**What is the Advanced Encryption Standard?**

The AES Encryption algorithm (also known as the Rijndael algorithm) is a symmetric block cipher algorithm with a block/chunk size of 128 bits. It converts these individual blocks using keys of 128, 192, and 256 bits. Once it encrypts these blocks, it joins them together to form the ciphertext.

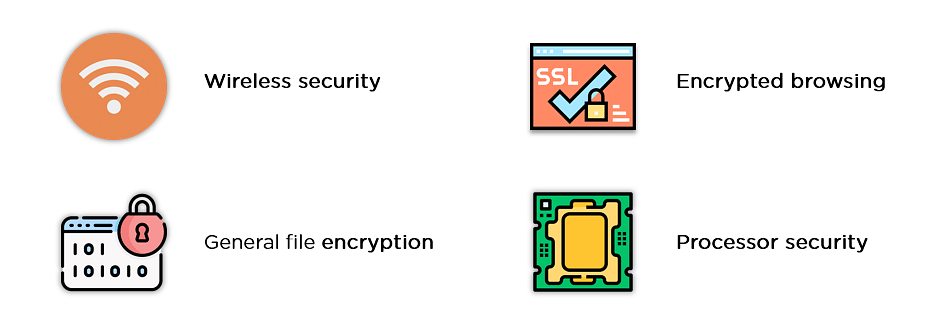
It is based on a substitution-permutation network, also known as an SP network. It consists of a series of linked operations, including replacing inputs with specific outputs (substitutions) and others involving bit shuffling (permutations).

In this tutorial, you will go through some of the standout features that AES offers as a globally standardized encryption algorithm.

**What are the Features of AES?**

1. SP Network: It works on an SP network structure rather than a Feistel cipher structure, as seen in the case of the DES algorithm.
2. Key Expansion: It takes a single key up during the first stage, which is later expanded to multiple keys used in individual rounds.
3. Byte Data: The AES encryption algorithm does operations on byte data instead of bit data. So it treats the 128-bit block size as 16 bytes during the encryption procedure.
4. Key Length: The number of rounds to be carried out depends on the length of the key being used to encrypt data. The 128-bit key size has ten rounds, the 192-bit key size has 12 rounds, and the 256-bit key size has 14 rounds.

**What Are the Applications of AES?**

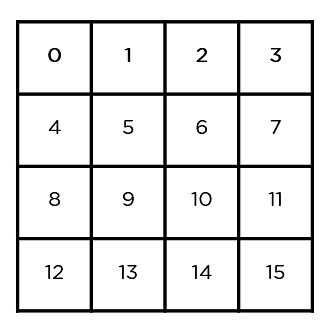


The applications of the AES Encryption algorithm are as follows:

1. Wireless Security: Wireless networks are secured using the Advanced Encryption Standard to authenticate routers and clients. WiFi networks have firmware software and complete security systems based on this algorithm and are now in everyday use.
2. Encrypted Browsing: AES plays a huge role in securing website server authentication from both client and server end. With both [symmetric](https://www.simplilearn.com/tutorials/cryptography-tutorial/symmetric-encryption) and [asymmetric encryption](https://www.simplilearn.com/tutorials/cryptography-tutorial/asymmetric-encryption) being used, this algorithm helps in SSL/TLS encryption protocols to always browse with the utmost security and privacy.
3. General File Encryption: Apart from corporate necessities, AES is also used to transfer files between associates in an encrypted format. The encrypted information can extend to chat messages, family pictures, legal documents, etc.
4. Processor Security: Many processor manufacturers enable hardware-level encryption using the likes of AES encryption to bolster security and prevent meltdown failures, among other low-profile risks.

