The ISC Int'l Journal of Information Security

July 2012, Volume 4, Number 2 (pp. 107–114)

http://www.isecure-journal.org

Cryptanalysis of GSM Encryption Algorithm A5/1☆

Vahid Amin Ghafari [1,2,*], Ali Vardasbi [3], and Javad Mohajeri [3]

[1] *Research Center of Intelligent Signal Processing (RCISP), Tehran, Iran*

[2] *Department of Information and Communication Technology, Malek ashtar University of Technology, Tehran, Iran*

[3] *Electronics Research Institute, Sharif University of Technology, Tehran, Iran*

ARTICLE INFO.

Article history:

Received: 20 November 2011

Revised: 25 August 2012

Accepted: 9 December 2012

Published Online: 25 May 2013

Keywords:

A5/1, Precomputed Table, Useless States, Internal State Transition, Ultimately Periodic.

ABSTRACT

The A5/1 algorithm is one of the most famous stream cipher algorithms used for over-the-air communication privacy in GSM. The purpose of this paper is to analyze several weaknesses of A5/1, including an improvement to an attack and investigation of the A5/1 state transition. Biham and Dunkelman proposed an attack on A5/1 with a time and data complexity of $2^{39.91}$and $2^{21.1}$, respectively. In this paper, we propose a method for identification and elimination of useless states from the pre-computed tables and a new approach to access the table in the online phase of the attack which reduces the time complexity to $2^{37.89}$ and the required memory in half. Furthermore, we discuss another weakness of A5/1 by investigating its internal state transition and its keystream sequence period. Consequently, the internal states are divided into two classes, initially periodic and ultimately periodic. The presented model is verified using a variety of simulations which are consistent with the theoretical results.

1 Introduction

A5 is a family of encryption algorithms which 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 the use of A5/1, a stronger version of this family. Since 1999 when Briceno *et al.* published their paper on the accurate design of A5/1 via reverse engineering; many attacks have been proposed on A5/1 [5]. The attacks against A5/1 can be divided into two categories: active and passive. The passive attacks can be divided into three classes: guess-and-determine (GD), time-memory-data tradeoff (TMDTO) and correlation attacks. The GD attacks on A5/1 require high time complexity [7, 8, 10, 11], while the TMDTO attacks on A5/1 require high precomputation or data complexity [2, 4, 7], and correlation attacks on A5/1 require numerous known plaintexts or ciphertexts [1, 6, 9].

The first official cryptanalysis on A5/1 stream cipher was presented in 1997 by Golić, who proposed two known plaintext attacks on the cipher: guess-and-determine and time-memory tradeoff attack [12]. However, neither of these was practical.

Biryukov, Shamir and Wagner presented another time-memory tradeoff attack on A5/1 in 2000 [4]. Their cryptanalysis requires a large amount of known plaintext as well as a great deal of precomputation time; hence, it is not practical either. At the same time, Biham and Dunkelman proposed a guess-and-determine attack on A5/1. 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

☆ This article is an extended/revised version of an ISCISC'11 paper.

* Corresponding author.

Email addresses: vahidaming@yahoo.com (V. Ghafari), vardasbi@alum.sharif.edu (A. Vardasbi), mohajer@sharif.edu (J. Mohajeri).

ISSN: 2008-2045

and $2^{39.91}$ of A5/1 clockings [3]. This attack suffers from requiring too many known plaintexts as well.

Afterward, three cryptanalysis were proposed on A5/1 in the form of correlation attacks; however, they also required too many known plain or ciphertexts or they assumed ideal conditions for the attack [1, 6, 9], which made them impractical similar to the previous ones above. Barkan, Biham, and Keller proposed a time-memory-data tradeoff attack on A5/1 in 2007 [2]. Although this attack contains a high complexity precomputation phase, it is more practical than its predecessors. In this paper, the attack by Biham and Dunkelman is improved by identification and elimination of useless states from the precomputed table. Nearly half of the cases of the precomputed tables are useless, and therefore can be eliminated. By eliminating these useless states and proposing a method for exploiting the obtained table in the online phase of the attack, the time complexity reduces to $2^{37.89}$, while the memory complexity decreases to half of its previous value.

