Blowfish

Sankeeth Kumar Chinta.
# 991730264
Sept 18, 2015

Contents

1 Introduction

In cryptographic circles, plaintext is the message you're trying to transmit. That message could be a medical test report, a firmware upgrade, or anything else that can be represented as a stream of bits. The process of encryption converts that plaintext message into ciphertext, and decryption converts the ciphertext back into plaintext. Encryption algorithms are technically classified in two broad categories- Symmetric key Cryptography and Asymmetric Key Cryptography.

In symmetric type of Cryptography, the key that is used for encryption is same as the key used in decryption. So the key distribution has to be made before transmission of any information. The key plays a very important role in this cryptography since the nature of the key length has an impact on security. Examples of various symmetric key algorithms are Data encryption standard(DES), Triple DES, Advanced Encryption Standard(AES) and Blowfish Encryption Algorithm.

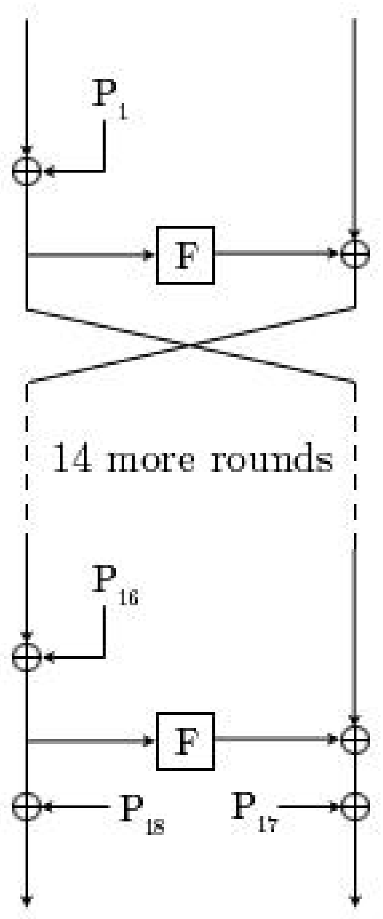
In Asymmetric Cryptography, two unique keys are used for encryption and decryption. One is public and the other one is private. The public key is available to anyone on the network i.e., whoever wants to encrypt the plaintext should know the Public Key of receiver. Decryption of the cipher text through his/her own private key is by authorizes personnel. Private Key is kept secretly from the others. Symmetric Encryption Algorithm is much faster as compared to Asymmetric key algorithm. The memory requirement of Symmetric algorithm is comparatively less than asymmetric algorithm. Eg: RSA

DES (Data Encryption Standard) is the first encryption standard recommended by NIST (National Institute of Standards and Technology). It was developed by IBM in 1974 and in 1977 it was adopted as a national standard.DES is a 64-bit block cipher under 56-bit key. The algorithm computes with a permutation of sixteen rounds block cipher and a final permutation. DES application is popularly used in the commercial, military, and other domains. DES standard is public.

In the year 2000, AES was invented by scientists named Joan and Vincent Rijmen. AES makes use of the Rijndael block cipher, Rijndael key and block length can be of 128, 192 or 256-bits, If both the key-length and block length are 128-bit then Rijndael performs 9 processing rounds. If the block/key is 192-bit, it will perform 11 processing rounds and in the same way for 256-bits, Rijndael performs 13 processing rounds..

Block Cipher is a deterministic algorithm operating on fixed-length groups of bits, called blocks, with an unvarying transformation that is specified by a symmetric key. Block ciphers are important elementary components in the design of many cryptographic protocols. The modern design of block ciphers is based on the concept of an iterated product cipher. Iterated product ciphers carry out encryption in multiple rounds, each of which uses a different sub-key derived from the original key. One widespread implementation of such ciphers is called a Feistel network.

One of the most popular fiestel network cipher is Blowfish. Blowfish is a symmetric-key block cipher proposed as a new encryption standard. It is a 16- round fiestel system, which uses large key-dependent S-boxes and iterates a simple encryption 16 times. The block size is 64 bits and the key-length may varies from 32 bits upto 448bits.Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date.



2 History

DES is the factotum of cryptography algorithms which was a long past time to replace the old standard. In DES, the length of the key is too small for today emerging technologies. The world just incompletely trusted DES in light of the fact that it survived the investigation of the NSA. Specialists trusted DES on the grounds that it was a distributed standard, and in light of the fact that it survived 20 years of escalated cryptanalysis by cryptographers around the globe. Cryptography is similar to that: trust in an algorithm develops as many groups tries to break it and fails Contenders for a substitution are rising, yet none has taken far reaching hold. Triple-DES is the moderate methodology; IDEA (utilized as a part of PGP) is the most encouraging new algorithm. What's more, there is a gathering of unpatented additionally rans: RC4 (once a prized formula of RSA Data Security, Inc. be that as it may, now openly accessible on the Internet), SAFER, and Blowfish. In 1993, Scheneir first presented Blowfish at the Cambridge Algorithms Workshop ("Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)," Fast Software Encryption, R. Anderson, ed., Lecture Notes in Computer Science 809, Springer-Verlag, 1994) and in Dr. Dobb's Journal (April 1994). From the start Blowfish was intended to be a completely free–unpatented, unlicensed, and uncopyrighted–alternative to DES. Since then it has been analyzed by some people and has started to see use in some systems, both public and private.

