

Module 10

Cryptography and PKI

Module Objectives

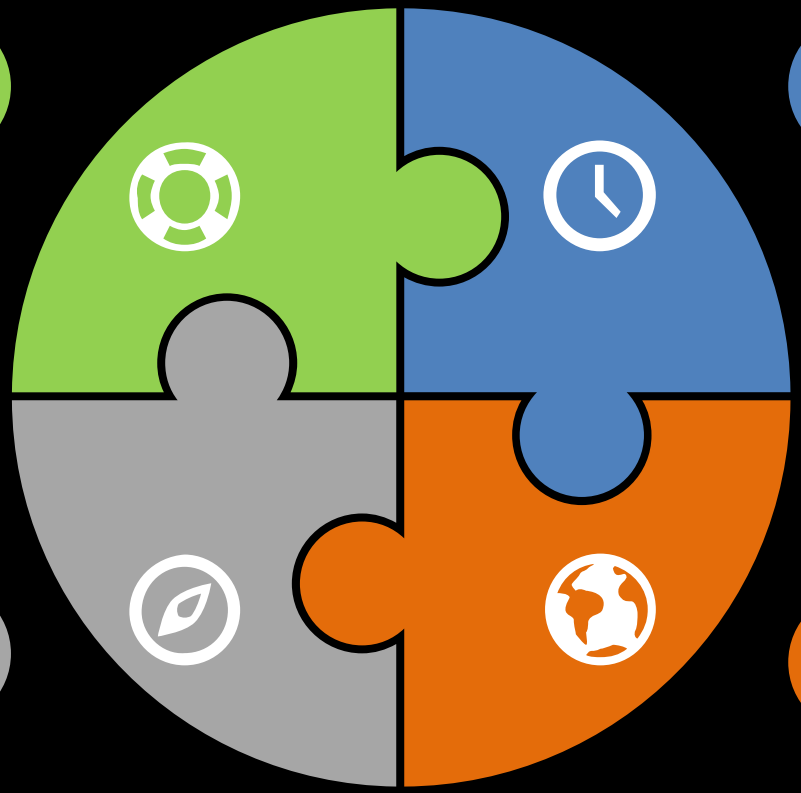
1. Understanding Cryptographic Techniques
2. Understanding the Different Encryption Algorithms
3. Understanding the Different Hashing Algorithms
4. Overview of Different Cryptography Tools and Hash Calculators
5. Understanding Public Key Infrastructure (PKI)
6. Understanding Digital Signatures and Digital Certificates



Module Flow

Discuss Cryptographic Techniques

01



02

03

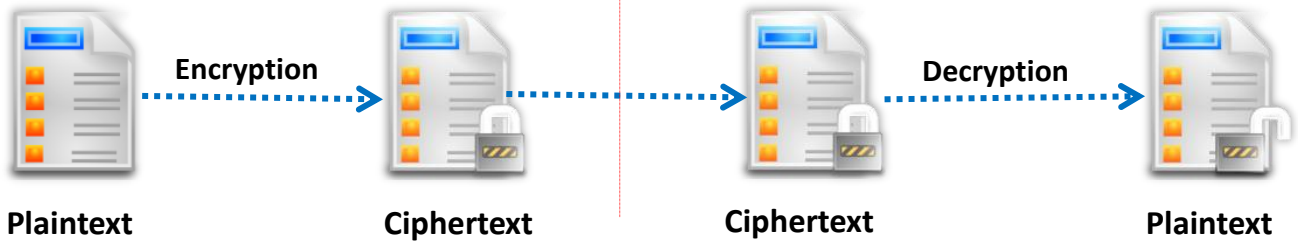
04

Cryptography

- Cryptography is the **conversion of data** into a scrambled code that is encrypted and sent across a private or public network
- Cryptography is used to protect confidential data, such as **email messages**, chat sessions, **web transactions**, personal data, **corporate data**, and e-commerce applications

Objectives of Cryptography

- **Confidentiality**
- **Authentication**
- **Integrity**
- **Nonrepudiation**





Encryption

- Encryption is a way of **protecting information** by transforming it in such a way that the resulting transformed form is unreadable to an unauthorized party
- To encrypt data, an encryption algorithm uses a **key** to perform a transformation on the data

