International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 5, May - 2013

Symmetric Key Cryptography Algorithm Using Complement For Small Data Security

N. Bhaskar
Asst.professor, CMR Technical Campus

Abstract

During data transmission between the source and the destination in computer network, the data is exposed to external modifications with malicious intentions. In today's world, most of the means of secure data and code storage and distribution rely on using cryptographic schemes such as certificates or encryption keys. Cryptography is widely used to protect sensitive data from unauthorized access and modifications while on transit. There are two basic types of cryptography: i. Symmetric key and ii. Asymmetric key algorithms. Symmetric algorithms are the quickest and most commonly used type of encryption. Here, a single key is used for both encryption and decryption. There are few well-known symmetric key algorithms i.e. DES, IDEA, AES, RC2, RC4 etc. In this paper, a new symmetric key algorithm is proposed. The advantages of this new algorithm are also explained.

Keywords: Information Security, Encryption, Decryption, Symmetric key, Cryptography, Confidentiality, Integrity.

1. Introduction

During this time when the Internet provides essential communication between tens of millions of people and is being increasingly used as a tool for commerce, security becomes a tremendously important issue to deal with. There are many aspects to security and many applications, ranging from secure commerce and payments to private communications and protecting passwords.

One essential aspect for secure communications is that of Cryptography. The concept of securing messages through cryptography has a long history. Indeed, Julius Caesar is credited with creating one of the earliest cryptographic systems to send military messages to his generals.

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis.

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key—a word, number, or phrase—to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key. A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a cryptosystem.

2. Brief History of Cryptography

Cryptography, the science of encrypting and decrypting information, dates as far back as 1900 BC when a scribe in Egypt first used a derivation of the standard hieroglyphics of the day to communicate.[2] There are many notable personalities who participated in the evolution of Cryptography. For example, "Julius Caesar (100-44 BC) used a simple substitution with the normal alphabet (just shifting the letters by 3 positions) in government communications", [2] and later, Sir Francis Bacon in 1623, who described a cipher is known today as a 5-bit binary encoding. He advanced it as a steganographic device by using variation in type

face to carry each bit of the encoding". For all the historical personalities involved in the evolution of Cryptography, it is William Frederick Friedman, founder of Riverbank Laboratories, cryptanalyst for the US government, and lead code-breaker of Japan's World War II Purple Machine, who is "honored as the father of US cryptanalysis". In 1918 Friedman authored The Index of Coincidence and Its Applications in Cryptography, which is still considered by many in this field as the premiere work on cryptography written this century.

During the late 1920s and into the early 1930s, the US Federal Bureau of Investigation (FBI) established an office designed to deal with the increasing use of cryptography by criminals. At that time the criminal threat involved the importation of liquor. According to a report written in the mid-1930s by Mrs. Elizabeth Friedman, a cryptanalyst employed by the US government like her husband, William F. Friedman, the cryptography employed by bootleggers. Although cryptography was employed during World War I, two of the more notable machines were employed during World War II: the Germans' Enigma machine, developed by Arthur Scherbius, and the Japanese Purple Machine, developed using techniques first discovered by Herbert O. Yardley.

In the 1970s, Dr. Horst Feistel established the precursor to today's Data Encryption Standard (DES) with his 'family' of ciphers, the 'Feistel ciphers', while working at IBM's Watson Research Laboratory. In 1976, The National Security Agency (NSA) worked with the Feistel ciphers to establish FIPS PUB-46, known today as DES. Today, triple-DES is the security standard used by U.S. financial institututions. Also in 1976, two contemporaries of Feistel, Whitfield Diffie and Martin Hellman first introduced the idea of public key cryptography in a publication entitled "New Directions in Cryptography". Public key cryptography is what PGP, today's industry standard, uses in its software. In the September, 1977 issue of The Scientific American, Ronald L. Rivest, Adi Shamir and Leonard M. Adleman introduced to the world their RSA cipher, applicable to public key cryptography and digital signatures. The authors offered to send their full report to anyone who sent them self-addressed stamped envelopes, and the ensuing international response was so overwhelming the NSA balked at the idea of such widespread distribution of cryptography source code.

