A Survey on Different Secret Key Cryptographic Algorithms

Harshala B. Pethe[1] and Dr. S. R. Pande[2]

[1]*Department of Electronics & Comp. Sc. RTMNU, Nagpur (India).*

[2]*Department of Computer Science, SSESA's Science College, Nagpur (India)*

ABSTRACT

Today's internet world is very competitive and to survive in such a competitive world there must be a secure environment to communicate. Internet and network applications are growing fast day by day. For this purpose there is a requirement of an efficient and strong algorithm which will provide strong encryption. This paper presents a detailed study of various secret key cryptographic algorithms.

Key words: Cryptographic algorithms, Encryption, Secret key

Introduction

The art and science of keeping messages secure is cryptography and it is practiced by cryptographers. Cryptanalysts are practitioners of cryptanalysis, the art and science of breaking ciphertext; that is, seeing through the disguise. The branch of mathematics encompassing both cryptography and cryptanalysis is **cryptology** and its practitioners are **cryptologists**. The goals of cryptography are:

1) Confidentiality: Keeping messages secret.
2) Origin authentication: verifying the message's source.
3) Integrity: assuring that a message has not been modified.
4) Key management: distributing the secret "keys" for cryptographic algorithms.

Secret key encryption is also referred to as conventional encryption or single key encryption, was the only type of encryption in use prior to the development of public key encryption (Surya, Diviya) (Kaliski 1993).

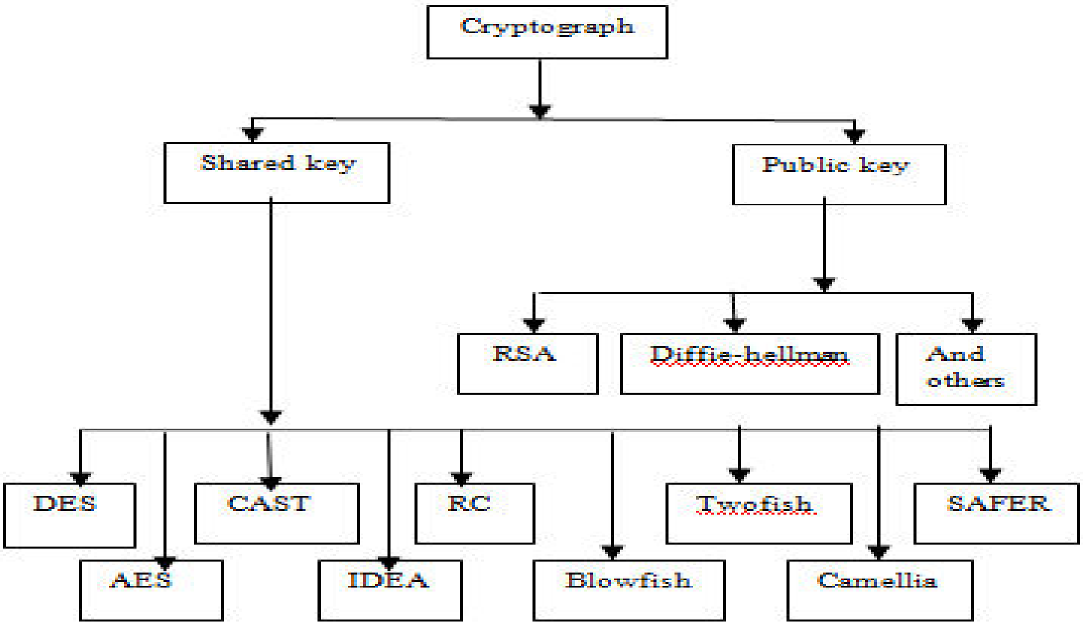

Chart-1: Symmetric key encryption

The shared or symmetric key encryption is a cryptography technique that uses a shared secret key to encrypt and decrypt the data. Symmetric encryption algorithms are very efficient at processing large amounts of information and computationally less intensive than asymmetric encryption algorithms.

There are two types of symmetric encryption algorithms: stream ciphers and block ciphers which provide bit-by-bit and block encryption respectively.

