Initialisation Flaws in the A5-GMR-1 Satphone Encryption Algorithm

V. Bhartia
Queensland University of Technology
GPO Box 2434,
Brisbane, Queensland 4001
Australia
vishesh.bhartiya@connect.qut.edu

L. Simpson
Queensland University of Technology
GPO Box 2434,
Brisbane, Queensland 4001
Australia
lr.simpson@qut.edu.au

ABSTRACT

A5-GMR-1 is a synchronous stream cipher used to provide confidentiality for communications between satellite phones and satellites. The keystream generator may be considered as a finite state machine, with an internal state of 81 bits. The design is based on four linear feedback shift registers, three of which are irregularly clocked. The keystream generator takes a 64-bit secret key and 19-bit frame number as inputs, and produces an output keystream of length berween 2^8 and 2^{10} bits.

Analysis of the initialisation process for the keystream generator reveals serious flaws which significantly reduce the number of distinct keystreams that the generator can produce. Multiple (key, frame number) pairs produce the same keystream, and the relationship between the various pairs is easy to determine. Additionally, many of the keystream sequences produced are phase shifted versions of each other, for very small phase shifts. These features increase the effectiveness of generic time-memory tradeoff attacks on the cipher, making such attacks feasible.

CCS Concepts

•**Security and privacy** → *Symmetric cryptography and hash functions;* **Block and stream ciphers;** *Cryptanalysis and other attacks;*

Keywords

Stream ciphers; satellite phones; initialisation; cryptanalysis; time-memory attacks.

1. INTRODUCTION

Satellite phones, or satphones, use satellites in Earth's orbit to establish network connections, providing phone coverage without the terrestrial infrastructure required for cellular mobile phones. There are two main satphone standards, known as GMR-1, and GMR-2. Details of the encryption algorithms used to protect the communication between satphone and satellite have been made public recently. These were obtained by reverse engineering satphones. Descriptions of the stream ciphers used with GMR-1 and GMR-2, known as A5-GMR1 and A5-GMR2, respectively, are given in [3], along with preliminary cryptanalysis of the algorithms. In this paper we focus on the initialisation process of the A5-GMR-1 algorithm.

Satphone communications are divided into frames. Each frame is encrypted by XORing with a binary keystream sequence. The A5-GMR-1 keystream generator uses a secret 64-bit key, K, for all frames in a conversation. This secret key is combined with a known 19-bit frame number, E, which differs for each frame. Initialisation using K and E must be performed before a keystream segment of the required length can be generated and used for encryption or decryption of the frame. An outline of the initialisation process is provided in [3] with additional detail given in [4]. A good initialisation process should ensure that each (K,E) pair generates a distinct and unpredictable keystream sequence, and that multiple sequences produced with the same key but different frame numbers appear unrelated. Also, if the internal state of the keystream generator is revealed at some time during keystream generation, it should be difficult to establish the secret key from the known state.

In this paper, we consider the description of A5-GMR-1 gleaned from [3] and [4]. We investigate vulnerabilities introduced by poor design choices in the initialisation process. We consider the keystream generator as a finite state machine, and examine the state cycles formed by tracing paths of state transitions that occur during initialisation and keystream generation. We make observations about possible initial states and the subsequent paths of internal states and relate these to the corresponding keystream sequences. Unfortunately, for A5-GMR-1 the initialisation process results in multiple (K,E) pairs that produce the same keystream sequence, and it is simple to determine relationships between the values of K and E for which this occurs. We identify features of the initialisation process that greatly reduce the keystream variability and therefore reduce the resistance of A5-GMR-1 to common forms of known-plaintext attack.

2. DESCRIPTION OF A5-GMR-1

The A5-GMR-1 keystream generator is a finite state machine with the state contained in four binary linear feedback shift registers (LFSRs) [3]. The LFSRs are denoted R_1, R_2, R_3 and R_4; and have 19, 22, 23 and 17 binary stages, respec-

AISC 2016 February 2–5, 2016, Canberra, Australia

DOI: 10.475/123_4

tively. Thus the keystream generator has an internal state size of 81 bits. Figure 1 shows the four LFSRs and the associated functions for keystream generation mode. Table 1 gives the feedback polynomials for the LFSRs.