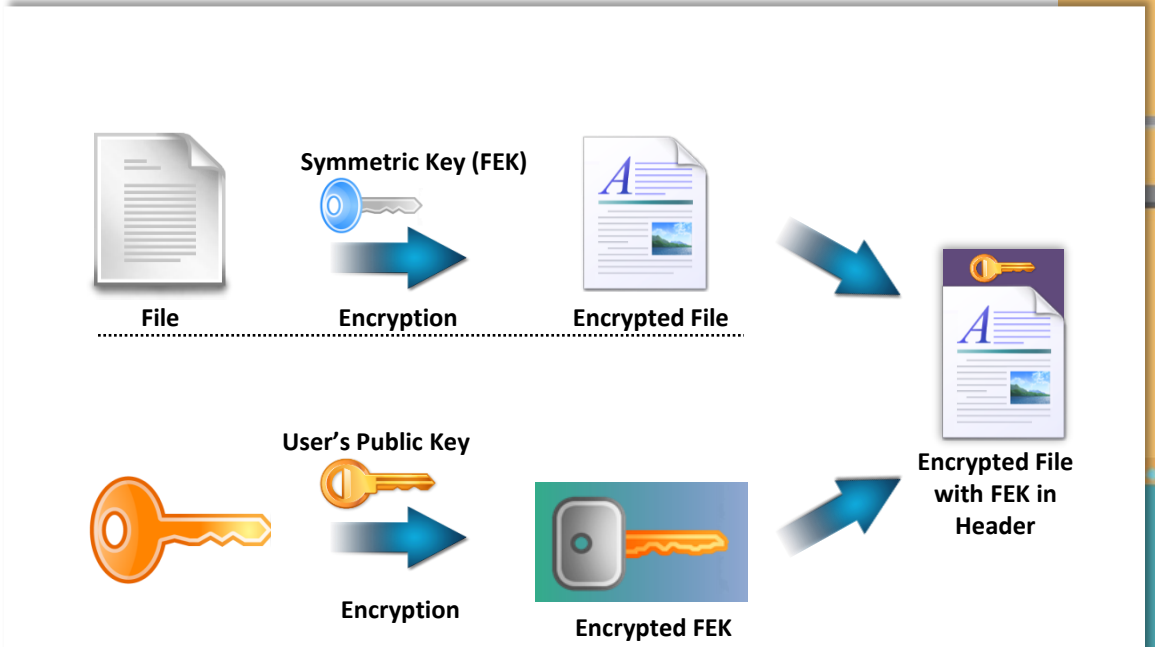


Types of Encryption

- Symmetric Encryption
- Asymmetric Encryption



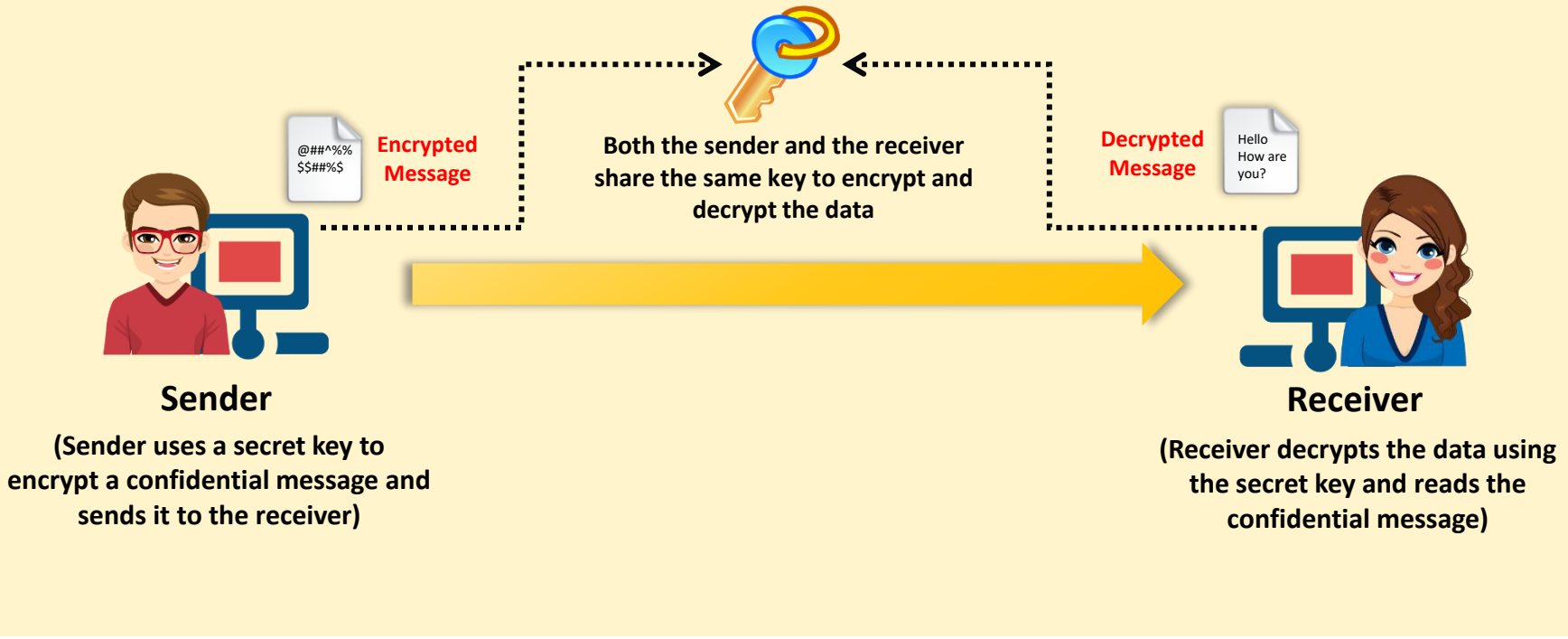
Encryption



Symmetric Encryption

- Symmetric encryption is the oldest cryptographic technique used for **encrypting digital data** in order to **ensure data confidentiality**
- It is called as symmetric encryption since a **single key** is used for encrypting and decrypting the data
- It is used to encrypt **large amounts of data**

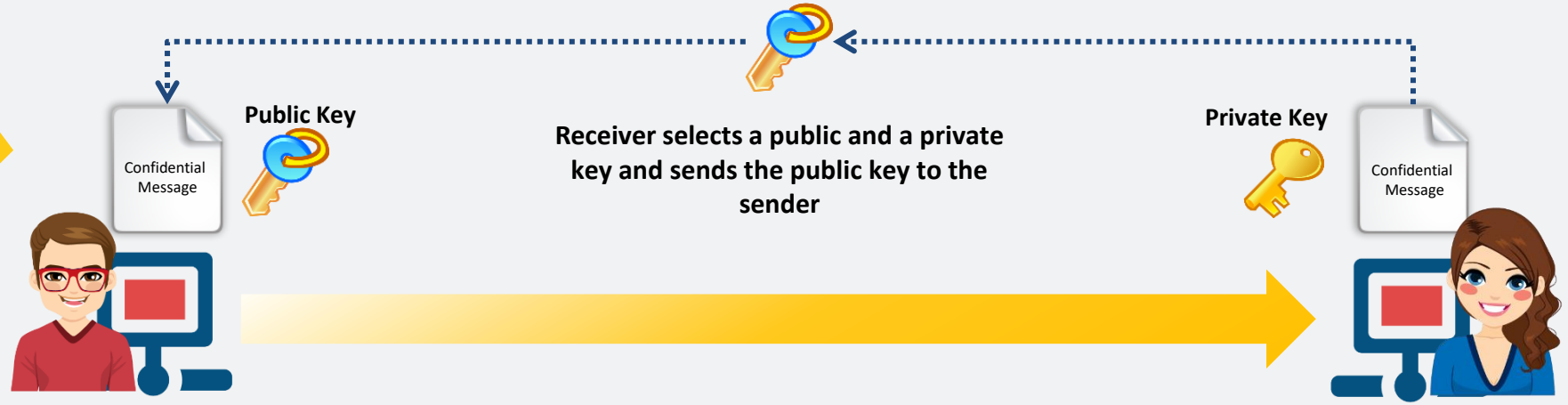




Asymmetric Encryption

- Unlike symmetric encryption, asymmetric encryption **uses two separate keys** to carry out encryption and decryption; one key, called the **public key**, is used for encrypting messages, whereas the second key, called the **private key**, is used for decrypting messages
- It is also called **public key encryption** and is used to **encrypt small amounts of data**





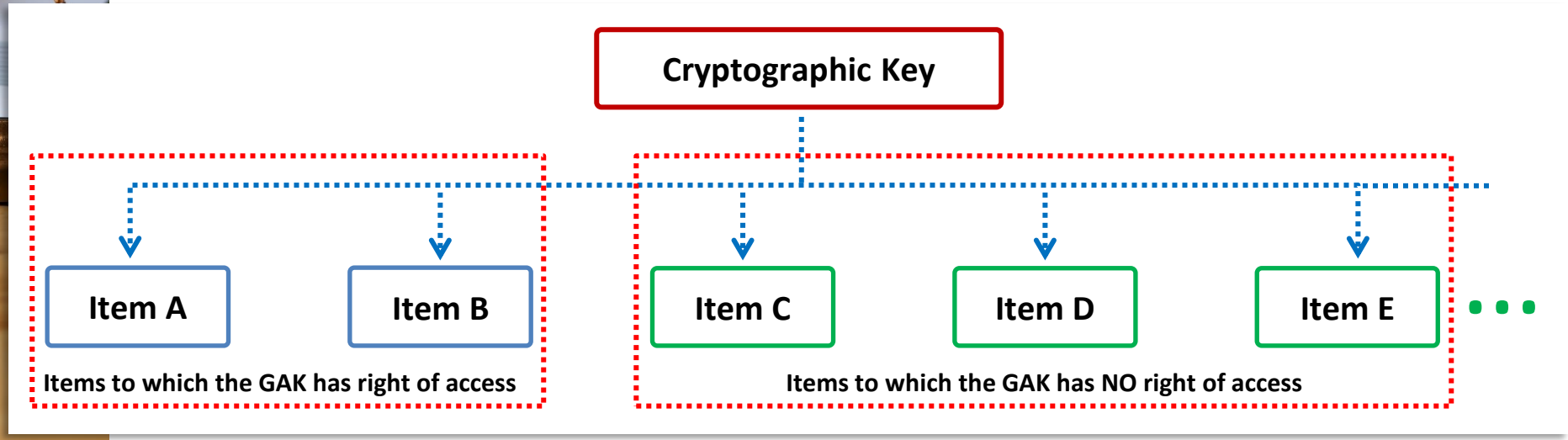
Government Access to Keys (GAK)



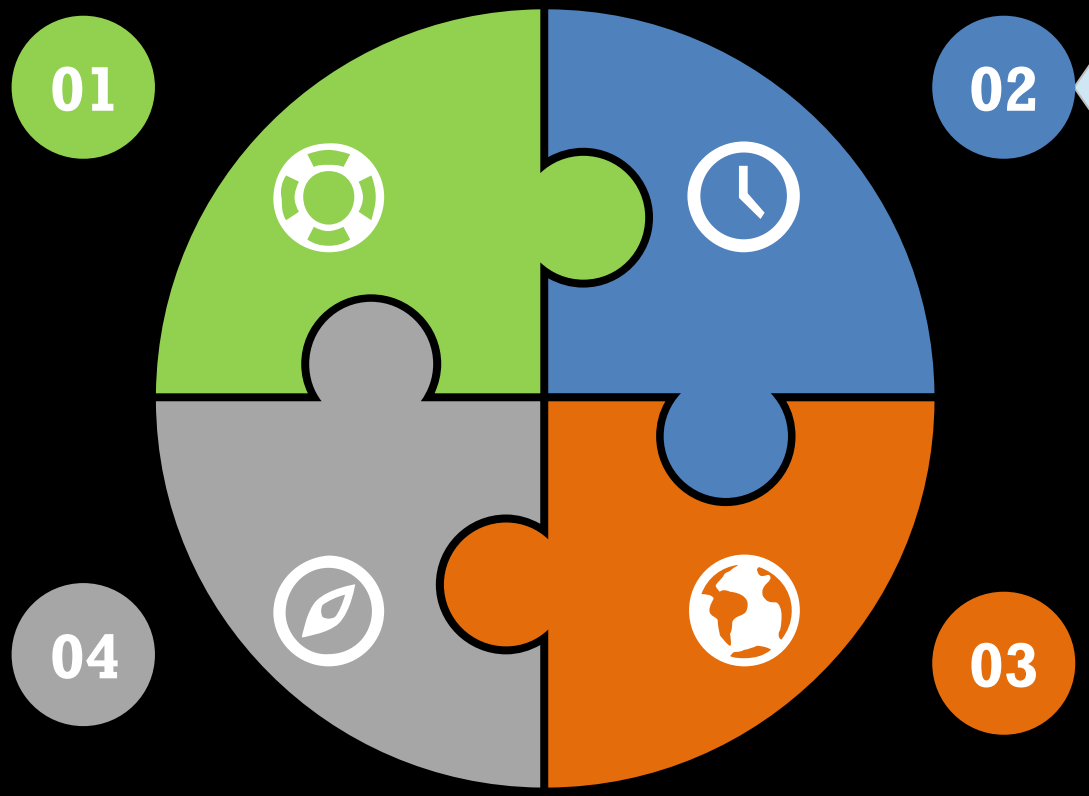
GAK means that software companies will give **copies of all keys** (or at least a sufficient proportion of each key that the remainder could be cracked) to the government

