IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 23, 2020, accepted September 7, 2020, date of publication September 11, 2020,
date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023425

On the Estimation of Synchronous Scramblers in Direct Sequence Spread Spectrum Systems

DONGYEONG KIM[1], JUNGHWAN SONG[1], (Member, IEEE), AND DONGWEON YOON[2], (Senior Member, IEEE)

[1]Department of Mathematics, Research Institute for Natural Sciences, Hanyang University, Seoul 04763, South Korea
[2]Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Dongweon Yoon (dwyoon@hanyang.ac.kr)

ABSTRACT In a non-cooperative context, the receiver has no information about communication parameters; therefore, it must perform communication forensics, which is the process of identifying what information it can from the collected data. This paper proposes a novel method for parameter estimation of synchronous scramblers in direct sequence spread spectrum systems. For the estimation, we use the bitwise relations inherent in the scrambling sequence and the repetitive patterns by the spreading code inherent in the scrambled sequence. Regarding computational complexity, previous studies of parameter estimation of synchronous scramblers require exponential computational complexities. Unlike the existing methods, our proposed method can practically estimate the feedback polynomial and initial state of the synchronous scrambler with polynomial computational complexity.

INDEX TERMS Linear feedback shift register, spread spectrum, scrambler, estimation.

I. INTRODUCTION

Communication forensics, which identify information from collected data by blind estimation of communication parameters, make essential contributions to both cooperative and non-cooperative contexts, such as wireless mobile communications, cognitive radios, and surveillance systems [1]. Particularly, blind estimation of communication parameters has played a more important role in non-cooperative contexts, such as spectrum surveillance systems and cognitive radio systems, where the receiver lacks all information about communication parameters. The receiver must therefore perform communication forensics. This is a tremendous work, and the estimation of even a single communication parameter is very difficult.

Research on the blind estimation of communication parameters in non-cooperative contexts has been separately conducted in its various aspects, including but not limited to source coding [2]–[6], channel coding [7]–[10], interleaving [11]–[17], modulation [18]–[24], spreading sequence [25]–[29], and scrambling [30]–[37]. In this paper, we focus on the estimation of scrambling parameters.

The direct sequence spread spectrum (DSSS) is widely used in commercial and military communication systems for its features such as anti-multipath and anti-jamming [38]–[42]. DSSS systems commonly employ scrambling with a maximal length sequence (m-sequence), which is generated by a linear feedback shift register (LFSR) having a period $2^n - 1$, where n is the degree of the feedback polynomial of the LFSR. A synchronously scrambled sequence is generated by modulo-2 addition of an input sequence and a scrambling sequence, generated by an LFSR, to the scrambler.

To recover a scrambled DSSS signal in a non-cooperative context, the parameters used for scrambling must first be blindly estimated on the receiver side. When a scrambler uses an m-sequence, the enormous number of candidates for the feedback polynomials and the initial state make parameter estimation a very challenging task. Scrambler parameter estimation is a favorite subject of inquiry [30]–[37]. Algorithms relevant to blind estimation of scrambler parameters include the following.

Under the assumption of a biased input sequence, [30]–[33] estimated scrambler parameters, the feedback polynomial and the initial state of the scrambler, by using the statistical difference between a truly random and a biased input sequence; [30] estimated scrambler parameters based on searching for sparse multiples of the feedback polynomial with the degree of the sparse multiples; [31]–[33] improved on the algorithm of [30]; [31] proposed a blind reconstruction method with the estimation of bias value; [32] estimated

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

the scrambler by adding a boundary condition to the upper estimation of standard deviation and considering flipped bits; and [33] proposed an improved algorithm using a statistical variable and the median value estimation.

Instead of relying on the bias, [34], [35] presented algorithms for estimation of scrambler parameters by using dual words of the channel encoder: [34] studied the problem of reconstruction of the LFSR in a synchronous scrambler placed after a channel encoder by using the orthogonal property of dual words; and [35] proposed a scheme to improve the detection capability of [34] by using a mean value and matched filter.

Those methods of estimation in [30]–[35] that discover sparse multiples of the feedback polynomial require a full search of the possible sparse multiples. Their computational complexity therefore increases exponentially with the scrambler parameters.

References [36] and [37] estimated the parameters of a synchronous scrambler in DSSS systems by using a triple correlation function (TCF) and eigenvalue decomposition (EVD); [36] investigated the properties of TCF and estimated scrambling sequence using TCF. To find the triple correlation value, [36] requires a full search for all possible integer pairs of $\left\{ (p, q) \in \{1, \cdots, 2^n - 1\}^2 \middle| p < q \right\}$, therefore, its computational complexity increases exponentially with regard to the degree of the feedback polynomial n; [37] used EVD to estimate the parameters of a synchronous scrambler, making a matrix whose column length is the period of the scrambling sequence. Therefore, EVD of the matrix also needs exponential computational complexity. In summary, the previous studies [30]–[37] blindly estimate the scrambler parameters; however, their computational complexities all increase exponentially, regardless of the methods used.

In this paper, we propose a novel method for parameter estimation of a synchronous scrambler in DSSS systems. Unlike previous studies having exponential computational complexity, our proposed method is based not on a full search nor huge matrix computation, but on the bitwise linear relations inherent in the scrambling sequence itself, and therefore has merely polynomial computational complexity.

For parameter estimation of a synchronous scrambler with an m-sequence in DSSS systems we reconstruct the scrambling sequence with the following properties: the repetitive patterns inherent in the scrambled sequence as a consequence of the spreading code used in the DSSS system, the well-known ''shift and add property'' of the m-sequence, and the linear relations made by the feedback polynomial between the bits in the m-sequence used for scrambling. We then show the estimation performance in terms of computational complexity, the required minimum scrambled sequence length, execution time, and detection probability.

This paper is organized as follows: Section II summarizes the system model. Section III first shows how to obtain the scrambling sequence by removing the message and the spreading code in the scrambled sequence. And we then present the algorithm for estimating the feedback polynomial and the initial state of the scrambler from the obtained scrambling sequence. Section IV presents estimation performance of the algorithm and Section V presents conclusions.

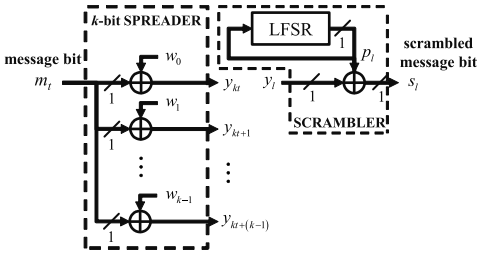


FIGURE 1. Simplified system model for the spreader and scrambler.

II. SYSTEM SETUP

Fig. 1 shows a typical spreader and scrambler in a DSSS system, where we assume a synchronous scrambler with an m-sequence that is the output of the LFSR as a scrambling sequence $(p_l)_{l\geq 0}$. In Fig. 1, a message bit m_t is spread into $(y_i)_{i=kt}^{kt+(k-1)}$ by an arbitrary k-bit spreading code $\left(w_j\right)_{j=0}^{k-1}$, that is

$$y_{kt+j} = m_t \oplus w_j, \quad \text{for } 0 \leq j \leq k-1 \tag{1}$$

where $\oplus$ denotes the modulo-2 addition (exclusive-or). Note that all the k bits in the spreading code $\left(w_j\right)_{j=0}^{k-1}$ are modulo-2 added with the message bit m_t. Therefore, the input sequence of the synchronous scrambler $(y_l)_{l\geq 0}$ consists of repetitions of the spreading code $\left(w_j\right)_{j=0}^{k-1}$ or $\left(w_j \oplus 1\right)_{j=0}^{k-1}$, since a sequence $(y_i)_{i=kt}^{kt+(k-1)}$ of length k is a form of spreading code or its complementary form according to the message bit m_t.

