State Transition Analysis of GSM Encryption Algorithm A5/1

Praveen Kumar Gundaram, Appala Naidu Tentu, and Swamy Naidu Allu

Abstract—A5/1 stream cipher is used in Global System for Mobile Communication(GSM) phones for secure communication. A5/1 encrypts the message transferred from a mobile user. In this paper, we present the implementation of cryptanalytic on A5/1 techniques such as minimized state recovery for recovering the session key. The number of state transitions/updations needed for a state S to reoccur is maintained in the lookup table. This table can be used to recover the initial state from which the keystream was produced. Experiments are carried out for reduced version, full A5/1 cipher on 3.20 GHz machine, and cluster computing facility.

Index Terms—A5/1 stream cipher, Cryptanalysis, Precomputed Tables, Keystream, Initial State Transition, Periodicity.

I. INTRODUCTION

MORE than 5 a billion users of GSM mobile phones use A5/1 Stream Cipher [19] to protect confidentiality communication. In GSM, the data is transmitted as 228-bit block frames. Over the air, every 4.615-millisecond frame is sent and received.

GSM [7] is composed of three main algorithms [5], the A3 algorithm used for authentication, the A5 algorithm used for encryption, and the A6 algorithm for key generation [3]. Many of these algorithms are comparatively weak and have therefore been successfully targeted in the past years. The internal architecture of the two algorithms (i.e, A3 and A6) in GSM is not described. The operators may additionally adopt the exact configuration of the stream cipher algorithms [2] on their personal. In 1994 the approximate design of A5/1 was disclosed [20]. In 1999 the complete design of both stream ciphers A5/1 and A5/2 was discovered by Briceno[4].

A5/1 Stream Cipher [1] produces a 228-bit keystream denoted as PRAND using a 64-bit session key denoted as $K_c$ and 22-bit frame counter also known as IV denoted as $F_i$. A ciphertext of length 228-bits is produced after XORing the 228-bit plaintext with the 228-bit keystream.

Several cryptanalytic techniques were proposed on the A5/1 cipher include Anderson [2], Golic [19] and Babbage [11]. In 2001 Biryukov, Shamir, and Wagner [12], in 2000 Biham and Dunkelman [10], in 2003 Ekdahl and Johansson [9], in 2005 Maximov, Johansson and Babbage [11], in 2008 Barkan and Biham Keller [16] and a few other researchers examined A5/1 after reverse-engineered. For more detail about cryptanalytic techniques go through our previous paper [24]. For understanding the behavior of the A5/1 stream cipher, we analyze the present state of the cipher [13]. The obvious presumption is that the state space is $2^{24}$. However, a closer study of the clocking mechanism reveals that a significant proportion of the potential internal states are inaccessible from any valid state [8]. So many experiments have been conducted to evaluate the failure of probable states in the stream cipher A5/1, all of these experiments conclude that only about 15% of all desirable states remain applicable after the beginning 100 clockings. In practice, any attacker [17] wants to cover 15% of the state capacity: $N \approx 2^{61.26}$.

A. Our Contribution

In this paper, we constructed a minimized lookup table for recording the periodicity of each state used to design and implement a reduced version of A5/1 stream cipher. The summary of our contribution is given below:

- We proposed the Floyd cycle-detection algorithm and its implementation, which is used for an attack on the A5/1 stream cipher.
- Procedure for recovering the session key from the initial state.
- Implemented an attack using the constructed table on the reduced version of the A5/1 stream cipher, which recovers the initial state of the stream cipher given the keystream.

B. Organization of the Paper as follows

Section II describes the design of the A5/1 stream cipher. Section III describes the reduced version A5/1 stream cipher. In section IV, we explained the proposed attack i.e., minimized internal state recovery attack with experiment results, and section V concludes the work.

