A Study on Reconstruction of Linear Scrambler Using Dual Words of Channel Encoder

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

***Abstract*—In this paper, the reconstruction of the feedback polynomial as well as the initial state of a linear feedback shift register (LFSR) in a synchronous scrambler placed after a channel encoder is studied. The study is first based on the assumption that the channel is noiseless and then extended to the noisy channel condition. The dual words, which are orthogonal to the codewords generated by the channel encoder, are used in the reconstruction algorithm. The number of bits required by the new algorithm is compared with another recently proposed algorithm and results show that the number of bits required to do the reconstruction can be significantly reduced.**

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

I. Introduction

A LINEAR scrambler is usually used in a communication system to convert a data bit sequence into a pseudo-random sequence that is free from long strings of 1 s and 0 s. It is easy to implement with a wide variety of scrambler polynomials to choose from and the choice of which one to use has relatively little impact on the performance of the communication system. However, basing on the scrambler reconstruction technique detailed in [1], it is found in [2] that not all scrambler polynomials offer equal protection against reconstruction. In this work, we examined further the reconstruction of the feedback polynomial of a linear scrambler assuming the source bits are being encoded with forward error correction coding before being scrambled. The findings of this work are envisaged to aid the design of secured digital communication systems implemented in a flexible platform such as software defined radio (SDR). Our results point out what can be done to prevent reconstruction of a communication system; for example, various scrambler reconstruction techniques were proposed in [1]–[5]. The proposed approach will also add to the plethora of techniques for designing an intelligent receiver which can adapt itself to the different building blocks of the transmitter such as those proposed in [6]–[8]. It is also an extension of the results and findings on recovery of error-correcting codes which include linear block codes [9]–[11] and convolutional codes [12]–[16].

There are generally two types of linear scrambler, namely synchronous scrambler and self-synchronized scrambler. Both types of scrambler usually consist of a 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 modulo 2 summation. In this paper, for simplicity, only synchronous scramblers are considered. Reconstruction of a synchronous scrambler consists of reconstructing the feedback polynomial of the LFSR as well as its initial state. When some input and scrambled bits are known, the Berlekamp-Massey algorithm [3] can be used to reconstruct the feedback polynomial of the LFSR. In [4], a method is proposed to estimate the initial state of the LFSR from the scrambled sequence only, and by assuming that the feedback polynomial of the LFSR is also known. Recently, in [1], an algorithm is proposed by Cluzeau for reconstructing the feedback polynomial of the LFSR by only using the scrambled sequence. In the following, this algorithm will be referred to as Cluzeau's algorithm.

Although Cluzeau's algorithm is much more efficient than the brute force search algorithm in the recovery of the feedback polynomials of the LFSR, it is based on the critical assumption that the source bits, which XOR directly with the outputs of the LFSR, are distributed with a biased probability $\Pr(x_t = 1) = (1/2) - \varepsilon$, where $\varepsilon \neq 0$. Although this assumption usually holds for natural sources, when the source bits pass through a channel encoder before they are scrambled, the bias existing in the bit sequence might become very small. Consequently, the number of bits required to do the reconstruction becomes exorbitantly large. To deal with this problem, in this paper, a scheme is proposed to use the property of "dual words", which are orthogonal to the codewords generated by the channel encoder, instead of the bias existing in the encoded bit sequence, to achieve reconstruction of the scrambler. It can be observed that by using the proposed scheme, the number of bits required for reconstruction is reduced drastically.

The paper is organized as follows. In Section II, Cluzeau's algorithm is reviewed. In Section III, the bias existing in the encoded bit sequence after a channel encoder is analyzed. In Section IV, the scheme to recover the feedback polynomial as well as the initial state of the LFSR in a linear scrambler placed after a channel encoder is proposed. In Section V, the problem of reconstruction of the scrambler in the presence of channel noise is investigated. Some security propositions are given in the concluding section in Section VI.

Manuscript received May 17, 2012; revised November 27, 2012; accepted February 03, 2013. Date of current version February 27, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Y.-W. Peter Hong.

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

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

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

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.2013.2246515

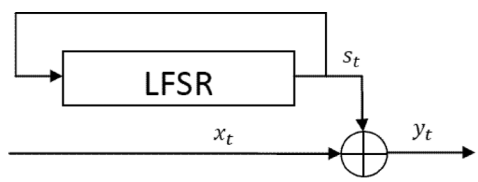


Fig. 1. Structure of synchronous scrambler.