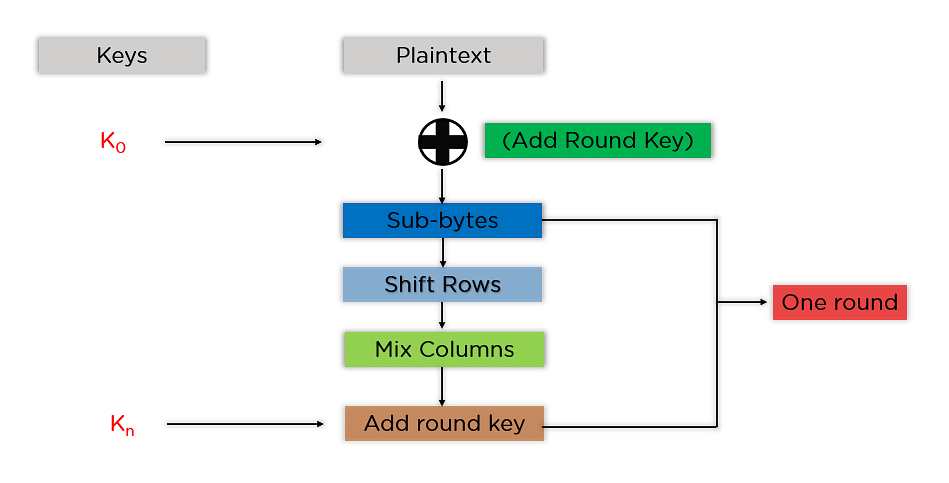
**How Does AES Work?**

To understand the way AES works, you first need to learn how it transmits information between multiple steps. Since a single block is 16 bytes, a 4x4 matrix holds the data in a single block, with each cell holding a single byte of information.



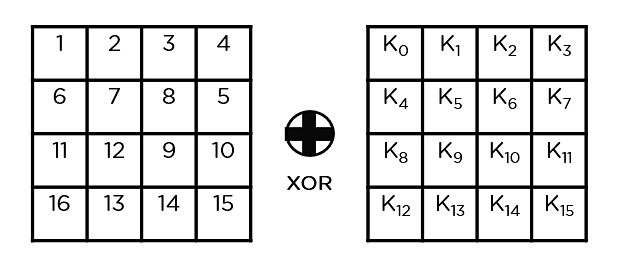
The matrix shown in the image above is known as a state array. Similarly, the key being used initially is expanded into (n+1) keys, with n being the number of rounds to be followed in the encryption process. So for a 128-bit key, the number of rounds is 16, with no. of keys to be generated being 10+1, which is a total of 11 keys.

The steps to be followed in AES are in the below image.

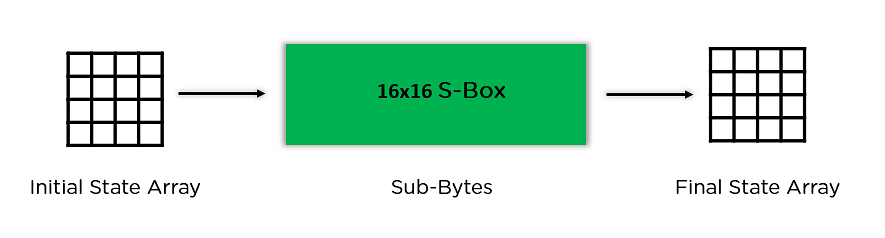


The mentioned steps are to be followed for every block sequentially. Upon successfully encrypting the individual blocks, it joins them together to form the final ciphertext. The steps are as follows:

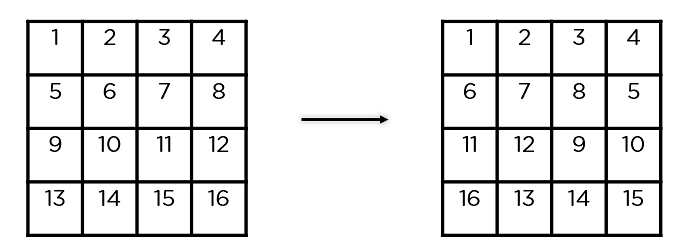
* Add Round Key: You pass the block data stored in the state array through an XOR function with the first key generated (K0). It passes the resultant state array on as input to the next step.



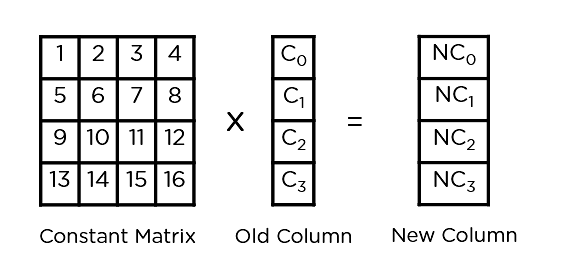
* Sub-Bytes: In this step, it converts each byte of the state array into hexadecimal, divided into two equal parts. These parts are the rows and columns, mapped with a substitution box (S-Box) to generate new values for the final state array.