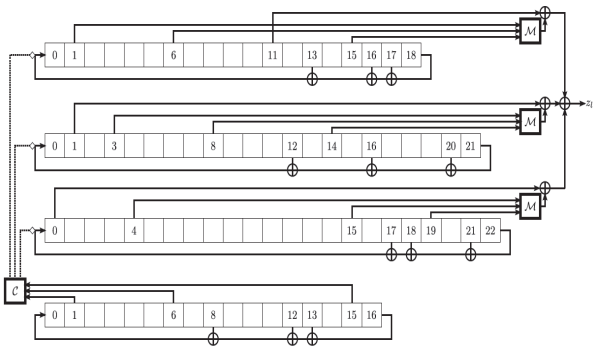

Figure 1: The A5-GMR1 keystream generator

The A5-GMR-1 cipher has two modes of operation: initialisation and keystream generation. The same LFSRs are used in each mode, but the state update functions vary depending on whether clocking of the registers is regular or irregular. Note that R_4 is always regularly clocked.

Let $R_{i,j}$ denote the j^{th} stage of the i^{th} register, for appropriate i and j, and $R_{i,j}(t)$ denote the contents of this stage at time t. When irregular clocking is used, the contents of three stages from R_4 (namely $R_{4,1}$, $R_{4,6}$ and $R_{4,15}$) are used as inputs to a majority function $\mathcal{M}$ to determine which of the other three registers will be clocked. The majority function is a quadratic boolean function represented algebraically as $\mathcal{M}(x_1, x_2, x_3) = x_1x_2 \oplus x_2x_3 \oplus x_1x_3$. This function controls the clocking of registers R_1, R_2 and R_3 as follows:

$$\text{Iff } \mathcal{M}(R_{4,1}, R_{4,6}, R_{4,15}) = \begin{cases} R_{4,15} \text{ then } R_1 \text{ is clocked.} \\ R_{4,6} \text{ then } R_2 \text{ is clocked.} \\ R_{4,1} \text{ then } R_3 \text{ is clocked.} \end{cases} \quad (1)$$

The use of the majority function means that at least two of the three registers, R_1, R_2 and R_3, will be clocked at each time step. The clock control is denoted $\mathcal{C}$ in Figure 1. We describe the steps in the initialisation and keystream generation processes below.

2.1 Initialisation

The A5-GMR-1 cipher uses a 64-bit key K and 19-bit frame number E. Let k_i represent the i^{th} bit of K, where $0 \leq i < 64$, so that K is represented by $K = (k_0k_1 \ \ldots \ k_{63})$. Similarly, let e_j represent the j^{th} bit of E, where $0 \leq j < 19$, so that E is represented by $E = (e_0e_1 \ \ldots \ e_{18})$. To produce the keystream sequence that will be used to encrypt frame number E in the communication, both K and E are input to the initialisation mode.

Initialisation is performed in two phases, which we refer to as *loading* and *diffusion*. First the 64-bit K and 19-bit E are combined to form a 64-bit initialisation vector I, where I is represented by $I = (v_0v_1 \ \ldots \ v_{63})$. This vector is loaded into the internal state, and then a nonlinear update function is applied for diffusion.

Note that the bits in I are obtained as linear combinations of the bits of K and E [4]:

$$\begin{aligned} I &= [v_0v_1 \ldots v_{63}] \\ &= [(k_{15} \oplus e_{18})(k_{14} \oplus e_{17})(k_{13} \oplus e_{16})(k_{12} \oplus e_{15})(k_{11} \oplus e_{14}) \\ &(k_{10} \oplus e_{13})(k_9 \oplus e_{12})(k_8 \oplus e_{11})(k_7 \oplus e_{10})(k_6 \oplus e_9)(k_5 \oplus e_8) \\ &(k_4 \oplus e_7)(k_3 \oplus e_6)k_2k_1k_0k_{31}k_{30}k_{29}k_{28}k_{27}k_{26}k_{25}k_{24}(k_{23} \oplus e_5) \\ &(k_{22} \oplus e_4)k_{21}k_{20}k_{19}k_{18}k_{17}k_{16}k_{47}k_{46} \ldots k_{33}k_{32}(k_{63} \oplus e_3) \\ &(k_{62} \oplus e_2)(k_{61} \oplus e_1)(k_{60} \oplus e_0)k_{59}k_{58}k_{57}k_{56}k_{55}k_{54}k_{53}k_{52} \\ &k_{51}k_{50}k_{49}k_{48}) \end{aligned}$$