In the sequel, another weakness of A5/1 is presented which may be useful in some cryptanalysis methods such as time-memory tradeoff attacks. The problem of specifying the period of a stream cipher keystream sequence and its internal state is crucial in evaluating the security of that stream cipher. Although several references stated that the period of an A5/1 keystream sequence is approximately 2^{23} [7, 12, 13], there is no published paper regarding the exact structure of the A5/1 internal state transition. Studying the state transition of this algorithm is useful in a number of cryptanalysis methods such as rainbow attacks, in which knowing the quality and probability of a collision in the produced table can be of great importance [2]. Another example is the time-memory tradeoff attack in [4], where the states which can be clocked backwards are a minimum of 100 clocks and the methods which cover more states after 328 back-clocking are preferred. In fact, if there was a feasible method to distinguish the states which can be back clocked to more than 100 previous states, from the states lacking this property, then, choosing the points in the tables could be performed more efficiently. This would lead to more reasonable results in time-memory tradeoff attacks [4].

The present paper is organized as follows: Section 2 contains a description of the A5/1 algorithm. An improved attack on A5/1 is presented in Section 3, while Section 4 deals with the state transition in the A5/1 algorithm. Finally, we will summarize and conclude the paper in Section 5.

2 Description of the A5/1 Stream Cipher

The A5/1 consists of three LFSRs (Linear Feedback Shift Registers) with the lengths of 19, 22, 23, denoted by R_1, R_2, and R_3, respectively. The output is generated by XOR-ing of the most significant bits (MSBs) of the three registers. Subsequently, the values of three bits $R_1[8]$, $R_2[10]$, and $R_3[10]$ (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. It is noteworthy that at least two registers are clocked at each clock cycle, and each register will be clocked with the probability of $\frac{3}{4}$. The numerical value of zero is allocated to the least significant bit of each register as indicated in Figure 1 . The 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 are 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 the 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; moreover, the second 114 bits are used for encryption data from the mobile phone to the network. Note that we suppose, an attacker can access a single direction in each loading of A5/1. Thus each 114 bits is relevant to one loading.

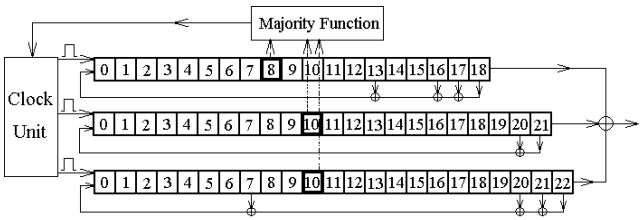

Figure 1. The A5/1 internal structure.

3 An Improved Attack on A5/1

Biham and Dunkelman proposed a guess-and-determine attack on the A5/1 in which they exploited a precomputed table to improve the performance

of their attack. First, they proposed a guess-and-determine attack on the A5/1, claiming that the complexity of the attack is similar to the earlier attack by Golić [7, 12]. In the second step, they improved the attack by exploiting a precomputed table. This attack, which can recover the internal state of the algorithm, requires $2^{21.1}$ bits of known plaintext, $2^{33.6}$ preprocessing of the A5/1 clockings, 2 GB memory and has a time complexity of $2^{40.97}$ A5/1 clocking. Subsequently, the attack is improved by utilizing even a larger table [3]. Their final attack requires $2^{21.1}$ bits of known plaintext, 2^{38} preprocessing of the A5/1 clockings, 32 GB memory and has a time complexity of $2^{39.91}$ A5/1 clockings.

In the following sections, this attack is improved by identifying and eliminating nearly half of the states from the precomputed table, which are revealed to be useless. By eliminating these useless states and proposing a method to exploit the resulting table during the online phase of the attack, the time complexity reduces to $2^{37.89}$, and the memory complexity decreases to 16 GB.

The effect of eliminating the useless states from precomputed tables is displayed in Table 1. The results in Table 1 are based on the assumption that each frame (of length 114 bits) is related to one loading.

Table 1. Attack on A5/1 presented in [3], and the improved proposed attack.

Attack	Precomp. Compl.	Time Compl.	Data Compl. (known bits)	Memory Compl.	Success Rate
Biham & Dunkelman	2^{38}	$2^{39.91}$	$2^{21.1}$	32 GB	63%
Proposed attack	2^{38}	$2^{37.89}$	$2^{21.1}$	$\approx 16GB$	63%

3.1 Early Attack on A5/1

The main idea of the attack was to wait until an event occurs which leaks a large amount of information about the internal state. 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 the attack, $R_1[9, 10, 11, 12, 14, 15, 16, 17, 18]$, $R_2[0]$ and $R_3[10, 22]$ are guessed; then all bits of R_1 and R_2 are recovered.

