PROBABILISTIC RELATIONS FOR THE SOLITAIRE KEYSTREAM GENERATOR

Marina Pudovkina
Moscow Engineering PhysicsInstitute
maripa@online.ru

Abstract: Stream ciphers are often used in applications where high speed and low delay are a requirement. The Solitaire keystream generator was developed by B. Schneier as a paper-and-pencil cipher. Solitaire gets its security from the inherent randomness in a shuffled deck of cards. In this paper we present probabilistic relations for the Solitaire keystream generator and describe their properties.

Keywords: Solitaire. Probabilistic relations.

1. INTRODUCTION

Many keystream generators proposed in the literature consist of a number of possibly clocked linear feedback shift registers (LFSEs) that are combined by a function with or without memory. LFSR-based generators are often hardware oriented and for a variety of them it is known how to achieve desired cryptographical properties [6]. For software implementation, a few keystream generators have been designed which are not based on shift registers. Such generators with mixing next-state functions are RC4[7], IA, IBAA, ISAAC [8], SCOP [9].

The Solitaire keystream generator was developed by B. Schneier [1] as a paper-and-pencil cipher. Solitaire gets its security from the inherent randomness in a shuffled deck of cards. By manipulating this deck, a communicant can create a string of "random" letters that he then combines with his message. Solitaire can be simulated on a computer, but it is designed

to be implemented by hand. It was designed to be secure even against the most well–funded military adversaries with the biggest computers and the smartest cryptanalysts. It's not fast, though it can take an evening to encrypt or decrypt a reasonably long message.

Solitaire is an output–feedback mode stream cipher. The next–state function F is a composition of four transformations $F = F_4 F_3 F_2 F_1$ which permute of elements of a deck.

In [2] is considered cycle structure of Solitaire. It is proved that Solitaire is not reversible and described all irreversible states. In [3] are analyzed properties of the key scheduling algorithm which derives the initial state from a variable size key, and described weaknesses of this process. One of these weaknesses is the existence of large classes of equivalent keys.

In this paper we present probabilistic relations for the Solitaire keystream generator and stress some their properties. These relations describe the jokers location in a deck at any time t. We show that the number of elements between the jokers at time t depends on t and the initial number of elements between the A joker and the B joker.

The paper is organized in the following way. In section 2 we describe the Solitaire cipher. In section 3 we consider probabilistic relations for the next-state function and in section 4 we give them for the key scheduling algorithm. We conclude in section 5 .

2. DESCRIPTION OF THE SOLITAIRE CIPHER

Solitaire is in fact a family of algorithms indexed by parameter n, which is a positive integer. Let m be a cardinality of an alphabet of a plaintext, then n=2m+2. The internal state of Solitaire at time t consists of a table $S_t=\{s_t[0],\ldots,s_t[n-1]\}$ of n values. S is a permutation of integers between zero and n-1.

B. Schneier takes n=54, m=26.

The next–state function F is a composition of four transformations $F = F_4 F_3 F_2 F_1$, which correspond to items 1–4 of the description given in [1]. The transformations F_4, F_3, F_2, F_1 permute elements of a table $S=(s[0],\ldots,s[n-1])$.

Let one joker A=n-2 and the other B=n-1.

The next-state function F

1. The transformation $F_1: S_i \rightarrow X=(x[0],\ldots,x[n-1])$. Let s[j]=n–1(A). If j≠ n–1 then move the A joker one element down: x[j]=s[j+1], xfi+l]=n-I, and x[k]=s[k], k=O…n–1, k≠j, j+1. If j=n-I move it just below s[0]: x[0]=s[0], x[l]=A, x[2]=s[1],…, x[k]=s[k-I],…,x[n-l]=s[n-2].

