Reconstructing a Linear Scrambler With Improved Detection Capability and in the Presence of Noise

Xiao-Bei Liu, Soo Ngee Koh, Xin-Wen Wu, *Member, IEEE*, and Chee-Cheon Chui

***Abstract*—In this paper, the problem of reconstruction of the feedback polynomial in a linear scrambler is studied. Our work contains two parts. In the first part, schemes to improve the performance of an existing reconstruction algorithm are proposed. Simulation results show that both the detection capability and speed of the existing algorithm are significantly improved by using our proposed schemes. In the second part, the reconstruction of linear scramblers in the presence of channel noise is investigated. We consider flipped bits due to noise as well as insertion of bits in the scrambled bit sequence. For both cases, factors which affect the performance of the reconstruction algorithm are discussed.**

***Index Terms*—Binary symmetric channel, linear feedback shift register, scrambler.**

I. Introduction

In A digital communication system, the constituent elements used by the transmitter and the specifications of each element are normally known by the receiver. In this paper, we consider a scenario wherein the specifications of the elements used by the transmitter are not completely known to the receiver. The capability of reconstructing transmitter elements when their specifications are not perfectly known is envisaged to be an enabling technology in digital communication systems with a flexible platform such as software defined radio (SDR), as it will reduce the overheads needed and make the design of the system more flexible. Similar application is also envisaged in [1] for "multistandard adaptive receivers."

The challenge of reconstructing transmitter elements when their specifications are unknown has attracted considerable research interests in the last few years. For example, results and findings on recovery of error-correcting codes have been published in [2]–[7]. In this paper, we focus on the reconstruction of another element which is commonly used in digital communication systems, i.e., the linear scrambler. A linear scrambler is usually used in a communication system to convert a data bit sequence into a pseudorandom sequence that is free from long strings of 1s or 0s. There are generally two types of linear scrambler, namely the synchronous scrambler and the self-synchronized scrambler. Both types of scrambler usually consist of a linear feedback shift register (LFSR) whose output sequence $(s_t)_{t\geq 0}$ is combined with the input sequence $(x_t)_{t\geq 0}$ and the result is the scrambled sequence $(y_t)_{t\geq 0}$, i.e.,

$$y_t = x_t \oplus s_t, \quad t \geq 0 \tag{1}$$

where $\oplus$ denotes module 2 summation. In this paper, for simplicity, only the synchronous scrambler is considered. However, our ideas on the reconstruction of synchronous scramblers with the presence of channel noise can also be extended to self-synchronized scramblers.

In most communication systems, to achieve the maximum period for the sequences produced by the LFSRs, binary primitive polynomials are used as the feedback polynomials. Reconstructing a linear scrambler consists of reconstructing the feedback polynomial of the LFSR as well as its initial state in the case of a synchronous scrambler. In this paper, we will focus on reconstructing the feedback polynomial of the LFSR, as reconstructing the initial state of the LFRS is a well-known problem for stream cipher and it has been extensively studied in the literature [8]–[11]. When some input and scrambled bits are known, the Berlekamp–Massey algorithm [12] can be used to reconstruct the feedback polynomial of the LFSR. Recently, an algorithm is proposed by Cluzeau for reconstructing the feedback polynomial of the LFSR by using only the scrambled bits [13]. In the following, this algorithm will be referred to as "Cluzeau's algorithm."

Although Cluzeau's algorithm can be used to reconstruct most of the feedback polynomials of the LFSR very efficiently, it can be observed from the simulation results shown in [13] that in some cases, the algorithm cannot make a correct detection of the feedback polynomial even when the false-alarm probability P_f is set to a very small value. Furthermore, the algorithm proposed in [13] assumed that all the scrambled bits are correctly received. In practical situations, the communication channels always have some types of noise, which will lead to errors in the received bits. In this paper, the above-mentioned two problems are investigated. In the first part of this paper, a scheme to improve the detection capability of Cluzeau's algorithm, with only marginal increase in complexity is proposed. Following that, an approach to reduce the number of operations required by Cluzeau's algorithm to do the recovery without affecting the detection capability is described. In the second part of this paper, the problem of reconstruction of scramblers in the presence of noise is studied. Two kinds of channel errors are considered; one is bit flipping due to channel noise and the second is insertion of bits in the scrambled bit sequence.

Manuscript received April 17, 2011; revised August 03, 2011; accepted September 12, 2011. Date of publication September 29, 2011; date of current version January 13, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Z. Jane Wang.

X.-B. Liu and S. N. Koh are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

X.-W. Wu is with the School of Information and Communication Technology, Griffith University, Gold Coast, QLD 4222, Australia.

C.-C. Chui is with the Temasek Laboratories at Nanyang Technological University, Singapore 639798.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2011.2169790

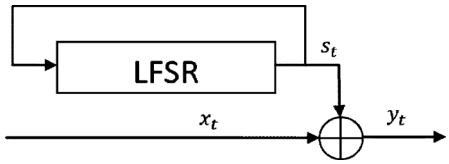

Fig. 1. Structure of synchronous scrambler.

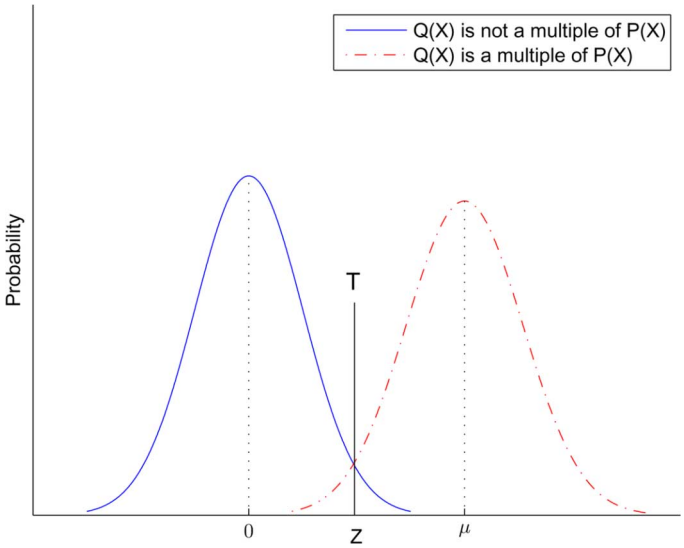

Fig. 2. Distributions of Z.

The paper is organized as follows. In Section II, Cluzeau's algorithm is reviewed. In Section III, a scheme is proposed to improve the detection capability of Cluzeau's algorithm. In Section IV, the approach to reduce the number of operations required to detect the feedback polynomial will be described. In Section V, reconstructions of the scrambler in the presence of channel noise is investigated. Conclusions are drawn in Section VI.

II. Cluzeau's Algorithm for Reconstructing a Synchronous Scrambler

In this section, Cluzeau's algorithm which recovers the feedback polynomial $P(X)$ of a synchronous scrambler by using only the scrambled bits will be reviewed. In a synchronous scrambler, s_t is generated independently of x_t and y_t, as shown in Fig. 1.

Instead of searching for the feedback polynomial $P(X)$ directly, Cluzeau's algorithm searched for sparse multiples of $P(X)$ with the degree of the sparse multiples varying from low to high. After two multiples of $P(X)$ are detected, it returns the nontrivial greatest common divisor (gcd) of the two detected multiples as the detected feedback polynomial. The determination of whether a sparse polynomial is a multiple of $P(X)$ or not is based on a statistical test on the absolute value of a variable Z, which is given by

$$Z = \sum_{t=i_{d-1}}^{N-1} (-1)^{z_t} \tag{2}$$

where z_t is a module 2 summation of d scrambled bits, i.e., $z_t = y_t \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j}, (0 < i_1 < i_2 < \cdots < i_{d-1})$. Let $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$, when $Q(X)$ is a multiple of $P(X)$, we have

$$\begin{aligned} z_t &= y_t \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j} \\ &= x_t \oplus \bigoplus_{j=1}^{d-1} x_{t-i_j} \end{aligned} \tag{3}$$

since $s_t \oplus \bigoplus_{j=1}^{d-1} s_{t-i_j} = 0$ and $y_t = x_t \oplus s_t$. According to the statistical analysis results given in [7], when $Q(X)$ is a multiple of $P(X)$ and if the input bits are biased distributed with $\Pr(x_t = 1) = (1/2) - \varepsilon$, where $\varepsilon \neq 0$, z_t is also biased distributed with $\Pr(z_t = 1) = (1/2)[1-(2\varepsilon)^d]$. Then according to Theorem 1 given in [7], the value of Z, i.e., $\sum_{t=i_{d-1}}^{N-1}(-1)^{z_t} = (N - i_{d-1}) - 2\sum_{t=i_{d-1}}^{N-1} z_t$, is Gaussian distributed with mean value μ given by

$$\mu = (N - i_{d-1})(2\varepsilon)^d \tag{4}$$

and variance σ^2 given by

$$\begin{aligned} \sigma^2 = (N - i_{d-1})(1 - (2\varepsilon)^{2d}) \\ + \sum_{u=1}^{d-1} N_u \left((2\varepsilon)^{2(d-u)} - (2\varepsilon)^{2d}\right) \end{aligned} \tag{5}$$

where N_u denotes the number of pairs $(z_t, z_{t'}), (0 < |t' - t| \leq i_{d-1})$ which share exactly u terms of x_t. For different $Q(X)$, the values of N_u are different, and hence there is not a fixed value of σ^2. However, according to [7], an upper bound of σ^2 can be derived, since in the worst case, $N_1 = N_2 = \cdots = N_{d-2} = 0$ and $N_{d-1} \leq (N - i_{d-1})2d(d-1)$, which leads to

$$\begin{aligned} \sigma^2 &\leq (N - i_{d-1})[1 + 2d(d-1)(2\varepsilon)^2 \\ &\qquad - (2\varepsilon)^{2d}(1 + 2d(d-1))] \\ &\leq (N - i_{d-1})[1 + 2d(d-1)](1 - (2\varepsilon)^{2d}). \end{aligned} \tag{6}$$

