On the CLD Attack to a Statistical Model of a Key Stream Generator

Shaoquan Jiang, Zailiang Tang, and Mingsheng Wang
(Corresponding author: Shaoquan Jiang)

Institute of Information Security, Mianyang Normal University
166 Mianxing Rd. West, High-Tech District, Mianyang 621000, China
(Email: shaoquan.jiang@gmail.com)
(Received Apr. 16, 2015; accepted Dec. 16, 2015)

Abstract

An embedding attack based on constraint Lenvenshtein distance was proposed by Golić and Mihaljević to analyze a statistical model of a key stream generator which contains an additive noise of probability p, where any value of $p < 1/2$ is possible. This attack is significant only if the embedding error caused by the noise is less than that caused by an incorrect candidate initial state. We show that this condition is not satisfied when $p \geq 1/4$.

Keywords: Cryptanalysis, LFSR, probabilistic analysis, stream cipher

1 Introduction

A key stream generator is a function that maps a short key into a long stream, which can be used to efficiently encrypt a plaintext stream by bit-wise XORing with the latter. The main purpose is to make it fast. The lightweight key stream generator Sprout [1] is such an example. However, proposing a secure key stream generator is very tricky. In fact, Sprout has been effectively attacked [12, 17]. See [2, 4, 5] for other examples of key stream generators. In this paper, we consider the key stream generator based on a linear feedback shift register (LFSR) sequence [10] which is a very efficient mechanism for a key stream generator (of course, LFSR has many applications such as frequency-hopping communication [3]).

In fact, most of generators in the literature are designed using it. However, many of them are broken by exploiting the linearity of LFSRs; see the correlation attack [14] for an example. A popular method for this type of attack is to reduce a complicated generator to a statistical model $Y_i = X_i + E_i, i = 1, 2, \cdots$, where $\{X_i\}_{i\geq 1}$ is a secret LFSR sequence, $\{Y_i\}_{i\geq 1}$ is the key stream and E_i is a binary noise with $P(E_i = 1) = p < 1/2$. For instance, Zeng et al. [15] reduced generators [6] to this model and completely broke them using a linear syndrome attack.

Generators subject to this attack usually have a common feature: Its input LFSR is regularly clocked. An irregularly clocked key stream generator is desired. Golić et al. [7] studied the security of this type of generator by considering the model $Y_i = X_{f(i)} + E_i, i = 1, 2, \cdots$, where $P(E_i = 1) = p < 1/2$, $f(i) = i + \sum_{j=1}^{i} a_j$ and $\{a_j\}$ is another LFSR. They proposed a constrained embedding attack to this model. When $p = 0$, this model degenerates to a decimation generator.

Golić and O'Connor [8] proposed an (un)constrained embedding attack to this generator when the irregularly clocking step is bounded by D. The embedding probability for $D = 2$ was given in [9]. Zhang [16] proposed a new attack to the decimation generator.

Given a partial key stream $Y_1, \cdots, Y_n$ of the model $Y_i = X_{f(i)} + E_i$ with $\Pr(E_i = 1) = p < 1/2$, Golić and Mihaljević considered a noisy embedding attack: Try to embed Y^n into the prefix $\hat{X}_1, \cdots, \hat{X}_{2n}$ of a candidate LFSR $\hat{X}$ for X (assume the resulting error sequence is $\hat{E}^n$) and find $\hat{X}$ with the least $\sum_{i=1}^{n} \hat{E}_i$ as the solution for X. The attack succeeds if it gives the solution $\hat{X} = X$. Notice that when generating Y^n from X^{2n}, the noise sequence is $E_1, \cdots, E_n$. Hence, this attack is significant only if $\sum_{i=1}^{n} E_i < \sum_{i=1}^{n} \hat{E}_i$ for any LFSR $\hat{X}$ other than X (otherwise, $Y_1, \cdots, Y_n$ is less noisy when considered as generated from a wrong LFSR $\hat{X}$). In this paper, we show that this condition is invalid when $p \geq 1/4$.

2 Preliminaries

Notions: We will use the following notions.

