3 May 1998
Source: Hardcopy from Anonymous
Thanks to the authors and Springer

Add: On Random Mappings and Random Permutations
W. G. Chambers

Walter Fumy (Ed.)

Advances in Cryptology -- EUROCRYPT ' 97

International Conference on the Theory and Application of Cryptographic Techniques Konstanz, Germany, May 11-15, 1997 Proceedings

Springer

[Excerpt, Pages 239-255]

[Note: Please verify equations by reference to the originals]

239

Cryptanalysis of Alleged A5 Stream Cipher

Jovan Dj. Golic *

School of Electrical Engineering, University of Belgrade
Bulevar Revolucije 73, 11001 Beograd, Yugoslavia

Abstract. A binary stream cipher, known as A5, consisting of three short LFSRs of total length 64 that are mutually clocked in the stop/go manner is cryptanalyzed. It is allegedly used in the GSM standard for digital cellular mobile telephones. Very short keystream sequences are generated from different initial states obtained by combining a 64-bit secret session key and a known 22-bit public key. A basic divide-and-conquer attack recovering the unknown initial state from a known keystream sequence is first introduced. It exploits the specific clocking rule used and has average computational complexity around 2^{40}. A time-memory trade-off attack based on the birthday paradox which yields the unknown internal state at a known time for a known keystream sequence is then pointed out. The attack is successful if $T \cdot M \geq 2^{63.32}$ where T and M are the required computational time and memory (in 128-bit words), respectively. The precomputation time is $O(M)$ and the required number of known keystream sequences generated from different public keys is about T/102. For example, one can choose $T \simeq 2^{27.67}$ and $M \simeq 2^{35.65}$. To obtain the secret session key from the determined internal state, a so-called internal state reversion attack is proposed and analyzed by the theory of critical and subcritical branching processes. *["$\simeq$" here means approximately]*

Form SF298 Citation Data

Report Date *("DD MON YYYY")* 11051997	**Report Type** N/A	**Dates Covered (from... to)** *("DD MON YYYY")*

Title and Subtitle Cryptanalysis of Alleged A5 Stream Cipher	**Contract or Grant Number**
	Program Element Number
Authors	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) IATAC Information Assurance Technology Analysis Center 3190 Fairview Park Drive Falls Church VA 22042	**Performing Organization Number(s)**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Monitoring Agency Acronym**
	Monitoring Agency Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms	
Document Classification unclassified	**Classification of SF298** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** unlimited
Number of Pages 21	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	5/11/97	White Paper

4. TITLE AND SUBTITLE
Cryptanalysis of Alleged A5 Stream Cipher

5. FUNDING NUMBERS

6. AUTHOR(S)
Not provided

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Information Assurance
Technology Analysis Center
(IATAC)
3190 Fairview Park Drive
Falls Church, VA 22042

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Technical
Information Center
DTIC-AI
8725 John J. Kingman Road,
Suite 944

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE
A

13. ABSTRACT ***(Maximum 200 Words)***

The objective of this paper is to discuss the Theory and Application of Cryptographic Techniques used to develop crypt analytic attacks on A5 that can reconstruct the 64-bit secret key in the known plain text scenario with the computational complexity smaller than 264.

14. SUBJECT TERMS
CRYPT

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

1 Introduction

A common type of keystream generators for additive stream cipher applications consists of a number of possibly irregularly clocked linear feedback shift registers (LFSRs) that are combined by a function with or without memory. Standard cryptographic criteria such as a large period, a high linear complexity, and good statistical properties are thus relatively easily satisfied, see [12]. However, such a generator may in principle be vulnerable to various divide-and-conquer attacks in the known plaintext (or ciphertext-only) scenario, where the objective is to reconstruct the secret key controlled LFSR initial states from the known keystream sequence, for a survey see [12] and [5]. In practice, for resynchronization purposes, the internal state of a keystream generator is reinitialized once in

---

* This work was done while the author was with the Information Security Research Centre, Queensland University of Technology, Brisbane, Australia. Part of this work was carried out while the author was on leave at the Isaac Newton Institute for Mathematical Sciences, Cambridge, United Kingdom. This research was supported in part by the Science Fund of Serbia, grant #04M02, through the Mathematical Institute, Serbian Academy of Science and Arts.

---

a while by combining the same secret session key with different randomizing keys (typically transmitted in the clear and called here *public)* into the secret message keys defining different initial internal states. This may open new possibilities for the secret key recovery cryptanalytic attacks, see [3].

In this paper, a keystream generator consisting of three short binary LFSRs with known primitive feedback polynomials that are mutually clocked in the stop/go manner is cryptanalyzed. The LFSR lengths are 19, 22, and 23, respectively, and the total length is thus 64. Middle taps in each of the LFSRs are used to define the clock-control sequence, the clocking rule is such that at least two LFSRs are effectively clocked per each output bit, and the keystream sequence is formed as the bitwise sum of the three stop/go clocked LFSR sequences. The 64-bit long secret key is nonlinearly combined with a 22-bit long public key (frame number) to form the LFSR initial states. The first 100 output bits are discarded and the message length is only 114 bits (frequent resynchronization). However, the full-duplex communication mode makes the effective message length of 228 bits. The scheme along with the code has been made public in [1] and is allegedly used under the name A5 for stream cipher encryption in the GSM standard for digital cellular mobile telephones, see [13]. For simplicity, the name A5 is used here throughout. In a yet unpublished paper [14], it has been observed, perhaps surprisingly, that the period of the keystream sequence is only slightly bigger than the period, $\simeq 2^{23}$, of the longest LFSR. A possibility for a divide-and-conquer attack of average complexity 2^{40} has been mentioned in [1] and [13]. The attack would consist in guessing the initial states of the two shorter LFSRs and, then, in computing the longest LFSR sequence from the known keystream sequence. However, this attack can not work, because the clocking depends on the unknown longest LFSR sequence as well. In addition, one has to take care of the first 100 output bits being discarded as well.

Although one may in principle imagine that edit distance or edit probability correlation attacks [4] can be adapted to deal with stop/go clocking, such attacks are not likely to be successful on A5, because of a very short available keystream sequence. Due to the bitwise summation, to achieve a divide-and-conquer effect, one or two LFSRs have to be replaced by their linear models [7], where linear models of individual LFSRs can be based on the repetition property only, while linear models of pairs of the LFSRs must involve their feedback polynomials as well. Instead of the so-called shrunk feedback polynomials [7], we now have to introduce the expanded feedback polynomials. If the whole scheme is replaced by the corresponding linear model, one may then even conceive of a fast correlation attack framework similar to the one from [6], but the required keystream sequence

length would be much bigger than the one at disposal. On the other hand, the conditional correlation attack [11] based on the repetition property can not be extended to deal with A5, because of the specific clocking rule.

The objective of this paper is to develop cryptanalytic attacks on A5 that can reconstruct the 64-bit secret key in the known plaintext scenario with the computational complexity smaller than 264. In Section 2, a more detailed description of the A5 stream cipher is presented. It is shown that the known plaintext

---

241

attacks are very realistic in the GSM applications. In Section 3, a basic divide-and-conquer attack on A5 with the average computational complexity $2^{40.16}$ is introduced. It essentially consists in guessing some bits of the LFSR states, in recovering the others by solving appropriate linear equations, and in the LFSR states reversion via the unknown binary clocking sequences to obtain the LFSR initial states. The last step is needed since the first 100 output sequence bits are discarded. In Section 4, a time-memory trade-off attack based on the birthday paradox probabilistic argument is pointed out. This attack is feasible due to relatively short internal state size of 64 bits. It can recover the LFSR internal states for a particular keystream sequence at a particular time and is successful if $T \cdot M \geq 2^{63.32}$, where T and M are the required computational time and memory, respectively. The precomputation time is $O(M)$ and a sample of T/102 228-bit long observed keystream sequences generated from the same secret session key and different public keys is needed. To obtain the secret key, a low-complexity internal state reversion attack is then proposed in Section 5. It consists in the reversion of the LFSR internal states, first when the output sequence is known, then when the output sequence is unknown, and finally when the secret key is nonlinearly combined with the known public key. The complexity of the attack is analyzed by the theory of critical and subcritical branching processes, briefly outlined in the Appendix. Conclusions are given in Section 6.