In the second step, the attacker refers to the precomputed table and recovers the remaining bits of R_3. This table contains the possible values for the 10 bits from R_3(5 MSBs and 5 CCBs), 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). The average number of the 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 the third register is not clocked in one clock cycle is 1/4 and the probability that the third register is not clocked for 10 consecutive clock cycles is equal to 2^{-20}). Each 64 known bits of the successive output stream is sufficient for recovering the internal state of the algorithm [7]. Each 114 output stream bits contains $114 - 64 + 1 = 51$ (overlapping) strings of 64 consecutive bits. Therefore, by using $114.(2^{20})/51 = 2^{21.1}$ known bits, one can apply the attack with a success rate of $1 - (1 - 2^{-20})^{2^{20}} = 63\%$.

For each 2^{20} string, there are 2^{12} possible cases to be guessed (in the first step of the attack) and for each case, in the first access to the table, there are $2^{1.53}$ candidates on average (note that the first time that the attacker accesses the table, three bits of R_3 are known). The attacker clocks the registers for a required number of times according to the table access. The cost of the first clock is equivalent to two A5/1 clockings. In the second access to the table, the attacker obtains an average of $2^{4.53}$ candidates and clocks the registers as many times as required [3].

For the other 3 bits of R_3, the attacker builds another table width 6 bits of each register (since another access to the previous table will cost a great deal). Through each access, this table provides the attacker an average of $2^{2.82}$ candidates. Thus an average of $2^{0.18}$ candidates remain for the three unknown bits of R_3.

Next, the wrong candidates must be discarded. To check whether this is a right state, two clock cycles for each candidate are required. Therefore, the time complexity of the attack is $2^{40.97}$ A5/1 clockings which is obtained from the following:

$$2^{20} \times 2^{12} \times 2^{1.53} \times 2 \times 2^{4.53} \times ((1+1)+2 \times 0.88) = 2^{40.97} \quad (1)$$

where $(1 + 1)$ is due to the work which is performed before accessing the second table [3].

The authors indicated that this attack can be improved by using a large table. The time complexity of the attack decreases 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].

3.2 Contradictory States

There is an important point which has not been mentioned in [3]. In the first access to the table, the MSB of R_3 can be obtained using MSBs of R_1, and R_2 as well as the output bit which are all known. During

the next clock cycle of A5/1, the new MSBs of R_1, R_2 and the output bit are XORed, thus a new value for MSB of R_3 will be obtained. In the case where R_3 is not clocked in this clock cycle, a contradiction will occur with a probability of 12, since the newly obtained MSB of R_3 may not be equal to the old one. The states in which the above event occurs are useless and can be eliminated from the table.

The probability of contradiction after the first clock cycle is $1/4 \times 1/2 = 1/8$ (because during the A5/1 clocking, R_3 is not clocked with a probability of $1/4$ and the probability which two random bits are not equal is $1/2$). As a result, the probability of a contradiction after the n^{th} clock cycle is $(7/8)^{n-1} \times 1/8$. Thus, the probability of contradiction in n consecutive clock cycles is obtained by summing the probability of contradictions in the first n clock cycles.

In a table which is prepared for the 10 unknown bits from R_3 (5 MSBs and 5 CCBs) by using 5 bits of the output stream and 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. Finally, nearly 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), the length of the generated output stream can be 5, 6 and 7 bits in 15968, 14280 and 2520 of the cases, respectively [3].

In Table 2, the percentage of consistency for different lengths of output is presented according to 15 CCBs. 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 bit.To distinguish the consistent states and the contradictory states, one should evaluate the MSB of R3 according to the given bits of R1, R2 and the keystream. The contradictory states are the ones whose R3 is not to be clocked and it has a MSB value different from this evaluated value. Note that the percentages of consistencies are given according to the length of output bits which depend on the control bits.