2. The transformation F_2: $X \rightarrow Y=(y[0],\ldots,y[n-1])$. Let x[j]=n-2(B). If $j \neq n-1$ and n–2 then move the B joker two elements down: y[j]=x[j+1], **y[j+1]=x[j+2],** y[j+2]=n-2 (B). If j=n-1, move the B joker just below x[1]. If j=n-2, move it just below x[0].
3. The transformation F_3: $Y \rightarrow Z=(z[0],\ldots,z[n-1])$. Perform a triple cut. That is, swap the elements above the first joker with the elements below the second joker. "First" and "second" jokers refer to whatever joker is nearest to, and furthest from, the top of the deck. Ignore the "A" and "B" designations for this step. The jokers and the elements between them don't move; the other elements move around them.
4. The transformation F_4: $Z \rightarrow S_{i+1}$. Perform a count cut. Let z[n-1]=k. Swap the elements z[0],...,z[k] with the elements z[k+1],..., z[n-2]. The element z[n-1] does not swap. A deck with a joker as z[n-1] will remain unchanged by this step.

The output function f

Let $s_{i+1}[0]=q$.

If $s_{i+1}[0]=A$ or $s_{i+1}[0]=B$ then we have not an output element.

If $s_{i+1}[0] \neq A, B$ then the output element $k_i=s_{i+1}[q] \pmod m$.

Let $M=m_1 m_2 \ldots m_L$ be a plaintext and $C=c_1 c_2 \ldots c_L$ be a ciphertext.

Encryption:

$$c_i=(m_i+k_i) \pmod m.$$

Decryption:

$$k_i=(c_i-m_i) \pmod m.$$

Key Scheduling Algorithm

Key is an initial deck ordering. A passphrase is used to order the deck. This method uses the Solitaire algorithm to create an initial deck ordering. Both the sender and receiver share a passphrase. (For example, "SECRET KEY.") Start with the deck in a fixed order; (0, 1, 2, . . ., n-3, A, B). Perform the Solitaire operation, but instead of Step 4, do another count cut based on the first character of the passphrase. In other words, do step 4 a second time, using the character of the passphrase as the cut number instead of the last card.

Repeat the four steps of the Solitaire algorithm once for each character of the passphrase. That is, the second time through the Solitaire steps use the second character of the passphrase, the third time through use the third character, etc. Use the final two characters to set the positions of the jokers.

3. PROBABILISTIC RELATIONS FOR THE NEXT-STATE FUNCTION

In this section we describe probabilistic relations and their properties for the next-state function. Let d_A be the number of elements before the A joker and d_B be the number of elements before the B joker. We shall say that d_A is called the A joker distance and d_B is called the B joker distance.

THEOREM 1
Let $d_A(j)$ be the A joker distance at time j and $d_B(j)$ be the B joker distance at time j. Let $S_j=(F)^j(S_0)$ and $k_j=s_j[n-1]$. If for any $t<j$: $d_B[t]\neq n-1$, $d_B[t]\neq n-2$, $d_A[t]\neq n-1$, $d_B[t]\neq d_A[t]+1$ and $d_A[t]\neq d_B[t]+1$, then the A joker distance and the B joker distance satisfy the following relations.

1. If $j=2i$ then

$$d_A(2i)=[d_A(0)+\sum_{j=0}^{i-1}(k_{2j+1}-k_{2j+2})-i] \pmod{n-1}$$

$$d_B(2i)=[d_B(0)+\sum_{j=0}^{i-1}(k_{2j+1}-k_{2j+2})+i] \pmod{n-1} \quad (1)$$

2. If $j=2i+1$ then

$$d_A(2i+1)=[-d_B(0)+\sum_{j=0}^{i-1}(k_{2j+2}-k_{2j+1})-(1+k_{2i+1})\pmod{n}-i-2] \pmod{n-1}$$

$$d_B(2i+1)=[-d_A(0)+\sum_{j=0}^{i-1}(k_{2j+2}-k_{2j+1})+i-1-(1+k_{2i+1})\pmod{n}] \pmod{n-1}. \quad (2)$$

Proof.
We conduct the proof by induction. Let d'_A be the A joker distance in a permutation Y and d'_B be the B joker distance in a permutation Y Let d''_A be the A joker distance in a permutation Z and d''_B be the B joker distance in a permutation Z. Consider $j=2k+1$.