Specifically, the initialisation is performed as follows:

- *Loading phase:*
  1. The content of each stage in each of the registers is set to 0.
  2. The 64-bit initialization vector I is formed from K and E.
  3. Over the next 64 clocks I is loaded into each of the four registers separately, using regular clocking. For R_i, where $i = 1, 2, 3, 4$ the loading is performed by XORing a bit of I with the linear feedback for R_i to form the new contents of $R_{i,0}$.
  4. The contents of register stages $R_{i,0}$, for $i = 1, 2, 3, 4$ are set to 1.
- *Diffusion phase:*
  1. The cipher is clocked 250 times. R_4 is regularly clocked each time, and controls the clocking of the other registers as in Equation 1. During this time no keystream output is produced.

For a given K and E, we refer to the internal state of the keystream generator at the end of the loading phase as the *l*oaded state, and at the end of the diffusion phase as the *i*nitial state, respectively. Once the initial state is formed, the generator can begin to produce the keystream sequence.

2.2 Keystream Generation

In keystream generation mode, the state update function for the cipher is the same as during the diffusion phase of initialisation. Register R_4 is regularly clocked and the majority function $\mathcal{M}$ is applied to the contents of three stages of R_4 to determine which of the other three registers will be clocked, as outlined in Equation 1.

Each time the internal state is updated, the output function is applied and a keystream bit produced. The output function takes inputs from 12 stages in the internal state: four from each of registers R_1,R_2 and R_3. Three of the stages in each register are used as input to the majority function $\mathcal{M}$ and the output of $\mathcal{M}$ is XORed with the contents of another stage in the register. Then the resultant values from each register are linearly combined. Specifically,

$$\begin{aligned} z_t = &\mathcal{M}(R_{1,1}(t), R_{1,6}(t), R_{1,15}(t)) \oplus R_{1,11}(t) \\ &\oplus \mathcal{M}(R_{2,3}(t), R_{2,8}(t), R_{2,14}(t)) \oplus R_{2,1}(t) \\ &\oplus \mathcal{M}(R_{3,4}(t), R_{3,15}(t), R_{3,19}(t)) \oplus R_{3,0}(t). \end{aligned} \quad (2)$$

Register	Length	Feedback polynomial
$R1$	19	$x^{19} + x^{18} + x^{17}x^{14} + 1$
$R2$	22	$x^{22} + x^{21} + x^{17}x^{13} + 1$
$R3$	23	$x^{23} + x^{22} + x^{19}x^{18} + 1$
$R4$	17	$x^{17} + x^{14} + x^{13}x^{9} + 1$

Table 1: LFSR feedback polynomials

The amount of keystream required to encrypt or decrypt a frame depends on the transmission channel. In [3] frame lengths for the TCH3, TCH6 and TCH9 channels are given as $m = 208$, 420 and 648 bits, respectively. For a given K and E, $2m$ bits are produced, with either the first m or second m bits of the keystream being used, depending on the direction of the transmission (satphone to satellite, or vice versa). Following this, the frame number is incremented and the keystream generator reinitialised. Thus for a particular frame in a conversation, the amount of keystream available for a known plaintext attack is quite small: between 2^8 and 2^{10} bits.

3. OBSERVATIONS

In this section we consider the A5-GMR-1 keystream generator as a finite state machine. We make a series of observations about the number of distinct internal states of the keystream generator that can be obtained at various points of the initialisation process; during the loading phase, through the diffusion phase and also during keystream generation. As the A5-GMR-1 state update function is the same during diffusion and keystream generation, we determine the possible cycles of internal states that can occur during both the diffusion phase of initialisation and keystream generation.

1. **Number of distinct initialisation vectors** After Step 2 of the loading phase in the initialisation process, a 64-bit vector I is formed, with elements which are simple linear combinations of bits of the 64-bit K and 19-bit E. There are $2^{64+19} = 2^{83}$ distinct (K, E) pairs, but only 2^{64} distinct I. We note that, given the linear combination, there are 2^{19} distinct (K, E) pairs for each I.

