Cryptography Features and Objectives

Reading

35m

Status | Incomplete

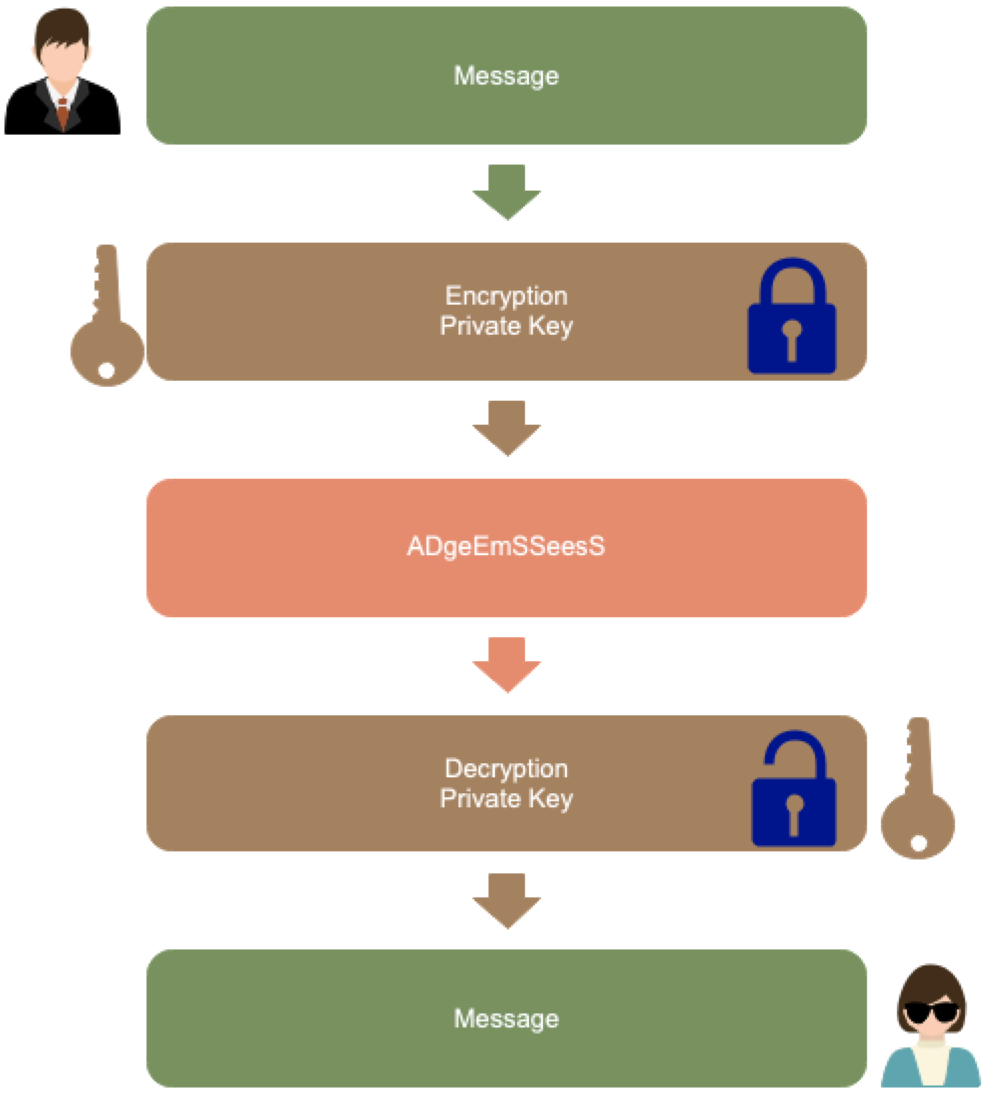
Introduction

The purpose of cryptography is to transform a message into coded text, ensuring the confidentiality of the information is contained. The primary use of cryptography is for secure exchange of sensitive information, such as usernames and/or passwords.

