A stream cipher encrypts text by applying a key and algorithm to each bit of a data stream one by one. In contrast, block ciphers process data in blocks of multiple bits at once. Stream ciphers are mainly used to encrypt one byte (8 bits) at a time.

Since stream ciphers are linear, messages are encrypted and decrypted with the help of the same key. And, while cracking them is difficult, hackers will have to manage to do it.

In this, a keystream, a random series of bits, is generated from a key. To encrypt the data stream, each bit is XORed with an equivalent bit from the keystream.

## How does it work?

Stream ciphers make use of a common key (symmetric key) to code their data. Encryption and decryption processes of the data are handled by this symmetric key. Unlike public-key ciphers, stream ciphers utilize one key for encryption as well as decryption, eliminating the need for different keys for each task (for instance, using one key to encrypt and another to decrypt).

Cryptographic methods generally conceal data from unauthorized access by scrambling it. However, stream ciphers differ by processing data bit-by-bit, unlike block ciphers that operate on collections of data known as blocks.

## Stream ciphers involve

**Plaintext** − The original message to be encrypted.

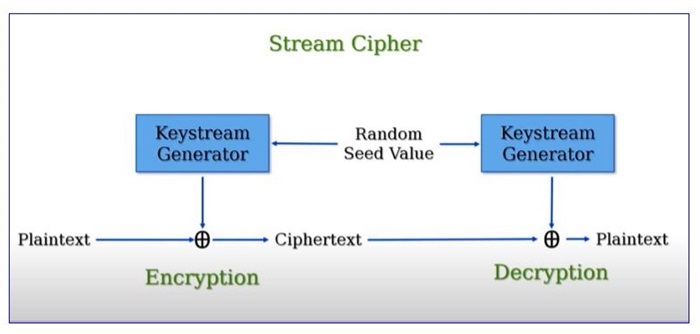
**Keystreams** − Random sequences of characters (e.g., numbers, letters, symbols) that replace the plaintext characters.

**Ciphertext** − The encrypted message. Key generation is a complex mathematical operation, but modern computers can perform it quickly.

In a stream cipher, individual bits of plaintext are inputted and subjected to a mathematical operation. The result is jumbled text which needs the right key to decode.

Using the proper key, the receiver can reverse the process and convert the scrambled text back to its plaintext.

In stream cipher encryption, the key known as a one-time pad is exceptionally secure due to its unique property. It is designed to be equivalent in length to the message being encrypted, ensuring that an attacker cannot mathematically decipher the message without having the original key.



## Keystream Generation

Let us see at a basic example of keystream creation with the help of an XOR-based stream cipher.

Let's say we have the below data −

Key: 101011

Initialization Vector (IV): 110100

To create a stream of encrypted data (keystream):

Set up the encryption algorithm with a secret key and initialization vector (IV).

If needed, adjust the key and IV to be the same length as the message being encrypted.

Combine the key and IV using an exclusive OR (XOR) operation to generate the keystream.

Here is the step by step process −

Key: 101011

IV: 110100

### Keystream: 011111

Now, let us say we have a plaintext message as: 1100101.

To encrypt this plaintext using the keystream −

Plaintext: 1100101

Keystream: 011111

### Ciphertext: 1011010

To decrypt the ciphertext, we would use the same keystream −

Ciphertext: 1011010

Keystream: 011111

### Plaintext: 1100101