A NEW APPROACH FOR GENERATING STRONG KEY IN RIVESTCIPHER4 ALGORITHM

[1]MANIKANDAN GANESAN, [2]MANIKANDAN RAMACHANDRAN, [3]SUNDARGANESH GOVINDKUMAR

[1]Assistant Professor., School of Computing, SASTRA University Tamil Nadu, India

[2]Assistant Professor, School of Computing, SASTRA University Tamil Nadu, India

[3]Student , School of Computing, SASTRA University Tamil Nadu, India

E-mail: manikandan@it.sastra.edu , manikandan@core.sastra.edu , sundarganesh2012@gmail.com

ABSTRACT

In the era of digital transaction, in order to transact the data in a more secured manner the need for a cryptographic algorithm is inevitable. There are numerous numbers of cryptographic algorithms which makes the system invulnerable from the attacks of intruders and eavesdroppers.RivestCipher4 algorithm is one such cryptographic algorithm which is very well known for its performance and simplicity. In this paper, we propose a software toolkit for increasing the key strength so that it will be very hard for the intruder to break the key and this technique will act as black box so that intruder will have no idea about the key formation. So it will lead to increased key complexity which obviously results the intruder nothing else than confusion and frustration.

Keywords: *Cryptography, Encryption, key strength, key complexity***.**

1. INTRODUCTION

Most of the existing systems are vulnerable to attacks and it can be broken at some point of time by crypt analyzing it. There are various cryptanalysis techniques available to break most of the encryption algorithms at one point of time. Each and every algorithm either it may be block cipher or stream cipher or any other cipher types can be easily attacked by performing various cryptanalysis techniques like brute force attack, n-gram analysis, linear and non linear cryptanalysis, meet in the middle attack, Man in the middle attack etc..we are evidencing the intruders intruding the systems which possesses even a complex algorithmic design. Mostly the algorithms of different ages are broke easily by eavesdroppers at one stage and we are witnessing it in our day-to-day daily life. This is because the algorithmic developers always believe in their self developed encryption formulas and we the users firmly attached of following a single algorithm which is no more secure after a period of time. It is quite obvious to digest the fact that it is easy to cryptanalysis any algorithm within months as soon as they are adapted to practical use. Since the intruders and eavesdroppers had shown their excelling skills towards breaking the encryption algorithms almost in all important and sensible areas like Banking, Military, Defense, Networks, a need for "practically strong and infeasible to get attacked" algorithm becomes vital. This paper suggests one such cryptographic technique which never ever gives a clue of neither the encryption pattern adopted nor the number of iterations that will carry out to obtain the high end cipher text.[1]

Cryptography is a well known and widely used technique that manipulate information in order to crypt their existence. To be more specific, cryptography protects information by transforming it into an unreadable format [1]. The original text is transformed into a scramble equivalent text called cipher text and this process is called as "Encryption". This is achieved via an Encryption Algorithm. Only those who possess a secret key can decrypt the cipher text into plaintext. Simply it scrambles a message so it cannot be understood.

Cryptography deals with protecting information by encoding or transformation of data [1].There are two types of cryptographic schemes available on the basis of key [1].

1. *Symmetric key Cryptography*: This is the cryptographic scheme which uses a common key for enciphering and deciphering the message.
2. *Asymmetric or Public Key Cryptography*: This type of cryptographic scheme uses two keys for encryption and decryption called Public key and Private Keys.

We adopted Symmetric key cryptographic scheme and hence only one key is needed for communication. So, the chosen cryptographic scheme involves,

1. *Plaintext*: The original message that has to be communicated to receiver.
2. *Encryption*: Enciphering of data by using a key via a desired encryption algorithm at sender side.
3. *Transmission*: Transfer of cipher message to receiver through a public communication channel.
4. *Decryption*: Deciphering of the cipher text thus received via the same algorithm (reverse Encryption) by using the key.

