# Twofish Encryption Algorithm

Twofish is a symmetric-key block cipher with a block size of 128 bits and variable-length key of size 128, 192 or 256 bits. This encryption algorithm is optimized for 32-bit central processing units and is ideal for hardware and software environments. It is open source (unlicensed), unpatented and freely available for use. Twofish is similar to an earlier block cipher, Blowfish. It also includes advanced functionalities to replace the Data Encryption Standard (DES) algorithm.

## Understanding Twofish

The Twofish encryption algorithm is an open source symmetric key block cipher developed for the 1997 National Institute of Standards and Technology (NIST) competition to determine the successor to the aging Data Encryption Standard (DES). Twofish was the brainchild of Bruce Schneier, John Kelsey, Chris Hall, Niels Ferguson, David Wagner, and Doug Whiting.

As the name suggests, Twofish is based on Blowfish, an earlier block cipher designed by Schneier. While Blowfish has never been cracked officially, its 64-bit block size was deemed insufficient by the NIST for security. This prompted Schneier's team to base Twofish around 128-bit blocks and make a number of important improvements.

Twofish ultimately lost the 1997 competition to Rijndael, which became known as the Advanced Encryption Standard (AES) and went on to dominate the cryptography space in the coming decades. However, unlike many encryption algorithms of the era, Twofish didn't just fade into history — due the cipher's open source nature and exceptional security, it continues to be used to this day.

Although not considered an advanced encryption standard, Twofish does provide a more efficient and secure alternative to the DES algorithm. One reason is its block size of 128 bits, which makes it resistant to brute-force attacks. Another is its complex key schedule with support for multiple key sizes.

## Key Features of the Twofish Algorithm

Twofish stands out from the other accepted encryption algorithms in several ways. We have summarized the key features of the Twofish algorithm below for your convenience:

- **High block size.**
  It has a block size of 128 bits, making it very resistant to brute force attacks.

- **S-boxes.**
  Twofish uses pre-computed, key-dependent substitution boxes (S-boxes) to obscure the relationship between the encryption/decryption key and the ciphertext. Each S-box consists of three 8-by-8 bit fixed permutations.

- **Feistel network.**
  Twofish is based on the Feistel network structure, just like DES and Blowfish before it. This structure involves dividing input data into equal blocks, then processing each block through multiple encryption rounds. According to Schneier, the Feistel network was chosen because it was thoroughly studied and understood — the developers deliberately chose not to tamper with the formula to avoid introducing vulnerabilities.

- **Complex key schedule.**
  Compared to other encryption algorithms, Twofish has a relatively complex key schedule (the algorithm for deriving round keys from the main key during iterative encryption processes). Twofish encryption always goes through 16 rounds, which makes it resistant to cryptanalysis techniques like related-key attacks and slide attacks.

- **Inherent flexibility.**
  Like AES, Twofish can accept keys of 128, 192, and 256 bits in length, making it suitable for a wide range of security operations. In addition, Twofish is designed to be able to trade key setup time or ROM and RAM for encryption speed, optimizing performance based on the resources it has access to.

- **Open source.**
  The creators of Twofish declined to patent the encryption algorithm — as a result, Twofish has passed into the public domain. There is now a thriving community of developers using Twofish in their work.

## Twofish Architecture

Twofish consists of a number of building blocks, such as the following:

- **Feistel network.**
  A method of transforming any function (F function) into a permutation that forms the basis of many block ciphers.

- **S-boxes.**
  Table-driven, nonlinear substitution operations. Twofish uses four precomputed, key-dependent, bijective, 8-by-8-bit S-boxes. They are commonly used in block ciphers.

- **Maximum distance separable (MDS) matrices.**
  A common feature of Reed-Solomon error-correcting codes. Twofish uses a single 4-by-4 MDS matrix over Galois field ($2^8$) for computations.

- **Pseudo-Hadamard transform (PHT).**
  Simple mixing operation that runs quickly in software. Twofish uses a 32-bit PHT to mix the outputs from its two parallel 32-bit g functions.

- **Whitening.**
  The technique of XORing key material before and after the first and last rounds, respectively. Twofish XORs 128 bits of subkey before the first Feistel round and after the last Feistel round.

- **Key schedule.**
  The means by which key bits are transformed into round keys that the cipher can use. Twofish has a complicated key schedule, which contributes to its security capabilities.

## How does Twofish encryption work?

Like all encryption algorithms, Twofish takes plaintext information (raw data that represents the true values entered, such as the text in a message) and turns it into ciphertext.

