8th International ISC Conference
on Information Security and Cryptology
(I S C I S C ' 1 1)
Ferdowsi University of Mashhad-September 14-15, 2011

An Improved Attack on A5/1

Vahid Amin Ghafari
Information and Communication Technology complex
Malek Ashtar University of Technology
Tehran, Iran
vahidaming@yahoo.com

Javad Mohajeri
Electronic Institute
Sharif University of Technology
Tehran, Iran
mohajer@sharif.ir

***Abstract*—A5/1 is a stream cipher used in GSM to provide over-the-air communication privacy. Biham and Dunkelman proposed an attack on A5/1 with time complexity of 2^(39.91) and data complexity of 2^(21.1) known bits and memory complexity of 32 GB. In this paper, we propose an improvement on their attack. Our improvement is identification and elimination of useless states from the precomputed table. Furthermore, we propose another way for use of table in online phase of attack that causes decreasing in the time complexity to 2^(37.89) and memory complexity decreases to half.**

***Keywords-** A5/1; GSM; stream cipher; precomputed table; useless states;*

I. INTRODUCTION

A5 is a family of encryption algorithms that are used to protect the privacy of conversation in the GSM mobile phone system. Over half a billion customers in the world are protected from eavesdropping by using, A5/1, a stronger version of this family. Since Briceno et al published their paper in 1999 (that the design of A5/1 was pointed out with reverse engineering); many attacks have been proposed on A5/1 [5]. Attacks against A5/1 are divided into two categories: active and passive. Passive attacks can be divided into three classes: guess-and-determine (GD), time-memory-data tradeoff (TMDTO) and correlation attacks. GD attacks on A5/1 require high time complexity [7,10,8,11], while TMDTO attacks on A5/1 require high precomputation or data complexity [7,4,2], and Correlation attacks on A5/1 require many known plaintexts or ciphertexts [6,9,1].

Biham and Dunkelman proposed a guess-and-determine attack on A5/1 in 2000. They improved this attack by exploiting a precomputed table in their paper. Their attack on A5/1 requires 2^(21.1) bits of known plaintext and 2^(39.91) of A5/1 clockings [3].

In this paper their attack is improved by identification and elimination of useless states from the precomputed table. Almost half cases of the precomputed tables are useless, and they can be eliminated from the table. By using elimination of these useless states and a proposed way for use of table in the online phase of attack, the time complexity decreases to 2^37.89, and memory complexity decreases to half.

Table I is presented to compare the effect of eliminating the useless states from precomputed tables. The results in Table I are based on the assumption that each frame (of length 114 bits) is related to one loading.

TABLE I. ATTACK ON A5/1 THAT PRESENTED IN [3] AND IMPROVEMENT OF THEM

Attack	Precomputation complexity	Time complexity	Data complexity (known bits)	Memory Complexity	Success Rate
Biham & Dunkelman	2^38	2^39.91	2^21.1	32 GB	63%
Our improvement	2^38	2^37.89	2^21.1	≈ 16 GB	63%

The paper is organized as follows: Section 2 contains a description of the A5/1 algorithm. Early attack on A5/1 is surveyed in Section 3 and contradictory states are presented in Section 4. Improvement of the attack in online phase is discussed in Section 5. Finally, we summarize and conclude the paper in section 6.

II. DESCRIPTION OF THE A5/1 STREAM CIPHER

A5/1 consists of 3 LFSR of lengths 19, 22, 23, which are denoted by R_1, R_2, R_3 respectively. The output is generated by XOR-ing of the most significant bits (MSBs) of the three registers. Then, the value of three bits $R_1[8]$, $R_2[10]$, $R_3[10]$ (clock-controlling bits (CCBs)) enter into the clock controlling unit and their majority value is obtained. Each LFSR is clocked if its clock bit is equal to this majority value. Note that at each clock cycle at least two registers are clocked, and each register will be clocked with the probability of $3/4$. In Figure 1, number 0 is allocated to the least significant bit of each register. A5/1 takes two parameters as input for initialization, a 64 bit secret session key K_c and a 22 bit frame number F_n. First, the LFSRs are initialized by zero. Then all registers are clocked 64 times regularly, and the successive bits of K_c are consecutively XORed into the LSB of each registers in parallel. In the second step, the registers are clocked 22 times regularly and the successive bits of F_n again XORed into the LSB of each registers in parallel. In the third step, the algorithm is clocked for 100 clocks with the majority clocking mechanism, and discards the output. Finally, the algorithm produces 228 bits of running key.

