"CRYPTOLOGY"@Network Security and Computer Algorithm

Garima Ojha (Assistant Professor)

Harshit Pratap Shahi

Abstract

Cryptography prior to the modern age was effectively synonymous with encryption. Encryption is converting the information to be sent from a readable state to apparent nonsense. The originator of an encrypted message have to share the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons to do the same. In technical term it is about constructing i.e. making or generating and Analyzing protocols that overcome the influence of adversaries i.e. the third parties and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. In fact, it provided a way of secure talks in even the era of Hitler and Julius Caesar.

Keywords- cryptography; encryption; protocols;

Introduction:

Cryptography is concerned with the conceptualization, definition, and construction of computing systems that address security matters. Foundations of Cryptography present a complex and systematic treatment of foundational issues: like defining cryptographic tasks and solving new cryptographic problems using existing tools. Basically the encryption schemes used today (modern schemes) and the attacks they suffer are the main concern of this paper and consequently what are the attacks are also discussed. Along with it, it is to be noted that some of the encryption schemes (classical) like rotor machine, Caesar technique, steganography(basically it's not a encryption) which is important as everything is evolved from that basic work of which kings, queens and generals have used for efficient communication in order to govern their countries and command their armies. At the same time, they have all been aware of the consequences of their messages falling into the false hands, revealing precious secrets to rivals and vital information to opposing forces. The matter discussed inside assumes basic familiarity with the analysis of algorithms, some knowledge of complexity theory and probability will be a advantage for fast understanding.

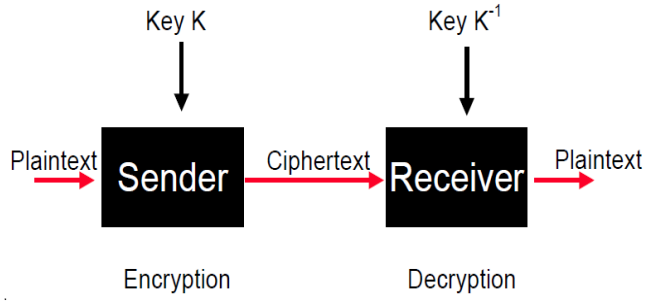

Fig 1. Process of encrypting and decrypting message

Types of Key scheme model:

Symmetric Key Model (Modern schemes):

Symmetric-key algorithms are a type of algorithms for cryptography that uses the similar cryptographic keys for both encryption of plaintext at the input side and decryption of cipher text at the output side. The keys are identical or there may be a simple transformation to go between the two keys. The keys represent a shared secret between two or more parties (anyone who have that same key) that can be used to maintain a private information link. While it provides a good scheme but this requirement that both parties have access to the secret key is one of the main drawbacks in itself of symmetric key encryption, in comparison to public-key encryption.

There are five ingredients of symmetric encryption scheme:

- Plaintext: This is the original message or data that is fed into the algorithm as input.
- Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.
- Secret key: The secret (latent) key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time and every time. The exact substitutions and transformations performed by the algorithm depend on the key.
- Ciphertext: This is nothing but the scrambled message produced as output. It depends on the plaintext and the secret key both. For a given message, two different keys will produce two entirely different cipher texts. The ciphertext is an apparently random stream of data which make no sense and, can be said unintelligible.
- Decryption algorithm: This is essentially the encryption algorithm which only runs in reverse order. It takes the cipher text and the secret key and produces the original plaintext as a output.

Types of symmetric key algorithms

Block Cipher:

A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically, a block size of 64 or 128 bits is used. The reason for taking the larger bits is to increase the complexity of the key and it cannot be guessed by any either method i.e. cryptanalyst and brute force attack. Using some of the modes of operation, a block cipher can be used to achieve the same effect as a stream cipher. Block ciphers are important elementary components in the design of many cryptographic protocols, and are widely used to implement encryption of bulk data.

