A MODIFIED STREAM GENERATOR FOR THE GSM ENCRYPTION ALGORITHMS A5/1 AND A5/2

Imran Erguler[1,2], and Emin Anarim[2]

[1]Department of Electronics and Communications Engineering, Dogus University
Acibadem 34722 Istanbul-Turkey
phone: + (90) -216-3271106 , email: ierguler@dogus.edu.tr
[2]Electrical-Electronics Engineering Department, Bogazici University
Bebek 34342 Istanbul-Turkey
phone: + (90) -212- 3595400-6414, email: anarim@boun.edu.tr

ABSTRACT

A5/1 and A5/2 are the GSM encryption algorithms that protect user data transmission over air. However, both of the A5/1 and A5/2 were cryptanalized by using different attack techniques such as time-memory trade off, divide and conquer and correlation attacks. In this study, we present a modified version of the A5/1 and A5/2 with offering security improvements to the vulnerabilities of the algorithms. By changing just the clocking mechanism of the shift registers used in the algorithms, it is shown that known attacks techniques become impractical.

1. INTRODUCTION

Encryption in mobile communication is very crucial to protect information of the subscribers and avoid fraud. GSM uses A5 stream generator to encrypt digital user data transmitted from mobile station to the base station and base station to the mobile station. A5 stream cipher has two major variants: A5/1 is the stronger version used in western European countries and A5/2 is the weaker version used in the other countries.

Attacks against A5 algorithms have been presented in different papers [1-6]. In [1], it is shown that cryptanalysis of A5/1 can be performed on a single PC with a few minutes of computational time and about 150-300 Gbytes of memory. An attack against A5/2 has been also discussed in [5]. According to the paper, A5/2 can be easily cryptanalized about 2^{17} time complexity.

In this paper, we propose a modified version of A5 stream cipher with providing security enhancements that make the time-memory trade-off attacks and divide & conquer attacks impractical. The improvements that increase the security of the algorithm mainly based on the characteristic clocking mechanism of the proposed model.

2. A BRIEF DESCRIPTION OF A5/1 AND A5/2

In GSM, each conversation is sent as a sequence of frames every 4.6 millisecond. Each frame contains 114 bits for communication from the mobile to base station and 114 bits for communication from base station to the mobile. For each conversation user data can be encrypted by a new secret session key K_c and for each frame, K_c is mixed with a publicly known frame counter F_n. The result of this process serves as the initial state of A5 stream generator which produces 228 pseudo random bits. These generated 228 bits are XOR'ed by 228 bits of the plaintext to produce 228 bits of the cipher text.

2.1 Description of A5/1

A5/1 stream cipher is a binary linear feedback shift register (LFSR) based key stream generator. It combines three LFSRs of lengths 19, 22 and 23 bits which are denoted by R1, R2 and R3 respectively [1]. All of these registers have primitive feedback polynomials and each register is updated according to its own feedback polynomial. The taps of R1 are at bit positions 13, 16, 17, 18; the taps of R2 are at bit positions 20, 21; and the taps of R3 are at bit positions 7, 20, 21, 22. The three registers are maximal length LFSRs with periods 2^{19} -1, 2^{22} - 1, and 2^{23} -1, respectively. The output of A5/1 is produced by XOR'ing the most significant bit of each register as shown in Figure 1.

Each LFSR has a single clocking tap in bit 8 for R1, bit 10 for R2 and bit 10 for R3; denoted as C1, C2 and C3 respectively. Clocking mechanism of each LFSR is determined according to the majority rule: Each clock cycle majority of C1, C2, and C3 is calculated and two or three registers whose clocking tap value is the same as majority bit are clocked [1]. Since at each clock cycle at least two LFSRs are clocked, an individual LFSR moves with probability 3/4 and stops with probability 1/4.

The initial state of A5/1 is carried out as follows: All of the registers are first zeroed and then 64 bit secret session key K_c and 22 bit frame number F_n XOR'ed (ignoring majority rule) in parallel into the least significant bits (lsb) of the three registers. In the next step all LFSRs are clocked for 100 clock cycles according to majority rule, however no output is produced. Finally, three LFSRs are clocked according to majority rule to generate 228 bits of key stream sequence.