The government promises that they will hold on to the keys in a **secure manner** and will only use them when a **court issues a warrant** to do so

To the government, this is similar to the **ability to wiretapping phones**



Module Flow



Ciphers



Types of ciphers

Classical Ciphers

Substitution cipher

A block of plaintext is replaced with ciphertext

Transposition cipher

The letters of the plaintext are shifted about to form the cryptogram

Modern Ciphers

Based on the type of key used

Private Key

Same key is used for encryption and decryption

Public Key

Two different keys are used for encryption and decryption

Based on the type of input data

Block Cipher

Encrypts blocks of data of fixed size

Stream Cipher

Encrypts continuous streams of data

Ciphers are **algorithms** used to encrypt or decrypt the data

Data Encryption Standard (DES)



- DES is designed to **encipher** and **decipher** blocks of data consisting of **64 bits** under control of a 56-bit key
- DES is the **archetypal block cipher** — an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length
- Due to the **inherent weakness** of DES with today's technologies, some organizations triple repeat the process (3DES) for added strength until they can afford to update their equipment to AES capabilities

Advanced Encryption Standard (AES)



- AES is a **symmetric-key** algorithm used by the US government agencies to secure sensitive but unclassified material
- AES is an **iterated block cipher** that works by repeating the same operation **multiple** times
- It has a **128-bit** block size with key sizes of 128, 192, and 256 bits for AES-128, AES-192, and AES-256, respectively



RC4, RC5, and RC6 Algorithms

RC4

- A variable key size **symmetric key stream cipher** with byte-oriented operations and is based on the use of a random permutation

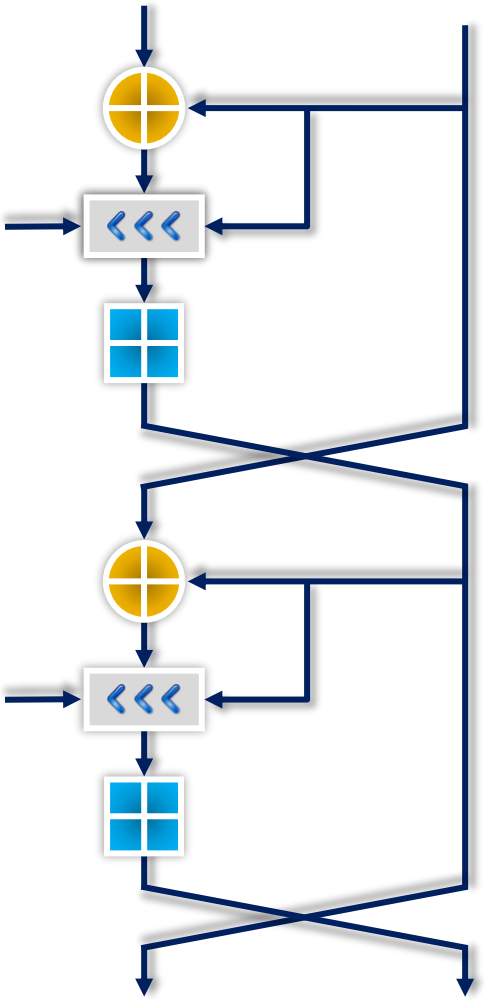
RC5

- It is a **parameterized algorithm** with a variable block size, variable key size, and variable number of rounds. The key size is **128 bits**

RC6

- RC6 is a **symmetric key block cipher** derived from RC5 with two additional features:
  - ✓ **integer multiplication**
  - ✓ **four 4-bit working registers** (RC5 uses two 2-bit registers)

RC5 Algorithm



Digital Signature Algorithm (DSA) and Rivest Shamir Adleman (RSA)

Digital Signature Algorithm (DSA)

- Federal Information Processing Standard (FIPS) 186-2 specifies the digital signature algorithm (DSA) that can be used in the **generation and verification of digital signatures** for sensitive, unclassified applications

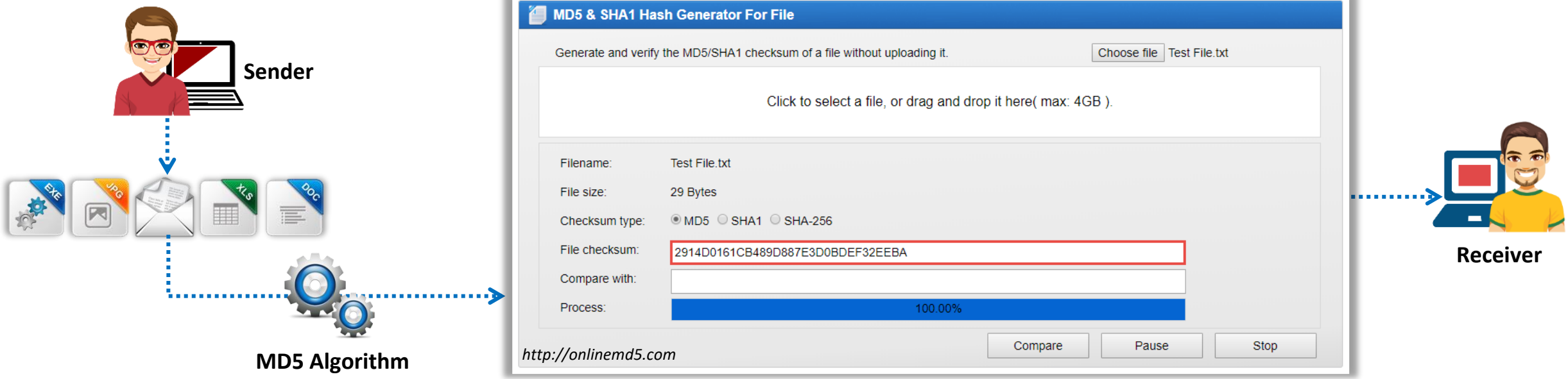


Rivest-Shamir-Adleman (RSA)

- RSA is an **Internet encryption and authentication system** that uses an algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman
- It is commonly used and is one of the **de-facto encryption standards**

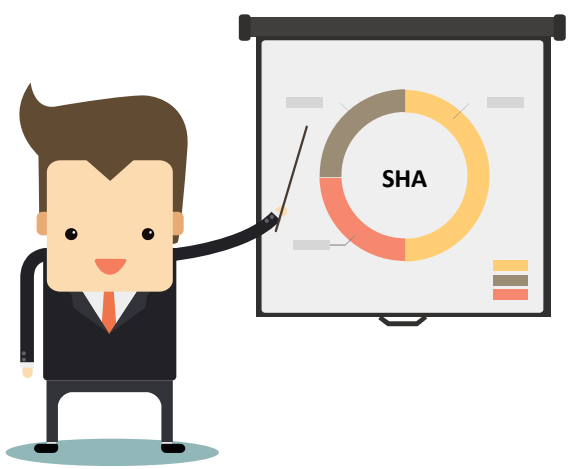
MD5 and MD6

- The MD5 algorithm takes a message of **arbitrary length** as the input and then outputs a **128-bit fingerprint** or message digest of the input
- MD5 is not collision resistant; use of the latest algorithms, such as **MD6**, **SHA-2** and **SHA-3**, is recommended
- **MD6** uses a Merkle tree-like structure to allow for immense parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks
- MD5 and MD6 are deployed for digital signature applications, file integrity checking, and storing passwords



Secure Hashing Algorithm (SHA)

