Cryptography Introduction

Last Updated: 13-02-2020

**Cryptography** is the study and practice of techniques for secure communication in the presence of third parties called adversaries. It deals with developing and analyzing protocols which prevents malicious third parties from retrieving information being shared between two entities thereby following the various aspects of information security.

Secure Communication refers to the scenario where the message or data shared between two parties can’t be accessed by an adversary. In Cryptography, an Adversary is a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security.

Data Confidentiality, Data Integrity, Authentication and Non-repudiation are core principles of modern-day cryptography.

1. **Confidentiality** refers to certain rules and guidelines usually executed under confidentiality agreements which ensure that the information is restricted to certain people or places.
2. **Data integrity** refers to maintaining and making sure that the data stays accurate and consistent over its entire life cycle.
3. **Authentication** is the process of making sure that the piece of data being claimed by the user belongs to it.
4. **Non-repudiation** refers to ability to make sure that a person or a party associated with a contract or a communication cannot deny the authenticity of their signature over their document or the sending of a message.

Consider two parties Alice and Bob. Now, Alice wants to send a message m to Bob over a secure channel.  
So, what happens is as follows.  
The sender’s message or sometimes called the Plaintext, is converted into an unreadable form using a Key k. The resultant text obtained is called the Ciphertext. This process is known as Encryption. At the time of receival, the Ciphertext is converted back into the plaintext using the same Key k, so that it can be read by the receiver. This process is known as Decryption.

Alice (Sender) Bob (Receiver)

C = E (m, k) ----> m = D (C, k)

Here, C refers to the Ciphertext while E and D are the Encryption and Decryption algorithms respectively.

Let’s consider the case of Caesar Cipher or Shift Cipher as an example.  
As the name suggests, in Caesar Cipher each character in a word is replaced by another character under some defined rules. Thus, if A is replaced by D, B by E and so on. Then, each character in the word would be shifted by a position of 3. For example:

Plaintext : Geeksforgeeks

Ciphertext : Jhhnvirujhhnv

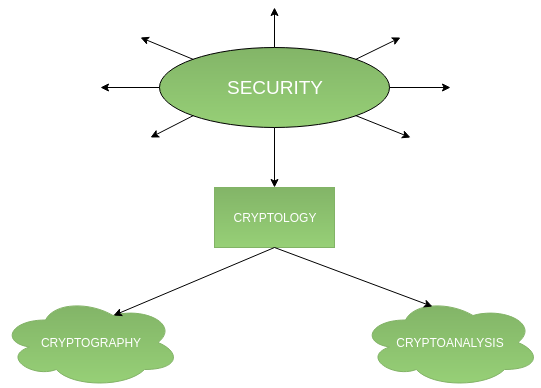
**Note** that even if the adversary knows that the cipher is based on Caesar Cipher, it cannot predict the plaintext as it doesn’t have the key in this case which is to shift the characters back by three places.

Introduction to Crypto-terminologies

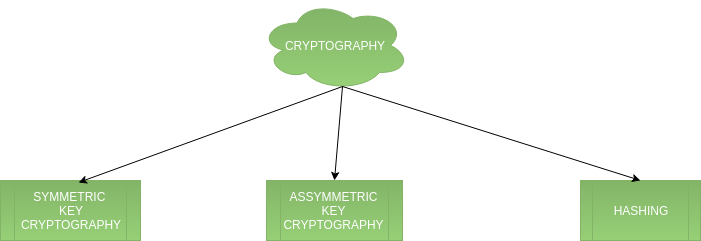
Last Updated: 09-08-2019

Cryptography is an important aspect when we deal with network security. ‘Crypto’ means secret or hidden. Cryptography is the science of secret writing with the intention of keeping the data secret. Cryptanalysis, on the other hand, is the science or sometimes the art of breaking cryptosystems. These both terms are a subset of what is called as Cryptology.

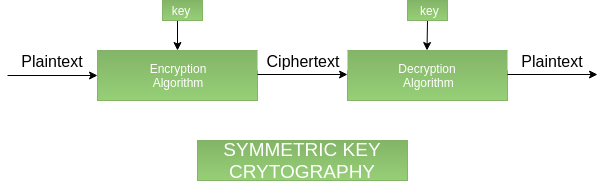
**Classification –**  
The flowchart depicts that cryptology is only one of the factors involved in securing networks. Cryptology refers to study of codes, which involves both writing (cryptography) and solving (cryptanalysis) them. Below is a classification of the crypto-terminologies and their various types.



