Blowfish Algorithm

A symmetric-key block cipher called blowfish encryption is frequently used for password hashing, VPNs, and file encryption. Since its introduction in 1993, this encryption method has gained popularity due to its effective encryption and decryption operations. However, more recent, more secure algorithms like AES are gradually taking the place of Blowfish.

Blowfish is a 64-bit block cipher that uses symmetric encryption and a key that can be up to 448 bits long. It was created in 1993 by Bruce Schneier to replace the outdated Data Encryption Standard (DES) and International Data Encryption Algorithm (IDEA) encryption methods.

Though its popularity has decreased recently, blowfish is well known for its ease of use and efficiency. It is being replaced by more recent, stronger encryption methods like the Advanced Encryption Standard (AES).

Features of Blowfish

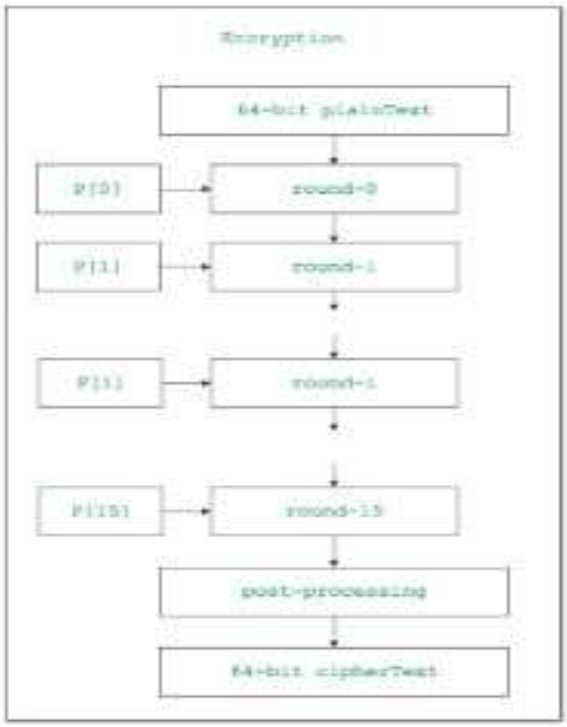
Some of the main features of the Blowfish algorithm are as follows −

- Block Cipher − Data in Blowfish is encrypted using a block cipher technique using symmetric keys, resulting in 64-bit blocks of encryption.
- Symmetric key algorithm − The Blowfish approach encrypts and decrypts data with the same symmetric encryption key.
- Different length keys − Blowfish offers key lengths ranging from 32 bits to 448 bits. The longer the key, more secure the data. However, processing longer keys usually requires more resources and time.
- Feistel Code − The Feistel cipher development divides the plaintext in half and jumbles each half independently using a sequence of mathematical operations.

Working of Blowfish

An SP network is used by Blowfish; the substitution box (S-box) and permutation box (P-box) must be started first. There are four 32-bit S-boxes with 256 entries each and eight P-arrays with 32-bit subkeys.

- **Step 1** − First, we divided the 64-bit plaintext into two equal blocks, L and R, each containing 32 bits.
- **Step 2** − The following actions are taken in each of the 16 encryption cycles that we begin in the following step −
  - Now, the L and the first member of the P-array (P1) are XORed.
  - Then XOR R with F, where F is a function of L and uses the four blocks that make up the S-box. Below is a summary of function F in entirety.
  - The next iteration of the loop starts once L and R are switched.
- **Step 3** − L and R are switched again after the loop is completed.
- **Step 4** − XOR R with P17 and L with P18 to get the final two unused P-box entries (P17 & P18).
- **Step 5** - The cipher text is obtained by combining L and R in the final step.