II. A5/1 STREAM CIPHER

In a digital mobile network, over-the-air (OTA) transmissions are encrypted with a stream cipher to ensure their security. The A5/1 stream cipher [4] design by using three linear feedback shift registers (LFSRs), table - I as feedback polynomials and figure-1 demonstrates the specifications of three shift registers. Each register has one clocking bit associated with it. A majority function is used to clock all three registers, stop and go fashion.
A. Procedure

The A5/1 stream cipher [24] is formed by three Linear Feedback Shift Registers (LFSRs) that use majority clocking. These LFSRs have a total bit count of 19 + 22 + 23 = 64. When the LFSR is moved, a few tapped bits of the LFSR’s are XORed together to supply the later bit as shown in the table - I. Majority clocking ensures that only LFSRs are clocked when the majority value of three clock bits is the same as the clock bit. For randomly distributed clocking bits, the probability of a register being shifted is 75%. There are only 8 possible conditions out of one, three registers to generate a keystream. The condition is XORed out upon initialize. The 64-bit secret key $K_c$ (known to handset and BTS) is loaded by XORing the bits to the rightmost bit of each LFSR before shifting them. Irregular clocking rule says that, at every cycle, the given register is clocked if its clocking bit is equal to the majority of all 3 clocking bits. At each step at least 2 or 3 registers are clocked as shown in the below majority function equation.

$$f(x, y, z) = x \cdot y \oplus y \cdot z \oplus z \cdot x$$

where $x$, $y$, and $z$ are the clocking bits of the registers.

The above function takes three register’s clocking bits as input and produces the majority bit as output.

| LFSR | Length in bits | Feedback Polynomial | Control bit | Tapped positions |
|------|----------------|---------------------|-------------|------------------|
| 1    | 19             | $x^{19} + x^{18} + x^{17} + x^{14} + 1$ | 8           | 13, 16, 17, 18  |
| 2    | 22             | $x^{22} + x^{21} + 1$ | 10          | 20, 21          |
| 3    | 23             | $x^{23} + x^{22} + x^{21} + x^{13} + 1$ | 10          | 7, 20, 21, 22   |

Table I: Parameters of A5/1 Stream Cipher

During these processes, all LFSRs are clocked, so majority clocking is allowed only after 64+22 clockings [24]. After that, the machine is clocked forward 100 times using the majority rule, storing the output of 114 bits which is a keystream as shown in figure - 2.

III. REDUCED VERSION OF A5/1 STREAM CIPHER

A. Description

In this section, we discuss about design new reduced version of A5/1 stream cipher and their cryptanalysis which is used for help to recover the key of A5/1 [18]. We named the new designed cipher is Tiny A5/1 stream cipher which follows parameters of A5/1 [22]. Tiny A5/1 (16-bits) is a reduced version of A5/1 for understanding the behaviour of full A5/1 stream cipher (64-bit) [21]. It uses 3 LFSRs $R_1$, $R_2$, and $R_3$ are of the lengths 4, 5 and 7 respectively, similar to A5/1 as shown in the table - II. The feedback polynomials for the 3 LFSRs are given by $x^4+x+1$, $x^4+x^3+x^2+x+1$ and $x^7+x^3+x^2+x+1$ for $R_1$, $R_2$, and $R_3$ respectively, these polynomials decide the tapping positions so the tapping positions of LFSR$R_1$ are 3, 0; LFSR$R_2$ are 4,3,1,0; LFSR$R_3$ are 6,2,1,0 as show in the figure - 3.

