International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 7, July - 2014

# An Approach For Securing RC4 Algorithm

Priya Mishra
M.Tech Scholar
JECRC University

Vijay Prakash Sharma
Assistant Professor
JECRC University

**Abstract: In few past decades various internet traffic security algorithms gained quick popularity and RC4 algorithm for internet traffic can be cited as most prominent or proficient technology today. The Transport Layer Security (TLS) provides confidentiality and integrity of data when data transmits across unsecure network. TLS supports several encryption methods but in this paper TLS uses RC4 for encryption. RC4 is a stream cipher. It mainly consists of two algorithms. First one is KSA and another is PRGA. RC4 is extremely fast when implemented in software but at the cost of lower security. This paper present a work on KSA and PRGA that helps in improving the RC4 algorithm for secure the internet traffic. Main work focus on the key length and the variable multiplication in the both algorithms that generates the encrypted secure text. The proposed work provide the secure encrypted algorithm for achieving more security in RC4.**

***Keywords: TLS(Transport Layer Security), RC4(Rivest Cipher), KSA(Key Scheduling Algorithm), PRGA(Pseudo Random Generation Algorithm).***

## I. INTRODUCTION

In today's world, Internet is playing a very important role in our life. It became a part of human body. A human can't live without breathing, in the same way, today he can't live without Internet [1].

TLS is debatably the most widely used secure communication protocol on the Internet in the present day [2]. As we know RC4 is a cryptographic algorithm. **Cryptography** is an art and science of preparing coded communications intended to be intelligible only to the person possessing a key. Cryptography refers both to the process or skill of communicating in or deciphering secret writings and to the use of codes to convert computerized data so that only a specific recipient will be able to read it using a key. Cryptographers call an original communication the cleartext or plaintext. Once the original communication has been scrambled or enciphered, the result is known as the ciphertext or cryptogram. The enciphering process usually involves an algorithm and a key. An encryption algorithm is a particular method of scrambling a computer program or a written set of instructions. The key specifies the actual scrambling process. The original communication may be a written or broadcast message or a set of digital data. Cryptography is important for more than just privacy, however. Cryptography protects the world's banking systems as well. Many banks and other financial institutions conduct their business over open networks, such as the Internet. Without the ability to protect bank transactions and communications, criminals could interfere with the transactions and steal money without a trace.

There are many types of cryptography, including codes, steganography, and ciphers. Codes rely on codebooks. Steganography relies on different ways to hide or disguise writing. Ciphers include both computer generated ciphers and those created by encryption methods.

### *1) Codes and Codebooks*

A well-constructed code can represent phrases and entire sentences with symbols, such as five-letter groups, and is often used more for economy than for secrecy. A properly constructed code can give a high degree of security, but the difficulty of printing and distributing codebooks books of known codes under conditions of absolute secrecy limits their use to places in which the books can be effectively guarded. In addition, the more a codebook is used, the less secure it becomes.

### *2) Steganography*

Steganography is a method of hiding the existence of a message using tools such as invisible ink, microscopic writing, or hiding code words within sentences of a message. Cryptographers may apply steganography to electronic communications. This application is called transmission security.

*3) Ciphers*

Ease of use makes ciphers popular. There are two general types of ciphers. Substitution ciphers require a cipher alphabet to replace plaintext with other letters or symbols. Transposition ciphers use the shuffling of letters in a word to make the word incomprehensible. Ciphers are the secret codes used to encrypt plaintext messages. Ciphers of various types have been devised, but all of them are either substitution or transposition ciphers. Computer ciphers are ciphers that are used for digital messages. Computer ciphers differ from ordinary substitution and transposition ciphers in that a computer application performs the encryption of data. The term cryptography is sometimes restricted to the use of ciphers or to methods involving the substitution of other letters or symbols for the original letters of a message.

**Cryptanalysis** is the art of analyzing ciphertext to extract the plaintext or the key. In other words, cryptanalysis is the opposite of cryptography. It is the breaking of ciphers. Understanding the process of code breaking is very important when designing any encryption system. The science of cryptography has kept up with the technological explosion of the last half of the 20th century. Current systems require very powerful computer systems to encrypt and decrypt data. While cryptanalysis has improved as well, some systems may exist that are unbreakable by today's standards. Today's cryptanalysis is measured by the number and speed of computers available to the code breaker. Some cryptographers believe that the National Security Agency (NSA) of the United States has enormous, extremely powerful computers that are entirely devoted to cryptanalysis.