Adding a bit y_l in the scrambler input sequence $(y_l)_{l\geq 0}$ to a bit p_l in the scrambling m-sequence $(p_l)_{l\geq 0}$ yields a bit s_l in the scrambled sequence $(s_l)_{l\geq 0}$, i.e.,

$$s_l = y_l \oplus p_l = m_t \oplus w_j \oplus p_l \quad \text{for} \quad l = kt + j. \tag{2}$$

If we obtain a scrambling sequence by removing the message bit m_t and the spreading code bit w_j in (2), it is noteworthy that it is possible to estimate the scrambler parameters for generating p_l by using the well-known Berlekamp-Massey (BM) algorithm [43].

In the following sections, we propose and analyze an algorithm to estimate the feedback polynomial and the initial state of LFSR, obtaining p_l by eliminating m_t and w_j in (2).

III. PROPOSED ALGORITHM

A. CANCELLATION OF THE INPUT SEQUENCE OF THE SCRAMBLER

We investigate how to remove the message bit m_t and spreading code $\left(w_j\right)_{j=0}^{k-1}$ from the scrambled sequence $(s_l)_{l\geq 0}$ to obtain a scrambling sequence $(p_l)_{l\geq 0}$ in (2). Removal relies on the repetitive patterns by the spreading code inherent in the scrambled sequence.

We first show how to remove the message bit. The sequence $(y_i)_{i=kt}^{kt+(k-1)}$, which is generated by adding the message bit m_t to the spreading code $(w_j)_{j=0}^{k-1}$, is an input to the scrambler as in (2). In this case, if we take the modulo-2 addition to the adjacent bits y_{kt+j} and $y_{kt+(j+1)}$, which are generated from the same message bit m_t, we have

$$\begin{aligned} y_{kt+j} \oplus y_{kt+(j+1)} &= (m_t \oplus w_j) \oplus (m_t \oplus w_{j+1}) \\ &= w_j \oplus w_{j+1} \quad \text{for } 0 \le j \le k-2. \end{aligned} \tag{3}$$

The message bit m_t is removed in (3).

On the other hand, for $j = k-1$, the message bits m_t and m_{t+1} are not removed when we perform the modulo-2 addition on two adjacent bits y_{kt+j} and $y_{kt+(j+1)}$ as follows:

$$\begin{aligned} y_{kt+j} \oplus y_{kt+(j+1)} &= y_{kt+(k-1)} \oplus y_{k(t+1)} \\ &= (m_t \oplus w_{k-1}) \oplus (m_{t+1} \oplus w_0) \\ &\quad \text{for } j = k-1. \end{aligned} \tag{4}$$

If the message bits m_t and m_{t+1} are involved as in (4), then a scrambling sequence cannot be obtained. We will show how to remove the message bits m_t and m_{t+1} remaining in (4) in Section III(B).

The process for removing the spreading code is as follows. As in (1), a bit w_j is modulo-2 added to the message bit m_t with period k. In this case, if two bits y_{kt+j} and $y_{k(t+1)+j}$ are modulo-2 added, which are k bits in position apart and generated by the same spreading code bit w_j, then we have

$$y_{kt+j} \oplus y_{k(t+1)+j} = (m_t \oplus w_j) \oplus (m_{t+1} \oplus w_j) = m_t \oplus m_{t+1}. \tag{5}$$

The spreading code bit w_j is removed in (5).

From (3) and (5), we see that, to obtain the scrambling sequence $(p_l)_{l\ge 0}$ by removing the message bits and the spreading code bits in (2), we need 4 input sequence bits. This is one of the key ideas in this paper. By taking modulo-2 additions with the 4 bits of y_l, y_{l+1}, y_{l+k}, and $y_{l+(k+1)}$ for $l = kt + j$ $(0 \le j \le k-2)$, we obtain

$$\begin{aligned} &y_l \oplus y_{l+1} \oplus y_{l+k} \oplus y_{l+(k+1)} \\ &\quad = y_{kt+j} \oplus y_{kt+(j+1)} \oplus y_{k(t+1)+j} \oplus y_{k(t+1)+(j+1)} = 0 \\ &\qquad \text{for } l = kt + j \ (0 \le j \le k-2). \end{aligned} \tag{6}$$

Note that the message bits m_t and m_{t+1} and the spreading code bits w_j and w_{j+1} are simultaneously removed in (6).

Similarly, we define u_l and $\widetilde{p}_l$ as modulo-2 additions with the 4 bits of $s_l, s_{l+1}, s_{l+k}, s_{l+(k+1)}$ for the scrambled sequence and $p_l, p_{l+1}, p_{l+k}, p_{l+(k+1)}$ for the scrambling sequence, respectively, as

$$u_l = s_l \oplus s_{l+1} \oplus s_{l+k} \oplus s_{l+(k+1)} \quad \text{for } l = kt + j \tag{7}$$

$$\widetilde{p}_l = p_l \oplus p_{l+1} \oplus p_{l+k} \oplus p_{l+(k+1)} \quad \text{for } l = kt + j. \tag{8}$$

Then, substituting (2) and (6) into (7), yield

$$\begin{aligned} u_l &= s_l \oplus s_{l+1} \oplus s_{l+k} \oplus s_{l+(k+1)} \\ &= p_l \oplus p_{l+1} \oplus p_{l+k} \oplus p_{l+(k+1)} \\ &= \widetilde{p}_l \quad \text{for } l = kt + j\,(0 \le j \le k-2)\,. \end{aligned} \tag{9}$$

From (9), we see that u_l is equal to $\widetilde{p}_l$, and this will play an important role in estimation of scrambler parameters.

Note that, $(\widetilde{p}_l)_{l\ge 0}$ obtained in (9) is another shifted version of the scrambling m-sequence $(p_l)_{l\ge 0}$ due to the "shift and add property" of the m-sequence. Consequently, the feedback polynomial of the LFSR for $(\widetilde{p}_l)_{l\ge 0}$ is the same as that of $(p_l)_{l\ge 0}$.

Eq. (9) is valid for $l = kt + j\,(0 \le j \le k-2)$; however, it is not guaranteed for $l = kt + j$ $(j = k-1)$. Therefore, the maximum bit length of the possible consecutive sequence of $\widetilde{p}_l$ that can be obtained from (9) is $k-1$. Thus, when the degree of the feedback polynomial of the scrambler is n, if $k - 1 \ge 2n$, then the feedback polynomial of the scrambler can be estimated by using the BM algorithm directly from (9). On the other hand, if $k - 1 < 2n$, it is difficult to obtain the feedback polynomial of the scrambler by using the BM algorithm. In the following subsection, to solve this problem, we propose a method to obtain the consecutive sequence $(\widetilde{p}_l)_{l\ge 0}$ for $l = kt + j$ $(0 \le j \le k-1)$.

B. ESTIMATION OF THE FEEDBACK POLYNOMIAL OF THE LFSR FOR SYNCHRONOUS SCRAMBLER

In subsection III(A), the message bits m_t and m_{t+1} are removed for $l = kt + j$ $(0 \le j \le k-2)$ in (3); however, they are left in place for $l = kt + j$ $(j = k-1)$ in (4). Therefore, the consecutive scrambling sequence $(\widetilde{p}_l)_{l\ge 0}$ obtained in (9) is valid for $l = kt + j\,(0 \le j \le k-2)$. In this subsection, to obtain the scrambling sequence of $(\widetilde{p}_l)_{l\ge 0}$ valid for $l = kt + j$ $(0 \le j \le k-1)$, we propose an additional method to (9) by using the linearity among the scrambling sequence bits.

To do this, we define decimation and show the linear relation among the scrambling sequence bits in Theorem 1.

Definition 1 [44]: Let $(u_l)_{l\ge 0}$ and $(v_l)_{l\ge 0}$ be the sequences and d be the positive integer. We define the d-bit decimated sequence $(v_l)_{l\ge 0}$ of $(u_l)_{l\ge 0}$ as $v_l = u_{dl}$.