Table 2. The percentage of consistency for different lengths for 15 bits of clock controlling

Length of output stream	# possible values for the CCBs	Consistency (%)
5 bits	15968	64.3%
6 bits	14280	47%
7 bits	2520	36.2%

It should be noted that by increasing the length of the output stream, the percentage of consistency is decreased which is normal since the probability of contradiction is increased.

The average number of candidates for the 20 bits of R_1, R_2 and 5 bits of the output stream is $2^{3.75}$:

$$2^{-10} \times (15968 \times 0.64 + \frac{14280}{2} \times 0.47 + \frac{2520}{4} \times 0.362) = 13.52 \quad (2)$$

However in [3], without eliminating useless states, the corresponding average is 23.2.

The result can be improved by using more bits in the table [3]. The percentage of consistency for different lengths of output according to 17 CCBs is presented in Table 3. This 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 the possible value of the 10 bits from R_3 in each entry.

Table 3. The percentage of consistency for different lengths for 17 bits of clock controlling.

Length of output stream	# possible values for the CCBs	Consistency (%)
5 bits	23328	100%
6 bits	59808	46.2%
7 bits	41496	29.6%
8 bits	6440	21.7%

The average number of candidates for each access to the table is $2^{3.3} \approx 9.86$:

$$2^{-12} \times (23328 + \frac{59808}{2} \times 0.462 + \frac{41496}{4} \times 0.296 + \frac{6440}{8} \times 0.217) = 9.86 \quad (3)$$

In [3] where useless states are not eliminated, the corresponding average is 16.

Another table is required to recover the 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), the length of the generated output stream is 3 bits in 392 cases and 4 bits in the remaining 120 cases. This table provides us an average of $2^{2.43}$ candidates through each access:

$$2^{-6} \times (392 \times 0.78 + 120/2 \times 0.64) = 5.4 \quad (4)$$

Thus,an average of $2^{-0.57}$ candidates remain for three unknown bits of R_3.

The time complexity of the attack is obtained by:

$$2^{20} \times 2^{12} \times 2^{0.3} \times 2.2^{3.3} \times ((1+1) + 2 \times 0.67) \quad (5)$$

which is $2^{38.34}$ A5/1 clockings.

It is noteworthy that identifying and eliminating the useless states will not increase the time complexity of precomputation phase; because during the table's generation, we primarily assume the fixed values for R_1, R_2 and output bits and then we obtain MSBs of R_3, for all possible choices for CCBs of R_3. During this process a contradictory state is eliminated whenever encountered. Therefore, no extra time is imposed by the elimination.

3.3 The improved Attack in Online Phase

The time complexity is based on access to the tables without using any memory in the online phase of the attack. Using memory in the online phase means that after accessing the tables, all candidates obtained from the table must be saved in a negligible memory. For example, by using a negligible memory and getting a candidate from the 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 (In fact all candidates obtained from the tables must be saved in a memory). Thus the number of accesses to the table is decreased and the time complexity will equal to $2^{37.89}$, which is obtained from the following:

$$2^{20} \times 2^{12} \times 2^{0.3} \times 2 \times (1 + 2^{3.3}(1 + 2 \times 0.67)) \quad (6)$$

Hitherto, two improvements have been made on the attack of [3]. All the previous attacks, including the original attack and the improved version, assumed that R_3 was not clocked for ten clock cycles. Due to this assumption, the attack requires $2^{21.1}$ bits of known plaintext which may not be available in the GSM. If it is assumed that R3 is not clocked for less than ten clock cycles, then the data complexity can be decreased by increasing the time complexity, i.e. exploiting a time-data tradeoff. Therefore, we can propose two modifications to previous attacks which require less known plaintext bits.

In Table 4, attacks A and B are based on the 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 4. Attacks on A5/1 and their complexity.

Type of Attack	Precomp. Compl.	Time Compl.	Data Compl. (frame)	Memory Compl. (GB)	Success Rate
A	2^{38}	$2^{44.19}$	4	16	55 %
B	2^{38}	$2^{47.19}$	4	16	96 %

4 Period of GSM's A5/1 Stream Cipher

After proposing an improved attack on the A5/1 algorithm, this section aims to find a structure for the state transition in the algorithm.