Overview of Algorithms

Secret key cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit at a time and implement some form of feedback mechanism so that the key is constantly changing. A block cipher is so called because the scheme encrypts one block of data at a time using the same key on each block.

Block ciphers can operate in one of several modes.

The following are the four most important modes:

A. Electronic Codebook (ECB): This mode is the simplest and most obvious application. The secret key is used to encrypt the plaintext block to form a ciphertext block. Two identical plaintext blocks, then, will always generate the same ciphertext block. Although, this is the most common mode of block ciphers, it is vulnerable to a variety of brute-force attacks.

B. Cipher Block Chaining (CBC): This mode adds a feedback mechanism to the encryption scheme. In CBC, the plaintext is exclusively–ORed (XORed) with the previous ciphertext block prior to encryption. In this mode, two identical blocks of plaintext never encrypt to the same ciphertext.

C. Cipher Feedback (CFB): This mode is a block cipher implementation as a self-synchronizing stream cipher. CFB mode allows data to be encrypted in units smaller than the block size, which might be useful in some applications such as encrypting interactive terminal input. If we were using 1-byte CFB mode, for example, each incoming character is placed into a shift register the same size as the block, encrypted, and the block is transmitted. At the receiving side, the ciphertext is decrypted and the extra bits in the block (i.e., everything above and beyond the one byte) are discarded.

D. Output Feedback (OFB) mode is a block cipher implementation conceptually similar to a synchronous stream cipher. OFB prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that is independent of both the plaintext and ciphertext bit streams (Marcella, Menendez,2008).

Secret Key Cryptographic Algorithms

Following are some most common secret key cryptographic algorithms

Data Encryption Standard (DES):

It is a Feistel-type Substitution-Permutation Network (SPN) cipher, specified in FIPS PUB 46. The result of a 1970s effort to produce a U.S. encryption standard. DES uses a 56-bit key which can be broken using brute-force methods, and is now considered obsolete. A 16 cycle Feistel system is used, with an overall 56-bit key permuted into 16 48-bit sub keys, one 154 for each cycle. To decrypt, the identical algorithm is used, but the order of sub keys is reversed. The L and R blocks are 32 bits each, yielding an overall block size of 64 bits. The hash function "f", specified by the standard using the so-called "S-boxes", takes a 32-bit data block and one of the 48-bit sub keys as input and produces 32 bits of output. Sometimes DES is said to use a 64-bit key, but 8 of the 64 bits are used only for parity checking, so the effective key size is 56 bits [9]. Since the time DES was adopted (1977), it has been widely speculated that some kind of backdoor was designed into the cryptic S-boxes, allowing those "in the know" to effectively crack DES. Time has proven such speculation idle. Regardless of any backdoors in the hash function, the rapid advances in the speed of electronic circuitry over the last 20 years, combined with the natural parallelism of Feistel ciphers and DES's relatively small key size, have rendered the algorithm obsolete. In 1998, the Electronic Frontier Foundation built a DES Cracker for less than $250,000 that can decode DES messages in less than a week.

Triple DES

Triple DES was developed to address the obvious flaws in DES without designing a whole new cryptosystem. Triple DES simply extends the key size of DES by applying the algorithm three times in succession with three different keys. The combined key size is thus 168 bits (3 times 56), beyond the reach of brute-force techniques such as those used by the EFF DES Cracker. Triple DES has always been regarded with some suspicion, since the original algorithm was never designed to be used in this way, but no serious flaws have been uncovered in its design, and it is today available cryptosystem used in a number of Internet protocols (Kumar, Thakur, Kalia, 2011).