- For a set $\mathcal{S}$, $s \leftarrow \mathcal{S}$ samples an element s from $\mathcal{S}$ uniformly randomly.
- For $j \leq n$, u_j^n denotes sequence $u_j, u_{j+1}, \cdots, u_n$, and sequence u_1^n is simply denoted by u^n.
- For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \cdots, n\}$.
- i.i.d. is a well-known abbreviation of "*independently identically distributed*".

2.1 LFSR and Key Stream Generator

A binary *linear shift register sequence* (LFSR) $S = s_0, s_1, \cdots$ is a sequence generated using a linear recursive relation $s_{j+k} = s_j c_{k-1} + s_{j+1}c_{k-2} + \cdots + s_{j+k-1}c_0$ over $\mathbb{F}_2$, starting with an *initial state* $(s_0, \cdots, s_{k-1})$, where $c_0, \cdots, c_{k-1} \in \mathbb{F}_2$ are called *connection coefficients*.

A *key stream generator* is a function $f : \{0,1\}^k \to \{0,1\}^*$ that maps a secret key $\mathbf{w} \in \{0,1\}^k$ into a long binary stream $z_1, z_2, \cdots$. It can be used to encrypt a plaintext stream $m_1, m_2, \cdots$ by simply bit-wise XORing: $m_1 \oplus z_1, m_2 \oplus z_2, \cdots$. When a receiver with the secret key $\mathbf{w}$, receives the ciphertext, he can recover the plaintext in an obvious way. For the generator to be useful, we must make sure it is secure against some attacks.

A relatively weak attack is a *ciphertext-only attack*, which requires an adversary to recover the secret key $\mathbf{w}$ when only a partial ciphertext stream is given. A widely considered attack is a *known plaintext attack*: The adversary is given a partial ciphertext and its corresponding plaintext and his objective is to recover the secret $\mathbf{w}$. Equivalently, the attacker is given a partial key stream $z_1, \cdots, z_n$, from which he tries to recover the secret $\mathbf{w}$. It is well-known from Berlekamp-Massey algorithm [13] that LFSR with an initial state and connection coefficients as the secret key is not a secure key stream generator. However, LFSR is a very useful tool to construct a reasonably secure key stream generator.

2.2 Hoeffding Inequality

We now introduce the famous Hoeffding inequality. For details, see [11].

Lemma 1. *[Hoeffding] Let $X_1, \cdots, X_n$ be n independent RVs with $a_i \le X_i \le b_i$ for $i = 1, \cdots, n$. Then, for $\forall t > 0$,*

$$P\Big(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \ge t\Big) \le e^{-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}},$$

$$P\Big(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \le -t\Big) \le e^{-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}},$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(X_i)$.

3 Problem Statement

3.1 A Statistical Model of a Clock-controlled Generator

Golić and Mihaljević [7] considered the following statistical model of a key stream generator. Let $X = \{x_i\}_{i\ge1}$ and $A = \{a_i\}_{i\ge1}$ be two LFSR sequences. The key stream $Z = \{z_i\}_{i\ge1}$ is generated noisily as follows.

$$z_i = x_{f(i)} + e_i, i = 1, 2, \cdots, \quad (1)$$

where $f(i) = i + \sum_{t=1}^{i} a_t$ and $e_1, e_2, \cdots$, are i.i.d. with $P(e_i = 1) = p < 0.5$. Strictly, this is not a key stream generator as it involves an additive noise e_i. However, this could be a useful abstraction of a key stream generator. Specifically, e_i could be an a statistical approximation to a complicated structure.

The initial state recovering problem for this generator is to find the initial states of X and A, assuming the noise probability p, a partial key stream z^n and connection coefficients of X and A are known. A näive approach for this is to search for all possible initial states of X and A and verify whether $x_{f(i)}$ matches with z_i for $i = 1, \cdots, n$ with probability roughly p. However, if each of X and A has an initialize state length k, then it requires $O(2^{2k})$ times of tests.

3.2 Constrained Levenshtein Distance Attack

Golić and Mihaljević [7] proposed a noisy embedding attack based on *Constrained Levenshtein Distance* (CLD) to recover $\{x_i\}_{i\ge1}$ from z^n. We call it a CLD *attack*. Their approach is as follows. For each candidate $\hat{X}$ of X, they generate a partial sequence $\hat{x}^{2n}$ and compute the CLD between $\hat{x}^{2n}$ and the known key stream z^n, where CLD is defined as follows.

