Volume 7, No. 7, Nov-Dec 2016
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
REVIEW ARTICLE
Available Online at www.ijarcs.info

STREAM CIPHER

Deepak Lalar
M.Tech
Sardar Patel University of Police
Security and Criminal justice Jodhpur 342304

Ravi Nahta
Guest Faculty
Sardar Patel University of Police
Security and Criminal justice Jodhpur 342304

Abstract: In this paper we discuss about the Stream Cipher. Stream cipher encrypt single bit at a time. This will happen to adding a key stream to a plain text bit. Stream cipher is small and fast, therefore its use in small embedded system. There are main two stream cipher:- A5/1 cipher for GSM mobile. Over the air for the GSM encryption is also needed to protect the conversation. Approximately the design of A5 is leaked in 1994. A5/1 was developed in 1987 and A5/2 was developed in 1989. This both were kept secret.It leaked in 1994 and reverse engineered in 1999 by Marc Briceno. RC4 is widely used stream ciher for its simplicity, speed and efficiency. RC4 is used for secure the Internet traffic. RC4 was designed in 1987 by Ron Rivest. RC4 was at first secret, however an outline of it absolutely was announce to the Cypherpunks mails in 1994. There are many attacks happens on both stream cipher, we discuss two of them in this paper.

Keywords: Encryption, PNGR, decryption, A5/1 plaintext, ciphertext, RC4.

I. INTRODUCTION

There is different types of cryptography algorithm, lets have some look of them. Cryptography divided into three types: Symmetric cipher, asymmetric cipher and protocol. Stream cipher has two parts, block cipher and stream cipher. In the diagram1 we can see that symmetric cipher have two parts, Block Cipher and stream cipher:

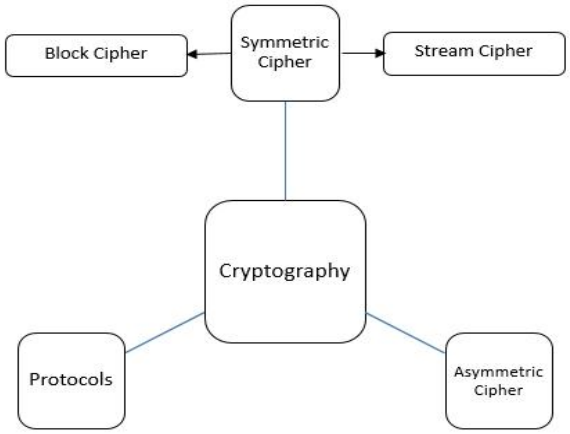

Fig. 1: area within cryptography

Stream cipher is an important class of encryption algo-rithm. Stream ciphers were fabricated in 1917 by Gilbert Vernam. Stream cipher is additionally known as Vernam ci-phers. Stream cipher encrypt plaintext bit by bit individually by adding the key stream in it. Whereas block cipher encrypt plaintext in a bit of block. Stream cipher is quicker than block cipher. Stream cipher may be either symmetric key or public key. There are synchronous stream cipher and asynchronous stream cipher. Stream cipher is small in size and it is fast then other algorithm therefore its uses in small embedded systems, e.g., cell phone. Example of stream cipher: - A5/1 cipher that is employed in GSM for cypher voice. For cypher the net trafc RC4 is employed, that an additional example of stream cipher. Stream cipher merely cypher plaintext into cipher text with adding key stream.

Stream cipher simply encrypt plaintext into cipher text with adding key stream. It XOR (modulo2) the plaintext with key stream and get cipher text. At the receiver side reverse process is done for decryption. Receiver add key into cipher text and XOR it and receive plaintext. This process is called decryption. Lets have a look for it:

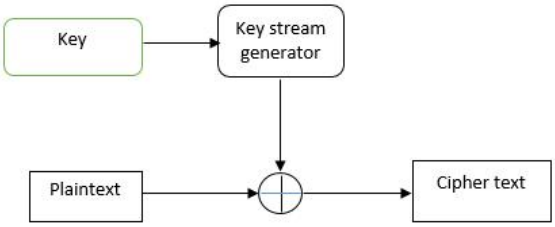

Fig. 2: Stream cipher encryption

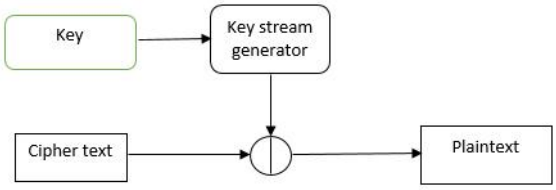

Fig. 3: Stream cipher decryption

In diagram as we see that key will enter into the key stream, where some operation will generate key stream. After that this key stream and plaintext can XOR and generate ciphertext. Its very simple to encrypt and decrypt the plaintext and cipher text, just XOR plaintext and get cipher text same for decryption, XOR cipher text with key stream and get plaintext.

Security of stream cipher is purely based on key stream. Thats why most important and difficult part of stream cipher is to generate key stream for encryption and decryption. In stream cipher randomness is use for generating key stream,

calls random number generator. Which gives more security to it, as compare to others.

A. Random Number Generator: -

Security of stream cipher is based on suitable key stream s_0; s_1; s_2.... Randomness play a major role in stream cipher. There are three types of random number generator:

1) True Random Number Generator (TRNG): [1]

True Random Number Generator produce the output, which cannot be rebuilt. Example: Example: If we tend to ip the coin a hundred times and record the ensuing sequence of a hundred bits, it'll nearly not possible to breed an equivalent bit sequence. The probability of success is extremely small, which is $1=2^{100}$.

*2) Pseudorandom Number Generators (PRNG): [1]*Pseudorandom Number Generators (PNGRs) generate keys in the sequence by using initial seed value. Its computed in following way:

s_0 = seed

$s_{i+1} = f(s_i)$; i = 0, 1…

A generalization of this are generators of the form

$s_{i+1} = f(s_i; s_{i1}, \ldots, s_{i\ t})$

Where t is a fix integer. A popular example is the linear congruential generator:

s_0 = seed

$s_{i+1}as_i + b(\text{mod}m)$; i = 0, 1, …

where a; b; m are integer constants. Most popular example is the rand() function used in ANSI C. It has the parameters:

$s_0 = 12345$

s_{i+1} 1103515245s_i + 12345(mod2^{31}); i = 0, 1; …

Mainly the PNRGs is commonly used because of its good statical properties, means their output is the approximately a sequence of true random number.

3) Cryptographically Secure Pseudorandom Number Generators (CSPRNGs)[1]:

Cryptographically Secure Pseudoran-dom Number Generators (CSPRNGs) kind a type of PNGR which have a special property, which is unpredictable. In-formally, this means that given n output bits of the key streams$_i$, s_{i+1},…, s_{i+n1}, where n is some integer, it is computationally infeasible to compute the subsequent bitss_{i+n}, s_{i+n+1},… Another denition, if n consecutive bits of key stream, there's no formula that may predict ensuings$_{n+1}$with higher than 50-50 chance of success. Another property of CSPRNG is that given the above sequence, it should be computationally impossible to compute any preceding bits si1, si2, ...s_{i1}, s_{i2}, …

B. One Time Pad [1]

It is a steam cipher in which, we use true random number generator for key stream and every key stream bit is use only once. Stream cipher which have these properties is called one time pad. Assume that we have a key length of 10,000 bits, and only brute force attack will work on it for an exhaustivekey search. If attacker want to compute the key steam than attacker needs 2^{10000} computer and every single computerwill compute single key. Of course, this is impossible to built that much computer because its too large in number. This is computationally secure not unconditionally.

OTP is unconditionally secure because we get the equation for every ciphertext bit in such form:

y_0 $x_0 + s_0(\text{mod}2)$

y_1 $x_1 + s_0(\text{mod}2)$

As we see that all the equation of one time pad is linear equation which have two unknown variables.If an attacker can get the value of any one them variable, he can't able to determine the value of another one.

C. Shift Register-Based Stream Ciphers[1]-[2]

For generating long pseudorandom sequence we use linear feedback shift register (LFSRs). It is easy to implement in hardware. Example is the A5/1 stream cipher. Number of the flip-flop in LFSR is degree of LFSR.