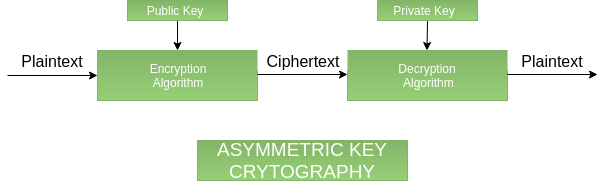
**1. Cryptography –**  
Cryptography is classified into symmetric cryptography, asymmetric cryptography and hashing. Below are the description of these types.



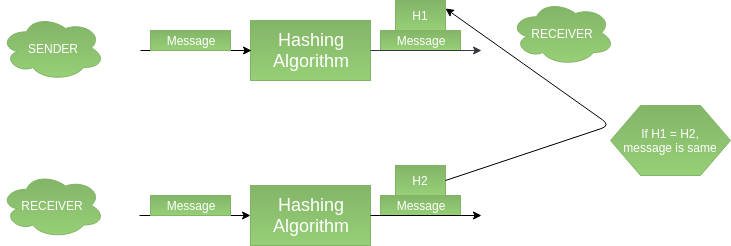
1. **Symmetric key cryptography –**  
   It involves usage of one secret key along with encryption and decryption algorithms which help in securing the contents of the message. The strength of symmetric key cryptography depends upon the number of key bits. It is relatively faster than asymmetric key cryptography. There arises a key distribution problem as the key has to be transferred from the sender to receiver through a secure channel.



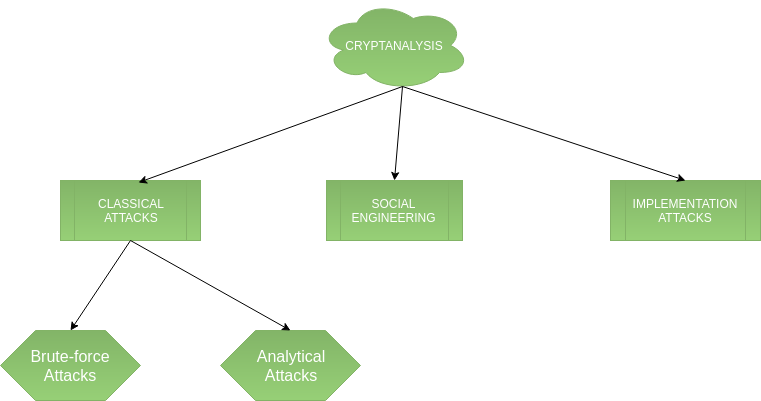
1. **Assymetric key cryptography –**  
   It is also known as public key cryptography because it involves usage of a public key along with secret key. It solves the problem of key distribution as both parties uses different keys for encryption/decryption. It is not feasible to use for decrypting bulk messages as it is very slow compared to symmetric key cryptography.



1. **Hashing –**  
   It involves taking the plain-text and converting it to a hash value of fixed size by a hash function. This process ensures integrity of the message as the hash value on both, sender\’s and receiver\’s side should match if the message is unaltered.



**2. Cryptanalysis –**



1. **Classical attacks –**  
   It can be divided into a)Mathematical analysis and b) Brute-force attacks. Brute-force attacks runs the encryption algorithm for all possible cases of the keys until a match is found. Encryption algorithm is treated as a black box. Analytical attacks are those attacks which focuses on breaking the cryptosystem by analysing the internal structure of the encryption algorithm.
2. **Social Engineering attack –**  
   It is something which is dependent on the human factor. Tricking someone to reveal their passwords to the attacker or allowing access to the restricted area comes under this attack. People should be cautious when revealing their passwords to any third party which is not trusted.
3. **Implementation attacks –**  
   Implementation attacks such as side-channel analysis can be used to obtain a secret key. They are relevant in cases where the attacker can obtain physical access to the cryptosystem.