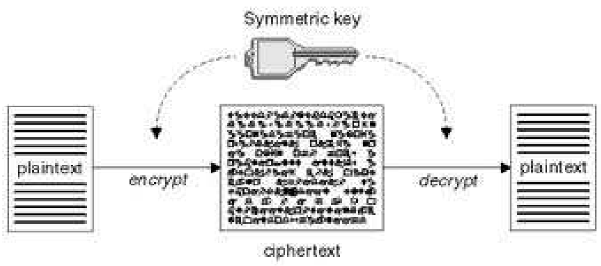


Fig 1: Symmetric Key Cryptography

We can also classify symmetric key cryptography into two types on the basis of their operations as

1. *Stream Ciphers*: It is a symmetric key cipher where stream of plaintext are mixed with a random cipher bit stream (key stream), typically by any logical operation.
2. *Block Ciphers*: Block cipher encryption algorithm takes an n-bit block of plaintext as input, and produces a corresponding n-bit output block of cipher text.[2]

We have chosen stream cipher for our cryptographic operation since it is the main tool for implementing private key encryption in practice. The original RC4 stream cipher is famous but it is easy to be attacked by hardware based key search algorithms. Nevertheless, many applications are still using RC4 stream cipher as security protection for small and portable devices. For example, Wired Equivalent Privacy (WEP) , Wi-Fi Protected Access and WPA2 are using RC4 stream cipher with improve features such as the addition of IV to form the RC4 traffic key, increased key and IV sizes, and temporal key integrity protocol. Currently, some works was also done for improving the RC4 stream cipher core algorithm itself. [7]

2. EXSISTING SYSTEM

RC4 is a synchronous stream cipher designed to satisfy both security and efficiency for lightweight algorithms, dedicated to hardware. Environments where the available resources are restricted. RC4 stream ciphers have weaknesses on the key size. If the key size is short, attacker can easily obtain the key by using the key recovery algorithms. In this case, re-keying is necessary.[10] Generally, re-keying is done by using the internal state in each packet to reinitialize the large internal state. The newly initialized internal state is dependant; therefore more variation and randomness can be achieved for the internal state. The value of internal state must be unique, and must not be used twice or more although the messages to be encrypted are different. Stream cipher is the important class of encryption and they encrypt each digit of plain text one at a time using a simple time dependent encryption transformation in practice ,the digit is single bit or byteRc4 is most widely used stream cipher nowadays due to its simplicity and high efficiency .rc4 is a variable key size key size stream cipher based on a 256 byte internal state and two one byte indexes I and j.rc4 consist of two parts namely key scheduling algorithm and pseudo random generation[2].
The key-scheduling algorithm consists of following steps in order to generate a key we should start the permutation in the array "S". "Key length" is stated as the number of bytes in

the key and it usually range 1 ≤ key length ≤ 256, typically between 5 and 16, accordance to a key length of 40 – 128 bits. Initially, the array "S" is initialized to the recognized permutation. Array "S" operated for 256 iterations in a similar way to the main PRGA, but also mixes in bytes of the key[3].

```
for (i=0;i<=255;i++)
    S[i] = I;

j = 0;
for  (i=0;i <=255;i++)
{
    j = (S[i] + i + key[i mod key length]) mod 256;
    swap(S[i] , S[j]);
}
```

The Pseudo random generation algorithm (PRGA) modifies the state and outputs a byte of the key stream. In each iteration, the PRGA increments *i*, adds the value of S pointed to by *i* to *j*, exchanges the values of S[*i*] and S[*j*], and then outputs the element of S at the location S[*i*] + S[*j*] (modulo 256). Each element of S is swapped with another element at least once every 256 iterations.[3]

```
i = 0;
j = 0;
while
{
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    swap( S[i] , S[j] );
    K = S[(S[i] + S[j]) mod 256];

}[1]
```

KGA:

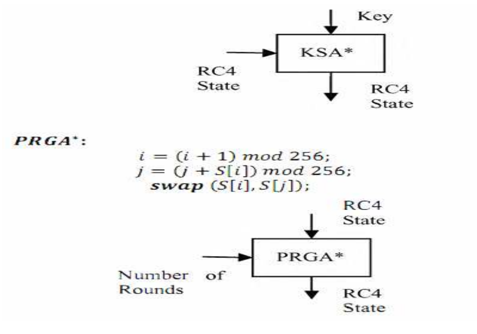


Fig 2: KSA and PRGA schedule

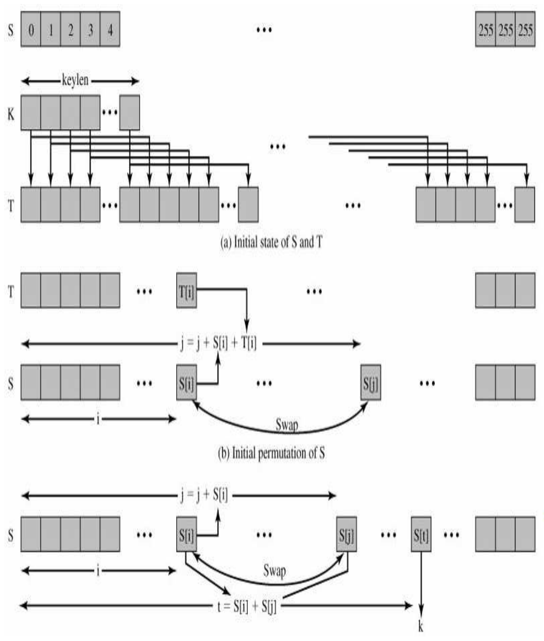


Fig 3. Steps in RC4 Algorithm

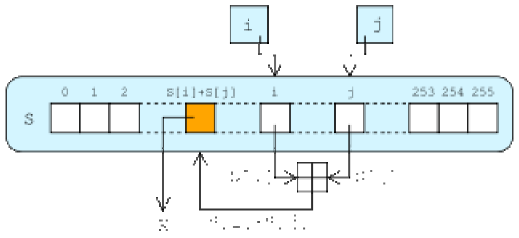


Fig 4. RC4 stream generation

3. PROPOSED SYSTEM

In this paper we propose a black box tool for generating a fresh key from the actual key provided by the user. The fresh key thus obtained is hard to crack because of our complex black box design. In general black box may be of anything which has a set of confused and diffused mathematical formulas. For our publication purpose we propose a set of steps which is to be carried out in a black box. The following block diagram illustrates our modified approach to the Rivestcipher4 algorithm, and the steps which have to take place inside a black box.

Let K be the original key, CK be the concatenated key, NK be the ASCII conversion of the original key K, X be the resultant new key resulting from the black box.