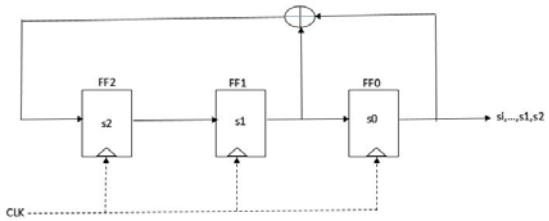

Fig. 4: Linear feedback shift register [1]

This is the LFSR with the degree of m=3 because of 3 flip-flop. Here s_i refers an internal state and it's shifted to right with each clock tick. Output bit is denoted by the rightmost bit. XOR sum of output and previous values of all flip-flop is denoted by leftmost bit. Since XOR is a linear operation, therefore its called linear feedback shift register. Let initial state of LFSR is $s_2 = 1$, $s_1 = 0$, $s_0 = 0$, than how its computes sequence of state.

In this table rightmost column is the output of LFSR. In this as we can see that after 6 clock cycle its repeat itself. It means period of length is 7 and has form: 0010111 0010111

TABLE I: Table for sequence of state of LFSR [1]

clk	FF_2	FF_1	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

0010111 ...

Let's have a look how output is computed

s_3 $s_1 + s_0$ (mod2)

s_4 $s_2 + s_1$(mod2)

s_5 $s_3 + s_2$(mod2)

.
.
.

In general:

S_{i+3} $s_{i+1} + s_i$(mod2)

where$i$ = 0,1,2,...

D. A5/1 stream cipher

A5/1 is used for securing the GSM conversation over the air. Strong version of A5 stream cipher is A5/1 and other is A5/2, which is less secure. Mostly European countries use A5/1 for encryption and most other countries uses A5/2 [3].For a long time period GSM companies kept A5/1 and A5/2 secretly. After that Briceno, Goldberg and Wagner

published reverse engineered A5/1 and A5/2 [5].

It sends the data in the form of frames and each frame contain 114 bit. This will tales 4.6 milliseconds.Every time it will use new session key for encrypt the new conversation. K will mash with the frame counter F_n, and the output will use as the initial state of a generator, that will produce pseudo random bits [4]. For generating cipher text of 114+114 bits, this bits XOR'ed by two parties with the 114+114 bits of the plain text [4].

The LFSRs are clocked in an irregular fashion. It is a stop/go clocking with the majority rules as follows[6]. Each register has a certain clocking tap, denoted C1; C2; C3; respectively. Every time the LFSRs are clocked, the three taps C1; C2; C3 determines that which LFSRs are clocked. R1 and R2 are clocked, but not R3, if C1= C2 = C3 + 1; R1 and R3 clocked, but not R2, if C1= C2 + 1 = C3; R2 and R3 clocked, but not R1, if C1 + 1 = C2 = C3; finally R1, R2, R3 are all clocked, if C1 = C2 = C3. As we see that in each step at-least two LFSRs are clocked, and the probability for an individual LFSR being clocked is 3/4 [6].

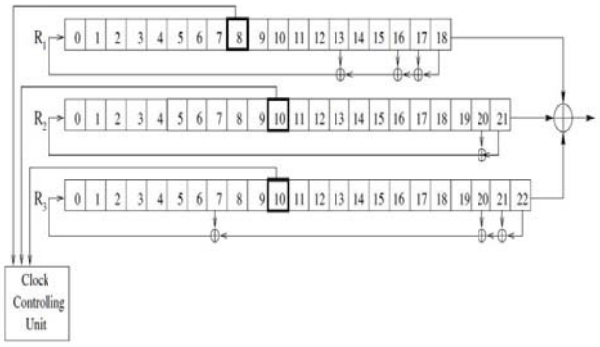

Fig. 5: A5/1 structure [3]

E. RC4

Most of the encryption in wireless is based on symmetric keyencryption, e.g. RC4. RC4 is designed by Ron Rivest in 1987 and widely used in many application and in wireless networks such as IEEE 802.11 WEP and CDPD [8]. RC4 have mainly two workingmodules: one is KSA with key k as input(40-256 bit) and another one is PRGA which generates a pseudo-random output sequence [7]. Stream of pseudo-random number generator is generated with the unique key [8]. For initialising a 256-byte array, RC4 uses a variablelength key from 1 to 256 bytes. For generating pseudorandom bytes and psedurandom stream array is used, which XOR with the plaintext and give ciphertext or XOR with ciphertext and gives plaintext [8]. RC4 is fast and efficient. It is also very fast since it uses only 7 CPU clock cycles per byte of output on Pentium CPU architecture [8]. RC4 use Wired Equivalent Privacy (WEP) protocol for providing security inwireless local are network (WLANs).