Each mobile phone in GSM network sends frames every 4.6 millisecond to network and each frame consists of 228 bits.

The first 114 bits are used for encryption data from network to mobile phone, and the second 114 bits used for encryption data from the mobile phone to network. Note that we suppose in each loading of A5/1, an attacker can access a single direction. Thus each 114 bits is relevant to one loading.

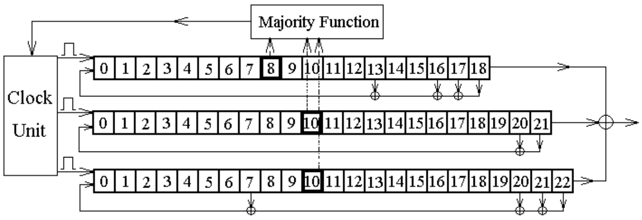

Figure 1. The A5/1 internal structure

III. EARLY ATTACK ON A5/1

In [3] an attack was applied to A5/1 that requires 2^(21.1) bits of known plaintext, 2^38 preprocessing of A5/1 clockings, 32 GB memory and has a time complexity of 2^(39.91) A5/1 clockings, which recovers the internal state of the algorithm. The attack's main idea was to wait until an event which leaks a large amount of information about the internal state occurs. The proposed attack can be described in the following steps. Suppose that for 10 consecutive clock cycles, register R_3 is not clocked.

In the first step of attack, R_1[9,10,11,12,14,15,16,17,18], R_2[0] and R_3[10,22] are guessed and then all bits of R_1 and R_2 are recovered. In the second step, the attacker refers to the precomputed table and recovers remaining bits of R_3. This table, by the 5 known bits of the output stream and the 20 bits of R_1 and R_2 (5 MSBs and 5 CCBs from each); contain the possible values for the 10 bits from R_3(5 MSBs and 5 CCBs). The average number of candidates for each access to this table is 2^(4.53) [3].

The attacker should examine 2^20 possible starting locations until, in one of them with a high probability, for 10 consecutive clock cycles the third register is not clocked (the probability that third register is not clocked in one clock cycle is equal to $1/4$ and probability that the third register is not clocked for 10 consecutive clock cycles is equal to 2^(−20)). Each 64 known bits of successive output stream is sufficient for recovering the internal state of the algorithm [7]. In each 114 output stream bits, there are 51 $114-64+1=51$ (overlapping) strings of 64 consecutive bits. Therefore, using 114(2^20)/51 = 2^21.1 known bits, one can apply the attack with a success rate of about $1-(1-2^{-20})^{2^{20}}=63\%$.

For each 2^20 strings, there are 2^12 possible cases for guessing (in the first step of attack) and for each case, in first access to the table, there are 2^(1.53) candidates on average (note that the first time attacker access to the table three bits of R_3 are known). Then, the attacker clocks the registers as many times as needed according to the table access. The cost of the first clock is equivalent to two A5/1 clockings. Then, in the second access to the table, attacker gets 2^(4.53) candidates on average, and attacker clocks the registers as many times as needed [3].

For the other 3 bits of R_3, attacker builds another table by 6 bits of each register (because access to the previous table again would cost us too much). This table gives the attacker for each access 2^(2.82) = 7.1 candidates on average. Thus attacker have on average 2^(−0.18) ≈ (0.88) candidates for the rest of the 3 unknown bits of R_3.

Now, the wrong candidates must be discarded. For checking whether this is a right state, two clock cycles for each candidate are required. Therefore, the time complexity of the attack is 2^20 × 2^12 × 2^1.53 × 2 × 2^4.53((1 + 1) + 2 × 0.88) = 2^40.97 A5/1 clockings ((1+1) is relevant to work before access to the second table) [3].

The authors showed that this attack can be improved by using a large table. The time complexity of the attack decrease to 2^(39.91) A5/1 clockings by using the table based on 12 bits from R_1, R_2 and 5 bits of the output stream [3].

IV. CONTRADICTORY STATES

There is an important point that has not been mentioned in [3]. In the first access to the table, MSBs of R_1, R_2 as well as the output stream are known, thus MSB of R_3 can be obtained. Then if R_3 is not clocked in the next clock cycle of A5/1, new bits of MSBs of R_1, R_2 and output stream are XORed and a new value for MSB of R_3 will be obtained. In this situation, with probability $1/2$ a contradiction occurs, because R_3 has not been clocked in previous clock cycle and new and old MSB of R_3 may not be equal. These states are useless and they can be eliminated from the table.