2 Description of the Stream Cipher

The stream cipher algorithm to be defined is for simplicity called A5 according to [1], [13]. The A5 type keystream generator considered is shown in Fig. 1.

Fig. 1. Alleged A5 type keystream generator

Let $f_i(z) = \text{Sigma}^{<r_i}_{>l=0}\, f_{i,l}\, z^l$ denote a known binary primitive feedback polynomial of LFSR_i of length r_i, $i = 1$, 2, 3, and let $r_1 = 19$, $r_2 = 22$, and $r_3 = 23$. The feedback polynomials specified in [1], [13] are sparse, but our

cryptanalytic methods to be presented do not depend on their choice. Let $S_i(0) = (x_i(t))^{<r_i-1>}{}_{t=0}$ denote the initial state of LFSR$_i$ and let $x_i = (x_i(t)^{oo}{}_{t=0}$ denote the corresponding maximum-length sequence of period 2^{r_i} -1 produced by LFSR$_i$ via the linear recursion $x_i(t) = \text{Sigma}^{r_i}{}_{l=1}\, f_{i,t}\, x_i(t - l)$, $t \geq tau_i$.

Let $S_i(t) = s_{i,l}(t))^{r_i}{}_{l=1}$ denote the state of LFSRi at time $t \geq 0$ in a scheme with stop/go clocking to be defined below, and let tau_i denote a middle tap from LFSR$_i$ used for clock-control. The values suggested in [1] are tau_1 = 10, tau_2 = 11, and tau_3 = 2. Then the clock-control sequence $C = (C(t))^{oo}{}_{t=1}$ is defined by

$$C(t) = g(s_{1,tau1}(t - 1) + s_{2,tau2}(t - 1) + s_{3,tau3}(t - 1) \qquad (1)$$

where g is a 4-valued majority function of three binary variables such that $g(s_1, s_2, s_3) + \{i, j\}$ if $s_i = s_j \neq s_k$ for $i < j$ and $k \neq i, j$, and $g(s_1, s_2, s_3) = \{1, 2, 3\}$ if $s_1 = s_2 = s_3$. The clock-control value *C(t)* defines which LFSRs are clocked to produce an output bit *y(t)* as the sum

$$y(t) = s_{1,l}(t) + s_{2,l}(t) + s_{3,l}(t), \quad t \geq 1. \qquad (2)$$

Let $c_i = (c_i(t))^{oo}{}_{t=1}$ denote the binary clocking sequence for LFSR$_i$ (it is clocked if $c_i(t) = 1$ and not clocked if $c_i(t) = 0$) which is derived from the clock-control sequence C in an obvious way. Equation (2) can formally be used to generate the initial bit $y(0)$ from $S(0)$, so that $y = (y(t))^{oo}{}_{t=0}$ is called the output sequence. The first 100 output bits, $(y(t))^{100}{}_{t=1}$, are discarded, the following 114 bits are used as the keystream for one direction of communication in the full-duplex mode, then the next 100 bits are again discarded, and the following 114 bits are used as the keystream for the reverse direction of communication. The encrypted messages are thus very short and the resynchronization is frequent.

The LFSR initial states are defined in terms of the secret and public keys. The public key is a known 22-bit frame number generated by a counter and hence different for every new message. The 64-bit secret session key is first loaded into the LFSRs (the all-zero initial state is avoided by setting the output of the last stage to 1) and the 22-bit public key is then bitwise added into the feedback path of each of the LFSRs that are mutually clocked as above. More precisely, if $p = (p(t))^0{}_{t=-21}$ denotes the public key, then for every $-21 \leq t \leq 0$, the LFSRs are first stop/go clocked as before and, then, the bit *p(t)* is added to the last stage of each of the LFSRs. The LFSR states after these 22 steps, as a secret message key, represent the initial LFSR states for the keystream generation.

The A5 stream cipher is allegedly used to encrypt the links between individual cellular mobile telephone users and the base station in the GSM system, see [13]. Therefore, if two users want to communicate to each other via their base station(s), the same messages get encrypted twice which makes the known plaintext cryptanalytic attack possible, provided a cooperative insider user can be established. Note also that the links between the base stations are not encrypted. For

---

any user, a 64-bit secret session key is generated by another algorithm from the secret master key specific to the user and a public random 128-bit key transmitted in the clear from the base station to the user. So, a possible reconstruction of one or more session keys for a user opens a door for a cryptanalytic attack on the master key of that user.

3 Basic Attack

The objective of a divide-and-conquer attack to be presented in this section is to determine the LFSR initial states from a known keystream sequence corresponding to only one known plaintext-ciphertext pair. In fact, only about 64 known successive keystream bits are required. Let $S(t) = (S_1(t), S_2(t), S_3(t))$ denote the whole internal state of A5 at time $t \geq 0$, where $S(0)$ is the initial internal state defined by the secret message key. The known keystream sequence is in fact composed of two segments $(y(t))^{214}_{t=101}$ and $(y(t))^{428}_{t=315}$. The first goal is to reconstruct the internal state $S(101)$ and the second one is to determine $S(0) = (S_1(0), S_2(0), S_3(0))$ from $S(101)$.

Recall that $c_i = (c_i(t))^{\infty}_{t=1}$ denotes the binary clocking sequence for $LFSR_i$, which is clocked if $c_i(t) = 1$ and not clocked if $c_i(t) = 0$. If A_i denotes the state-transition matrix of regularly clocked $LFSR_i$, then

with the integer summation in the exponent. Also, let $\underline{x}_i = (\underline{x}_i(t))^{\infty}_{t=0}$ denote the stop/go clocked LFSRi sequence, where $\underline{x}_i(t) = s_{i,1}(t)$. In the probabilistic analysis to follow, a sequence of independent uniformly distributed random variables, over any finite set, is called purely random. As usual, we keep the same notation for random variables and their values.

Proposition 1. *Assume that the three regularly clocked LFSR sequences are mutually independent and purely random. Then the 4-valued clock-control sequence C is purely random and, hence, the binary clocking sequence ci is a sequence of independent identically distributed binary random variables with the probability of zero being equal to* 1/4.

Proposition 2. Assume that the three regularly clocked LFSR *sequences are mutually independent and purely random. Then the bitwise sum of any two or more stop/go clocked sequences* $\underline{x}_i$ *is purely random.*

It is shown in Section 5 that the state-transition function of A5 is not one-to-one, so that the set of all reachable internal states at time t, $t \geq 1$, is a subset of the set S_0 of all 2^{64} initial states. In particular, only $5 \cdot 2^{61} \simeq 2^{63.32}$ internal states are reachable for $t = 1$. As a consequence, different initial states can give rise to the same internal state at some time in future or even to the same output sequence too. This is explained in terms of the theory of branching processes in Section 5. More precisely, the number of different initial states giving rise to

---

the same internal state at some time in future is very likely linear in that time and, therefore, relatively small for the times of interest (internal state reversion when the output is not known, Subsection 5.1). On the other hand, the number of different initial states yielding the same internal state at some time in future and the same output sequence is very likely to be a very small integer (internal state reversion when the output is known, Subsection 5.2). In addition, since the individual LFSR sequences are maximum-length sequences with good (low)

autocorrelation and crosscorrelation properties and the combining function is maximum-order correlation immune, it is highly likely that different output sequences $y = (y(t))^{\infty}_{t=0}$ are different on the first successive 64 positions, $(y(t))^{63}_{t=0}$.