Theorem 1: Let n be the degree of the primitive feedbackpolynomial of LFSR that generates the sequence $(p_i)_{i\ge 0}$, and $(p_{ki})_{i\ge 0}$ be the k-bit decimated sequence of $(p_i)_{i\ge 0}$. If the degree of a minimal polynomial that generates $(p_{ki})_{i\ge 0}$ is n, then the bits of the sequence $(p_i)_{i\ge 0}$ are represented as the linear combination of $(p_0, p_k, \cdots, p_{(n-1)k})$ with unique linear relation coefficients $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$ by

$$p_{i+1} = e_n p_i \oplus e_{n-1} p_{i+k} \oplus \cdots \oplus e_1 p_{i+(n-1)k}. \tag{10}$$

The proof of Theorem 1 is in Appendix A. Eq. (10) of Theorem 1 is another key idea in the paper. To obtain the scrambling sequence of $(\widetilde{p}_l)_{l\ge 0}$ valid for $l = kt + j$ $(0 \le j \le k-1)$, that is, to obtain the scrambling sequence bit $\widetilde{p}_l$ in place for $l = kt + j\,(j = k-1)$, we have to obtain the linear relation coefficients in (10).

First, we set the system of linear equations about the linear relation coefficients to obtain the linear relation coefficients $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$. If all the subscripts of $\widetilde{p}_l$ bits in (10) are in the range of $l = kt + j\,(0 \le j \le k-2)$, then,

the unknowns in (10) will be only the linear relation coefficients because we can obtain the bits $\widetilde{p}_l$ for $l = kt + j$ $(0 \le j \le k-2)$ from (9).

In this case, by substituting $\widetilde{p}_{kt}$ for p_i in (10), we can generate the linear equations about the linear relation coefficients $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$ as follows:

$$\widetilde{p}_{kt+1} = e_n\widetilde{p}_{kt} \oplus e_{n-1}\widetilde{p}_{k(t+1)} \oplus \cdots \oplus e_1\widetilde{p}_{k(n+t-1)}. \quad (11)$$

Since all the subscripts l of $\widetilde{p}_l$ in (11) are in the range of $l = kt + j\,(0 \le j \le k-2)$, substituting (9) into (11) yields

$$u_{kt+1} = e_n u_{kt} \oplus e_{n-1}u_{k(t+1)} \oplus \cdots \oplus e_1 u_{k(t+n-1)}. \quad (12)$$

There are n unknowns $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$ in (12) because u_l can be obtained by modulo-2 additions with the 4 bits of $s_l, s_{l+1}, s_{l+k}, s_{l+(k+1)}$ as in (7). These unknowns can be determined by solving the system of n linear equations.

Finally, by substituting $\widetilde{p}_l$ in (9), which is equal to u_l for $l = kt + j\,(0 \le j \le k-2)$, and the obtained value of $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$ in (12) into (10), we can calculate the $\widetilde{p}_l$ for $l = kt + j$ $(j = k-1)$ with

$$\widetilde{p}_{kt+(k-1)} = e_n u_{kt+k-2} \oplus e_{n-1}u_{kt+2k-2} \oplus \cdots \oplus e_1 u_{kt+nk-2} \quad \text{for } t \ge 0. \quad (13)$$

Note that, using (9) and (13), we can have the consecutive scrambling sequence of $(\widetilde{p}_l)_{l\ge 0}$ for $l = kt+j$ $(0 \le j \le k-1)$. By using the obtained scrambling sequence of $(\widetilde{p}_l)_{l\ge 0}$, we can obtain the feedback polynomial of the scrambler with the BM algorithm, where we need a $2n$-bit length scrambling sequence of $(\widetilde{p}_l)_{l\ge 0}$.

Using the above results, we summarize the estimation of the feedback polynomial of LFSR as the following three steps.

1) By applying (10) into the scrambling sequence $(\widetilde{p}_l)_{l\ge 0}$ for $l = kt+j$ $(0 \le j \le k-2)$, determine the coefficients $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$.
2) By using the obtained linear relation coefficients in Step 1, and $u_l = \widetilde{p}_l$ for $l = kt + j\,(0 \le j \le k-2)$, find $\widetilde{p}_l$ for $l = kt + j\,(j = k-1)$.
3) By applying $(\widetilde{p}_l)_{l\ge 0}$ into the BM algorithm, estimate the feedback polynomial of LFSR.

C. ESTIMATION OF INITIAL STATE OF THE LFSR FOR SYNCHRONOUS SCRAMBLER

Now, we estimate the initial state of the LFSR, the other of the scrambler parameters. Let $c_n x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + 1$ be the feedback polynomial obtained in Section III(B). Then, the bitwise relations in the scrambling sequence can be expressed as

$$p_{l+n} = c_n p_l \oplus c_{n-1}p_{l+1} \oplus \cdots \oplus c_1 p_{l+n-1} \quad \text{for } l \ge -n \quad (14)$$

where $c_1, \cdots, c_n \in \{0, 1\}$ are the coefficients of the feedback polynomial.

Note that there is a linear relation between the scrambling sequence and the initial state because the initial state becomes the part of the scrambling sequence. After constructing a system of linear equations by using the linear relation, it is possible to estimate the initial state by solving the system of linear equations with Gaussian elimination.

We propose Theorem 2, and the approach to obtain the initial state follows from the theorem.

Theorem 2: For $(p_l)_{l\ge 0}$, which is the output sequence of an LFSR, each bit p_l is represented as the linear combination of $(p_{-1}, p_{-2}, \cdots, p_{-n})$, which is the initial state of the LFSR, as follows:

$$p_l = e_n^{(l)}p_{-n} \oplus e_{n-1}^{(l)}p_{1-n} \oplus e_{n-2}^{(l)}p_{2-n} \oplus \cdots \oplus e_1^{(l)}p_{-1}. \quad (15)$$

Note that, $\left(e_1^{(l+1)}, e_2^{(l+1)}, \cdots, e_n^{(l+1)}\right) \in \{0, 1\}^n$, which are the coefficients of representing p_{l+1} as the linear combination of $(p_{-1}, p_{-2}, \cdots, p_{-n})$, are unique for a given l and uniquely determined from $\left(e_1^{(l)}, e_2^{(l)}, \cdots, e_n^{(l)}\right) \in \{0, 1\}^n$, which are the coefficients of representing p_l as the linear combination of $(p_{-1}, p_{-2}, \cdots, p_{-n})$ as follows:

$$e_i^{(l+1)} = e_{i+1}^{(l)} \oplus e_1^{(l)}c_i \quad \text{for } 1 \le i \le n, \;\; e_{n+1}^{(l)} = 0. \quad (16)$$

To the best of our knowledge, Theorem 2 has not been reported in the literature. We give a proof of Theorem 2 in Appendix B. Theorem 2 is one of the clues for estimation of the initial state of LFSR.

We set the system of linear equations for the initial state by using Theorem 2. Note that we have already obtained the coefficients $c_1, \cdots, c_n \in \{0, 1\}$ of the feedback polynomial in Section III(B). When $l = -n$, (14) becomes

$$p_0 = c_n p_{-n} \oplus c_{n-1}p_{1-n} \oplus \cdots \oplus c_1 p_{-1}. \quad (17)$$

When $l = 0$, (15) becomes

$$p_0 = e_n^{(0)}p_{-n} \oplus e_{n-1}^{(0)}p_{1-n} \oplus e_{n-2}^{(0)}p_{2-n} \oplus \cdots \oplus e_1^{(0)}p_{-1}. \quad (18)$$

We see that, from (17) and (18), $\left(e_1^{(0)}, e_2^{(0)}, \cdots, e_n^{(0)}\right) \in \{0, 1\}^n$, which are the coefficients representing p_0 as the linear combination of the initial state, satisfy the condition:

$$e_i^{(0)} = c_i \quad \text{for } 1 \le i \le n. \quad (19)$$

For any integer l, we can obtain $\left(e_1^{(l)}, e_2^{(l)}, \cdots, e_n^{(l)}\right) \in \{0, 1\}^n$, which are the coefficients representing p_l with the initial state $(p_{-1}, p_{-2}, \cdots, p_{-n})$ in (15), by mathematical induction from (16) and (19).