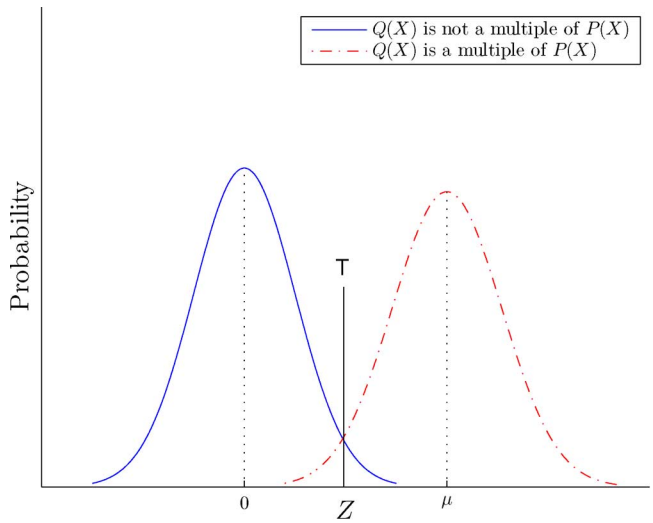


Fig. 2. Distributions of Z.

II. Cluzeau's Algorithm for Reconstructing a Synchronous Scrambler

In a synchronous scrambler, s_t is generated independently of x_t and y_t, as shown in Fig. 1.

Instead of brute force searching for the feedback polynomial $P(X)$ directly, Cluzeau's algorithm searches 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 modulo 2 summation of d scrambled bits, i.e., $z_t = y_t \oplus \bigoplus_{j=1}^{d-1} y_{t-i_j}$, $(0 < i_1 < i_2 < \cdots < i_{d-1})$, and N is the number of bits required for the reconstruction. Let $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$. When $Q(X)$ is a multiple of $P(X)$, we have

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

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 [1], z_t is biasedly distributed with $\Pr(z_t = 1) = (1/2)[1 - (2\varepsilon)^d]$, if the input bits are biasedly distributed with $\Pr(x_t = 1) = (1/2) - \varepsilon$, where $\varepsilon \neq 0$. Consequently, 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 the mean value μ given by

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

and the variance σ^2 [5] given by

$$\sigma^2 \leq (N - i_{d-1}) \left[1 + d\left((2\varepsilon)^2 - (2\varepsilon)^{2d}\right)\right]. \tag{5}$$

It can also be shown 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 the mean value 0 and the 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 $P(X)$, i.e., when $|Z| < T$, $Q(X)$ is not a multiple of $P(X)$; otherwise, $Q(X)$ is a multiple of $P(X)$. The threshold T and the number of bits required for the reconstruction N depend on two factors, i.e., the false-alarm probability P_f and the nondetection probability P_n. Let

$$a = \Phi^{-1}\left(1 - \frac{P_f}{2}\right) = \frac{T}{\sqrt{N - i_{d-1}}} \tag{6}$$

and

$$b = -\Phi^{-1}(P_n) = \frac{T - |\mu|}{\sigma}, \tag{7}$$

where Φ denotes the normal distribution function. From (6) and (7), it can be derived that the threshold T is

$$T = \frac{a(a + b\bar{\sigma}_l)}{(2|\varepsilon|)^d}, \tag{8}$$

and the number of bits required for the reconstruction is

$$N = i_{d-1} + \frac{(a + b\bar{\sigma}_l)^2}{(2\varepsilon)^{2d}}, \tag{9}$$

where $\bar{\sigma}_l$ is the normalized upper bound of σ, which is given by

$$\bar{\sigma}_l = \sqrt{[1 + d\left((2\varepsilon)^2 - (2\varepsilon)^{2d}\right)]}. \tag{10}$$

More detailed description of Cluzeau's algorithm can be found in [1] and [5].

III. Bias After Channel Encoder

In many communication systems, error correcting codes are used to combat errors introduced by the communication channel. In this work, we considered the case when the channel

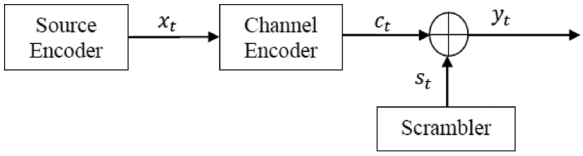


Fig. 3. Chain of scrambler and channel encoder.

encoder is placed between the source and the scrambler as shown in Fig. 3.

In the following, the bias existing in the encoded bit sequence after a channel encoder will be analyzed. Two commonly used error correcting codes are considered, i.e., linear block code and convolutional code.

A. Bias of a Bit Sequence After a Linear Block Encoder

Generally, for a (n, k) binary linear block code $\mathcal{C}$, where k is the number of information bits and n is the number of coded bits, a $k \times n$ generator matrix can be defined by the following $k \times n$ array:

$$\mathbf{G_b} = \begin{bmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & & & \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}, \tag{11}$$

where $g_{i,j} = (0 \text{ or } 1)$ and $\mathbf{g_0}, \mathbf{g_1} \cdots \mathbf{g_{k-1}}$ are linearly independent n-tuples that form a basis for $\mathcal{C}$. Considering a k-tuple message, i.e.,

$$\mathbf{x} = (x_0, x_1, \ldots, x_{k-1}),$$

the encoder transforms the message $\mathbf{x}$ independently into an n-tuple codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ by

$$\mathbf{c} = \mathbf{x} \cdot \mathbf{G_b} = (x_0, x_1, \ldots, x_{k-1}) \begin{bmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{bmatrix}. \tag{12}$$