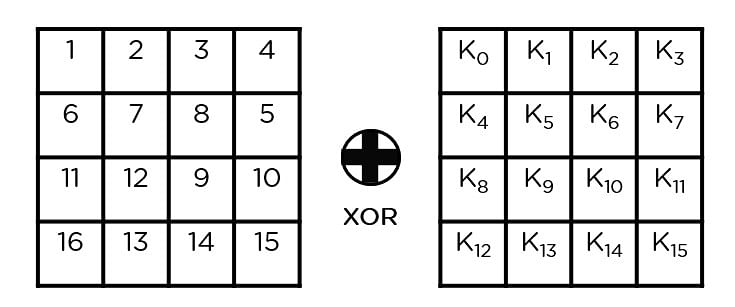
* Shift Rows: It swaps the row elements among each other. It skips the first row. It shifts the elements in the second row, one position to the left. It also shifts the elements from the third row two consecutive positions to the left, and it shifts the last row three positions to the left.



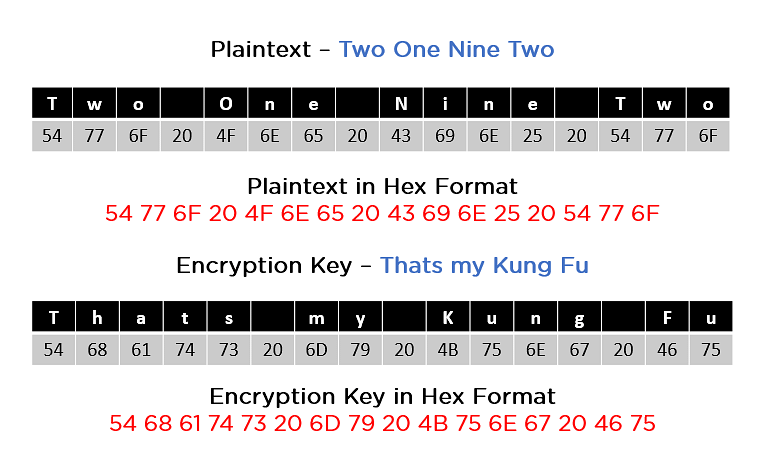
* Mix Columns: It multiplies a constant matrix with each column in the state array to get a new column for the subsequent state array. Once all the columns are multiplied with the same constant matrix, you get your state array for the next step. This particular step is not to be done in the last round.



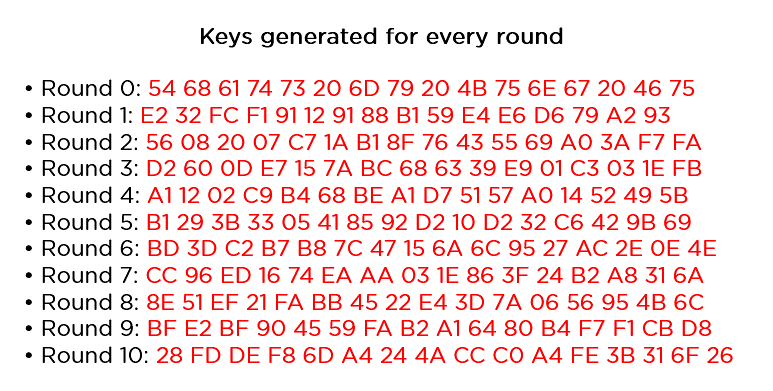
* Add Round Key: The respective key for the round is XOR’d with the state array is obtained in the previous step. If this is the last round, the resultant state array becomes the ciphertext for the specific block; else, it passes as the new state array input for the next round.



Now that you understand the basic steps needed to go through the encryption procedure, understand this example to follow along.