If the A5/1 registers were not clocked with respect to a majority function, i.e., all three LFSRs were clocked in all the algorithm clocks, due to the LFSRs' primitive characteristic function and their relatively prime size, the period of the algorithm's generated keystream will be $\approx 2^{64}$. However, the majority function makes it hard to comment about the period of the keystream sequence. In [13]., the period of an algorithm "like A5/1" was observed to be near $\frac{4}{3}(2^{23} - 1)$. This observation was later referenced by Golić in [7, 12] for the A5/1 algorithm.

Our investigation, however, revealed that with a high probability, a randomly selected initial state will never be repeated; suggesting the keystream sequence is ultimately periodic. We tested a set of 10,000 randomly selected initial states and the first 64 keystream bits were repeated in none of them.

4.1 Internal State Transition

After numerous experiments and simulations on the internal state transition of the A5/1 algorithm, the states were observed to enter a loop after an average number of $2^{26.17}$ algorithm clocks. These simulations indicated that a large proportion of all the internal states will never be repeated. In other words, the loop states (i.e., the states which are repeated during the run of the algorithm) constitute only a small part of the internal states.

Therefore, the space of the algorithm internal states can be divided into several independent pages. Each page containing one loop, in which there are several branches entering the loop. A scheme of the internal states in a page is depicted in Figure 2.

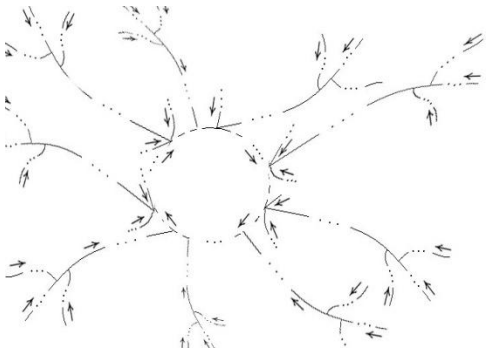

Figure 2. A scheme of the internal states in a page

All the states in each page will ultimately enter into a loop. Throughout this paper, the number of

clocks after which a state meets the loop is called the distance of that state to its loop.

In one of our simulations, 100 initial states were selected randomly and the distance of each state to a loop as well as the period of the loop was measured. The average distance of a randomly selected state to its loop was $2^{26.17}$, while the average period of each loop was $2^{25.42}$. The minimum and maximum of the measured values are presented in Table 5.

Table 5. The minimum and maximum of the distance of each state to a loop and the period of the loop.

Sample with max/min period/distance of/to the loop	Distance to loop	Loop period
maximum period	$2^{26.19}$	$2^{28.41}$
minimum period	$2^{27.32}$	$2^{23.41}$
maximum distance	$2^{28.08}$	$2^{23.41}$
minimum distance	$2^{17.27}$	$2^{24.41}$

Table 6 contains the ratio of the number of states with one, two, three and four possible predecessors to the total states in a loop. It is interesting to note that despite the different period of loops, this ratio remains approximately constant for all the loops. This occurs because the states in a loop can be considered to have a uniform distribution in this context [1].

Table 6. The ratio of the states with one, two, three and four possible predecessors to the total states in a loop.

One state	Two states	Three states	Four states
0.65	0.125	0.156	0.062

Our results are consistent with the following proposition about the number of possible predecessors [7, 12].

Proposition 1: If an internal state $S(t)$ is randomly chosen according to a uniform distribution, then the number of solutions for the predecessor $S(t-1)$ is a nonnegative integer random variable Z with the probability distribution [7]:

$$\Pr\{Z=0\}=\frac{3}{8},\Pr\{Z=1\}=\frac{13}{32}, \qquad (7)$$
$$\Pr\{Z=2\}=\Pr\{Z=3\}=\frac{3}{32},\Pr\{Z=2\}=2^{-5}$$

Table 7 shows the theoretical ratio of the number of states with one, two, three and four possible predecessors to the 62.5% states which have at least one possible state. Evidently, the states in a loop have at least one predecessor and constitute a portion of these 62.5% states. Having a distribution sufficiently close to uniform, the states in a loop have an experimental ratio (Table 6) which is consistent with the theoretical ratios in Table 7.

Table 7. The theoretical ratio of the states with one, two, three and four possible predecessors to the states with at least one predecessor.

One State	Two States	Three states	Four states
0.65	0.15	0.15	0.05

It is notable that 37.5% of all the states have no possible predecessors. In other words, 37.5% of all the states can only be a starting point and they are never produced. This was noted in [4] and the paper was based on the elimination of the states with no possible states after 100 backward clocking and exploiting the states which cover more possible states when they are clocked back. Evidently, the states in a loop are more suitable to be used in time-memory-data tradeoff tables, due to their more comprehensive coverage of the states when they are clocked back.