Any encoded bit c_i $(i = 0, 1, \ldots, n-1)$ can be written as a linear binary summation of the message bits, i.e.,

$$c_i = g_{0,i}x_0 \oplus g_{1,i}x_1 \oplus \cdots \oplus g_{k-1,i}x_{k-1}. \tag{13}$$

Suppose the source bit sequence is produced by a biased and memoryless source with bias ε, and the number of nonzero terms (the weight) in the ith column of $\mathbf{G_b}$ is L_i $(i = 0, 1, \ldots, n-1)$, then the probability that $c_i = 1$ is given by

$$\begin{aligned} \Pr(c_i = 1) &= \sum_{l=1,3,\ldots}^{L_i} \binom{L_i}{l} \left(\frac{1}{2} - \varepsilon\right)^l \left(\frac{1}{2} + \varepsilon\right)^{L_i - l} \\ &= \frac{1}{2}\left[1 - (2\varepsilon)^{L_i}\right]. \end{aligned} \tag{14}$$

According to (14), the bias existing in the ith encoded bit c_i is $\varepsilon_{c_i} = 1/2(2\varepsilon)^{L_i}$. As $L_i \geq 1$ and $\varepsilon \leq 0.5$, we have $\varepsilon_{c_i} \leq \varepsilon$.

TABLE I
BIAS AFTER SOME BCH ENCODERS

(n,k)	Code rate	Bias obtained by (16)	Bias obtained by simulations
(7,4)	0.57	0.057	0.059
(15,11)	0.73	0.073	0.073
(15,7)	0.47	0.047	0.048
(31,26)	0.84	0.084	0.084
(31,21)	0.68	0.068	0.068
(31,16)	0.52	0.052	0.052
(63,57)	0.90	0.090	0.091
(63,51)	0.81	0.081	0.081
(63,45)	0.71	0.071	0.071
(63,39)	0.62	0.062	0.062
(63,36)	0.57	0.057	0.059

The bias existing in the whole encoded bit sequence, ε_{bc}, can be expressed by

$$\varepsilon_{bc} = \frac{1}{n}\sum_{i=0}^{n-1} \varepsilon_{c_i} \leq \frac{1}{n}\sum_{i=0}^{n-1} \varepsilon = \varepsilon. \tag{15}$$

From the above equation, it can be observed that the bias existing in the encoded bit sequence is less than or equal to the bias existing in the bit sequence before the encoder. Consider the systematic encoder, for which $L_0 = L_1 = \cdots = L_{k-1} = 1$ and $L_k, L_{k+1}, \cdots, L_{n-1} > 1$. The bias existing in the encoded bit sequence can be roughly estimated by

$$\varepsilon_{bc} = \frac{1}{n}\sum_{i=0}^{n-1} \varepsilon_{c_i} = \frac{k}{n}\varepsilon + \frac{1}{2n}\sum_{i=k}^{n-1} (2\varepsilon)^{L_i} \approx \frac{k}{n}\varepsilon. \tag{16}$$

To verify (16), the bias existing in the bit sequences of the output of the BCH encoders are obtained by computer simulations and results are shown in Table I. In each simulation, a bit sequence which contains $10000 \times k$ information bits is input into a BCH encoder (systematic encoder) and the simulation is repeated 100 times. The bias existing in the bit sequence before the encoder is set to 0.1. From Table I, it can be observed that the bias after the BCH encoder determined by the simulation results matches very well with that computed by (16).

B. Bias of a Bit Sequence After a Convolutional Encoder

An (n, k, m) convolutional code, where k is the number of information bits, n is the number of coded bits and m is the constraint length, can be defined by a $k \times n$ generator matrix $\mathbf{G_c}$ which consists of $k \times n$ binary "impulse responses" $\mathbf{g_j^{(i)}}$, where i denotes the ith input $(0 \leq i < k)$ and j denotes the jth output $(j \leq j < n)$, i.e.,

$$\mathbf{G_c} = (\mathbf{G_0}, \ldots, \mathbf{G_{n-1}}) = \begin{bmatrix} \mathbf{g_0^{(0)}} & \mathbf{g_1^{(0)}} & \cdots & \mathbf{g_{n-1}^{(0)}} \\ \mathbf{g_0^{(1)}} & \mathbf{g_1^{(1)}} & \cdots & \mathbf{g_{n-1}^{(1)}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g_0^{(k-1)}} & \mathbf{g_1^{(k-1)}} & \cdots & \mathbf{g_{n-1}^{(k-1)}} \end{bmatrix}, \tag{17}$$

where

$$\mathbf{g_j^{(i)}} = \left(g_j^{(i)}(0), g_j^{(i)}(1), \ldots, g_j^{(i)}(m-1)\right). \tag{18}$$