The modern design of block ciphers is based totally on the concept of an iterated product cipher. Claude Shannon in his seminal 1949 publication Communication Theory of Secrecy Systems suggested Product ciphers. Iterated product ciphers carry out encryption in multiple rounds, i.e. each which it uses a different sub key which is derived from the original key. A widespread implementation of such ciphers is called a Feistel network, named after Horst Feistel, and notably implemented in the DES (data encryption standard) cipher. Many other realizations of block ciphers, such as the AES (advanced encryption standard), etc.

The publication of the DES cipher by the U.S. National Bureau of Standards (now National Institute of Standards and Technology, NIST) in 1977 was helpful in understanding of modern block cipher design. It influenced the academic development of cryptanalytic attacks. Both linear and differential cryptanalysis arose out of studies on the DES design. Today, there are plenty of attack techniques that a block cipher must be secure against, in addition to being robust against brute force attacks specially.

It is a matter of fact that, even a secure block cipher is suitable only for the encryption of a single block under a fixed key. A multitude of operations mode have been designed to allow their repeated use in a secure way, commonly to achieve the security goals of encryption along with authentication. However, block ciphers may also be used as building blocks in other cryptographic protocols, e.g. universal hash functions and pseudo-random number generators.

Suggested Method:

In this technique I am using a random number for generating the initial key, where this key will use for encrypting the given source file using proposed encryption algorithm with the help of encryption number. It's represented by a new block based symmetric cryptography algorithm. Basically In this technique a block based substitution method will be use. In the present technique I will provide for encrypting message multiple (like block) times. The proposed key blocks contains all possible words comprising of number (n) of characters and are generated from all characters whose ASCII code is from 0 to 255 in a random order. The pattern of the key blocks will depend on text key entered by the user. This system using 512 bit key size to encrypt a text message. It wills us very difficult to find out two same massages using this parameter. Initially, to decrypt any kind of file one has to know exactly what the key blocks are and to find the random blocks theoretically. One has to apply 2_{256} trial run and which is intractable. Initially that technique is only possible for some files such as Microsoft word file, excel file, text file.

Encryption used:

Here we are using symmetric encryption approach. We already know that symmetric encryption approach is divide in two type one is block cipher symmetric cryptography technique and another is stream cipher symmetric cryptography but here we

are choosing block cipher type because its efficiency and security. Basically private key concept is the symmetric key concepts where plain text is converting into encrypted text known as cipher text using private key where cipher text decrypted by same private key into plane text. The encryption key is related to the decryption key, in that they may be identical or there is a simple transformation to occur between the two keys.

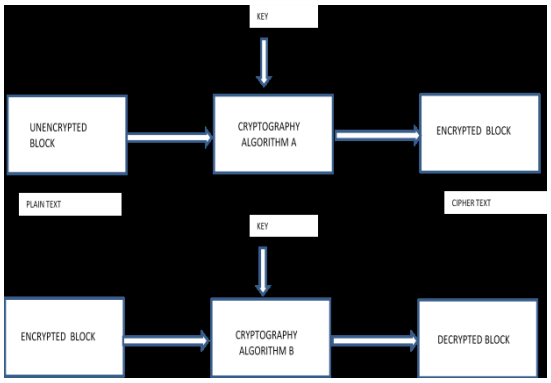

Figure 2: Basic Concept for Symmetric Cryptography

Reasons for Using Symmetric Approach for Encryption and Decryption:

- Keys for symmetric-key ciphers are relatively short.
- Each trading partner can use the same encryption algorithm no need to develop and exchange secret algorithms. This reduces the efforts for the new algorithm.
- Security is dependent only and only on the length of the key.
- High rates of data throughput (output created in given time period).
- Symmetric-key ciphers can be used as primitives to construct various cryptographic mechanisms.
- Symmetric-key ciphers can be composed to produce stronger ciphers like a complex algorithm created output.
- The encryption process is very simple.
- Symmetric-key encryption is perceived to have an extensive history.
- There are no complexities as such.