Since 1993, it was analysed considerably and gaining acceptance as a strong encryption algorithm. The first implementation of blowfish was done in LabVIEW.. This was proposed as the world needs a new encryption standard as the workhorse encryption algorithm is near ending of its useful life.

3 Description of Algorithm:

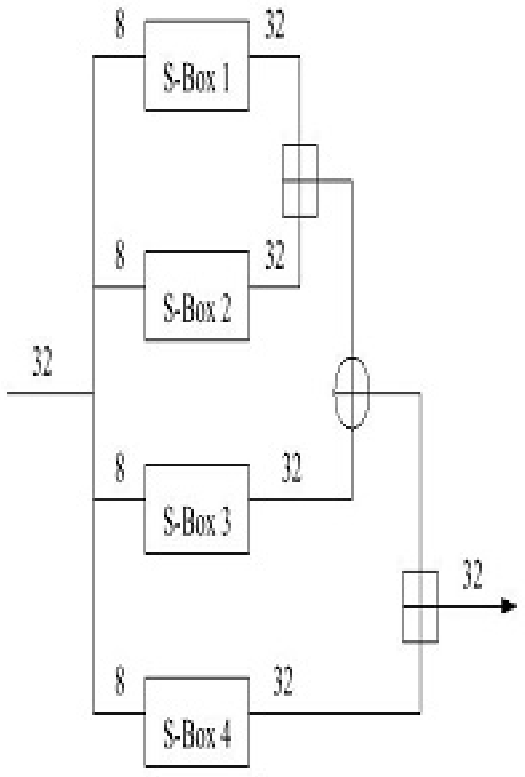
Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time. The algorithm follows fiestal network and is divided into 2 main parts:

1. Key-expansion
2. Data Encryption
3. Data Decryption

3.1 Key Expansion

Prior to any data encryption and decryption, these keys should be computed before-hand.

The p-array consists of 18, 32-bit sub-keys:
P1, P2,., P18
Four 32-bit S-Boxes consist of 256 entries each:
S1, 0, S1, 1,. S1, 255
S2, 0, S2, 1,.. S2, 255
S3, 0, S3, 1,.. S3, 255
S4, 0, S4, 1 ....S4, 255



Generating the Sub-keys:
The sub-keys are calculated and generated using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.

2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)

3. Encrypt the all-zero string with the Blowfish algorithm, using the sub-keys described in steps (1) and (2).

4. Replace P1 and P2 with the output of step (3).

5. Encrypt the output of step (3) using the Blowfish algorithm with the modified sub-keys.

6. Replace P3 and P4 with the output of step (5).

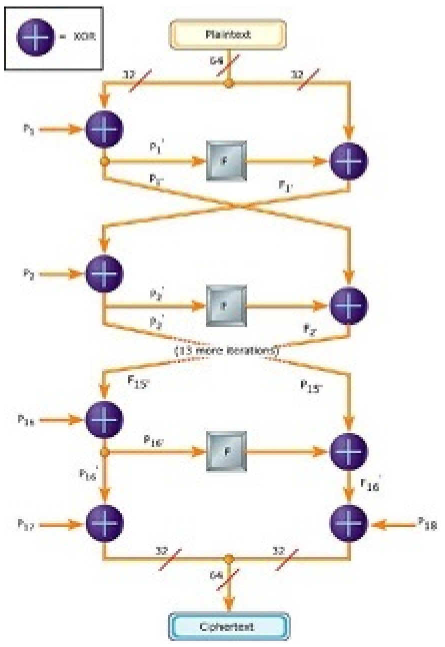
7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub-keys. Applications can store the sub-keys rather than execute this derivation process multiple times.

3.2 Data Encryption

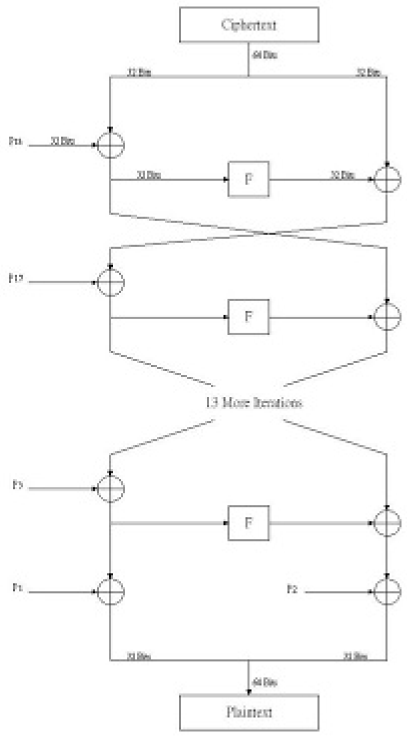
It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.

```
Algorithm: Blowfish Encryption
Divide x into two 32-bit halves: xL, xR
For i = 1to 16:
xL = XL XOR Pi
xR = F(XL) XOR xR
Swap XL and xR
Swap XL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR
```



3.3 Data Decryption

Decryption is exactly the same as encryption, except that P1, P2 ..... P18 are used in the reverse order.