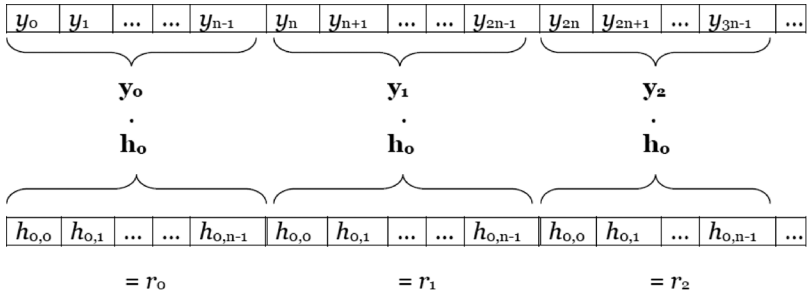


Fig. 4. Dot product of a dual word of a linear block code with the received bit sequence.

TABLE II
BIAS AFTER SOME RATE 1/2 CONVOLUTIONAL ENCODERS

m	G_0	G_1	Bias obtained by (20)	Bias obtained by simulations
3	111	101	0.012	0.012
4	1111	1011	0.0024	0.0024
5	10111	11001	0.0024	0.0024
6	101111	110101	$4.8e-4$	$4.8e-4$
7	1001111	1101101	$1.6e-4$	$1.6e-4$
8	10011111	11100101	$9.6e-5$	$9.3e-5$
9	110101111	100011101	$8.3e-5$	$7.6e-5$

Supposing the bit sequence at the ith input of the convolutional encoder is $\mathbf{x_i} = (x_{i,0}, x_{i,1}, \ldots)$, the bit sequence at the jth output is given by

$$\mathbf{c_j} = \mathbf{x_0} * \mathbf{g_j^{(0)}} \oplus \mathbf{x_1} * \mathbf{g_j^{(1)}} \oplus \cdots \oplus \mathbf{x_{k-1}} * \mathbf{g_j^{(k-1)}} = \sum_{i=0}^{k-1} \mathbf{x_i} * \mathbf{g_j^{(i)}}, \tag{19}$$

where $*$ is the convolution operation. Suppose the number of nonzero terms in $\mathbf{g_j^{(i)}}$ is $\tilde{L}_{i,j}$, then the bias of the whole encoded bit sequence, ε_{cc}, can be expressed as

$$\varepsilon_{cc} = \frac{1}{kn} \sum_{i=0}^{k-1} \sum_{j=0}^{n-1} \frac{1}{2} (2\varepsilon)^{\tilde{L}_{i,j}}. \tag{20}$$

To verify (20), the bias existing in the bit sequences after some optimum rate 1/2 convolutional code encoders [17] are obtained by computer simulations and results are shown in Table II. In each simulation, a bit sequence which contains 1,000,000 information bits is input into a convolutional encoder and the simulation is repeated 1000 times. The bias existing in the bit sequence before the encoder is assumed to be 0.1.

From Table II, it can again be observed that in general, the bias existing in the bit sequence after the sequence has passed through a convolutional encoder is very low as $\tilde{L}_{i,j}$ is normally > 2.

IV. RECONSTRUCTION OF THE SCRAMBLER AFTER A CHANNEL CODE

In the last section, our analysis shows that after passing through a channel encoder, the bias existing in the bit sequence drops, especially when convolutional codes are used. In this section, a novel scheme for reconstruction of the feedback polynomial and initial state of the LFSR in a scrambler which is placed after a channel encoder is proposed. This scheme exploits the property of dual words instead of the bias existing in the encoded bit sequence. In the following, the reconstruction of the scrambler placed after a linear block code will be considered first and after that, the proposed scheme will be extended to the case of convolutional code.

A. Reconstruction of the Scrambler After Linear Block Code

1) Reconstruction of the Feedback Polynomial of the LFSR: Consider a (n, k) binary linear block code $\mathcal{C}$ with $k \times n$ generator matrix $\mathbf{G_b}$. Rows in $\mathbf{G_b}$ form a basis for $\mathcal{C}$. The parity-check matrix for $\mathcal{C}$ is a $(n-k) \times n$ matrix $\mathbf{H_b}$ whose rows span the dual code $\mathcal{C}^{\perp}$, i.e.,

$$\mathbf{H_b} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & & & \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{bmatrix} \tag{21}$$

and $\mathbf{G_b} \cdot \mathbf{H_b^T} = 0$. $\mathbf{h_0}, \mathbf{h_1}, \ldots, \mathbf{h_{n-k-1}}$ denote rows $0, 1, \ldots, n-k-1$ in $\mathbf{H_b}$ and they are called dual words of $\mathcal{C}$.