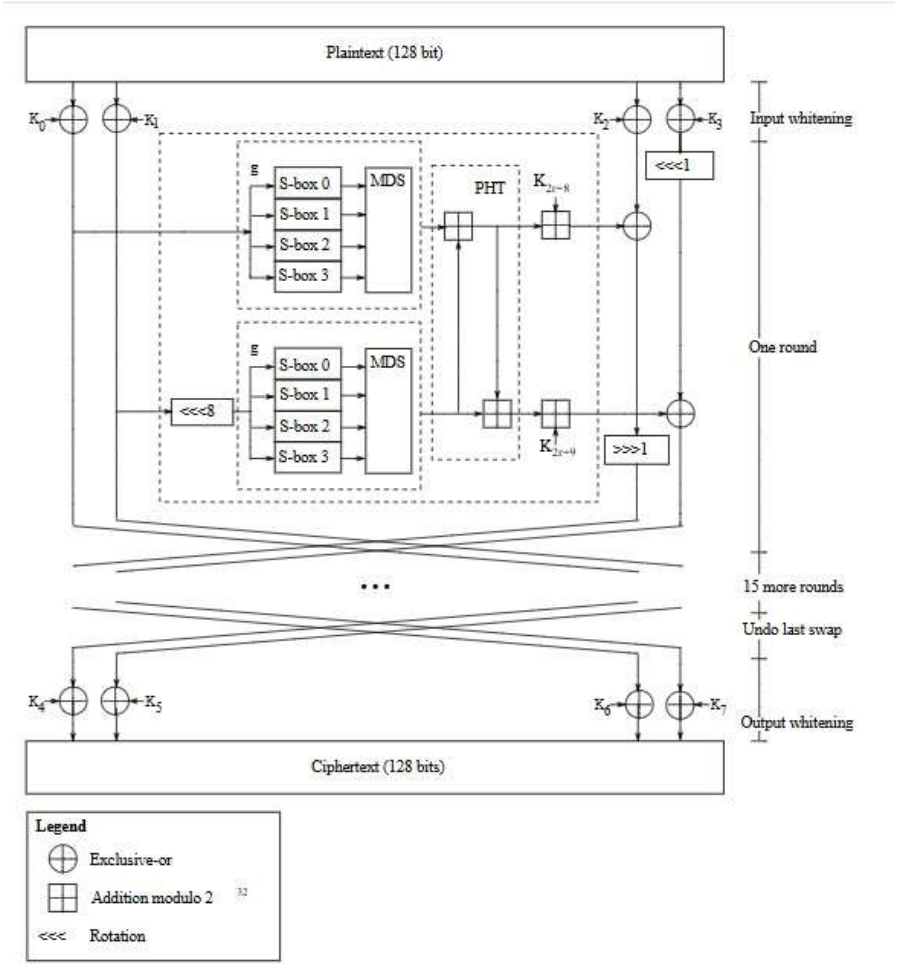
Here is what the Twofish encryption process looks like:

1. When a message is entered, Twofish first breaks the plaintext into equal 128-bit blocks. These 128-bit blocks are then further divided into 32-bit parts.
2. Each 32-bit part undergoes input whitening technique using subkeys generated from the main key. To make the following processes clearer, we'll call these whitened parts R0, R1, R2, and R3.
3. Twofish can now proceed to encrypt the data in rounds. This is what a single Twofish round looks like:
    a) Each round begins with the F function, which uses the leftmost 32-bit box values (in this case, R0 and R1) to encrypt the rightmost 32-bit boxes (R2 and R3):
        - First, Twofish rotates R1 left by 8 bits, yielding RL1.
        - Then, Twofish takes R0 and RL1 and breaks each up into four-byte sections.
        - Each 4-byte section is processed through a different key-dependent S-box.
        - The outputs of these S-boxes are combined back into two 32-bit parts using a Maximum Distance Separable (MDS) matrix. The result of R0 is T0, while the result of RL1 is T1.
        - T0 and T1 are combined using a Pseudo-Hadamard Transform (PHT), which functions like a mixer. This produces a single value, which is replicated twice and combined with two different round keys (based on the key schedule) to yield F0 and F1.
    b) After the F function is complete, F0 is first XORed to R2 and then rotated right by one bit. We'll call the result C2 because it takes the position of R2.
    c) F1 is XORed to R3, but the result is rotated by one bit to the left. This yields C3, which takes the position of R3.

d) While we used R0 and R1 values for the F function, we do not replace them at this point. Instead of the original R0, R1, R2, and R3, we now have R0, R1, C2, and C3.

e) At the end of the round, C2 and C3 switch places with R0 and R1, resulting in the order of C2, C3, R0, and R1. This means that in the next round, C2 and C3 will be subject to the F function and modify the original R0 and R1.