Let us remark that

$$d'_A(2k+1)=d_A(2k)+1,$$

$$d'_B(2k+1)=d_B(2k)+2.$$

Recall that $F_2\ F_1(S_{2k}) = Y$ and $F_3(Y) = Z$. Note that the distances in permutations S_{2k}, Y, and Z satisfy the following relations.

$$d''_A(2k+1)=n-1-d'_B(2k+1)=n-3-d_B(2k),$$

$$d''_B(2k+1)=n-1-d'_A(2k+1)=n-2-d_A(2k).$$

The distances in permutations S_{2k}, Y, Z and S_{2k+1} satisfy the following relations.

$$d_A(2k+1)=[d''_A(2k+1)-(k_{2k+1}+1) \pmod n] \pmod{n-1}=[n-1-d'_B(2k+1)-(k_{2k+1}+1) \pmod n] \pmod{n-1}= [-d'_B(2k+1)-(k_{2k+1}+1)] \pmod{n-1}= [-d_B(2k)-2-(k_{2k+1}+1)] \pmod{n-1},$$

$$d_B(2k+1)=[d''_B(2k+1)-(k_{2k+1}+1) \pmod n] \pmod{n-1}= [n-1-d'_A(2k+1)-(k_{2k+1}+1) \pmod n] \pmod{n-1}=[-d'_A(2k+1)-(k_{2k+1}+1) \pmod n] \pmod{n-1} =[-d_A(2k)-1-(k_{2k+1}+1) \pmod n] \pmod{n-1}.$$

It follows that

$$d_A(2k+1)= -[d_B(2k)+(1+k_{2k+1}) \pmod n+2] \pmod{n-1},$$

$$d_B(2k+1)= -[d_A(2k)+(1+k_{2k+1}) \pmod n+1] \pmod{n-1}.$$

Therefore,

$$d_A(2k+2)=-[d_B(2k+1)+(1+k_{2k+2}) \pmod n+2] \pmod{n-1}= -[-d_A(2k)-(1+k_{2k+1}) \pmod n-1+(1+k_{2k+2}) \pmod n+2] \pmod{n-1}= [-d_A(2k)+(k_{2k+2}-k_{2k+1})+1] \pmod{n-1}= [d_A(2k)+(k_{2k+1}-k_{2k+2})-1] \pmod{n-1},$$

$$d_B(2k+2)=-[d_A(2k+1)+(1+k_{2k+2}) \pmod n+1] \pmod{n-1}= -[-d_B(2k)-(1+k_{2k+1}) \pmod n-2+(1+k_{2k+2}) \pmod n+1] \pmod{n-1}=-[-d_B(2k)+(k_{2k+2}-k_{2k+1})-1] \pmod{n-1}= [d_B(2k)+(k_{2k+1}-k_{2k+2})+1] \pmod{n-1}.$$

We apply an induction over k and obtain.

$d_A(2k+2)=[d_A(2k) + (k_{2k+1}-k_{2k+2})-1] \pmod{n-1}= [d_A(2k-2)+(k_{2k-1}+k_{2k+1}-k_{2k}-k_{2k+2})-2] \pmod{n-1}=. \ldots=\{d_A(0)+ \sum_{j=0}^{k} (k_{2j+1} - k_{2j+2})-k+1\} \pmod{n-1}.$

$d_B(2k+2)= [d_B(2k) + (k_{2k+1}-k_{2k+2})+1] \pmod{n-1}= [de(2k-2) + (k2k-1+kzk+1-k_{2k}-k_{2k+2})+2] \pmod{n-1}=\ldots=[dB(0)+ \sum_{j=0}^{k} (k_{2j+1} - k_{2j+2})+k+1] \pmod{n-1}.$