Therefore, the upper bound σ_l of σ is given by

$$\sigma \leq \sigma_l = \sqrt{(N - i_{d-1})[1 + 2d(d-1)](1 - (2\varepsilon)^{2d})} \tag{7}$$

and the normalized upper bound $\bar{\sigma}_l$ is given by

$$\bar{\sigma}_l = \frac{\sigma_l}{\sqrt{N - i_{d-1}}} = \sqrt{[1 + 2d(d-1)](1 - (2\varepsilon)^{2d})}. \tag{8}$$

According to the above description, when $Q(X)$ is a multiple of $P(X)$, Z has a Gaussian distribution with mean value μ and variance σ^2; it is also obvious that when $Q(X)$ is not a multiple of $P(X)$, $\Pr(z_t = 0) = (1/2)$, implying that Z has a Gaussian distribution with mean value 0 and variance $N - i_{d-1}$. The two distributions are depicted in Fig. 2.

From Fig. 2, it can be observed that when the two distributions of Z have a small enough intersection, a threshold T can be used to determine whether $Q(X)$ is a multiple of $P(X)$ or not: i.e., when $|Z| < T$, $Q(X)$ is not a multiple of $P(X)$; otherwise, $Q(X)$ is a multiple of $P(X)$. T depends on the false-alarm probability $P_f = \Pr(|Z| \geq T|$ when $Q(X)$ is not a multiple of

TABLE I
SIMULATION RESULTS OF CLUZEAU'S ALGORITHM

Feedback Polynomial	Detected polynomial	Time
$x^8+x^4+x^3+x^2+1$	$x^8+x^4+x^3+x^2+1$	13.7s
$x^9+x^6+x^4+x^3+1$	$x^9+x^6+x^4+x^3+1$	40.9s
$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{23}+x^7+1$	49.8s
$x^{12}+x^{11}+x^{10}+x^8+x^7+x^2+1$	$x^{19}+x^{12}+1$	34.2s
$x^{23}+x^{18}+1$	$x^{23}+x^{18}+1$	50.8s
$x^{29}+x^2+1$	$x^{29}+x^2+1$	1 min 19.2s

$P(X)$) and on the nondetection probability $P_n = \Pr(|Z| < T|$ when $Q(X)$ is a multiple of $P(X)$). In the following, Cluzeau's algorithm is outlined for reader's convenience.

1) Compute the threshold T as follows:

$$T = \frac{a(a+b\bar{\sigma}_l)}{(2|\varepsilon|)^d} \tag{9}$$

where

$$a = \Phi^{-1}\left(1-\frac{P_f}{2}\right) \tag{10}$$

and

$$b = -\Phi^{-1}(P_n). \tag{11}$$

In the above equations, Φ denotes the normal distribution function, i.e.,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left(-\frac{t^2}{2}\right)dt \tag{12}$$

and d denotes the weights of the sparse multiples of $P(X)$ and typical values of d are 3, 4, and 5.

2) For $(i_1,\ldots,i_{d-1})$, $0 < i_1 < \cdots < i_{d-1} \le D$ (D is the maximum degree of the sparse multiple we want to search), compute the number of bits N required to recover the feedback polynomial as follows:

$$N = i_{d-1} + \frac{(a+b\bar{\sigma}_l)^2}{(2\varepsilon)^{2d}}. \tag{13}$$

3) Initialize Z with $Z = 0$.
For t from i_{d-1} to N, compute

$$z_t = y_t \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j} \tag{14}$$

and

$$Z = Z + (-1)^{z_t}. \tag{15}$$

4) If $|Z| > T$, store $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$ in a table.
5) For $Q'(X) \neq Q(X)$ in the table, compute the nontrivial gcd of $(Q(X), Q'(X))$.

Steps 2–5 are repeated until a $\gcd(Q(X), Q'(X)) = P(X)(P(X) \neq 1)$ is found or all combinations of $(i_1,\ldots,i_{d-1})$ are tested.

According to [7], if using the algorithm described above, for a randomly chosen primitive polynomial $P(X)$ of degree L, the number of operations performed by the algorithm is

$$W_p = d\cdot 2^L\frac{(a+b\bar{\sigma}_l)^2}{(2\varepsilon)^{2d}} \tag{16}$$

TABLE II
MULTIPLES OF THE FEEDBACK POLYNOMIALS

Feedback Polynomial	Detected Trinomial Multiples
$x^8+x^4+x^3+x^2+1$	$x^{21}+x^{10}+1$, $x^{25}+x+1$
$x^9+x^6+x^4+x^3+1$	$x^{36}+x^{19}+1$, $x^{42}+x^5+1$
$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{23}+x^7+1$, $x^{46}+x^{14}+1$
$x^{12}+x^{11}+x^{10}+x^8+x^7+x^2+1$	$x^{19}+x^{12}+1$, $x^{38}+x^{24}+1$
$x^{23}+x^{18}+1$	$x^{23}+x^{18}+1$, $x^{46}+x^{36}+1$
$x^{29}+x^2+1$	$x^{29}+x^2+1$, $x^{58}+x^4+1$

To test the accuracy and runtime of Cluzeau's algorithm, it is applied to some feedback polynomials and results are shown in Table I. We will propose improved reconstruction procedures in the following sections. To make a fair comparison with Cluzeau's algorithm [13], we run Cluzeau's algorithm with the parameters $\varepsilon = 0.1$, false-alarm probability $P_f = 2\cdot 10^{-7}$, nondetection probability $P_n = 10^{-5}$, and weight of the sparse multiples of the feedback polynomial $d = 3$, which are the same as those in [13].

In Table I, the first, third, and last feedback polynomials are the same as those shown in Table III in [13]. Due to the difference in hardware (Intel dual core, 2.5 GHz), when testing Cluzeau's algorithm, the runtimes are a bit different from those presented by Cluzeau. In the following sections, we will test our improved reconstruction algorithm and compare the results, using the same parameters and hardware as those for Table I.

III. IMPROVE THE DETECTION CAPABILITY OF CLUZEAU'S ALGORITHM

From Table I, it can be observed that the third and fourth polynomials are not correctly recovered. To further understand why the original feedback polynomials are not detected, we look into the detected multiples of the feedback polynomials and results are shown in Table II.

From Table II, it can be observed that for the third and fourth feedback polynomials, their second detected trinomial multiple is a multiple (square) of the first one. Therefore, their gcd is the first detected multiple instead of the feedback polynomial. For the feedback polynomials $x^{23}+x^{18}+1$ and $x^{29}+x^2+1$, their second detected trinomial multiple is also a multiple of the first one. However, as shown in Table I, they are correctly detected. This is because their first detected trinomial multiples, i.e., $x^{23}+x^{18}+1$ and $x^{29}+x^2+1$, are exactly the feedback polynomials already.