RC4 is used in Transport Layer Security. Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide endpoint authentication and secure communications over any transport. TLS is normally associated with Internet communication but can be applied to any transport layer, including sockets and HTTP. TLS allows for two levels of security: Server Authentication and Mutual Authentication.

*4) Server Authentication*

Server Authentication authenticates the server to the client. When server authentication is used, the end user, or client, verifies that the server they are communicating with is actually who it says that it is. In the Internet world, your browser is the client, and a website such as Amazon is the server. Millions of clients need to be able to prove that the site to which they are giving financial information is really Amazon.

*5) Mutual Authentication*

Mutual Authentication authenticates the server to the client, and the client to the server. When Mutual Authentication is used, both the client and the server provide and validate certificates in order to verify each other's identity.

TLS is encryption for data in transit, not data at rest. That means that the end host or recipient in a TLS connection must be able to decrypt the encrypted traffic sent to it in order to be processed and/or displayed in the web browser. When you capture your own data using an Internet browser from a secure connection that you initiated, you are able to get at all the pieces involved in this process to examine them regardless of the methods used for encryption.

## II. REVIEW OF LITERETURE

The Transport Layer Security (TLS) protocol provides secure channels between browsers and web servers, making it critical step, TLS enables clients to verify a server's identity by validating its public-key certificate against a set of trusted root authorities. If that validation fails browsers cannot distinguish between actual attacks and benign errors (such as a server mis-configuration). Instead, browsers display a warning and ask the user to decide whether continuing is safe.

There are many researcher worked on the RC4 security areas and their presented works are as:

*1) Literature of review on Cryptography*

**Allam Mousa and Ahmad Hamad**, in 2006, Encryption is the process of transforming plaintext data into ciphertext in order to conceal its meaning and so preventing any unauthorized recipient from retrieving the original data. Hence, encryption is mainly used to ensure secrecy. Companies usually encrypt their data before transmission to ensure that the data is secure during transit. The encrypted data is sent over the public network and is decrypted by the intended recipient. Encryption works by running the data (represented as numbers) through a special encryption formula. Both the sender and the receiver know this key which may be used to encrypt and decrypt the data. Cryptography is a tool that can be used to keep information confidential and to ensure its integrity and authenticity. All modern cryptographic systems are based on Kerckhoff's principle of having a publicly-known algorithm and a secret key. Many cryptographic algorithms use complex transformations involving substitutions and permutations to transform the plaintext into the ciphertext.

They said that encryption keys fall into two catagories: Public key encryption and Private key encryption. Private

keys are also known as *symmetrical keys.* In private key encryption technology, both the sender and receiver have the same key and use it to encrypt and decrypt all messages. This makes it difficult to initiate communication for the first time. And Public key encryption, or a Diffie-Hellman algorithm, uses two keys to encrypt and decrypt data: a public key and a private key. Public keys are also known as *asymmetrical keys.* The receiver's public key is used to encrypt a message then this message is sent to the receiver who can decrypt it using its own private key. This is a one-way communication. If the receiver wants to send a return message, the same principle is used. The message is encrypted with the original sender's public key and can only be decrypted with his or her private key.

There are a variety of different types of encryption methods. But basically there are two methods of producing ciphertext: Stream cipher and Block cipher. Stream cipher is one of the simplest methods of encrypting data where each bit of the data is sequentially encrypted using one bit of the key. In order to make a stream cipher more difficult to crack, one could use a crypto key which varies in length. The main advantage of the stream cipher is that it is faster and more suitable for streaming application but its main disadvantage is that it is not suitable in some architecture. One example of the stream cipher method is the RC4 technique. Block ciphers are designed to encrypt data in chunks of a specific size. A block cipher specification will identify how much data should be encrypted on each pass (called a block) as well as what size key should be applied to each block. For example, the Data Encryption Standard (DES) specifies that DES encrypted data should be processed in 64-bit blocks using a 56-bit key.

A paper published by **James L. Massey**, in 1988, described concepts of both secret-key and public-key cryptography. Paper also described the need of cryptology. In this paper James use the Shannon's theory of secrecy and Simmons's theory of authenticity to use it into practical cryptographic systems. James used the public key concepts through consideration of Diffie-Hellman public key distribution system and the Rivest- Shamir- Adleman cryptosystem. He also described about some cryptographic protocols.

*2) Literature of review on TLS*

According to **Pratik Guha and Shawn Fitzgerald**, in 2013, various types of attacks have been designed to take advantage of select properties of the SSL/TLS architecture, design and weaknesses of the cipher suites used in SSL/TLS for encryption and key establishment.

