A Secure Private Key Encryption Technique for Data Security in Modern Cryptosystem

Dilbag Singh[1] and **Ajit Singh**[2]

Abstract - ***The present paper provides a conceptual framework on the proposed private key encryption technique that can be used for data security in modern cryptosystem. This encryption technique uses the concept of arithmetic coding and can be used as an independent system as well as can be clubbed with any of the encryption system that works on floating point numbers. It provides you with a 256 character key (length of the key can be increased or decreased on the basic of the character set required) that can be used as a one time resident key or a key for every message depending on the level of security required. The proposed technique converts a word of text into a floating-point number that lie in between 0 and 1. This floating point no. is obtained on the basis of the probability of characters contained within the word of text and the one time key which has been provided .The security level can be further increased by generating different floating point number every time when a word repeat and increasing the length of the key.***

Index Terms - Arithmetic Coding, Encryption, Decryption, Floating point number, Resident and Regular key.

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. To provides the security cryptography come into the existence [1].Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription. Some experts argue that cryptography appeared spontaneously sometime after writing was invented, with applications ranging from diplomatic missives to war-time battle plans. It is no surprise, then, that new forms of cryptography came soon after the widespread development of computer communications. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet. Within the context of any application-to-application communication, there are some specific security requirements, including:

a) 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.)
b) Privacy/confidentiality*:* Ensuring that no one can read the message except the intended receiver.
c) Integrity: Assuring the receiver that the received message has not been altered in any way from the original.
d) Non-repudiation*:* A mechanism to prove that the sender really sent this message.

Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication. There are, in general, two types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, and public-key (or asymmetric) cryptography, each of which is described below. In all cases, the initial unencrypted data is referred to as plaintext. It is encrypted into ciphertext, which will in turn (usually) be decrypted into usable plaintext [2].

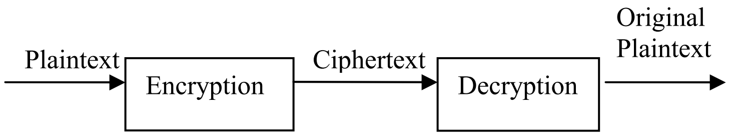

Figure 1: Basic Operation of Cryptography

1.1 Symmetric Encryption [2]

Symmetric Encryption (also known as symmetric-key encryption, single-key encryption, one-key encryption and private key encryption) is a type of encryption where the same secret key is used to encrypt and decrypt information or there is a simple transform between the two keys as shown in fig.2.
A secret key can be a number, a word, or just a string of random letters. Secret key is applied to the information to change the content in a particular way. This might be as simple as shifting each letter by a number of places in the alphabet. Symmetric algorithms require that both the sender and the receiver know the secret key, so they can encrypt and decrypt all information.
There are two types of symmetric algorithms: Stream algorithms (Stream ciphers) and Block algorithms (Block ciphers).

[1]*Department of Computer Science & Engineering, Choudhary Devi Lal University, Sirsa, Haryana (India)*
[2]*Department of Computer Science & Engineering, BPS Mahila Vishwavidyalaya, Khanpur Kalan, Sonepat, Haryana (India)*
E-mail: [1]*dbs_beniwal@rediffmail.com and* [2]*ghanghas_ajit@rediffmail.*com

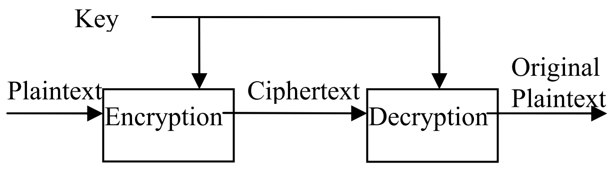

Figure 2: Symmetric Cryptosystem: KE= KD (KE Encryption Key and KD: Decryption Key

1.1.1 Types of Symmetric algorithms

Symmetric algorithms (Symmetric-key algorithms) use the same key for encryption and decryption. Symmetric-key algorithms can be divided into Stream algorithms (Stream ciphers) and Block algorithms (Block ciphers).