Recall that in (9), when $l = kt + j\,(0 \le j \le k-2)$, $u_l = s_l \oplus s_{l+1} \oplus s_{l+k} \oplus s_{l+(k+1)} = p_l \oplus p_{l+1} \oplus p_{l+k} \oplus p_{l+(k+1)}$ holds. Therefore, by substituting (15) into (9) with the obtained coefficients $\left(e_1^{(l)}, e_2^{(l)}, \cdots, e_n^{(l)}\right) \in \{0, 1\}^n$, we can obtain the linear equations about $(p_{-1}, p_{-2}, \cdots, p_{-n})$:

$$\begin{aligned} u_l &= p_l \oplus p_{l+1} \oplus p_{l+k} \oplus p_{l+(k+1)} \\ &= \left(e_n^{(l)}p_{-n} \oplus e_{n-1}^{(l)}p_{1-n} \oplus e_{n-2}^{(l)}p_{2-n} \oplus \cdots \oplus e_1^{(l)}p_{-1}\right) \\ &\quad \oplus \left(e_n^{(l+1)}p_{-n} \oplus e_{n-1}^{(l+1)}p_{1-n} \oplus e_{n-2}^{(l+1)}p_{2-n}\right. \\ &\quad \left. \oplus \cdots \oplus e_1^{(l+1)}p_{-1}\right) \\ &\quad \oplus \left(e_n^{(l+k)}p_{-n} \oplus e_{n-1}^{(l+k)}p_{1-n} \oplus e_{n-2}^{(l+k)}p_{2-n}\right. \end{aligned}$$

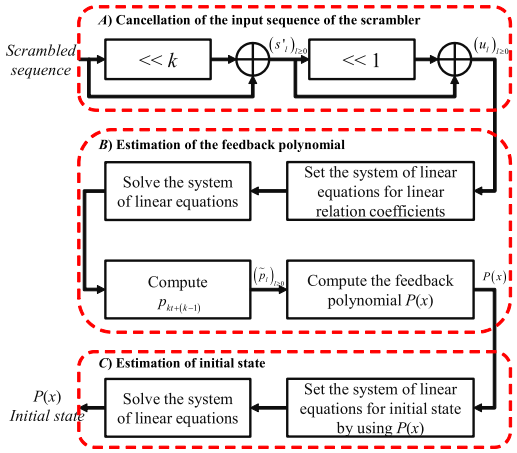


FIGURE 2. Block diagram of the proposed method.

$$
\begin{aligned}
&\oplus \cdots \oplus e_1^{(l+k)} p_{-1}\Big) \\
&\oplus \Big(e_n^{(l+(k+1))} p_{-n} \oplus e_{n-1}^{(l+(k+1))} p_{1-n} \oplus e_{n-2}^{(l+(k+1))} p_{2-n} \\
&\oplus \cdots \oplus e_1^{(l+(k+1))} p_{-1}\Big) \\
&= \left(e_n^{(l)} \oplus e_n^{(l+1)} \oplus e_n^{(l+k)} \oplus e_n^{(l+(k+1))}\right) p_{-n} \\
&\oplus \left(e_{n-1}^{(l)} \oplus e_{n-1}^{(l+1)} \oplus e_{n-1}^{(l+k)} \oplus e_{n-1}^{(l+(k+1))}\right) p_{1-n} \\
&\oplus \cdots \oplus \left(e_1^{(l)} \oplus e_1^{(l+1)} \oplus e_1^{(l+k)} \oplus e_1^{(l+(k+1))}\right) p_{-1} \\
&\quad \text{for } l = kt + j\,(0 \le j \le k-2)\,. \qquad (20)
\end{aligned}
$$

Since the unknown in (20) is only the initial state $(p_{-1}, p_{-2}, \cdots, p_{-n})$, by using it, we construct a system of linear equations for finding the initial state. We set the system of n linear equations from (20) and solve them by using Gaussian elimination. We then obtain $(p_{-1}, p_{-2}, \cdots, p_{-n})$, the initial state of the scrambler. Note that there is a rare case where the solution is not unique because of the dependency of the equations. In this case, we add more linear equations by (20) into the system of linear equations, so that the resulting system has a unique solution.

D. ESTIMATION OF THE SYNCHRONOUS SCRAMBLER

We present the results of the previous subsections as the block diagram of the proposed estimation process in Fig. 2, and finally summarize the estimation of synchronous scrambler parameters, the feedback polynomial and the initial state, in Algorithm 1.

In Step 3 of Algorithm 1, to get the unique linear relation coefficients $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$, the number of input bits of Algorithm 1 should be at least $(2n-1)k + 2$. To make the scrambled sequence length of $(s_l)_{l\ge 0}$ longer than $(2n-1)k + 2$, we have to set n_{Th} to the upper bound of n, and set the minimum scrambled sequence length $\min_{len}$ to $(2n_{Th}-1)k + 2$. The reason for setting the minimum scrambled sequence length to $(2n_{Th}-1)k + 2$ will be explained in Section IV.

Here is a simple example to show how Algorithm 1 works.

Example 1) We assume the following:

Algorithm 1 Estimation of the Synchronous Scrambler in Direct Sequence Spread Systems

Input:
- $(s_l)_{l\ge 0}$: spread sequence longer than $\min_{len}$ bits
- k: spreading code length
- n_{Th} : upper bound of the feedback polynomial degree

Output:
- $P(x)$: feedback polynomial
- $(p_{-n}, p_{1-n}, \cdots, p_{-1})$: initial state

1. $s'_l \leftarrow s_l \oplus s_{l+k}$
2. $u_l \leftarrow s'_l \oplus s'_{l+1}$
3. From (12), generate and solve the system of linear equations to compute linear relation coefficients $e_n, e_{n-1}, \cdots, e_1 \in \{0, 1\}$
4. Compute $\widetilde{p}_{kt+(k-1)}$ for $t \ge 0$ according to (13)
5. Compute $P(x)$ by using the BM algorithm for $(\widetilde{p}_l)_{l\ge 0}$
6. Set the system of linear equations using (20) and get the initial state by using Gaussian elimination

The feedback polynomial of the scrambler:

$$P(x) = x^5 + x^3 + 1$$

The initial state of the scrambler: (01100)
Spreading code: (110)
Message sequence: (011 100 011 11…)
Input of the scrambler: message spread by the spreading code

$$n_{Th} : 5$$

In this case, the scrambled sequence $(s_l)_{l\ge 0}$, which is made by the modulo-2 addition of the input sequence $(y_l)_{l\ge 0}$ and the scrambling sequence $(p_l)_{l\ge 0}$ becomes

$$
\begin{aligned}
(s_l)_{l\ge 0} &= (y_l)_{l\ge 0} \oplus (p_l)_{l\ge 0} \\
&= (1101100010011101101100010010 01\ldots) \\
&\quad \oplus (1111100011011101010000100101 10\ldots) \\
&= (0010000001000000111100110111 11\ldots).
\end{aligned}
$$

These are the steps of Algorithm 1:

1.

$$
\begin{aligned}
(s'_l)_{l\ge 0} &= (s_l)_{l\ge 0} \oplus (s_{l+3})_{l\ge 0} \\
&= (y_l \oplus y_{l+3})_{l\ge 0} \oplus (p_l \oplus p_{l+3})_{l\ge 0} \\
&= (0001110001110000001110000000\ldots) \\
&\quad \oplus (0011111000110111010100001 00\ldots) \\
&= (0010001001000111011010001 00\ldots)
\end{aligned}
$$

2.