Based on the above observation, to avoid making the wrong detection, the second detected multiple of the feedback polynomial should be ignored if it is a multiple of the first one. The only exception is that when the first detected multiple of the feedback polynomial is irreducible, then the feedback polynomial is nothing else but the detected multiple itself. Consequently, we

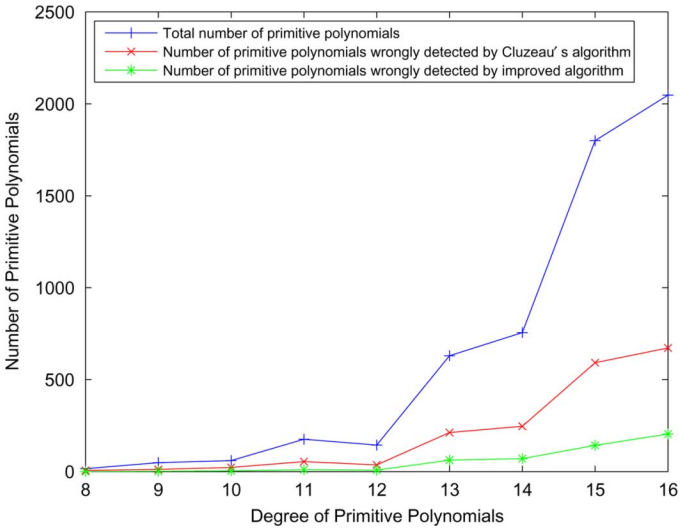

Fig. 3. Comparison of Cluzeau's and improved algorithms.

can modify sightly Cluzeau's algorithm with an improved performance as follows:

1) Follow the algorithm described in Section II until a sparse multiple $Q(X)$ is found.
2) If $Q(X)$ is irreducible, then stop and return $Q(X)$ as the detected feedback polynomial; otherwise, store $Q(X)$ in a table and go to the next step.
3) Search for another sparse multiple $Q'(X)$; if $Q'(X)$ is a multiple of $Q(X)$, ignore it and keep on searching until a $Q'(X)$ which is not a multiple of $Q(X)$ is found. Store $Q'(X)$ in the table.
4) Compute gcd $(Q(X), Q'(X)) = P(X)$. If $P(X) \neq 1$, return $P(X)$ as the detected feedback polynomial. Otherwise, go to Step 3.

In the following, the algorithm described above will be called "improved algorithm," to differentiate it from Cluzeau's algorithm. The improved algorithm includes two extra steps, as compared with Cluzeau's algorithm. The first is to determine if the detected multiple is irreducible. The second is to determine if the second multiple is a multiple of the first one. The second step is rather simple because the number of operations (XORs) used by it is bounded from above by $O(d \cdot D)$. For the first step, there are a lot of algorithms proposed in the literature to test the reducibility of a binary polynomial [14], [15]. We use the algorithm proposed in [15] and the number of operations is bounded from above by $O(d \cdot D^2)$. Compared with Cluzeau's algorithm, for which the number of operations is bounded from above by $O((D^{d-1})/((2\varepsilon)^{2d}))$, the increase in complexity is trivial.

To evaluate the performance of the improved algorithm, the scrambler using each of all the primitive polynomials of degrees from 8 to 16 as the feedback polynomial is simulated. The weight of the sparse multiple is chosen to be $d = 3$ and the results obtained are plotted in Fig. 3.

From Fig. 3, it is apparent that the detection performance is significantly improved by using the improved algorithm. It should be noted that there are still feedback polynomials which cannot be correctly recovered by using the improved algorithm. This situation occurs if the first detected multiple is $Q(X) = P(X)F(X)D(X)$, where $F(X)$ and $D(X)$ are any binary polynomials not equal to 1, and the second detected multiple is $Q'(X) = P(X)F(X)D'(X)$. If $D'(X)$ is not a multiple of $D(X)$, $Q'(X)$ is not a multiple of $Q(X)$ either. Therefore, $Q'(X)$ will not be ignored in the improved algorithm, and the gcd of $Q(X)$ and $Q'(X)$ is $P(X)F(X)$ instead of $P(X)$ itself. In any case, if the detected feedback polynomial is not primitive, say it is equal to $P(X)F(X)$, we can find the correct feedback polynomial by trying to recover the source bits using the polynomials $P(X)F(X)$, $P(X)$ and $F(X)$ as the feedback polynomials, respectively, and see which one would lead to a sensible source sequence. The procedure above includes the factorization of $P(X)F(X)$ which can be easily achieved with Berlekamp's algorithm [16].

IV. Improve the Speed of Cluzeau's Algorithm

According to (13) and (16), the number of bits and operations required to recover the feedback polynomial depends on the normalized upper bound $\bar{\sigma}_l$. The bigger the value of $\bar{\sigma}_l$, the larger the number of bits and operations required. In the following, we will show that the upper bound of σ can be improved, and a new upper bound σ'_l, which approaches the actual value of σ more closely, will be derived.

As described in Section II, σ^2 becomes the maximum value in the worst case, i.e., $N_1 = N_2 = \cdots = N_{d-2} = 0$ and $N_{d-1} \leq (N - i_{d-1})2d(d-1)$, where $N_u (u = 1, 2, \ldots, d-1)$ denotes the number of pairs $(z_t, z_{t'})$ which share exactly u terms of x_t. It is obvious that N_u also denotes the number of pairs $(X^t Q(X), X^{t'} Q(X))$ which share exactly u terms of X^i. Let M_u denote the number of pairs $(Q(X), X^n Q(X)), (0 < n \leq i_{d-1})$, which share exactly u terms of $X^i (i = i_1, i_2, \ldots i_{d-1})$. Since at each time instant t, $(X^t Q(X), X^{t'} Q(X)), (i_{d-1} \leq t < t' \leq N-1, 0 < t'-t \leq i_{d-1})$, can be taken as a time shift version of $(Q(X), X^n Q(X)), (0 < n \leq i_{d-1})$, we have

$$N_u = 2(N - i_{d-1})M_u. \tag{17}$$

The factor 2 in (17) arises because t and t' are interchangeable.

In the following, the number of terms that the pair $(Q(X), X^n Q(X))$ have in common is studied. $Q(X)$ and $X^n Q(X)$ are denoted by $Q(X) = 1 + X^{i_1} + X^{i_2} + \cdots + X^{i_{d-1}}$ and $X^n Q(X) = X^n + X^{n+i_1} + X^{n+i_2} + \cdots + X^{n+i_{d-1}}$, respectively. Obviously, the constant term 1 will not be shared by the pair $(Q(X), X^n Q(X))$. So, first, how many times X^{i_1} appears in $X^n Q(X)$ is studied. It can be observed that there is only one time, i.e., when $n = i_1$, X^{i_1} appears in $X^{i_1} Q(X)$. Then, for X^{i_2}, there are two times it appears in $X^n Q(X)$, i.e., when $n = i_2$ or $n = i_2 - i_1$, X^{i_2} is shared by the pair $(Q(X), X^n Q(X))$. For the same reason, for X^{i_3}, $X^{i_4}, \ldots, X^{i_{d-1}}$, the number of times they appear in $X^n Q(X)$ are $3, 4, \ldots, d-1$, respectively. In Table III, a summary of the terms that the pair $(Q(X), X^n Q(X))$ have in common, with their shared times and the corresponding values of n are shown.

Obviously, Table III includes all the terms that the pair $(Q(X), X^n Q(X))$ share for $0 < n \leq i_{d-1}$. It is also noted that for the same shared term, the possible values of n are different since $0 < i_1 < i_2 < \cdots < i_{d-1}$. It means that there is no double counting of the shared terms. Therefore, the total

TABLE III
SHARED TERMS OF THE PAIR $(Q(X), X^nQ(X))$ AND THE CORRESPONDING VALUE OF n

Shared term	Shared times	Value of n
1	0	N.A
X^{i_1}	1	i_1
X^{i_2}	2	i_2, $i_2 - i_1$
X^{i_3}	3	i_3, $i_3 - i_1$, $i_3 - i_2$
...	...	...
$X^{i_{d-1}}$	$d-1$	i_{d-1},$i_{d-1} - i_1$,..., $i_{d-1} - i_{d-2}$

number of shared terms for the pair $(Q(X), X^nQ(X))$ for $0 < n \leq i_{d-1}$ is

$$M = \frac{[1 + (d-1)](d-1)}{2} = \frac{d(d-1)}{2}. \tag{18}$$

For different shared terms, the value of n may be the same, e.g., when $i_1 = i_2 - i_1 = i_3 - i_2$, $Q(X)$ and $X^{i_1}Q(X)$ will share X^{i_1}, X^{i_2}, and X^{i_3}, i.e., three terms. Obviously, the maximum number of terms $Q(X)$ and $X^nQ(X)$ can share for the same value of n is $d-1$. Since M_u denotes the number of pairs $(Q(X), X^nQ(X)), (0 < n \leq i_{d-1})$ which share exactly u terms of X^i, we have

$$\sum_{u=1}^{d-1} u \cdot M_u = M = \frac{d(d-1)}{2}. \tag{19}$$

Based on (17) and (19), we have

$$\begin{aligned} \sum_{u=1}^{d-1} u \cdot N_u &= 2(N - i_{d-1}) \sum_{u=1}^{d-1} u \cdot M_u \\ &= (N - i_{d-1}) \cdot d \cdot (d-1). \end{aligned} \tag{20}$$

From (5) and (20), it can be observed that the maximum value of σ can be achieved only when

$$N_u = \begin{cases} 0, & \text{when} \quad u < d-1 \\ (N - i_{d-1})d, & \text{when} \quad u = d-1. \end{cases} \tag{21}$$