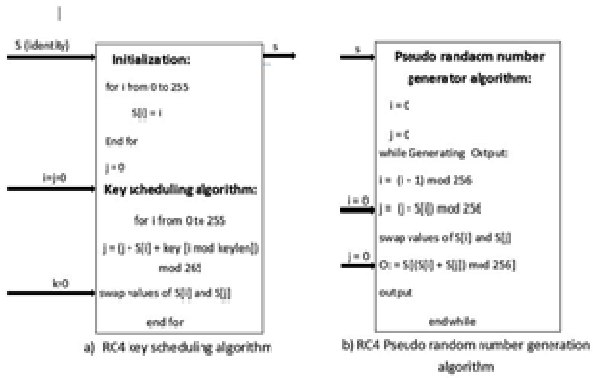

Fig. 6: RC4 steam cipher [7]

II. ATTACKS ON STREAM CIPHER

In this part we are discuss about some of the attacks which happen on stream cipher.

A. Golic Time-Memory Trade-off Attack [4][9]

A5/1 has a weakness, since it has $n = 2^{64}$ and state define by the19+22+23=64 bits in its three registers [4]. The base of the Golictime-memory tradeoff is to get the maximum set of states which ismaximum states of algorithm when output is generating. Anyinterruption between both state will help to get actual state of algorithm by stored data [4].

If the attacker can determined internal state at the time 101 t 151, then the attack consist with the reverse pro-cessing of the internalstate to S(101) based on known output, then to S(0) when the output is unknown, and finally to the secret session key when the known public key is incorporated [9].

If the attacker can determined internal state at the time 315 t 365, then the attack consist with the reverse pro-cessing of the internal state to S(315) based to known output, then to S(214) when the output is unknown, and the rest is same aswell as in case fist with S(214) as the internal state[9]. More than one candidates for the secret session key are then easily reduced to only one[9], correct solution by comparing a small number of already known keystream sequences with the ones generated from the assumed candidates and known public keys [9].

B. Brute force attack [10]

1) Software implementation:RC4 is mainly designed for software, therefore this implementation is very easy. The parameters of the ciphertext and the plaintext is taking by the program which implement the RC4 algorithm. Starting with the smallest possible key (0x00000) and incrementing by one each iteration, the program eventually reaches the key which properly decrypts the message [10]. This implementation is very scaleable and easily extended to a cluster of PCs by dividing up the keyspace among participating computers [10].

2) Hardware Implementation: The implementation was re-alized on a Xilinx Virtex II XC2V1000 FPGA using tools from Celoxica [10]. It have two distinct components: Key-Checker Unit and Controller. The checker-unit has pair of adders, XOR gate, and pair of counter.Checker unit will

check a single bit independently. Using multiple checkers in a network therefore results in an adjustable level of parallelism easily modified by adding or removing key-checker units from the design[10].

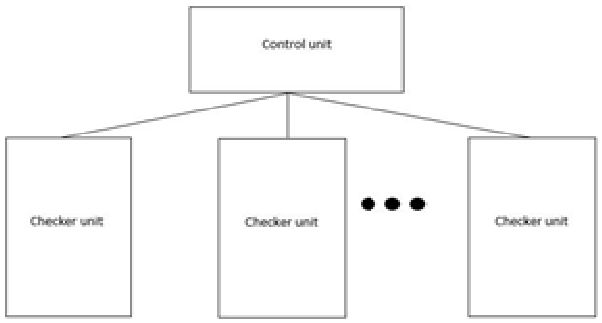

Fig. 7: Block Diagram of the Hardware RC4 Key-Search Machine [10]