Table II: Tiny A5/1 Stream Cipher Parameters

| LFSR | Length in bits | Feedback Polynomial | Clocking bit | Tapped bits |
|------|----------------|---------------------|--------------|-------------|
| $R_1$| 4              | $x^4+x+1$           | 2            | 3.0         |
| $R_2$| 5              | $x^4+x^3+x^2+x+1$   | 2            | 4,3,1,0     |
| $R_3$| 7              | $x^7+x^3+x^2+x+1$   | 3            | 6,2,1,0     |

Keystream generation [15] procedure as follows, initially all 3 registers load with zeros. Then fill the session key ($K_c$) and frame counter ($F_n$)(22 bits) into three registers bit by bit [24]. Then clock the registers 100 times irregularly with the majority rule. Now we get a state called the initial state. From the current state, we generate a keystream of 114+114 bits by irregular clocking mechanism, then clock as same for the later frame. The same procedure is followed for the next 22 bit Frame Number, which varies with each burst but is publicly known.
All the parameters of Tiny A5/1 are shown in table - II and the length of the LFSRs used in Tiny A5/1 are co-prime to each other. So the period of Tiny A5/1 is \((2^4 - 1) \times (2^5 - 1) \times (2^7 - 1) = 15 \times 31 \times 127 = 59055 < 2^{16} - 1\). The clocking positions of registers \(R_1, R_2\) and \(R_3\) are 2, 2, and 3. A register is clocked (updated) if and only if the majority of all the three clocking bits are equal to its clocking bit. So at least two registers are clocked at each iteration (clock).

### B. Cryptanalysis of Tiny A5/1

Stepwise procedure to estimate the period of each initial state as follows.

1) For each initial state in \(2^{16}\) states: load the initial values of 16 bit in Tiny A5/1.
2) Check whether the state is connecting to loop. If so calculate the distance between the state and the loop. Then calculate the periodicity using floyd cycle detection algorithm is shown in figure - 6.

### C. Internal State Transition

Among all possible \(2^{16}\) states, consider only states in which state of each register should have at least one (that is non-zero state) [23]. So that all possible non-zero states such that each register will have a non-zero state are 59055 \((2^4 - 1) \times (2^5 - 1) \times (2^7 - 1) = 15 \times 31 \times 127 = 59055\). This will be the theoretical period of the keystream generated by Tiny A5/1. But experimentally, if we consider an initial state, after certain clocks it moves towards a loop. We did this experiment by taking each state from all 59055 states. We could get two loops such that each state, after a certain number of iteration, will move towards any one of the loops. The period of the two loops is 353 and 860. We found only two loops in the entire keyspace, those loops periods are 353 and 860. We observed after simulation of Tiny A5/1 algorithm’s internal states periodicity, that all states are eventual periodicity, in all the states on an average after 270 clocks it will fall into the loop out of two loops. In the below table - IV, minimum, maximum clock values are shown.

The following table - III shows that some initial states with periodicity in terms of distance to loop and period of loop.

### D. Linear Complexity

Linear complexity of a sequence(S) is denoted as \(LC(S)\), which is defined as the length of the shortest LFSR which generates the given sequence \(S\), where \(S = z_0, z_1, \ldots\) be a finite or infinite sequence. In this section we calculate the linear complexity of all the states of Tiny A5/1 ciphers (all states are fall into 353 and 860 loops). Then measure the linear complexity of all lines for bot loops, and observe the maximum, minimum, and average values in table- V.

#### Linear Complexity of All States

Linear complexity of all lines for bot cycles graph shown in figure - 4 below.

![Linear complexity of all states](image.png)

X-axis represents the number of states of Tiny A5/1 stream cipher and Y-axis represents linear complexity of the state.

### IV. Minimized Pre-Computation Table Attack

The Minimised Pre-computation Table Attack (MPTA) proposes an enhanced existing attack on the A5/1 algorithm, with the goal of working out how to transform the algorithm’s state. The time of the algorithms generated keystream would be roughly \(2^{64}\), if the A5/1 registers were not clocked according to a majority rule, i.e. all three LFSRs were clocked in all algorithm clocks, due to the LFSR’s primitive characteristic function and their comparatively prime size. Our analysis found that a randomly selected initial state will almost definitely never be repeated and has no predecessors. However, the majority feature makes it hard to comment on the keystream sequence’s period.