2.2 Description of A5/2

A5/2 is built from four LFSRs of lengths 19, 22, 23, 17 bits denoted by R1, R2, R3 and R4 respectively, shown in Figure 2. Each register is updated by its own primitive feedback polynomial which is given in [6]. The feedback functions of

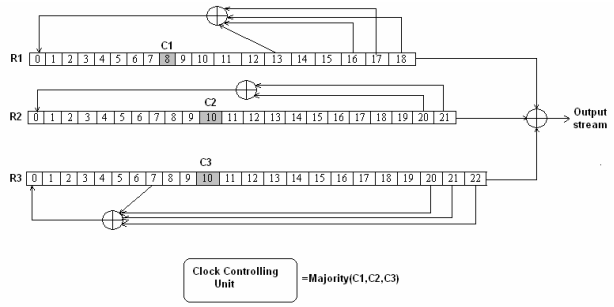

Figure 1 - The A5/1 Stream Cipher

R1, R2 and R3 of A5/2 are the same as those of R1, R2 and R3 of A5/1. Clocking of R1, R2 and R3 is controlled by R4 and R4 is regularly clocked in each clock cycle. Majority of the bits, R4(3), R4(7) and R4(10), is calculated and a binary result according to majority rule is obtained. If the result is the same as R4(3), then R2 is clocked, if the result is the same as R4(7), then R3 is clocked and if the result is the same as R4(10) then R1 is clocked. After the clocking of R1, R2 and R3, R4 are clocked. It is obvious that, the clock control mechanism of both A5/1 and A5/2 depends on majority rule. However inputs to clocking control mechanism are given from R4 in case of A5/2, while in A5/1 inputs are from R1, R2 and R3. The output bit is generated as follows: in each register majority of two bits and complementary of a third bit is calculated; the results of all the majorities and the right most bit from each register are XOR'ed to form the output bit.

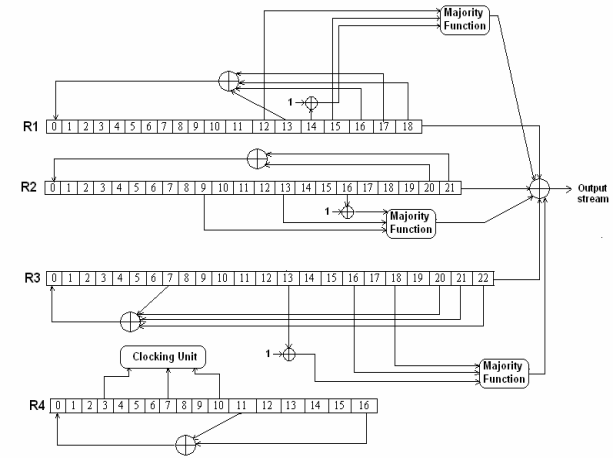

Figure 2 - The A5/2 Stream Cipher

3. THE PROPOSED STREAM CIPHER MODEL

Most of attacks against A5/1 and A5/2 make use of the security flaws in clocking mechanisms of the algorithms. For A5/2, R4 controls clocking of the other LFSRs. So if the initial state of R4 is determined, for each known output bit, exact number of times that a register is clocked to generate the output bit can be known. As a result with the knowledge of feedback polynomial of the four LFSR's, every bit of the output can be expressed as a quadratic multivariate function in the initial state and the attack presented in [6] can be applied. There are also attacks on A5/1 due to its poor clocking mechanism, although it is the strong version. In [2] and [3] divide & conquer attacks have been applied. According to these studies: Since the clocking tap positions of R1, R2 and R3 are known, linear equations about the LFSRs content can be obtained by guessing the some bits before the clocking tap bits. It is shown that by using these linear equations A5/1 can be cryptanalized.

To overcome the above problems, we offer a new clocking mechanism for A5 stream generator. The proposed generator consists of four LFSRs denoted as R1, R2, R3 and R4 respectively, depicted in Figure 3. These registers are chosen the same as those of A5/2 to avoid hardware efficiency loss.

The new stream generator works as follows:

For each clock cycle output of R4, R4(16), is checked; if it is 1 then majority of clocking taps R1(14), R2(9) and R3(3) denoted as C1,C2 and C3 respectively is calculated. The registers whose clocking tap value agrees with the majority result are clocked. However if the R4(16) is 0 then majority result is not applied, instead the following match rule is used. In the match rule three function values, F1, F2 and F3 that are related to R1, R2 and R3 respectively are computed as shown below.

$$F1 = R1(3) \oplus R3(7) \oplus R4(8)$$

$$F2 = R2(16) \oplus R1(6) \oplus R4(12)$$

$$F3 = R3(11) \oplus R2(13) \oplus R4(3)$$

If F1 is 1, then R1 is clocked, if F2 is 1 then R2 is clocked and if F3 is 1, then R3 is clocked. The register, whose related function value is 0, is not clocked. However, if all of F1, F2 and F3 are 0, all of the three registers are clocked to avoid stopping of R1, R2 and R3 at the same time.

