

Comments on “Theory and Applications of Cellular Automata in Cryptography”

S.R. Blackburn*, S. Murphy† and K.G. Paterson‡

Information Security Group, Royal Holloway,
University of London, Surrey TW20 0EX, U.K.

June 26, 1995

Abstract — The cipher systems based on Cellular Automata proposed by Nandi *et al.* [3] are affine and are insecure.

Index Terms — Cryptography, block ciphers, stream ciphers, cellular automata, affine group.

*This author’s research supported by EPSRC Research Grant No. GR/H23719.

†This author acknowledges the support of the Nuffield Foundation

‡This author’s research supported by a Lloyd’s of London Tercentenary Foundation Research Fellowship.

In [3], the authors present cryptographic transformations based on Cellular Automata. These transformations are used to define block ciphers and stream ciphers. It is claimed that the cryptographic transformations generate the alternating group and that the cryptosystems are secure. Both of these claims are incorrect.

The systems in [3] are based on permutations of the vector space V_N of dimension N over $GF(2)$. These permutations are obtained from Cellular Automata. The specific permutations used (called *fundamental transformations* in [3]) are in fact affine. Thus the group generated by these permutations is a subgroup of the Affine group on V_N and not the Alternating group A_{V_N} as claimed. The affine group is a small subgroup of the alternating group. In any case, knowing that the group generated by the cryptographic transformations contains the alternating group gives no guarantee of security (see [2]).

The various ciphers proposed are cryptographically weak as they depend on affine transformations. We examine each in turn.

1 The Block Cipher

We first note that there appears to be no mechanism for changing the key as the authors propose that it is stored in ROM. The ability to change the key is essential for any cipher. Our cryptanalysis is independent of this observation.

The block cipher proposed in [3] uses p different enciphering functions E_0, \dots, E_{p-1} . Plaintext block $i + jp$ ($0 \leq i < p$) is enciphered using E_i . Each E_i is an affine cipher, as it is a composition of affine transformations. $(N + 1)$ chosen plaintext-ciphertext pairs encrypted under E_i suffice to deduce E_i , see [1]. In any case, the probability of being able to deduce E_i given k known plaintext-ciphertext pairs is at least $1 - 2^{N+1-k}$. For example, given $2N$ plaintext-ciphertext pairs encrypted under E_i , we can deduce E_i with probability at least $1 - 2^{-(N-1)}$. Thus this block cipher is insecure against both chosen and known plaintext attacks.

2 The Stream Ciphers

Two stream ciphers are proposed in [3]. Both produce L bits each time they are clocked. Four consecutive bits are extracted and XORed with a block of four bits of the plaintext, to produce the ciphertext. We cryptanalyse each of the stream ciphers in turn.

2.1 PCA with ROM Generator

The PCA with ROM generator consists of l invertible $L \times L$ binary matrices R_0, \dots, R_{l-1} stored in ROM. Let V be an L dimensional binary vector space. The state of the generator $v(t) := (v_1(t), \dots, v_L(t))$ at time t is an element of V . At time $(t + 1)$, the new state of the generator is defined by $v(t + 1) = R_{t'}v(t)$, where $t' = t \bmod l$. At time t , the generator outputs the four bits $v_h(t), v_{h+1}(t), v_{h+2}(t)$ and $v_{h+3}(t)$ where h is time independent.

This generator is insecure as decimations of its output sequence have low linear complexity. For example, consider the sequence $w = w(0), w(1), \dots$ defined by

$$w(j) := v_h(jl).$$

This sequence is v_h decimated by l . If we set $A = \prod_{i=0}^{l-1} R_i$, so

$$v((j + 1)l) = Av(jl),$$

and set π to be the linear map from V to $GF(2)$ mapping (v_1, \dots, v_L) to v_h , then we may write

$$w(j) = \pi A^j v(0).$$

Suppose A has characteristic polynomial $c(X) := \sum_{i=0}^L c_i X^i$. By the Cayley–Hamilton theorem $c(A) = 0$, so

$$\sum_{i=0}^L c_i w(j + i) = \sum_{i=0}^L c_i \pi A^{j+i} v(0) = \pi A^j c(A) v(0) = 0.$$

Hence w has linear complexity at most L . A similar argument shows that any decimation by l of any of the four sequences produced by the generator has linear complexity at most L . Thus the keystream consists of an interleaving of $4l$ sequences, each of linear complexity at most L . By using the Berlekamp–Massey algorithm on $2L$ bits of each of the $4l$ decimated sequences, the keystream generator can therefore be determined in complexity $O(L^2l)$. Since both l and L are small (the values $L = 8$, $l = 16$ are suggested in [3]), the generator is insecure.

2.2 The Two Stage PCA Generator

The second generator consists of two cellular automata PCA_1 and PCA_2 . PCA_2 has a fixed next state function and is regularly clocked. Its output is used to control the next state function of PCA_1 .

We can define the automaton PCA_2 as follows. Its current state $k(t) = (k_1(t), k_2(t), \dots, k_L(t))$ is a binary vector of length L . The next state $k(t+1)$ is $M(x_1, \dots, x_L)k(t)$, where the elements $x_i \in GF(2)$ are part of the key and $M(x_1, \dots, x_L)$ is the $L \times L$ matrix defined as follows. Let P be the $L \times L$ binary matrix having ones on its super- and sub-diagonals and zeros

elsewhere, so

$$P_{ij} = \begin{cases} 1 & \text{when } j = i \pm 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then $M(x_1, \dots, x_L)$ is the matrix $P + \text{diag}(x_1, \dots, x_L)$.