In the period of an algorithm “like A5/1” was observed to be near \(4/3\) \((2^{23} - 1)\). suggesting the keystream sequence is ultimately periodic. We tested a set of \(2^{25}\) randomly selected
initial states and the first 64 keystream bits were repeated in none of them.

A. Internal State Transition

In the experimental results, we can observe that there are a finite number of internal states [25]. All internal state sequences eventually are periodic, and all these 37.5 percents of the states have no possible predecessors, these can be used as an initial state [1]. We perform various experiments and simulate various internal states. We observed that the A5/1 algorithm behaves an average of $2^{26.17}$ algorithm clocks required to calculate the period, as seen in the table. According to these simulation results, observed that an important proportion of all internal states will never be repeated. In another way, the states that are repeated during the algorithm’s execution makeup just a limited portion of the internal states. As a consequence, the internal state space of the algorithm can be separated into many separate state loops. Each state includes a single loop through which multiple branches join. Each state on every circle will conclusively arrive into a loop. Distance of that state to its loop [26] is defined as several clocks after which a state meets the loop as shown in the table - VI.

### Table VI

| Distance of initial state to a loop and its period | Avg. distance of state to a loop | 62390635.86 $\approx 2^{25.65}$ |
|---|---|---|
| Avg. period of the loop | 43577701.376979 $\approx 2^{23.29}$ |
| Min. distance of state to a loop | 810 $\approx 2^{7.08}$ |
| Min. period of the loop | 11182509 $\approx 2^{18.29}$ |
| Max. distance of state to a loop | 845572755 $\approx 2^{32.57}$ |
| Max. period of the loop | 167773089 $\approx 2^{34.25}$ |

B. Stepwise Procedure of Internal State Transition

1) Phase-I: Pre-computation phase.
   - Generate off-line data for cryptanalysis.
   - Approach: load random state, then detect cycle with corresponding length from loaded state.
   - Stepwise procedure:
     - Load 64-bit random numbers into $R_1$, $R_2$ and $R_3$ registers.
     - Clock $n$ times based on majority rule to find the period of the state using floyd’s-cycle-detection-of-state (where $n$ will get from the floyd algorithm).
     - Find intersection state of cycle.
     - Find the lengths (in terms of clocks) of loop and line.
     - Store random state, loop intersection state, length of loop and line in the table.

2) Phase-II: Online phase.
   - Perform Initial state computations to get Initial state by using Lookup Table [14].
   - Recover the session key($K_{\text{enc}}$) by Reversal clocking or SAT solvers.

Experiment time analysis, for choosing random 64-bit state from the total space of $2^{64}$, then finding associative loop measuring as in the following table VII.

### Table VII

| S.No | Key space covered | time taken | Memory |
|---|---|---|---|
| 1 | Min - $2^{25}$ | 2.6 min | 16 bytes |
| 2 | Avg - $2^{29}$ | 3.5 min | 16 bytes |
| 3 | Max - $2^{37}$ | 4.8 min | 16 bytes |

Sample loop diagram as shown figure - 5 for understanding, loop-1 is having five states, loop-2 is having one state, and loop-3 is having two states. In this figure loop 1, 2 and 3 lengths are distinct, in other cases before intersecting loop, it may also intersect lines.

![Sample loop diagram](image)

Fig. 5. Sample loops diagram

C. Stepwise Procedure for Floyds-cycle-detection of State

In this proposed, we calculate the period and number of clocks required to generate the same sequence for the particular random state of A5/1 stream cipher (state which as no predecessors clock on A5/1 stream cipher).

- Let us take a state as variables $p$ and $q$ of the 64-bit value.
- Initially $p$ and $q$, both pointing at the random state.
- $p$ forward clock one time and $q$ forward clock two times at some point of time.
- $q$ is running at double speed, so definitely it will be ahead of $p$, so here it contains a loop, then $q$ at some point will enter in the loop. Sometime later $p$ will also enter in the loop.
- Now, when both $p, q$ are in the loop, and if they continue to clock at the same speed then eventually they will meet at the same state as shown in the figure - 6.