4. Twofish repeats this process until it has gone through 16 rounds of encryption.
5. The resulting 32-bits are subjected to output whitening (again, using subkeys generated from the main key) and combined to produce a 128-bit block of ciphertext.
6. The 128-bit ciphertext blocks are put together to produce the encrypted message.



## How Twofish Works (Repeated again)

Twofish is a 128-bit symmetric block cipher that splits plaintext into four 32-bit words. These words are XORred with four keywords -- a step known as *input whitening*. Sixteen rounds then follow, with the two words on the left used as input to the *g* functions (four byte-wide key-dependent S-boxes followed by linear mixing based on an MDS matrix) in each round.

Next, a PHT combines the results of the two g functions. One of the words on the right is rotated left by 1 bit, while the other is rotated right afterward. Following this, the two results (of adding the two keywords) are XORed into the words on the right. The left and right halves are swapped for the next round. Once all the rounds are complete, the swap of the last round is reversed. This step is known as *output whitening*. Finally, the four words are XORed with four more keywords to produce the ciphertext, whose four words are written as 16 bytes. The same little-endian conversion used for the plaintext is also used to write the ciphertext.

In sum, the encryption process in Twofish includes the following steps:

1. Twofish breaks the input plaintext into 128-bit blocks that are then divided into 32-bit words.
2. In each round, two 32-bit words act as inputs into the F function. This function, a key-dependent permutation on 64-bit values, also takes the round number r used to select appropriate subkeys as input. The subkeys generated from the main key are used to whiten each 32-bit word.
3. Each word is broken up into 4 bytes, which are then sent through four key-dependent S-boxes. These S-boxes have 8-bit input/output (I/O).
4. The MDS matrix combines the 4 output bytes into a 32-bit word.
5. The two 32-bit words are combined using a PHT.
6. Finally, they are added to two round subkeys and XORed into the right half.

## Main Advantages and Disadvantages of Twofish Encryption

- The main advantage of Twofish is its robust security. Twofish is theoretically more resistant to cryptanalysis than AES.

- As previously stated, support for multiple key lengths provides the flexibility needed to implement the algorithm for differing security requirements.

- Another advantage is its democratic nature. Since there is no patent or license, Twofish can be freely implemented by anyone for almost any kind of application.

- On the flip side, Twofish's complexity can be a disadvantage. Since it can be resource-intensive and require a lot of computational overhead, it can be hard to implement the algorithm for low-power devices. Speed can also be a limiting factor when Twofish is implemented in apps with limited resources.

- Also:

| Advantage | Disadvantage |
|---|---|
| Twofish is considered highly secure and has withstood extensive cryptanalysis. | The algorithm is relatively complex, which can make implementation and analysis challenging. |
| Twofish supports key sizes of 128, 192, and 256 bits, providing flexibility based on security requirements. | Twofish might be resource-intensive, making it less suitable for low-power devices or applications with limited computing resources. |
| Twofish is designed for efficient performance in software and hardware implementations. | Although the patent on Twofish has expired, some developers may still be cautious about potential legal issues. |

## The Security of Twofish

With a 128-bit block size and variable-length encryption key, Twofish is one of the most secure encryption protocols. In theory, its high block size means that Twofish is safe from brute-force attacks since such an attack would require tremendous processing power to decrypt a 128-bit encrypted message.


Diagram shows how a side-channel attack works.

The larger size of the encrypted data also makes Twofish secure. When different encryption algorithms are compared in terms of plaintext size after encryption,

Twofish converts 240 KB of plaintext information into a massive 955 KB. Only Blowfish can match Twofish in this aspect. Even the Advanced Encryption Standard (AES) algorithm is limited in that it can convert 250 KB of plaintext into a maximum encrypted size of 847 KB.

The only problem with this large size is that it can slow down program execution, especially if the algorithm is applied to massive quantities of plaintext data.

## Vulnerabilities in Twofish

The Twofish encryption algorithm, while generally considered secure, is not without its potential vulnerabilities:

- **Susceptibility to Side-Channel Attacks**
  Twofish is vulnerable to side-channel attacks, including timing and power analysis attacks. Attackers may exploit information leaked through these channels to gain insights into the cryptographic processes, potentially compromising the algorithm's security.