In the mid-1980s ROT13 was employed by USENET groups to prevent the viewing of "objectionable material [by] innocent eyes", and soon thereafter, a 1990 discovery by Xuejia Lai and James Massey proposed a new, stronger, 128-bit key cipher designed to replace the aging DES standard named International Data Encryption Algorithm (IDEA). This algorithm was designed to work more efficiently with "general purpose" computers used by everyday households and businesses. Concerned by the proliferation of cryptography, the FBI renewed its effort to gain access to plaintext messages of US citizens. In response, Phil Zimmerman released his first version of Pretty Good Privacy (PGP) in 1991 as a freeware product, which uses the IDEA algorithm. PGP, a free program providing military-grade algorithm to the internet community, has evolved into a cryptographic standard because of such widespread use. The initial versions of PGP were geared towards the more computer literate individual, but to the individual nonetheless. Phil Zimmerman could be compared to Henry Ford in his efforts to provide PGP to every home by making it free, and therefore, affordable. Today, PGP's updated version is offered free to the public. In 1994, Professor Ron Rivest, co-developer of RSA cryptography, published a new algorithm, RC5, on the Internet. It had been claimed that RC5 is stronger than DES. [2]

3. Cryptography

Data that can be read and understood without any special measures is called plaintext or clear-text. The method of disguising plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable gibberish called ciphertext. The process of reverting cipher-text to its original plaintext is called decryption.

Within the context of any application-to-application communication, there are some specific security requirements, including:

- *Authentication:* The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.).
- *Privacy/confidentiality:* Ensuring that no one can read the message except the intended receiver.
- *Integrity:* Assuring the receiver that the received message has not been altered in any way from the original.
- *Non-repudiation:* A mechanism to prove that the sender really sent this message.

In a typical situation where cryptography is used, two parties (X and Y) communicate over an insecure channel. X and Y want to ensure that their communication remains incomprehensible by anyone who might be listening. Furthermore, because X and Y are in remote locations, X must be sure that the information she receives from Y has not been modified by anyone during

transmission. In addition, she must be sure that the information really does originate from Y and not someone impersonating Y.

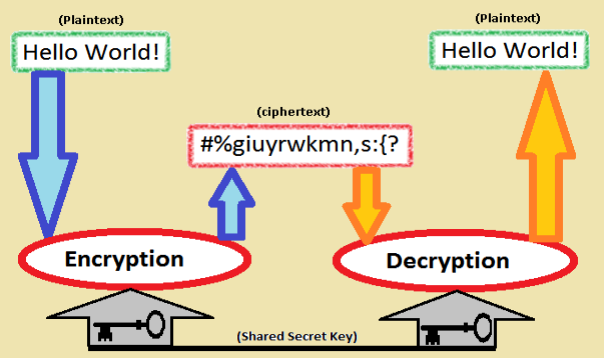

Fig 1: Cryptography concept

Cryptography is used to achieve the following goals:

3.1. Confidentiality

To ensure data remains private. Confidentiality is usually achieved using encryption. Encryption algorithms (that use encryption keys) are used to convert plain text into cipher text and the equivalent decryption algorithm is used to convert the cipher text back to plain text. Symmetric encryption algorithms use the same key for encryption and decryption, while asymmetric algorithms use a public/private key pair.

3.2. Data integrity

To ensure data is protected from accidental or deliberate (malicious) modification. Integrity is usually provided by message authentication code or hashes. A hash value is a fixed length numeric value derived from a sequence of data. Hash values are used to verify the integrity of data sent through insecure channels. The hash value of received data is compared to the hash value of the data as it was sent to determine if the data was altered.

3.3. Authentication

To assure that data originates from a particular party. Digital certificates are used to provide authentication. Digital signatures are usually applied to hash values as these are significantly smaller than the source data that they represent.

3.4. Non reputation

This gives assurances to the receiver of a message that it actually came from the sender and no one is faking the identity of the sender. This function of cryptography is provided with Public Key System only.

4. Types of Cryptography

Cryptography is a process which is associated with scrambling plaintext (ordinary text, or cleartext) into ciphertext (a process called encryption), then back again (known as decryption). There are several ways to classify the various algorithms. The most common types are i) Secret Key Cryptography which is also known as Symmetric Key Cryptography and ii) Public Key Cryptography which is also known as Asymmetric Key Cryptography.