Cryptographic methods are the techniques and algorithms used to encrypt and decrypt information in order to secure it from unauthorized access or modification. These methods can be divided into two main categories: symmetric cryptography and asymmetric cryptography. Symmetric cryptography uses a shared secret key to encrypt and decrypt data, while asymmetric cryptography uses a pair of public and private keys to perform the same functions. Understanding the strengths and weaknesses of these different cryptographic methods is essential for new cyber security professionals, as it enables them to select and implement the appropriate cryptographic techniques to protect their organization's data and assets

Symmetric Cryptography

Symmetric cryptography is one of the most widely used forms of cryptography. As you can see in the image, the encrypted message is encrypted with an algorithm, which generates a new message, called the encoded message, from the combination of the original message and a secret key. After, the message is decrypted using the same key.



This coded message can be sent or stored, and only those with the same private key can decode it. An analogy to the real world, symmetric cryptography is similar to putting a message inside a safe and closing it with a padlock. Only people with the keys to the padlock can access the message.

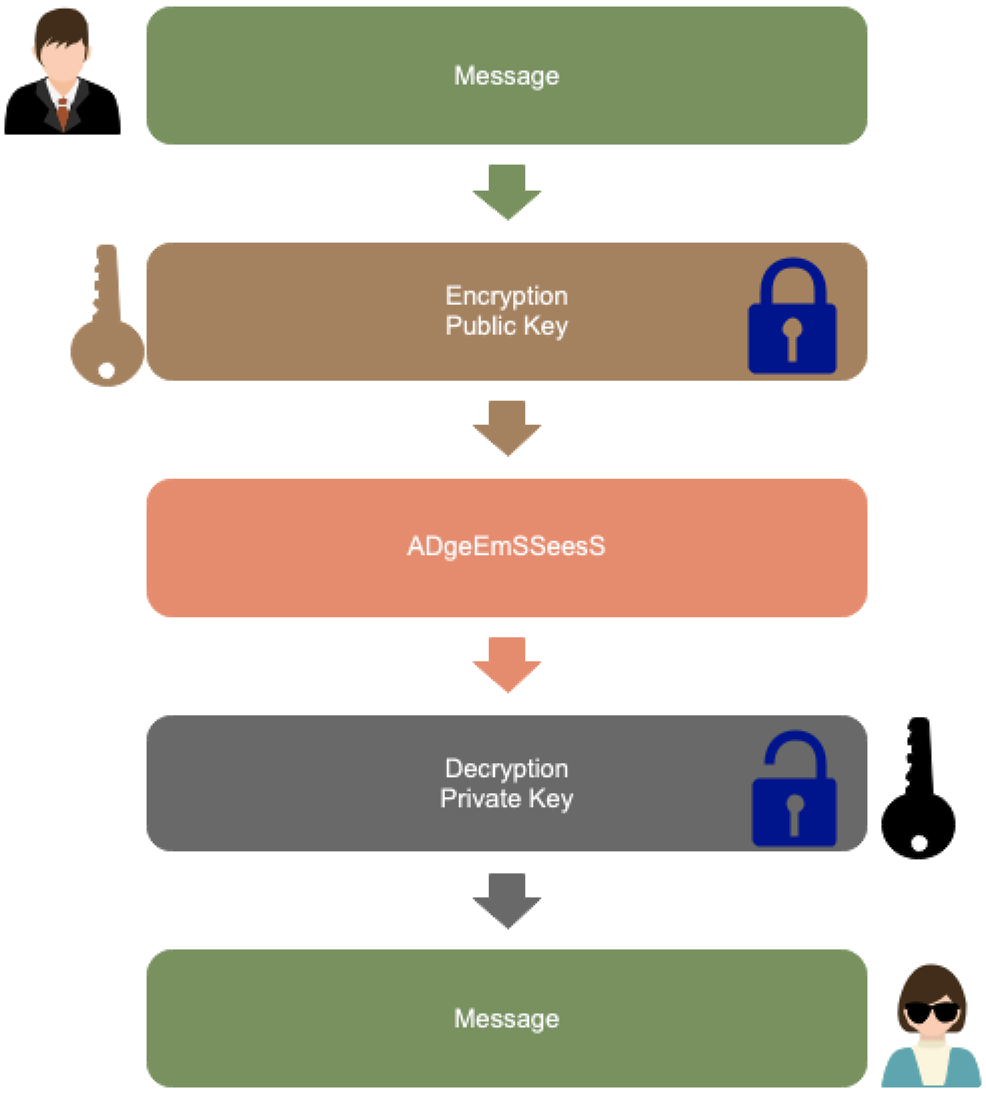
The cost of encrypting and decrypting symmetric keys is low in the computational world. The big problem in this type of cryptography is developing safe ways to transmit these keys. If the key transmission is not secure, the security of the entire cryptography system is compromised. Some public key infrastructure (PKI) systems implement mechanisms that allow the secure exchange of these keys.