The attacks are:

Browser Exploit against SSL/TLS (BEAST) attack

• Compression Ratio Info-leak Made Easy (CRIME) attack

• Timing Info-leak Made Easy (TIME) attack

• Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext (BREACH) attack

• Lucky 13 attack

• RC4 biases in TLS

According to **Ivan Ristić,** in 2013, SSL/TLS is a deceptively simple technology. It is easy to deploy, and it just works except that it does not, really. The first part is true—SSL is easy to deploy—but it turns out that it is not easy to deploy *correctly*. To ensure that SSL provides the necessary security, users must put extra effort into properly configuring their servers. In 2009, he began his work on SSL Labs because he wanted to understand how SSL was used and to remedy the lack of easy-to-use SSL tools and documentation.

In 2006, **Klein's** work presented a new attack strategy on WEP, which claimed to improve the complexity to 25000 packets for a 0.5 probability of success. However, these estimates were only theoretical, as no practical implementation of this attack was made in. Later in 2010-11, Sepehrdad, Vaudenay and Vuagnoux implemented this attack and verified that the practical complexity was approximately 60000 packets for Klein's attack.

In 2007, the first hands-on practical attack on WEP was demonstrated by **Tews, Weinmann and Pyshkin,** where the attack complexity is reduced to 40000 packets, with a brute force on 106 most probable secret keys. No theoretical analysis was presented in this work , but the practical results could mount a key recovery attack on WEP in under 60 seconds.

In 2013, **Sally Vandeven** said that TLS is encryption for data in transit, not data at rest. That means that the end host or recipient in a TLS connection must be able to decrypt the encrypted traffic sent to it in order to be processed and/or displayed in the web browser. Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are both protocols used for the encryption of network data. He described that how TLS works when it is used for encryption of transit data. Wireshark is a feature-rich, free tool that captures and dissects network traffic. Current versions of both the Firefox and Chrome browsers will easily save encryption keys in a file that can then be imported to Wireshark.

3) *Literature of review on RC4*

According to **Takanori Isobe, Toshihiro Ohigashi, Yuhei Watanabe, and Masakatu Morii** RC4, designed by Rivest in 1987, is one of most widely used stream ciphers in the world. It is adopted in many software applications and standard protocols such as SSL/TLS, WEP, Microsoft Lotus, Oracle secure SQL and more. RC4 consists of a key scheduling algorithm (KSA) and a pseudo-random generation algorithm (PRGA). The KSA converts a user-provided variable-length key (typically, 5–32 bytes) into an initial state *S* consisting of a permutation of *{*0, 1, 2, . . . ,*N* − 1*}*, where *N* is typically 256. The PRGA generates a keystream *Z*1, *Z*2, . . ., *Zr*, . . . from *S*, where *r* is a round number of the PRGA.

In 2004, **Korek's** work initiated the next generation of WEP attacks, where practical key recovery algorithms took the center stage. In fact, there was no theoretical analysis in, and only practical implementations of WEP attacks in the form of the Air cracking software were put forward. The attack complexity was reduced to about 100,000 packets for a success probability of 0.5.

In 2005, **Mantin** utilized the well-known Jenkins' correlation or the glimpse bias to mount a key recovery attack on RC4 in the WEP mode. Mantin's attack improved that of Fluhrer, Mantin and Shamir in terms of complexity, and remained effective even if the first *N* = 256 bytes of the keystream were discarded. No practical timings were reported for general WEP scenario, but it was claimed that 16-byte key recovery was possible in 248 steps using 217 short keystreams generated from different chosen IVs. The data complexity was estimated as 222 if the IV was concatenated after the secret key. This work also introduced the notion of fault injection in WEP attacks, and claimed that only 214 faulted keystreams could be used to recover the internal state and the secret key.

According to **Scott Fluhrer, Itsik Mantin and Adi Shamir** there are several weaknesses in the key scheduling algorithm of RC4. They identify a large number of weak keys, in which knowledge of a small number of key bits suffices to determine many states and output bits with non-negligible probability. And by using these weak keys they construct new distinguishers for RC4. They analyze the key scheduling algorithm which derives the initial state from a variable size key, and give two significant weaknesses of this process. The first weakness is the existence of large classes of weak keys, in which a small part of the secret key determines a large number of bits of the initial permutation. The second weakness is a related key vulnerability which applies when part of the key presented to the KSA is exposed to the attacker. Finally they find that RC4 is completely insecure in a common mode of operation which is used in the widely deployed Wired Equivalent Privacy Protocol.

