a **de-facto encryption** standard.
- The MD5 algorithm is intended for digital signature applications, where a large file must be compressed securely before being encrypted.
- The **SHA algorithm** takes a message of arbitrary length as input and outputs a 160-bit message digest of the input.
- Secure Sockets Layer (SSL) is a protocol for transmitting private documents via the Internet.
- RC5 is a fast block cipher designed by RSA Security.