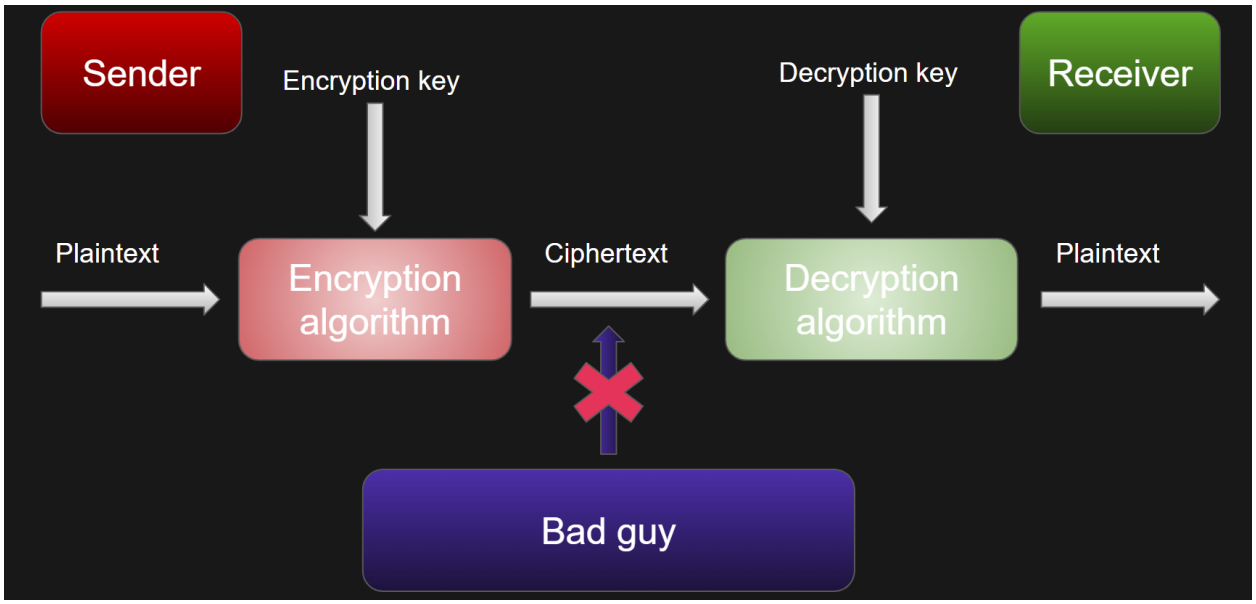
📝 - Cryptography

Complete

What is Cryptography?

Cryptography is the practice and study of techniques for secure communication in the presence of third parties called adversaries. Cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography



- Plaintext: is the data to be protected during the transmission
- Encryption algorithm: mathematical process that produces a cipherthext for any given plaintext and encryption key
- Encryption key: value known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext to obtain the ciphertext
- Ciphertext: scrambled version of the plaintext
- Decryption algorithm: mathematical process that produces a plaintext for any given ciphertext and decryption key
- Decryption key: value known to the receiver. It is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the ciphertext to obtain the plaintext back

The sender of a confidential message should be able to encrypt the message easily. It is presumed that adversaries could easily capture the ciphertext, but the encryption should be such that an adversary should not be able to easily extract the plaintext from the ciphertext
The intended recipient should be able to decipher ciphertext easily and obtain the plaintext using the decryption key. For this to be feasible sender and receiver first exchange the keys; decryption key may