In other words, if the same key is used for encryption and decryption, we call the mechanism as Symmetric Key Cryptography. However, if two different keys are used in a cryptographic mechanism, wherein one key is used for encryption, and another, different key is used for decryption; we call the mechanism as Asymmetric Key Cryptography. This is shown in Figure 2 [2].

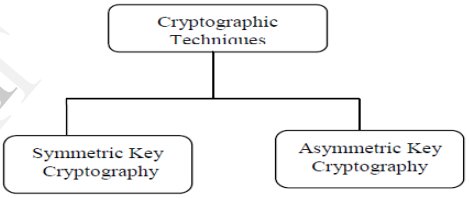

Fig 2: Cryptography techniques

4.1. Secret key cryptography

In secret key cryptography, a single key is used for both encryption and decryption. As shown in Figure 3, the sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key [5].

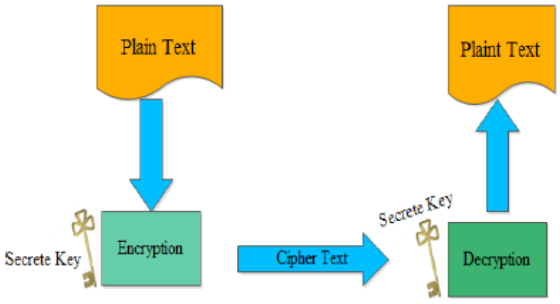

Fig 3: Secrete key algorithm

Secret key cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit (byte or computer word) 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. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher [1].Stream ciphers come in several flavors but two are worth mentioning here. Self-synchronizing stream ciphers calculate each bit in the keystream as a function of the previous n bits in the keystream. It is termed "self-synchronizing" because the decryption process can stay synchronized with the encryption process merely by knowing how far into the n-bit keystream it is. Synchronous stream ciphers generate the keystream in a fashion independent of the message stream but by using the same keystream generation function at sender and receiver. While stream ciphers do not propagate transmission errors, they are, by their nature, periodic so that the keystream will eventually repeat.

Block ciphers can operate in one of several modes; the following four are the most important:

- *Electronic Codebook (ECB) mode* is the simplest, 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 susceptible to a variety of brute-force attacks.
- *Cipher Block Chaining (CBC) 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.
- *Cipher Feedback (CFB) 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 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.
- *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 bitstreams.

The most common secret-key cryptography scheme used today is the Data Encryption Standard (DES), designed by IBM in the 1970s and adopted by the National Bureau of Standards (NBS) [now the National Institute for Standards and Technology (NIST)] in 1977 for commercial and unclassified government applications. DES has been adopted as Federal Information Processing Standard 46 (FIPS 46- 3) and by the American National Standards Institute as X3.92). DES is a blockcipher employing a 56-bit key that operates on 64-bit blocks [2].

There are a number of other secret-key cryptography algorithms that are also in use today like CAST-128 (block cipher), RC2 (block cipher) RC4 (stream cipher), RC5 (block cipher), Blowfish (block cipher), Two fish (block cipher). In 1997, NIST initiated a process to develop a new secure cryptosystem for U.S. government applications. The result, the Advanced Encryption Standard (AES), became the official successor to DES in December 2001.

4.2. Public key cryptography

Public or asymmetric key cryptography involves the use of key pairs: one private key and one public key. Both are required to encrypt and decrypt a message or transmission. The private key, not to be confused with the key utilized in private key cryptography, is just that, private. It is not to be shared with anyone. The owner of the key is responsible for securing it in such a manner that it will not be lost or compromised. On the other hand, the public key is just that, public. Public key cryptography intends for public keys to be accessible to all users. In fact, this is what makes the system strong. If a person can access anyone public key easily, usually via some form of directory service, then the two parties can communicate securely and with little effort, i.e. without a prior key distribution arrangement. Figure 4 describes the Public Key Cryptography [3].