The Advanced Cryptography Standard (AES) and Blowfish are examples of symmetric cryptography algorithms.

Asymmetric Cryptography

Asymmetric cryptography or public key cryptography gets around the big problem found in symmetric cryptography. This cryptography method uses a pair of distinct keys, public and private keys, to promote encryption and decryption.

The private key must be accessible only to the user who owns it. Every message encrypted with the private key can only be decrypted with its respective public key. And every message encoded with the public key can only be decrypted using its corresponding private key. This way, if user A wants to send an encrypted message to user B, they can encode it with user B's public key and send it to user B. Only user B, who has user B's private key, can read the content of the message sent by A.



As the name implies, the public key can be disclosed to all entities with which the key owner wants to relate. Meaning, it doesn't need to be kept secret. Asymmetric cryptography algorithms are based on solving very complex problems that may involve elliptic curves and the factorization of vast prime numbers.

Due to this, the computational cost required for the encryption and decryption of asymmetric keys is high. The RSA and the Digital Signature Algorithm (DSA) are among the most used asymmetric cryptography algorithms.

Important Security Aspects

The main objective of a public key infrastructure is to allow security within insecure environments and provide means to identify the origin of exchanged information. This infrastructure uses a series of buttons to promote the operation of the following essential security points:

- Authentication:- consists of verifying that a given user is really who they claim to be. An attacker must not be able to impersonate another user. In the real world, it can be exercised using an identity document, or a driver's license, for example.
- Confidentiality: means ensuring that users who do not have access permissions cannot read the content of exchanged messages.
- Integrity: allows the user to verify that the received message has not been modified during transmission. In the public key infrastructure, integrity is closely related to the digital signature of documents.
- Non-repudiation: guarantees that if a user sends a message, they will have no way of denying the sending of this message later.
- Access control: the ability to allow or deny the use of a service to a given entity, such as a user or a company.