Consequently, it takes about 64 successive keystream bits to check if an assumed preceding internal state is consistent with the subsequent output sequence. The expected number of solutions for $S(101)$ is with high probability a small integer, whereas the the number of solutions for $S(0)$ (equivalent initial states) is very likely to be relatively small.

3.1 Internal state reconstruction

Let $S(101)$ be the internal state to be determined in the first stage of the attack. Since the number of reachable states $S(101)$ is not bigger than $2^{63.32}$ and the unreachable ones can be simply characterized by a set of linear equations, in the average complexity analysis given below we can simply take 63.32 instead of 64. For every $i = 1, 2, 3$, first guess n bits $(s_{i,l}(101))^{r_i + n-1}_{l=taui}$ if $n \leq r_i - tau_i + 1$,and, if not, then also guess the next $n - r_i + tau_i - 1$ bits produced by the linear recursion from $S_i(101)$. In any case, one thus obtains $3n$ linearly independent equations for unknown bits of $S(101)$, provided that $n \leq 19$. Since the assumed bits on average define $4n/3$ elements of the clock-control sequence, one can thus form $1 + 4n/3$ additional linear equations, where the first one is clearly obtained from the first keystream bit $y(101)$ without using the clock-control sequence. The additional linear equations are mutually linearly independent, provided that $n \leq 18$, because each one then contains at least two new bits that have not appeared before. They are linearly independent of the first $3n$ equations if and only if each of them contains at least one new bit that is not already guessed. This happens with high probability if

$$n < \max(tau_1, tau_2, tau_3) - 1. \qquad (4)$$

If not, then the last among the additional equations will necessarily involve some of the already guessed bits and will with high probability be linearly dependent on the first $3n$ equations. Suppose first that the condition (4) *is* satisfied. Then all the obtained linear equations are with high probability linearly independent, so that the internal state can be determined uniquely if $1 + 3n + 4n/3 \geq 63.32$, that is, if $n \geq 14.38$ (it follows that $\max(tau_1, tau_2, tau_3) \geq 16$). The obtained state should then be tested for correctness on additional $3n$ keystream bits on average The computational complexity is then $\simeq 2^{43.15}$ and the total required keystream

---

sequence length is about 64 successive bits (we keep the fractions since we deal with the average case complexity).

Suppose now that $\max(tau_1, tau_2, tau_3) \leq 15$, which means that the condition (4) is not satisfied, as is the case in the particular proposal from [1], where $\max(tau_1, tau_2, tau_3) = tau_3 = 12$. In this case, the last of the additional equations are with high probability linearly dependent and as such can not be used as before, but can be used to test the linear consistency of the initial guess. If the previous analysis was extended, then one would get that n has to be bigger than 14.38 and that the average complexity would hence increase, contrary to the intuition. Indeed, one can do better than that. Let initially $n = 10$, so that (4) is satisfied. One thus obtains the total of $1 + 3n + 4n/3 \simeq 44.3$ linearly independent equations on average. Now, instead of guessing the next $m \simeq 19.02/3$ bits on average in each of the LFSR sequences, we will build a tree structure to sequentially store all the possibilities for the next bits that are consistent with the additional linear equations. In each node of the tree one stores the

next three input bits to the majority clock-control function such that the resulting clocking is consistent with the equations. This approach is in spirit similar to the inversion attack [8] on nonlinear filter generators. The average number of branches leaving each node would have been $3/4 \cdot 4 + 1/4 \cdot 8 = 5$ if it were not for the additional equations. They on average reduce this number to 2.5. The required depth of the tree should on average be $4m/3$ to obtain the next m guessed bits in each of the LFSR sequences. So, instead of 2^{3m} possibilities for the next m bits, we have to check only $2.5^{4m/3} \simeq 2^{1.76m} \simeq 2^{11.16}$ possibilities on average, under the reasonable independence assumption valid for the so-called supercritical branching processes, see Theorem 6 from the Appendix. The overall complexity is then $2^{30+11.16} \simeq 2^{41.16}$. For comparison, suppose that the clock-control bits are used to produce the output, that is, $tau_1 = tau_2 = tau_3 = 1$. Then, clearly, only the part of the process involving the tree applies and the overall complexity is minimum possible, that is, $2^{1.76 \cdot 63.32/3} \simeq 2^{37.15}$.

To get the average number of trials needed to find the correct internal state $S(101)$, one should in fact divide by two the complexity figures given above, e.g., $2^{41.16}$ thus reduces to $2^{40.16}$.

3.2 Internal state reversion via clocking sequences

In the second stage, our objective is to recover the initial LFSR states from $S(101)$. In view of (3), this can be done by guessing the number of ones in individual binary clocking sequences, that is, the number of clocks needed to get $S_i(101)$ from $S_i(0)$, for each $i = 1, 2, 3$. According to Proposition 1, the underlying probability distribution is binomial with the average number of clocks $0.75 \cdot 101 \simeq 76$ and the standard deviation $0.25 \cdot$ square root $303 \simeq 4.35$ for each of the LFSR sequences. If the search is organized in order of decreasing probabilities for each of the LFSR sequences independently, the number of trials required to find the correct numbers of clocks is with high probability not bigger than about 10^4 and is at worst about 10^6. For each guess, one first recovers $S_i(0)$ from $S_i(101)$ by backward linear recursion, for each $i = 1, 2, 3$, and then tests

---

the guess by running the keystream generator forwards to obtain $S(101)$. Note that multiple solutions for $S(0)$, if they exist, are all obtained by checking all $\simeq 10^6$ possibilities for the clocking sequences, for any possible $S(101)$ obtained in the first stage. This number can clearly be reduced by assuming the mutually constrained rather than independent clocking sequences for individual LFSRs. In any case, reconstructing the initial state $S(0)$ from $S(101)$ is much faster than obtaining $S(101)$ itself.

4 Time-Memory Trade-Off Attack

As was already explained in the previous section, the first 64 successive output bits of A5, $(y(t))^{63}_{t=0}$, represent a vectorial boolean function of 64 initial state bits $S(0)$ such that the number of different initial states $S(0)$ producing the same 64-bit initial output block is in most cases only 1 or a very small integer. In fact, since the initial 101 output bits are not used for the keystream, the initial state bits $S(0)$ should be confined to the $2^{63.32}$ values achievable by $S(1)$ which are easily characterized. As a consequence, for any observed 64 successive keystream bits, one can find all the preceding internal states yielding these bits either by exhaustive search over all reachable internal states requiring $2^{63.32}$ 64-bit computations and bitwise comparisons or by only one table lookup requiring $2^{63.32}$ 64-bit words of memory to store the inverse of the vectorial boolean function considered. The inverse function, with multiple preimages if they exist, is found and stored in $2^{63.32}$ precomputation time. Let the time and memory required in these two extreme cases be denoted as $T = 2^{63.32}$, $M = 1$ and $T = 1$, $M = 2^{63.32}$, respectively. Is any meaningful time-memory trade-off based on the birthday paradox possible?