We calculate the periodicity with their lengths corresponding time in seconds of initial states. These initial states are randomly chosen from the A5/1 stream cipher which has no predecessors. The results are stored in the form of a lookup table as shown in the table - IX, this pre-computed lookup table is used in the attack phase. While experimenting we observed that minimum and maximum cycles for a particular state are 011182406, 469758320 respectively. The following table - VIII shows the minimum and maximum clocks and their corresponding state with loop intersection state.
is. It took a maximum of 239 clocks, to get the internal state. We covered 254 keyspace and stored it in table [6] with the size of 6 GB and total loops 17,715 as shown in the table X. Then with these partial results, we need to search for that sequence parallel in the 17,715 odd loops run on parallel threads.

One of these provides the concurrence and then we know the modified key. Once we identify the sequence in our thread, we know the previous 100 bits also from that point, we backtrack to the original session key using matrix multiplication.

### Table IX

**SAMPLE RESULT FOR MINIMUM LOOKUP TABLE**

| Initial state | Length of line | Keep intersections | length of loop | time (sec) |
|---------------|----------------|--------------------|----------------|------------|
| 0x8432a86a7631 | 140571836 | 0x1fd5794825908 | 0x89490957 | 20.60 secs |
| 0x542289ec6de91b18 | 0x0555585c70a64e2a | 0x22a3758f23f4162 | 0x077710786 | 20.60 secs |
| 0x3b08545c3ecae15a | 0x4a869337f58f2779 | 0x5e46a3258db5521 | 0x349478268 | 20.60 secs |
| 0x519b500d431bd7b7 | 0x643c986966334873 | 0x79e2a9e37545e146 | 0x3f2dba317c83e458 | 20.60 secs |
| 0x41a7c4c96b68079a | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 0x71f324542ca88611 | 20.60 secs |
| 0x333ab105721da317 | 0x08edbdab79838cb2 | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x257130a362bbd95a | 0x2443a8582d1d5ae9 | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x3804823e77465f01 | 0x6763845e75a2a8d4 | 0x2443a8582d1d5ae9 | 0x71f324542ca88611 | 20.60 secs |
| 0x436c6125628c895d | 0x2443a8582d1d5ae9 | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x218065232 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x1b7f9e55dab41e36 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x1a7b5c0dc9fcde24 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x268c169e0445f107 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x1fb93432ec26d636 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x6b3d2e4deec96f4 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x68b3d2e4deec96f4 | 0x189a769b54e49eb4 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 20.60 secs |
| 0xed87533626ed912b | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0xe60db4f81fa8c169 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x528a795f1a8d49eb | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0xf5b2c810cf694eff | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x56c9dff113abf8c4 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x554531071a61119d | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x90343aa8cddc4678 | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0xe2f3f383d81cde4e | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0x2085161a255c711e | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |
| 0xaff79482ec4ae60b | 0x42b2b1b6dd61969d | 0x71f324542ca88611 | 0x189a769b54e49eb4 | 20.60 secs |

### Table X

**TABLE GENERATIONS FOR MINIMUM LOOKUP TABLE**

| Computational facility | #lines | #loops | Key space covered | Time | Memory |
|------------------------|--------|--------|-------------------|------|--------|
| HPC                    | 288    | 17,715  | 254               | 20   | 6 GB   |