Cryptographic Hash

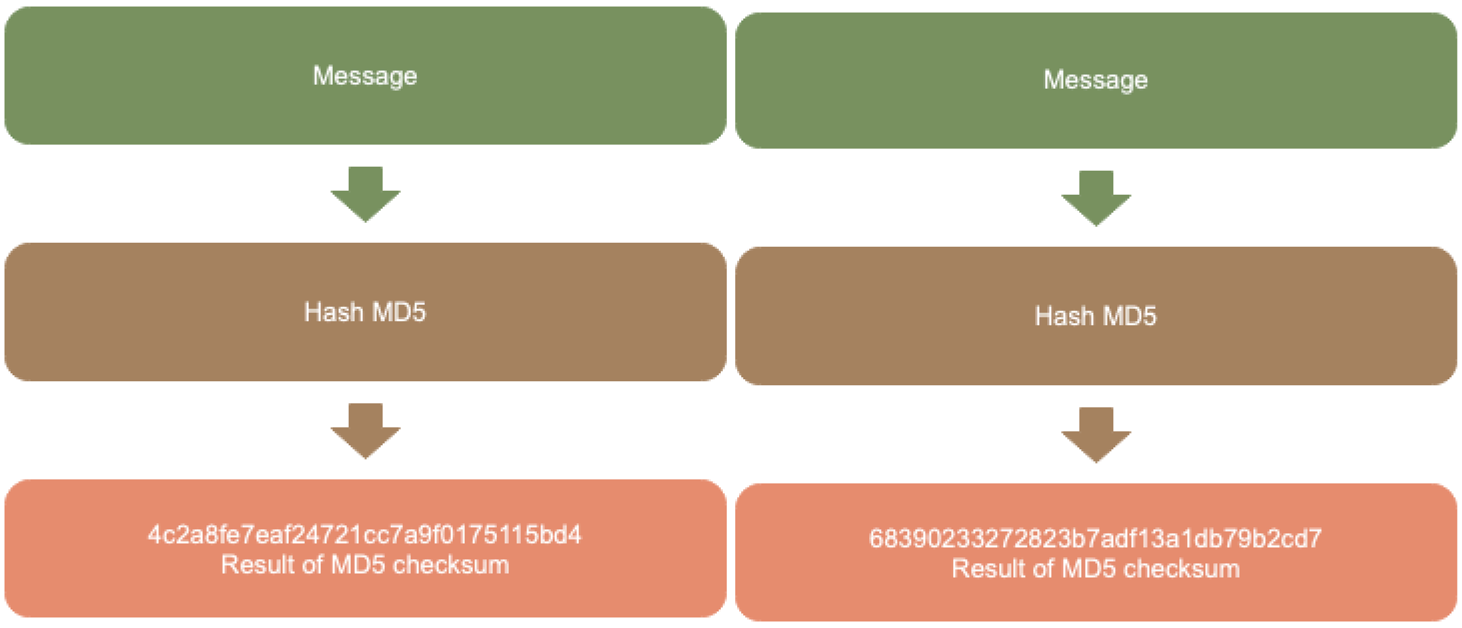
The hash technique does not use a key like the techniques seen previously. It uses a fixed-length hash value, a mathematical value computed over top of the plain text using specific algorithms. Hash functions are used to verify the integrity of data to ensure that it has not been inadvertently changed or tampered with.

It is a cryptographic function that aims to compute a message digest when creating a digital signature. This hash function, also called cryptographic digest, is used in conjunction with asymmetric cryptography to guarantee a digital document's integrity.

Imagine the following situation: a specific file is being shared. However, one of the people modifies its content and continues to distribute it over the network. In this way, whoever downloaded this content would end up with a contaminated file, which could be corrupted or even hiding some type of malware.

To ensure the integrity of shared data, generating a hash from the original file, where the software calculates a unique sequence of letters and numbers and assigns it to the files or folders that will start to be shared.