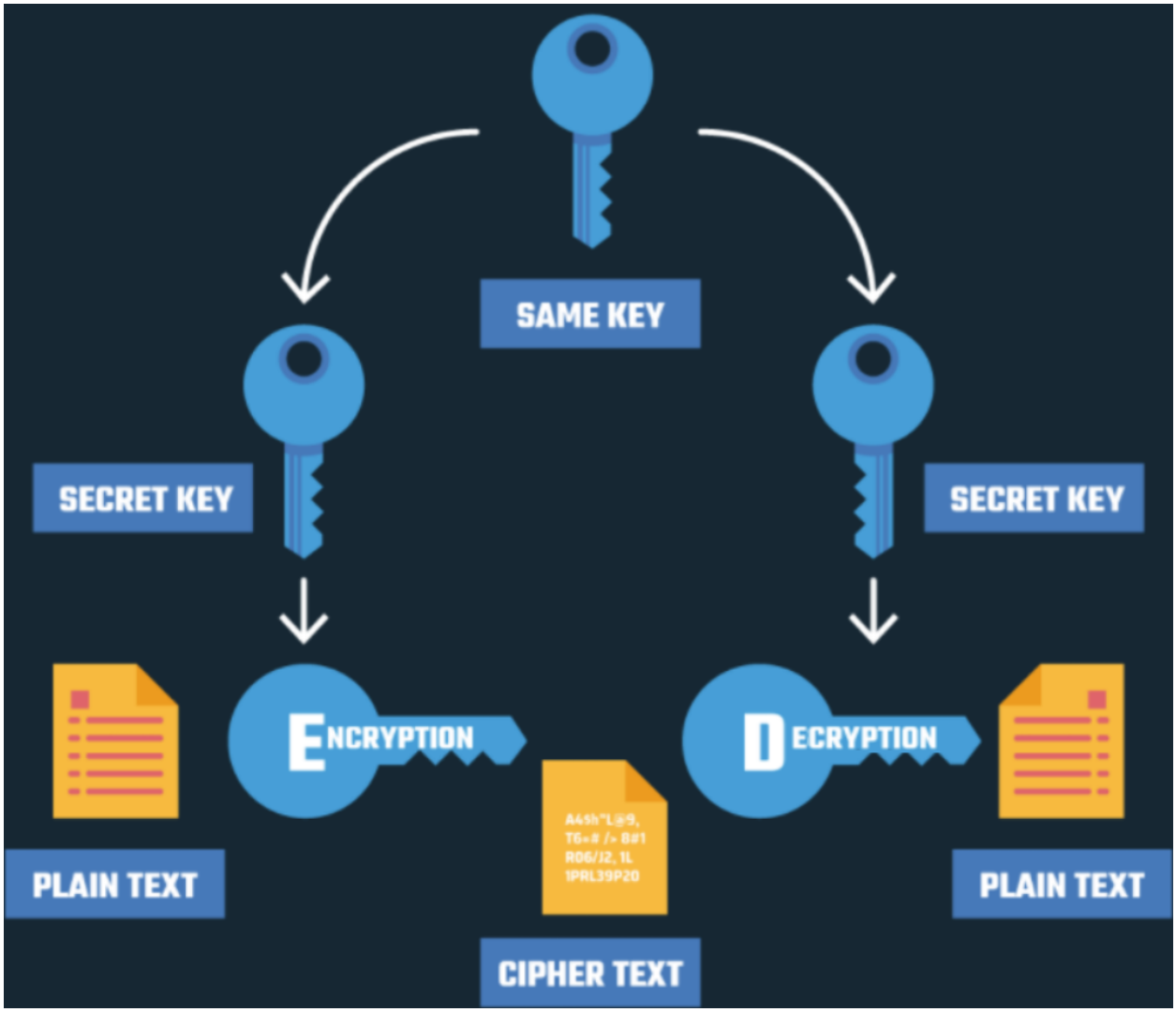
be same as encryption key in the case of symmetric cryptography, or they may be different in the case of asymmetric cryptography.
Another wanted feature is that if ciphertext gets modified in transit, the receiver should be able to detect that has been modified (data integrity).
The sender of a message should not be able to later deny having sent the message, this functionality is known as non-repudiation

Modern cryptography

The modern encryption schemes can be broadly classified into two categories—Symmetric Schemes and Asymmetric (or Public Key) Schemes. In symmetric schemes, there is only one key that is kept secret between the sender and recipient of secure messages. The secret key is used both for encryption and decryption. In asymmetric schemes, each user generates a pair of related keys, one that is made public and one that is kept secret (private) by the owner

Symmetric Encryption

Symmetric encryption is a simple cryptographic algorithm by today’s standards, however, it was once considered state of the art. Julius Caesar is believed to have used this technique in his confidential communications, and even during medieval times it had been largely used. In more recent times he German army used it to send private communications during World War II.