$$
\begin{aligned}
(u_l)_{l\ge 0} &= (s'_l)_{l\ge 0} \oplus (s'_{l+1})_{l\ge 0} \\
&= ((y_l \oplus y_{l+3}) \oplus (y_{l+1} \oplus y_{l+4}))_{l\ge 0} \oplus ((p_l \oplus p_{l+3}) \\
&\quad \oplus (p_{l+1} \oplus p_{l+4}))_{l\ge 0} \\
&= (0010010010010000010010000 0\ldots) \\
&\quad \oplus (0100001001011001111100011 0\ldots) \\
&= (0110011011001001101110011 0\ldots)
\end{aligned}
$$

3.

$$M = \left(\begin{array}{ccccc|c} u_0 & u_3 & u_6 & u_9 & u_{12} & u_1 \\ u_3 & u_6 & u_9 & u_{12} & u_{15} & u_4 \\ u_6 & u_9 & u_{12} & u_{15} & u_{18} & u_7 \\ u_9 & u_{12} & u_{15} & u_{18} & u_{21} & u_{10} \\ u_{12} & u_{15} & u_{18} & u_{21} & u_{24} & u_{13} \end{array}\right)$$
$$= \left(\begin{array}{ccccc|c} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{array}\right)$$

% Generation of the matrix M from (12)

$$M' = \left(\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

% Computation of M' by using Gaussian elimination

$$(e_5, e_4, e_3, e_2, e_1) = (0, 1, 1, 0, 0)$$

$$\begin{aligned} \widetilde{p}_{t+1} &= 0 \cdot \widetilde{p}_t \oplus 1 \cdot \widetilde{p}_{t+k} \oplus 1 \cdot \widetilde{p}_{t+2k} \oplus 0 \cdot \widetilde{p}_{t+3k} \oplus 0 \cdot \widetilde{p}_{t+4k} \\ &= \widetilde{p}_{t+k} \oplus \widetilde{p}_{t+2k} \end{aligned} \tag{21}$$

% Derivation of (21) from the linear relation coefficients $(e_5, e_4, e_3, e_2, e_1)$.

4. $(\widetilde{p}_l)_{l\geq 0} = (01\underline{0}\ 00\underline{0}\ 10\underline{0}\ 10\ldots)$

% Computation of the underlined $\widetilde{p}_{kt+(k-1)}$ by (21) from the linear relation coefficients $(e_5, e_4, e_3, e_2, e_1)$.

5. $P(x) = x^5 + x^3 + 1$

% Computation of the feedback polynomial of the scrambler by using the BM algorithm for $(\widetilde{p}_l)_{l\geq 0}$.

6. $p_{-5} = 0, p_{-4} = 1, p_{-3} = 1, p_{-2} = 0, p_{-1} = 0$ from

$$\begin{cases} u_0 = 0 = (0, 0, 0, 1, 1) \cdot (p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})^T \\ u_1 = 1 = (1, 0, 1, 0, 1) \cdot (p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})^T \\ u_3 = 0 = (0, 1, 1, 1, 1) \cdot (p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})^T \\ u_4 = 0 = (1, 0, 0, 1, 1) \cdot (p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})^T \\ u_6 = 1 = (1, 1, 0, 1, 0) \cdot (p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})^T \\ u_7 = 0 = (0, 1, 1, 0, 1) \cdot (p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})^T, \end{cases}$$

$$\left(\begin{array}{ccccc|c} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{array}\right)$$

$$\xrightarrow{G.E.} \left(\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

TABLE 1. Computational complexity of algorithm 1 for bitwise operation.

Step	Required number of bitwise operations
1	$(2n_{Th}-2)k+2$
2	$(2n_{Th}-2)k+1$
3	$O(n_{Th}^3)$ for Gaussian elimination
4	$(n_{Th}-1)\cdot\left\lfloor\frac{2n_{Th}}{k}\right\rfloor$
5	$O(n_{Th}^2)$ for the BM algorithm
6	$O(n_{Th}^3)$ for Gaussian elimination
Total	$O(2n_{Th}^3+11n_{Th}^2)$

For the total complexity, we assume that $k < 2n_{Th}$.

% Generation of the system of linear equations for the initial state $(p_{-5}, p_{-4}, p_{-3}, p_{-2}, p_{-1})$ and solution of the initial state by using Gaussian elimination.

Note that from the results of Example 1, we obtain the feedback polynomial $P(x) = x^5 + x^3 + 1$ and the initial state of the scrambler (01100).

IV. PERFORMANCE OF THE ALGORITHM

In this section, we investigate the estimation performance of the proposed algorithm in terms of computational complexity, the required minimum scrambled sequence length, execution time, and the detection probability.

We first examine the computational complexity, the required minimum scrambled sequence length, and execution time of Algorithm 1 for the error-free case. Then we apply the Algorithm 1 to a noisy channel with statistical technique and show the detection probability in an additive white Gaussian noise (AWGN) channel.

Regarding the computational complexity of Algorithm 1, the numbers of bitwise operations in Step 1 and Step 2, are determined by the required minimum number of bits to generate a matrix for Gaussian elimination in Step 3. The number of bitwise operations in Step 4 is calculated by the required minimum input length $2n_{Th}$ for the BM algorithm in Step 5, where the BM algorithm has a complexity $O\left(n_{Th}^2\right)$ for input sequence length $2n_{Th}$ in Step 5. For Step 3 and Step 6, the computational complexities become $O\left(n_{Th}^3\right)$ for $n_{Th} \times n_{Th}$ matrix by Gaussian eliminations. We summarize the computational complexities of each step and the total computational complexity for the bitwise operations of Algorithms 1 in Table 1, where $\lfloor\cdot\rfloor$ is the floor function.

Table 1 shows that the proposed algorithm requires $O\left(2n_{Th}^3 + 11n_{Th}^2\right)$ bitwise operations. Unlike conventional methods having exponential computational complexity, the proposed algorithm can estimate the feedback polynomial and the initial state of a scrambler effectively with polynomial computational complexity, because the algorithm is based not on a full search for sparse multiples of the feedback polynomial, nor on huge matrix computation for EVD, but on the bitwise relations inherent in the scrambling sequence itself.

For a practical application, the minimum length of a scrambled sequence required to estimate the scrambler

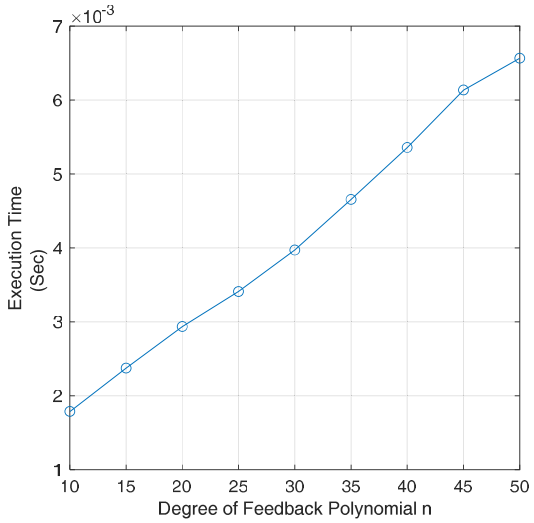


FIGURE 3. Execution time of Algorithm 1.