1.1.1.1StreamCiphers

Stream ciphers as shown in fig.3 encrypt the bits of information one at a time - operate on 1 bit (or sometimes 1 byte) of data at a time (encrypt data bit-by-bit). Stream ciphers are faster and smaller to implement than block ciphers, however, they have an important security gap. If the same key stream is used, certain types of attacks may cause the information to be revealed.

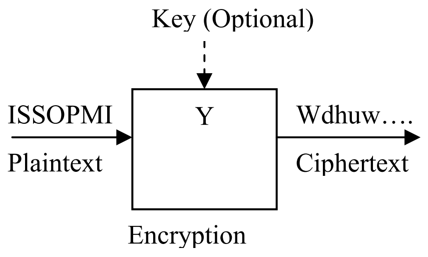

Figure 3: Stream Cipher-Convert one symbol of plaintext immediately into a symbol of ciphertext

1.1.12BlockCiphers

Block cipher (method for encrypting data in blocks as shown in fig.4) is a symmetric cipher which encrypts information by breaking it down into blocks and encrypting data in each block. A block cipher encrypts data in fixed sized blocks (commonly of 64 bits). The most used block ciphers are Triple DES and AES.

Some examples of symmetric encryption algorithms:

AES/Rijndael
Blowfish
CAST5
DES
IDEA
RC2
RC4
RC6
Serpent
Triple DES

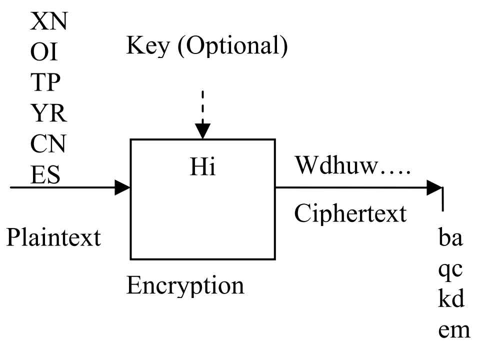

Figure 4: Block Cipher- Convert a group of plaintext symbols as one block

1.2 Asymmetric Encryption [8]

Asymmetric encryption (Also called Public Key Encryption) uses different keys for encryption and decryption. The decryption key is very hard to derive from the encryption key. The encryption key is public so that anyone can encrypt a message. However, the decryption key is private, so that only the receiver is able to decrypt the message. It is common to set up "key-pairs" within a network so that each user has a public and private key. The public key is made available to everyone so that they can send messages, but the private key is only made available to the person it belongs to.

1.2.1 Working of Asymmetric Encryption System[8]:

The sender and the recipient must have the same software. The recipient makes a pair of keys - public key and private key (both keys can be unlocked with a single password). Public key can be used by anyone with the same software to encrypt a message. Public keys can be freely distributed without worrying since it is only used to scramble (encrypt) the data. The sender does not need the recipient's password to use his or her public key to encrypt data. The recipient's other key is a private key that only he or she can use when decrypting the message. Private key should never be distributed since the private key assures that only the intended recipient can unscramble (decrypt) data intended for him or her.

To understand asymmetric encryption better consider an example, Jack makes public key A and private key A, and Jill makes public key B and private key B. Jack and Jill exchange their public keys. Once they have exchanged keys, Jack can send an encrypted message to Jill by using Jill's public key B to scramble the message. Jill uses her private key B to unscramble it. If Jill wants to send an encrypted message to Jack, she uses Jack's public key A to scramble her message, which Jack can then unscramble with his private key A. Asymmetric cryptography is typically slower to execute electronically than symmetric cryptography.

Some Asymmetric Algorithms (public key algorithms) such as RSA allow the process to work in the opposite direction as well: a message can be encrypted with a private key and decrypted with the corresponding public key. If the recipient wants to decrypt a message with Bob's public key he/she must know that the message has come from Bob because no one else has sender's private key.

