# Symmetric Encryption

**Rijndael** is an algorithm for Advanced Encryption Standard (AES). The algorithm was designed by two Belgian cryptologists, Vincent Rijmen and Joan Daemen, whose surnames are reflected in the cipher's name. Rijndael has its origins in Square, an earlier collaboration between the two cryptologists.

The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in 128-bit blocks - however, in excess of AES design criteria, the block sizes can mirror those of the keys. Rijndael uses a variable number of rounds, depending on key/block sizes, as follows:

9 rounds if the key/block size is 128 bits

11 rounds if the key/block size is 192 bits

13 rounds if the key/block size is 256 bits

Rijndael is a substitution linear transformation cipher, not requiring a Feistel network. It uses triple discreet invertible uniform transformations (layers). Specifically, these are: Linear Mix Transform; Non-linear Transform and Key Addition Transform. Even before the first round, a simple key addition layer is performed, which adds to security. Thereafter, there are Nr-1 rounds and then the final round. The transformations form a State when started but before completion of the entire process.

The State can be thought of as an array, structured with 4 rows and the column number being the block length divided by bit length (for example, divided by 32). The cipher key similarly is an array with 4 rows, but the key length divided by 32 to give the number of columns. The blocks can be interpreted as unidimensional arrays of 4-byte vectors.

The exact transformations occur as follows:

1. The byte sub transformation is nonlinear and operates on each of the State bytes independently - the invertible S-box (substitution table) is made up of 2 transformations.
2. The shiftrow transformation sees the State shifted over variable offsets. The shift offset values are dependent on the block length of the State.
3. The mixcolumn transformation sees the State columns take on polynomial characteristics over a Galois Field values (28), multiplied x4 + 1 (modulo) with a fixed polynomial.
4. Finally, the roundkey transform is XORed to the State. The key schedule helps the cipher key determine the round keys through key expansion and round selection.

Overall, the structure of Rijndael displays a high degree of modular design, which should make modification to counter any attack developed in the future much simpler than with past algorithm designs.

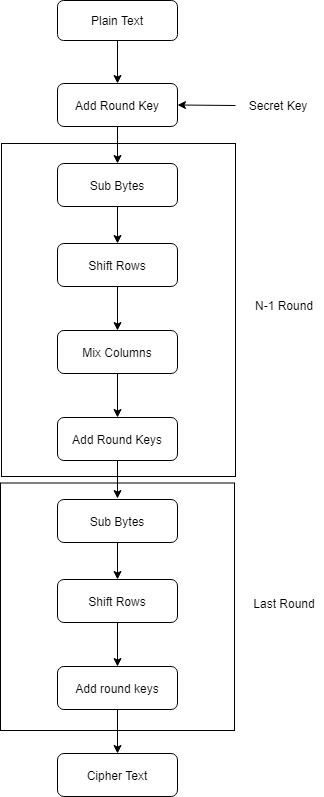


Figure: Rijndael Algorithm flowchart

# Asymmetric Encryption

Asymmetric encryption is also referred to as public key encryption. In asymmetric encryption, both the encrypting and decrypting systems have a set of keys. One is called the public key, and another is called the private key. If the message is encrypted with one key in the pair, the message can be decrypted only with the other key in the pair.

Asymmetric key algorithms are not quite as fast as symmetric key algorithms. This is partially because asymmetric key algorithms are generally more complex, using a more sophisticated set of functions. Asymmetric key algorithms are not as widely used as their symmetric counterparts. **Diffie-Hellman** is one of the most used Asymmetric encryption algorithms.

The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

For the sake of simplicity and practical implementation of the algorithm, we will consider only 4 variables one prime P and G (a primitive root of P) and two private values a and b.

P and G are both publicly available numbers. Users (say Alice and Bob) pick private values a and b and they generate a key and exchange it publicly, the opposite person received the key and from that generates a secret key after which they have the same secret key to encrypt.

Step by Step Explanation

|  |  |
| --- | --- |
| ALICE | BOB |
| Public Keys available = P, G | Public Keys available = P, G |
| Private Key Selected = a | Private Key Selected = b |
| Key generated = x = Ga mod P | Key generated = y = Gb mod P |
| Exchange of generated keys takes place | |
| Key received = y | key received = x |
| Generated Secret Key = ka = ya mod P | Generated Secret Key = kb = yb mod P |
| Algebraically it can be shown that ka = kb | |
| Users now have a symmetric secret key to encrypt | |
|  |  |

# Difference Between Symmetric and Asymmetric Encryption

* Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetrical encryption uses a pair of a public key and a private key to encrypt and decrypt messages when communicating.
* Asymmetric encryption was introduced to complement the inherent problem of the need to share the key in symmetrical encryption model, eliminating the need to share the key by using a pair of public-private keys.
* Asymmetric encryption takes relatively more time than the symmetric encryption.

# Using Hashtable for encryption

Hashing is using a special cryptographic function to transform one set of data into another of fixed length by using a mathematical process. The process involves mapping data of any size to a fixed length using a hash table and then storing the output data in the digest.

Encryption encodes data so that only the authorized party can read it by possessing the correct key, which will decrypt the data into a readable message. The major difference between hashing and encryption is that encrypted data can be made readable again whereas hashed data cannot.

But there is a tricky way to use hashing as encryption. After data has been converted to a fixed length key by hash function using another mathematical hash function that data can be converted into a mathematical number if needed. And then storing that data in a hash table with the index of hashed data can provide the opportunity to someone to retrieve the actual data from the hash table using the hashed data as a key. Following this technique hash table can be used as an encryption tool. In the context of security, it is virtually impossible to reconstruct the input data from the output, even if the hash function is known. It’s worth mentioning that even if someone were to compromise the database, they would not be able to do anything immediately with the passwords, as there is no simple method to find the password that produced the hash that they now possess. They are also useful for comparing the contents of two files or a collection of files without comparing the entire files.

<https://www.sciencedirect.com/topics/computer-science/asymmetric-key-algorithm>

<https://searchsecurity.techtarget.com/definition/Rijndael>

<https://www.geeksforgeeks.org/implementation-diffie-hellman-algorithm/>

<https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences>

<https://sandstormit.com/data-hashing-and-encryption-and-how-they-enhance-security/>