Steps for Key Generation:

1. Select or create (choice) any private key of Size 256 X 2 bits or 64 characters. It's must.
2. Size of selected key can vary from 128 bits to 512 bits or 16 to 64 characters.
3. Choose any character from 0 to 255 ASCII code.
4. Use 512 bits in length i.e. of 64 * 8 key.
5. Divide 64 bytes into 4 blocks of 16 bytes likes Key_Block1, Key_Block2, Key_Block3, and Key_Block4.
6. Apply XOR operation between Block1 and Block3. Store the results in new Key_Block13.
7. Again apply XOR operation between Block2 and Block13 and store the result in new Key_Block213.
8. Again apply XOR operation between Key_Block213 and Key_Block4. Results will store in new Key_Block4213.
9. Repeat Step 7, 8, 9 till (random number / 4).
10. Exit

Steps of proposed Algorithm:

1. First we have to select plane text of 16 bytes (or it can vary from 16 to 64 depend on requirement).
2. Initially insert key of size 16 bytes (entirely depends on plane text value)
3. Then apply XOR operation between key (Key_Block4213) and plane text block (Text_Block). Result will store in Cipher Block1.
4. Apply right circular shift (crcl) with 3 values. Store the result in new Cipher_Block2.
5. Again apply XOR operation between Cipher_Block2 and Key_Block2. Store the result in new Cipher_Block3.
6. Then again Apply XOR operation between Cipher_Block3 and Key_Block4. Result will store in Cipher_Block4.
7. Cipher_Block4 contains the input of the next round as a plane text block.
8. Repeat the step 1 to 7 till (Encryption Number / 4).
9. Exit.

Results Comparison:

We are taking two parameters for execution time one is encryption value and second is decryption time which is shown in table 1 and table 2. Here
Comparison of execution time of encrypting plaintext on different existing cryptographic is done. Here, The Proposed Algorithm (with 265 bit block size is used) and "A new Symmetric key Cryptography Algorithm by using the extended MSA method: DJSA symmetric key algorithm" algorithm (with 128-bit

block size) have been implemented on a number of different data files like text, pdf and image varying types of content and sizes of a wide range. Encryption and Decryption time of various Text files comparisons shown in table 1 and table 2 respectively.

Table 1: Time comparisons for encrypting text files

Plain Text Size	DJSA algorithm	Data Encryption through AES Methodology	Proposed Algorithm
1.66 mb	0:01:34	0:01:32	0:01:25
560 kb.txt	0:00:37	0:00:35	0:00:28
187 kb.txt	0:00:18	0:00:16	0:00:09
46 kb.txt	0:00:11	0:00:09	0:00:02
16 kb.txt	0:00:10	0:00:08	0:00:01

Table 2: Time comparisons for decrypting text files

Plain Text in Size	DJSA symmetric key algorithm	Data Encryption through AES Methodology	Proposed Algorithm
1.66 mb.txt	0:01:34	0:01:32	0:01:25
560 kb.txt	0:00:37	0:00:35	0:00:28
187 kb.txt	0:00:18	0:00:16	0:00:09
46 kb.txt	0:00:11	0:00:09	0:00:02
16 kb.txt	0:00:10	0:00:08	0:00:01

Advantages and Characteristic of Proposed Method:

1. Simplicity- As it is based on the symmetric key scheme it uses one single key at both the sides so makes the work simple.
2. Security- A key with 128 bit is almost not possible to break.
3. Efficiency: It can make the opposition feel embarrass.
4. Robustness: Quite appreciable
5. Time efficiency: From Table3 it's clear that it's quite hypothetical to break the key.