$d_A(2k+1)=-[d_B(2k)+(1+k_{2k+1}) \pmod{n}+2] \pmod{n-1}= -[d_B(0)+ \sum_{j=0}^{k-1} (k_{2j+1}-k_{2j+2})+k+2+(1+ k_{2k+1}) \pmod{n}] \pmod{n-1}= -[d_B(0)+ \sum_{j=0}^{k-1} (k_{2j+2}-k_{2j+1})-k-2-(1+k_{2k+1}) \pmod{n}] \pmod{n-1}.$

$d_B(2k+1)=-[d_A(2k)+(1+k_{2k+1}) \pmod{n}+1] \pmod{n-1}= -[d_B(0)+ \sum_{j=0}^{k-1} (k_{2j+1}-k_{2j+2})-k+1+(1+k_{2k+1}) \pmod{n}] \pmod{n-1}=[-d_B(0)+ \sum_{j=0}^{k-1} (k_{2j+2}-k_{2j+1})+k-1-(1+k_{2k}+1) \pmod{n}] \pmod{n-1}.$

This completes the proof.

REMARK 1

Let $P\{d_B=n-1, d_B=n-2, d_A=n-1, d_B= d_A +1, d_A= d_B +1\}$ be a probability that $d_B=n-1$, or $d_B=n-2$, or $d_A=n-1$, or $d_B= d_A +1$, or $d_A= d_B +1$ then

$$P\{d_B=n-1,d_B=n-2,d_A=n-1,d_B= d_A +1,d_A= d_B +1\}\leq 4/n.$$

Proof.

Really, $P\{d_B=n-1,d_B=n-2,d_A=n-1,d_B= d_A +1,d_A=d_B +1\}\leq 3\cdot(n-1)!/n! +2*(n-2)!/n!=3/n+2/(n-1)n\leq 4/n.$

Let Prob(j) be a probability that the probabilistic relations at time j are true.

REMARK 2

$$\text{Prob}(j) \geq (1-4/n)^j$$

Proof.

Note that the probabilistic relations at time j are true if for any t<j: $d_B[t] \neq n-1$, $d_B[t] \neq n-2$, $d_A[t] \neq n-1$, $d_B[t] \neq d_A[t]+1$ and $d_A[t] \neq d_B[t]+1$. Using remark 1, we have $P \geq (1-4/n)^j$.

Let us consider some properties that obtained from the presented probabilistic relations. By $dist_{AB}(t)$ denote the number of elements between the A joker and the B joker at time t. Proposition 1 and proposition 2 show that $\mathbf{dist_{AB}(t)}$ depends on t and $d_A(0)-d_B(0)$.

PROPOSITION 1

Let $x=(d_A(i)-d_B(i))(\text{mod } n-1)$ then

$$\mathbf{dist_{AB}(i)} \in \{x-1,\ n-2-x\}$$

Proof.

Let us remark that $\mathbf{dist_{AB}(i)}=|d_A(i)-d_B(i)|-1$.

Consider two possible cases.

a) If $\mathbf{d_A(i)>d_B(i)}$ then $\mathbf{dist_{AB}(i)}=(d_A(i)-d_B(i))(\text{mod } n-1)-1=x-1$

b) If $\mathbf{d_A(i)<d_B(i)}$ then $x=(d_A(i)-d_B(i))(\text{mod } n-1)=n-1+d_A(i)-d_B(i)$. Therefore, $\mathbf{dist_{AB}(i)}=d_B(i)-d_A(i)-1=n-2-X$

The proposition is proved.

PROPOSITION 2

Let $\mathbf{y=}(d_A(0)-d_B(0)-k)\ (\text{mod } n-1)$ then

$$\mathbf{dist_{AB}(k)} \in \{\mathbf{y-1},\ n-2-y\}$$

Proof.

