CRYPTANALYSIS OF THE A5/2 ALGORITHM

Slobodan Petrović* and Amparo Fúster-Sabater*

Abstract - An attack on the A5/2 stream cipher algorithm is described, that determines the linear relations among the output sequence bits. The vast majority of the unknown output bits can be reconstructed. The time complexity of the attack is proportional to 2^{17}.

Introduction: A5 is the stream cipher algorithm used to encrypt the link from the telephone to the base station in the GSM system. According to [1], two versions of A5 exist: A5/1, the 'stronger' version, and A5/2, the 'weaker' version. The attacks on the A5/1, utilizing the birthday paradox, are described in [2, 3]. The attack on the A5/2 presented here is of algebraic nature.

The scheme of the A5/2 algorithm is given in the Fig. 1. The LFSR R_4 clocks the LFSRs $R_1, \ldots, R_3$ in the stop/go manner. The feedback polynomials of the registers are: $g_1(x) = 1 + x^{14} + x^{17} + x^{18} + x^{19}$, $g_2(x) = 1 + x^{21} + x^{22}$, $g_3(x) = 1 + x^8 + x^{21} + x^{22} + x^{23}$, $g_4(x) = 1 + x^{12} + x^{17}$. The function F is the majority function $F(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3$.

The communication in the GSM system is performed through frames. Each frame consists of 228 bits. For every frame to be enciphered, the initialization procedure takes place, that yields the initial state of the LFSRs on the basis of the 64-bit secret key K and the 22-bit frame number $\mathcal{F}$. During the initialization, the bits of the secret key are first imposed into all the LFSRs, at every clock pulse, without the stop/go clocking, starting from the LSB of each key byte. Then the bits of the frame number are imposed into all the LFSRs in the

*Instituto de Física Aplicada (CSIC), Serrano 144, 28006 Madrid, Spain

same way, starting from the LSB. Finally, the algorithm is run for 100 clock pulses utilizing the stop/go clocking, but producing no output.

Cryptanalytic attack: The attack consists of updating the system of linearized equations that relate the state variables of the LFSRs $R_1, \ldots, R_3$ with the output bits, on the basis of the clock-control sequence produced by the LFSR R_4, for its initial state picked from the set of 2^{17} possible states. The linearization of the equations is performed by substitution of the nonlinear terms by the new variables. Due to the frequent reinitializations, small number of skipped bits in the initialization process and the distribution of the feedback taps, many linearly dependent equations appear, and almost all the unknown output bits, that come after very few known output bits, can be reconstructed without solving the system at all.

For the analysis of the system, we start from the analysis of the rank of a matrix to which a random last row is added. Namely, we prove the following

Proposition 1 - Let $\mathbf{W} = [w_{i,j}]_{i=1,\ldots,m;j=1,\ldots,n}$ be a matrix over GF(2), whose rank is $r(\mathbf{W}) = m$. Let $\mathbf{U} = [u_{i,j}]_{i=1,\ldots,m+1;j=1,\ldots,n}$ be a matrix over GF(2), whose first m rows are respectively equal to the rows of $\mathbf{W}$, and the elements of the last row are generated independently at random, with the probability $Pr(u_{m+1,j} = 1) = 0.5$, $1 \leq j \leq n$. Then the probability that $r(\mathbf{U}) = r(\mathbf{W})$ is

$$Pr(r(\mathbf{U}) = r(\mathbf{W})) = 2^{m-n}. \tag{1}$$

Proof: The first m linearly independent rows of the matrix $\mathbf{U}$ span the vector space, whose cardinality is 2^m. The claim that $r(\mathbf{U}) = r(\mathbf{W})$ means that the last row of $\mathbf{U}$ must belong to the vector space spanned by the first m rows. Since $Pr(u_{m+1,j} = 1) = 0.5$, the required probability is 2^{m-n}. Q.E.D.

Due to the nonlinear order of the majority function, the maximum number of variables in the system will be $n = 719$. Consider now the process of adding equations to this system. Suppose that for some clock pulse c_t of the algorithm, the system consists of m linearly independent equations. If the contribution of

R_i to the new equation does not depend on k_{i_t} state variables, $i = 1, \ldots, 3$, then the number of equations that can be added to the system reduces at least from 2^{719} to 2^{719-d_t}, where $d_t = \sum_{i=1}^{3} \left(k_{i_t} + \binom{k_{i_t}}{2}\right)$. In such a way, the probability that the rank of the new system is the same as the rank of the previous system can be significantly greater than that for the equation generated at random.