Several statistical tests such as the ordinary test, NIST tests and the overall test were applied on the states both in and out of the loop in order to find a method to distinguish the loop states from the other states. Unfortunately, these tests had no significance in distinguishing among the states in the loops and the other states.

Proposition 1 only deals with one step back-clocking. Following a similar approach as the one in [4, 12], one can propose the distribution of the number of possible predecessors for more than only one step of back-clocking. In other words, one can consider all the $2^{3(d+1)}$ different values for $d+1$ bits of each LFSR in A5/1 which is shown in Figure 3 as the shaded area.

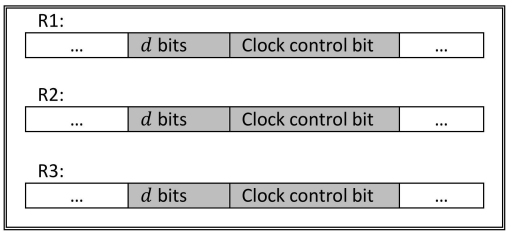

Figure 3. Changing bits in three LFSRs to obtain the number of possible predecessors in d clocks back.

Thus, to further study the behavior of the A5/1 algorithm state transition, we calculated the percentage of states with different numbers of possible predecessors after d ($d=1,\ldots,6$) back clockings. This process is performed as follows: Each one of the possible $2^{3(d+1)}$ states are back-clocked for d clocks and the number of possible predecessors are monitored. At the end, the number of all the states with k ($k=0,\ldots,4^d$) possible predecessors is divided by $2^{3(d+1)}$ which is the

[1] The bias which consecutive A5/1 states have from a uniform distribution is well below these ratios and therefore can be neglected.

total number of states in this process. Table 8 shows the percentage of states with k ($k = 0, 1$) predecessors for d ($d = 1, \ldots, 6$) back clocking. The last column of Table 8 indicates the total number of states with less than four possible predecessors.

According to Table 8, the sum of ratios for $k < 2$ remains near the constant value of 80% for different 'd's. Furthermore, approximately 94% of all states have less than four possible predecessors for different numbers of clocks back.

Table 8. The theoretical percentage of states with k ($k = 0, 1, 2$) predecessors for d ($d = 1, \ldots, 6$) Clocks Back

# back clocking \ # predecessors	$k = 0$	$k = 1$	$k < 4$
$d = 1$	37.5	40.6	96.9
$d = 2$	42.2	37.9	95.3
$d = 3$	43.9	35.8	94.9
$d = 4$	45.3	34.2	94.7
$d = 5$	46.6	32.8	94.4
$d = 6$	47.9	31.5	94.1

Our simulations lead us to the conclusion that the average number of each page states is approximately $2^{51.6}$ and consequently, there are about $2^{12.4}$ pages. Next, we give a brief overview of the number of pages.

The first step is to determine the number of each of the loop branches. The states in a loop can be categorized into four groups, based on the number of their possible predecessors. Each state in a group g ($g = 1, \ldots, 4$), introduces $g - 1$ branches to the loop. Therefore, one can estimate the number of branches in a loop as follows:

$$B = L \cdot \sum_{g=1}^{4} \{Pr(Z = g) \cdot (g - 1)\}, \tag{8}$$

where B and L represent the average number of branches in one loop and the average number of states in each loop, respectively.

Using the values in Table 7, and $L \approx 2^{25.42}$ the average number of branches in a loop will be:

$$B \approx 2^{24.68} \tag{9}$$

Now, only the average number of states in each branch (which we denote it n) needs to be obtained. Based on our simulations, the average distance of a state to its loop is $2^{26.17}$; we call this quantity D. In order to estimate the number of states in each branch (n), it should be noticed that approximately 40% of the states have no possible predecessors; thus they are on the starting points of the branches. Assuming this is the case for the points in each branch (i.e. around 40% of all the points of each branch are starting points), the distance of each point to the loop will be through the remaining 60% of the branch points. This observation suggests that D constitutes about 60% of n, leading to the following conclusion:

$$n \approx \frac{D}{0.6} \approx 2^{26.91} \tag{10}$$

Finally, each page contains an average of N states, where N can be estimated by:

$$N = n.B + L \approx 2^{51.6} \tag{11}$$

and there are approximately $\frac{2^{64}}{2^{51.6}} = 2^{12.4}$ different pages. It has to be noted that this approach provided us a rough estimation and more research is required in order to improve these estimations.