Let $\mathbf{x=}(d_A(i)-d_B(i))(\text{mod } n-1)$. By proposition 1 and (1), (2) we obtain.

a) If k=2i+1 then

$$x=(d_A(k)-d_B(k))(\text{mod } n-1)=(-d_B(0)+\sum_{j=0}^{i-1}(k_{2j+2}-k_{2j+1})-(1+k_{2i+1})(\text{mod } n)$$

$$-i-2-(-d_A(0)+\sum_{j=0}^{i-1}(k_{2j+2}-k_{2j+1})+i-1-(1+k_{2i+1})\ (\text{mod } n)))\ (\text{mod } n-1)=\ldots$$

$$=(d_A(0)-d_B(0)-2i-1)(\text{mod } n-1)=(d_A(0)-d_B(0)-k)(\text{mod } n-1).$$

b) If k=2i then

$$x=(d_A(k)-d_B(k))(\text{mod } n-1)=(d_A(0)+\sum_{j=0}^{i-1}(k_{2j+1}-k_{2j+2})-i-d_B(0)+\sum_{j=0}^{i-1}(k_{2j+1}-k_{2j+2})+i)\ (\text{mod } n-1)=(d_A(0)-d_B(0)-2i)(\text{mod } n-1)=(d_A(0)-d_B(0)-k)\ (\text{mod } n-1).$$

Therefore, $x=(d_A(k)-d_B(k))(\text{mod } n-1)=(d_A(0)-d_B(0)-k)(\text{mod } n-1)=y$.
We have $\mathbf{dist_{AB}(k)}\in\{y-1, n-2-y\}$
The proposition is proved.

REMARK 3

If we take $x=(d_B(k)-d_A(k))(\text{mod } n-1)$ and $y=(d_B(0)-d_A(0)+k)$ (mod n-1) then proposition 1 and proposition 2 remain true.

PROPOSITION 3

If $d_B(0)-d_A(0)=2$ then

$$\mathbf{dist_{AB}}(k)\in\{k+1,\ n-k-4\}.$$

This proposition can be proved by direct calculations.

In propositions 4–7 we describe some properties which allow finding the jokers location or elements of the permutation with high probabilities. Let S'_0, S''_0 be two initial states. By d'_A, d'_B denote the A joker distance and the B joker distance for S' and by d''_A, d''_B denote distances for S''.

PROPOSITION 4

1. If $d'_A(0)=d''_A(0)$, $d'_B(0)=d''_B(0)$ and $d'_A(k)=d''_A(k)$ then $d'_B(k)=d''_B(k)$.
2. If $d'_A(0)=d''_A(0)$, $d'_B(0)=d''_B(0)$ and $d'_B(k)=d''_B(k)$ then $d'_A(k)=d''_A(k)$.

Proof.

Let us prove item 1.

a) Let k=2i.

By $d'_A(k)=d''_A(k)$ and (1) we get

$$[d'_A(0)+\sum_{j=0}^{i-1}(k'_{2j+1}-k'_{2j+2})-i]=[d''_A(0)+\sum_{j=0}^{i-1}(k''_{2j+1}-k''_{2j+2})-i]\ (\text{mod } n-1).$$

Therefore,

$$\sum_{j=0}^{i-1}(k'_{2j+1}-k'_{2j+2})=\sum_{j=0}^{i-1}(k''_{2j+1}-k''_{2j+2})\ (\text{mod } n-1).$$

Note that

$$d'_B(2i)= [d'_B(0)+ \sum_{j=0}^{i-1} (k'_{2j+1}- k'_{2j+2})+i] \text{ (mod n-1) and } d''_B(2i)= [d''_B(0)+ \sum_{j=0}^{i-1} (k''_{2j+1} -k''_{2j+2})+i] \text{ (mod n}-1).$$

Therefore, $d'_B(k)= d''_B(k)$.

b) Let k=2i+1.