- This algorithm generates a cryptographically secure one-way hash; it was published by the **National Institute of Standards and Technology** as a **US Federal Information Processing Standard**



SHA-1

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

SHA-2

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

SHA-3

- SHA-3 uses the **sponge construction**, in which message blocks are **XORed** into the initial bits of the state, which is then invertibly permuted

HMAC

The hash-based message authentication code (HMAC) is a type of **message authentication code** (MAC) that makes use of a **cryptographic key** in combination with a cryptographic hash function

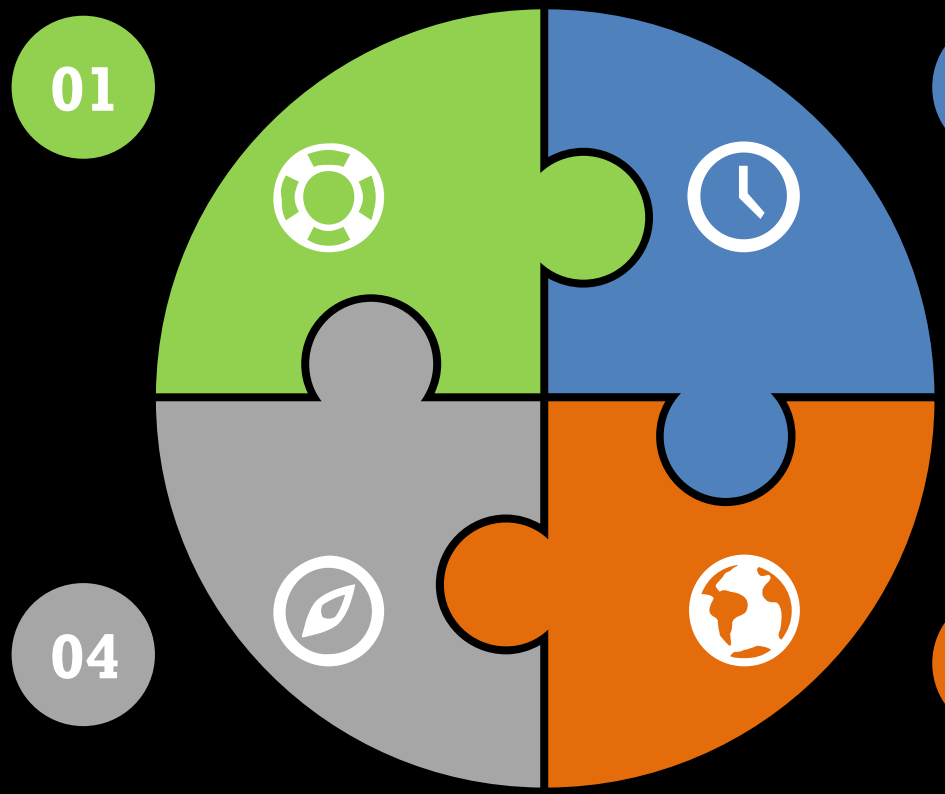
This algorithm includes an embedded hash function such as **SHA-1** or **MD5**

The strength of HMAC depends on the **embedded hash function**, key size, and the size of the hash output

As the HMAC executes the underlying hash function twice, it protects the data from various **length extension attacks**