Put the value of $u = d-1$ and $N_u = (N - i_{d-1})d$ into (5), we get

$$\begin{aligned} \sigma^2 &\leq (N - i_{d-1})(1 - (2\varepsilon)^{2d}) \\ &\quad + (N - i_{d-1})d((2\varepsilon)^2 - (2\varepsilon)^{2d}) \\ &\leq (N - i_{d-1})[1 + d((2\varepsilon)^2 - (2\varepsilon)^{2d})]. \end{aligned} \tag{22}$$

According to (22), the new upper bound σ_l' of σ is given by

$$\sigma_l' = \sqrt{(N - i_{d-1})[1 + d((2\varepsilon)^2 - (2\varepsilon)^{2d})]} \tag{23}$$

and the new normalized upper bound $\bar{\sigma}_l'$ is given by

$$\bar{\sigma}_l' = \frac{\sigma_l'}{\sqrt{N - i_{d-1}}} = \sqrt{1 + d((2\varepsilon)^2 - (2\varepsilon)^{2d})}. \tag{24}$$

To see how closely the new upper bound of σ derived above approaches the actual value, the Gaussian distributions with μ calculated by using (4) and standard deviation σ equal to σ_l and σ_l', respectively, are plotted in Fig. 4. In Fig. 4, the actual distribution of Z, which is obtained by using $Q(X) = 1 + x^{10} + x^{21}$, $N = 10\,000$, $\varepsilon = 0.1$, and $d = 3$, is also plotted. A total of 50 000 values of Z are collected and a low-pass filter is used to smooth the curve. Other trinomials for $Q(X)$ are also tested in our simulations and it is found that there is no big difference in the distribution of Z for different trinomials of $Q(X)$. Therefore, the dashed curve in Fig. 4 can be taken as an approximation of the actual distribution of Z for any $Q(X)$ with $d = 3$.

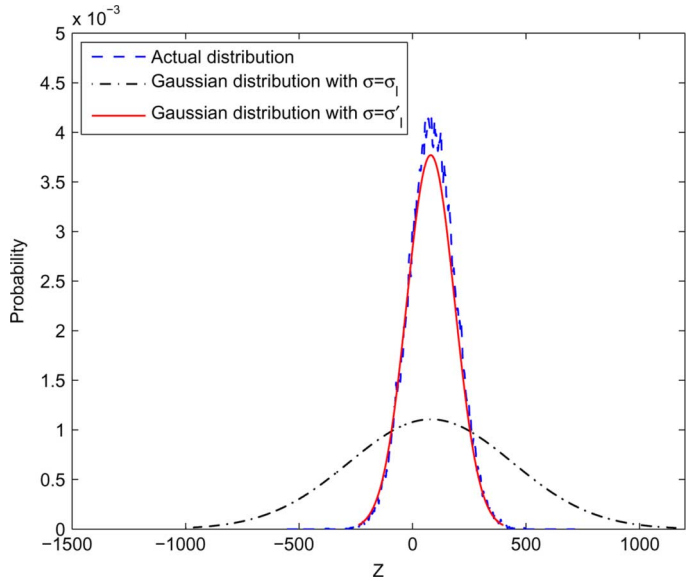

Fig. 4. Distribution of Z with different $\sigma(N = 10\,000, i_{d-1} = 21, \epsilon = 0.1, d = 3)$.

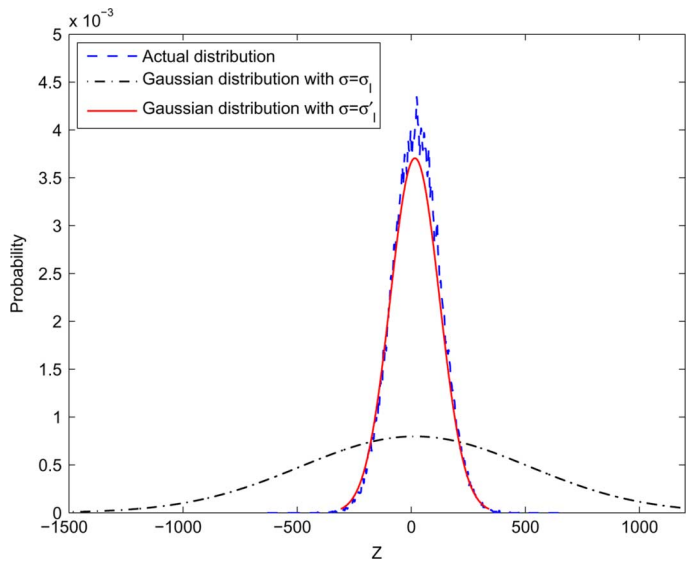

Fig. 5. Distribution of Z with different $\sigma(N = 10\,000, i_{d-1} = 31, \epsilon = 0.1, d = 4)$.

From Fig. 4, it can be observed that the Gaussian distribution with $\sigma = \sigma_l'$ approaches the actual distribution of Z very closely. The Gaussian distribution with $\sigma = \sigma_l$ deviates from the actual distribution by quite a margin, as shown in Fig. 4. It should be noted that when d increases, σ_l deviates even more from σ, as shown in Fig. 5, but σ_l' still approaches σ very closely.

TABLE IV
SIMULATION RESULTS OF THE IMPROVED ALGORITHM USING $\tilde{T}$ AND $\tilde{N}$ INSTEAD OF T AND N

Feedback Polynomial	Detected polynomial	Time (using T&N)	Time (using $\tilde{T}$&$\tilde{N}$)
$x^8+x^4+x^3+x^2+1$	$x^8+x^4+x^3+x^2+1$	13.7s	3.0s
$x^9+x^6+x^4+x^3+1$	$x^9+x^6+x^4+x^3+1$	40.9s	9.0s
$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{10}+x^6+x^5+x^3+x^2+x+1$	1 min 40.7s	22.5s
$x^{12}+x^{11}+x^{10}+x^8+x^7+x^2+1$	$x^{12}+x^{11}+x^{10}+x^8+x^7+x^2+1$	5 min 48.4s	1 min 16.7s
$x^{23}+x^{18}+1$	$x^{23}+x^{18}+1$	12.3s	2.7s
$x^{29}+x^2+1$	$x^{29}+x^2+1$	18.8s	4.1s

When using the new upper bound, both the threshold T, the number of bits N, and the number of operations W_p required by the algorithm can be reduced and consequently, the runtime of the algorithm can be reduced. The new threshold $\tilde{T}$ can be obtained by replacing $\bar{\sigma}_l$ in (9) by $\bar{\sigma}'_l$ and the result is given by

$$\tilde{T} = \frac{a(a+b\bar{\sigma}'_l)}{(2|\varepsilon|)^d} = \frac{a^2+ab\sqrt{1+d((2\varepsilon)^2-(2\varepsilon)^{2d})}}{(2|\varepsilon|)^d}. \quad (25)$$

Similarly, the revised number of bits $\tilde{N}$ and operations $\tilde{W}_p$ required by the algorithm are given by

$$\tilde{N} = i_{d-1} + \frac{\left(a+b\sqrt{1+d((2\varepsilon)^2-(2\varepsilon)^{2d})}\right)^2}{(2\varepsilon)^{2d}} \quad (26)$$

and

$$\tilde{W}_p = d\cdot 2^L \frac{\left(a+b\sqrt{1+d((2\varepsilon)^2-(2\varepsilon)^{2d})}\right)^2}{(2\varepsilon)^{2d}}. \quad (27)$$

Comparing (16) and (27), it can be observed that the reduction factor R for the number of operations required to recover the feedback polynomial is given by

$$R = \frac{W_p}{\tilde{W}_p} \approx \frac{(\gamma+q)^2}{(1+q)^2} \quad (28)$$

where $\gamma = \sqrt{1+2d(d-1)}$ and $q = (a/b)$.

In Fig. 6, the values of R against d are plotted. According to the simulation setup described in Section II, $q \approx 1.2$. It can be observed that R increases with the increase in d. This is because γ increases with the increase in d.

In the following, the improved algorithm which uses $\tilde{T}$ and $\tilde{N}$ instead of T and N are tested by using the same feedback polynomials as those in Table I, and results are shown in Table IV. In the second and third column of Table IV, the detected feedback polynomials and runtime required by the improved algorithm are shown. In the fourth column of Table IV, the runtime required by the improved algorithm using $\tilde{T}$ and $\tilde{N}$ instead of T and N is shown.