2. **Number of distinct loaded states** In Step 3 of the loading phase in the initialisation process, I is loaded into each of the registers linearly. The last iteration of this process includes the final bit of I ($v_{63} = k_{48}$) in the new value of register stages $R_{i,0}$, for $i = 1, 2, 3, 4$. However, in Step 4, the value of $R_{i,0}$, for $i = 1, 2, 3$, and 4 is set to 1. Thus $v_{63} = k_{48}$ is ineffective in forming the loaded states (equivalent to using a 63-bit key). That is, although there are 2^{83} distinct (K, E) pairs, and 2^{64} distinct values for I, there are only 2^{63} distinct loaded states in total. Therefore there are 2^{20} distinct (K, E) pairs for each loaded state of A5-GMR-1.

3. **Maximum number of possible states:** The final step of the loading phase sets $R_{i,0} = 1$ for $i = 1, 2, 3$ and 4. Thus no LFSR can be in an all-zero state at the start of the diffusion phase. During both diffusion and keystream generation, $R_{4,0}$ is autonomous and regularly clocked. Aside from the clock control, registers R_i for $i = 1, 2, 3$ are also autonomous in these phases. If clocked, the new value for $R_{i,0}(t + 1)$ is computed by applying the feedback function for R_i (given in Table 1) to $R_i(t)$, where $i = 1, 2, 3$ and 4. Given the non-zero loaded states for each register and the use of primitive polynomials for the feedback functions, no register will revert to an all zero state. The number of possible states for each component register R_i is given by $2^{|R_i|} - 1$, where $|R_i|$ denotes the length of R_i, for $i = 1, 2, 3$ and 4. Therefore, after the loading phase, the maximum number of possible states the keystream generator can be in at any time is given by:

$$\begin{aligned} G &= (2^{|R_1|} - 1)(2^{|R_2|} - 1)(2^{|R_3|} - 1)(2^{|R_4|} - 1) \\ &= (2^{19} - 1)(2^{22} - 1)(2^{23} - 1)(2^{17} - 1) \approx 2^{81} \end{aligned}$$

4. **Cyclic structure of state transition graph:** The irregular clocking of R_1, R_2 and R_3 is under the control of regularly clocked R_4. There are $2^{17} - 1$ possible non-zero states for R_4, each with one successor state and one predecessor state. Therefore each state in the A5-GMR-1 state transition graph has exactly one successor state and one predecessor state. That is, all of the states in the state transition graph lie on cycles: either one cycle that encompasses all G states, or on several disjoint cycles.

5. **Number of distinct cycles:** After the loading phase of initialisation, no register can be in an all-zero state. Thus the binary sequence produced by R_4 has period $P_{R_4} = 2^{17} - 1 = 131,071$. Consider possible values of the three stages of R_4 providing inputs to the majority function as a triplet. In one complete cycle of the non-zero states of R_4, each triplet occurs 2^{14} times, except for the all zero triplet, which occurs $2^{14} - 1$ times. Thus completing one cycle of R_4 states results in each of the other three registers being clocked exactly $6 \times 2^{14} - 1 = 98,303$ times. The state cycles for R_1, R_2 and R_3, if regularly clocked, contain $(2^{19} - 1)$, $(2^{22} - 1)$ and $(2^{23} - 1)$ states, respectively. All of these values are relatively prime to 98, 303, and to each other. Therefore, the structure of the state transition graph for the A5-GMR-1 cipher during the diffusion and keystream generation phases is one big cycle that encompasses all G states.

6. **Number of distinct initial states** Consider the loaded state corresponding to a (K, E) pair as a starting point on a path of internal state values. Under the state update function for both diffusion and keystream generation, the internal states form a single cycle containing G states. In the diffusion phase the state update function is bijective, and is applied 250 times.

Therefore every distinct loaded state has a corresponding initial state which is a distance of 250 steps away on the state cycle.

7. **Number of distinct internal states visited during keystream generation** Consider the production of keystream segments of length 2^{10} bits (longer than used in the satphone application described). Generating a keystream segment of this length involves stepping through 2^{10} distinct internal states. Suppose this path of 2^{10} internal states contains no states which are themselves initial states (that is, no paths have overlapping segments). As there are only 2^{63} distinct initial states, this assumption implies that 2^{73} distinct internal states can be visited during keystream generation, over all possible (K, E) values. Note that this is much less than the value of G; given the number of distinct initial states and the length of the keystream required for the application, if the initial states are uniformly scattered across the state cycle then the proportion of internal states ever visited during keystream generation is approximately $2^{73}/2^{81} = 2^{-8}$. If the assumption that there are no overlapping paths does not hold, then the number of distinct initial states that will ever be visited during keystream generation will be further reduced. Clearly, a large proportion of possible internal state values for A5-GMR1 will never be reached in the satphone application.