Advanced encryption standard (aes) / rijndael

In the late 1990s, the U.S. National Institute of Standards and Technology (NIST) conducted a competition to develop a replacement for DES. The winner, announced in 2001, is the Rijndael algorithm, destined to become the new Advanced Encryption Standard. Rijndael mixes up the SPN model by including Galios field operations in each round. Somewhat similar to RSA modulo arithmetic operations, the Galios field operations produce apparent gibberish, but can be mathematically inverted. AES have Security is not an absolute; it's a relation between time and cost. Currently, there are speculations that military intelligence services possibly have the technical and economic means to attack keys equivalent to about 90 bits, although no civilian researcher has actually seen or reported of such a capability. Actual and demonstrated systems today, within the bounds of a commercial budget of about 1 million dollars can handle key lengths of about 70 bits. An aggressive estimate on the rate of technological progress is to assume that technologies will double the speed of computing devices every year at an unchanged cost. If correct, 128-bit keys would be in theory be in range of a military budget within 30-40 years. An illustration of the current status for AES is given by the following example, where we assume an attacker with the capability to build or purchase a system that tries keys at the rate of one billion keys per second. This is at least 1 000 times faster than the fastest personal computer in 2004..The key length should thus be chosen after deciding for how long security is required, and what the cost must be to brute force a secret key. In some military circumstances a few hours or days security is sufficient – after that the war or the mission is completed and the information uninteresting and without value. In other cases a lifetime may not be long enough. There is currently no evidence that AES has any weaknesses making any attack other than exhaustive search, i.e. brute force, possible. Even AES-128 offers a sufficiently large number of possible keys, making an exhaustive search impractical for many decades, provided no technological breakthrough causes the computational power available to increase dramatically and that theoretical research does not find a short cut to bypass the need for exhaustive search. There are many pitfalls to avoid when encryption is implemented, and keys are generated. It is necessary to ensure each and every implementations security, but hard since it requires careful examination by experts. An important aspect of an evaluation of any specific implementation is to determine that such an examination has been made, or can be conducted.

If both block length and key length are 128 bits, AES will perform 9 processing rounds. If the block and key are 192 bits, AES will perform 11 processing rounds. If the block and key are 256 bits, then it performs 13 processing rounds. Each processing rounds involves four steps:

1. Substitute bytes: Uses an S-box to perform a byte by byte substitution of the block.
2. Shift rows: A simple permutation.
3. Mix column: A substitution method where data in each column from the shift row is multiplied by the algorithm's matrix.
4. Add round key: The key for the processing round is XORed with the data [4].

CAST: CAST was designed in Canada by **C**arlisle **A**dams and **S**tafford **T**avares. They claim that the name refers to their design procedure and should conjure up images of randomness, but note the authors' initials. The example CAST algorithm uses a 64-bit block size and a 64-bit key.

The structure of CAST should be familiar. The algorithm uses six S-boxes with an 8-bit input and a 32-bit output. Construction of these S-boxes is implementation-dependent and complicated CAST-128, is a DES–like substitution-permutation crypto algorithm, employing a 128-bit key operating on a 64-bit block. CAST-256 is an extension of CAST-128, using a 128-bit block size and a variable length (128, 160, 192, 224, or 256 bit) key. CAST is named for its developers, Carlisle Adams and Stafford Tavares and is available internationally. CAST-256 was one of the Round 1 algorithms in the AES process.

International Data Encryption Algorithm (IDEA)

International Data Encryption algorithm (IDEA) is a block cipher algorithm designed by Xuejia Lai and James L. Massey of ETH-Zürich and was first described in 1991.The original algorithm went through few modifications and finally named as International Data Encryption Algorithm (IDEA). The mentioned algorithm works on 64-bit plain text and cipher text block (at one time). For encryption, the 64-bit plain text is divided into four 16 bits sub-blocks. In our discussion, we denote these four blocks as P1 (16 bits), P2 (16 bits), P3 (16 bits) and P4 (16 bits). Each of these blocks goes through 8 ROUNDS and one OUTPUT TRANSFORMATION phase. In each of these eight rounds, some (arithmetic and logical) operations are performed. Throughout the eight ROUNDS, the same sequences of operations are repeated. In the last phase, i.e., the OUTPUT TRANSFORMATION phase, we perform only arithmetic operations. At the beginning of the encryption process, the 64 bit plain text is divided in four equal size blocks and ready for ROUND1 input. The output of ROUND1 is the input of ROUND2. Similarly, the output of ROUND2 is the input of ROUND3, and so on. Finally, the output of ROUND8 is the input for OUTPUT TRANSFORMATION, whose output is the resultant 64 bit cipher text (assumed as C1 (16bits), C2 (16 bits), C3 (16 bits) and C4 (16 bits)). As the IDEA is a symmetric key algorithm, it uses the same key for encryption and for decryption. The decryption process is the same as the encryption process except that the sub keys are derived using a different algorithm. The size of the cipher key is 128bits. In the entire encryption process we use total 52 keys (ROUND1 to ROUND8 and OUTPUT TRANSFORMATION phase); generated from a 128 bit cipher key. In each round (ROUND1 to ROUND8) we use six sub keys. Each sub-key consists of 16bits. And the OUTPUT TRANSFORMATION uses 4 sub-keys (Basu 2011).