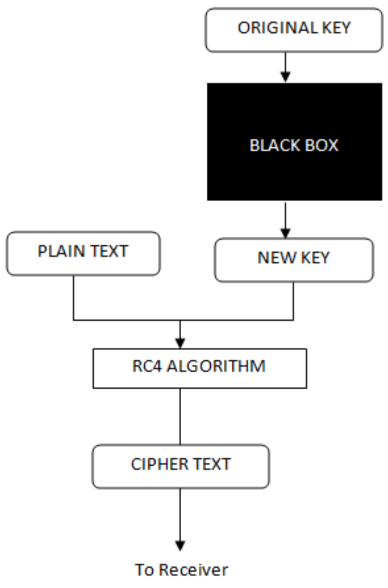


Fig 5:Block diagram of encryption phase

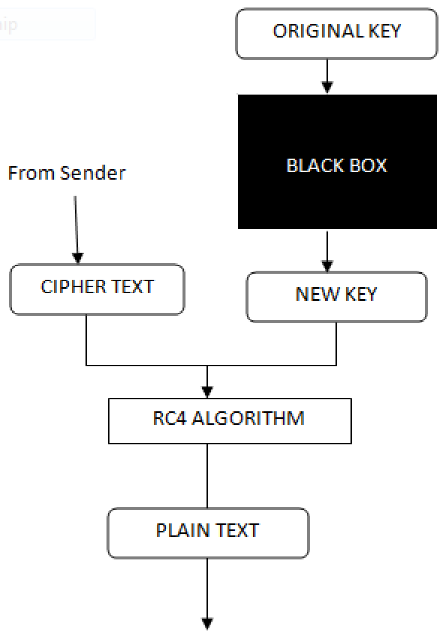


Fig 6: Block diagram of decryption phase

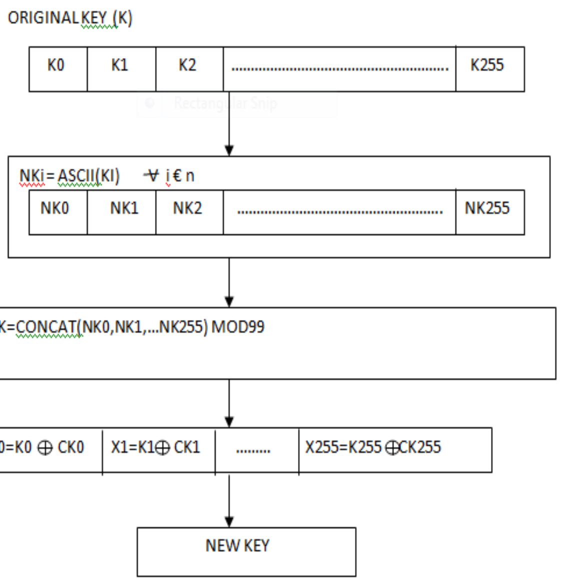


Fig7: Generation of new key in black box technique

In order to derive a new key(X) from the already existing key (k) of the RC4 the following steps are followed

1. From the key (K) which is given by the particular user, we get the ASCII values of the particular characters.
2. The ASCII values are concatenated which is called as the concatenated key (CK)
3. The Concatenated Key (CK) is now manipulated inside a particular modular function (mod 99) for the purpose of ending up with two digit number and the resulted value is Y.
4. The particular value Y is now XORed with the each ASCII value of the given character.
5. The resultant of the XORed product is the new fresh key which is fed into the RC4 algorithm.

4. CASE STUDY

In order to increase the key strength of RC4 we should generate a new key in order to increase the complexity of the key which is fed as the input to the actual RC4 algorithm. This can be explained with a simple example.

For instance if the user uses a key called “SASTRA”, in the actual RC4algorithm this key is fed into the key generation algorithm which is followed by the pseudorandom random generation algorithm so in this paper the black box technique is proposed in order to increase the key strength. This black box technique is explained as follows. If the user gives the particular key called “SASTRA”, in that case from given key (k) is the ASCII value of the each and every character of the key is derived.

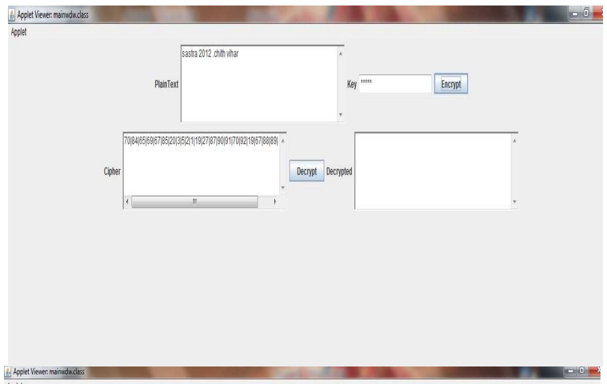
1. From the key (K) here SASTRA , we get the ASCII values for all characters present in the string SASTRA.
Here, S-83, A-65, S-83, T-84, R-82, A-65
2. The ASCII values are then concatenated which we are referring as the concatenated key (CK).
Here the CK is 836583848265
3. The concatenated key (CK) is now manipulated inside a particular modular function which will result in a particular value Y.
Here (836583848265) mod 99 will be67.
4. The particular value Y is now XOR with the each ASCII value of the given character.
(83,65,83,84,82,65) $\oplus$ (66,66,66,66,66,66) = (17,01,17,22,16,01)
5. The resultant of the XOR product i.e., 170117221601 is the new fresh key which is fed into the RC4 algorithm.