Assume that,the objective is to recover the preceding internal states for any observed 64 successive keystream bits in the known plaintext scenario. Each known keystream sequence of effective length 228 bits provides 102 $\simeq 2^{6.67}$ 64-bit blocks, and, due to the very small keystream sequence length, it is very likely that the cryptanalyst knows either all 228 bits or none of them. So, any time-memory trade-off solely based on these 102 keystream blocks is meaningless. However, we may consider a sample of all the keystream sequences corresponding to different initial states (secret message keys) derived from K (at most 2^{22}) different known public keys and a single secret session key. The reconstruction of any internal state corresponding to a particular public key is then meaningful if $K < 2^{22}$ and if it leads to the recovery of the secret session key, which can then be used to decrypt the ciphertexts obtained from the remaining public keys.

Let the cryptanalyst form a table of M possibly multiple 64-bit words defining the reachable initial states corresponding to a random sample of M different 64-bit output blocks, and let the table be then sorted out with respect to the output blocks, which are also stored. Multiple preimages are all obtained by the internal state reversion given a known output, in $O(M)$ time, see Subsection 5.2. The required precomputation time for sorting is M log M or, approximately, just M if the logarithmic factor, smaller than 64, is neglected. Altogether, the required

---

247

precomputation time is thus $O(M)$. *By* the standard birthday paradox (used in meet-in-the-middle attacks), it then follows that with high probability at least one of the $102 \cdot K$ keystream blocks in the observed sample will coincide with one of the output blocks used to form the table if

$$102 \cdot K \cdot M \geq 2^{63.32} \qquad (5)$$

where a small multiplicative constant is neglected for simplicity. The time T needed to find such a keystream block is $102 \cdot K \log M$ or simply $102 \cdot K$ neglecting the logarithmic factor. Then only one table lookup gives the desired internal state(s). So, the time-memory trade-off is possible with $T \cdot M \geq 2^{63.32}$ and $T < 102 \cdot 2^{22}$. For example, if $K = 2^{15,}$ then the time and memory required are $T \simeq 2^{21.67}$ and $M \simeq 2^{41.65}$ (in 128-bit words), respectively, and the precomputation time is $O(M)$. In an extreme case, when $K = 2^{21}$, we get $T \simeq 2^{27.67}$ and $M \simeq 2^{35.65} \simeq 862$ Gbytes, but the secret session key to be determined can then only be used to decrypt ciphertexts obtained from the remaining half of the public keys.

A more general approach for the cryptanalyst would be to analyze the traffic corresponding to L different sessions for each out of N users. This increases the sample size (and time) to $102 \cdot K \cdot L \cdot N$, so that further reduction in M is possible, which makes the attack quite realistic. In this case, a particular user whose secret session key is to be determined is not known in advance. This, of course, does not make a difference if the objective is cloning rather than decryption. Even more generally, one may also allow that K be maximum possible, 2^{22}, if the cryptanalyst is capable of attacking the algorithm that combines the secret master key of a user and a public random 128-bit key into the secret session key. Namely, the determined session key may be useless for decryption, but may be used for the secret master key reconstruction with devastating consequences regarding both decryption and cloning.

The time-memory trade-off attack described clearly applies to arbitrary keystream generators, and is feasible in the case of A5 because of its relatively short memory size of only 64 bits. It yields an internal state of A5 at a known time and is meaningful when coupled with a cryptanalytic attack to be introduced in the next section which gives all the candidates for the secret session key. If the internal state is determined at time $101 \leq t \leq 151$, then the attack consists in the reversion of the internal state to $S(101)$ based on known output, then to $S(0)$ when

the output is not known (due to the first 100 output bits discarded), and finally to the secret session key when the known public key is incorporated. If the internal state is determined at time $315 \leq t \leq 365$, then the attack consists in the reversion of the internal state to $S(315)$ based on known output, then to $S(214)$ when the output is not known, and the rest is the same as in the first case with $S(214)$ as the internal state. Note that possible multiple solutions are all obtained. Multiple candidates for the secret session key are then easily reduced to only one, correct solution by comparing a small number of already known keystream sequences with the ones generated from the assumed candidates and known public keys.

248

5 Internal State Reversion via Branching

The objective of the internal state reversion attack to be described in this section is to find all the secret session keys that combined with a known public key give rise to a given internal state at a known time. All the internal states at a known time that are consistent with a known keystream sequence can be obtained either by the basic internal state reconstruction attack from Subsection 3.1 or by the time-memory trade-off attack from Section 4.

The performance of the attack is analyzed by the theory of critical and subcritical branching processes and its time and space complexities are thus shown to be both small. Extensive computer experiments on nonlinear filter generators regarding the so-called generalized inversion attack [9] (where the whole internal state is recovered starting from its finite input memory part in a way similar to the internal state reversion) show that the size of the generated search trees can be well described by the theory of branching processes.

5.1 Unknown output

Given an internal state $S(t)$ at time t, $t \geq 1$, $S(t) \in S_0$ the objective of the reversion attack when the output sequence is not known is to determine all the internal states $S(t')$ at a given previous time $t' < t$ that produce $S(t)$ at time t by the state-transition function, whereas the output sequence is not considered at all. For the reversion to work, the state-transition function, F, must be easily computable in the reverse direction. Letting F^{-1} denote the reverse state-transition function, $F^{-1}(S(t))$ denotes the set of all $S(t - 1)$ such that $F(S(t - 1) = S(t)$. The reversion attack then consists in the recursive computation of the reverse state-transition function starting from $S(t)$ and up to $S(t')$. The internal states obtained can all be stored as nodes in a tree with $t - t' + 1$ levels where the initial level, $n = O$, has one initial node representing $S(t)$, and the level n, $1 \leq n \leq t - t'$ contains the nodes representing all possible $S(t - n)$ giving rise to $S(t)$. The end nodes thus give all the desired internal states $S(t')$. The main problem here is to estimate the size of the trees arising from a random $S(t)$, that is, the number of the nodes obtained at each level n if $S(t)$ is chosen uniformly at random, and especially if n is not small.

The state-transition function of A5-is essentially determined by the clock-control sequence, see (1) and (3). Accordingly, the number of different states $S(t - 1)$ in $F^{-1}(S(t))$ is derived by backward clocking from all the possibilities for $C(t - 1)$ and hence only depends on the following six bits: the three bits $(s_{1,tau1}(t), s_{2,tau2}(t), s_{3,tau3}(t))$ which define the clock-control sequence at the current time t, $C(t)$, and the three preceding bits in the regularly clocked LFSR sequences which, if $\min(tau_1, tau_2, tau_3) \geq 2$, all belong to $S(t)$ and are given as $(s_{1,tau1 - 1}(t), s_{2,tau2 - 1}(t), s_{3,tau3 -1}(t))$. Denote these bits by s_1, s_2, s_3 and s'_1, s'_2, s'_3, respectively.