$D^*(\hat{x}^{2n}, z^n)$ = minimal number of deleting and complementation operations required to produce z^n from $\hat{x}^{2n}$, by first deleting an arbitrary prefix of $\hat{x}^m$ and then following the model at Equation (1).

Let $\mathcal{X}$ be the set of candidate $\hat{X}$ for X. If the initial state of X has k bits, then $|\mathcal{X}| = 2^k$. Given z^n, the attack outputs X^* that minimizes CLD (among all possible sequences in $\mathcal{X}$) as its solution for X. It succeeds if $X^* = X$.

For each $\hat{X}$, [7] showed that CLD$(\hat{x}^m, z^n)$ can be computed in $O(mn)$ time and hence is efficient.

Note that the number of deletions in producing z^n from $\hat{x}^{2n}$ is the constant n. This attack is equivalent to minimize the number of complementing operations. We denote the number of complementing operations in $D^*(\hat{x}^{2n}, z^n)$ by $D(\hat{x}^{2n}, z^n)$. In the sequel, instead of $D^*(\hat{x}^{2n}, z^n)$, we will focus on $D(\hat{x}^{2n}, z^n)$.

Note that z^n is generated from X, A in the real process with the complementing sequence e^n. So the number of complementing operations in this process is $\sum_{i=1}^{n} e_i$. Hence, the CLD attack is meaningful only if $\sum_{i=1}^{n} e_i < D(\hat{x}^{2n}, z^n)$ for each $\hat{X} \in \mathcal{X} - \{X\}$. That is, the real number of complementing operations should not be greater than that under a wrong candidate sequence $\hat{X}$. Hence, we consider

$$\alpha = P\Big(D(\hat{x}^{2n}, z^n) \le \sum_{i=1}^{n} e_i\Big). \quad (2)$$

As in [7], we model an LFSR as a purely random sequence. So α is defined over the uniform random $\hat{x}^{2n}, x^{2n}$

and the randomness of z^n and e^n.

The expected number of $\hat{X}$ in $\mathcal{X}$ with $D(\hat{x}^{2n}, z^n) \leq \sum_{i=1}^{n} e_i$ is $2^k\alpha$, as $|\mathcal{X}| = 2^k$. Thus, the CLD attack is meaningful only if $2^k\alpha$ is small. The problem in this paper is to lower bound α and show that $\alpha > constant$ when $p \geq 1/4$. In this case, $2^k\alpha$ is large, which makes the attack fail to identify which $\hat{X}$ will be the true X.

Finally, we notice that $D^*(\hat{x}^{2n}, z^n)$ permits deleting an arbitrary *prefix* of $\hat{x}^m$. However, here the prefix can be changed to postfix without affecting α in Equation (2). Indeed, we can convert a postfix into a prefix version in the following way. We can start to embed z^n reversely to $\hat{x}^{2n}$. That is, we can embed $z_n, z_{n-1}, \cdots, z_1$ into $\hat{x}_{2n}, \cdots, \hat{x}_1$. If z_1 is embedded at $\hat{x}_j$, then $\hat{x}^{j-1}$ can be deleted by the convention.

Since z^n and $\hat{x}^{2n}$ are uniformly random and independent, the distribution of $z_n, \cdots, z_1, \hat{x}_{2n}, \cdots, \hat{x}_1$ and the distribution of $z_1, \cdots, z_n, \hat{x}_1, \cdots, \hat{x}_{2n}$ are exactly the same. So the two ways give the same α. In this paper, for convenience, we use the postfix version for $D(\hat{x}^{2n}, z^n)$ (i.e., we revise "prefix" in the definition of $D^*(\hat{x}^{2n}, z^n)$ to "postfix").

4 Lower Bound on α When $p \geq 1/4$

In this section, we show that α is larger than a constant when $p \geq 1/4$. Our strategy is as follows. For an embedding algorithm $\mathcal{E}$ that embeds z^n into u^{2n}, we use $\mathcal{E}(u^{2n}, z^n)$ to denote the number of flips in the embedding process. Then, $\mathcal{E}(u^{2n}, z^n) \geq D(u^{2n}, z^n)$. It follows that $\alpha \geq P(\mathcal{E}(u^{2n}, z^n) \leq \sum_{i=1}^{n} e_i)$. Hence, it suffices to show that $P(\mathcal{E}(u^{2n}, z^n) \leq \sum_{i=1}^{n} e_i) > constant$ for some algorithm $\mathcal{E}$. So the main task is to design $\mathcal{E}$.