In general, d_t depends on the initial state of the algorithm. Let us call a *dependency* the set of state variables on which a particular position of an LFSR depends. Due to the concentrations of the feedback taps of the LFSRs R_1 and R_2 to the right of the inputs to the majority function, the input positions to this function and the last positions of these LFSRs depend on very few initial state variables after every initialization. The cardinalities of the dependencies of these positions will grow much slower than in the case when the feedback taps are not concentrated at the ends (R_3). Thus, the total cardinalities of the dependencies for the LFSR R_1 and R_2 can be very small and the probability that the newly added equation will be linearly dependent on the others can become very close to one.

Let $\mathbf{AX} = \mathbf{B}$ be a linear system over GF(2), where $\mathbf{A} = [a_{i,j}]_{i=1,\ldots,m;j=1,\ldots,n}$, $\mathbf{X} = [x_i]_{i=1,\ldots,n}$, and $\mathbf{B} = [b_i]_{i=1,\ldots,m}$. Denote by $\mathbf{A}'$ the extended matrix of the system. Let us transform the matrix $\mathbf{A}'$ into the trapezoidal form using the Gauss algorithm. Denote the state of the matrix $\mathbf{A}'$ after the performing of the k-th set of such transformations by $\mathbf{A}'_k$. Thus, $\mathbf{A}' = \mathbf{A}'_0$. Let us define the matrix $\mathbf{P} = [p_{i,j}]_{i,j=1,\ldots,m}$ in the following way: $p_{0_{i,i}} = 1$, $p_{0_{i,j,i\neq j}} = 0$; at the k-th step of the transformation, if i-th and j-th rows of the matrix $\mathbf{A}'_k$ are interchanged or summed, so are the respective rows of $\mathbf{P}_k$. In such a way, the nonzero elements of the i-th row of the matrix $\mathbf{P}_k$, $i = 1, \ldots, m$ at the k-th step of the transformation, point to the ordinal numbers of the rows of the original system $\mathbf{A}'_0$, on which the i-th row of the matrix $\mathbf{A}_k$ linearly depends.

Let $\mathbf{A}'_m$ be the extended matrix of the system $\mathbf{AX} = \mathbf{B}$ in its trapezoidal form and suppose that the rank of this system is $r(\mathbf{A}'_m) = m$. Let us add the new equation $\mathbf{WX} = \mathbf{Z}$ to the system, where $\mathbf{W} = [w_i]_{i=1,\ldots,n}$, and $\mathbf{Z} = [z_1]$. Let

$*$ denote the operation of adding a row to a matrix. Apply the process of transformation to the trapezoidal form to the new system $\mathbf{CX} = \mathbf{D}$, where $\mathbf{C} = \mathbf{A}_m * \mathbf{W} = [c_{i,j}]_{i=1,\ldots,m+1;j=1,\ldots,n}$ and $\mathbf{D} = \mathbf{B}_m * \mathbf{Z} = [d_i]_{i=1,\ldots,m+1}$. Suppose $r(\mathbf{C}'_{m+1}) = m$ and denote by q the biggest row index for which $p_{m+1_{q,m+1}} = 1$.

If $q = m+1$ and z_1 is known, then $c'_{m+1_{q,n+1}} = 0$ and

$$z_1 = \sum_{i=1}^{m} b_i p_{m+1_{q,i}}. \tag{2}$$

If $q = m+1$ and z_1 is not known, we can guess z_1 and transform the matrix $\mathbf{C}'$ in the same way as if z_1 were known. Since $r(\mathbf{C}'_{m+1}) = m$, for the correct value of z_1, $c'_{m+1_{q,n+1}}$ must be zero. Thus, in this case, z_1 is also given by (2).

If $q < m+1$ and z_1 is known, then the following relation will obviously hold:

$$z_1 = c'_{m+1_{q,n+1}} + \sum_{i=1}^{m} b_i p_{m+1_{q,i}}. \tag{3}$$

But if $q < m+1$ and z_1 is not known, the relation (3) will still hold, but the calculated value for z_1 can be incorrect.

The degenerate cases when $q < m+1$ will be rare [4]. To guess the bits that cannot be reconstructed, the runs of these cases should be short. Experiments performed on a great number of frames show that approximately 70% of these runs are of length less than 10.

The attack on the A5/2 consists of the following major steps:

Input: 4 frames of the output sequence and their corresponding frame numbers; frame numbers of the output sequence frames to be reconstructed; threshold T chosen according to the actual bit error ratio in the channel;

Output: reconstructed frames of the output sequence, except of the bits that correspond to the degenerate cases and to the linearly independent equations.