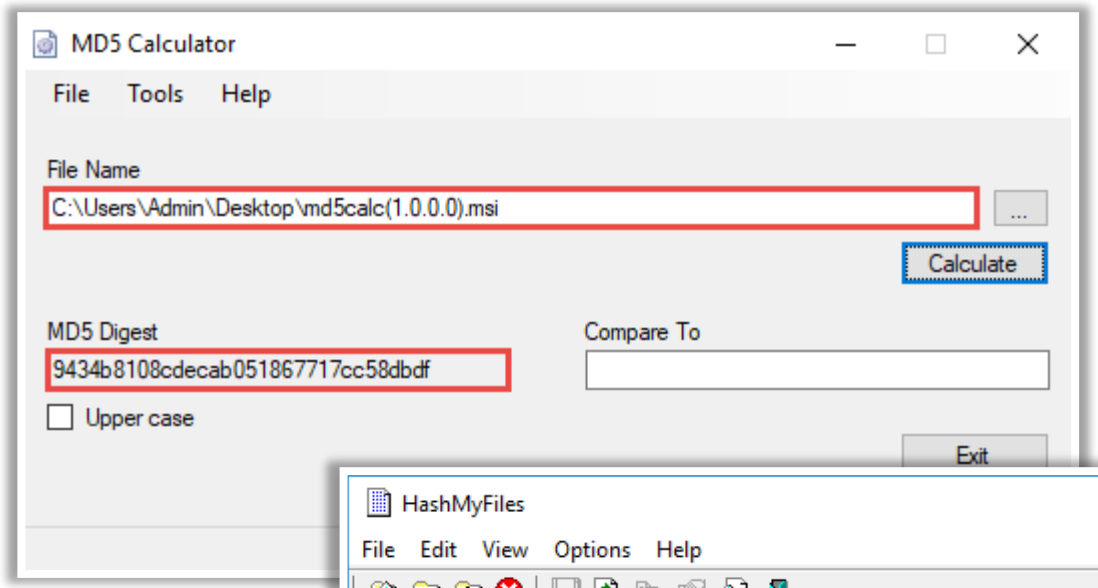
Module Flow



Discuss Various Cryptography Tools

MD5 and MD6 Hash Calculators

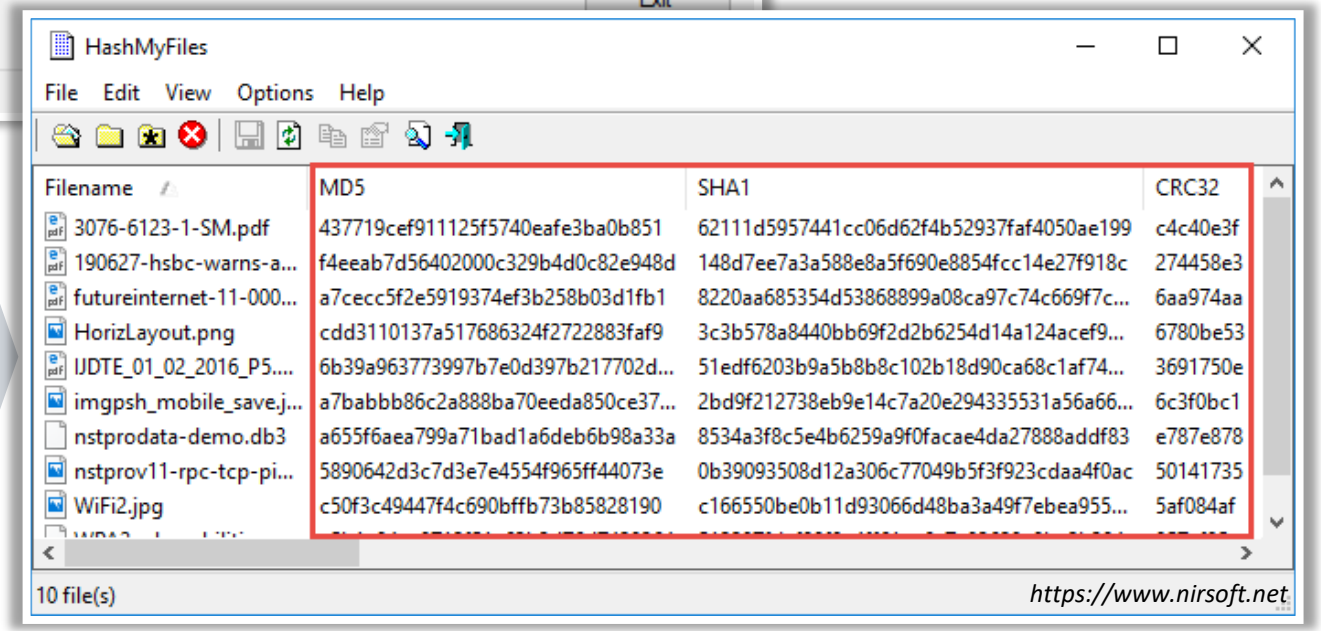
MD5 Calculator



https://www.bullzip.com



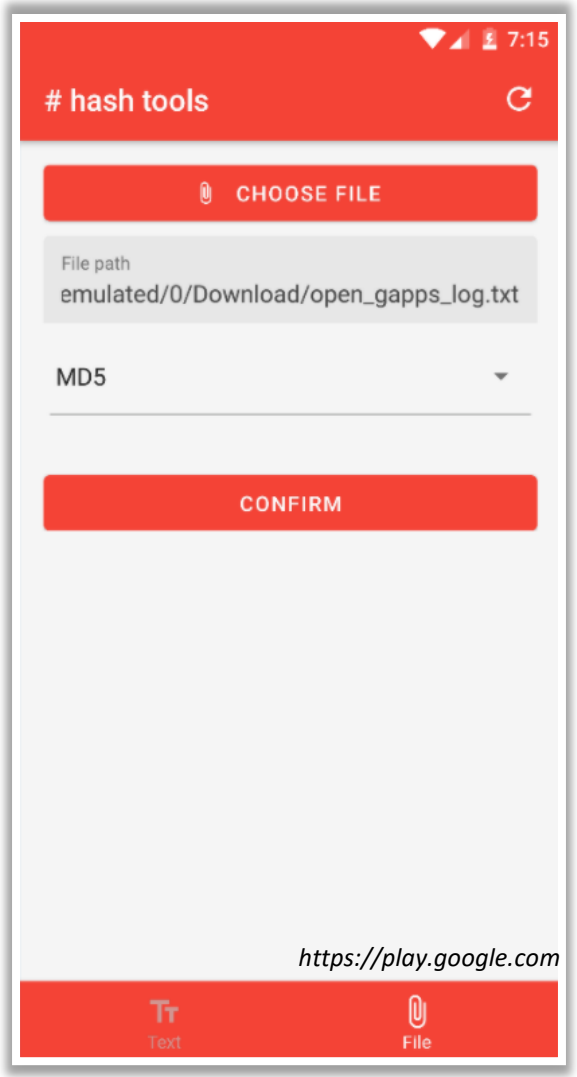
HashMy Files



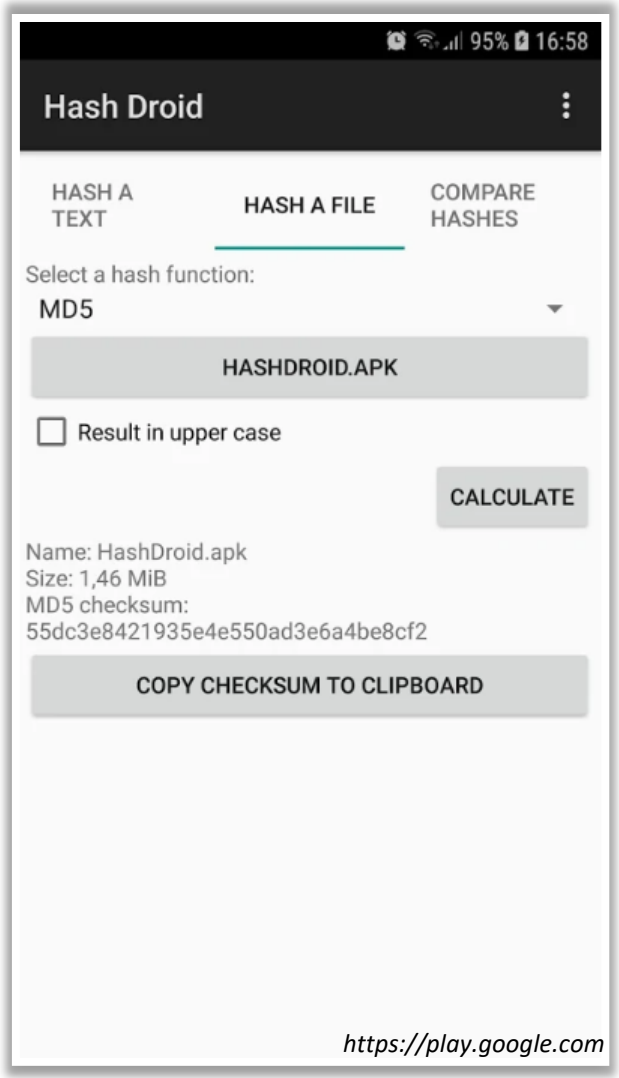
- **MD6 Hash Generator**
  https://www.browserling.com
- **All Hash Generator**
  https://www.browserling.com
- **MD6 Hash Generator**
  https://convert-tool.com
- **md5 hash generator**
  https://onlinehashtools.com
- **HashCalc**
  https://www.slavasoft.com

Hash Calculators for Mobile

Hash Tools



Hash Droid

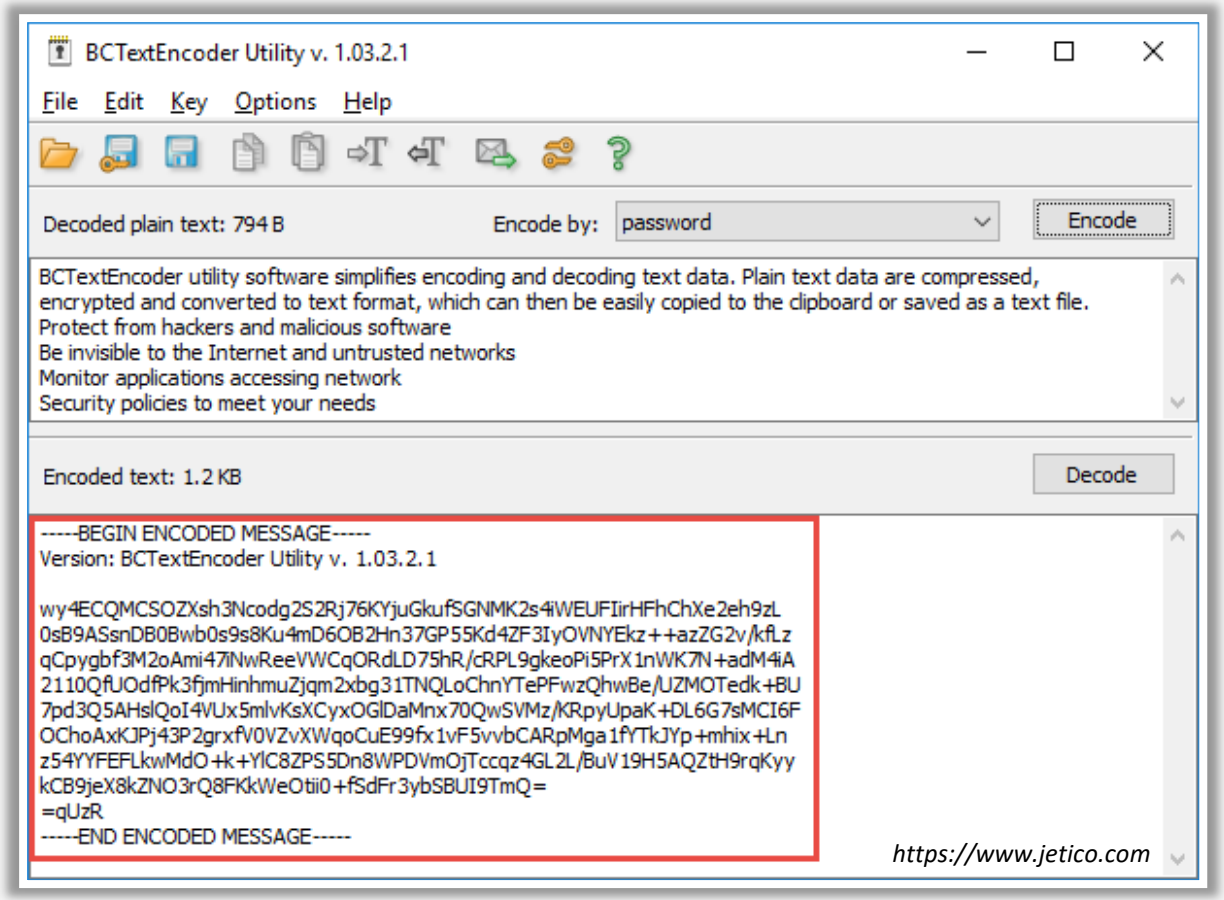


- **Hash Checker**
  https://play.google.com
- **Hashr - Hash & Checksum Calculator**
  https://play.google.com
- **Hash Calc**
  https://play.google.com
- **Hash Generator - Checksum Calculator**
  https://play.google.com
- **Hash Smart Checker**
  https://play.google.com

Cryptography Tools

BCTextEncoder

- Encrypts **confidential text** in your **message**
- Uses strong symmetric and public-key algorithms for **data encryption**