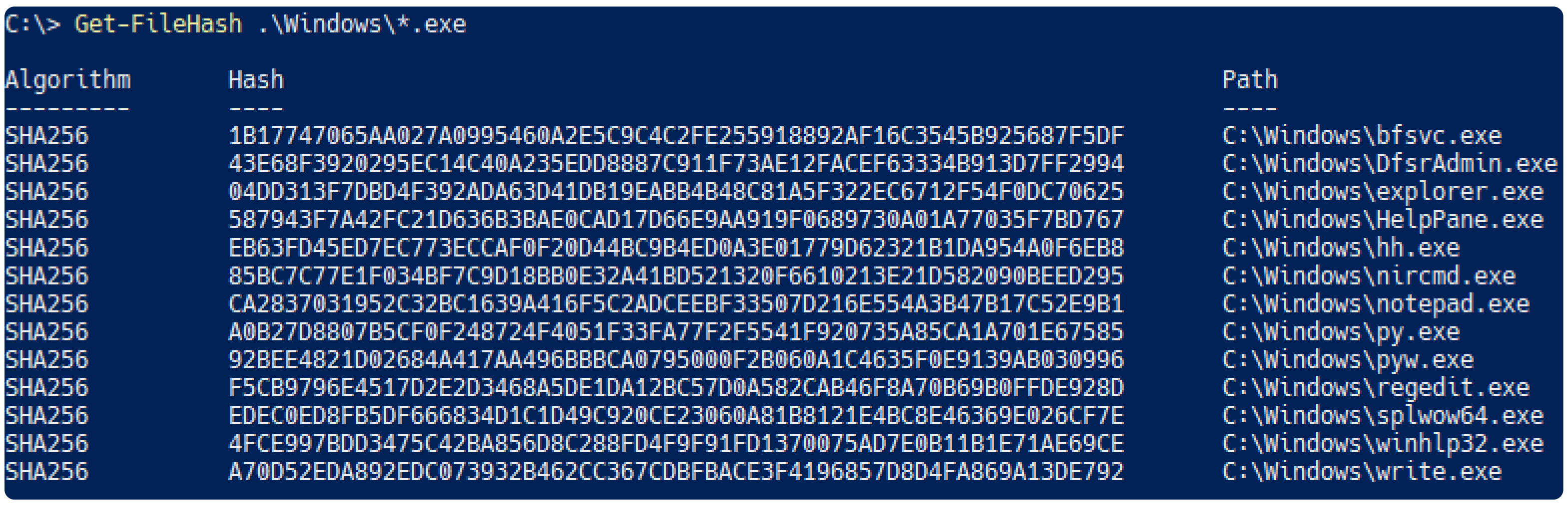
The exact process can be made using messages, as seen in the image below.



Data Integrity Guarantee

By downloading an ISO image, such as the installation DVD of a Linux distribution, for example, the same type of conference can be performed. Before burning the media and starting the installation, always check if the hash (or checksum) assigned to the original image is the same as the copy on your machine's hard drive. If the sequence of numbers and letters match, the file has not been corrupted and is a perfect copy of the online content.

For example, in the image below, you can see a command where it is possible to verify the HASH (fingerprint) of each file in the main Microsoft Windows folder.



```
C:\> Get-FileHash .\Windows\*.exe

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          1B17747065AA027A0995460A2E5C9C4C2FE255918892AF16C3545B925687F5DF       C:\Windows\bfsvc.exe
SHA256          43E68F3920295EC14C40A235EDD8887C911F73AE12FACEF63334B913D7FF2994       C:\Windows\DfsrAdmin.exe
SHA256          04DD313F7DBD4F392ADA63D41DB19EABB4B48C81A5F322EC6712F54F0DC70625       C:\Windows\explorer.exe
SHA256          587943F7A42FC21D636B3BAE0CAD17D66E9AA919F0689730A01A77035F7BD767       C:\Windows\HelpPane.exe
SHA256          EB63FD45ED7EC773ECCAF0F20D44BC9B4ED0A3E01779D62321B1DA954A0F6EB8       C:\Windows\hh.exe
SHA256          85BC7C77E1F034BF7C9D18BB0E32A41BD521320F6610213E21D582090BEED295       C:\Windows\nircmd.exe
SHA256          CA2837031952C32BC1639A416F5C2ADCEEBF33507D216E554A3B47B17C52E9B1       C:\Windows\notepad.exe
SHA256          A0B27D8807B5CF0F248724F4051F33FA77F2F5541F920735A85CA1A701E67585       C:\Windows\py.exe
SHA256          92BEE4821D02684A417AA496BBBCA0795000F2B060A1C4635F0E9139AB030996       C:\Windows\pyw.exe
SHA256          F5CB9796E4517D2E2D3468A5DE1DA12BC57D0A582CAB46F8A70B69B0FFDE928D       C:\Windows\regedit.exe
SHA256          EDEC0ED8FB5DF666834D1C1D49C920CE23060A81B8121E4BC8E46369E026CF7E       C:\Windows\splwow64.exe
SHA256          4FCE997BDD3475C42BA856D8C288FD4F9F91FD1370075AD7E0B11B1E71AE69CE       C:\Windows\winhlp32.exe
SHA256          A70D52EDA892EDC073932B462CC367CDBFBACE3F4196857D8D4FA869A13DE792       C:\Windows\write.exe
```

Hash and Practical Applications

As was said, hash codes and functions are very useful in cryptography and electronic signatures.