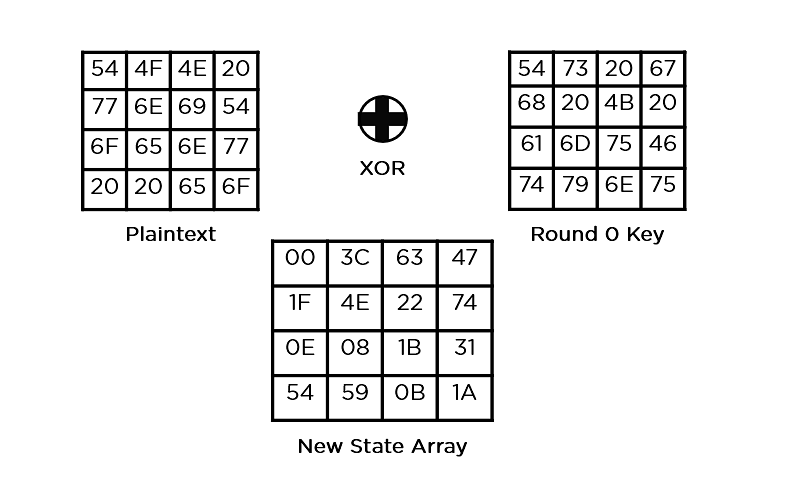


As you can see in the image above, the plaintext and encryption convert keys to hex format before the operations begin. Accordingly, you can generate the keys for the next ten rounds, as you can see below.

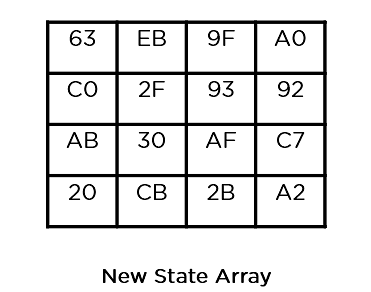


You need to follow the same steps explained above, sequentially extracting the state array and passing it off as input to the next round. The steps are as follows:

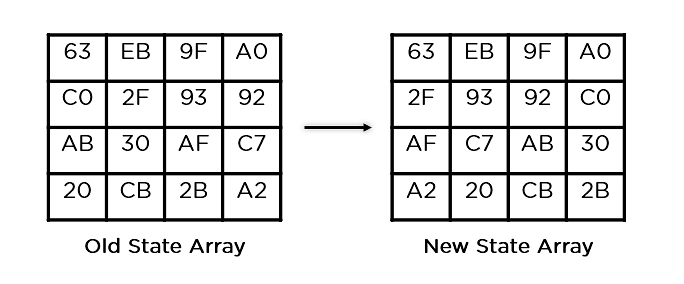
* Add Round Key:



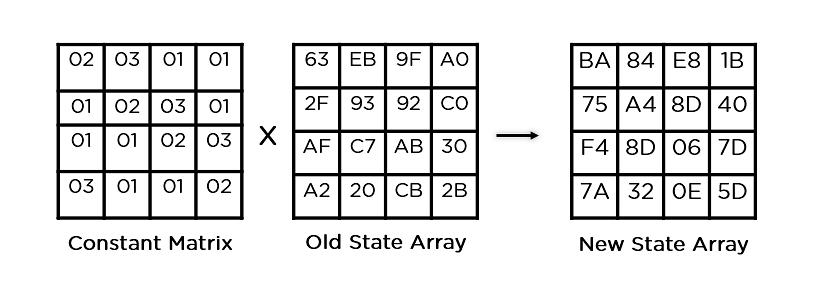
* Sub-Bytes: It passes the elements through a 16x16 S-Box to get a completely new state array.



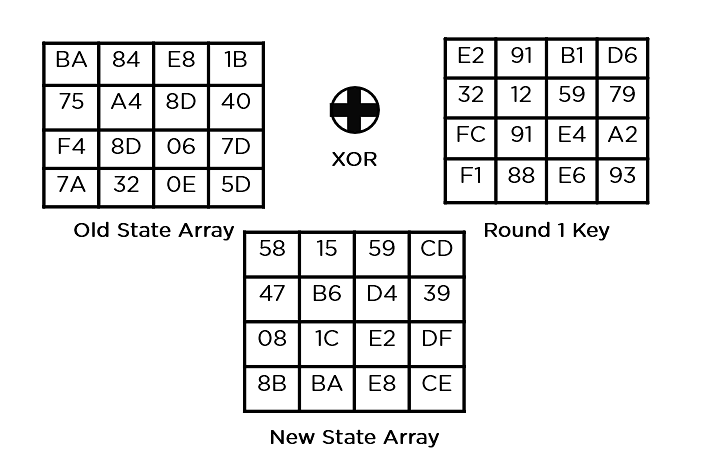
* Shift Rows:



* Mix Columns:



* Add Round Key:



This state array is now the final ciphertext for this particular round. This becomes the input for the next round. Depending on the key length, you repeat the above steps until you complete round 10, after which you receive the final ciphertext.