Table 3. Average Time Required for Exhaustive Key Search (Brute Force attack)

Key size (bits)	Number of alternative keys		Time required at 1 decryption/ms		Time required at 10^6 decryption/ms
32	2^{32}	= 4.3×10^9	2^{31} µs	= 35.8 minutes	2.15 milliseconds
56	2^{56}	= 7.2×10^{16}	2^{55} µs	= 1142 years	10.01 hours
128	2^{128}	= 3.4×10^{38}	2^{127} µs	= 5.4×10^{24} years	5.4×10^{18} years
168	2^{168}	= 3.7×10^{50}	2^{167} µs	= 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	26!	= 4×10^{26}	2×10^{26} µs	= 6.4×10^{12} years	6.4×10^6 years

Structure of Stream Cipher:

A typical cipher (stream) encrypts plaintext one byte at a time, although it may be designed to operate on one bit at a time or on units larger than a byte at a time. In this structure a key is input to a pseudorandom bit generator that produces a kind of stream of 8-bit numbers that are apparently random. The output of the generator, called a key stream, is combined one byte at a time with the plaintext using the bitwise exclusive-OR (XOR) operation. For example, if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting cipher text byte is. Decryption requires the use of the same pseudorandom sequence:

The difference is that a one-time pad uses a genuine random number as stream, whereas a stream cipher uses a pseudorandom number stream.

Following important design considerations for a stream cipher must be kept in mind while designing:

1. The encryption sequence should have a large period. A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats. Remember, the longer the period of repeat the more difficult it will be to do cryptanalysis.
2. The key stream should approximate the properties of a true random number stream as close as possible. For example, there should be an approximately equal number of 1s and 0s. If the key stream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often.
3. From the figure above output of the pseudorandom number generator is conditioned on the value of the input key. To guard against brute-force attacks, the key needs to be sufficiently long. The same considerations as apply for block into ciphers are valid here. Thus, with current technology, a key length of at least 128 bits is desirable for confusing opponent badly.

With a properly designed pseudorandom number generator, a stream cipher can be as secure as block cipher of comparable key length. The primary advantage of a stream cipher is that stream ciphers are almost always faster and use far less code than do block ciphers.

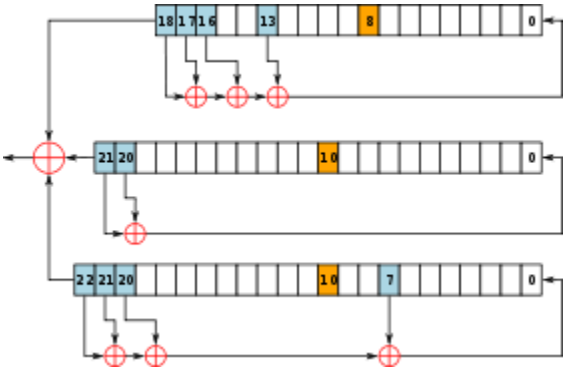

Figure 3: Stream ciphers diagram

The interface of the application is simple enough to be used by any user. The next figure shows the interface with the encryption and a decryption buttons. The encryption is performed simply by choosing any kind of file while decryption is executed by choosing an encrypted file with an appropriate.

Figure 4: Shows the Encryption/Decryption interface.

Figure 5: Shows a successful encryption and the encryption progress bar

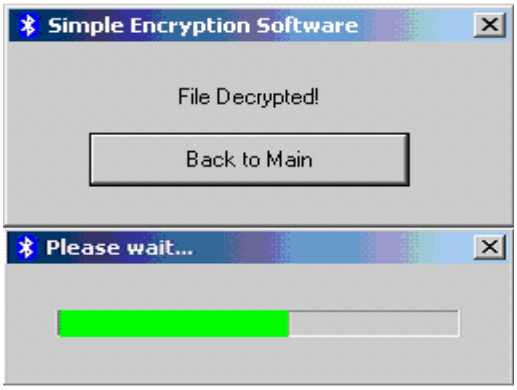

Figure 6: Successful decryption going and the decryption progress bar

Types of attack or security breaches:

Security attacks are classified in two types of terms one is the passive attack and other is the active attack. A passive attack attempts to learn or make use of the information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation other way round. So by the term it is clear that the encryption protects the message to be delivered from those who are not concerned to it but also if they get or decipher the message they can only use it to track the movements to be done according to the message and does not affect the resource or the system directly and once it is known that there is a seepage than the key which was used can be changed again to a more difficult one. So it can be counted under the passive attack.