### E. Comparative Analysis

In this section, we discuss the comparative analysis of the A5/1 stream cipher and Tiny A5/1 stream cipher. In the table - XI, we compare all the parameters of the existing A5/1 and Tiny A5/1 stream cipher in terms of time, memory, and stream generation.
TABLE XI

| Description                        | A5/1 Stream cipher | Tiny A5/1 stream cipher |
|------------------------------------|---------------------|-------------------------|
| Size of the session key ($K_c$)   | 64-bit              | 16-bit                  |
| Size of the frame number ($F_c$)  | 22-bit              | 6-bit                   |
| Internal state size ($state$)     | 64-bit              | 16-bit                  |
| Keystream of each frame           | 114+114 bits        | 32+32 bits              |
| Execution for 1KB encryption file | 15 sec              | 10 sec                  |
| Time for Lookup table generation  | ≈ 20 day            | ≈ 1 day                 |
| Storage space                     | ≈ 6 GB              | ≈ 2 MB                  |
| Key space covered                 | 2^7                 | 2^6                     |
| Execution time for key recovery   | 90 sec to 10 min    | Max 30 sec with         |
|                                   | with 80% success    | 95% success probability |

V. CONCLUSION

The time-memory tradeoff attack retrieves the internal state which is afterload $K_c$, as well as deciphering a conversation, given ciphertext and known-plaintext bits. Previous attacks also often represented a high amount of precomputation and/or memory, as well as having a high time complexity. We present a minimised pre-computation table attack of recovering A5/1 stream cipher session key. The current attack is straightforward to execute. It has been introduced and uses a parallel method to accomplish the mission in less time. Finally, the presented attack reveals new implementation weaknesses in A5/1 that should be taken into account when creating new stream ciphers.