Rivest Cipher (RC)

Named for Ron Rivest, a series of SKC algorithms.

RC1: Designed on paper but never implemented.

RC2: A 64-bit block cipher using variable-sized keys designed to replace DES. Its code has not been made public, although many companies have licensed RC2 for use in their products. RC2 is designed by Ron Rivest for RSA Data Security, Inc. (RSADSI). Apparently, “RC” stands for “Ron’s Code” although it officially stands for “Rivest Cipher.” (RC3 was broken at RSADSI during development; RC1 never got further than Rivest’s notebook.) It is proprietary, and its details have not been published. Don’t think for a minute that this helps security. RC2 has already appeared in commercial products. RC2 has not been patented and is only protected as a trade secret. The two operations are “mix” and “mash,” and one is chosen in each round. RC2 is not an iterative block cipher. This suggests that RC2 offers more protection against differential and linear cryptanalysis than other block ciphers which have relied for their security on copying the design of DES [8].

RC3: Found to be breakable during development.

RC4: A stream cipher using variable-sized keys; it is widely used in commercial cryptography products, although it can only be exported using keys that are 40 bits or less in length. RC4 is a stream cipher, symmetric key algorithm. The same algorithm is used for both encryption and decryption as the data stream is simply XORed with the generated key sequence. The key stream is completely independent of the plaintext used. It uses a variable length key from 1 to 256 bit to initialize a 256-bit state table. The state table is used for subsequent generation of pseudo-random bits and then to generate a pseudo-random stream which is XORed with the plaintext to give the ciphertext.

The algorithm can be broken into two stages: initialization, and operation. In the initialization stage the 256-bit state table, **S** is populated, using the key, **K** as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted.

The steps for RC4 encryption algorithm is as follows:

1- Get the data to be encrypted and the selected key.

2- Create two string arrays.

3- Initiate one array with numbers from 0 to 255.

4- Fill the other array with the selected key.

5- Randomize the first array depending on the array of the key.

6- Randomize the first array within itself to generate the final key stream.

7- XOR the final key stream with the data to be encrypted to give cipher text. (Mousa, Hamad, 2006)

RC5: A block cipher supporting a variety of block sizes, key sizes, and number of encryption passes over the data. RC5 should be a symmetric block cipher. The same secret cryptographic key is used for encryption and for decryption. The plaintext and cipher text are fixed length bit sequences (blocks) (Rivest 1997).

RC6: An improvement over RC5, RC6 was one of the AES Round 2 algorithms.

BLOWFISH: A symmetric 64-bit block cipher invented by Bruce Schneier; optimized for 32-bit processors with large data caches, it is signifi cantly faster than DES on a Pentium or Power PC-class machine. Key lengths can vary from 32 to 448 bits in length [4]. Blowfish, available freely and intended as a substitute for DES or IDEA, is in use in over 80 products. Blowfish is an algorithm of my own design, intended for implementation on large microprocessors. The algorithm is unpatented.

Blowfish is designed to meet the following design criteria.

1. Fast. Blowfish encrypts data on 32-bit microprocessors at a rate of 26 clock cycles per byte.
2. Compact. Blowfish can run in less than 5K of memory.