First, we compare the first three columns of Table I with Table IV. It can be seen that for the first and second feedback polynomials which are correctly detected by using Cluzeau's algorithm, they can still be correctly detected by using the improved algorithm and the runtime is not affected. For the third and fourth feedback polynomials which are wrongly detected by using Cluzeau's algorithm, they are correctly detected by using the improved algorithm, while the runtime is also increased. For the last two feedback polynomials, the runtime is even reduced by using the improved algorithm, because they are irreducible and the algorithm is stopped after the first step. We then compare the third and fourth columns in Table IV. It can be observed that the time required to do the detection is reduced by a factor of about 4.5 for all the primitive polynomials, when using $\tilde{T}$ and $\tilde{N}$ instead of T and N in the detection process. This result matches very well with the reduction factor for the number of operations required to recover the feedback polynomial shown in Fig. 6. It should be noted that when using $\tilde{T}$ and $\tilde{N}$, the detected polynomials are still the same as those shown in the second column of Table IV. In general, comparing Tables I and IV, it is clear that with our proposed schemes, both the detection capability and speed of Cluzeau's algorithm are improved significantly.

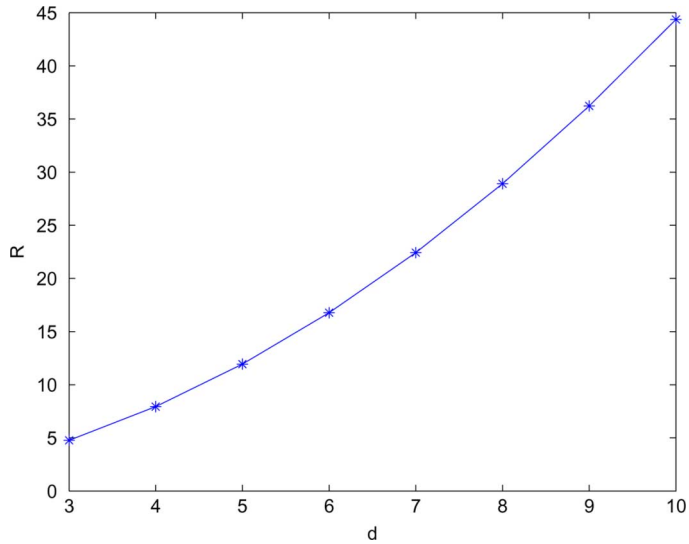

Fig. 6. Reduction factor (R) for the number of operations for different values of $d(q \approx 1.2)$.

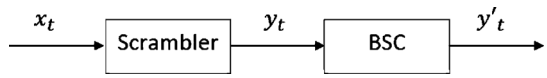

Fig. 7. Chain of scrambler and channel.

V. RECOVERY OF SCRAMBLER IN THE PRESENCE OF CHANNEL NOISE

A. Recover the Scrambler With Flipped Bits

In the algorithm described in Section II, the scrambled bit sequence $y_t(t>0)$ is assumed to be correctly received at the receiver. In this section, we consider the situation that channel noise is present, as depicted in Fig. 7.

In Fig. 7, the channel is modeled as a binary symmetric channel (BSC). The input to the BSC is denoted by y_t and the output of the channel is denoted by y'_t, which is given as

$$y'_t = y_t \oplus e_t \tag{29}$$

where e_t is the channel error at time instant t. Suppose the channel error probability (or crossover probability) is p, then at each time instant t we have

$$\Pr(e_t = 1) = p \tag{30}$$

and

$$\Pr(e_t = 0) = 1 - p. \tag{31}$$

Similar to (14), we have

$$\begin{aligned} z'_t &= y'_t \oplus \bigoplus_{j=1}^{d-1} y'_{t-i_j} \\ &= y_t \oplus e_t \bigoplus_{j=1}^{d-1} y_{t-i_j} \oplus e_{t-i_j} \end{aligned} \tag{32}$$

where $y_t = x_t \oplus s_t$. When $Q(X)$ is a multiple of $P(X)$, we have $s_t \oplus \bigoplus_{j=1}^{d-1} s_{t-i_j} = 0$ and thus

$$z'_t = x_t \oplus e_t \oplus \bigoplus_{j=1}^{d-1} x_{t-i_j} \oplus e_{t-i_j}. \tag{33}$$

Suppose $x'_t = x_t \oplus e_t$, (33) becomes

$$z'_t = x'_t \oplus \bigoplus_{j=1}^{d-1} x'_{t-i_j}. \tag{34}$$

Comparing with the noiseless case, in which we have

$$z_t = x_t \oplus \bigoplus_{j=1}^{d-1} x_{t-i_j} \tag{35}$$

we can see that the only difference is that x_t is replaced by x'_t, which can be taken as passing x_t through a BSC channel. As stated previously, the reconstruction of the scrambler is based on the assumption that x_t is biasedly distributed with $\Pr(x_t = 0) = (1/2) + \varepsilon$. When x_t is replaced by x'_t, we need to look into the distribution of x'_t and see if it is biasedly distributed also.

Let us introduce another variable δ such that $\delta = (1/2) - p$. We then have

$$\Pr(e_t = 1) = \frac{1}{2} - \delta \tag{36}$$

and

$$\Pr(e_t = 0) = \frac{1}{2} + \delta. \tag{37}$$

According to the distribution property of x_t and e_t, we have

$$\Pr(x'_t = 1) = \frac{1}{2} - 2\varepsilon\delta \tag{38}$$

and

$$\Pr(x'_t = 0) = \frac{1}{2} + 2\varepsilon\delta. \tag{39}$$

From the above equations, we can see that x'_t is also biasedly distributed, with a new bias $\varepsilon' = 2\varepsilon\delta$. It means that when channel noise is present, we still can recover the synchronous scrambler by using the method proposed for noiseless condition. The only difference is that the source bias is changed from ε to $2\varepsilon\delta$, where δ is determined by the channel error probability p.

As $p \leq 0.5$ and $\delta = (1/2) - p$, we have $0 \leq 2\delta \leq 1$ or $2\varepsilon\delta \leq \varepsilon$. The "=" sign only holds when $\delta = 0.5$ or $p = 0$. Therefore, when $p > 0$, we will have $\varepsilon' < \varepsilon$. According to (26) and (27), the smaller the bias, the larger the number of bits and operations are required in the reconstruction. Therefore, when ε is changed to ε', the number of bits required in the reconstruction becomes larger and the time required to do the detection becomes longer. When $\tilde{N} \gg i_{d-1}$, which is usually true for practical systems, the factor of increase (I) in the number of bits can be derived as

$$I = \frac{1}{(2\delta)^{2d}} = \frac{1}{(1-2p)^{2d}}. \tag{40}$$

In Fig. 8, the values of I are plotted for different values of p and d when channel errors are present. It can be observed that the factor of increase in the number of bits used in the reconstruction grows with increases in p and d. In practical situations, the channel error probability can vary from a very small value to about 0.2 when $E_b/N_o = 0$ dB [17], and, therefore, I may vary from 1 to 100 depending on the values of p and d. To make an appropriate choice of I, an accurate estimation of the statistical properties of the channel is, therefore, needed.

Next, the impact on P_f and P_n when channel errors are present but the number of bits used is not increased correspondingly is investigated. According to (10) and the new upper bound of σ derived in Section IV, we have

$$a = \Phi^{-1}\left(1 - \frac{P_f}{2}\right) = \frac{\tilde{T}}{\sqrt{\tilde{N} - i_{d-1}}}. \tag{41}$$

As $\tilde{T}$ and $\tilde{N}$ are precalculated, they will not change with the change in the source bias; therefore, P_f will not be affected by the channel noise.

According to (11) and the distribution of Z, we have

$$-b = \Phi^{-1}(P_n) = \frac{\tilde{T} - \mu}{\sigma} \tag{42}$$

where $\tilde{T}$ is given by (25), μ by (4), and σ by (22). When channel noise is present, the source bias will change from ε to $2\varepsilon\delta$. Therefore, the ε in (4) and (22) must be replaced by $2\varepsilon\delta$, and the resulting new mean value μ_e of Z is

$$\mu_e = (\tilde{N} - i_{d-1})(4\varepsilon\delta)^d = (2\delta)^d \mu \tag{43}$$

and the new upper bound σ_e of the standard deviation of Z is

$$\begin{aligned} \sigma_e &= \sqrt{(\tilde{N} - i_{d-1})[1 + d((4\varepsilon\delta)^2 - (4\varepsilon\delta)^{2d})]} \\ &\approx \sqrt{\tilde{N} - i_{d-1}} \cdot \bar{\sigma}'_l. \end{aligned} \tag{44}$$

From (43) and (44), it can be observed that when channel noise is present, the standard deviation of Z will not be significantly

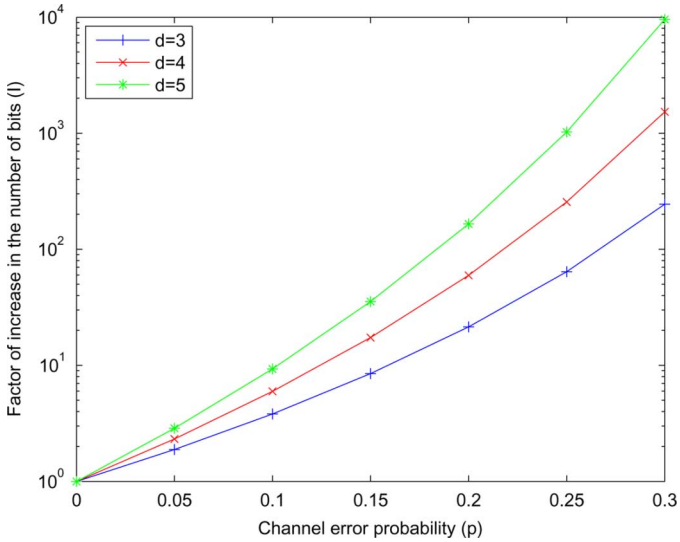

Fig. 8. Factor of increase (I) of the number of bits for different p and d when channel errors are present ($\tilde{N} \gg i_{d-1}$).