Proposition 3. *Let (i, j, k) denote a permutation of* (1, 2, 3) . *Then the following six events can occur:*

249

- *A : for any k, if* $s'_i = s'_j \neq s'_k = s_k$, *then* $C(t-1) = \{i,j\}$

- *B : for any k, if* $s'_i = s'_j \neq s'_k \neq s_k$, *then* $C(t-1)$ can take no values

- *C : if* $s'_1 = s'_2 \neq s'_3 = s_1 = s_2 = s_3$, *then* $C(t-1) = \{1, 2, 3\}$

- *D : if* $s'_1 = s'_2 \neq s'_3 \neq s_1 = s_2 = s_3$, *then* $C(t-1)$ can take every of the four values {1, 2}, {1,3}, {2,3}, and {1, 2, 3}

- *E : for any k, if* $s'_1 = s'_2 \neq s'_3 = s_i = s_j \neq s_k$, *then* $C(t-1)$ can take every of the two values $\{i, j\}$ and {1, 2, 3}

- *F : for any i, if* $s'_1 = s'_2 \neq s'_3 = s_i \neq s_j = s_k$, *then* $C(t-1)$ can take every of the three values $\{i, j\}$, $\{i, k\}$ and {1, 2, 3}

Proposition 4. *If an internal state* $S(t)$ *is randomly chosen from* S_0 *according to uniform distribution, then the number of solutions for* $S(t-1)$ *is a nonnegative integer random variable Z with the probability distribution*

$$\Pr\{Z = 0\} = 3/8, \Pr\{Z = 1\} = 13/32,$$

$$\Pr\{Z = 2\} = \Pr\{Z = 3\} = 3/32, \Pr\{Z = 4\} = 1/32. \qquad (6)$$

It follows that the state-transition function of A5 is not one-to-one and that the fraction of the internal states from S_0 not reachable in one step is 3/8 (they are simply characterized by a set of three linear equations). Let $\{S(t-n)\}$ denote the set of all the internal states/nodes at level n in the tree spanned by the reversion from a given $S(t)$, and let $Z_n = |\{S(t-n)\}|$ and $Y_n = \text{Sigma}^n_{l=1} Z_l$. Both the time and space complexities of the reversion attack are determined by Y_n. Our objective now is to estimate how large Z_n and Y_n can be when $S(t)$ is randomly chosen. Of course, each particular $S(t)$ uniquely determines the tree (model **M'**), and if we assume that regularly clocked LFSR sequences are mutually independent and purely random (model **M**), then the tree is random rather than unique even when $S(t)$ is fixed. From the internal state reversion via the clocking sequences, Subsection 3.2, we know that in both the models $Z_n \leq n^3$ necessarily holds. The trees spanned in both the models are expected to be similar if the depth n *is* smaller than 4/3 of the period of the shortest LFSR, $\simeq 4/3\ 2^{19}$, which is when on average the LFSR sequences start to repeat themselves in model **M'**.

Proposition 4 shows that the associated Galton-Watson *branching process,* described in the Appendix, has the branching probability distribution defined by (6), with the expected value and variance $\mu = 1$ and $sigma^2 = 9/8$, respectively. The branching process is *critical.* The random trees produced by model **M** and by the associated branching process are not exactly the same, as random variables. The reason for this is that in the branching process the branching probability distribution for a given node is independent of the nodes at the same or the preceding levels (the history), whereas in model **M** there is a weak dependence between the nodes as a result of different internal states having some clock-control bits in common. This weak dependence affects the expected values and variances of both Z_n and Y_n, but insignificantly.

250

Consequently, if $S(t)$ is uniformly distributed over S_0, then in model **M**, in view of Theorem 6 from the Appendix, $E(Z_n) \simeq 1$, $\mathrm{Var}(Z_n) \simeq sigma^2 n$, and $\Pr\{Z_n > 0\} \simeq 2/(sigma^2 n)$, where $\mathrm{sigma}^2 = 9/8$. So, the fraction of the internal states reachable in n steps is about $2/(sigma^2 n)$. On the other hand, both the computational time and the storage required for the reversion attack are determined by the total number of nodes Y_n. Theorem 7 from the Appendix then gives that $E(Y_n) \simeq n$ and $\mathrm{Var}(Y_n) \simeq sigma^2 n^3/3$. In view of the Chebyshev's inequality $\Pr\{|Y_n - E(Y_n)| > e\} \leq \mathrm{Var}(Y_n)/e^2$, we then get that the total number of nodes Y_n is with high probability $O(n$ square root $n)$ and the multiplicative constant is not big. Note that in the case of interest, $n = 101$, and the approximations are expected to be very good.

It is also interesting to see how large Z_n and Y_n can grow when conditioned on the event that the internal state $S(t)$ is reachable in n steps. We know that at least one such state results from both the basic internal state reconstruction attack and the time-memory trade-off attack. Theorem 8 from the Appendix yields that in this case $E(Z_n|Z_n > 0) \simeq sigma^2 n/2$ and $\mathrm{Var}(Z_n|Z_n > 0) \simeq sigma^4 n^2/4$. This means that the number of solutions for $S(t - n)$is with high probability linear in n, provided at least one such solution exists. As for the total number of nodes Y_n, we noted in the Appendix that $E(Y_n|Z_n > 0) = O(\mathrm{sigma}^2 n^2)$ and $\mathrm{Var}(Y_n|Z_n > 0) = O(sigma^2 n^4)$, so that Y_n is then with high probability $O(sigma^2 n^2)$. So, for $n = 101$ both the time and space complexities are small, although somewhat bigger than in the case of a uniformly distributed $S(t)$.

The number, N, of starting internal states in the real reversion attack may be bigger than just one, but is still small, as will be shown in the following subsection. The time complexity clearly increases proportionally with N, whereas the space complexity, determined as the maximum tree size over all the starting states, increases only logarithmically with N, due to the exponential probability distribution (21) in Theorem 8 from the Appendix.

5.2 Known output

Given an internal state $S(t)$ at time t, $t \geq 1$, $S(t) \in S_0$, the objective of the reversion attack when the output sequence is known is to determine all the internal states $S(t')$ at a given previous time $t' < t$ that produce $S(t)$ at time t by the state-transition function as well as the known output sequence $(y(t - l))_{l=1}^{t-t'}$. This reversion attack then goes along the same lines as the one when the output sequence is not known, with a difference that from each level in the tree spanned, the nodes whose internal states produce the output bits different from the one known are all removed. The size of the resulting tree is hence much smaller.

The output bit produced from $S(t - 1)$ at time $t - 1$ depends on the following six bits: $(s_{1,1}(t), s_{2,1}(t), s_{3,1}(t))$ and the preceding three bits in the regularly clocked LFSR sequences. They are denoted as z_1, z_2, z_3 and z'_1, z'_2, z'_3, respectively. The produced output bit is then equal to $z'_i + z'_j + z_k$ if $C(t - 1) = \{i, j\}$, for any $\{i, j\}$ (as usual, (i, j, k) is a permutation of $(1, 2, 3)$), and to $z'_1 + z'_2 + z'_3$ if $C(t - 1) = \{1, 2, 3\}$. An analog of Proposition 3 can then be established, with a difference that in each of the given six events, $C(t - 1)$ can take every specified value for which, in addition, the produced output bit coincides with the one

---

known, $y(t - 1)$. If $\min(tau_1, tau_2, tau_3) \geq 3$, one can then derive the following analog of Proposition 4.