4. INVESTIGATING THE INITIAL STATE DISTRIBUTION

In this section we investigate the distribution of initial states on the A5-GMR1 state transition graph to determine whether paths of internal states do overlap. In that case, segments of the corresponding keystreams also overlap and the keystreams produced from these initial states will be phase-shifted versions of one another. There is a direct correspondence between loaded and initial states, with initial states at a fixed distance of 250 steps ahead of the respective loaded state on the state cycle. Hence the distribution of initial states on the A5-GMR1 state transition graph is the same as the distribution of loaded states. As loaded states have a readily detectable format, we investigate the distribution of loaded states instead.

4.1 Distribution of loaded states

Loaded states in A5-GMR1 have a particular format, due to both the construction of I and the loading phase during initialisation being entirely linear. Although there are 81 stages in the state, the contents of four of these stages are set to 1, and the other 77 stages are linear combinations of the 63 bits of I: $v_0 v_1 \dots v_{62}$. Thus the loaded state can be described by a system of 77 linear equations in 63 variables. As this system is overdetermined by fourteen linear equations, these fourteen equations and also $R_{i,0} = 1$, for $i = 1, 2, 3, 4$ can be used as a sufficient condition to determine whether a particular internal state is a loaded state for some set of (K, E) values.

Consider the possibility that the distance between two loaded states L_1 and L_2 is one. That is, L_2 is the state obtained after the state update function has been applied to L_1. We derive a system of linear equations to identify loaded states L_1 which, when the cipher is clocked once, produce another loaded state, L_2 (so the required format holds for L_2 also). We refer to L_1 and L_2 as a slid pair with distance 1. As the clocking during diffusion is irregular, there are four cases to consider, based on the possible values of the inputs to $\mathcal{M}$. These are given in Table 2, where for brevity we use $a, b \in \{0, 1\}$ with b denoting the binary complement of a.

Considering the pair-wise XOR of the R_4 input values to $\mathcal{M}$ in L_1, we have three linear equations in each of these four cases. For L_2 to also be a loaded state two conditions have to be met. Firstly, we require the contents of $R_{i,0} = 1$, for $i = 1, 2, 3, 4$. This provides four equations in the case where all four registers are clocked, and three equations in cases where one of the registers is not clocked. Secondly, we have the 14 linear equations relating state bits as a result of the overdefined equation system for loaded states. Therefore, we have 21 or 22 linear, though not neccesarily independent, equations. To determine the number of linearly independent equations, and hence the number of free variables, we can calculate the ranks of these matrices. The results are listed in Table 3. Thus there are $2^{53} + 2^{44} + 2^{44} + 2^{44} \approx 2^{53.0084}$ pairs of loaded states with a distance of 1.

A similar method can be used to derive equations for slid pairs L_1 and L_2 with a distance of two, or three, or greater. However, this will result in some double counting. Consider three loaded states; L_1, L_2 and L_3; where L_1 and L_2 have distance 1, and L_2 and L_3 have distance 1. Then the distance between L_1 and L_3 is two. To count loaded states that are exclusively at distance 2 from other loaded states, we need to adjust to account for the mumber of slid pairs of distance 1 also included. That is, the number of slid pairs of exactly distance 2 is given by the number of slid pairs of distance two less the number of consecutive slid pairs of distance 1. This approach is extended to determine the number of slid pairs of loaded states that are exclusively of distance 3. The number of slid pairs exclusively at distance 1, 2 and 3 are given in Table 4.

4.2 Distribution of initial states

From Table 4 it is clear that the 2^{63} distinct loaded states for A5-GMR1 are not uniformly distributed around the internal state cycle. Many loaded states are only a short distance from each other. As there are exactly 250 state transitions from each loaded state to the corresponding initial state, then the 2^{63} distinct initial states are similarly distributed. The number of distinct internal states that will ever be visited during keystream generation, across all possible (K, E) values, is clearly much less than 2^{73}, and represents much less than 2^{-8} of the possible 81-bit internal state values.

For some initial states, the subsequent state is also an initial state. This implies that the two keystreams produced by a slid pair of initial states will be phase shifts of each other, for very short shift distances. Hence many of the keystream segments produced will have substantial overlapping segments. This reduces the variability of the keystreams produced to encrypt each frame.