5. SIGNIFICANCE OF PROPOSED SYSTEM

1. The newly formed key which is generated will not be known even to sender and receiver.
2. It is worth a million to say in a line that more the complex black box higher will be its key strength.
3. In general, Key strength which is one of the salient feature of block cipher which is missing in all stream ciphers. This scenario will come to an end by the usage of black box technique which will increase the key strength that is ever provided by other stream ciphers
4. Black Box is nothing but a complex formula based logical transformation of key in to an another form which will be more complex than the original key and hence the logic flow of black box can be changed by any means by following any complex mathematical or analytical transformations rather restricting to one that we provided above.
5. Since this black box execution is independent of the original RC4 algorithm, there is no chance for affecting the data security of RC4 algorithm.
6. We propose this black box approach as a open software tool so that it will be suitable in the mere future to hold any algorithm like RC4 which lags in their key strength.

6. SIMULATION AND RESULTS

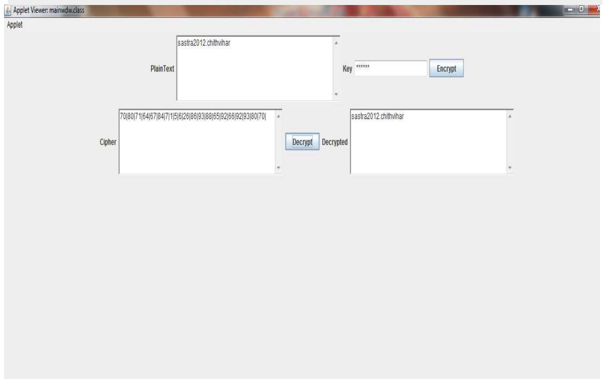
For the purpose of simulating the black box and RC4 algorithm we used Java which is known for its platform independency and better GUI features. The following figures illustrates about the encryption and decryption phase of Rc4 algorithms.



sastra 2012 .chith vihar

*****

70|84|65|69|67|85|20|3|5|2|1|19|27|87|90|91|70|92|19|67|88|89|



sastra2012.chithvihar

******

70|80|71|64|67|84|7|1|5|6|26|86|93|88|65|92|66|92|93|80|70|

sastra2012.chithvihar

Fig 8& 9: Rc4 Encyption& Decryption

7. FUTURE ENHANCEMENTS

The black box tool that is designed can be easily modified to accept any encryption algorithm which is framed in future. Just by adding or removing another module in the main function and also by increasing or decreasing the hash limit, any number of iteration can be included or reduced in the particular algorithm or other in particular they are used to increase the key length especially in the case of RC4[9]. Though the system is designed for stream cipher but the modules can be used in block cipher also in which key length is already in high complexity. By adding a new authentication between the sender and receiver sockets, the system can also be improved to work as secure WAP and WLAN. Moreover, we currently concentrate on our next work which adopts integration of block and stream cipher in order to produce a very strong key such that it will lead to the increase in the key complexity as well as the data security and hence very difficult to interoperate.

8. CONCLUSION

Key strength of RC4 has the main concern of security weakness , so that they can be easily breached, the tool that we developed is used to increase the complexity of the key .The intruder will have no idea about the existence of the particular black box tool and before his realization the particular data will be transferred to the receiver. Thus the black box tool that is being implemented will increase the key complexity without affecting the performance up to a maximum level that ever offered by the existing stream ciphers.

REFERENCES:

[1] B. Schneier, *Applied Cryptography*, 2nd edition ed. New York:Wiley, 1996.