Difference Between Symmetric and Asymmetric Key Encryption

**Symmetric Key Encryption:**  
[Encryption](https://www.geeksforgeeks.org/encryption-its-algorithms-and-its-future/) is a process to change the form of any message in order to protect it from reading by anyone. In Symmetric-key encryption the message is encrypted by using a key and the same key is used to decrypt the message which makes it easy to use but less secure. It also requires a safe method to transfer the key from one party to another.

**Asymmetric Key Encryption:**  
Asymmetric Key Encryption is based on public and private key encryption technique. It uses two different key to encrypt and decrypt the message. It is more secure than symmetric key encryption technique but is much slower.

| **SYMMETRIC KEY ENCRYPTION** | **ASYMMETRIC KEY ENCRYPTION** |
| --- | --- |
| It only requires a single key for both encryption and decryption. | It requires two key one to encrypt and the other one to decrypt. |
| The size of cipher text is same or smaller than the original plain text. | The size of cipher text is same or larger than the original plain text. |
| The encryption process is very fast. | The encryption process is slow. |
| It is used when a large amount of data is required to transfer. | It is used to transfer small amount of data. |
| It only provides confidentiality. | It provides confidentiality, authenticity and non-repudiation. |
| Examples: 3DES, AES, DES and RC4 | Examples: Diffie-Hellman, ECC, El Gamal, DSA and RSA |
| In symmetric key encryption, resource utilization is low as compared to asymmetric key encryption. |  |

# What is the RSA algorithm?

The **RSA algorithm** is an asymmetric cryptography algorithm; this means that it uses a public key and a private key (i.e two different, mathematically linked keys). As their names suggest, a public key is shared publicly, while a private key is secret and must not be shared with anyone.

The RSA algorithm is named after those who invented it in 1978: Ron Rivest, Adi Shamir, and Leonard Adleman.

The following illustration highlights how asymmetric cryptography works:

## How it works

The RSA algorithm ensures that the keys, in the above illustration, are as secure as possible. The following steps highlight how it works:

### 1. Generating the keys

1. Select two large prime numbers, x*x* and y*y*. The prime numbers need to be large so that they will be difficult for someone to figure out.
2. Calculate n = x \* y*n*=*x*∗*y*.
3. Calculate the ***totient*** function; \phi(n) = (x-1)(y-1)*ϕ*(*n*)=(*x*−1)(*y*−1).
4. Select an integer e*e*, such that e*e* is ***co-prime*** to \phi(n)*ϕ*(*n*) and 1 < e < \phi(n)1<*e*<*ϕ*(*n*). The pair of numbers (n,e)(*n*,*e*) makes up the public key.

**Note:** Two integers are co-prime if the only positive integer that divides them is 1.

1. Calculate d*d* such that e.d = 1*e*.*d*=1 mod*mod* \phi(n)*ϕ*(*n*).

d*d* can be found using the ***extended euclidean algorithm***. The pair (n,d)(*n*,*d*) makes up the private key.

### 2. Encryption

Given a plaintext P*P*, represented as a number, the ciphertext C*C* is calculated as:

C = P^{e}*C*=*P*​*e*​​ mod*mod* n*n*.

### 3. Decryption

Using the private key (n,d)(*n*,*d*), the plaintext can be found using:

P = C^{d}*P*=*C*​*d*​​ mod*mod* n*n*.

## Pseudocode

Consider an example of the RSA algorithm through the following pseudocode:

int x = 61, int y = 53;  
int n = x \* y;  
// n = 3233.  
   
// compute the totient, phi  
int phi = (x-1)\*(y-1);  
// phi = 3120.  
   
int e = findCoprime(phi);  
// find an 'e' which is > 1 and is a co-prime of phi.  
// e = 17 satisfies the current values.  
   
// Using the extended euclidean algorithm, find 'd' which satisfies   
// this equation:  
d = (1 mod (phi))/e;  
// d = 2753 for the example values.  
   
public\_key = (e=17, n=3233);  
private\_key = (d=2753, n=3233);  
   
// Given the plaintext P=123, the ciphertext C is :  
C = (123^17) % 3233 = 855;  
// To decrypt the cypher text C:  
P = (855^2753) % 3233 = 123;