## III. PROPOSED WORK

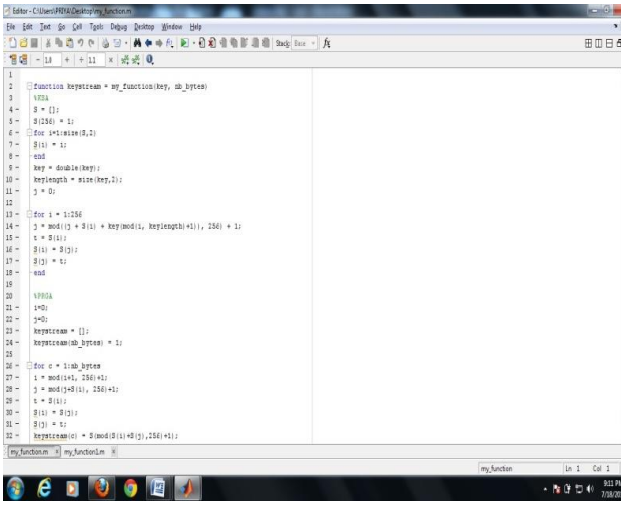
In this proposed work I make a algorithm that cause a different output then a original with better results in respect to complex to break or recover the original text from the encrypted text.

*1) Original Work of RC4 algorithm and the Results*

The RC4 algorithm

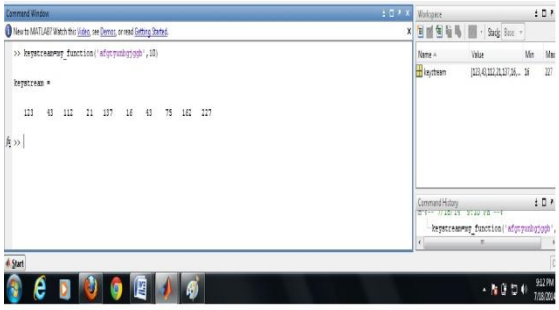
```
function keystream = my_function(key, nb_bytes)
%KSA
S = [];
S(256) = 1;
for i=1:size(S,2)
S(i) = i;
end
key = double(key);
keylength = size(key,2);
j = 0;

for i = 1:256
j = mod((j + S(i) + key(mod(i, keylength)+1)), 256) + 1;
t = S(i);
S(i) = S(j);
S(j) = t;
end

%PRGA
i=0;
j=0;
keystream = [];
keystream(nb_bytes) = 1;

for c = 1:nb_bytes
i = mod(i+1, 256)+1;
j = mod(j+S(i), 256)+1;
t = S(i);
S(i) = S(j);
S(j) = t;
keystream(c) = S(mod(S(i)+S(j),256)+1);
end
end
my_function("afgtyunbgjggb", 10)
```

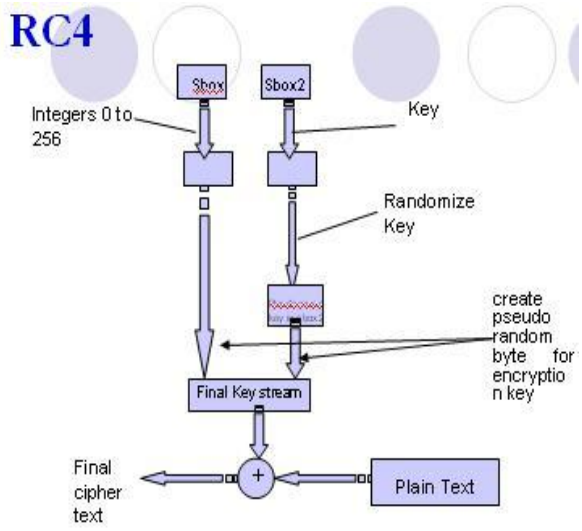
That Produces the **output** of length 10



for key "afgtyunbgjggb" and length 10

123 43 112 21 137 16 43 75 162 227






*2) Presented Work*

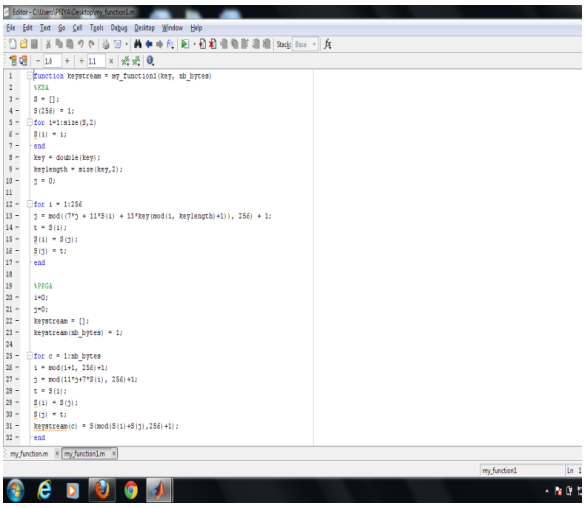
While Using this algorithm we can change the output length by changing the variable length input. But it will recover very easily by simply ARP poisoning attacks.