Digital signatures work this way.

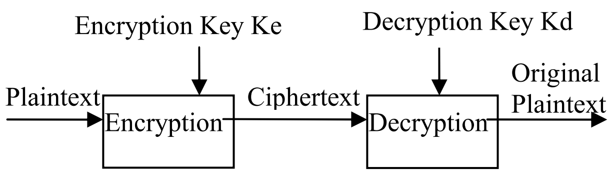

Figure 5: Asymmetric Cryptosystem: KE≠KD (KE Encryption Key and KD: Decryption Key)

Some examples of popular asymmetric encryption algorithms:
RSA
DSA
PGP

2. ARITHMETIC CODING

In arithmetic coding, a message is encoded as a real number in an interval from one to zero. The idea behind arithmetic coding is to have a probability line, 0-1, and assign to every symbol a range in this line based on its probability [6], the higher the probability, the higher range which assigns to it. Once we have defined the ranges and the probability line, start to encode symbols, every symbol defines where the output floating point number lands.

The coding algorithm is symbol wise recursive; i.e., it operates upon and encodes (decodes) one data symbol per iteration or recursion. On each recursion, the algorithm successively partitions an interval of the number line between 0 and 1, and retains one of the partitions as the new interval. Thus, the algorithm successively deals with smaller intervals, and the code string, viewed as a magnitude, lies in each of the nested intervals. The data string is recovered by using magnitude comparisons on the code string to recreate how the encoder must have successively partitioned and retained each nested subinterval. Arithmetic coding differs considerably from the more familiar compression coding techniques, such as prefix (Huffman) codes [4].

Arithmetic coding typically has a better compression ratio than Huffman coding, as it produces a single symbol rather than several separate codeword and can be use in compression based encryption system [12]. There are a few disadvantages of arithmetic coding. One is that the whole codeword must be received to start decoding the symbols, and if there is a corrupt bit in the codeword, the entire message could become corrupt. Another is that there is a limit to the precision of the number which can be encoded, thus limiting the number of symbols to encode within a codeword. [7]

3. SHANNON CHARACTERISTICS OF A GOOD ENCRYPTION SYSTEM [5]

1. The amount of security needed should determine the amount of labor appropriate for the encryption and decryption.
2. The set of keys and enciphering algorithm should be free from complexity.
3. The implementation of the process should be as simple as possible.
4. Errors in ciphering should not propagate and cause corruption of further information in the message.
5. The size of the enciphered text should be no larger than the text of the original message.

4. PERPOSED PRIVATE KEY ENCRYPTION TECHINQUE

The proposed technique is based on the concept of arithmetic coding [9] in which a word of text is converted into a floating-point number that lie in the range between 0 and 1. Private Key encryption system based on this technique can be used as an independent system as well as can be clubbed with any of the encryption system that works on floating point numbers [7]. The probability table can also be set according to the user requirement and the combination of two keys working one over the other makes it extremely difficult to break as the total no of exhaustive cases shoot up tremendously.

4.1 Requirements

A. Two Keys

The system is implemented using two keys: -

1. Resident key: - It's a one-time key provided by the user when the software is installed or initiated.
2. Regular key: -This key is subjected to change as and when the user thinks that the previous keys have been disclosed. Length varies with security requirements.

Note: Regular key used when the system work in independent mode.

B. A Table

A table containing all symbols along with probability of occurrence [3,9].

Symbol	Probability of Occurrence	Symbol	Probability of Occurrence
^	0.0001	N	0.0380
A	0.0500	O	0.0320
B	0.0500	P	0.0300
C	0.0450	Q	0.0380
D	0.0455	R	0.0400
E	0.0380	S	0.0400
F	0.0360	T	0.0320
G	0.0400	U	0.0300
H	0.0360	V	0.0350
I	0.0380	W	0.0300
J	0.0320	X	0.0300
K	0.0400	Y	0.0250
L	0.0360	Z	0.0779
M	0.0360		

4.2 Implementation

In the Add on to the Existing Encryption System mode of implementation the technique converts a word of text into a floating-point number. This floating point no. is obtained on the basis of the probability of characters contained within the word of text and the one time key which has been provided[10, 11].