4.2 Theoretical View of the Result

To explain this situation for the internal states of the A5/1 algorithm, two points should be noted. First, there are a finite number of internal states and this means, at some point, the internal state will be repeated, i.e., the internal states sequence is ultimately periodic. Second, 37.5% of the states have no possible predecessors, meaning that these states can only be the starting points. They will never be generated in the clocking process (an example of this status is shown in Figure 4). Therefore, the scheme in Figure 1 is theoretically explainable as well. This behavior may also be observed in nonlinear shift registers.

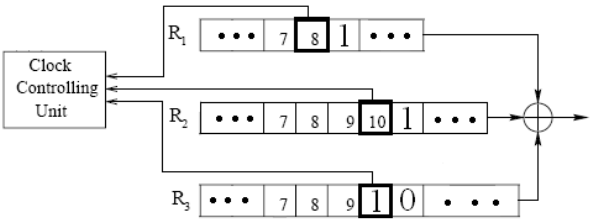

Figure 4. An example of the states have no possible predecessors.

5 Conclusion

The Biham and Dunkelman's attack on the A5/1 stream cipher requires $2^{21.1}$ bits of known plaintexts, 2^{38} preprocessing of the A5/1 clockings, 32 GB of memory and $2^{39.91}$ time complexity of the A5/1 clockings. We have discovered that nearly half of the cases of the precomputed tables (that are used in the online phase of the attack) are useless and can be eliminated from the table. Hence, the time complexity of the proposed attack decreases to $2^{38.34}$ and that of the memory decreases to 16 GB. Moreover, the time complexity is reduced to $2^{37.89}$ using negligible memory in the online phase of the attack. In addition, the state transition of the A5/1 algorithm was investigated. One of the important applications of this discussion can be

in time-memory-data tradeoff attacks (especially rainbow attacks), where the states with possible previous 100 states and the ones which cover more states after 328 backwards clocking are preferred.

Moreover, some simulations were performed along with the investigation of the internal state transition of A5/1. A set of 10,000 randomly selected samples was tested in one of them. It was observed that none of these samples were repeated in the clocking process. An extensive range of experiments has led us to the conclusion that after an average of approximately $2^{26.17}$ clocks, the internal states of the algorithm will enter several loops and only the loop states are repeated during the clocking process. Based on this observation, the structure of the A5/1 internal states was conjectured in Figure 1.

Finally, several estimates of the total number of pages and each page points were presented.

Acknowledgment

The authors would like to thank Narges Abdi for her precious effort in some of simulations.

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. Golić, "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.

[12] J. Golić, "Cryptanalysis of Alleged A5 Stream Cipher," *Advances in Cryptology, proceedings of Eurocrypt'97, LNCS 1233*, pp. 239–255, Springer-Verlag, 1997.

[13] W. G. Chambers, "On random mappings and random permutations," *in Fast Software Encryption '94 (Lecture Notes in Computer Science)*, Berlin, Germany: Springer-Verlag, 1995, vol. 1008, pp. 22–28.

Vahid Amin Ghafari received the B.S. and M.S. degrees in Electrical Engineering from Imam Hossein University (IHU) in 2005 and 2007, respectively. Since 2006, he has been a research assistant at Research Center of Intelligent Signal Processing (RCISP). His research interests include cryptography, design and analysis of algorithms, and data security.

Ali Vardasbi received his B.S. and M.S. degrees both from Sharif University of Technology in 2008 and 2011, respectively. Since then, he has been a research assistant at the Electronics Research Institute of Sharif University of Technology. His research interests include cryptography, design and analysis of algorithms, and signal processing.

Javad Mohajeri received the B.S. degree from Isfahan University in 1986 and the M.S. degree from Sharif University of Technology in 1989, both in mathematics. He has been a faculty member at Electronics Research Institute of Sharif University of Technology since 1990. His research interests include cryptography and data security. He is the author/co-author of over 60 research articles in refereed Journals/ Conferences.