AxCrypt
https://axcrypt.net

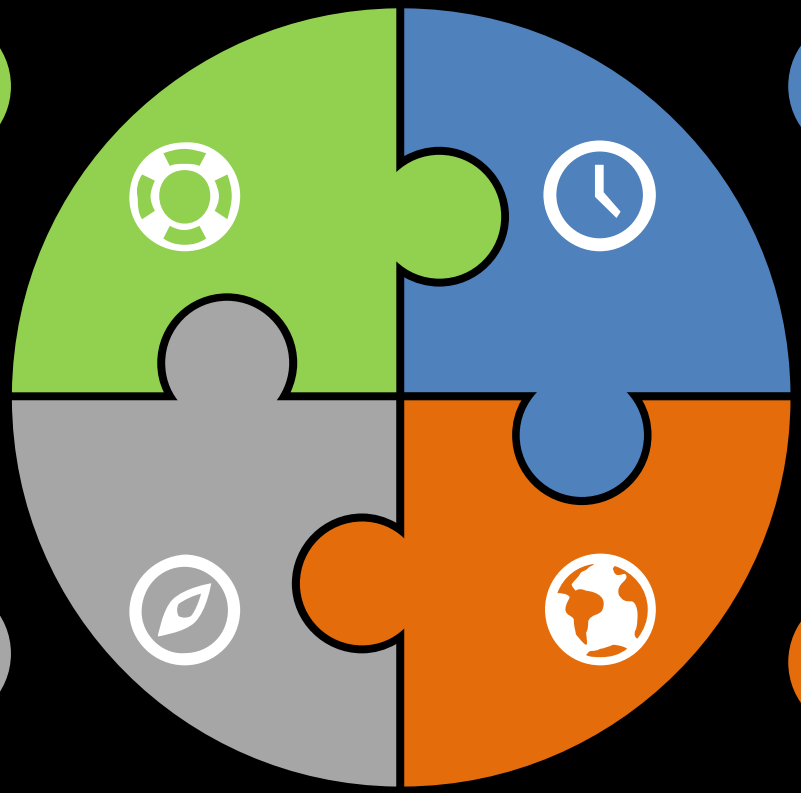
Microsoft Cryptography Tools
https://docs.microsoft.com

Concealer
https://www.belightsoft.com

CryptoForge
https://www.cryptoforge.com

Cyphertop
https://cyphertop.com

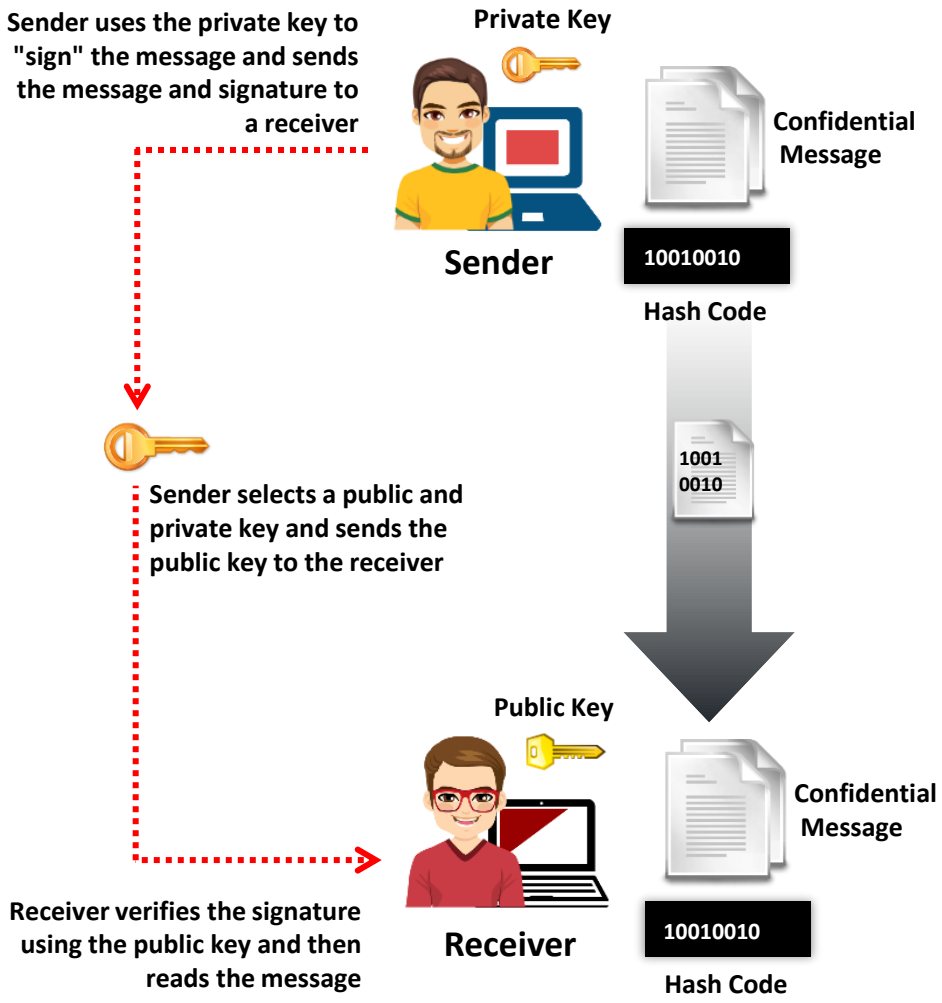
Module Flow



Discuss Public Key Infrastructure (PKI)

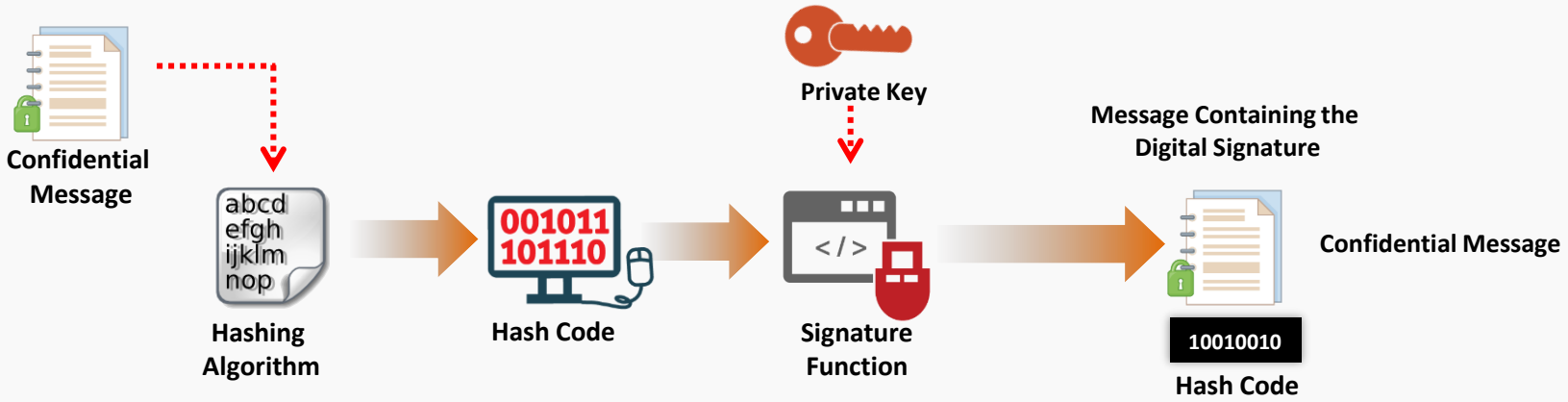
Digital Signature

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

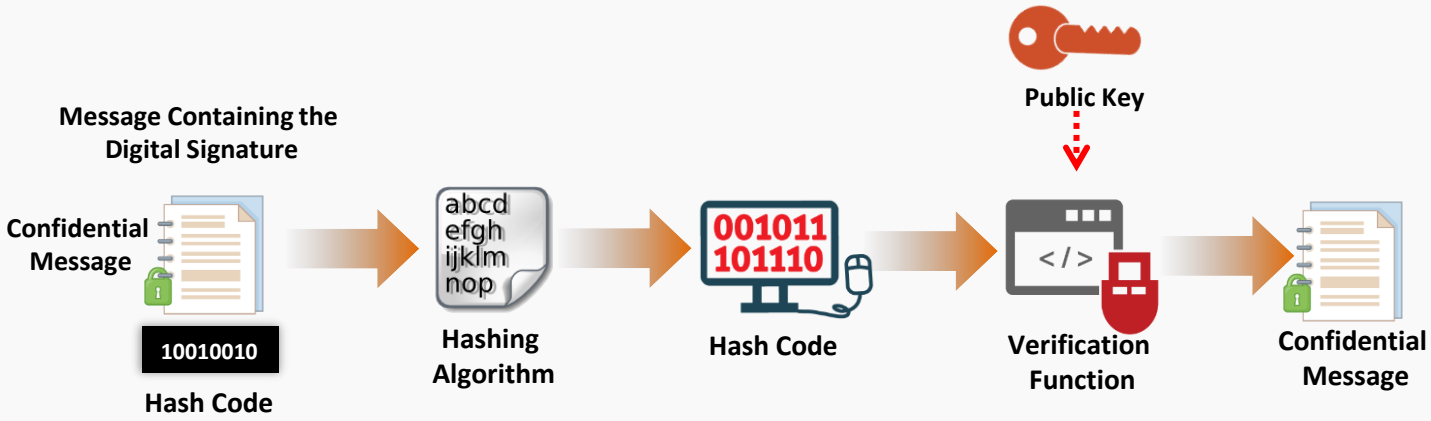


Digital Signature (Cont'd)