Proposition 5. *If an internal state S(t) is randomly chosen from S_0 according to uniform distribution, then the number of solutions for S(t - 1) is a nonnegative integer random variable Z with the probability distribution*

$$\Pr\{Z = 0\} = 315/512,\ \Pr\{Z = 1\} = 75/256,\ \Pr\{Z = 2\} = 9/128,$$

$$\Pr\{Z = 3\} = 5/256,\ \Pr\{Z = 4\} = 1/512. \qquad (7)$$

The probabilistic models **M** and **M'** are defined in the same way as before, with a difference that the known output sequence is assumed to be either fixed or purely random and independent of the LFSR sequences. The dependence between the nodes in the trees produced by **M** and **M'**, although still relatively weak, is stronger than before due to the six additional bits controlling the output. The associated branching process is now *subcritical* with $\mu = 1/2$ and $sigma^2 = 17/32$. The results regarding the probability distribution, the expected values, and the variances for the random variables Z_n and Y_n are then obtained analogously, by applying the parts of Theorems 6-8 from the Appendix relating to subcritical branching processes. Consequently, we get that in model **M**, $E(Z_n) \simeq 2^{-n}$, $\mathrm{Var}(Z_n) \simeq sigma^2 2^{-(n-2)}$, and $\Pr\{Z_n > 0\} \simeq c2^{-n}$, where c is a positive constant that is obtained numerically as $c = \lim_{n>\infty} 2^n (1 - f^{(n)}(0)) \simeq 0.63036$, where $f^{(n)}$ is the self-composition of the generating function f of the probability distribution defined by (7), see the Appendix. Also, $E(Y_n) \simeq 1$ and $\mathrm{Var}(Y_n) \simeq 8\ sigma^2$.

Conditioning on the event that the starting internal state is reachable in n steps, we get $E(Z_n|Z_n > 0) \simeq 1/c \simeq 1.586$, $\mathrm{Var}(Z_n|Z_n > 0) \simeq 4\ sigma^2/c - 1/c2 \simeq 0.854$, $E(Y_n|Z_n > 0) = O(n)$, and $\mathrm{Var}(Y_n|Z_n > 0) = O(n^2)$. The size Y_n of the resulting tree is then $O(n)$ with high probability. In the case of interest, resulting from the time-memory trade-off attack, we have that $n \leq 50$, so that the obtained trees are very small, whereas the number of possible solutions for $S(t - n)$ ($S(101)$) is with high probability only 1 or a very small positive integer.

5.3 Secret key reconstruction

Our goal now is to obtain all possible secret session keys from all the determined initial states $S(0)$ given a known public key $p = (p(t))^0_{t=-21}$. Recall that the secret session key is in fact an internal state of the initialization scheme, which works in the same way as the keystream generator A5, except that the public key is bitwise added, in 22 steps, into the feedback path of each of the LFSRs. Given an initial state $S(0)$, $S(0) \in S_0$, the objective of the secret key reconstruction attack is to determine all the internal states $S(t')$ at the previous time $t' = -22$ that produce $S(0)$ by the modified state-transition function $S(t) = F_0(S(t - 1), p(t))$, $-21 \leq t \leq 0$, which also depends on the known public key sequence

---

p. The modified reverse state-transition function $F^{-1}_0(S(t), p(t))$ then consists of two stages: first, the bit $p(t)$ is added to the last stage of each of the LFSRs and, second, the LFSRs are clocked backwards according to all possible values $C(t - 1)$ for the clock-control sequence.

It is readily seen that the secret key reconstruction can be achieved by the reversion attack when the output sequence is not known in which the reverse state-transition function is modified according to the public key p as explained above. Consequently, both the analysis based on the theory of critical branching processes and the conclusions derived remain valid for the secret key reconstruction attack. Since now $n = 22$ instead of $n = 101$, the trees spanned are much smaller in size. Multiple solutions for the secret session key $S(-22)$ giving rise to the

same $S(0)$ are still possible, but their number is relatively small. All the resulting candidates for the secret session key are consistent with the used keystream sequence. These multiple candidates for the secret session key are then easily reduced to only one, correct solution by comparing a small number of already known keystream sequences with the ones generated from the assumed candidates and known public keys.

6 Conclusions

Several cryptanalytic attacks on a binary stream cipher known as A5 are proposed and analyzed. The objective of the attacks is to reconstruct the 64-bit secret session key from one or several known keystream sequences produced by different 22-bit (randomizing) public keys, in the known plaintext scenario which is shown to be very realistic in the GSM applications. A basic divide-and-conquer attack with the average computational complexity $2^{40.16}$ and negligible memory requirements is first introduced. It requires only about 64 known successive keystream bits and gives all possible LFSR initial states consistent with a known keystream sequence. A time-memory trade-off attack based on the birthday paradox is then pointed out. The objective of the attack is to find the LFSR internal states at a known time for a known keystream sequence corresponding to a known public key. The attack is feasible as the internal state size of A5 is only 64 bits.

To obtain the secret session key from the determined LFSR internal states, an internal state reversion attack is proposed and analyzed by the theory of critical and subcritical branching processes. It is shown that there typically exist multiple, but not numerous, candidates for the secret session key that are all consistent with the used keystream sequence. The unique, correct solution is then found by checking on a small number of additional keystream sequences. The secret session key recovered can be used to decrypt the ciphertexts obtained from the remaining public keys and, possibly, to mount a cryptanalytic attack on the secret master key of the user as well.

A simple way of increasing the security level of the A5 stream cipher with respect to the cryptanalytic attacks introduced is to make the internal memory size larger. For example, doubling the memory size, from 64 to 128 bits, is very

---

253

likely to push the attacks beyond the current technological limits. Note that the secret session key size need not be increased to 128 bits. In addition, one can make the clock-control dependent on more than just a single bit in each of the shift registers by using a balanced nonlinear filter function applied to each of them individually. The inputs to the filter functions should be spread over the shift register lengths, respectively, and their outputs can be combined in the same way as in A5. This increases the complexity of the basic internal state reconstruction attack.

Appendix

Branching processes

The so-called Galton-Watson process, see [10], [2], is a Markov chain $\{Z_n\}^{\infty}_{n=0}$ on the nonnegative integers whose transition function is defined in terms of a given probability distribution $\{p_k\}^{\infty}_{k=0}$. The initial random variable Z_0 takes value 1 with probability 1, and for any $n \geq 1$, the random variable Z_n conditioned on $Z_{n-1} = i$ is the sum of i independent identically distributed random variables with the probability distribution $\{p_k\}^{\infty}_{k=0}$. The process can be regarded as a random (finite or infinite) tree with Z_n being the number of nodes at level $n \geq 0$ where the number of branches leaving any node in the tree is equal to k with probability p_k, independently of

other nodes at the same or previous levels. The generating function characterizing the probability distribution of Z_n can be expressed as the self-composition of the generating function $f(s) = \text{Sigma}^{oo}_{k=0}\, p_k s^k$ of $\{p_k\}^{oo}_{k=0}$, which is the probablility distribution of Z_1. Precisely, if $f^{(n)}(s)$, $0 \leq s \leq 1$, denotes the generating function of the probability distribution of Z_n and if $f^{(0)} = s$, then for every $n \geq 1$, $f^{(n)}(s) = f(f^{(n-1)}(s))$.