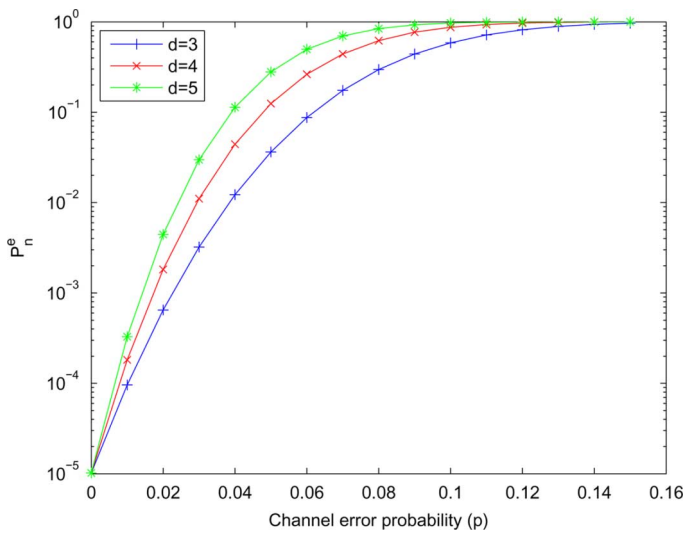

Fig. 9. Values of P_n^e for different values of p and $d(\varepsilon = 0.1, P_f = 2.10^{-7})$.

affected, but the mean value of Z will become smaller. This will result in a smaller value of b, and thus P_n will increase. As $2\delta \leq 1$, we have

$$\frac{\tilde{T} - \mu_e}{\sigma} = \frac{\tilde{T} - (2\delta)^d \mu}{\sigma} \geq \frac{(2\delta)^d (\tilde{T} - \mu)}{\sigma} = (2\delta)^d (-b). \tag{45}$$

Therefore, the new value of P_n when noise is present, i.e., P_n^e, can be roughly estimated by

$$P_n^e = \Phi\left(\frac{\tilde{T} - \mu_e}{\sigma}\right) \geq \Phi((2\delta)^d (-b)). \tag{46}$$

In Fig. 9, the values of P_n^e are plotted for different values of p and d. The number of bits $\tilde{N}$ used in the recovery is set to satisfy the condition that when $p = 0$, the nondetection probability is equal to 10^{-5}.

From the figure, it is clear that with the increase in the channel error probability, the nondetection probability P_n^e increases rapidly if the number of bits used in the recovery process is not increased correspondingly. When the channel error probability is 0.1, more than half of the sparse multiples of the feedback polynomial will not be detected. When the channel error probability becomes equal to or larger than 0.15, almost all the sparse multiples will not be detected. Therefore, as stated before, in practical situations, channel estimation should be used to make sure that the number of bits used in the recovery process is properly chosen.

B. Recovery of Scrambler When Insertion of Bits Occurs

During the data transmission, there is another kind of error, namely insertion/deletion of one or more bits into/from the scrambled bit sequence. In this section, for simplicity, only insertion of one bit is considered. However, our ideas can easily be extended to insertion of more than one bits and also deletion of one or more than one bits. Suppose c_x is inserted at time index $t_x (t_x \geq 0)$, the received sequence becomes

$$y_0, y_1, y_2, \ldots y_{t_x - 1}, c_x, y_{t_x}, y_{t_x + 1}, \ldots$$

Suppose the received sequence is denoted by y'_t, we have

$$y'_t = \begin{cases} y_t, & t \leq t_x - 1 \\ c_x, & t = t_x \\ y_{t-1}, & t \geq t_x + 1. \end{cases} \tag{47}$$

For $(i_1, \ldots, i_{d-1})$, when $t \leq t_x - 1$, we then have

$$z'_t = y'_t \oplus \bigoplus_{j=1}^{d-1} y'_{t-i_j} = y_t \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j}. \tag{48}$$

When $t \geq t_x + i_{d-1} + 1$, we will have

$$z'_t = y'_t \oplus \bigoplus_{j=1}^{d-1} y'_{t-i_j} = y_{t-1} \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j-1}. \tag{49}$$

Comparing with the noiseless case, in which we have $z_t = y_t \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j}$, we can see that $z'_t = z_t$ when $t \leq t_x - 1$, and $z'_t = z_{t-1}$ when $t \geq t_x + i_{d-1} + 1$. As the final decision is based on the value of Z, we are more interested in the difference between the summations $\sum_{t=i_{d-1}}^{\tilde{N}-1} z_t$ and $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$. Assuming the number of bits of $\{z_{i_{d-1}}, \ldots, z_{\tilde{N}-1}\}$ that are different from $\{z'_{i_{d-1}}, \ldots, z'_{\tilde{N}-1}\}$ is d_z, then the density, $P_{(t_x, i_{d-1}, \tilde{N})} = (d_z)/(\tilde{N} - i_{d-1})$, is dependent on the values of t_x, i_{d-1}, and $\tilde{N}$. For different values of t_x, i_{d-1}, and $\tilde{N}$, $P_{(t_x, i_{d-1}, \tilde{N})}$ has the following four different expressions:

1) $0 \leq t_x < i_{d-1} + 1$ and $t_x + i_{d-1} < \tilde{N} - 1$.
   In this case, $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ can be written as

$$\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t = \sum_{t=i_{d-1}}^{t_x + i_{d-1}} z'_t + \sum_{t=t_x+i_{d-1}+1}^{\tilde{N}-1} z'_t = \sum_{t=i_{d-1}}^{t_x + i_{d-1}} z'_t + \sum_{t=t_x+i_{d-1}}^{\tilde{N}-2} z_t. \tag{50}$$

   Considering the worst case, i.e., z'_t is independent of $z_t (\Pr(z'_t = z_t) = 0.5)$ when $i_{d-1} \leq t \leq t_x + i_{d-1}$, it can be seen that the number of bits that are different in

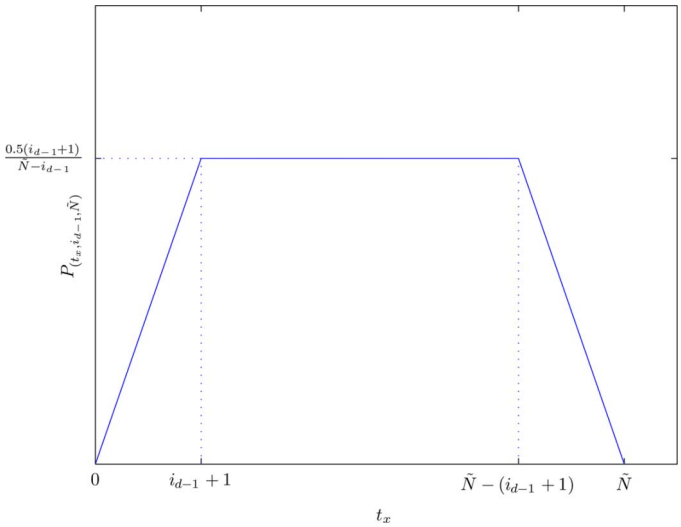

Fig. 10. Variations of $P_{(t_x,i_{d-1},\tilde{N})}$ versus $t_x(\tilde{N} \geq 2(i_{d-1}+1))$.

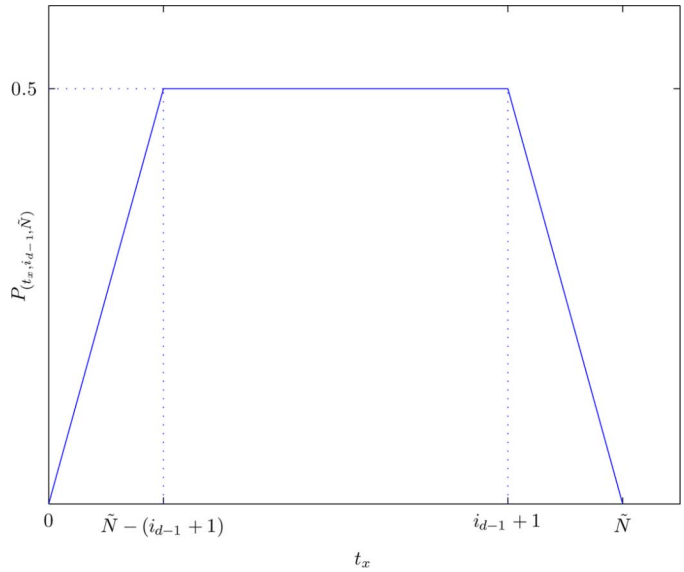

Fig. 11. Variations of $P_{(t_x,i_{d-1},\tilde{N})}$ versus $t_x(\tilde{N} < 2(i_{d-1}+1))$.