REFERENCES

[1] Vahid Amin Ghafari, Ali Vardarsi, and Javad Mohajeri, Cryptanalysis of GSM Encryption Algorithm A5/1- July 2012, Volume 4, Number 2 (pp. 107 - 114).
[2] Vadim Baluninsev, Alexander Semenov, Oleg Zaikin, and Stepan Kuchemazov, A Bitslice Implementation of Anderson’s Attack on A5/1, J.Open Engineering, Vol 8, Issue 1, 2018.
[3] Qurike, Jeremy (2004-05-01), Security in the GSM system, AusMobile.
[4] Marc Briceno., Ian Goldberg., David Wagner.: A pedagogical implementation of the GSM A5/1 and A5/2 “voice privacy” encryption algorithms,(1999). http://cryptome.org/gsm-a512.htm (originally on www.scard.org).
[5] ETSI: European Telecommunications Standards Institute- GSM Architecture:https://www.etsi.org.
[6] Jiqiang Lu., Zhou Li., and Matt Henrickson.: Time-Memory Trade-Off Attack on the GSM A5/1 Stream Cipher Using Commodity GPU, ACNS 2015, LNCS 9092, pp. 350-369, Springer International Publishing Switzerland (2015).
[7] Thomas Stockinger: GSM Network and its privacy-the A5 Stream Cipher, November (2005).
[8] Karsten Nohl.: Attacking phone privacy, BlackHat 2010 Lecture Notes, Security Research Labs, Berlin (2010).
[9] Patrik Ekdahl., Thomas Johansson.: Another Attack on A5/1, IEEE Transactions on Information Theory, Volume 49, Issue 1, pp. 284-289 (2003).
[10] Eli Biham., Orr Dunkelman.: Cryptanalysis of the A5/1 GSM Stream Cipher, Progress in Cryptology, proceedings of Indocrypt’00, Lecture Notes in Computer Science 1977, Springer-Verlag, pp. 43-51 (2000).
[11] Alexander Maximov., Thomas Johansson., Steve Babbage.: An improved correlation attack on A5/1, proceedings of SAC’04, LNCS 3357, pp. 1-18, Springer-Verlag, (2005).
[12] Alex Biryukov., Adi Shamir., David Wagner.: Real Time Cryptanalysis of A5/1 on a PC, Advances in Cryptology, proceedings of Fast Software Encryption’00, Lecture Notes in Computer Science 1978, Springer-Verlag, pp. 1-18 (2001).
[13] Y.Nagendar., Kamakshi Prasad V., Allam Appa Rao., Padmavathi G.: Applications of Stream ciphers in wireless communications, International Journal of Computer Sciences and Engineering, Vol.6, Issue 6, 2012, pp. 1121-1126 (2012).
[14] Elad Barkan., Eli Biham.: Conditional Estimators: An Effective Attack on A5/1,SAC 2005, LNCS 3897, pp. 1-19, Springer-Verlag Berlin Heidelberg (2006).
[15] Matthias Krause.; BDD-Based Cryptanalysis of Keystream Generators. EUROCRYPT 2002, LNCS 2332, pp. 222-237 (2002).
[16] Elad Barkan., Eli Biham., Nathan Keller.: Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. Journal of Cryptology, 21; 392-429,(2008).
[17] Maria Kalender., Dominis Pneumatikatos., Ioannis Pappaefstathiou., Charalampos Manifavas.: Breaking the GSM A5/1 cryptography algorithm with rainbow tables and high-end FPGAS. 22nd International Conference on Field Programmable Logic and Applications (FPL), 29-31 Aug. (2012).
[18] Meyer, S.: Breaking GSM With Rainbow Tables, arXiv preprint arXiv:1107.1086, (2011).
[19] Golic, J.,Cryptanalysis of Alleged A5 Stream Cipher, In Proc. of Eurocrypt’97, vol 1233 of LNCS, pp 239-255. Springer-Verlag, 1997.
[20] S. Babbage., A Space/Time Tradeoff in Exhaustive Search Attacks on Stream Ciphers, European Conference on Security and Detection, IEE Conference Publication No. 408 May 1995.
[21] A. Biryukov and A. Shamir., Cryptanalytic time/memory/data tradeoffs for stream ciphers, ASIACRYPT 2000, Lecture Notes in Computer Science, vol. 1976, pp. 1-13, 2000.
[22] A. Biryukov, A. Shamir, and D. Wagner., Real time cryptanalysis of A5/1 on a PC, FSE 2001, Lecture Notes in Computer Science, vol. 1978, pages 37-44, 2002.
[23] M. E. Hellman., A cryptanalytic time-memory-trade-off, IEEE Transactions on Information Theory, vol. 26, pp. 401-406, July 1980.
[24] Gundaram P.K., Allu S.N., Yenikala N., Tentu A.N. (2021) Rainbow Tables for Cryptanalysis of A5/1 Stream Cipher. In: Palesi M., Trajkovic M., (eds) Cryptography and Network Security, 10th International Conference, CANS 2021, Brno, Czech Republic, September 10-12, 2021, Proceedings, pp. 392-409. Springer, Cham.
[25] Wang M., Hao Y. (2021) Revisit Two Memoryless State-Recovery Cryptanalysis Methods on A5/1. In: Yu Y., Yang M. (eds) Information Security and Cryptology. Inscrypt 2021. Lecture Notes in Computer Science, vol 13007. Springer, Cham.
[26] A. Beckmann, J. Federowicz, J. Keller, and U. Meyer.A structural analysis of the A5/1 state transition graph”, Germany, 2012.

G Praveen Kumar is pursuing his Ph.D in computer science & engineering, university college of science from Acharya Nagarjuna University, Guntur, Andhra Pradesh and also working as Research Associate in CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS), University of Hyderabad Campus, Hyderabad. He did his MCA from Osmania University. His areas of interest are network security, cryptology and cryptanalysis and design of security protocols.

Appala Naidu Tentu is an Associate Professor at CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS), University of Hyderabad Campus. He obtained Ph.D in Computer Science and Engineering (specialisation in Cryptography and Information Security) from JNTU Hyderabad. His current research interests are in the areas of cryptography, cryptanalysis and design of security protocols.

Allu Swamy Naidu is pursuing his Ph.D in computer science & engineering, university college of science from Acharya Nagarjuna University, Guntur, Andhra Pradesh and also working as Research Associate in CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS), University of Hyderabad Campus, Hyderabad. He did his M.Tech from IIT Kharagpur. His areas of interest are Cryptography and Cryptanalysis, Quantum and Post Quantum Cryptography, Coding theory.