- **Implementation Challenges**
  Implementing Twofish correctly can be challenging. Errors in the implementation may introduce vulnerabilities that attackers could exploit. The algorithm's complexity requires careful and accurate coding to ensure its secure deployment.

- **Resource Intensiveness**
  Twofish's computational complexity might make it less suitable for low-power devices or applications with limited computing resources. The algorithm's resource-intensive nature could impact the efficiency of its implementation in scenarios where computational overhead needs to be minimized.

Twofish's design aimed to provide a secure and efficient alternative to DES, considering the evolving landscape of cryptographic requirements. While it was not selected as the Advanced Encryption Standard (AES), **Twofish** remains a respected and secure cipher. Its development and participation in the NIST competition contributed to advancing cryptographic algorithms.

## Comparing Twofish to other encryption algorithms

To better understand the advantages and disadvantages of Twofish, let's compare it to two other popular encryption algorithms, its chief competitor AES and its predecessor Blowfish.

## Twofish vs AES

- Twofish and AES both made it to the finals of the 1997 NIST competition, but ultimately, AES was chosen as the future of cybersecurity. In the end, victory was decided by one thing — speed. The future encryption standard would have to deal with continuous streams of data, so even nanosecond delays could quickly add up to interfere with communications.
- The gulf between the two further increased when AMD and Intel began making processors with hardware acceleration for AES. On most modern devices that use AMD or Intel chipsets, AES is orders of magnitude faster than Twofish.
- While in theory Twofish is more secure than AES (for instance, it uses 16 rounds for encryption rather than 10, 12, or 14 in AES), this added security doesn't have much practical effect — both encryption algorithms would take so long to crack that the supercomputers performing the calculations would turn to dust before they were finished.

## Twofish vs Blowfish

- In many ways, Twofish is simply a more advanced version of Blowfish. Bruce Scheneir intended Twofish to become Blowfish's successor, just like AES replaced DES, but history proved otherwise — Blowfish is even more popular than Twofish today, although the disparity is likely the result of Blowfish simply being out for longer.
- Blowfish is faster than many modern encryption algorithms (including Twofish), making it popular for bulk encryption and password storage. Furthermore, despite using only 64-bit blocks for encryption, Blowfish has not been cracked yet. That doesn't mean it's completely safe, however — for example, the small block size may be exploited by birthday attacks. As a result, Schneier recommends using Twofish instead for its enhanced security.

## Twofish Encryption in Use

By leveraging its inherent flexibility and security in areas where speed is not essential, the Twofish encryption algorithm has managed to carve out a niche for itself.

Here are some popular applications of Twofish:

- **Password managers.**
  Twofish is a natural fit for password storage solutions. First, password managers prize security above all else, taking advantage of Twofish's complex key schedule and S-blocks. Second, the fact that password managers are only used to access credentials means that the slower speed of Twofish isn't as noticeable.

- **Virtual private networks.**
  Some virtual private networks (VPNs) use Twofish to encrypt your internet connection, although the top brands typically prefer AES-256 due to the latter's speed. Ultimately, the choice of encryption algorithm depends on many factors, including the provider's own infrastructure and the VPN protocol used.

- **OpenPGP (Pretty Good Privacy).**
  PGP is a tool for electronically signing and encrypting files, directions, and emails. True to its name, OpenPGP is an open source standard that helps implement PGP in software. One popular OpenPGP implementation that uses Twofish is the GNU Privacy Guard (GnuPG), which focuses on user communications data.

- **TrueCrypt.**
  TrueCrypt was a free file encryption tool that used Twofish and other secure ciphers to protect data. The original TrueCrypt was officially discontinued in 2014, although an independent audit in 2015 determined that there were no significant flaws with the software. VeraCrypt, one of TrueCrypt's successors, also uses Twofish encryption in its operations.

## Is Twofish encryption secure to use?

**Yes**, Twofish encryption is secure to use. However, like all encryption algorithms, they are only a part of the larger cybersecurity ecosystem and may be defeated by something as simple as human error. Several attacks have targeted Twofish systems in the past, but, according to Bruce Schneier, they were not practical breaks in cryptanalysis.

## Has Twofish been cracked?

**No**, Twofish has not been cracked yet, despite extensive cryptanalysis attempts. In theory, the Twofish algorithm may be susceptible to side-channel attacks, cyberattacks using observable system information (such as power consumption or timing) due to its reliance on precomputed S-boxes, but no successful side-channel attack on Twofish has been carried out yet.