1. `SET` $s = 0$; { `Ordinal number of the initial state of` R_4 }
   `SET` $i = 0$; { `Frame number index` }

SET $m = 0$; { The number of linearly independent equations }

2. Choose the s-th state of the LFSR R_4; SET $d = 0$;

3. SET $i = i + 1$; Complete the initialization process, starting from the state s of R_4, imposing the frame number $\mathcal{F}_i$ into all the LFSRs, and keeping track of the dependencies;

4. IF $d > T$ THEN SET $s = s + 1$, and go to Step 2; if the end of the frame is reached, then go to Step 3; otherwise, run the algorithm A5/2 for one cycle, keeping track of the dependencies, and setting the equation that relates these dependencies and the corresponding output bit;

5. Linearize the obtained equation, by substituting the nonlinear terms by the new variables; add this equation to the system;

6. Check the current system for its rank, updating the matrix $\mathbf{P}$; if the current rank is greater than the previous rank, then SET $m = m + 1$ and go to Step 4; if the current rank is equal to the previous rank and the current output bit is known, check whether the known bit is equal to the bit calculated by the relation (3); if not, then SET $d = d + 1$, return to the previous state of the system and go to Step 4; if the current rank is equal to the previous rank and the output bit is unknown, find the biggest q such that $p_{m+1_{q,m+1}} = 1$; IF $q = m + 1$, then calculate the unknown bit by the relation (2), return to the previous state of the system and go to Step 4; IF $q < m + 1$ then return to the previous state of the system and go to Step 4.

Our algorithm examines all the possible 2^{17} initial states of the LFSR R_4 in the worst case. For each such state and for all the checks after the first one, the system already has the trapezoidal form, except for the newly added last row. So, the complexity of these checks will be linear in m.

Acknowledgement

This work was supported by C.A.M., Spain, under grant 07T/0044/1998.

References

[1] http://cryptome.org/gsm-a512.htm, 1999.

[2] Biryukov A., Shamir A., Wagner D., 'Real Time Cryptanalysis of A5/1 on a PC', in Proceedings of Fast Software Encryption, New York, 2000, Lecture Notes in Computer Science, Berlin: Springer Verlag, in press.

[3] Golić J. Đ., 'Cryptanalysis of Alleged A5 Stream Cipher', in Advances in Cryptology - EUROCRYPT '97, Lecture Notes in Computer Science 1233, W. Fumy (ed.), Berlin: Springer-Verlag, 1997, pp. 239-255.

[4] Parker D. S., Dinh L., 'How to Eliminate Pivoting from Gaussian Elimination - by Randomizing Instead', Technical Report No. CSD-950022, Computer Science Department, University of California, Los Angeles, 1995.

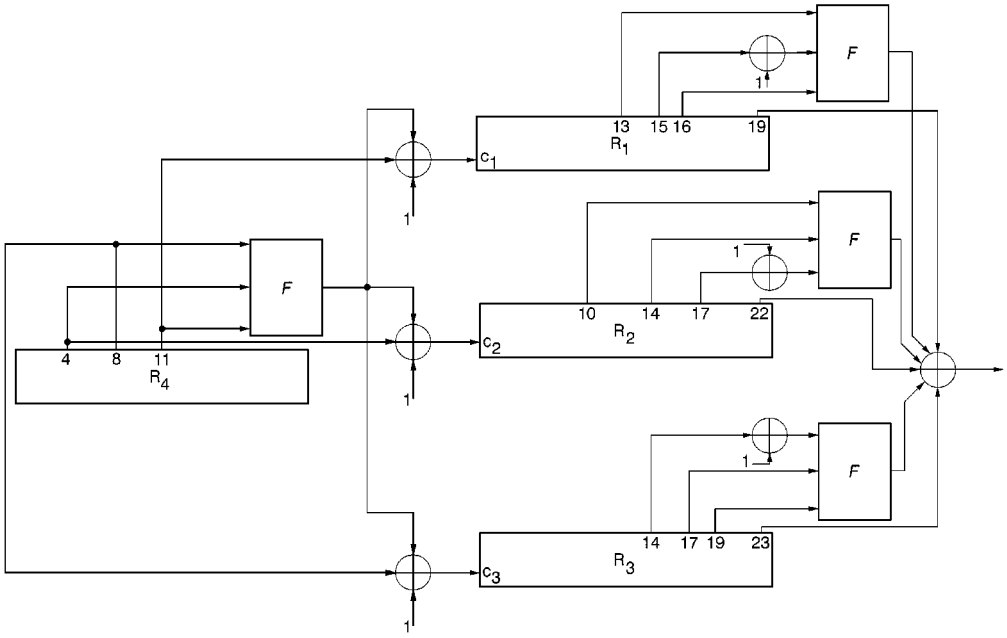

Fig. 1 - The scheme of the A5/2 algorithm