Implementation of the RC4 decryption algorithm is contains by key-checker.Each unit has two port, input and output. Input port get the single key and after that decrypted result sendto theoutput port. Algorithm will start with the initialization of array S and K. S and K is very small memory unit. Initialization of S involves assigning the values of S to 0 through 255 sequentially and takes 256 clock cycles due to the restrictions on memory access [10].Similarly, bytes of the key fill K. After optimization, size of key is reduced by the K (40 bytes in this case), and read with the help of modulo addressing. This was done by reduce the number of clockcycles which is required to check a single key. After initializing S and K, the next step is to permute S based on the values contained in K. If swap takes 4 clock cycle, one to read S[i], one to read S[j] and a clock to rewrite each of them to the opposite locations[10], initialization is perform by using many clock. Overall, 1300 clock cycle are used by single key for checking process; initialization of S and K takes 516 clock cycles, pseudo-random permutation of S takes 770 cycles and 7 cycles for generation of the pseudo-random bytes[10].

During the design, each step of decryption must consider number of clock cycles, From this the performance of key-search machine is effected. A large circuit which have more key-checker units will negatively effect the clock rate and should be operate, such that perfect solution can be implemented.

III. CONCLUSION

In this paper we discussed about stream cipher. We read about how encryption and decryption will happen in stream cipher. Streamcipher use different types of randomness for encryption, which makes its more secure. It uses True random number generator, pseudo random number generator (PRNG), cryptographically secure pseudo random number generator (CSPRNG) and one time pad. For generating pseudo random number it uses three linear shift registers

As we seen that there mainly are two types of stream cipher: A5/1 and RC4, which are respectively use for security in GSM and Internet traffic. After that attacker can easily able to break the security of it. There are many attack which applies on this, we discussed two of them. Golic time memory trade off attack and brute force attack. In brute force as se saw that it can easily make by the software as well as hardware. If we increase the size of hardware we can increase the security of RC4,but the clock rate decrease if increase the size of circuit. Therefore we have to find such types of checker-units which will increase the performance. My direction of work is to prevent A5/1 from Golic Time-Memory Trade-off Attack and RC4 from Brute force attack

IV REFERENCES

[1] ChristofPaar and Jan Pelzl, ”Understanding Cryptography”, Springer-Verlag Berlin Heidelberg, London New York, pp. 29-54, 2010.

[2] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, ”Handbook of applied cryptography”, CRC Press, August 2001

[3] E. Biham, O. Dunkelman, ”Cryptanalysis of the A5/1 GSM stream Cipher”, Lecure Notes in Computer Science, vol. 1977, 2000, pp. 43 51, (Indocrypt 2000).

[4] Marc Briceno, Ian Goldberg, David Wagner, ”A Pedagogical Implementation of A5/1”, 1999.

[5] PatrikEkdahl and Thomas Johansson, ”Another Attack on A5/1”, Submission to IEEE Transactions on information theory, 2002.

[6] Poonam Jindal, Brahmjit Singh, ”A Survey on RC4 Stream Cipher”, Morden Education and Computer science Press, I. J. Computer Network and Information Security, 2015

[7] P. Prasithsangaree and P. Krishnamurthy, ”Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs”, IEEE, 2003.

[8] Jovan Dj. Golit,”Cryptanalysis of Alleged A5 Stream Cipher”, Springer, Verlag Berlin Heidelberg, 1997.

[9] Nathaniel Couture and Kenneth B. Kent, ”The Effectiveness of Brute Force Attacks on RC4”, IEEE, Proceedings of the Second Annual Conference on Communication Networks and Services Research, 2004.

[10] Celoxica, HandelC Reference Manual.

[11] P. Kundarewich, S. Wilton and A.J. Hu, A CPLDbased, ”RC-4 Cracking System”, Canadian Conference on Electrical and Computer Engineering 1999, 1999.

[12] B.Schneier, ”Applied Cryptography”, John Wiley & Sons, Second Edition, 1996, pp 397-400.

[13] Scott Fluhrer, ItsikMantin, and Adi Shamir, ”Weaknesses in the Key Scheduling Algorithm”, Springer, Springer-Verlag Berlin Heidelberg, 2001.