Cryptanalysis:

Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single cipher text. So it shows here that it is a type of passive attack and does not affect the system directly in any way. There are two general approaches to attacking a conventional encryption scheme:

- Cryptanalysis: Its attacks solely rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-cipher text pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. But the opponent is required to have a insight of the word formation in the cipher text as it is necessary to guess the next optimum option to fill.

  Requirements that a cryptanalyst need to know to think the options available to him/her.

Table 4: Requirements for a cryptanalyst

Type of Attack	Known to Cryptanalyst
Ciphertext	• Encryption algorithm
only	• Ciphertext
Known plaintext	• Encryption algorithm • Ciphertext • One or more plaintext-ciphertext pairs formed with the secret key
Chosen plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	• Encryption algorithm • Ciphertext • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Note: In general we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the cipher text itself, generally applying various statistical tests to it.

The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of

information to work with. In many cases, however, the analyst has more information. The analyst may be able to capture one or more plaintext messages as well as their encryptions. Or the analyst may know that certain plaintext patterns will appear in a message. For example, a file that is encoded in the Postscript format always begins with the same pattern, or there may be a standardized header or banner to an electronic funds transfer message, and so on. All these are examples of known plaintext. This is the knowledge with which, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed. Let's say for example;

1. If an entire accounting file is being transmitted, the opponent may know the placement of certain key words in the header of the file. For example,
2. The source code for a program developed by any Corporation X might include a copyright statement in some standardized position which can have a specified format and the cryptanalysis. But if the analyst is able somehow to get the source system to insert into the system a message chosen by the analyst, then a chosen-plaintext attack is possible. It's an example of differential cryptanalysis

Brute force: A brute-force attack is a very common technique that involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table 3 can be referenced to understand the working of the brute force attack and it advisable to use large number of bits to generate a key that correspondingly shows how much time is involved for various key spaces. Results are shown for four binary key sizes. The 56-bit key size is used with the DES algorithm, and the 168-bit key size is used for triple DES (Data Encryption Standard). The minimum key size specified for AES (Advanced Encryption Standard) is 128 bits.

Results are also shown for substitution codes that use a 26-character key, in which all possible permutations of the 26 characters serve as keys. For each key size, the results are shown assuming that it takes 1 ms to perform a single decryption, which is a reasonable order of magnitude for today's machines. With the use of massively parallel processing organizations of microprocessors, it may be possible to achieve processing rates many orders of magnitude greater. The final column of (Table 3) considers the results for a system that can process 1 million keys per microsecond. As you can understand that, at this performance level, DES can no longer be considered computationally secure. Because other NSA (National Security Agencies) like one in U.S.A. (Digital Fortress by Dan Brown) is only one to do this herculean task. And moreover no one knows does that actually exist in this real world or are just cocktails of the novel.

Below shows the difference between old and latest contemporary GPU small and versatile.

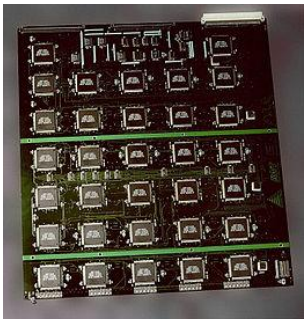

Figure 7: The DES cracking machine contained over 1,800 custom chips and could brute-force a DES key in a matter of days. The photograph shows a DES Cracker circuit board fitted on both sides with 64 Deep Crack chips.

Figure 8: Modern GPUs are well-suited to the repetitive tasks associated with hardware-based password cracking.

References:

[1] www.wikipedia.com
[2] www.google.com
[3] Cryptology and Network Security Principles and Practices by William Stallings.
[4] Digital fortress by Dan Brown (novel)
[5] Foundations of Cryptography: Basic Applications, by Oded Goldreich