Explanation

Encryption: - Each character in the 256-character key is associated with its corresponding probability of occurrence using the table. The sequence of these characters along with their probabilities acts as the basis for Algorithm [9] of the technique to work.

Algorithm

In order to implement the algorithm array data structure can be use for the resident key, priority table and message. To encrypt the message, algorithm includes the following steps:

```
Step 1:- On the basis of the resident key and priority table derive another table (Range table)
  a)Initialize range_from=0, range_to=probability [first element of the key], counter=1
  b) Repeat steps for all the characters in the one time key
     range_from [counter] =range_to[counter-1]
     range_to[count]=range_from[count]+ probability[count]
         Count=count + 1;
Step2:- Read the word to be encrypted.
Step3:-Initialize Low_value=0, High_value=1,
        difference=1, count=1
Step4:- Repeat for every character of the word
        Temp=Low_value[count-1]
        Low_value[count]=low_value[count–1]+ difference[count-1] * range_from[symbol]
        High value[count]=temp + difference[count –1]* range_to[symbol]
 Difference[count]=high_value[count]- low_value[count]
        Count=count+1
 Step5:- float_code=low_value
 Step6:- Repeat process for every word in the file
```

Output: Output is a floating point no. that is corresponding to the inputted word. It can be provided further to any other encryption algorithm that works on floating point numbers

Decryption: - Getting the floating points its time now that we convert it into original text.

Algorithm

To deecrypt the message, algorithm include the following steps:

```
Step 1:- while flote_code ! = 0.0 repeat step 3 to 5
Step 2:- initialize count=1
Step3:- find range[symbol] where float_code lies and set count accordingly.
Step4:-float_code
        =(float_code - range_from[count])/prob[count]
Step5:- store symbol in a character string
Srep6:- count=count+1
Step7:- repeat process for every word in the file
```

Output: Output is the original text.

Example:

Encryption:

```
The resident key used is
a b c d e f g h i j k l m n o p q r s t u v w x y z
Enter the word to be encrypted
anil
Symbol   Low_value   High_value    Difference
  a        0.0001      0.0501        0.05
  n        0.026205    0.028105      0.0019
  i        0.026851    0.026923      7.22e-05
  l        0.026884    0.026886      2.5992e-06
Equivalent floating point number is  0.026884
```

Decryption:

```
Symbol  Range_From  Range_To  Prob.   Flot_Code
 a       0.0001      0.0501    0.05    0.535674
 n       0.5221      0.5601    0.038   0.357204
 i       0.3401      0.3781    0.038   0.4501
 l       0.4501      0.4861    0.36    6.53798e-13
The word  is anil
```

5. KEY FEATURES

1. It's a Private Key Encryption system.
2. Make the system that can operates in different modes

2.1 Add on to the existing encryption system

2.2 Independent system

3. Flexibility: - The system is extremely flexible as allows the length of both the keys to be changed, the length of the resident key depends on the character set required. The probability table can also be set according to the user requirement.
4. Extremely difficult to break the code by brute force attack: - The combination of two keys working one over the other makes it extremely difficult to break as the total no of exhaustive cases shoot up tremendously.

6. EFFICIENCY

The proposed technique satisfies all the Shannon Characteristics of a Good Encryption System as shown in the table given below. It makes the system faster, portable and requires memory space of less than 15kb and also provides an efficient data security during communications.

Sr. No.	**Shannon Characteristics of a Good Encryption System**	**Proposed Technique**
1	The amount of security needed should determine the amount of labor appropriate for the encryption and decryption.	√
2	The set of keys and enciphering algorithm should be free from complexity.	√

Continued on page no. 270