4 Design Decisions

A 64-bit block size yields a 32-bit word size, and maintains block-size compatibility with existing algorithms. Blowfish is easy to scale up to a 128-bit block, and down to smaller block sizes. The fundamental operations were chosen with speed in mind. XOR, ADD, and MOV from a cache are efficient on both Intel and Motorola architectures. All sub-keys fit in the cache of a 80486, 68040, Pentium, and PowerPC. The Feistel Network that makes up the assemblage of Blowfish is intended to be as straightforward as would be prudent, while as yet holding the desirable cryptographic properties of the structure. In calculation outline, there are two fundamental approaches to guarantee that the key is sufficiently long to guarantee a specific security level. One is to painstakingly plan the algorithm so that the whole entropy of the key is safeguarded, so there is no better approach to cryptanalyze the algorithm other than brute force. The other is to plan the algorithm with such a variety of key bits that attacks that diminish the powerful key length by a few bits are unimportant. Since Blowfish is intended for huge microprocessors with a lot of memory, the recent has been picked. Be that as it may, it works just as well on Handheld frameworks with a better than average microprocessors. The sub key generation procedure is intended to save the whole entropy of the key and to convey that entropy consistently all through the Sub-keys. It is likewise intended to appropriate the arrangement of permitted Sub-keys arbitrarily all through the domain of Sub-keys. The digits of pi were picked as the starting Sub-key table for two reasons: on the grounds that it is an random grouping not identified with the algorithm, and in light of the fact that it could either be put away as a major aspect of the algorithm or derived when required. In the Sub-key generation prepare, the Sub-keys change somewhat with each pair of Sub-keys created. This is primarily to protect against any attacked of the Sub-key generation process that exploit the fixed and known Sub-keys. It also reduces storage requirements. The 448 limit on the key size ensures that the every bit of every Sub-key depends on every bit of the key. The key bits are over and over XORed with the digits of pi in the starting P-array to keep the accompanying potential attack: Assume that the key bits are not rehashed, but rather padded with zeros to stretch out it to the length of the P-array. An attacker may discover two keys that vary just in the 64-bit value XORed with P1 and P2 that, utilizing the initial known sub-keys, produce the same encrypted value. Assuming this is the case, he can discover two keys that deliver all the same sub-keys. This is an exceedingly enticing assault for a malignant key generator. To keep this same kind of attack, the starting plain text value in the sub-key generation procedure is fixed. The sub-key generation algorithm does not expect that the key bits are arbitrary. Indeed, even exceptionally related key bits, for example, an alphanumeric ASCII string with the bit of each byte set to 0, will deliver random sub-keys. On the other hand, to create sub-keys with the same entropy, a more longer alphanumeric key is required. The time consuming sub-key generation procedure includes significant unpredictability for a brute- force attack. The sub-keys are too long to be put away on a massive tape, so they would need to be produced by a brute-force breaking machine as required. A total of 522 iterations of the encryption calculation are required to test a single key, successfully adding 29 stages to any brute-force attack.

5 Areas Of Applications

A standard encryption algorithm must be suitable for many different applications:

1. Bulk encryption: The algorithm should be efficient in encrypting data files or a continuous data stream.

2. Random bit generation: The algorithm should be efficient in producing

single random bits.

3. Packet encryption: The algorithm should be efficient in encrypting packet-sized data. (An ATM packet has a 48- byte data field.) It should implementable in an application where successive packets may be encrypted or decrypted with different keys.

4. Hashing: The algorithm should be efficient in being converted to a one-way hash function.

5.1 Products Using Blowfish Algorithm

1. Blowfish Advanced CS by Markus Hahn: File encryption and wipe utility for all Win32 systems. File browser, job automation, auto password confirmation, secure key setup with SHA-1, and data compression with LZSS. Uses Blowfish, Twofish, and Yarrow. Open source.

2. Access Manager by Citi-Software Ltd: A password manager for Windows. Free for personal use.

3. AEdit: A free Windows word processor incorporating text encryption.

6 Conclusion

The proposed algorithm of the Blowfish can achieve an efficiency of data encryption up to 4 bits per clock. We avoid I/O limited constraint by changing the I/O from 64 bits to 16 bits in this design. The proposed architecture ought to fulfill the need of high data encryption and can be connected to different devices separately. Blowfish algorithm, it is a variable-length key block cipher. Applications where there is a strong communication link and where the key will not be changed too often there we will be using the Blowfish algorithm It is quicker than DES. Blowfish is a 16 pass block encryption algorithm that can be never broken. BLOWFISH is used frequently because:

1. It is fast as it encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte.

2. It is compact as it can run in less than 5K of memory.

3. It simply uses addition, XOR, lookup table with 32-bit operands.

4. It is secure as the key length is variable ,it can be in the range of 32 448 bits: default 128 bits key length.