Tech Juris

A White Paper prepared for
Asian School of Cyber Laws
www.asianlaws.org

The Blowfish algorithm

ASCL White Papers can be downloaded from:
www.asianlaws.org/whitepapers

Tech Juris
The Technology Lawyers
www.techjuris.com

ASCL-WP-IS-Tec-Algo-03

The Blowfish program was developed by author and computer security and cryptography consultant Bruce Schneier. Blowfish is a cipher based on Feistel rounds, and the design of the f-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software. The block ciphers Khafre and CAST have somewhat similar rounds.

The main claim to fame of Blowfish, however, is its method of key scheduling. The round keys, and the entire contents of all the S-boxes, are created by multiple iterations of the block cipher. This enhances the security of the block cipher, since it makes exhaustive search of the keyspace very difficult, even for short keys.

Description of Blowfish

Unlike DES, Blowfish applies the f-function to the *left* half of the block, obtaining a result XORed to the *right* half of the block. This departure from convention may cause confusion in reading the description of Blowfish. However, upon further reflection, it is really DES that is creating confusion; the time sequence of events should move from left to right (particularly in a design that is otherwise big-endian); this is generally what happens in more recent designs, such as the AES candidates, and particularly in ciphers with unbalanced Feistel rounds.

Blowfish consists of sixteen rounds. For each round, first XOR the left half of the block with the sub key for that round. Then apply the f-function to the left half of the block, and XOR the right half of the block with the result. Finally, after all but the last round, swap the halves of the block. There is only one sub key for each round; the f-function consumes no sub keys, but uses S-boxes, which are key dependent.

After the last round, XOR the *right* half with subkey 17, and the *left* half with subkey 18.

The f-function

Blowfish uses four S-boxes. Each one has 256 entries, and each of the entries is 32 bits long. To calculate the f-function: use the first byte of the 32 bits of input to find an entry in the first S-box, the second byte to find an entry in the second S-box, and so on. The value of the f-function is (S1(B1) + S2(B2)) XOR (S3(B3) + S4(B4)) where addition is performed modulo 2^32.

Decryption

Decryption is the same as encryption, with the 18 sub keys used in reverse order. At first, this seems unbelievable (although not quite as bad as understanding the decryption of IDEA), because there are two XOR operations following the last use of the f-function, and only one preceding the first use of the f-function.

However, if you modify the algorithm so that use of sub keys 2 through 17 takes place before the output of the f-function is XORed to the right half of the block, and is done to the same data before that XOR, although that means it is now on the right half of the block, since the XOR of the sub key has been moved before a swap of the halves of the block, you have not actually changed anything since the same information is XORed to the left half of the block between each time it is used as input to the f-function.

In fact, you can even move the XOR still earlier, before the preceding swap of block halves. Once you do that, you have the exact reverse of the decryption sequence.

Sub key generation

Begin by initializing sub keys 1 through 18, followed by elements zero through 255 of the first S box, then elements zero through 255 of the second S box, all the way to element 255 of the fourth S box, with the fractional part of pi. The most significant bit of the fractional part of pi becomes the most significant bit of the first sub key.

Then, take the key, which may be of any length up to 72 bytes, and, repeating it as often as necessary to span the entire array of 18 sub keys, XOR it with the sub key array contents. Then execute the Blowfish algorithm repeatedly, with an initial input of a 64-byte block of all zeroes as plaintext input. After each execution, replace part of the sub keys or S boxes with the successive outputs of Blowfish, in the same order as the digits of pi in binary (or hexadecimal) form were placed in them; after the first iteration, replace sub keys 1 and 2; after the tenth iteration, replace the first two entries (0 and 1) in S-box 1; and so on.

For each iteration of Blowfish in key generation, also use the output of the preceding iteration as input. (The original description of Blowfish in the April 1994 issue of *Dr. Dobb's Journal* could be interpreted to imply that zero should be used as input for every iteration. As the later iterations only change individual S-box entries, this could lead to large stretches of identical data in the S-boxes, and is thus a misreading of the directions, not a slightly different original form of the algorithm.)

Thus, loading Blowfish with a new key takes as much time as encrypting 521 blocks ((256*4+18)/2) in Blowfish. This gives Blowfish an extra 9 bits of security against a brute force search for keys shorter than maximum length, which makes the 32-bit, instead of 40-bit limit for export versions of Blowfish reported by one individual, just about exactly right.

Asian School of Cyber Laws

Pune | Bangalore | Kolkata | Nagpur

OUR SERVICES

Information Security

- Training
- Consultancy
- White papers
- Workshops

Technology Law

We provide training, consultancy, workshops, and white papers in the following areas of law:

- Media Laws
- Semi-conductor Law
- Intellectual Property Law
- PKI Law
- Cyber Law
- Drafting
- Software valuation
- Audits
- Arbitration
- E-contracts

In addition, we conduct a Diploma course in Information Technology Law.

Cyber Crime Investigation

- Training
- Consultancy
- Search and seizure operations
- White papers
- Certified Courses

CONTACT US

Regd. Office

6, Rajas, Above IDBI, Pashan Road, Pune 411008
Ph: 91 20 5890894 / 95
Fax: 91 20 5675600

Email: info@asianlaws.org
URL: www.asianlaws.org