To use the property of dual words to reconstruct the feedback polynomial of the LFSR, firstly, the received bit sequence $\mathbf{y} = (y_0, y_1, \ldots)$ is divided into blocks $\mathbf{y_0}, \mathbf{y_1}, \ldots$, with each block containing n bits, i.e., $\mathbf{y_t} = (y_{nt}, y_{nt+1}, \ldots, y_{(n+1)t-1})$. Then, a new sequence $\mathbf{r} = (r_0, r_1, \ldots)$ can be generated, in which each bit r_t is the dot product of $\mathbf{y_t}$ with a dual word, say $\mathbf{h_0}$, as shown in Fig. 4.

From Fig. 4, it can be seen that

$$\begin{aligned} r_0 &= \mathbf{y_0} \cdot \mathbf{h_0} = \sum_{i=0}^{n-1} y_i \cdot h_{0,i} \\ &= y_0 \cdot h_{0,0} \oplus y_1 \cdot h_{0,1} \oplus \cdots \oplus y_{n-1} \cdot h_{0,n-1} \\ r_1 &= \mathbf{y_1} \cdot \mathbf{h_0} = \sum_{i=0}^{n-1} y_{n+i} \cdot h_{0,i} \\ &= y_n \cdot h_{0,0} \oplus y_{n+1} \cdot h_{0,1} \oplus \cdots \oplus y_{2n-1} \cdot h_{0,n-1} \\ &\vdots \end{aligned} \tag{22}$$

As $\mathbf{y_t} = \mathbf{c_t} \oplus \mathbf{s_t}$, $(t = 0, 1, 2 \ldots)$, where $\mathbf{c_t}$ is the n-tuple codeword at time index t and $\mathbf{s_t} = (s_{nt}, s_{nt+1}, \ldots, s_{n(t+1)-1})$ are the outputs of the scrambler, we have

$$r_t = \mathbf{y_t} \cdot \mathbf{h_0} = \mathbf{c_t} \cdot \mathbf{h_0} \oplus \mathbf{s_t} \cdot \mathbf{h_0}. \tag{23}$$

According to the property of dual words, $\mathbf{c_t} \cdot \mathbf{h_0} = 0$; therefore, r_t can be written as

$$r_t = \mathbf{y_t} \cdot \mathbf{h_0} = \mathbf{s_t} \cdot \mathbf{h_0}, \tag{24}$$

i.e.,

$$\begin{aligned} r_0 &= s_0 \cdot h_{0,0} \oplus s_1 \cdot h_{0,1} \oplus \cdots \oplus s_{n-1} \cdot h_{0,n-1} \\ r_1 &= s_n \cdot h_{0,0} \oplus s_{n+1} \cdot h_{0,1} \oplus \cdots \oplus s_{2n-1} \cdot h_{0,n-1} \\ &\vdots \end{aligned} \tag{25}$$

Proposition 1: For a set of $d-1$ integers $(0 < i_1 < i_2 < \cdots < i_{d-1})$, if $r_t \oplus r_{t-i_1} \oplus r_{t-i_2} \oplus \cdots \oplus r_{t-i_{d-1}} \equiv 0$ for any $t \geq i_{d-1}$, then $1 + X^{ni_1} + X^{ni_2} + \cdots + X^{ni_{d-1}}$ is a multiple of the feedback polynomial $P(X)$.

Proof: According to (23), r_t can be written as

$$r_t = s_{nt} \cdot h_{0,0} \oplus s_{nt+1} \cdot h_{0,1} \oplus \cdots \oplus s_{n(t+1)-1} \cdot h_{0,n-1}. \tag{26}$$

Similarly,

$$\begin{aligned} r_{t-i_1} &= s_{n(t-i_1)} \cdot h_{0,0} \oplus s_{n(t-i_1)+1} \cdot h_{0,1} \oplus \\ &\quad \cdots \oplus s_{n(t-i_1+1)-1} \cdot h_{0,n-1} \\ &\vdots \\ r_{t-i_{d-1}} &= s_{n(t-i_{d-1})} \cdot h_{0,0} \oplus s_{n(t-i_{d-1})+1} \cdot h_{0,1} \oplus \\ &\quad \cdots \oplus s_{n(t-i_{d-1}+1)-1} \cdot h_{0,n-1}. \end{aligned} \tag{27}$$

Therefore,

$$\begin{aligned} &r_t \oplus r_{t-i_1} \oplus r_{t-i_2} \oplus \cdots \oplus r_{t-i_{d-1}} \\ &= \left(s_{nt} \oplus s_{nt-ni_1} \oplus \cdots \oplus s_{nt-ni_{d-1}}\right) \cdot h_{0,0} \\ &\quad \oplus \left(s_{nt+1} \oplus s_{nt+1-ni_1} \oplus \cdots \oplus s_{nt+1-ni_{d-1}}\right) \cdot h_{0,1} \oplus \cdots \\ &\quad \oplus \left(s_{n(t+1)-1} \oplus s_{n(t+1)-1-ni_1} \oplus \cdots \oplus s_{n(t+1)-1-ni_{d-1}}\right) \\ &\quad \cdot h_{0,n-1}. \end{aligned} \tag{28}$$