Now we present our algorithm $\mathcal{E}$ to embed z^n to u^{2n}, in which the average number of complements is $n/4$. The formal description is in Algorithm 1. The idea is as follows. It sequentially embeds $z_1, \cdots, z_n$ into u^{2n}. Let z_i be the current bit to be embedded and u_j be the currently unused bit awaiting to embed z_i. Initially, $i = j = 1$. If $z_i \neq u_j$ and $z_i \neq u_{j+1}$, then one complementing operation (recorded in a variable F) is used and z_i is embedded at u_{j+1}; otherwise, z_i is embedded to u_j when $z_i = u_j$ and embedded to u_{j+1} when $z_i = u_{j+1}$. Finally, increment i and update j to the next unused index.

Before analyzing our algorithm, we first present a general lemma. It considers a function $f : \{0,1\}^m \to [m-1]$. It states that if f satisfies a certain property, then for uniformly random U^m in $\{0,1\}^m$ and $J = f(U^m)$, we have that (U_J, U_{J+1}) is independent of U^{J-1}. We remark that this independency does not trivially follow from the uniformly randomness of U^m, as J depends on U^m and is implied from U^{J-1} (by looking at the dimension).

Lemma 2. *Let $f : \{0,1\}^m \to [m-1]$ be a function with the following property: if $f(u^m) = j$, then $f(u^{j-1}, v_j^m) = j$ for any $v_j^m \in \{0,1\}^{m-j+1}$. Let U^m be uniformly random in $\{0,1\}^m$ and $J = f(U^m)$. Then, (U_J, U_{J+1}) is independent of U^{J-1}.*

Algorithm 1 Embedding algorithm $\mathcal{E}$

Imput: u^{2n}, z^n;
Output: F
1: Begin
2: Set $i = 1, j = 1, F = 0$
3: **for** $i = 1$ to n **do**
4: **if** $z_i = u_j$ **then**
5: $j = j + 1$
6: **else**
7: **if** $z_i = u_{j+1}$ **then**
8: $j = j + 2$;
9: **else** $F = F + 1$ and $j = j + 2$;
10: **end if**
11: **end if**
12: **end for**
13: Return F
14: End

Proof. For $j \in [m-1]$, let $\mathcal{N}_j$ be the set of u^{j-1} such that $f(u^{j-1}, v_j^m) = j$ for some v_j^m. By the property of f, the set of all u^m with $f(u^m) = j$ is exactly $\mathcal{S}_j \stackrel{def}{=} \mathcal{N}_j \times \{0,1\}^{m-(j-1)}$. As any u^m must map to some $j \in [m-1]$, it follows that $\sum_{j=1}^{m-1} |\mathcal{N}_j| 2^{m-(j-1)} = 2^m$. So

$$\sum_{j=1}^{m-1} |\mathcal{N}_j| 2^{-(j-1)} = 1. \tag{3}$$

Notice that J can be derived from U^{J+1} by looking at the dimension. Hence, $U^{J+1} = u^{j+1}$ if and only if $J = j$ and $U^{j+1} = u^{j+1}$. So,

$$\begin{aligned} P(U^{J+1} = u^{j+1}) &= P(U^{j+1} = u^{j+1}, J = j) \\ &= P_{U^{j+1}}(u^{j+1}) P_{J|U^{j+1}}(j|u^{j+1}) \\ &= \begin{cases} 2^{-(j+1)}, & u^{j-1} \in \mathcal{N}_j \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{4}$$

where we have used the fact $P_{J|U^{j+1}}(j|u^{j+1}) = 1$ if $u^{j-1} \in \mathcal{N}_j$ and zero, otherwise. Similarly,

$$\begin{aligned} P(U^{J-1} = u^{j-1}) &= P(U^{j-1} = u^{j-1}, J = j) \\ &= P_{U^{j-1}}(u^{j-1}) P_{J|U^{j-1}}(j|u^{j-1}) \\ &= \begin{cases} 2^{-(j-1)}, & u^{j-1} \in \mathcal{N}_j \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