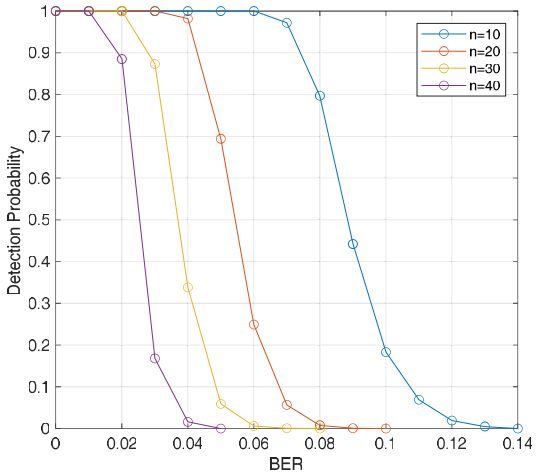


FIGURE 4. Detection probability versus BER for n in AWGN channel when $L = 1000$.

should be also considered. The required minimum length of a scrambled sequence for Algorithm 1 is determined by considering the BM algorithm in Step 5. To obtain $2n_{Th}$-bit input for the BM algorithm in Step 5, we should generate a $2n_{Th}$-bit sequence of $(\widetilde{p}_l)_{l\geq 0}$ in Step 4. For this, in Step 3, we need to generate and solve the system of linear equations from (12) with at least $(2n_{Th}-2)\,k+2$ bits. To get $(2n_{Th}-2)\,k+2$-bit input for Step 3, we need a scrambled sequence of $(2n_{Th}-1)\,k+2$-bit length in Step 1. Therefore, for Algorithm 1, the required minimum scrambled sequence length is $(2n_{Th}-1)\,k+2$ bits.

To examine the effectiveness of the proposed algorithm, we execute simulations with an Intel® Core™i7-6700K CPU of 4.00GHz and 64 GB RAM, and depict execution time in Fig. 3. The x-axis and y-axis represent the degree of feedback polynomial n and time (second), respectively, and execution time is measured by the average time of 10 000 runs for m-sequence of length 2^5-1 bits as a spreading code. In Fig. 3, we see that the proposed algorithm can estimate the feedback polynomial and initial state of the synchronous scrambler in 0.007 second even when the degree of feedback polynomial n is 50.

We also implemented and examined the algorithm for the various LFSR parameters and spreading codes: the spreading codes such as the m-sequence, Gold code [45], Kasami code [46], Walsh code [47], and OVSF code [48], and the LFSR parameters such as the primitive polynomials up to a degree of 50.

The proposed algorithm can be applied to a noisy channel with a statistical technique. If there are errors in the input sequence to the BM algorithm in Step 5 due to noise, it becomes highly probably that the estimated degree of the output feedback polynomial will be close to n_{Th}, which is the upper bound of the feedback polynomial degree in the algorithm. Otherwise, the degree of the output feedback polynomial becomes n, where $n < n_{Th}$. Therefore, in a noisy channel, we can estimate the feedback polynomial by repeating the algorithm L times and by comparing degrees of the output feedback polynomials where L is the number of repetitions.

We summarize the estimation of the feedback polynomial of LFSR in a noisy channel as the following steps.

1) After repeating the algorithm L times, select the polynomials that have appeared two or more times.
2) Find the smallest degree N in the selected polynomials.
3) Choose the candidate polynomials having the degree $[N, N+\alpha)$ from the selected polynomials in Step 1, where α is a design parameter.
4) Select the most commonly occurring polynomial from the candidate polynomials in Step 3.

We can also estimate the initial state of the LFSR by using the same method. The computational complexity, the required minimum scrambled sequence length, and execution time in a noisy channel increase L times compared to noise-free case. Nonetheless, they still have polynomial computational complexity, not exponential computational complexity.

To verify this, we simulate the proposed method in a noisy channel. For simulations, we set n_{Th} to $n+5$, α to 4, and spreading code length to 2^5-1, where we assume binary phase shift keying modulation and an AWGN channel. Fig. 4 depicts detection probabilities for the various values of degrees n when the number of repetitions L is 1000, and Fig. 5 shows detection probabilities for the various values of repetitions L when the degree of feedback polynomial n is 10, according to bit error rate (BER).

In Fig. 4, we find that detection probabilities of the proposed method can reach 0.9 at BERs of 0.08, 0.05, 0.03, and 0.02 for degrees $n = 10$, 20, 30, and 40, respectively. In Fig. 5, we see that detection probabilities of the proposed method increase as the number of repetitions L increases.

To validate the proposed algorithm, we compare the results of Algorithm 1 with the results of [32] and [34] in Fig. 6 for the various BERs in an AWGN channel, when the degree of feedback polynomial n is 8 and spreading code length k is 2^3-1, showing the number of bits available for reconstruction

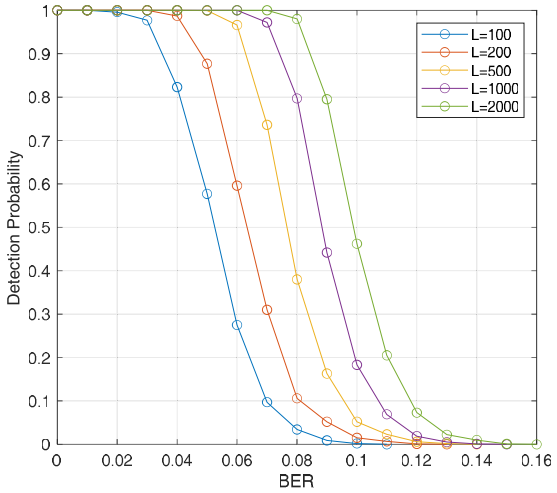
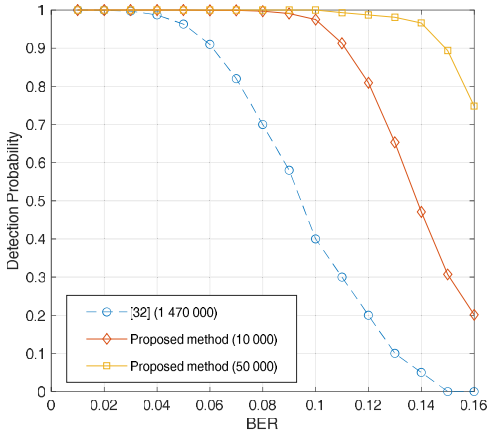
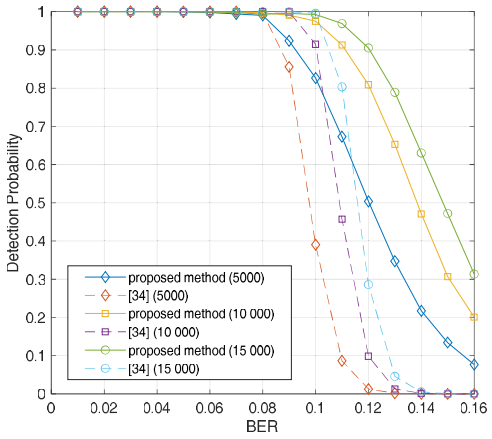


FIGURE 5. Detection probability versus BER for L in AWGN channel when $n = 10$.



(a) Comparison of the proposed method and the method of [32]



(b) Comparison of the proposed method and the method of [34]

FIGURE 6. Comparison of detection probabilities versus BER in AWGN channel when n = 8.

in parentheses. Fig. 6 (a) shows the comparison of detection probabilities from the proposed algorithm and [32]. In Fig. 6 (a), the number of bits available for reconstruction for [32] is set to 1 470 000 from [Eq. (26), 32] and those for the proposed method are set to 10 000 and 50 000 since our method requires far fewer bits. In Fig. 6 (b), we compare the results of the proposed Algorithm 1 with the results of [34] for the various numbers of bits available for reconstruction and BERs. From Figs. 6 (a) and (b) we see that our algorithm has a better detection performance than the methods of [32] and [34].

In this paper, we have considered the scrambling sequence as an m-sequence, but the proposed algorithm can straightforwardly be applied to non-maximal length scrambling sequences, such as Gold code and Kasami code, because these codes are generated by m-sequences. For example, Gold code with the length of $2^n - 1$ bits is generated by two preferred m-sequences with the primitive polynomials of degree n, and then we can consider that Gold code is generated by the LFSR with the multiplied polynomial of the two primitive polynomials of degree n. Therefore, we can find the feedback polynomial of degree $2n$ used for Gold code with the proposed algorithm by setting $n_{Th} > 2n$. Similarly, Kasami code with the length of $2^n - 1$ bits is generated by decimation and modulo-2 addition of an m-sequence with the primitive polynomial of degree n, and we can consider Kasami code to be generated by the LFSR with a feedback polynomial of degree $1.5n$. Therefore, we can find the feedback polynomial of Kasami code with the proposed algorithm by setting $n_{Th} > 1.5n$. Also, in this paper, we assumed that the duration of an information bit is equal to the period of the spreading code. However, the proposed algorithm can also be applied when that assumption does not hold. This is because the proposed method can also cancel message bits and spreading code bits even when the duration of an information bit is not equal to the period of the spreading code.