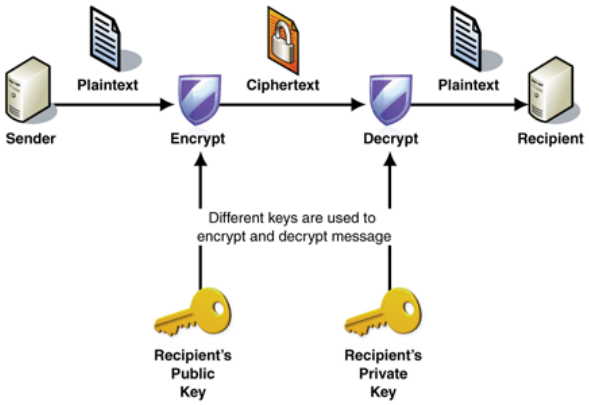

Fig 4: Public key algorithm

Public key cryptography depends upon the existence of so-called *one-way functions*, or mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute. Let me give you two simple examples:

- *Multiplication vs. factorization:* Suppose I tell you that I have two prime numbers, 3 and 7, and that I want to calculate the product; it should take almost no time to calculate that value, which is 21. Now suppose, instead, that I tell you that I have a number, 21, and I need you tell me which pair of prime numbers I multiplied together to obtain that number. You will eventually come up with the solution but whereas calculating the product took milliseconds, factoring will take longer. The problem becomes much harder if I start with primes that have 400 digits or so, because the product will have ~800 digits.
- *Exponentiation vs. logarithms:* Suppose I tell you that I want to take the number 3 to the 6th power; again, it is relatively easy to calculate $3^6 = 729$. But if I tell you that I have the number 729 and want you to tell me the two integers that I used, *x* and *y* so that $\log_x 729 = y$, it will take you longer to find the two values.

Public-key cryptography algorithms that are in use today for key exchange or digital signatures include:

- *RSA:* The first, and still most common, public key cryptography implementation, named for the three MIT mathematicians who developed it — Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key.
- *Diffie-Hellman*: After the RSA algorithm was published, Diffie and Hellman came up with their own algorithm. D-H is used for secret-key key exchange only, and not for authentication or digital signatures.
- *Digital Signature Algorithm (DSA):* The algorithm specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for the authentication of messages.
- *ElGamal:* Designed by Taher Elgamal, a PKC system similar to Diffie-Hellman and used for key exchange.
- *Elliptic Curve Cryptography (ECC):* A PKC algorithm based upon elliptic curves. ECC can offer levels of security with small keys comparable to RSA and other PKC methods. It was designed for devices with limited compute power and/or memory, such as smartcards and PDAs.
- *Public-Key Cryptography Standards (PKCS):* A set of interoperable standards and guidelines for public-key cryptography, designed by RSA Data Security Inc.
- *Cramer-Shoup*: A public-key cryptosystem proposed by R. Cramer and V. Shoup of IBM in 1998.
- *Key Exchange Algorithm (KEA):* A variation on Diffie-Hellman; proposed as the key exchange method for Capstone.
- *LUC:* A public-key cryptosystem designed by P.J. Smith and based on Lucas sequences. Can be used for encryption and signatures, using integer factoring.

5. Proposed Algorithm

In this section, we propose a new symmetric key algorithm that works with binary multiplication and division properties. It is restricted to 8-bit registers to store and maintain the binary data.

5.1. Encryption Algorithm

The entire process of encryption algorithm consists of the following steps. It implements stream cipher technique.

Step I: Accept the plain text letter.

Step II: Generate the ASCII value of the letter.

Step III: Convert ASCII value into binary format [which is 8-bit long, because the number of permitted ASCII letters is 256 only, i.e., $2^8 = 256$, for eg., the ASCII value 32 can be represented as 00100000 (underlined zeros are required)].

Step IV: Do 2's complement of that binary number.

Step V: Reverse the binary number.

Step VI: Take 10 as a secret key for encryption. Multiply that key with 2's complemented reversed binary number.

Step VII: The result comes from multiplication with secret key, the result will be converted into hexadecimal format. It gives cipher text.

5.2. Decryption Algorithm

Step I: The cipher text comes as a hexadecimal format, the result will be converted into binary format.

Step II: Take 10 as a secrete key for decryption and the result will be divided by the secrete key.

Step III: Reverse the binary number after divided by secret key.

Step IV: Do 2's complement of the binary number.

Step V: Convert the 2,s complement result into decimal format.

Step VI: Take the ASCII value of that decimal format and convert the given ASCII value into alphabet which is original plain text.

5.3. Practical Example

5.3.1. Encryption

Step I: Let the accepted letter be "T".

Step II: ASCII value of the letter "T" is 84.