Therefore, $P_{U^{J+1}}(u^{j+1}) = \frac{1}{4} P_{U^{J-1}}(u^{j-1})$. As

$$\begin{aligned} P_{U_J U_{J+1}}(a,b) &= \sum_{j=1}^{n} P_{U_j U_{j+1} J}(a,b,j) \\ &\stackrel{(*)}{=} \sum_{j} |\mathcal{N}_j| \cdot 2^{-(j+1)} \\ &= 1/4, \text{ (by Equation (3))} \end{aligned} \tag{6}$$

where equality $(*)$ follows from the fact that $(U_j, U_{j+1}, J) = (a, b, j)$ if and only if

$$U^m \in \mathcal{N}_j \times \{a\} \times \{b\} \times \{0,1\}^{m-(j+1)},$$

and the fact that U^m is uniformly random over $\{0,1\}^m$. Hence, Equations (4) (5) (6) imply

$$P_{U^{J+1}}(u^{j+1}) = P_{U_J U_{J+1}}(u_j, u_{j+1}) P_{U^{J-1}}(u^{j-1}). \quad (7)$$

That is, (U_J, U_{J+1}) is independent of U^{J-1}. □

We are ready to analyze our algorithm $\mathcal{E}$. We will use the following notations. Let $U^{2n} = U_1, \cdots, U_{2n}$ be a sequence of purely random binary stream. We define a binary RV F_i with respect to algorithm $\mathcal{E}(U^{2n}, z^n)$ such that $F_i = 1$ if and only if $F = F+1$ is executed in loop i. Then, $F = \sum_{i=1}^n F_i$. Since $\mathcal{E}$ is deterministic, the randomness of F is over U^{2n}. Let J_i be the index j at the beginning of loop i (e.g., $J_1 = 1$). Define function $\delta : \{0,1\}^2 \to \{0,1\}$ is defined such that $\delta(x, y) = 1$ if and only if $x = y$.

In the following, we show that $F_1, \cdots, F_n$ are independent and that $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_n}, z_n)$ are independent too, where the independency for both collections follows only from the randomness of U^{2n}.

Lemma 3. *Given $z^n \in \{0,1\}^n$, if U^{2n} is uniformly random in $\{0,1\}^{2n}$, then*

1) *RVs $F_1, \cdots, F_n$ are i.i.d. with $P(F_i = 1) = 1/4$.*
2) *RVs $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_n}, z_n)$ are i.i.d. with $P(\delta(U_{J_i}, z_i) = 1) = 1/2$.*

Proof. Notice that for any $i \le n$, we have $J_i < 2n$. By Lemma 2, if $f(U^{2n}) = J_i$, then (U_{J_i}, U_{J_i+1}) is independent of U^{J_i-1}. This will be used in our proof.

1) We start with the following claim.

   Claim 1. *For fixed z^n and i, we have that $(F_1, \cdots, F_{i-1})$ is deterministic in U^{J_i-1}.*

   Proof. Indeed, by our algorithm, if $z_{i-1} = U_{J_{i-1}}$, then $J_i = 1 + J_{i-1}$ and $F_{i-1} = 0$; otherwise, $J_i = 2 + J_{i-1}$, and $F_{i-1} = 1$ if and only if $z_{i-1} \neq U_{1+J_{i-1}}$. Here we can see that in any case, F_{i-1} is computed only using U^{J_i-1}. So for any $\ell < i$, $F_{\ell-1}$ is determined by $U^{J_\ell-1}$ (which in turn is determined by U^{J_i-1}). Thus, $(F_1, \cdots, F_{i-1})$ are deterministic in U^{J_i-1}, when z^n is fixed. This concludes the proof of our claim. □

   From the algorithm description, we can write $F_i = (z_i \oplus U_{J_i}) \wedge (z_i \oplus U_{1+J_i})$. Thus, F_i is deterministic in (U_{J_i}, U_{1+J_i}). From claim 1 and the fact that (U_{J_i}, U_{J_i+1}) is independent of U^{J_i-1} (see the beginning of the proof), we know that F_i is independent of $(F_1, \cdots, F_{i-1})$.