5. ATTACKING A5-GMR1

In this section we discuss several simple attacks on the A5-GMR1 keystream generator. Firstly, we describe attacks in the existing public literature: a guess and determine attack presented as the preliminary cryptanalysis of the keystream

$R_{4,15}$	$R_{4,6}$	$R_{4,1}$	$\mathcal{M}$	$R_{4,15} \oplus R_{4,6}$	$R_{4,6} \oplus R_{4,1}$	$R_{4,15} \oplus R_{4,1}$
a	a	a	a	0	0	0
a	a	b	a	0	1	1
a	b	a	a	1	1	0
a	b	b	b	1	0	1

Table 2: Register clocking

Registers Clocked	Rank	Free variables	Number of states
$R_1, R_2, R_3$and R_4	10	53	2^{53}
R_1, R_2 and R_4	19	44	2^{44}
R_2, R_3 and R_4	19	44	2^{44}
R_1, R_3 and R_4	19	44	2^{44}

Table 3: Equation systems for loaded states with distance 1

generator in [3], and an improved version in [4] that uses keystreams from multiple frames. Then we discuss the application of a generic time-memory attack to the cipher.

5.1 Previous Attacks

Preliminary cryptanalysis in [3] describes a known-plaintext attack on the A5-GMR-1 cipher that adapts the ideas of Petrovic and Fuster-Sabater [6]. The approach is to make a 16-bit guess of the contents of R_4 at the end of the loading phase, and then combine this with an algebraic attack. The contents of R_4 determine the clocking of the other registers so, for any particular guess, the A5-GMR1 state update function is essentially linear. Variables are assigned to the $(18 + 21 + 22 = 61)$ unknown contents of the stages in the other three registers. The internal state values at future time points can then be expressed as linear combinations of these variables. The output function is quadratic, so the known keystream bits can be expressed as a series of binary quadratic equations in terms of the assigned variables. Linearization can be applied to create a linear system in 655 variables. Using the known keystream to solve the system of equations yields values in a loaded state for R_1, R_2 and R_3 that correspond to the guessed R_4. The candidate solution should be tested to determine if it is correct (produces the same keystream). If not, the guessed value of R_4 is discarded and the process repeated for another 16-bit guess. If a suitable candidate is found, then I is obtained from the loaded state, and then K is obtained from I, given the known value of E.

A limitation of this approach is the keystream requirement: at least 655 consecutive bits must be produced from a single initialisation. This exceeds the frame lengths noted in [3] for the transmission channels. Thus the attack as described in [3] is of academic interest but not practically feasible. Also, apart from the condition that $R_{i,0} = 1$, for $i = 1, 2, 3, 4$, the attack does not consider the loading phase in the initialisation process. It begins by assigning variables to the loaded state. Thus for each 16 bit guess of R_4, 61 variables are required for the unknown stages in the remaining registers. This is almost as many variables as in the initialisation vector I. When the additional computation required to solve the equation systems is included, this approach is less effective than exhaustive search over the effective 63 bits of I or, for that matter, K.

To reduce the number of variables in these keystream equations and hence reduce the keystream requirement, the number of guessed state bits can be increased to include some of the stages in registers R_1, R_2 and R_3. For example, guessing an additional 17 bits can reduce the number of variables in the equation system to 345. This approach requires guessing 33 bits in total, and then constructing and solving a system of equations in 345 variables for each guess. Meeting the reduced keystream requirement is achieved at a cost of an increased number of candidate guesses, and so the attack remains impractical.

In [4] the limitation on obtaining sufficient keystream from a single frame is dealt with by noting that two initial states generated from the same key but with different frame numbers are linearly related (corresponds to our Observation 1). The XOR difference between two frame numbers thus translates to a linear difference in the 81-bit initial states in the different frames. This corresponds to a linear translation of the constructed equation system to be solved, as the matrix relating the initial state to the keystream will be different for different frames. Multiple consecutive frames (approximately 12 or 13) in the same communication were required, but this attack could be successfully implemented on obtained keystream. That is, it is the linear combination of the key bits and frame number bits that makes it possible to obtain sufficient keystream to perform an algebraic attack.