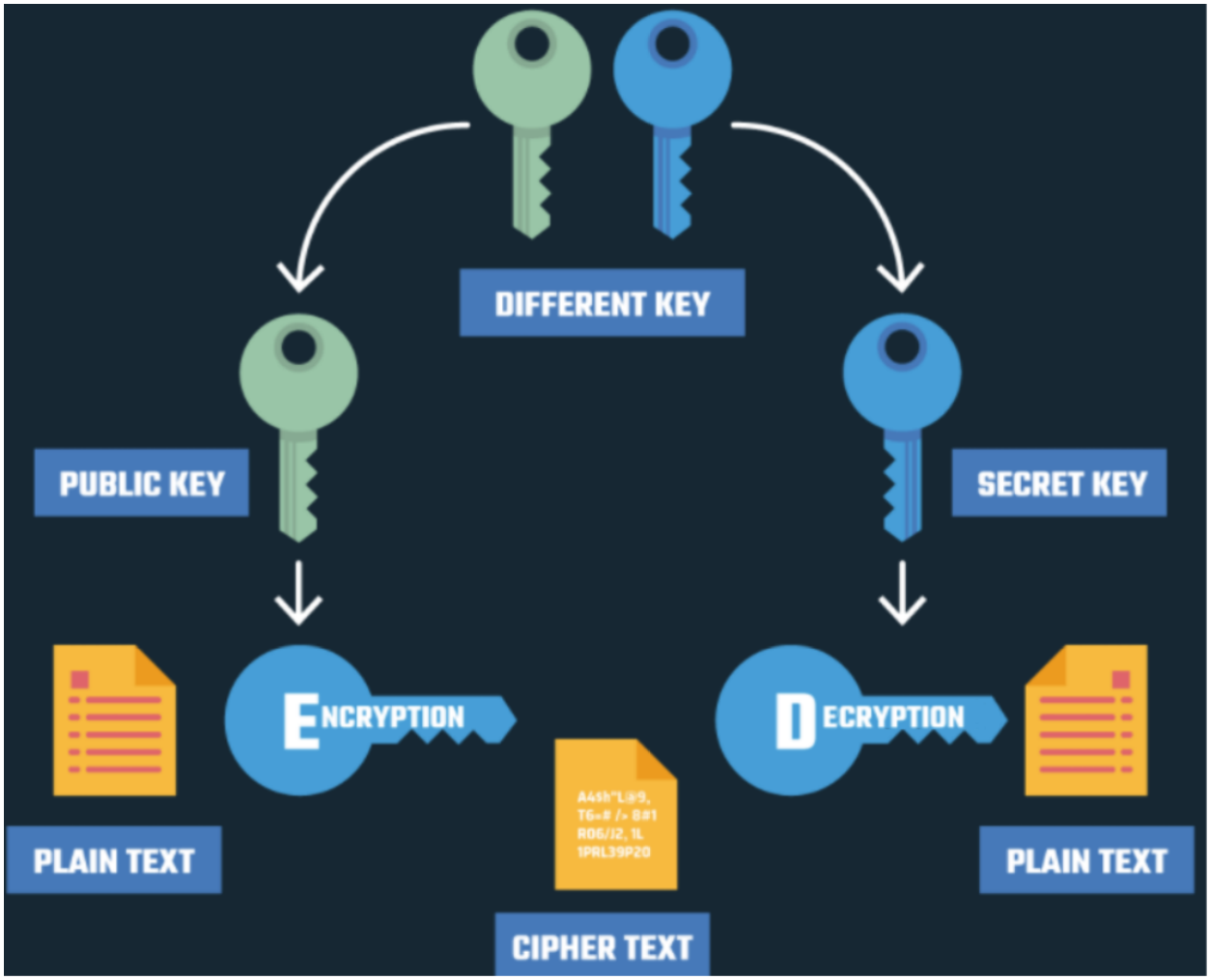
With symmetric encryption, a message that gets typed in plain text goes through mathematical permutations to become encrypted. The encrypted message is difficult to break because the same plain text letter does not always come out the same in the encrypted message. For example, the message “HHH” would not encrypt to three of the same characters.
To both encrypt and decrypt the message, you need the same key, hence the name symmetric encryption. While decrypting messages is exceedingly difficult without the key, the fact that the same key must be used to encrypt and decrypt the message carries significant risk. That’s because if the distribution channel used to share the key gets compromised, the whole system for secure messages is broken

Asymmetric Encryption

Asymmetric cryptography may be more advanced than symmetric cryptography, but both are still in use today -- and many times they get used in combination. Asymmetric encryption uses a mathematically related pair of keys for encryption and decryption: a public key and a private key.

The two participants in the asymmetric encryption workflow are the sender and the receiver; each has its own pair of public and private keys. First, the sender obtains the receiver's public key. Next, the plaintext is encrypted by the sender using the receiver's public key; this creates ciphertext. The ciphertext is then sent to the receiver, who decrypts the ciphertext with his private key and returns it to readable plaintext.