The basic characteristic of a branching process is the expected number of branches leaving any node, that is,

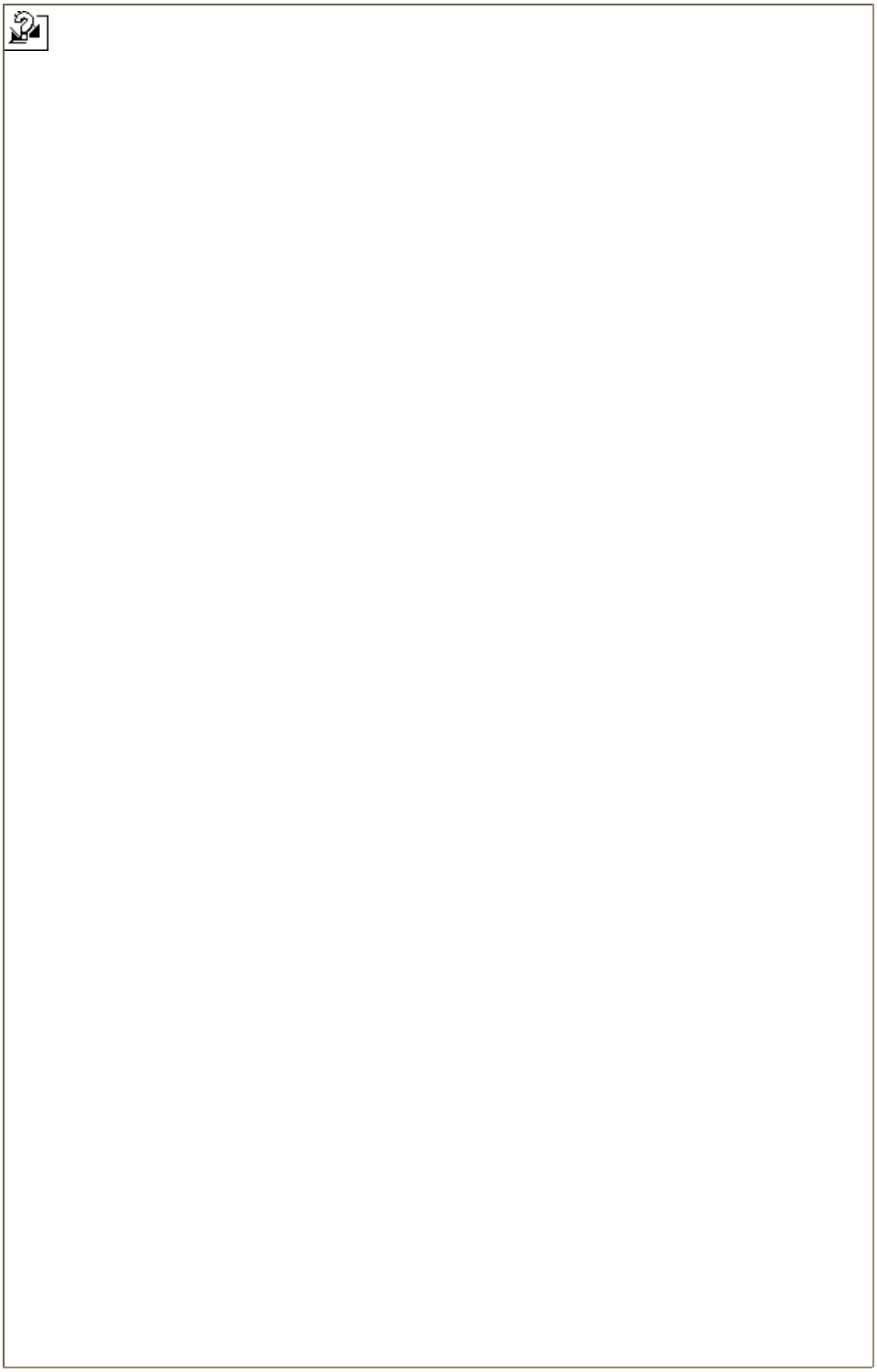
$$\mu = E(Z_1) = \text{Sigma}^{oo}_{k=0}\, kp_k. \qquad (8)$$

A branching process is called subcritical, critical, or supercritical if $\mu < 1$, $\mu = 1$, or $\mu > 1$, respectively. The extinction probability defined as the probability of a tree being finite is 1 for subcritical and critical (provided $p_0 > 0$) processes and smaller than 1 for supercritical processes. We are here only interested in subcritical and critical processes, whose main properties are given by the following theorem, see [2], [10]. Let $sigma^2 = \text{Var}(Z_1)$ be the variance of Z_1.

Theorem 6. *In the subcritical case, $\mu < 1$, for any $n \geq 1$,*

$$E(Z_n) = \mu^n \qquad (9)$$

$$\text{Var}(Z_n) = sigma^2 \mu^{n-1}\, 1\text{-}\mu^n / 1 - \mu \qquad (10)$$



255

$$E(Z_n|Z_n > 0) \sim sigma^2/2\ n \qquad (22)$$

$$\mathrm{Var}(Z_n|Z_n > 0) \sim sigma^4/4\ n^{2.} \qquad (23)$$

The probability distribution of the conditioned random variable $Y_n|\{Z_n > 0\}$ is not treated in the standard books on branching processes like [10] and [2]. Nevertheless, the previous theorems and the results regarding the conditioned random variable $Z_n|\{Z_{n+k} > 0\}$ presented in [2] lead us to conclude that in the subcritical case, $E(Y_n|Z_n > 0) = O(n)$ and $\mathrm{Var}(Y_n|Z_n > 0) = O(n^2)$, whereas in the critical case, $E(Y_n|Z_n > 0) = O(sigma^2 n^2)$ and $\mathrm{Var}(Y_n|Z_n > 0) = O(sigma^4 n^4)$.

References

1. R. J. Anderson, Internet communication.

2. K. B. Athreya and P. E. Ney, *Branching Processes.* Berlin: Springer-Verlag, 1972.

3. J. Daemen, R. Govaerts, and J. Vandewalle, "Resynchronization weakness in synchronous stream ciphers," Advances in Cryptology - EUROCRYPT '92, *Lecture Notes in Computer Science*, vol. 765, T. Helleseth ed., Springer-Verlag, pp. 159-167, 1994.

4. J. Dj. Golic and M. J. Mihaljevic, "A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance," *Journal of Cryptology, vol.* 3(3), pp. 201-212, 1991.

5. J. Dj. Golic, "On the security of shift register based keystream generators," Fast Software Encryption - Cambridge '93, *Lecture Notes in Computer Science,* vol. 809, R. J. Anderson ed., Springer-Verlag, pp. 90-100, 1994.

6. J. Dj. Golic, "Towards fast correlation attacks on irregularly clocked shift registers," Advances in Cryptology - EUROCRYPT '95, *Lecture Notes in Computer Science,* vol. 921, L. C. Guillou and J.-J. Quisquater eds., Springer-Verlag, pp. 248-262, 1995.

7. J. Dj. Golic, "Linear models for keystream generators," *IEEE Trans. Computers,* vol. C-45, pp. 41-49, Jan. 1996.

8. J. Dj. Golic, " On the security of nonlinear filter generators," Fast Software Encryption - Cambridge '96, *Lecture Notes in Computer Science,* vol. 1039, D. Gollmann ed., Springer-Verlag, pp. 173-188, 1996.

9. J. Dj. Golic, A. Clark, and E. Dawson, "Generalized inversion attack on nonlinear filter generators," submitted.

10. T. H. Harris, *The Theory of Branching Processes.* Berlin: Springer-Verlag, 1963.

11. R. Menicocci, "Cryptanalysis of a two-stage Gollmann cascade generator," *Proceedings of SPRC* '93, Rome, Italy, pp. 62-69, 1993.

12. R. A. Rueppel, "Stream ciphers," *Contemporary Cryptology: The Science of Information Integrity,* G. Simmons ed., pp. 65-134. New York: IEEE Press, 1991.

13. B. Schneier, *Applied Cryptography.* New York: Wiley, 1996.

14. S. Shepherd and W. Chambers, private communication.

[End]

Bart Preneel (Ed.)