When the key length and the size of variable will multiply by some variable to get different result Here I make some changes in the original work to get some new result and I found that the output produced by the new algorithm of a same length input is different and not such easier to recover it.

Algorithm

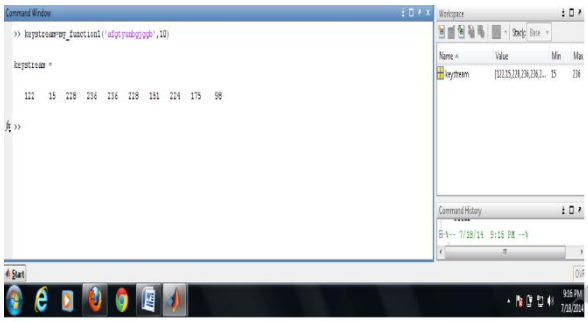
```
function keystream = my_function(key, nb_bytes)
%KSA
S = [];
S(256) = 1;
for i=1:size(S,2)
S(i) = i;
end
key = double(key);
keylength = size(key,2);
j = 0;

for i = 1:256
j = mod((7*j + 11*S(i) + 13*key(mod(i, keylength)+1)),
256) + 1;
t = S(i);
S(i) = S(j);
S(j) = t;
end

%PRGA
i=0;
j=0;
keystream = [];
keystream(nb_bytes) = 1;

for c = 1:nb_bytes
i = mod(i+1, 256)+1;
j = mod(11*j+7*S(i), 256)+1;
t = S(i);
S(i) = S(j);
S(j) = t;
keystream(c) = S(mod(S(i)+S(j),256)+1);
end
end
my_function("afgtyunbgjggb", 10)
```

That Produces the **output** of length 10



(for key "afgtyunbgjggb" and length 10)

122 15 228 236 236 228 151 224 175 98



## IV. RESULT

While evaluating the both I got some changes that if we change mod criteria by multiplying the key, variables and increment the key length by 1 the final output gives the more complex encrypted form that not been easily be recover by unauthentic person. So the key length is increased by *13 that makes key long and more powerful.

## V. CONCLUSION AND FUTURE WORK

This paper has shown that in today's world, What is the value of internet in a person's life. By using internet a person can send his/her data in any corner of the world, at any time. But internet carries many problems with it. To resolve these problems many cryptographic mechanisms are used. When RC4 is used in TLS for encryption then it is found that plaintext recovery attack for RC4 in TLS is possible for the first 256 bytes of the plaintext stream. So RC4 is not secure but still it is used for encryption due to its fast speed.

As the speed of encrypting the text is very fast by the RC4 but here it gives the good results as we need to protect the word, because today is the time for internet. In Future work I supposed to do some changes in encryption process with using the random pseudo codes or using some hash function with the both KSA and PRGA that may make a highly secure RC4.

## REFERENCES

[1] Nadhem AlFardan, Kenneth G. Paterson, Bertram Poettering, and Jacob C.N. Schuldt, Royal Holloway, University of London, Daniel J. Bernstein, University of Illinois at Chicago and Technische Universiteit Eindhoven, On the Security of RC4 in TLS, 2013.

[2] Matthew E. McKague, Design and Analysis of RC4-like Stream Ciphers, University of Waterloo, Canada, 2005.

[3] Santanu Sarkar, Sourav Sen Gupta, Goutam Paul, and Subhamoy Maitra, Chennai Mathematical Institute, Chennai, Indian Statistical Institute, Kolkata, Proving TLS-attack related open biases of RC4.

[4] Karthikeyan Bhargavan, Cedric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub, Implementing TLS with Verified cryptographic security, 2013.

[5] Marco de Vivo, Gabriela O. de Vivo, Germinal Isern, Internet Security Attacks at the Basic Levels.

[6] G. Julius Caesar, John F. Kennedy, Security Engineering: A Guide to Building Dependable Distributed Systems.

[7] ALFARDAN, N. J., BERNSTEIN, D. J., PATERSON, K. G., POETTERING, B., AND SCHULDT, J. C. N. On the security of RC4 in TLS and WPA. Information Security Group at Royal Holloway, University of London, 2013.