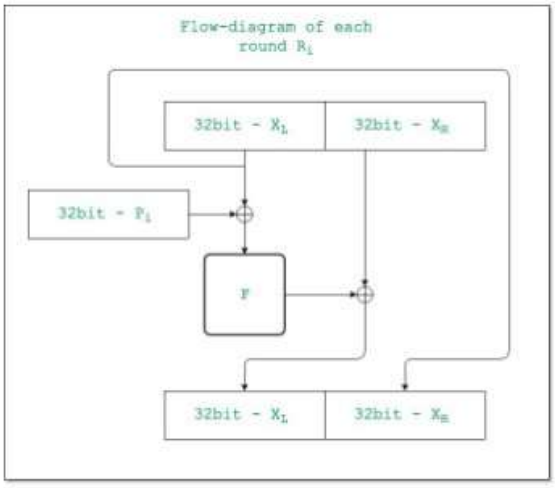


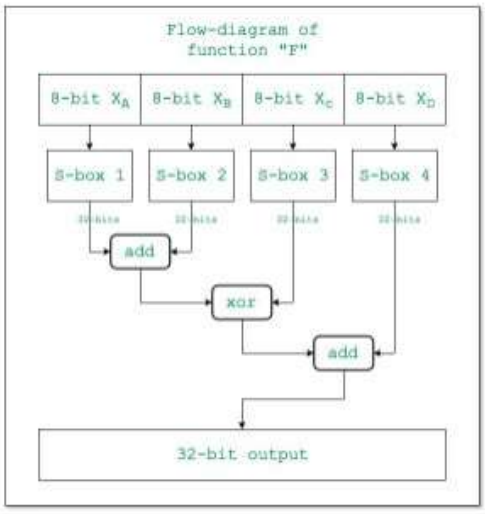
Encryption of Blowfish

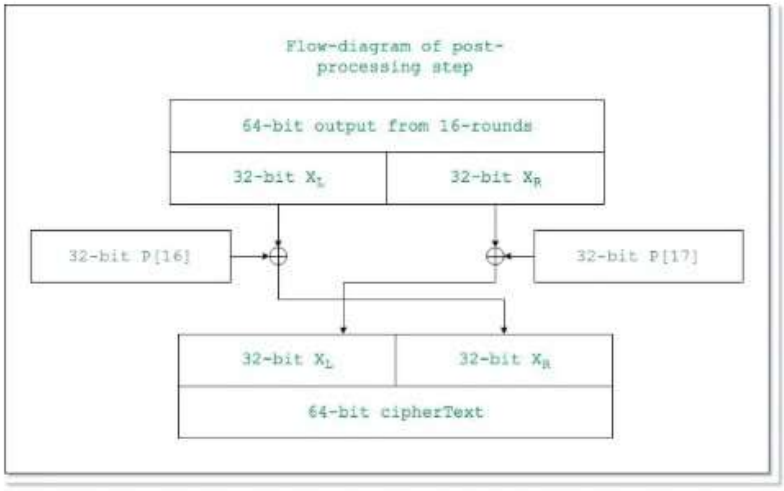
A symmetric key block cipher called Blowfish uses the same key for both encryption and decryption of data. Blowfish is quick and efficient mainly because it is simpler than other cryptography methods. While there are a few possible risks involved in achieving the highest level of data security, these risks cannot be ignored.

Here is an in-depth description of the Blowfish encryption technique −

- **Key expansion** − The initial component that Blowfish uses is a secret key, which can be anything between 32 and 448 bits long. The encryption key is then generated and extended using the P-array and S-boxes precomputation to generate several subkeys.
- **Subkeys Generation** − The 64-bit blocks that define the stretched-out key are divided into two 32-bit chunks. These components are joined with a few predetermined values to create a new set of subkeys.
- **Data Encryption** − This is when the exciting part starts. These two 32-bit segments are sixteen times encrypted. Every round involves a challenging set of transpositions and replacements (XOR operations, additions, and lookups in the S-boxes).
- **After processing** − The 32-bit scrambled bits are reconstructed to form 64-bit ciphertext blocks after 16 rounds.



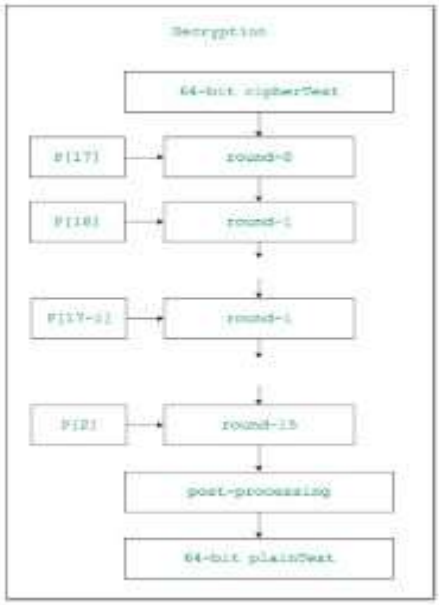


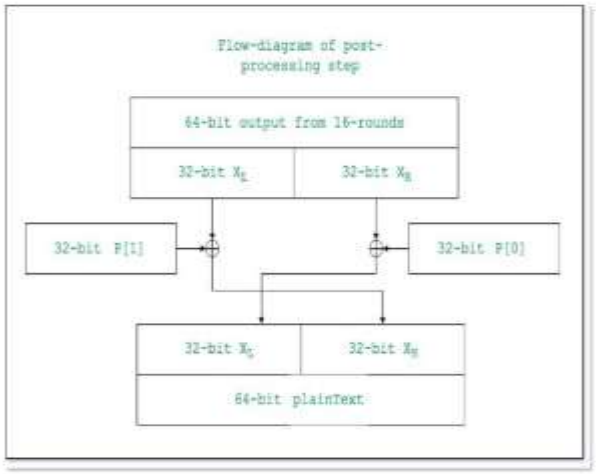


Decryption of Blowfish

In Blowfish, decryption is carried out by reversing the encryption process. Therefore, everything reverses until the ciphertext is converted back into plaintext.

This Blowfish encryption method uses your private key to protect your data. The best thing about Blowfish is that, if the material is encrypted, it can be challenging to decrypt it without the original key. These technologies from the 1990s are getting a little out of date, however, as more complex and secure encryption methods like AES or Twofish-a substitute for Blowfish-are replacing them.