$\sum_{t=i_{d-1}}^{\tilde{N}-1} z_t$ and $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ is approximately $0.5(t_x+1)$ and $P_{(t_x,i_{d-1},\tilde{N})}$ is given by

$$P_{(t_x,i_{d-1},\tilde{N})} = \frac{0.5(t_x+1)}{\tilde{N}-i_{d-1}}. \tag{51}$$

2) $0 \leq t_x < i_{d-1}+1$ and $t_x + i_{d-1} \geq \tilde{N}-1$.
In this case, the number of bits that are different in $\sum_{t=i_{d-1}}^{\tilde{N}-1} z_t$ and $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ is approximately $(\tilde{N}-i_{d-1})/(2)$ and $P_{(t_x,i_{d-1},\tilde{N})}$ is given by

$$P_{(t_x,i_{d-1},\tilde{N})} = 0.5. \tag{52}$$

3) $i_{d-1}+1 \leq t_x < \tilde{N}$ and $t_x + i_{d-1} < \tilde{N}-1$.
In this case, $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ can be written as

$$\begin{aligned}\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t &= \sum_{t=i_{d-1}}^{t_x-1} z'_t + \sum_{t=t_x}^{t_x+i_{d-1}} z'_t + \sum_{t=t_x+i_{d-1}+1}^{\tilde{N}-1} z'_t \\ &= \sum_{t=i_{d-1}}^{t_x-1} z_t + \sum_{t=t_x}^{t_x+i_{d-1}} z'_t + \sum_{t=t_x+i_{d-1}}^{\tilde{N}-2} z_t. \end{aligned} \tag{53}$$

The number of bits that are different in $\sum_{t=i_{d-1}}^{\tilde{N}-1} z_t$ and $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ is approximately $0.5(i_{d-1}+1)$ and $P_{(t_x,i_{d-1},\tilde{N})}$ is given by

$$P_{(t_x,i_{d-1},\tilde{N})} = \frac{0.5(i_{d-1}+1)}{\tilde{N}-i_{d-1}}. \tag{54}$$

4) $i_{d-1}+1 \leq t_x < \tilde{N}$ and $t_x + i_{d-1} \geq \tilde{N}-1$.
In this case, $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ can be written as

$$\begin{aligned}\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t &= \sum_{t=i_{d-1}}^{t_x-1} z'_t + \sum_{t=t_x}^{\tilde{N}-1} z'_t \\ &= \sum_{t=i_{d-1}}^{t_x-1} z_t + \sum_{t=t_x}^{\tilde{N}-1} z'_t. \end{aligned} \tag{55}$$

The number of bits that are different in $\sum_{t=i_{d-1}}^{\tilde{N}-1} z_t$ and $\sum_{t=i_{d-1}}^{\tilde{N}-1} z'_t$ is approximately $0.5(\tilde{N}-t_x)$ and $P_{(t_x,i_{d-1},\tilde{N})}$ is given by

$$P_{(t_x,i_{d-1},\tilde{N})} = \frac{0.5(\tilde{N}-t_x)}{\tilde{N}-i_{d-1}}. \tag{56}$$

In Figs. 10 and 11, the variations of $P_{(t_x,i_{d-1},\tilde{N})}$ versus t_x according to our analysis above are depicted. When $\tilde{N} \geq 2(i_{d-1}+1)$, the variation of $P_{(t_x,i_{d-1},\tilde{N})}$ versus t_x will follow the curve shown in Fig. 10 and when $\tilde{N} < 2(i_{d-1}+1)$, the variation of $P_{(t_x,i_{d-1},\tilde{N})}$ versus t_x will follow the curve shown in Fig. 11.

From both figures, it can be observed that the insertion of one bit will have the least impact to the summation $\sum_{t=i_{d-1}}^{\tilde{N}-1} z_t$ when the bit is inserted at the two ends of the bit sequence. When the insertion point moves from the two ends to the middle of the bit sequence, the impact will increase and finally reach a maximum value, which depends on the values of i_{d-1} and $\tilde{N}$. According to [7], for a randomly chosen primitive polynomial of degree L, the minimum value of i_{d-1} is

$$i_{d-1} = ((d-1)!)^{\frac{1}{d-1}} 2^{\frac{L}{d-1}}. \tag{57}$$

For a small value of L, normally $\tilde{N} \geq 2(i_{d-1}+1)$ will hold and from Fig. 10, it can be observed that the maximum value of $P_{(t_x,i_{d-1},\tilde{N})}$ is $(0.5(i_{d-1}+1))/(\tilde{N}-i_{d-1})$. When L increases in value, and as i_{d-1} increases exponentially with L, finally $\tilde{N} < 2(i_{d-1}+1)$ will be satisfied. In this case, when $\tilde{N}-(i_{d-1}+1) \leq t_x < i_{d-1}+1$, $P_{(t_x,i_{d-1},\tilde{N})}$ will be as high as 0.5.

In the following, the impact of $P_{(t_x,i_{d-1},\tilde{N})}$ on the performance of the reconstruction will be discussed. For simplicity, we consider insertion of one bit in the scrambled bit sequence to be equivalent to passing y_t through a BSC channel with channel error probability p_{eq}. Let $P_{(p_{\text{eq}})}$ denote the density of the number of different bits in $\{z_{i_{d-1}}, \ldots, z_{\tilde{N}-1}\}$ and

$\{z'_{i_{d-1}}, \ldots, z'_{\tilde{N}-1}\}$ when passing y_t through a BSC with channel error probability p_{eq}; from (32) and (35), we have

$$P_{(p_{\text{eq}})} = \Pr(z'_t \oplus z_t = 1) = \Pr\left(e_t \oplus \bigoplus_{j=1}^{d-1} e_{t-i_j} = 1\right) = \frac{1}{2}\left[1 - (2\delta_{\text{eq}})^d\right] \tag{58}$$

where $\delta_{\text{eq}} = \frac{1}{2} - p_{\text{eq}}$. Let $P_{(p_{\text{eq}})} = P_{(t_x, i_{d-1}, \tilde{N})}$, it can be derived that

$$\delta_{\text{eq}} = \frac{1}{2}\sqrt[d]{1 - 2P_{(t_x, i_{d-1}, \tilde{N})}} \tag{59}$$

and consequently we have

$$p_{\text{eq}} = \frac{1}{2} - \delta_{\text{eq}} = \frac{1}{2}\left(1 - \sqrt[d]{1 - 2P_{(t_x, i_{d-1}, \tilde{N})}}\right). \tag{60}$$

According to (60), for each value of $P_{(t_x, i_{d-1}, \tilde{N})}$, there is a corresponding channel error probability P_{eq}. The bigger the value of $P_{(t_x, i_{d-1}, \tilde{N})}$, the bigger the value of p_{eq}. Details on the impact of the channel error probability on the performance of the reconstruction can be found in Section V-A. For most practical systems, we have $\tilde{N} \gg i_{d-1}$ and $(0.5(i_{d-1}+1))/(\tilde{N} - i_{d-1}) \approx 0$. According to Fig. 10, $P_{(t_x, i_{d-1}, \tilde{N})} \leq (0.5(i_{d-1}+1))/(\tilde{N} - i_{d-1})$; therefore, for any value of t_x we have $P_{(t_x, i_{d-1}, \tilde{N})} \approx 0$ and hence $p_{\text{eq}} \approx 0$. It means that when $\tilde{N} \gg i_{d-1}$, the performance of the reconstruction will not be affected by the insertion of one bit, no matter where the bit is inserted. However, when i_{d-1} becomes larger, especially when $i_{d-1} \geq 0.5\tilde{N}$, the value of p_{eq} will vary from 0 to 0.5 depending on where the bit is inserted. In the worst case, i.e., $p_{\text{eq}} = 0.5$, the reconstruction will never succeed no matter how big the bias ε is, because as described in Section V-A, when $p_{\text{eq}} = 0.5$, we have $\delta_{\text{eq}} = 0$ and after passing through the BSC channel, the new bias becomes $2\varepsilon\delta_{\text{eq}} = 0$. In this case, the two distributions depicted in Fig. 2 will overlap and, therefore, no multiple of the feedback polynomial can be detected.