3. Simple. Blowfish uses only simple operations: addition, XORs, and table lookups on 32-bit operands. Its design is easy to analyze which makes it resistant to implementation errors.
4. Variably Secure. Blowfish's key length is variable and can be as long as 448 bits. Blowfish is optimized for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than DES when implemented on 32-bit microprocessors with large data caches, such as the Pentium and the PowerPC. Blowfish is not suitable for applications, such as packet switching, with frequent key changes, or as a one-way hash function. Its large memory requirement makes it infeasible for smart card applications (Kumar, Thakur, Kalia, 2011).

Twofish: A 128-bit block cipher using 128-, 192-, or 256-bit keys. Designed to be highly secure and highly flexible, well-suited for large microprocessors, 8-bit smart card microprocessors, and dedicated hardware and was one of the Round 2 algorithms in the AES process.

Camellia : A secret-key, block-cipher crypto algorithm developed jointly by Nippon Telegraph and Telephone (NTT) Corp. and Mitsubishi Electric Corporation (MEC) in 2000. Camellia has some characteristics in common with AES: a 128-bit block size, support for 128-, 192-, and 256-bit key lengths, and suitability for both software and hardware implementations on common 32-bit processors as well as 8-bit processors (e.g., smart cards, cryptographic hardware, and embedded systems). Camellia was jointly developed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation in 2000. Camellia specifies the 128-bit block size and 128-, 192-, and 256-bit key sizes, the same interface as the Advanced Encryption Standard (AES). Camellia is characterized by its suitability for both software and hardware implementations as well as its high level of security. From a practical viewpoint, it is designed to enable flexibility in software and hardware implementations on 32-bit processors widely used over the Internet and many applications, 8-bit processors used in smart cards, cryptographic hardware, embedded systems, and so on. Moreover, its key setup time is excellent, and its key agility is superior to that of AES. Camellia has been scrutinized by the wide cryptographic community during several projects for evaluating crypto algorithms. In particular, Camellia was selected as a recommended cryptographic primitive by the EU NESSIE (New European Schemes for Signatures, Integrity and Encryption) and also included in the list of cryptographic techniques for Japanese e-Government systems which were selected by the Japan CRYPTREC (Cryptography Research and Evaluation Committees.

Secure and Fast Encryption Routine (SAFER): Secret-key crypto scheme designed for implementation in software. Versions have been defined for 40-, 64-, and 128-bit keys. SAFER K-64 stands for Secure and Fast Encryption Routine with a Key of 64 bits. The government of Singapore is planning to use this algorithm—with a 128-bit key for a wide variety of applications. There are no patent, copyright, or other restrictions on its use.

The algorithm has a block and key size of 64 bits. It is not a Feistel network like DES, but an iterated block cipher: The same function is applied for some number of rounds. Each round uses two 64-bit sub keys, and the algorithm only uses operations on bytes [Schneier].

Conclusion

Encryption algorithm play a very important role in communication security where encryption time, memory usages, output byte and battery power is the major issue of concern. In this paper, we have given a detailed study of different secret key encryption algorithms like DES, AES, CAST, IDEA, RC, BLOWFISH, TWOFISH, Camellia and SAFER.

References

Basu, S.(2011), International Data Encryption Algorithm (Idea)-A Typical Illustration, *Journal of Global Research in Computer Science,* 2(7).

Marcella, A. J., Menendez D., (2008), Cyber Forensics: A field manual for collecting, examining and preserving evidence of computer crimes*,*2nd ed. Auerbach Publications.

Mousa, A., Hamad A.(2006), Evaluation of the RC4 Algorithm for Data Encryption. *International Journal of Computer Science and Applications*, 3(2).

Kaliski, B.(1993), A Survey of Encryption Standards. *IEEE*.

Kumar, N., Thakur, J. Kalia A.(2011), Performance Analysis of Symmetric Key Cryptography Algorithms:DES,AES and Blowfish, *An International Journal of Engineering Sciences* ISSN: 2229-6913, 4.

Rivest R. L. (1997), The RC5 Encryption Algorithm.

Schneier B., Applied cryptography.

Surya, E., Diviya C., A Survey on Symmetric Key Encryption Algorithms. *International Journal of Computer Science & Communication Networks*, 2(4), 475-477.