V. CONCLUSION

In this paper, we proposed a novel algorithm for estimating the parameters, the feedback polynomial, and the initial state of a synchronous scrambler in DSSS systems. We analyzed the estimation performance in terms of the computational complexity, the required minimum scrambled sequence length, execution time, and detection probability.

Unlike conventional algorithms having exponential computational complexity, the proposed algorithm could estimate practically the synchronous scrambler parameters, the feedback polynomial, and the initial state of the scrambler, based on the bitwise linear relations inherent in the scrambling sequence itself, with polynomial computational complexity.

In the proposed algorithm, most of the input sequence of the scrambler in the scrambled sequence could be removed by using the repeated patterns of the input sequence inherent in the scrambled sequence and the ''shift and add property'' of the m-sequence. To remove the remained input sequence in the scrambled sequence and get the scrambling m-sequence, we used the linear relations made by feedback polynomial among the m-sequence bits used for scrambling. From the algorithm, we were able to estimate the scrambler parameters and reconstruct the scrambling sequence.

The proposed algorithm could be applied to noisy channels with a statistical technique and could straightforwardly be

applied to non-maximal length scrambling sequences, such as Gold code and Kasami code.

APPENDIX

A. PROOF OF THEOREM 1

Since there are linear relationships among every n +1 bits of $(p_t)_{t\geq 0}$, if the degree of the minimal polynomial that generates the sequence $(p_{kt})_{t\geq 0}$ is n, then there is an $n \times n$ non-singular matrix $M \in \{0,1\}^{n\times n}$, which satisfies the following equation.

$$M \cdot \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{pmatrix} = \begin{pmatrix} p_0 \\ p_k \\ \vdots \\ p_{(n-1)k} \end{pmatrix}. \tag{22}$$

(If M is singular, then each entry of $\left(p_0, p_k, \cdots, p_{(n-1)k}\right)^T$ is represented by the number of bits less than n, where T denotes the transposition. Therefore, the degree of the minimal polynomial for LFSR, which generates the sequence $(p_{kt})_{t\geq 0}$, will be smaller than n, which is a contradiction. Therefore, M is non-singular.)

Since M is non-singular, there is an inverse matrix M^{-1} of M, and (22) is equivalent to the following (23).

$$\begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} p_0 \\ p_k \\ \vdots \\ p_{(n-1)k} \end{pmatrix}. \tag{23}$$

From (23), p_1 is uniquely represented as the linear combination of $\left(p_0, p_k, \cdots, p_{(n-1)k}\right)$. Similarly, for some t, p_{t+1} is also uniquely represented as a linear combination of $\left(p_t, p_{t+k}, \cdots, p_{t+(n-1)k}\right)$ □

B. PROOF OF THEOREM 2

We prove by mathematical induction. In (14), for $l = -n$, p_0 is represented as the linear combination of the initial state $(p_{-n}, p_{1-n}, \cdots, p_{-1})$ as follows:

$$p_0 = c_n p_{-n} \oplus c_{n-1} p_{1-n} \oplus \cdots \oplus c_1 p_{-1} \tag{24}$$

And for an arbitrary non-negative integer l, suppose p_l is represented as the linear combination of $e_1^{(l)}, e_2^{(l)}, \cdots, e_n^{(l)} \in \{0, 1\}$ and the initial state $(p_{-n}, p_{1-n}, \cdots, p_{-1})$ as follows:

$$p_l = e_n^{(l)} p_{-n} \oplus e_{n-1}^{(l)} p_{1-n} \oplus \cdots \oplus e_2^{(l)} p_{-2} \oplus e_1^{(l)} p_{-1} \tag{25}$$

Then p_{l+1} is represented for $(p_{1-n}, p_{2-n}, \cdots, p_0)$ as follows:

$$p_{l+1} = e_n^{(l)} p_{1-n} \oplus e_{n-1}^{(l)} p_{2-n} \oplus \cdots \oplus e_2^{(l)} p_{-1} \oplus e_1^{(l)} p_0 \tag{26}$$

Substituting (24) into (26), we obtain

$$\begin{aligned} p_{l+1} &= e_n^{(l)} p_{1-n} \oplus e_{n-1}^{(l)} p_{2-n} \oplus \cdots \oplus e_2^{(l)} p_{-1} \oplus e_1^{(l)} p_0 \\ &= e_1^{(l)} c_n p_{-n} \oplus \left(e_n^{(l)} \oplus e_1^{(l)} c_{n-1}\right) p_{1-n} \\ &\quad \oplus \left(e_{n-1}^{(l)} \oplus e_1^{(l)} c_{n-2}\right) p_{2-n} \\ &\quad \oplus \cdots \oplus \left(e_2^{(l)} \oplus e_1^{(l)} c_1\right) p_{-1} \end{aligned} \tag{27}$$

From (27), p_{l+1} is represented as the linear combination of the initial state $(p_{-n}, p_{1-n}, \cdots, p_{-1})$ with $e_i^{(l+1)} = e_{i+1}^{(l)} \oplus e_1^{(l)} c_i$ for $1 \leq i \leq n$ and $e_{n+1}^{(l)} = 0$ in (16). □

REFERENCES

[1] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*. Hoboken, NJ, USA: Wiley, 2015.
[2] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Codec and GOP identification in double compressed videos," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2298–2310, May 2016.
[3] D. Vazquez-Padin, M. Fontani, D. Shullani, F. Perez-Gonzalez, A. Piva, and M. Barni, "Video integrity verification and GOP size estimation via generalized variation of prediction footprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1815–1830, 2020.
[4] X. Jiang, P. He, T. Sun, F. Xie, and S. Wang, "Detection of double compression with the same coding parameters based on quality degradation mechanism analysis," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 170–185, Jan. 2018.
[5] B. Kwon, H. Song, and S. Lee, "Accurate blind lempel-ziv-77 parameter estimation via 1-D to 2-D data conversion over convolutional neural network," *IEEE Access*, vol. 8, pp. 43965–43979, 2020.
[6] X. Liang, Z. Li, Y. Yang, Z. Zhang, and Y. Zhang, "Detection of double compression for HEVC videos with fake bitrate," *IEEE Access*, vol. 6, pp. 53243–53253, 2018.
[7] R. Moosavi and E. G. Larsson, "Fast blind recognition of channel codes," *IEEE Trans. Commun.*, vol. 62, no. 5, pp. 1393–1405, May 2014.
[8] A. Bonvard, S. Houcke, R. Gautier, and M. Marazin, "Classification based on Euclidean distance distribution for blind identification of error correcting codes in noncooperative contexts," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2572–2583, May 2018.
[9] R. Swaminathan, A. S. Madhukumar, and G. Wang, "Blind estimation of code parameters for product codes over noisy channel conditions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1460–1473, Apr. 2020.
[10] A. D. Yardi, S. Vijayakumaran, and A. Kumar, "Blind reconstruction of binary cyclic codes from unsynchronized bitstream," *IEEE Trans. Commun.*, vol. 64, no. 7, pp. 2693–2706, Jul. 2016.
[11] C. Choi and D. Yoon, "Enhanced blind interleaver parameters estimation algorithm for noisy environment," *IEEE Access*, vol. 6, pp. 5910–5915, 2018.
[12] C. Choi and D. Yoon, "Novel blind interleaver parameter estimation in a noncooperative context," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 2079–2085, Aug. 2019.
[13] S. Ramabadran, A. S. Madhukumar, N. W. Teck, and C. M. S. See, "Parameter estimation of convolutional and helical interleavers in a noisy environment," *IEEE Access*, vol. 5, pp. 6151–6167, 2017.
[14] R. Swaminathan, A. S. Madhukumar, G. Wang, and T. S. Kee, "Blind reconstruction of Reed–Solomon encoder and interleavers over noisy environment," *IEEE Trans. Broadcast.*, vol. 64, no. 4, pp. 830–845, Dec. 2018.
[15] Y. Xu, Y. Zhong, and Z. Huang, "An improved blind recognition method of the convolutional interleaver parameters in a noisy channel," *IEEE Access*, vol. 7, pp. 101775–101784, 2019.
[16] G. Kim, M. Jang, and D. Yoon, "Improved method for interleaving parameter estimation in a non-cooperative context," *IEEE Access*, vol. 7, pp. 92171–92175, 2019.
[17] M. Jang, G. Kim, Y. Kim, and D. Yoon, "Blind estimation of interleaver parameter with a limited number of data," *IEEE Access*, vol. 8, pp. 69160–69166, 2020.
[18] Z. Wu, S. Zhou, Z. Yin, B. Ma, and Z. Yang, "Robust automatic modulation classification under varying noise conditions," *IEEE Access*, vol. 5, pp. 19733–19741, 2017.
[19] T. R. Kishore and K. D. Rao, "Automatic intrapulse modulation classification of advanced LPI radar waveforms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 2, pp. 901–914, Apr. 2017.
[20] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.
[21] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.
[22] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10760–10772, Nov. 2018.