[2] G. Gong, K. C. Gupta, M. Hell, and Y. Nawaz, "Towards a general RC4-like keystream generator," in *SKLOIS Conf. Inf. Security and Cryptol., CISC 2005 LNCS 3822*. New York: Springer-Verlag, 2005,pp. 162–174.

[3] S. Paul and B. Preneel, "On the (in)security of stream ciphers based on arrays and modular addition," in *Adv. Cryptol.—Asiacrypt 2006, LNCS4284*. New York: Springer-Verlag, 2006, pp. 69–83.

[4] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Sel. Areas in Cryptogr., SAC 2001, LNCS2259*. New York: Springer-Verlag, 2001, pp. 1–24.

[5] S. Fluhrer and D. McGrew, "Statistical analysis of the alleged RC4 keystream generator," in *Fast Software Encrypt., FSE 2000 LNCS1978*. New York: Springer-Verlag, 2000, pp. 19–30.

[6] S. Paul and B. Preneel, "A new weakness in the RC4 keystream generator,"in *Fast Software Encrypt., FSE 2004, LNCS 3017*. New York: Springer-Verlag, 2004, pp. 245–259.IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 53, NO. 9, SEPTEMBER 2007 3255

[7] I. Mantin, "Predicting and distinguishing attacks on RC4 keystream generator," in *Adv. Cryptol. — Eurocrypt 2005, LNCS 3494*. New York: Springer-Verlag, 2005, pp. 491–506.

[8] I. Mantin, "A practical attack on the fixed RC4 in the wep mode," in *Adv. Cryptol. — Asiacrypt 2005, LNCS 3788*. New York: Springer-Verlag, 2005, pp. 395–411.

[9] R. J. Jenkins, Jr., *ISAAC Fast Software Encrypt., FSE 1996, LNCS1039*. New York: Springer-Verlag, 1996, pp. 41–49.

[10] B. Zoltak, "VMPC one-way function and stream cipher," in *Fast SoftwareEncrypt., FSE 2004, LNCS 3017*. New York: Springer-Verlag,2004, pp. 210–225.

[11] A. Maximov, "Two linear distinguishing attacks on VMPC and RC4A and weakness of the RC4 family of stream ciphers," in *Fast Software Encrypt., FSE 2005, LNCS*

3557. New York: Springer-Verlag, 2005, pp. 342–358.

[12] Y. Tsunoo, T. Saito, H. Kubo, M. Shigeri, T. Suzaki, and T. Kawabata, "VMPC and RC4A attack," in *SKEW 2005 — Symmetric Key EncryptionWorkshop*, 2005 [Online].Available: http://www.ecrypt.eu.org/stream/papersdir/037.pdf

[13] E. Biham and J. Seberry, Py (Roo): A Fast and Secure Stream Cipher Using Rolling Arrays Report 2005/023, 2005 [Online]. Available:http://www.ecrypt.eu.org/stream/py.html, STREAM, the ECRYPT Stream Cipher Project

[14] S. Paul, B. Preneel, and G. Sekar, Distinguishing Attacks on the Stream Cipher Py Report 2005/081, 2005 [Online]. Available: http://www.ecrypt.eu.org/stream/py.html, STREAM, the ECRYPT Stream Cipher Project

AUTHOR PROFILES:

Mr. Manikandan Ganesan is working in SASTRA University as Assistant .Professor in school of computing SASTRA University. He has wide teaching experience of around 10 years. His areas of interest mainly include Data mining and warehousing, cryptography and network security.

Mr. Manikandan Ramachandran is working in SASTRA University in school of computing as Assistant .professor. He has teaching experience of 10 years .His area of interest mainly include VLSI, Embedded systems, and cryptography.

Mr.Sundarganesh Govindakumar is currently doing final year in Information and Communication Technology School of computing SASTRA University. His areas of interest mainly include ubiquitous computing, cryptography and network security, mobile ad hoc networks.