Blowfish encryption and decryption process example

Assume the message "Hi world" needs Blowfish encryption. The following are the steps involved:

1. Initially, the input "**Hi world**" consists of seven characters plus one space, which is equal to 64 bits, or 8 bytes.
2. The input is split into 32 bits. The left 32 bits -- "Hi w" -- are XORed with P1, which is generated by key expansion to create a value called P1 -- P denotes a prime number, a number that is not divisible except by 1 and itself.
3. Then, P1 runs through a transformative F-function (F In) in which the 32 bits are split into 4 bytes each and passed to the four S-boxes.
4. The first two values from the first two S-boxes are added to each other and XORed with the third value from the third S-box.
5. This result is added to the output of the fourth S-box to produce 32 bits as output.
6. The output of F In is XORed with the right 32 bits of the input message -- "orld" -- to produce output F1'.
7. Then, F1' replaces the left half of the message, while P1' replaces the right half.
8. This same process is repeated for successive members of P-array for 16 rounds in total.
9. Finally, after 16 rounds, the outputs P16' and F16' are XORed with the last two entries of the P-array, i.e., P17 and P18. They are then recombined to produce the 64-bit ciphertext of the input message.

Blowfish encryption Application examples

Wondering where you might bump into Blowfish? This speedy cipher is used in zipping up your data, safeguarding your passwords, and email messages. Blowfish has done a little bit of everything, so let's look at some of this encryption method's most popular use cases.

- **Network protocols.** The Blowfish algorithm has been used in network protocols like Secure Shell (SSH) and IPSec to secure internet communications.
- **Password hashing.** Some password managers use the bcrypt password hashing function based on Blowfish. While it is not as popular as AES and other modern encryption algorithms, password managers can use Blowfish for password encryption.
- **File encryption.** Certain file encryption software provides the option to use Blowfish for encryption to prevent unauthorized access.
- **Disk encryption.** Some disk encryption software uses Blowfish to safeguard your sensitive data from digital pirates and snoops.
- **Embedded systems.** Due to its simplicity and speed, Blowfish has been used in some embedded systems, such as devices with limited processing power.
- **Email encryption.** Blowfish has been used in some email encryption software to keep your messages under wraps.

For quite some time, Blowfish has been a digital hero. However, modern encryption methods are slowly pushing it out of the market. Although Blowfish does the job in data encryption, today's cyberspace is rife with threats that require modern and advanced solutions.

Advantages and disadvantages of Blowfish encryption

Besides being fast and efficient, the Blowfish encryption algorithm may not be as safe for your data encryption process as expected. So look at the main advantages and disadvantages of Blowfish yourself and consider the specific needs for your application.

Advantages	Disadvantages
Faster than previous encryption algorithms, such as DES and IDEA.	Less secure and efficient than modern encryption algorithms like AES.
Unpatented and free to use, making it accessible to anyone interested.	The initial setup process can be slow.
Fewer operations to complete compared to other encryption algorithms.	When handling larger quantities of data, the speed decreases significantly.
Generally fast, efficient, and secure.	Speed can be affected when changing keys.
Provides a variable encryption key size, making it more flexible compared to other encryption algorithms.	The small 32 and 64-bit size blocks make Blowfish more vulnerable to brute force attacks.
Compatible with various programming languages.	The key schedule takes a long time, which makes it unusable for some apps.
Supports secure user authentication for remote access.	Does not provide built-in authentication or data integrity checks.

Step-by-Step Example

Given Data

- Plaintext: 0x424C4F57 0x46495348 ("BLOWFISH" in ASCII)
- Key: 0x0123456789ABCDEFFEDCBA9876543210

Step 1: Key Expansion

- The key initializes the P-array and S-boxes.
- The plaintext (0x00000000 0x00000000) is encrypted iteratively to refine these tables.

Step 2: Encryption Rounds

1. Split input into L0 = 0x424C4F57 and R0 = 0x46495348.
2. Compute F(R0) and XOR with L0.
3. Swap halves and repeat for 16 rounds.
4. After 16 rounds, reverse the final swap.

Step 3: Output Ciphertext

After performing all rounds and final XORs, the ciphertext might look like 0x3FA40E8A 0x9D3F6BA5 (values vary based on key expansion).

Is Blowfish encryption safe to use?

Blowfish has been a trusty lock for your sensitive data for years. It's speedy, efficient, and offers variable length encryption, which helps protect your digital assets from brute force attacks. But it's not all sunshine and rainbows. The 64-bit block size can be an obstacle when dealing with heaps of data. So leave these resource-intensive encryption jobs for more modern encryption solutions like AES, offering a block size of 128, 192, and 256 bits.

The Blowfish algorithm can still be the way to go for some tasks. But you might want to adopt AES when it comes to large chunks of data or top-tier security. As always, assessing your needs is the key to picking the right encryption algorithm.