Fast Software Encryption

**Second International Workshop
Leuven, Belgium, December 14-16, 1994
Proceedings**

Springer

[Excerpt, Pages ?, 26-27]

[? page number]

On Random Mappings and Random Permutations

W. G. Chambers

Department of Electronic and Electrical Engineering,
King's College London, Strand, London WC2R 2LS, UK

w.chambers@kcl.ac.uk

---

1 Introduction

Much work has been done by many people, including the present author, to prove that certain classes of sequence-generator have guaranteed periods [1]. Here we examine what happens at the other extreme, with sequence generators which are finite-state machines modelled as having random next-state tables. The advantages are a lack of mathematical structure which might provide an entry for the cryptanalyst, and a huge choice of possibilities; the disadvantages are that there are no guarantees on anything, and as is well known there is a risk of getting a very short period.

Thus we consider a finite-state machine whose state is specified by an integer in the range 0, ...N - 1, and which has a next-state function F which specifies the n + 1-th state s_{n+1} as $F(s_n)$ with s_n the n-th state. The output can also be regarded as a function of the state, but we are not so much concerned with this. Evidently the function F can be represented by a look-up table with addresses and entries in the range 0, ...N - 1. Each function F corresponds to a state-diagram where the states correspond to N points 0, ...N - 1, with point i (the predecessor) joined to point j (the successor) by an arrowed line if $F(i) = j$. Points lying on a loop or *cycle* are called cyclic points; a point i satisfying $F(i) = i$ is regarded as lying on a cycle of length 1. Every point has a unique successor, but some points have no predecessors and some have more than one predecessor. In general the state diagram will consist of a number of cycles, plus a number of directed trees rooted on cyclic points; in these trees the arrows are pointing to the root.

If the next-state function F is invertible, so that for every j in 0, ...N - 1 there is an i satisfying $F(i) = j$, then the

function will be called a permutation (an *N*-permutation), and the state-diagram will consist of a number of cycles without any trees rooted in them.

There are altogether N^N *N*-functions, and *N*! *N*-permutations. Typically the state is represented by a number of bits, say *n*, so that we have $N = 2_n$. Normally *n* would be of the order of hundreds.

There is a considerable literature on this topic [2], [3], [4], [5], [6], [7].

[Unknown parts not provided]

26

as they run. There are certain practical objections to such a scheme of course, for instance if the generator needs to be restarted frequently or if random access to the key-stream is needed, but if one simply needs a long key-stream without any restarts such a method is feasible and of course it makes it harder for the cryptanalyst by presenting him as it were with a moving target.

If we have such a scheme then the internal description of the machine-state must include the look-up table. Now the point is the following: if the transformation from one state to the next is invertible, then we have a permutation for the next-state function, but otherwise we may be dealing with a general finitestate machine with its likelihood of much smaller loop sizes. Thus it would seem that the modification of the look-up tables should be carried out in a way that enables one to undo the transformation in a unique manner.

Thus as a simple example consider the following random-number generator proposed by Bob Jenkins and publicised on the Usenet service. The state variables are

```
a, b: Unsigned 32-bit integers
m[O] ..m[255]: a lookup table of unsigned 32-bit integers
p: a counter cycling from 0to 255
```

Initially *p* is set to zero and the other variables to random values. There are two internal unsigned 32-bit integer variables, *x* and *y*. Addition is carried out "modulo 2^{32}". We loop indefinitely on the following instructions:

```
x = m [p];
y = m[RS(x)]; /* RS(x) is a right-shift of x by 24, leaving an 8-bit result */
a = R(a); /* R() is a rotation left by 27 bits */
a = a + y; /* addition is mod 2^32 */
m[p] = a + b; /***/ /* see below */
output(b+y); /* put the value b+y on the output stream */
b = x;
p = (p + 1) mod 256; /* Step p */
```

The instruction marked /***/ evidently changes the lookup table. It is not hard to see that if we know the state variables at the end of this loop we could determine them at the start, including the value of the modified table-entry; thus we have what is in effect a permutation. The same thing would apply if we replaced /***/ by *m[p] = m[p] + a + b, or m[p] = mtp] XOR (a + b)*, but not if we replaced /***/ by *m[p] = m[p] + a*, as we could not then deduce the initial
value of b from the final values of the state variables.

8 A Cipher Claimed to Resemble A5

Two other encryption algorithms have recently been publicised on the Internet, without much theoretical backing. The first is "alleged RC4", which has similarities to the algorithm just described. Here the next state function is invertible.

27

The second (announced by Ross Anderson and Michael Roe) is purported to be very similar to the A5 cipher used in the GSM mobile telephone system [11]. It uses three binary linear feedback shift-registers with known (key-independent) primitive polynomials of degrees 19, 22 and 23 respectively. These registers are initially set in a key-dependent manner to non-zero values, and on each iteration they are stepped as follows: A control-bit is taken from a known position near the centre of each register. If two or three of the control bits are equal to 1, then the registers producing these bits are stepped. On the other hand if two or three of the control bits are 0, then the registers producing *these* bits are stepped. In effect the registers are mutually clock-controlled in a stop/go fashion, and it is easy to see that there is a probability of 3/4 that a register is stepped on any iteration of the algorithm. Thus the longest register would be expected to go through a complete cycle in roughly $P = (2^{23} - 1) \cdot 4/3$ iterations.

Regarded as a finite-state machine the system has just under $2^{19+22+23} = 2^{64}$ states, and the next-state function is non-invertible. Thus we would expect to find eventual periodicities of the order of 2^{32}, after a precursor sequence of the same order of length. Instead, in a search for eventual periodicities the author found 237 cases (all distinct), and all of them had periods very close to small multiples of $P = (2^{23} -1) \cdot 4/3$; moreover just over 40% of these cases had periods very close to P itself. (The precursor sequences were on the whole a little longer, of lengths something like $10P$ on average.) Evidently the shorter registers, one with a period very close to $P/16$ and the other with a period very close to $P/2$, are locking on to the period of the longest register, with respectively 16 cycles and 2 cycles for every cycle of the longest register. Further investigations have shown that this "lock-in" is a robust phenomenon, occurring independently of the choice of primitive polynomials, and even occurring if the three sequences of control bits are chosen as random periodic sequences with periods equal to numbers of the form $2^n - 1$. This topic is being pursued further.

The shortness of the period is probably not a hazard in normal use [11], where only a few hundred bits of output are required between key-changes, but perhaps one would be advised not to use this scheme as a random-number generator without further study.

9 Concluding Remarks

Encryption algorithms in which the look-up tables are continuously modified have the attractions of high speed and of making the analyst's task harder, but there may be a lingering doubt about the period, particularly when there is no significant theory available. The above discussion strongly suggests that the nextstate function should be invertible, but one might like some further reassurance. The use of a "rekeyed" cipher is a good way of obtaining a guaranteed huge period. Here there is "driving" sequence generator D whose output is used to modify the state of a second generator G which provides the final output. The generator D has a known or lower-bounded huge value for its period. Thus a 32-bit cascade generator with a cascade of length 8, as suggested in [1], will have

[Balance of article not provided]

[End]

HTML by JYA/Urban Deadline

See references to A5 cryptanalysis by Ross Anderson, Michael Roe, Bruce Schneier and Simon Shepherd: http://jya.com/crack-A5.htm

In particular, these classified papers by Simon Shepherd:

S J Shepherd, "Cryptanalysis of the GSM A5 Cipher Algorithm", IEE Colloquium on Security and Cryptography Applications to Radio Systems, Digest No. 1994/141, Savoy Place, London, 3 June 1994, (COMMERCIAL-IN-CONFIDENCE).

S J Shepherd, "An Approach to the Cryptanalysis of Mobile Stream Ciphers", IEE Colloquium on Security and Cryptography Applications to Radio Systems, Digest No. 1994/141, Savoy Place, London, 3 June 1994, (COMMERCIAL-IN-CONFIDENCE).