Today, hash codes usually have the following practical applications:

- Cryptography and electronic signature with blockchain technology: in this area, hash codes make it possible to unequivocally identify a file or document and verify that it has not undergone any alteration after its signature.
- Cryptocurrencies: hash codes are also an essential element in the cryptocurrency mining process through blockchain technology. Thus, for example, the calculation of hashes allows the creation of new Bitcoin blocks and the verification of the chain of transactions previously carried out.
- Password management: usually, online services store and manage passwords in hash format, and not in text, for greater privacy and security. These codes would also be used instead of plain text expressions in password recovery processes.
- Malware detection: hash codes that identify specific malicious programs allow them to be detected and removed more efficiently and accurately. Therefore, it is a widely used technology in the antivirus industry.
- Detection of copyright infringements: similar to the previous case, when specific services detect copyrighted content, they can automatically associate a hash to it that subsequently detects new uses of that material automatically and very efficiently.

As you can see, it is a very versatile technology, which has made it possible to implement significant digital security advances. A wide variety of online procedures and transactions, both in the public and private sections, directly depend on hash algorithms.

Types of Hash Algorithms

Over the years, different hash code generation protocols or algorithms have been developed. The goal has always been to improve the security and usability of this cryptographic tool.

Thus, some of the most used hash algorithms in recent years are the following:

- **MD5:** it was developed in 1991 to replace its predecessor, MD4.
- **SHA-1, SHA-2, and SHA-3 (pronounced "shaw one, shaw two, shaw three"):** it was created by the US National Security Agency (NSA) between 1993 and 2015. Today, they are still some of the most widely used algorithms.
- **BLAKE2 and BLAKE3:** these are recently created hash algorithms. In many cases, they offer greater efficiency and speed in code generation than previous hash functions.

Let's try Hashing

On CodeBeautify's Hash Generator, you can test hashing.

> Try hashing in the following phase.

Example 1:

Light House Labs

HASH Algorithm SHA-256

1bfe54b7acb832ca67536564013202ea7a0435e15d5d8a5832bbce757e34f5f1

Example 2:

light House Labs

HASH Algorithm SHA-256

3aaf9726e4ac9070bd3aa325e0d27878823e86acb7be7b24de1d55caa0f4f4c8

As you can see in the example, "L" was changed from uppercase to an "l" in lowercase. When you change just one letter in the phrase, this simple change can modify the whole HASH number or checksum, which guarantees the integrity of the message, because this checksum is unique for each one.

Conclusion

Cryptography is the practice of securing information by converting it into a code or cipher that can only be deciphered by authorized parties. It is a crucial component of cybersecurity, as it helps protect sensitive information from unauthorized access, modification, or theft.

As a new cyber security professional, it is important to understand the principles of cryptography, as it is an essential tool for safeguarding data and ensuring the confidentiality, integrity, and availability of information in today's digital world.

✓ Mark Completed

← Previous
Cryptanalysis Quiz

Next
Typical Levels of Cryptography →

How well did this activity help you to understand the content?

Let us know how we're doing

☆ ☆ ☆ ☆ ☆

W06D5

Fri Aug 2

Outline & Notes (1)

Lectures (1)

Work (10)

7 hrs

- Cryptanalysis
- The Use of the Historical vs. Modern Encryption
- ? Cryptanalysis Quiz
- Cryptography Features and Objectives
- Typical Levels of Cryptography
- The Use of Cryptography
- Code a Python Playfair Cipher
- Code a Python Playfair Cipher
- Cryptographic Encryption
- Cryptographic Methods with GPG

W06D5 Schedule »