5.2 Time/Memory/Data Attack

In this section we investigate the application of generic Time/Memory/Data (TMD) attacks to A5-GMR-1. TMD attacks have been proposed as a means to invert functions. Given the output of a function, an attacker can use a TMD attack to recover the input. The TMD attack can be applied to stream ciphers as a known-plaintext attack. The known plaintext is used to reveal keystream segments, and these are used to try to identify the underlying internal state. In Section 4 we showed that, given the initialisation procedure, the possible internal state values of A5-GMR-1 occuring during keystream generation represent only a small fraction of the state values. This resulted in reduced variability of the resultant keystreams, and indicates that A5-GMR-1 may be vulnerable to TMD attacks.

Time memory attacks are performed in two phases: a precomputation phase in which lookup tables are prepared, and an online phase in which the attacker obtains data and searches the lookup table to find a match that enables the function to be inverted. Time-Memory-Data attacks are ex-

Distance (exclusive)	Number of states	Proportion of states
1	$2^{53.0084}$	0.0982%
2	$2^{52.0278}$	0.0498%
3	$2^{51.2945}$	0.0299%
TOTAL	$2^{53.8657}$	0.1799%

Table 4: Pairs of loaded states with exclusive distance 1, 2 or 3

pressed in terms of the following parameters:

- N - Size of the search space
- P - Precomputation time (time required to generate lookup table)
- M - Amount of random access memory to store precomputed data
- T - Time required for online phase
- D - Amount of data available to the attacker in the online phase.

We consider two Time/Memory/Data Tradeoff attacks here: the original attack described independently by Babbage [1] and Golic [5], referred to as the Babbage-Golic attack, and the attack described by Biryukov and Shamir [2], referred to as the Biryukov-Shamir attack. The Babbage-Golic attack has a tradeoff curve $DM = N$, with $T = D$ and $P = M$. As noted in [2], T can be reduced if some of the input data is selectively ignored, but we don't consider that to be the case here. The tradeoff curve for the Biryukov-Shamir attack is $TD^2M^2 = N^2$, with $D^2 \leq T \leq N$ and $P = N/D$.

Consider the size of the search space. If we search for the input pair (K, E) to the cipher, then $N = 2^{64+19} = 2^{83}$. Note that this is greater than the set of all possible state values. If we search over all possible internal states of the cipher, then $N = 2^{81}$. However, from Section 4 it is clear that many internal states will never be visited during keystream generation, so this is a wasteful approach. If we search instead for initial states, this is reduced to $N = 2^{63}$. In this case, we construct the lookup table by randomly choosing M 63-bit initialisation vectors I_i (with v_{63} set to 0 since it is ineffective), and for each deriving an initial state, S_i, and producing a 63-bit keystream segment Z_i. The lookup table consists of pairs of entries (Z_i, S_i).

In the online phase, given some keystream data, we apply a 63-bit sliding window and consider all possible 63-bit keystream segments that can be formed from the data stream. If we find a Z_t that exists in our table, then we know the internal state at that time was S_t. As the state update function is bijective, this can be readily inverted to obtain the corresponding initial state S_i and the initialisation vector I_i. Given that the frame number E is known, the secret key K can be reconstructed (apart from the ineffective bit k_{48}, as noted in Observation 5). One lookup table with M initial state entries will suffice for all K and E values.

Now consider the data available to an attacker. If the keystream from only one frame is available, we have at least 208 bits, which provides 145 63-bit keystream segments, so $D = 145$. D can be readily increased by using data from multiple frames, with a maximum of 2^{19} frames. That is, the maximum value for $D \approx 2^{26}$. For the Babbage-Golic attack with $N = 2^{63}$, if $D = T = 2^{26}$ then $M = P = 2^{37}$. Clearly constructing a lookup table for this value for M is feasible, although D is extremely large. Reducing D to 2^{20} will increase M to 2^{43}, which is easily possible.

Similarly, for the Biryukov-Shamir attack if $D = 2^{26}$ then $TM^2 = 2^{74}$, with $T > 2^{52}$, so requirements for M are again very small. Tradoffs that reduce D substantially can still be performed for feasible values of M and T. For example, if $D = 2^{20}$ then $TM^2 = 2^{86}$, with $T > 2^{40}$, and $M < 2^{23}$.