As $\mathbf{h_0}$ is a dual word, $h_{0,0}, h_{0,1}, \ldots, h_{0,n-1}$ cannot be all 0. Therefore, $r_t \oplus r_{t-i_1} \oplus r_{t-i_2} \oplus \cdots \oplus r_{t-i_{d-1}} \equiv 0$ only holds when $s_k \oplus s_{k-ni_1} \oplus \cdots \oplus s_{k-ni_{d-1}} \equiv 0$, i.e., $s_k \equiv s_{k-ni_1} \oplus \cdots \oplus s_{k-ni_{d-1}}$. It means $1+X^{ni_1}+X^{ni_2}+\cdots+X^{ni_{d-1}}$ is a multiple of the feedback polynomial $P(X)$. □

It is interesting to note that since the encoded bits are removed according to (24), the sequence $\mathbf{r}$ can be taken as a combination of some nth decimated sequences of the original sequence produced by the LFSR. Some properties of such a decimated sequence have been found in [19]. Actually, proposition 1 can also be proved by using properties of the decimated sequence proposed in [19].

From Proposition 1, it can be observed that when the sequence $\mathbf{r}$ is obtained, Cluzeau's algorithm, with only minor changes, can be applied to $\mathbf{r}$ to find the feedback polynomial of the LFSR. In the following, the scheme to determine the feedback polynomial of the LFSR in a scrambler placed after a channel encoder is described:

1) Divide the received bit sequence $\mathbf{y} = (y_0, y_1, \ldots)$ into blocks $\mathbf{y_0}, \mathbf{y_1}, \ldots$, with each block containing n bits.
2) Generate a new bit sequence $\mathbf{r}$, in which each bit r_t is the dot product of the received block with a dual word.
3) For $(i_1, \ldots, i_{d-1})$, $0 < i_1 < \ldots < i_{d-1} \leq D$, compute the number of bits in $\mathbf{r}$, N_r, required for the summation of $\tilde{Z}$. How to compute N_r will be described later. Let $N_c = i_{d-1} + N_r$.
4) Initialize $\tilde{Z}$ with $\tilde{Z} = 0$.
5) For t varying from $i_{d-1}+1$ to N_c, compute

$$\tilde{z}_t = r_t \oplus \bigoplus_{j=1}^{d-1} r_{t-i_j} \tag{29}$$

and

$$\tilde{Z} = \tilde{Z} + (-1)^{\tilde{z}_t} \tag{30}$$

6) If $\tilde{Z} = N_r$, store $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j \cdot n}$ in a table.
7) For $Q'(X) \neq Q(X)$ in the table, compute the nontrivial greatest common divisor (gcd) of $(Q(X), Q'(X))$.

Steps 1 to 4 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.

The scheme proposed above is based on the fact that if $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j \cdot n}$ is a multiple of the feedback polynomial, $\tilde{z}_t$ will always be 0 for t varying from $i_{d-1}+1$ to N_c, and therefore, the value of $\tilde{Z}$ should be $N_c - i_{d-1} = N_r$. If $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j \cdot n}$ is not a multiple of the feedback polynomial, $\Pr(\tilde{z}_t = 1) = 0.5$ and $\tilde{Z}$ will be Gaussian distributed with the mean value 0 and the variance N_r. The distribution of $\tilde{Z}$ is shown in Fig. 5.

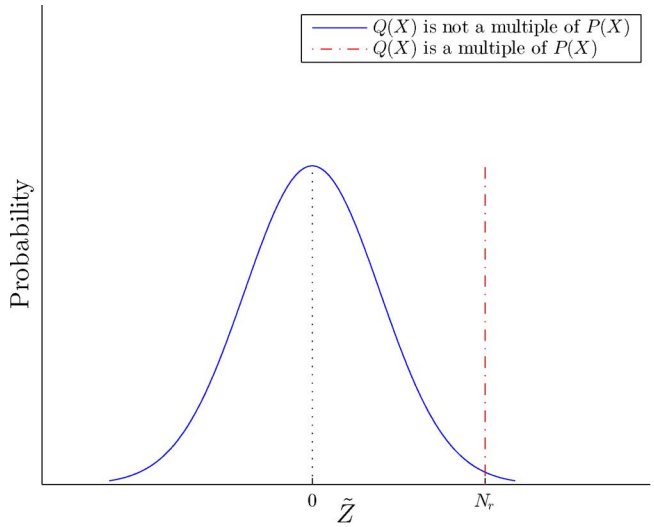


Fig. 5. Distributions of $\tilde{Z}$.