The probability of contradiction after first clock cycle is $1/4\times 1/2=1/8$ (because during A5/1 clocking, R_3 is not clocked with probability $1/4$ and the probability that two random bits are not equal to each other is $1/2$). So the probability of contradiction after n th clock cycle is (7 / 8)^(n − 1) × 1 / 8 . Thus, we sum the probability of contradictions in the first n clock cycles, until the probability of contradiction in n consecutive clock cycles is obtained.

In a table which is prepared for the 10 unknown bits from R_3(5 MSBs and 5 CCBs) with using of 5 bits of the output stream and the 10 bits of R_1 and R_2 (5 MSBs and 5 CCBs from each), we sum the probability of contradictions in the first 5 clock cycles to obtain the amount of useless cases in the table. After this, the result is that, almost half of the cases in the table are contradictory, and they can be eliminated from the table.

Note that the length of the generated output stream depends on CCBs. By using 15 CCBs (5 bits from each register), there are 15968 cases that generated output stream is in the length of 5 bits and there are in 14280 cases the output stream is in the length of 6 bits and there are in the remaining 2520 cases the output stream is in the length of 7 bits [3].

In Table II the percentage of consistency for different lengths of output are presented for 15 CCBs. The amount of the all possible cases is obtained from all possible cases for MSBs of R_1, R_2 and CCBs of R_1, R_2, R_3 and output stream ((15968 + 14280 × 2 + 2520 × 4) × 2^15). In order to obtain the percentage of consistency, we calculated the number of consistent states for all the possible values for the CCBs of R_1, R_2, R_3 and MSBs of R_1, R_2 and output stream.

Notice that with increasing the length of the output stream, the percentage of consistency is decreased which is normal because the probability of contradiction is increased.

TABLE II. THE PERCENTAGE OF CONSISTENCY FOR DIFFERENT LENGTHS FOR 15 BITS OF CLOCK CONTROLLING

	All possible cases	**Output stream with length 5 bits**	**Output stream with length 6 bits**	**Output stream with length 7 bits**
Number	1789394944	15968	14280	2520
Percentage of consistency	50.1%	64.3%	47%	36.2%

The average number of candidates for the 20 bits of R_1, R_2 and 5 bits of the output stream is $2^{3.75} \approx 13.52$ ($1/2^{10}(15968 \times 0.64 + 14280/2 \times 0.47 + 2520/4 \times 0.362) = 13.52$). This amount in [3] is 23.2 (without elimination of useless states).

The result can be improved by using more bits in the table [3]. In Table III the percentage of consistency for different lengths of output are presented for 17 CCBs. Our recent table is based on 5 bits of the output stream, and 24 bits from R_1 and R_2 (6 MSBs and 6 CCBs from each), which contains in each entry the possible value of the 10 bits from R_3.

TABLE III. THE PERCENTAGE OF CONSISTENCY FOR DIFFERENT LENGTHS FOR 17 BITS OF CLOCK CONTROLLING

	All possible cases	**Output stream with length 5 bits**	**Output stream with length 6 bits**	**Output stream with length 7 bits**	**Output stream with length 8 bits**
Number	47244640256	23328	59808	41496	6440
Percentage of consistency	38.6%	100%	46.2%	29.6%	21.7%

The average number of candidates for each access to the table is 2^(3.3) ≈ 9.86 ((1 / 2^12) × (23328 + 59808 / 2 × 0.462 + 41496 / 4 × 0.296 + 6440 / 8 × 0.217) = 9.86). This amount in [3] is 16 (without elimination of useless states).

Another table must be used for the recovery 3 unknown bits of R_3. This table is prepared by 6 bits of each register. By using 9 CCBs (3 bits from each register), there are 392 cases that generated output stream is in the length of 3 bits and there are in 120 cases the output stream is in the length of 4 bits. This table gives us for each access 2^(2.43) = 5.4 candidates on average ((1 / 2^6) × (392 × 0.78 + 120 / 2 × 0.64) = 5.4). Thus we have on average 2^(−0.57) ≈ (0.67) candidates for the rest of the 3 unknown bits of R_3.

Time complexity of the attack is 2^20 × 2^12 × 2^0.3 × 2 × 2^3.3((1 + 1) + 2 × 0.67) = 2^38.34 A5/1 clockings.