[23] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion methods for CNN-based automatic modulation classification," *IEEE Access*, vol. 7, pp. 66496–66504, 2019.
[24] H.-C. Wu, M. Saquib, and Z. Yun, "Novel automatic modulation classification using cumulant features for communications via multipath channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3098–3105, Aug. 2008.
[25] J. D. Vlok and J. C. Olivier, "Blind sequence-length estimation of low-SNR cyclostationary sequences," *IET Commun.*, vol. 8, no. 9, pp. 1578–1588, Jun. 2014.
[26] M. K. Tsatsanis and G. B. Giannakis, "Blind estimation of direct sequence spread spectrum signals in multipath," *IEEE Trans. Signal Process.*, vol. 45, no. 5, pp. 1241–1252, May 1997.
[27] J.-H. Liang, X. Wang, F.-H. Wang, and Z.-T. Huang, "Blind spreading sequence estimation algorithm for long-code DS-CDMA signals in asynchronous multi-user systems," *IET Signal Process.*, vol. 11, no. 6, pp. 704–710, Aug. 2017.
[28] H. M. Sarcheshmeh, H. K. Bizaki, and S. Alizadeh, "PN sequence blind estimation in multiuser DS-CDMA systems with multipath channels based on successive subspace scheme," *Int. J. Commun. Syst.*, vol. 31, no. 12, p. e3591, Aug. 2018.
[29] B. Shen and J.-X. Wang, "Chip rate and pseudo-noise sequence estimation for direct sequence spread spectrum signals," *IET Signal Process.*, vol. 11, no. 6, pp. 727–733, Aug. 2017.
[30] M. Cluzeau, "Reconstruction of a linear scrambler," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1283–1291, Sep. 2007.
[31] X.-B. Liu, S. N. Koh, X.-W. Wu, and C.-C. Chui, "Investigation on scrambler reconstruction with minimum a priori knowledge," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.
[32] X.-B. Liu, S. N. Koh, X.-W. Wu, and C.-C. Chui, "Reconstructing a linear scrambler with improved detection capability and in the presence of noise," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 208–218, Feb. 2012.
[33] H. WenJia, "Reconstructing the feedback polynomial of a linear scrambler with the method of hypothesis testing," *IET Commun.*, vol. 9, no. 8, pp. 1044–1047, May 2015.
[34] X.-B. Liu, S. N. Koh, C.-C. Chui, and X.-W. Wu, "A study on reconstruction of linear scrambler using dual words of channel encoder," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 542–552, Mar. 2013.
[35] Y. Ma, L.-M. Zhang, and H.-T. Wang, "Reconstructing synchronous scrambler with robust detection capability in the presence of noise," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 397–408, Feb. 2015.
[36] X. Gu, Z. Zhao, and L. Shen, "Blind estimation of pseudo-random codes in periodic long code direct sequence spread spectrum signals," *IET Commun.*, vol. 10, no. 11, pp. 1273–1281, Jul. 2016.
[37] H. Xie, F. Wang, and Z. Huang, "Blind reconstruction of linear scrambler," *J. Syst. Eng. Electron.*, vol. 25, no. 4, pp. 560–565, Aug. 2014.
[38] R. E. Ziemer and R. L. Peterson, *Digital Communications and Spread Spectrum Systems*. New York, NY, USA: Macmillan, 1985.
[39] R. C. Dixon, *Spread Spectrum Systems: With Commercial Applications*, vol. 994. New York, NY, USA: Wiley, 1994.
[40] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, vol. 1. Rockville, MD, USA: Computer science press, 1985.
[41] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications—A tutorial," *IEEE Trans. Commun.*, vol. COM-30, no. 5, pp. 855–884, May 1982.
[42] R. A. Scholtz, "The spread spectrum concept," *IEEE Trans. Commun.*, vol. COM-25, no. 8, pp. 748–755, Aug. 1977.
[43] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.
[44] P. F. Duvall and J. C. Mortick, "Decimation of periodic sequences," *SIAM J. Appl. Math.*, vol. 21, no. 3, pp. 367–372, Nov. 1971.
[45] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 4, pp. 619–621, Oct. 1967.
[46] T. Kasami, "Weight distribution formula for some class of cyclic codes," Coordinated Sci. Lab., Univ. Illinois, Urbana, IL, USA, Tech. Rep. R-285 (AD 632574), 1966.
[47] A. C. Grove, "An introduction to Walsh functions and their applications," *Int. J. Math. Educ. Sci. Technol.*, vol. 14, no. 1, pp. 43–53, Jan. 1983.
[48] F. Adachi, M. Sawahashi, and H. Suda, "Wideband DS-CDMA for next-generation mobile communications systems," *IEEE Commun. Mag.*, vol. 36, no. 9, pp. 56–69, Sep. 1998.

DONGYEONG KIM received the B.S. degree in mathematics from Hanyang University, Seoul, South Korea, in 2013, and the Ph.D. degree in mathematics from Hanyang University, under the supervision of Prof. J. Song, in 2020. His research interests include cryptanalysis of symmetric-key cyrptography, channel coding theory, and post quantum cryptography.

JUNGHWAN SONG (Member, IEEE) received the B.S. degree in mathematics from Hanyang University, Seoul, South Korea, in 1984, the M.S. degree in mathematics from Syracuse University, Syracuse, NY, USA, in 1989, and the Ph.D. degree in mathematics from the Rensselaer Polytechnic Institute, Troy, NY, in 1993. He was the Chairman of the Korea Cryptographic Forum. He is currently a Professor with the Department of Mathematics, Hanyang University. His current research interests include cryptanalysis of symmetric-key cryptography, mathematical optimization, and post quantum cryptography.

DONGWEON YOON (Senior Member, IEEE) received the B.S. *(summa cum laude)*, M.S., and Ph.D. degrees in electronic communications engineering from Hanyang University, Seoul, South Korea, in 1989, 1992, and 1995, respectively. From March 1995 to August 1997, he was an Assistant Professor with the Department of Electronic and Information Engineering, Dongseo University, Busan, South Korea. From September 1997 to February 2004, he was an Associate Professor with the Department of Information and Communications Engineering, Daejeon University, Daejeon, South Korea. Since March 2004, he has been a Faculty Member with Hanyang University, where he is currently a Professor with the Department of Electronic Engineering and the Director of the Signal Intelligence Research Center. His research interests include digital communications theory and systems, detection and estimation, satellite and space communications, and communication forensics.