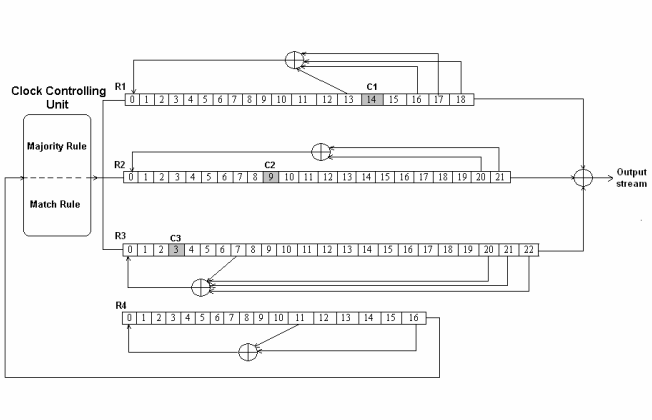

Figure 3 - The Proposed Stream Cipher

After the clocking of R1, R2 and R3 is done (according to majority rule or match rule), R4 is clocked once. The output of the generator is generated by XOR'ing the most significant bits of R1, R2 and R3, as in the case of A5/1.

The whole clocking mechanism is summarized in Table 1. As it is mentioned before, inputs to clocking control mechanism are given from only R4 in case of A5/2, and in A5/1 inputs are from R1, R2 and R3. On the other hand, for the new clocking model all of the four LFSRs give input to the

clocking mechanism which provides security enhancements against to divide and conquer attacks.

		C1	C2	C3	F1	F2	F3	One CLK
R4 (16) = 1 (Majority Rule)	Case 1	0	0	0				R1,R2,R3
	Case 2	0	0	1				R1,R2
	Case 3	0	1	0				R1,R3
	Case 4	1	0	0				R2,R3
	Case 5	0	1	1				R2,R3
	Case 6	1	0	1				R1,R3
	Case 7	1	1	0				R1,R2
	Case 8	1	1	1				R1,R2,R3
R4 (16) = 0 (Match Rule)	Case 9				0	0	1	R3
	Case 10				0	1	0	R2
	Case 11				1	0	0	R1
	Case 12				0	1	1	R2,R3
	Case 13				1	0	1	R1,R3
	Case 14				1	1	0	R1,R2
	Case 15				1	1	1	R1,R2,R3
	Case 16				0	0	0	R1,R2,R3

Table 1 - Clocking of R1, R2 and R3

4. SECURITY OF THE STREAM GENERATOR

In cipher design, the essential point that a designer has to consider is resistance of the cipher against different attacks. Therefore, in this section we investigate some known attacks applied on A5 with respect to the proposed stream generator to show its security enhancements.

4.1 Time memory trade-off attacks

In Eurocrypt 1997, Golic presented a time-memory trade-off attack against A5/1, based on birthday paradox yielding the unknown internal state at a known time for a known key stream sequence [2]. In fact the main principle of this attack is the same as time-memory tradeoff of S. Babbage's idea [7]. Therefore it is usually known as BG time memory trade-off attack. If N represents total number of solution space for LFSR's internal state, M represents amount of required memory and T represents required computational time of the attack, it has been shown that solution space N can be distributed between time T and memory M, where the inequality $TM \geq N$ must be hold for the success of the attack. Since total length of A5/1 stream cipher is 64-bit, there are 2^{64} different possibilities for the internal states of three LFSRs. However there are 3/8 unreachable states from internal states, so this number reduces to about $2^{63.32}$ as total solution space N according to [2]. For a given 2^{21} known different key stream sequences corresponding to 2^{21} different frames, the necessary memory and computational time are obtained as $M \approx 2^{35.65}$ and $T \approx 2^{27.67}$ respectively [2].

In the proposed stream generator, there are four LFSR's of total length 81, so according to this model N becomes as 2^{81}. Considering the new algorithm, the memory and time requirements according to the approach in [2], one can show that, cryptanalysis of the proposed model needs 2^{17} =131072 times more memory or 2^{17} times more time. Both of the cases seem infeasible, because as the memory increases from $2^{35.65}$ to $2^{52.65}$, it becomes about 130153 terabyte. Also according to this attack, computational time can not exceed $2^{28.67}$, since at most 2^{22} different frames can be used with the same K_c. Therefore required increase in computational time is also impossible with this idea. So Golic's attack is impractical with regarding the new stream generator.