From (2) and $d'_A(k)= d''_A(k)$ it follows that

$$[-d'_B(0)+\sum_{j=0}^{i-1} (k'_{2j+2} -k'_{2j+1})-(1+k'_{2i+1})(\text{mod n})\text{-i-2}] =$$

$$[-d''_A(0)+\sum_{j=0}^{i-1} (k''_{2j+2}- k''_{2j+1})+i-1-(1+k''_{2i+1}) \text{ (mod n)}] \text{ (mod n-1).}$$

By

$$d'_B(k)= [-d'_A(0)+\sum_{j=0}^{1-1} (k'_{2j+2}- k'_{2j+1})+i-1-(1+k'_{2i+1})(\text{mod n})](\text{mod n}-1),$$

$$d''_B(k)= [-d''_A(0)+ \sum_{j=0}^{i-1} (k''_{2j+2}-k''_{2j+1})+i-1-(1+k''_{2i+1})(\text{mod n})] \text{ (mod n-1)}$$

we have

$$\sum_{j=0}^{i-1} (k'_{2j+2}-k'_{2j+1})-(1+k'_{2i+1})(\text{mod n})= \sum_{j=0}^{i-1} (k''_{2j+2} - k''_{2j+1})+i-1-(1+k''_{2i+1}) \text{ (mod n)}.$$

Therefore, $d'_A(k)= d''_A(k)$.

Item 2 is proved similarly.

The proposition is proved.

PROPOSITION 5

If we know either $\{d_A(k), d_B(k+1)\}$ or $\{d_B(k), d_A(k+1)\}$ then we can determine the value of k_{k+1}.

This proposition can be proved by direct calculations.

PROPOSITION 6

1. If we know either $\{d_A(0), d_A(2i)\}$ or $\{d_B(0), d_B(2i)\}$ then we can determine

$$I=\sum_{j=0}^{i-1} (k_{2j+1}-k_{2j+2}).$$

2. If we know either $\{d_B(0), d_A(2i+1)\}$ or $\{d_A(0), d_B(2i+1)\}$ then we can determine

$$I=\sum_{j=0}^{i-1} (k_{2j+2}-k_{2j+1})-(1+k_{2i+1}).$$

Proof.
Let us prove item 1.
Note that (1) we can rewrite as

$$d_A(2i)=\sum_{j=0}^{i-1} (k_{2j+1}-k_{2j+2})+d_A(0)-i \pmod{n-1}=I+d_A(0)-i \pmod{n-1},$$

$$d_B(2i)=\sum_{j=0}^{i-1} (k_{2j+1}-k_{2j+2})+d_B(0)+i \pmod{n-1}=I+d_B(0)+i \pmod{n-1}.$$

This yields that

$$I=\sum_{j=0}^{i-1} (k_{2j+1}-k_{2j+2})=d_A(2i)-d_A(0)+i \pmod{n-1},$$

$$I=\sum_{j=0}^{i-1} (k_{2j+1}-k_{2j+2})=d_B(2i)-d_B(0)-i \pmod{n-1}$$

Item 2 is proved similarly.
The proposition is proved.

PROPOSITION 7

1. If $d_A(0)=d_A(k)$ then $d_B(k)=(d_B(0)+k) \pmod{n-1}$.
2. If $d_B(0)=d_B(k)$ then $d_A(k)=(d_A(0)-k) \pmod{n-1}$.

This proposition can be proved by direct calculations.

Let c be an element of the permutation S and $c \notin (A, B)$ and by $d_c(j)$ denote the number of elements before c at time j. In proposition 8 we find $d_c(1)$.

PROPOSITION 8

1. If either $d_A(0)< d_c(0)< d_B(0)$ or $d_B(0)< d_c(0)< d_A(0)$ then $d_c(1)=[-d_A(0) - d_B(0)-k_1+ d_c(0)-4] \pmod{n-1}$.
2. If either $d_c(0)< d_A(0)<d_B(0)$ or $d_B(0)< d_A(0)< d_c(0)$ then $d_c(1)= [d_c(0)- d_A(0)-k_1-2] \pmod{n-1}$.
3. If either $d_c(0)<d_B(0)<d_A(0)$ or $d_A(0)<d_B(0)< d_c(0)$ then $d_c(1)= [d_c(0)- d_B(0)-k_1-2] \pmod{n-1}$.