Because of the one-way nature of the encryption function, one sender is unable to read the messages of another sender, even though each has the public key of the receiver



The main advantage of this technique is that both the sender and the receiver they are not sharing their private keys, and the sender is certain that only the receiver is able to read the content of the message

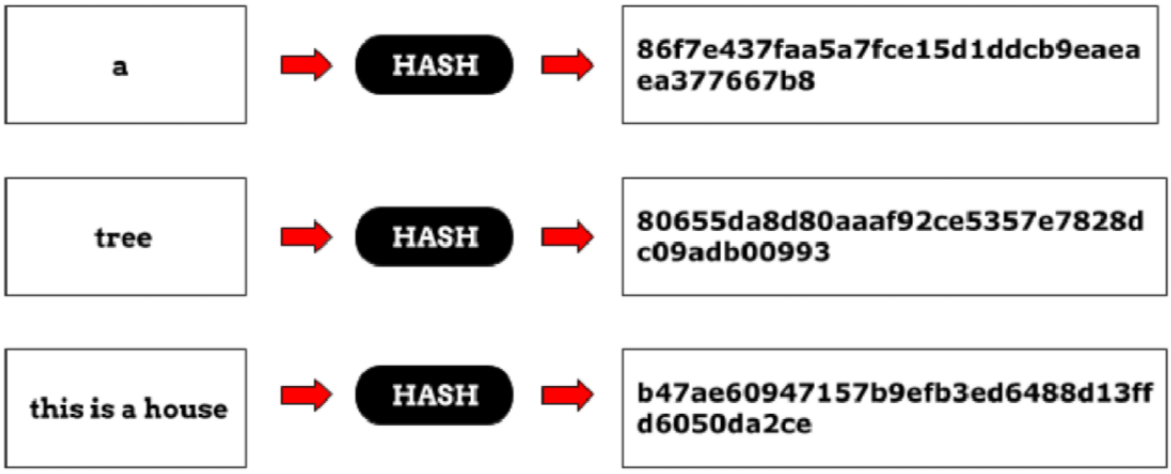
With this kind of cryptography even the use of digital signatures is enabled so that a recipient can verify that a message comes from a particular sender.
It also allows for non-repudiation so the sender can't deny sending a message
In this case the sender's private key is used to sign a message. To sign a message one first hashes (hashes are described next) the message and then encrypts the hash with the private key. A receiver can verify the hash by decrypting it using the signer’s public key and then compare the decrypted value to a computed hash of the message. If the values are equal, then the message is valid and came from the signer (assuming that the private key wasn’t stolen of course)

Symmetric vs Asymmetric

In general, asymmetric encryption it's a slow process compared to symmetric, so it's not appropriate for decrypting bulk messages. Due to this reason, most of the existing Cipher Systems are hybrid, involving both Symmetric and Asymmetric Cryptography. Public Key Cryptography is used to exchange only a secret key among the communicating parties. This is followed by Symmetric Cryptography to communicate the actual message, which is encrypted using the secret key that has been already exchanged using Public Key Cryptography. As a real world example take HTTPS. It uses asymmetric encryption to first establish the identity of one or both parties. Secondly, it uses asymmetric encryption to exchange a key to a symmetric cipher. So asymmetric is only used during the initial setup of communication. Symmetric encryption which is used through the rest is faster and more efficient with large amounts of data transfer. The keys are smaller which is generally why it's faster, but its algorithm is also easier to process

Hashing

Hashing is a mathematical operation that is easy to perform, but extremely difficult to reverse and it is used as a method of cryptography that converts any form of data into a unique string of text. Any piece of data can be hashed, no matter its size or type. In traditional hashing, regardless of the data's size, type, or length, the hash that any data produces is always the same length. A hash is designed to act as a one-way function—you can put data into a hashing algorithm and get a unique string, but if you come upon a new hash, you cannot decipher the input data it represents. A unique piece of data will always produce the same hash



Hashing can be used in various real world scenarios such as: storing safely passwords (to protect users, the passwords are not stored in plain text but they are converted into something not inteligible), file integrity checks (think of files you download, you can check if something went wrong during the download by calculating the hash of the file you have downloaded against a hash of the file which is online), Bitcoins' proof of work.

N.B.
Hashing !== Encryption, The difference between hashing and encryption is that encryption can be reversed, or decrypted, using a specific key, hashing is a one way function