A. Biryukov et al. have presented an improved time-memory trade-off attacks as biased birthday attack [1]. This attack makes an improvement to Golic's time-memory trade-off by defining a specific pattern which is generated by preprocessing many samples. Then in the case of occurrence of this pattern in output key stream sequence, possible internal states are found. According to [1], if R represents number of special states (red states) that are kept on disk, t represents available known conversation as number of known key stream corresponding to different frames, k represents length of specific pattern and $W(s)$ represents weight of a special state s, then the expected number of colliding special states in the disk and the actual data is expressed as:

$$\frac{RW(s)t}{2^N} \qquad (1)$$

where N is total number of solution space for LFSR's internal state. In [2], the results have been given for $R=2^{35}$, $W(s)=12500$, t = 120*1000/4.6= 26087 about 2 minutes of known conversation and $N=2^{64}$ due to total length of A5/1. For these values (1) becomes 0.61 which makes it quite likely that a collision will actually exist. The requirements of the attack are 146 gigabytes as memory and two minutes as length of known conversation.

This attack can be considered for the proposed generator. However as can be seen, (1) becomes 2^{17} times smaller due to large value of $N=2^{81}$ for the new model. To solve this problem, the attacker has to increase multiplication of number of special states kept on disk and the length of the conversation 2^{17} times more. If the attacker makes a trade off between time and memory and he can change the requirements as 18.25 terabytes for memory and 1.4 days for known conversation length. Neither requirement seems to be sensible, so we can say that this attack is also impractical with regarding the new model.

4.2 Divide-and-conquer attacks

Divide-and-conquer attacks work by guessing a part of the secret internal state of the stream cipher and then deducing the unknown part of the state. So it is something like; guess some bits determine the others. Golic has described a divide and conquer attack on A5/1 stream cipher in [2]. The main idea of the attack is getting 63.32 linear equations and then obtaining internal state of LFSR's by solving these equations. According to this attack, firstly n bits from each register is guessed and from these bits, attacker infer $3n$ linear equations, where n can not be bigger than the shortest distance between clocking tap bit and output bit of LFSR. From this, the attacker obtains about $4n/3$ linear equations, since assumed bits on average define $4n/3$ elements of clock control sequence and output is known [2]. Also known first output bit gives extra one linear equation, so totally attacker can have $3n+ 4n/3+1$ linear equations. In [2], n is chosen as 10, for this value number of total linear equations is 44.33. However required number of equations is 63.32, so 19 more equations are required. Golic noticed that if the attacker builds a tree structure that store all the valid possibilities for the three bits that are input to clock control function, not all 2^{19} options need to be considered. This factor results in about $2^{11.16}$ amount of time, hence the over all complexity becomes about $2^{40.16}$ linear equation set solving which is enough to obtain internal state of LFSRs.

In [3], E.Biham et al. described a different type of divide and conquer attack as "Basic Attack" which requires $2^{39.91}$ total work complexities with 2.36 minutes of conversation plaintext. According to the attack, it is assumed that 10 consecutive rounds the third register (R3) is not clocked may occur during the conversation. With this assumption and guessing 12 bits (9 bits of R1, R2(0) and R3(22)), the attacker recovers all the bits of R1 and R2. This step requires 2^{12} amount of time. In the second step attacker guesses another 10 bits of R3, to get the remained bits of R3. Second step costs to the attacker as 2^{15} workload. Therefore expected running time of this attack is 2^{27} given the R3 where location is static [3]. Of course location is unknown, so attacker needs to examine 2^{20} possible starting locations. Total time complexity of the attack becomes about 2^{47}. By presenting some techniques for the attack, they lower the time complexity of the attack about to $2^{39.91}$.

In both of the attacks mentioned above some bits are guessed, then by using clocking mechanism of A5/1, linear equations are obtained and unknown bits of LFSRs are recovered. If anyone attempts to apply a similar technique on the proposed cipher, he has to guess all bits of R4 to understand which rule for the clocking mechanism (majority rule or match rule) of R1, R2 and R3 is used. Otherwise guessing any small portion of the R1, R2 or R3 does not appear to help to determine the initial state of the generator. Guessing R4 results in 2^{17} extra computational time, so totally attack requires about 2^{57} time complexity. Furthermore, since the clocking rule changes depending on R4, some of the obtained linear equations from guessing process are not linearly independent, so it reduces efficiency of the attack. Therefore, divide-and conquer attacks on the proposed stream generator seem unlikely to work in practical conditions.