VI. Conclusion

Cluzeau's algorithm is very promising in reconstructing the feedback polynomial of the LFSR used in a linear scrambler. In this paper, a scheme to improve the detection capability of Cluzeau's algorithm is proposed. Simulation results show that the detection capability is significantly improved by using the proposed scheme. A tighter upper bound for σ which approaches the actual σ more closely has also been derived. By using the new upper bound, the number of bits and operations required by Cluzeau's algorithm to reconstruct the feedback polynomial of the LFSR can be reduced significantly without affecting the detection capability. As the number of operations is reduced, the time required for reconstruction is also reduced. According to our analysis, the higher the weight of the sparse multiples to be searched, the higher the time reduction factor.

It should be noted that even with the improved algorithm, the value of ε will affect the number of bits and operations and in turn the running time of the algorithm significantly. For example, when the source bias is reduced from 0.1 to 0.01, the number of operations required to do the reconstruction will increase by at least 10^6 times. In this case, even for a feedback polynomial of very small degree, the time to do the detection will become very long. For example, for the first feedback polynomial in Table IV, the detection time will increase from 3 s to about 35 days! Fortunately, for natural source in practical situations, the typical values of ε are 0.1 and 0.05 [7].

Another issue investigated in this paper is on how to recover the scrambler in the presence of noise. Our analysis results show that when passing the scrambled bits through a BSC channel, the feedback polynomial of the LFSR still can be recovered by using the same method as the one proposed for the recovery of the LFSR in noiseless condition. The only difference is that the effective source bias is changed which depends on the channel error probability p. As the effective source bias is smaller than the original source bias when bit errors are present, the number of bits required in the reconstruction becomes larger in order to maintain the detection capability. The larger the value of p, the larger the number of bits required for the reconstruction. As the factor of increase in the number of bits varies a lot for different values of p and d, channel estimation is proposed to be used to get the statistical properties of the channel.

We have also investigated the problem of reconstruction of the scrambler when there is an insertion of one bit in the scrambled bit sequence. What we have found is that when the number of bits used in the reconstruction $(\tilde{N})$ is much larger than the minimum degree of the multiple of the feedback polynomial (i_{d-1}), the performance of the reconstruction will not be affected by the insertion of one bit no matter where the bit is inserted. However, with the increase in the degree of the feedback polynomial, the degradation in the performance of the reconstruction algorithm will vary a lot depending on where the bit is inserted. When the bit is inserted at the two ends of the bit sequence, the performance degradation is small. When the insertion point moves from the two ends to the middle of the bit sequence, the performance degradation increases until a maximum is reached, and the maximum performance degradation is dependent on $\tilde{N}$ and i_{d-1}. In the worst case, i.e., $i_{d-1} \geq 0.5\tilde{N}$ and the insertion of the bit is at the mid point of the sequence, the reconstruction will never succeed even though only one bit is inserted.

References

[1] R. Gautier, G. Burel, J. Letessier, and O. Berder, "Blind estimation of scrambler offset using encoder redundancy," in *Proc. Thirty-Sixth Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 3–6, 2002, vol. 1, pp. 626–630.

[2] E. Filiol, "Reconstruction of convolutional encoder over $\text{GF}(q)$," in *Proc. Sixth IMA Conf. Cryptography and Coding*, 1997, Lecture Notes in Computer Science, no. 1355, pp. 100–110, Springer Verlag..

[3] E. Filiol, "Reconstruction of punctured convolutional encoders," in *Proc. IEEE Int. Symp. Information Theory and Applications (ISITA'00)*, 2000, pp. 4–7, SITA and IEICE Publishing..

[4] J. Barbier, "Reconstruction of turbo-code encoders," in *Proc. SPIE Defense and Security Symp., Space Communications Technologies Conf.*, Mar. 28–31, 2005, vol. 5819, pp. 463–473.

[5] J. Barbier, G. Sicot, and S. Houcke, "Algebraic approach for the reconstruction of linear and convolutional error correcting codes," *Int. J. Appl. Math. Comput. Sci.*, vol. 3, no. 3, pp. 113–118, 2006.

[6] M. Côte and N. Sendrier, "Reconstruction of convolutional codes from noisy observation," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seoul, Korea, Jun. 28–Jul. 3 2009, pp. 546–550.
[7] M. Cluzeau and M. Finiasz, "Reconstruction of punctured convolutional codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Taormina, Sicily, Italy, Oct. 2009.
[8] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. C-34, no. 1, pp. 81–84, Jan. 1985.
[9] W. Meier and O. Staffelbach, "Fast correlation attack on certain stream ciphers," *J. Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.
[10] A. Canteaut and E. Filiol, "Ciphertext only reconstruction of stream ciphers based on combination generators," in *Proc. Seventh Int. Workshop Fast Software Encryption (FSE'00)*, 2000, pp. 165–180.
[11] X. Wu, S. N. Koh, and C. C. Chui, "Primitive polynomials for robust scramblers and stream ciphers against reverse engineering," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Austin, TX, June 13–18, 2010, pp. 2473–2477.
[12] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.
[13] M. Cluzeau, "Reconstruction of a linear scrambler," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1283–1291, Sep. 2007.
[14] R. P. Brent, S. Larvala, and P. Zimmermann, "A fast algorithm for testing reducibility of trinomials mod 2 and some new primitive trinomials of degree 3021377," *Math. Comp.*, vol. 72, no. 243, pp. 1443–1452, 2003.
[15] M. Zivkovic, "Generation of primitive binary polynomials," in *Proc. Int. Conf. Algebra, Logic & Discrete Mathematics (Niš1995)*, no. 9, pp. 961–965, part 3.
[16] E. R. Berlekamp, "Factoring polynomials over finite fields," *Bell Syst. Tech. J.*, vol. 46, pp. 1853–1859, 1967.
[17] B. Sklar, *Digital Communications, Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

Xiao-Bei Liu received the B.S. degree in electrical and communication engineering from Fudan University, Shanghai, China, in 1998, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2004.

From 1998 to 2000, she was an engineer with Datang Mobile Communications Equipment Co., Ltd. and from 2007 to 2010 she was a senior digital signal processing engineer in Wireless Sound Solutions Pte. Ltd. She is currently a research fellow in the Positioning and Wireless Technology Centre of NTU and her research interests include digital signal processing in wireless communications, modulation/coding techniques, and secured communications.

Soo Ngee Koh received the B.Eng. degree from the University of Singapore and the B.Sc. degree from the University of London, both in 1979. He received the M.Sc. and Ph.D. degrees from Loughborough University, U.K., in 1981 and 1984, respectively.

Prior to his return to Singapore, he worked as a consultant at the British Telecom Research Laboratories in England. He joined Nanyang Technological University (NTU) of Singapore in 1985. He was the founding Head of the Communication Engineering Division of the School of Electrical and Electronic Engineering (EEE) of NTU from 1995 to 2005, founding Cochair of the International Conference on Information, Communications and Signal Processing, and Associate Chair (Academic) from 2005 to 2011. He is currently a Professor of the School and Director (Undergraduate Education) of the University. He has published more than 140 papers in international journals and conference proceedings, and holds two international patents on speech coder design. His research interests include speech processing, coding, enhancement and recognition, computer-aided language learning, blind source separation, and secured communication.

Xin-Wen Wu (M'00) received the B.S. and M.S. degrees in 1989 and 1992, respectively, from East China Normal University, Shanghai, and the Ph.D. degree in 1995 from the Institute of Systems Science, Chinese Academy of Sciences, Beijing.

From 1995 through 2003, he was affiliated with the Institute of Mathematics, Chinese Academy of Sciences. From January to October 1996, and from October 1997 to December 1998, he was a visiting research associate at the Center for Advanced Computer Studies at the University of Louisiana, Lafayette. From June 1999 to May 2000, he was a postdoctoral researcher at the Department of Electrical and Computer Engineering, University of California at San Diego. During February 2003–October 2005, he worked at the Department of Electrical and Electronic Engineering, University of Melbourne, holding a research fellowship. From November 2005 through April 2010, he was a faculty member at the Graduate School of Mathematics and Information Technology, University of Ballarat. Since April 2010, he has been with the School of Information and Communication Technology, Griffith University, Gold Coast, Australia. His research interests are in the areas of coding theory, cryptology, information theory with applications to bioinformatics, and other areas. He has authored or coauthored over 40 research papers and one book in the above-mentioned areas.

Chee-Cheon Chui received the B.Eng. degree from the National University of Singapore, Singapore, in 1994, and the M.Sc. and Ph.D. degrees from the University of Southern California, in 2001 and 2005, respectively, all in electrical engineering.

He is currently with Temasek Laboratories at Nanyang Technological University, Singapore, as a research scientist, engaging in research and development and management of numerous projects in the field of wireless communications. He has also held various positions in the executive committee of the IEEE Singapore local Communications Chapter. His current research interests include receiver synchronization, time-synchronization of wireless systems, physical-layer security, wireless communication signal processing, and forward error correction coding.