Creating a digital signature at the sender side

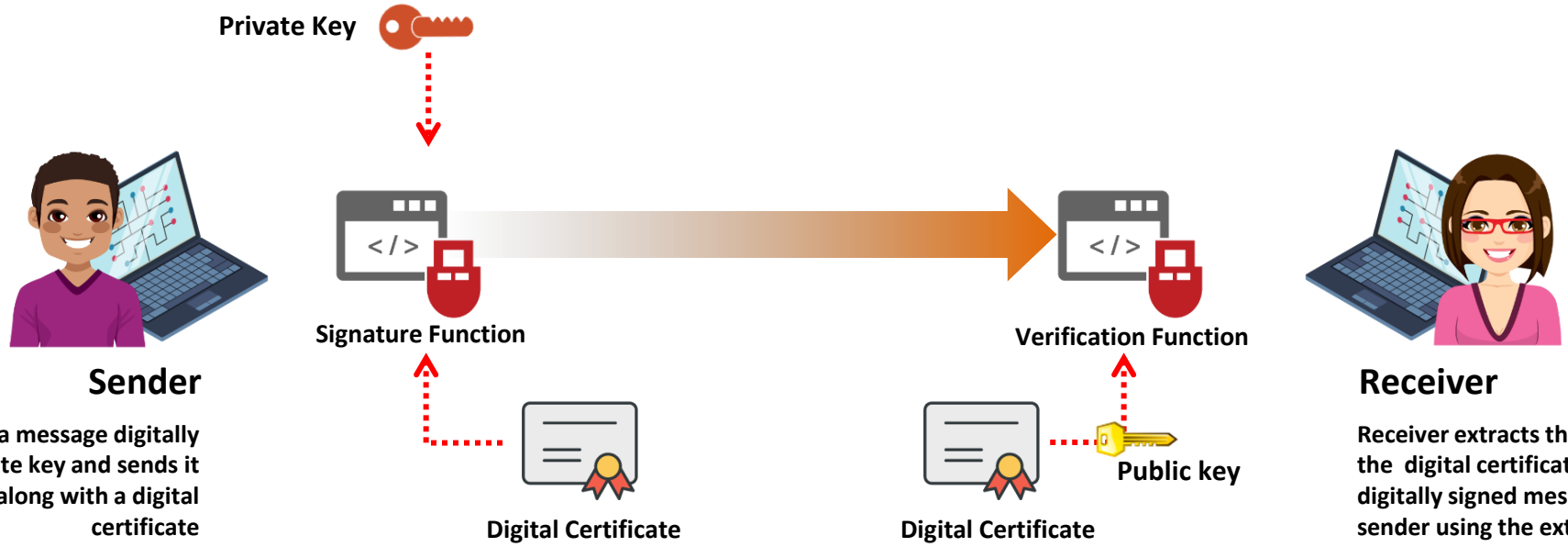


Verifying a digital signature at the recipient side



Digital Certificates

- The digital certificates are used for dealing with the security concerns regarding **transmission of public keys securely** to the receiver in the digital signature
- A **trusted intermediary solution** is used for securing the public keys, where the public key is bound with the name of its owner
- Owners of the public key need to acquire their public keys certified from the intermediary; the intermediary then issues certificates called **digital certificates** to the owners, which they can use to send the public key to a number of users



Digital Certificates (Cont'd)

Digital Certificate Attributes

- **Serial number**: Represents the **unique certificate identity**
- **Subject**: Represents the **owner of the certificate** which may be a person or an organization
- **Signature algorithm**: States **the name of the algorithm** used for creating the signature
- **Key-usage**: Specifies the **purpose of the public key**, whether it should be used for encryption, signature verification, or both
- **Public key**: Used for **encrypting** a message or **verifying** the signature of the owner
- **Issuer**: Provides the **identity of the intermediary** who issued the certificate
- **Valid from**: Denotes the date **from which** the certificate is valid
- **Valid to**: Denotes the date **till which** the certificate is valid
- **Thumbprint algorithm**: Specifies the **hashing algorithm** used for digital signatures
- **Thumbprint**: Specifies the **hash value** for the certificate, which is used for verifying the certificate's integrity

Public Key Infrastructure (PKI)

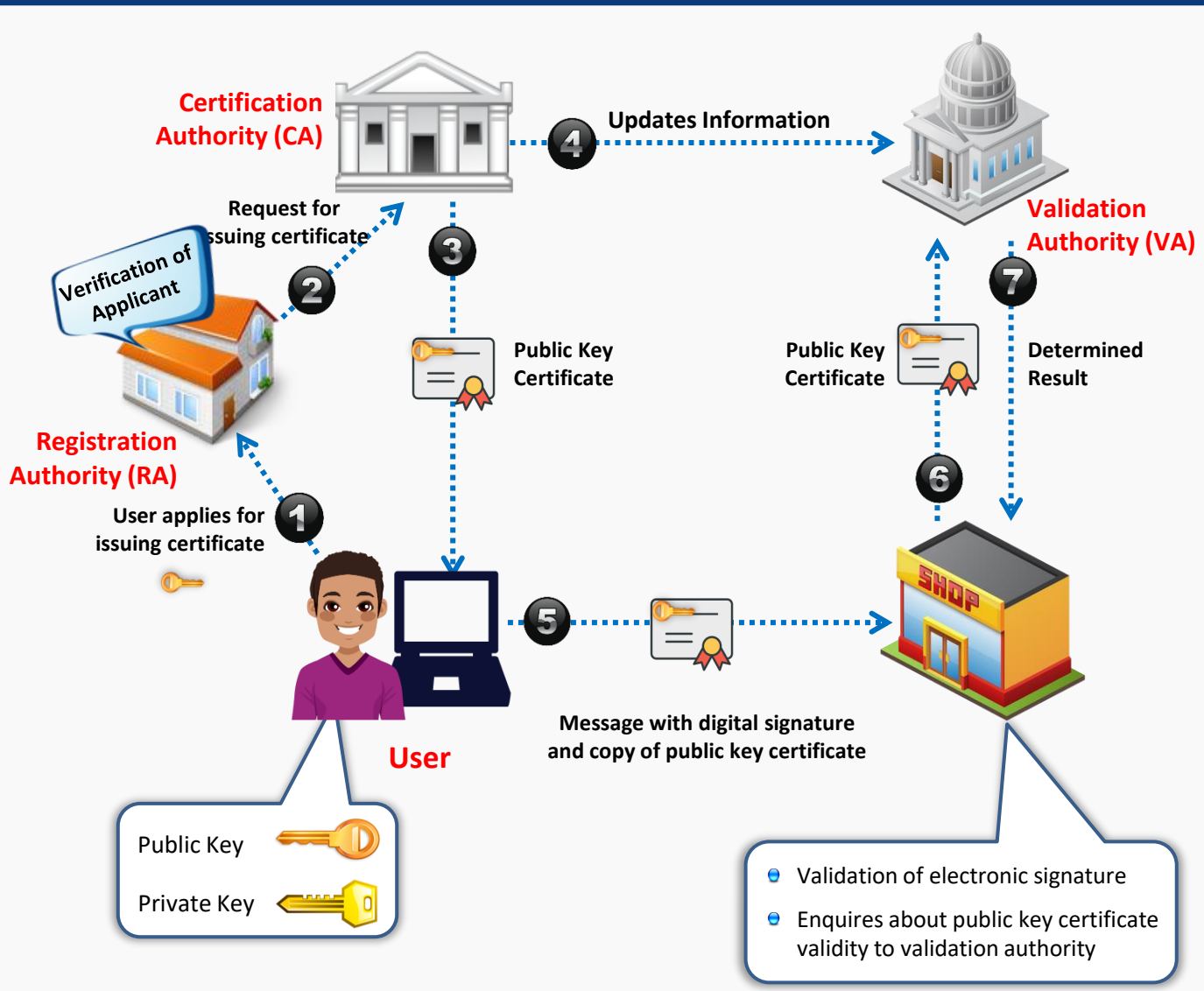
- A public key infrastructure (PKI) is a set of hardware, software, people, policies, and procedures required for creating, managing, distributing, using, storing, and revoking **digital certificates**