The state of PCA_1 at time t is a binary vector

$$v(t) := (v_1(t), \dots, v_L(t)).$$

Its next state $v(t + 1)$ is defined by

$$v(t + 1) = M(k(t))v(t).$$

Thus the output of PCA_2 controls the next state function of PCA_1 .

The generator outputs the four bits $(v_h(t), v_{h+1}(t), v_{h+2}(t), v_{h+3}(t))$ at time t , where h is time independent. The generator's key consists of the initial states $v(0)$ of PCA_1 and $k(0)$ of PCA_2 , together with the elements $x_1, \dots, x_L \in \{0, 1\}$ which control the next state function of PCA_2 .

We cryptanalyse the generator as follows. We assume we have a segment of the keystream (a small multiple of L bits should suffice). Firstly, we guess h . There are at most four choices for h in the system proposed in [3] and we try each in turn.

We then note that

$$v_{h+1}(t + 1) = v_h(t) \oplus v_{h+2}(t) \oplus k_{h+1}(t)v_{h+1}(t). \quad (1)$$

If we know all the output terms appearing in this equation, we can deduce $k_{h+1}(t)$ whenever $v_{h+1}(t) \neq 0$. The correlation given by this equation is a serious weakness of this type of cipher, whatever the size of L . For the purposes of this cryptanalysis, we assume that $v_{h+1}(t)$ is an approximately balanced sequence. If this is not the case, the output sequence is very weak, and is easily analysed. Using our knowledge of the keystream, we can calculate about half the bits in a segment of the sequence

$$k_{h+1}(0), k_{h+1}(1), \dots .$$

This sequence has linear complexity at most L , since it is generated by PCA_2 which has a linear next state function and output function. We may therefore recover all the sequence $k_{h+1}(0), k_{h+1}(1), \dots$ using standard techniques (see for example [1, pages 209-214]). Similarly, by examining the output sequences $v_{h+1}(t), v_{h+2}(t)$ and $v_{h+3}(t)$ we can recover the sequence $k_{h+2}(0), k_{h+2}(1), \dots$. We have obtained two of the output sequences for the generator PCA_2 .

We next guess values $x'_1, \dots, x'_L \in \{0, 1\}$ for x_1, \dots, x_L . There are 2^L possibilities and testing each in turn constitutes the main work needed in breaking the system. When $L = 16$, as suggested, this number of guesses is very manageable. For each guess x'_1, \dots, x'_L , we construct a system of $4L$ linear equations in $L - 2$ variables which correspond to the unknown bits of the initial state. If ρ is the linear map from V to $GF(2)^2$ which

maps (w_1, \dots, w_L) to (w_{h+1}, w_{h+2}) , then the $(2i)$ th and $(2i + 1)$ th equations correspond to the vector equation

$$\rho M(x'_1, \dots, x'_L)^i = (k_{h+1}(i), k_{h+2}(i)).$$

If these linear equations are inconsistent, our guess x'_1, \dots, x'_L is wrong. This consistency test rules out nearly all possibilities for x_1, \dots, x_L . If, however, the equations are consistent, they produce a unique value for the initial state.

Assuming that a particular choice of x'_1, \dots, x'_L has passed the test above, we then check if this choice is consistent with our knowledge of the output of PCA_1 . We do this as follows. Suppose that our guess is correct. In this case, we know the initial state $k(0)$ of PCA_2 , so we can generate the next state matrix of PCA_1 at any time t . We can use this information, together with our knowledge of the output of PCA_1 , to reconstruct the initial state $v(0)$ of PCA_1 by solving a set of linear equations in $L - 2$ variables. However, for an incorrect guess the linear equations we produce are likely to be inconsistent. We can generate more equations by using more cipher output bits. The probability of inconsistency then converges to one geometrically fast.

In summary, by checking each possibility for x_1, \dots, x_L for consistency with the known outputs of PCA_2 and then with the known outputs of PCA_1 , we can find — with very high probability — the next state function of PCA_2 . In accomplishing this, we have derived $k(0)$ and can easily calculate $v(0)$. We

have therefore found the complete key using a small amount of keystream and $4.2^L = 2^{L+2}$ trials. This compares with the value of approximately 2^{3L} trials given in [3]. Moreover, the correlation expressed by equation (1) makes the generator even more insecure than we have described. Since the suggested value of L is 16 [3], this shows that the two stage PCA generator as proposed is insecure.

References

- [1] H. Beker and F. Piper, *Cipher Systems* (Van Nostrand, London, 1982).
- [2] S. Murphy, K. Paterson and P. Wild, 'A Weak Cipher that Generates the Symmetric Group', *Journal of Cryptology* Vol. 7, 1994, pp61-65.
- [3] S. Nandi, B.K. Kar and P. Pal Chaudhuri, 'Theory and Applications of Cellular Automata in Cryptography', *IEEE Trans. Comp.* Vol. 43, No. 12 December 1994, pp. 1346–1357.