5.3 Vulnerabilities related to initialisation

There are several serious flaws in the initialisation process that make A5-GMR1 vulnerable to attack. These relate to design decisions in how the 64-bit key K and the 19-bit frame number E are used to form the initial state. In the loading phase, the 64-bit key K and the 19-bit frame number E are combined linearly to form a 64-bit value I, which is then loaded into the component LFSRs in a linear way to form an 81-bit state, and then four of those bits are set to a constant value of 1. Unfortunately the stages in each register which are set to the value 1 correspond to the bit in I formed from a single key bit k_{48}. This step effectively removes that key bit from the keystream generator. The loading phase has thus reduced 83 bits of input to 63 bits in forming the initial state. The diffusion phase of the initialisation process has no further external input, so cannot introduce any more variability.

The linearity of the combining function is a vulnerability exploited by the multi-frame algebraic attacks in [4]. It is also a contributing factor to the success of the generic TMD attacks, as a single lookup table can be used for data obtained from any frame. Essentially, the lookup table constructed in the precomputation stage of a TMD attack is based on I. Then in the online attack phase, once a match for a value of I is obtained in any frame, the known value of E can be substituted to obtain the corresponding values for K. Note that there will be two keys which are valid for any case, as the value of k_{48} can be either 0 or 1, since it is not actually effective. The TMD attacks are aided by the reduction in the effective initial state space, noted above, and also by the fact that a large proportion of the possible internal states of the cipher will never be visited during the keystream generation process. Different design choices regarding the loading and diffusion phases of initialisation may have made both of these attacks more difficult.

6. CONCLUSION

A5-GMR-1 is a synchronous stream cipher used to provide confidentiality for communications between satellite phones and satellites. The use of linear feedback shift registers in the design makes it vulnerable to algebraic attacks, although guessing a large number of state bits and solving large systems of equations is required in order for these attacks to be applied. These attacks recover the internal state of the ci-

pher, and then work backwards to determine the value of the secret key. Given the relatively small amount of keystream produced per frame, a naive implemetation of an algebraic attack is not feasible. A more complex algebraic attack which uses keystreams from multiple consecutive frames is possible, requiring the use of linear translations of the equation system for each frame.

In this paper, we consider A5-GMR1 as a finite state machine, and analyse the state transitions during both the initialisation and keystream generation processes. Our analysis reveals several serious flaws in the initialisation process which significantly reduce the number of distinct keystreams that the cipher can produce.

Although the keystream generator takes a 64-bit key and a 19-bit frame number as inputs to an 81-bit state, the flaws in the initialisation process mean that only 2^{63} distinct initial states can be formed, and hence only 2^{63} distinct keystreams can actually be produced. Many of these distinct keystreams are phase shifted versions of each other, for very small phase shifts (say 1, 2 or 3 bits). Thus, the internal state values that can occur during keystream generation represents only a small proportion of all possible internal state values. Additionally, the linear method for combining the key and frame number during initialisation makes it simple to determine (key, frame number) pairs which produce the same initial state, and hence produce exactly the same keystream. There are 2^{20} such (key, frame number) pairs for each keystream. Finally, we point out that although the cipher uses a 64-bit key, the initialisation process quickly cancels the effect of one of these key bits during the loading process. These flaws in the initialisation process reduce the effective state size from 81 bits to 63.

Given the key size, this reduction clearly shows that A5-GMR-1 will be vulnerable to simple generic time-memory tradeoff attacks. Multiple points on the TMD tradeoff curve (time, memory and known keystream requirements) show this approach to be feasible in the satellite phone application scenario.

7. REFERENCES

[1] S. Babbage. A space/time tradeoff in exhaustive search attacks on stream ciphers. In *European Convention on Security and Detection, 1995*, number 408. IEE, 1995.

[2] A. Biryukov and A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In *Advances in Cryptology ASIACRYPT 2000*, pages 1–13. Springer, 2000.

[3] B. Driessen, R. Hund, C. Willems, C. Paar, and T. Holz. Don't trust satellite phones: A security analysis of two satphone standards. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 128–142. IEEE, 2012.

[4] B. Driessen, R. Hund, C. Willems, C. Paar, and T. Holz. An experimental security analysis of two satphone standards. *ACM Transactions on Information and System Security (TISSEC)*, 16(3):10, 2013.

[5] J. D. Golić. Cryptanalysis of alleged a5 stream cipher. In *Advances in Cryptology EUROCRYPT97*, pages 239–255. Springer, 1997.

[6] S. Petrovic and A. Fuster-Sabater. Cryptanalysis of the a5/2 algorithm. *IACR Cryptology ePrint Archive*, 2000:52, 2000.