Step III: The binary value of 84 is 1010100.

Step IV: Do 2's complement of that binary number.

0	1	0	1	1	0	0

Step V: Reverse the binary number.

0	0	1	1	0	1	0

Step VI: Take 10 as a secret key for encryption. Multiply that key with 2's complemented reversed binary number.

0	0	1	1	0	1	0	0

Step VII: Whatever the result comes from multiplication with secret key, the result will be converted into hexadecimal format. The result is 34 which is ciphertext.

5.3.1. Decryption

Step I: The cipher text comes as a hexadecimal format that is 34; this result will be converted into binary format.

0	0	1	1	0	1	0	0

Step II: Take 10 as a secrete key for decryption and the result will be divided by the secrete key.

0	0	1	1	0	1	0

Step III: Reverse the binary number after divided by secret key.

0	1	0	1	1	0	0

Step IV: Do 2's complement of the reversed binary number.

1	0	1	0	1	0	0

Step V: Convert the 2,s complement result into decimal format.

That is 84.

Step VI: Take the ASCII value of that decimal format and convert the given ASCII value into alphabet which is original plain text.

Alphabet is T.

6. Advantages of Proposed Algorithm

1. This Algorithm implementation and using is very simple.
2. There are two reverse and 2's complement operations present in this algorithm which would make it more secured.
3. CRC checking in receiving ends is easier.
4. This algorithm is most suitable for small amount of data.

7. Conclusion

Cryptography is used to achieve few goals like Confidentiality, Data integrity, Authentication etc. of the data which has sent to the receiver from the sender. Now, in order to achieve these goals various cryptographic algorithms are developed by various people. It has been found that the algorithms which are available at this moment are more or less difficult or complex in nature, and of-course it is quite obvious. Because those algorithms are used to maintain high level of security against any kind of forgeries. For a very minimal amount of data those algorithms wouldn't be cost effective since those are not designed for small amount of data. It has been found that the present algorithms

are more or less difficult or complex in nature and not cost-effective. The study aimed at design and implementing a new algorithm to address this issue. Keeping this goal in mind, the proposed algorithm has been designed in a quite simple manner. A single key is used for both encryption and decryption, i.e., it has fallen under secret key cryptographic algorithm. The proposed algorithm is very simple in nature. The entire process depends on binary division, multiplication, 2's compliment and reverses which give better security. Further, the CRC checking at receiving end is easier. Finally, we claim that for a small amount data, the algorithm works very effectively in very less time.

8. Acknowledgement

The successful completion of any task would be incomplete without expression of simple gratitude to the people who encouraged our work. Though words are not enough to express the sense of gratitude towards everyone who directly or indirectly helped in this task. I thankful to this organization CMR Technical Campus, which provided good facilities to accomplish my work and would like to sincerely thank to our Management, Director Dr. A. Raji Reddy, Dean Dr. A. Ravi Purna Chandra Rao, HOD K. Srujan Raju, my colleagues and parents for giving great support, valuable suggestions and guidance in every aspect of my work.

9. References

1) S. William, Cryptography and Network Security: Principles and Practice, 2nd edition, Prentice-Hall, Inc., 1999 pp 23-50.
2) S. Hebert, "A Brief History of Cryptography", an article available at http://cybercrimes.net/aindex.html
3) Fundamentals of Computer Security, Springer publications "Basic Cryptographic Algorithms", an article available at www.itsc.state.md.us/oldsite/info/InternetSecurity/Crypto/CryptoIntro.htm#Algorithms
4) Andrew S Tanenbaum (2003), Fundamental of Computer Networks, 4th Edition, Pearson Education Asia Ltd., ISBN:0-13-046002-8.
5) "Introduction to Public-Key Cryptography", an article available at developer.netscape.com/docs/manuals/security/pkin/content.htm
6) K. Gary, *"An Overview of Cryptography"*, an article available at www.garykessler.net/library/crypto.html
7) Computer and Network security by ATUL KAHATE
8) S. Goldwasser and S. Micali, "Probabilistic Encryption", *Journal of Computer and System Sciences*, Vol 28, 1994 pp 270-299.

10. Author

N. Bhaskar

Asst.Professor
CSE Department
CMR Technical Campus
Kandlakoya, Medchal Road
Hyderabad, India.