5. PROPERTIES OF STREAM SEQUENCE

5.1 Statistical properties

A stream sequence can be unsuitable for cryptographic applications, if the sequence does not provide good statistics (randomness). The proposed stream cipher is investigated against some important statistical tests FIPS140-2 and autocorrelation test. FIPS140-2 statistical test contains the Monobit Test, the Poker Test, the Runs Test, and the Long Run Test [8]. These tests are based on performing a pass/fail statistical test on 10000 sequences of 20000 bits each produced by our proposed stream cipher (software implementation in MATLAB). The stream generator passes FIPS140-2 in proportion of 99.71%. Also autocorrelation test (Golomb's 3rd postulate) is applied to the new model and the resulting autocorrelation of one sequence is shown in Figure 4. The autocorrelation is normalized to one at the origin. As we can see, there is no significant peak compared to the value at the origin. From the test results, we can say that the output generated by the proposed generator satisfies the randomness properties.

5.2 Output rate

By looking at the Table 1, one can realize that each of R1, R2 and R3 is clocked once in 11 cases of the 16 cases. So, each of R1, R2 and R3 is clocked once with a probability of 11/16 which is very close to the probability 3/4 for A5/1.

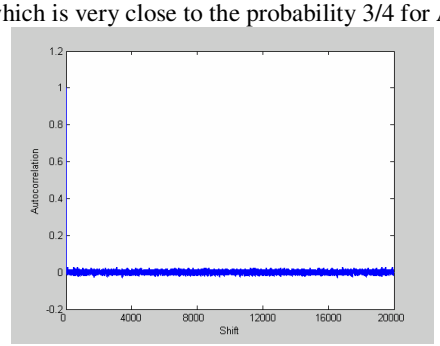

Figure 4 – Autocorrelation test result

Therefore for each clock cycle of a generator register (R1, R2 and R3), $16/11 \approx 1.45$ output bits are generated. Output rate for A5/1 is $4/3 \approx 1.33$. As can be seen the new algorithm is a little faster than A5/1 with respect to output rate.

6. CONCLUSIONS

In this paper, we have described a modified stream generator model for the GSM encryption algorithms A5/1 and A5/2. The LFSR's and primitive polynomials of this stream generator are the same as those of A5/2. By just changing the clocking mechanism of A5, the new generator provides a cryptographically more secure stream cipher with respect to some popular attacks as "divide and conquer" and "time memory trade-off". Furthermore, the period of the proposed generator is higher compared to A5/1. This is a nice design characteristic for a stream generator that has to be suitable for cryptographic applications.

REFERENCES

[1] A. Biryukov, A. Shamir and D. Wagner, "Real time cryptanalysis of A5/1 on a PC", Proceedings of FSE'2000, *Lecture Notes in Computer Science*, vol. 1978, pp. 1–18, 2001.

[2] J. Golic, "Cryptanalysis of alleged A5 stream cipher", Proceedings of Eurocrypt'97, *Lecture Notes in Computer Science*, vol. 1233, pp. 239-255, 1997.

[3] E. Biham and O. Dunkelman, "Cryptanalysis of the A5/1 GSM stream cipher", Proceedings of Indocrypt'2000, *Lecture Notes in Computer Science,* vol. 1977, pp. 43–51, 2000.

[4] P. Ekdahl and T. Johansson," Another Attack on A5/1", *IEEE Transactions on Inform. Theory,* vol. 49, pp. 1-7, 2003.

[5] S. Petrovic and A. Fuster-Sapater,"Cryptanalysis of the A5/2 Algorithm", Cryptology ePrint Archive, Report 2000/052, http://eprint.iacr.org,2000

[6] E. Barkan, E. Biham and N. Keller, "Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication", Proceedings of Crypto'03, *Lecture Notes in Computer Science.*, vol. 2729, pp. 600-616, 2003.

[7] S. Babbage, "A Space/Time Trade-Off in Exhaustive Search Attacks on Stream Ciphers", *European Convention on Security and Detection IEE Conf. Pub.,* no.408, May 1995.

[8] FIPS PUB 140-2 Security Requirements for Cryptographic Modules, http://csrc.nist.gov/cryptval/140-2.htm