Similar to Cluzeau's algorithm, the number of bits in $\mathbf{r}$ used in the summation of $\tilde{Z}$, N_r, will affect the false-alarm probability P_f and nondetection probability P_n. As shown in Fig. 5, the value of $\tilde{Z}$ is always equal to N_r when $Q(X)$ is a multiple of $P(X)$. That means $P_n = 0$ when the proposed scheme is used. The false-alarm can happen only when $\tilde{Z} = N_r$ but $Q(X)$ is not a multiple of $P(X)$, and the probability is given by

$$\begin{aligned} P_f &= \Pr\left(\tilde{Z} = N_r | \text{when } Q(X) \text{ is not a multiple of } P(X)\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(N_r-\mu)^2}{2\sigma^2}}\Big|_{\mu=0,\sigma^2=N_r} \\ &= \frac{1}{\sqrt{2\pi N_r}} e^{-\frac{N_r}{2}}. \end{aligned} \tag{31}$$

TABLE III
SIMULATION RESULTS FOR RECONSTRUCTION OF SCRAMBLERS PLACED AFTER LINEAR BLOCK CODES

Code type	Feedback polynomial	Detected multiples	N	N_c
Hamming(7,4)	$x^8+x^4+x^3+x^2+1$	$x^{112}+x^7+1$, $x^{266}+x^{245}+1$	$3.7e7$	2065
	$x^9+x^6+x^4+x^3+1$	$x^{182}+x^{77}+1$, $x^{294}+x^{91}+1$		2408
	$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{399}+x^{42}+1$, $x^{490}+x^{427}+1$		3780
BCH(15,11)	$x^8+x^4+x^3+x^2+1$	$x^{150}+x^{120}+x^{75}+x^{30}+1$, $x^{165}+x^{75}+x^{60}+x^{15}+1$	$9.5e6$	3225
	$x^9+x^6+x^4+x^3+1$	$x^{180}+x^{165}+x^{90}+x^{75}+1$, $x^{210}+x^{150}+x^{75}+x^{45}+1$		3900
	$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{195}+x^{150}+x^{105}+x^{60}+1$, $x^{270}+x^{150}+x^{120}+x^{15}+1$		4800
BCH(31,26)	$x^8+x^4+x^3+x^2+1$	$x^{248}+x^{155}+x^{93}+x^{62}+1$, $x^{403}+x^{155}+x^{124}+x^{93}+1$	$4.1e6$	14043
	$x^9+x^6+x^4+x^3+1$	$x^{372}+x^{217}+x^{186}+x^{31}+1$, $x^{434}+x^{310}+x^{62}+x^{31}+1$		15004
	$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{310}+x^{217}+x^{155}+x^{93}+1$, $x^{620}+x^{465}+x^{62}+x^{31}+1$		20770

It can be observed that a small value of N_r, say 50, can already make $P_f < 10^{-10}$. The total number of bits in $\mathbf{r}$ used in the reconstruction is $i_{d-1}+N_r$. According to (22) and Fig. 4, each bit in $\mathbf{r}$ is a dot product of a dual word with a received block consisting of n bits. Therefore, the total number of bits required by the proposed scheme is

$$N_c = (i_{d-1}+N_r)n \approx (i_{d-1}+50)n. \tag{32}$$

Comparing (32) with (9), it can be observed that the number of bits required to do the reconstruction by the proposed algorithm does not depend on the bias ε anymore. Obviously, when ε is small, it is most probably that $N_c < N$. To show this fact clearer, the proposed algorithm is applied to reconstruct some feedback polynomials of LFSR in synchronous scramblers placed after different linear block codes. The number of bits required by the proposed algorithm (N_c) are shown in Table III. The number of bits required by Cluzeau's algorithm (N) are also shown in Table III for comparison. In the simulation, it is assumed that the bias existing in the bit sequence before the block encoder is 0.1 and $d = 3$. For Cluzeau's algorithm, it is assumed that $P_f = 10^{-7}$ and $P_n = 10^{-5}$. For the proposed algorithm, it is assumed that $N_r = 50$, which will lead to $P_n = 0$ and $P_f < 10^{-10}$.

From Table III, it can be observed that the number of bits required by the proposed algorithm to do the reconstruction is much lower than that required by Cluzeau's algorithm, especially when Hamming (7,4) code is used. This is because the property of the dual word is exploited by the proposed algorithm instead of the bias in the encoded bit sequence. Since the code rate of Hamming (7,4) code is the lowest among the 3 types of codes shown in Table III, the bias existing in the encoded bit sequence is also the lowest, and the number of bits required to do the reconstruction is the longest when Cluzeau's algorithm is used.