Components of a PKI

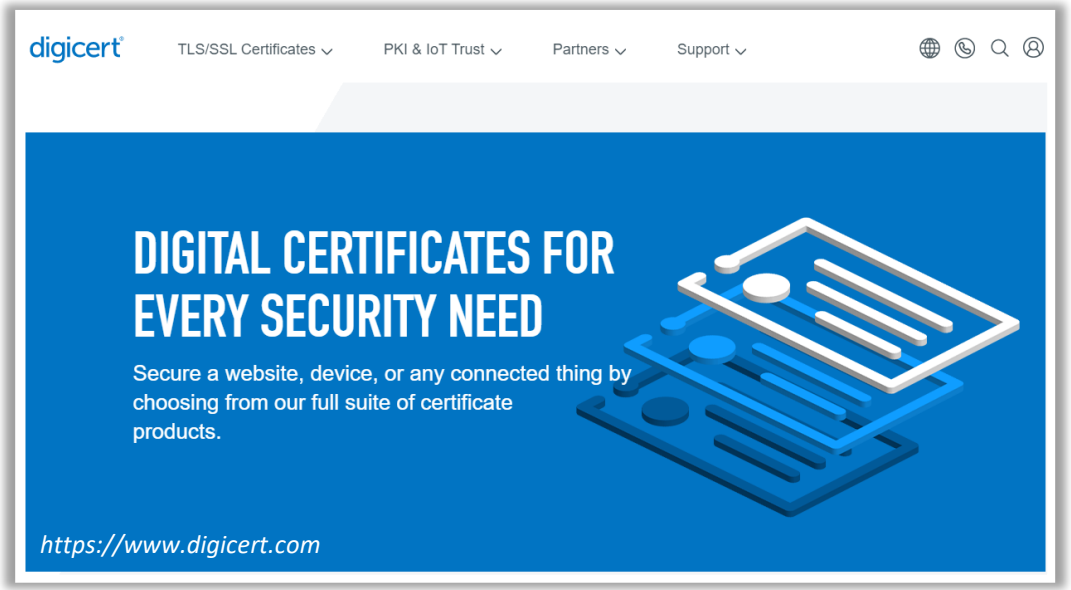
- **A certificate authority (CA) that issues and verifies digital certificates**
- **A certificate management system for generation, distribution, storage, and verification of certificates**
- **A registration authority (RA) that acts as the verifier for the CA**
- **One or more directories where the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

Public Key Infrastructure (PKI) (Cont'd)

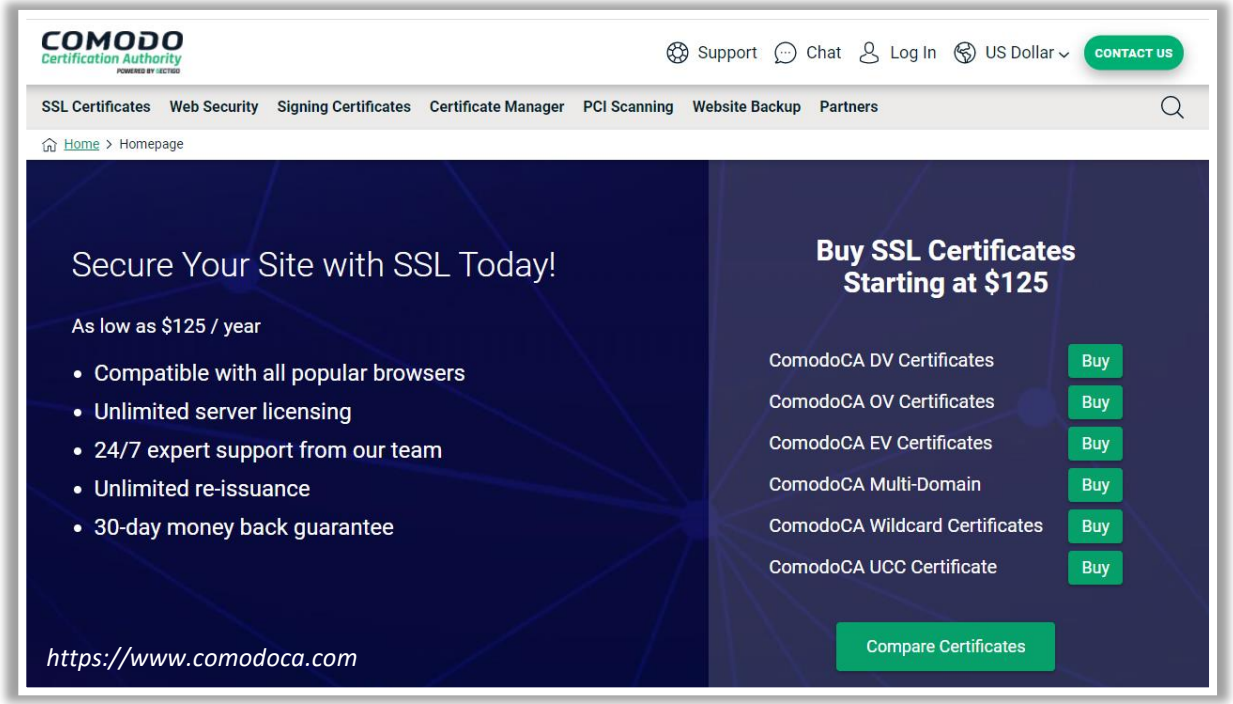


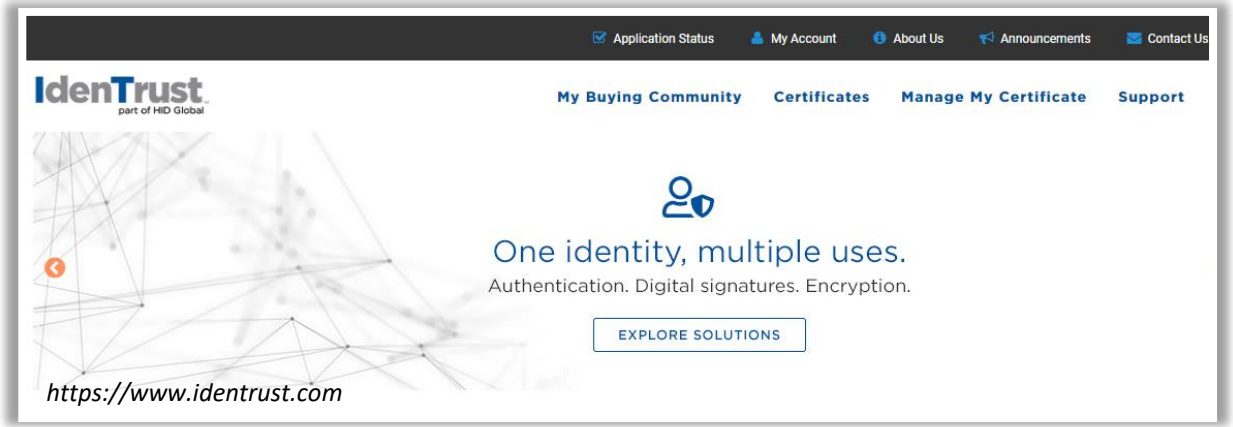


Certification Authorities









Module Summary



- This module has discussed cryptographic techniques, different encryption algorithms, and hashing algorithms
- It has discussed different cryptography tools and hash calculators
- It has also discussed the public key infrastructure (PKI) concepts
- Finally, this module ended with an overview of digital signatures and digital certificates
- In the next module, we will discuss on data security in detail