Note that identifying of useless states and eliminating them, will not cause increase the time complexity of precomputation step; because during of generating the tables, we first suppose fixed bits for R_1, R_2 and output stream, then, for all options of CCBs of R_3, we obtain MSBs of R_3 and when we encounter contradictory states, we eliminate these states.

V. IMPROVEMENT OF ATTACK IN ONLINE PHASE

The time complexity is based on access to the tables without using of a memory in online phase of attack. Using memory in online phase means that after each access to the tables, all candidates that obtained from the table in a negligible memory must be saved. For example, by using a negligible memory and getting a candidate from third access (to the second table), if the candidate was wrong, there is no need to access the first table again. In this situation, the next candidate from the memory will be accessed (indeed all candidates obtained from the tables must saved in a memory). Thus the number of accesses to the table is decreased and the time complexity will be equaled to 2^(37.89) (2^20 × 2^12 × 2^0.3 × 2 × (1 + 2^3.3 (1 + 2 × 0.67)) = 2^37.89).

VI. CONCLUSION

In [3] an attack was presented on A5/1 that requires 2^(21.1) bits of known plaintext, 2^38 preprocessing of A5/1 clockings, 32 GB memory and 2^(39.91) time complexity of A5/1 clockings. We find out that almost half cases of the precomputed tables (that use in online phase of attack) are useless and can be eliminated from the table. The time complexity of [3] with elimination of the useless states decreases to 2^(38.34) and memory complexity decreases to 16 GB. This time complexity is decreased to 2^(37.89) using of negligible memory in the online phase of the attack.

If 2^21.1 known bits are not available in GSM, we can decrease the amount of the known bits by time-data tradeoffs. Thus we can propose two attacks based on [3] that these attacks require a few known plaintexts. In Table IV, A and B attacks are based on this assumption that R_3 is not clocked for 4 and 3 clock cycles respectively. These attacks are similar to the previous attacks and use the same tables.

TABLE IV. ATTACKS AND THEIR COMPLEXITY ON A5/1

	Precomputation complexity	**Time complexity**	**Data complexity (frame)**	**Memory Complexity (GB)**	**Success Rate**
A	2^38	2^44.19	4	16	55%
B	2^38	2^47.19	4	16	96%

REFERENCES

[1] E. Barkan and E. Biham, "Conditional Estimators: An Effective Attack on A5/1", proceedings of SAC' 05, LNCS 3897, pp. 1–19, Springer-Verlag, 2006.

[2] E. Barkan, E. Biham and N. Keller, "Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication", Journal of Cryptology, Volume 21, Number 3, pp 392-429, July 2008.

[3] E. Biham and O. Dunkelman, "Cryptanalysis of the A5/1 GSM Stream Cipher", presented by INDOCRYPT 2000 , LNCS 1977, pp. 43–51, Springer-Verlag, 2000.

[4] A. Biryukov, A. Shamir and D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC", Advances in Cryptology, proceedings of Fast Software Encryption'00, LNCS 1978, pp. 1–18, Springer-Verlag, 2001.

[5] M. Briceno, I. Goldberg and D. Wagner, ‘‘A pedagogical implementation of the GSM A5/1 and A5/2 ‘‘voice privacy’’ encryption algorithms’’, http://cryptome.org/gsm-a512.htm (originally on www.scard.org), 1999.

[6] P. Ekdahl and T. Johansson, ‘‘Another Attack on A5/1’’, IEEE Transactions on Information Theory, Volume 49, Issue 1, pp. 284-289, 2003.

[7] J. Golic, “Cryptanalysis of Three Mutually Clock-Controlled Stop/Go Shift Registers”, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 3, MAY 2000

[8] J. Keller and Birgit Seitz. ‘‘A Hardware-Based Attack on the A5/1 Stream Cipher’’, In Proceedings of the 2001 Arbeitsplatzcomputer (APC), Munchen, Germany, http://pv.fernuni-hagen.de/docs/apc2001-final.pdf, 2001.

[9] A. Maximov, Thomas Johansson and Steve Babbage, ‘‘An improved correlation attack on A5/1’’, proceedings of SAC’04, LNCS 3357, pp. 1–18, Springer-Verlag, 2005.

[10] T. Pornin and J. Stern, “Software-hardware Trade-offs: Application to A5/1 Cryptanalysis”, CHES 2000, LNCS 1965, pp. 318-327, Springer-Verlag, 2000.

[11] E. Zenner, “On the Efficiency of the Clock Control Guessing Attack”, ICISC 2002, LNCS 2587, pp. 200–212, Springer-Verlag, 2003.