The proof is straightforward.

4. PROBABILISTIC RELATIONS FOR THE KEY SCHEDULING ALGORITHM

In this section we present probabilistic relations for the key scheduling algorithm.

THEOREM 2

Let $K=k_1,\ldots,k_L$ be a passphrase, where L is its length. Let $S_0=(0, 1, 2, \ldots,n-3, A, B)$ and $S_j=(F)^j(S_0)$. Let $d_A(j)$ be the A joker distance at time j and $d_B(j)$ be the B joker distance at time j. If for any $t<j$: $d_B[t]\neq n-1$, $d_B[t]\neq n-2$, $d_A[t] \neq n-1$, $d_B[t] \neq d_A[t]+1$ and $d_A[t] \neq d_B[t]+1$, then the A joker distance and the B joker distance satisfy the following relations.

1. If $j=2i$ then

$$d_A(2i)=[d_A(0)+\sum_{j=0}^{i-1} (k_{2j+1} - k_{2j+2}) - i] \pmod{n},$$

$$d_B(2i)=[d_B(0)+\sum_{j=0}^{i-1} (k_{2j+1}- k_{2j+2})+i] \pmod{n}.$$

2. If $j=2i+1$ then

$$d_A(2i+1) = [-d_B(0) + \sum_{j=0}^{i-1} (k_{2j+2} - k_{2j+1}) - k_{2i+1} - i - 3] \pmod n,$$

$$d_B(2i+1) = [-d_A(0) + \sum_{j=0}^{i-1} (k_{2j+2} - k_{2j+1}) + i - 2 - k_{2i+1}] \pmod n.$$

This theorem is proved as theorem 1 .

We stress that the propositions which are proved in the previous section remain true for the key scheduling algorithm but the operation "mod (n-1)" is changed by ''(mod n)".

5. CONCLUSION

In this paper we presented probabilistic relations for the jokers in Solitaire keystream generator and described some their properties. It analyzes the probability distribution of distance between the two jokers in a deck at different time periods. We found that the number of elements between jokers at time t depends on t and the initial number of elements between the A joker and the B joker. Presented results with bit changes are applied to the key-scheduling algorithm of Solitaire.

We hope that results described in [2], [3] and this paper allow mounting an attack on this cipher which is more effective than exhaustive search, and this will be the object of another paper.

REFERENCES

[1] Schneier B., "The Solitaire Encryption Algorithm", http://www.counterpane.com/solitaire.html.

[2] Pudovkina M, Varfolomeev A.A. "A Cycle Structure of the Solitaire Keystream Generator". 3nd International Workshop on Computer Science and Information Technologies CSIT'2001, YFA, 2001.

[3] Pudovkina M., "Weakness in the Key Scheduling Algorithm of the Solitaire Keystream Generator", SIBCOM-2001, TOMSK. (the paper is being published)

[4] Varfolomeev A.A., Zhukov A.E., Pudovkina M., "Analysis of Stream Ciphers ', Moscow, MEPhI, 2000.

[5] Crowley P. "Problems with Bruce Schneier "Solitaire"".,
http://www.hedonism.demon.co.uk/ paul/solitaire/

[6] Rueppel R.A. "Analysis and Design of Stream Ciphers", Springer-Verlag, Communications and Control Engineering Series, 1986.

[7] Rivest R.L.,"TheRC4 encryption algorithm",RSA Data Security, Inc., Mar. 1992
[8] R.J. Jenkins, "ISAAC", Fast Software Encryption -Cambridge 1996, vol. 1039, D. Gollmann ed.. Springer-Verlag.
[9] Simeon V. Maltchev and Peter T. Antonov. "The SCOP Stream Cipher", ftp://ftp.funet.fi/pub/crypt/cryptography/symmetric/scop/scop.tar.gz, Dec. 1997.