   Finally, notice that (U_{J_i}, U_{1+J_i}) is independent of J_i, as J_i is deterministic in U^{J_i-1} (by looking at the dimension). Hence,

$$\begin{aligned} & P((U_{J_i}, U_{1+J_i}) = (a,b)) \\ = & \sum_j P((U_j, U_{j+1}, J_i) = (a,b,j)) \\ = & \sum_j P((U_j, U_{j+1}) = (a,b)) P(J_i = j) \\ = & \frac{1}{4} \sum_j P(J_i = j) = 1/4. \end{aligned}$$

   Hence, from $F_i = (z_i \oplus U_{J_i}) \wedge (z_i \oplus U_{1+J_i})$, we have $P(F_i = 1) = 1/4$.

2) As U_{J_i} is independent of U^{J_i-1}, we have $\delta(U_{J_i}, z_i)$ is independent of $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_{i-1}}, z_{i-1})$ for any i. Hence, $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_n}, z_n)$ are independent. Finally, as J_i is deterministic in U^{J_i-1} (by looking at the dimension), U_{J_i} is independent of J_i. Thus,

$$\begin{aligned} P(U_{J_i} = 0) &= \sum_j P((U_j, J_i) = (0,j)) \\ &= \sum_j P(U_j = 0) P(J_i = j) \\ &= \frac{1}{2} \sum_j P(J_i = j) = 1/2. \end{aligned}$$

   This completes our proof.

□

We are ready to prove our theorem. This mainly is achieved using Heoffding inequality to $F_1, \cdots, F_n$ and the true error sequence $e_1, \cdots, e_n$ in producing z^n.

Theorem 1. *If $p = 1/4$, then $\alpha \ge 1/2$; if $p > 1/4$, then $\alpha \ge 1 - e^{-(p-.25)^2 n}$.*

Proof. Notice that $F = \sum_{i=1}^n F_i$ is the number of complements in a specific embedding process. Hence, $F \ge D(U^{2n}, z^n)$. Hence,

$$\begin{aligned} \alpha =& P\Big(D(U^{2n}, z^n) \le \sum_{i=1}^n e_i\Big) \\ \ge& P\Big(\sum_{i=1}^n F_i \le \sum_{i=1}^n e_i\Big) \end{aligned} \quad (8)$$

Since F^n only depends on U^{2n}, it is independent of e^n. If $p = 1/4$, then $F_1, \cdots, F_n$ are identically distributed with $e_1, \cdots, e_n$. Then, by symmetry, $\alpha \ge 1/2$. If $p > 1/4$, then since $F_1, \cdots, F_n, e_1, \cdots, e_n$ are independent, by Hoeffding inequality with $2n$ random variables,

$$\begin{aligned} & P(e_1 + \cdots + e_n - F_1 - \cdots - F_n \ge 0) \\ =& 1 - P(\sum_i (e_i - F_i) - n(p - .25) < -n(p - .25)) \\ \ge& 1 - e^{-(p-.25)^2 n}. \end{aligned} \quad (9)$$

This completes our theorem. □

Now we look at how many bits Algorithm $\mathcal{E}$ has used in order to embed z^n. In fact, after embedding z_n, the next available index of U_j is J_{n+1}. So the number of bits in embedding z^n is $N_n \stackrel{def}{=} J_{n+1} - 1$. From our algorithm description, $J_{\ell+1} = 1 + \delta(U_{J_\ell}, z_\ell) + J_\ell$. Thus,

$$N_n = n + \sum_{i=1}^{n} \delta(U_{J_i}, z_i). \tag{10}$$

Notice that in the real process in producing z^n, we know that $f(n) = n + \sum_{i=1}^{n} a_i$, where $a_1, a_2, \cdots$, are i.i.d. and uniformly random over $\{0, 1\}$ (as idealized in our analysis). Therefore, by Lemma 3, the distribution of N_n is identical to the real distribution. This demonstrates an interesting aspect of our algorithm.

5 Conclusion

In this paper, we revisited the noisy embedding attack from constraint Lenvenshtein distance by Golić and Mihaljević to a noisy key stream generator that contains an additive binary noise term of probability p, where any value of $p < 1/2$ is possible. We showed that this attack is not successful if $p \geq 1/4$. One immediate interesting question is to study the success for the case $p < 1/4$. When p is very small, the exponentially small embedding probability without a noise showed in [9] trivially implies the success of this algorithm. However, in general, this does not seem to be a trivial question. We leave it as an open question.

Acknowledgements

This work is supported by Open grant (No. 2015-MS-11) of State Key Lab of Information Security, Institute of Information Engineering, CAS.

References

[1] F. Armknecht and V. Mikhalev, "On lightweight stream ciphers with shorter internal states," in *Fast Software Encryption*, pp. 451–470, Springer-Verlag, 2015.

[2] Y. Asimi, A. Amghar, A. Asimi and Y. Sadqi, "New random generator of a safe cryptographic salt per session," *International Journal of Network Security*, vol. 18, no. 3, pp. 445–453, 2016.

[3] J. Chung, G. Gong and K. Yang, "New families of optimal frequency-hopping sequences of composite lengths," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3688–3697, 2014.

[4] H. El-Razouk, A. Reyhani-Masoleh and G. Gong, "New implementations of the wg stream cipher," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1865–1878, 2014.

[5] H. El-Razouk, A. Reyhani-Masoleh and G. Gong, "New hardware implementations of WG (29, 11) and WG-16 stream ciphers using polynomial basis," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 2020–2035, 2015.

[6] P. R. Geffe, "How to protect data with ciphers that are really hard to break," *Electronics*, vol. 4, pp. 129–156, 1973.

[7] J. D. Golić and M. J. Mihaljević, "A generalized correlation attack on a class of stream ciphers based on the levenshtein distance," *Journal of Cryptology*, vol. 3, no. 3, pp. 201–212, 1991.

[8] J. D. Golić and L. O'Connor, "Embedding and probabilistic correlation attacks on clock-controlled shift registers," in *Proceedings of Advances in Cryptology (Eurocrypt'94)*, pp. 230–243, Springer-Verlag, 1994.

[9] J. D. Golić, "Constrained embedding probability for two binary strings," *SIAM Journal on Discrete Mathematics*, vol. 9, no. 3, pp. 360–364, 1996.

[10] S. W. Golomb, *Shift Register Sequences*, San Francisco: Holden-Day, Inc., 1967.

[11] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[12] V. Lallemand and M. Naya-Plasencia, "Cryptanalysis of full sprout," in *Advances in Cryptology (Crypto'15)*, pp. 663–682, Springer-Verlag, 2015.

[13] J. L. Massey, "Shift-register synthesis and bch decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[14] T. Siegenthaler, "Decrypting a class of stream cipher using ciphertext only," *IEEE Transactions on Computers*, vol. 34, no. 1, pp. 81–85, 1985.

[15] K. Zeng and M. Huang, "On the linear syndrome method in cryptoanalysis," in *Proceedings of Advances in Cryptology (Crypto'88)*, pp. 469–478, Springer-Verlag, 1990.

[16] B. Zhang, "New cryptanalysis of irregularly decimated stream ciphers," in *Proceedings of 16th Annual International Workshop of Selected Areas in Cryptography*, pp. 449–465, Springer-Verlag, 2009.

[17] B. Zhang and X. Gong, "Another tradeoff attack on sprout-like stream ciphers," in *Advances in Cryptology (Asiacrypt'15)*, pp. 561–585, Springer-Verlag, 2015.

Shaoquan Jiang received the B.S. and M.S. degrees in mathematics from the University of Science and Technology of China, Hefei, China, in 1996 and 1999, respectively. He received the Ph.D degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, ON, Canada, in 2005. From 1999 to 2000, he was a research assistant at the Institute of Software, Chinese Academy of Sciences, Beijing; from 2005 to 2013, he was a faculty member at the University of Electronic Science and Technology of China, Chengdu, China; from 2013 to now, he is a faculty member at Mianyang Normal University, Mianyang, China. He was a postdoc at the

University of Calgary from 2006 to 2008 and a visiting research fellow at Nanyang Technological University from Oct. 2008 to Feb. 2009. His research interests are key stream generators, public-key based secure systems and secure protocols.