It should be noted that in Table III, the gcd of the two detected multiples is normally not the feedback polynomial but a multiple of the feedback polynomial. Suppose the gcd of the two detected multiples is $F(X)$. To find the correct feedback polynomial, $F(X)$ is firstly factorized. The correct feedback polynomial can then be found by descrambling the bit sequence by using each polynomial factor of $F(X)$ respectively, and see which one would lead to a descrambled bit sequence that satisfies the condition that the dot product of each codeword in the sequence with the dual words $\mathbf{h_i}$, $i = 0, 1 \ldots, n-k-1$ equals to 0. For example, the first two detected multiples in Table III are $x^{112}+x^7+1$ and $x^{266}+x^{245}+1$. Their gcd is $x^{56}+x^{42}+x^{35}+x^{21}+1$, which is the product of 3 polynomial factors $x^{24}+x^{20}+\cdots+1$, $x^{24}+x^{19}+\cdots+1$ and $x^8+x^4+x^3+x^2+1$. After descrambling the bit sequence by each polynomial factor, it is found that only $x^8+x^4+x^3+x^2+1$ leads to a sensible descrambled sequence. Hence, it is the correct feedback polynomial.

2) Reconstruction of the Initial State of the LFSR: After the feedback polynomial of the LFSR is determined, to descramble the received bit sequence, the initial state of the LFSR needs also to be recovered. In the following, a scheme to determine the initial state of the LFSR is described. This scheme is similar to the scheme proposed in [4], which also uses the encoder redundancy to determine the initial state of the LFSR.

Suppose the feedback polynomial of the LFSR is denoted by $P(X) = 1+a_1X+a_2X^2+\cdots+a_LX^L$, where L is the degree of the feedback polynomial and $a_i \in \{0,1\}$, then the output of the LFSR at time index t is

$$s_t = \sum_{i=1}^{L} a_i s_{t-i}. \tag{33}$$

Suppose the state of the LFSR at time index t is

$$\mathbf{S}_t = (s_t \quad s_{t+1} \quad s_{t+2} \quad \ldots \quad s_{t+L-1})^T \tag{34}$$

and a transition matrix F is defined as

$$F = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & a_{L-1} & a_{L-2} & \ldots & a_2 & a_1 \end{pmatrix}. \tag{35}$$

According to (33) and the property of the LFSR, the LFSR state at time index $t+i$, $(i = 0, 1, 2, \ldots)$ can be written as

$$\mathbf{S}_{t+i} = F^i \cdot \mathbf{S}_t. \tag{36}$$

Let the $1 \times L$ array U be defined as

$$U = (1 \quad 0 \quad 0 \quad \cdots \quad 0), \tag{37}$$

s_t can then be calculated by

$$s_t = U \cdot \mathbf{S}_t = U \cdot F^t \cdot \mathbf{S}_0. \tag{38}$$

According to (26) and (38), r_0 can be rewritten as

$$\begin{aligned} r_0 =& U \cdot \mathbf{S}_0 \cdot h_{0,0} \oplus U \cdot \mathbf{S}_1 \cdot h_{0,1} \oplus \cdots \oplus U \cdot \mathbf{S}_{n-1} \cdot h_{0,n-1} \\ =& U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \oplus \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \cdot \mathbf{S}_0 \end{aligned} \tag{39}$$

where I_L is a $L \times L$ identity matrix. Similarly, $r_1, r_2, \ldots, r_t$ can be rewritten as

$$\begin{aligned} r_1 =& U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \oplus \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \cdot F^n \cdot \mathbf{S}_0 \\ r_2 =& U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \oplus \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \cdot F^{2n} \cdot \mathbf{S}_0 \\ & \vdots \\ r_t =& U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \oplus \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \\ & \cdot F^{tn} \cdot \mathbf{S}_0. \end{aligned} \tag{40}$$

Suppose G is a $L \times L$ matrix that is given by

$$G = \begin{pmatrix} U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \cdot F^n \\ U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \cdot F^{2n} \\ \vdots \\ U \cdot (I_L \cdot h_{0,0} \oplus F \cdot h_{0,1} \cdots \oplus F^{n-1} \cdot h_{0,n-1}) \cdot F^{Ln} \end{pmatrix}. \tag{41}$$

Then the initial state $\mathbf{S}_0$ can be calculated by

$$\mathbf{S}_0 = G^{-1} \cdot (r_0 \quad r_1 \quad \cdots \quad r_{L-1})^T. \tag{42}$$

In many cases, there are more than one dual word for an error correcting code. According to (41), for the same feedback polynomial and different dual words, the matrices G are different. For each G and vector $(r_0\ r_1\ \cdots\ r_{L-1})$, an initial state $\mathbf{S}_0$ can be obtained by using (42). Obviously, if the feedback polynomial is the true feedback polynomial of the LFSR, $\mathbf{S}_0$ obtained from (42) are the same no matter which dual word is used. Otherwise, $\mathbf{S}_0$ obtained from different dual words are most likely to be different. This property can be used to determine the correct feedback polynomial of the LFSR without descrambling the bit sequence.

B. Reconstruction of the Scrambler After a Convolutional Code

Similar to linear block code, the generator matrix $\mathbf{G_c}$ of a (n, k, m) convolutional code generates a vector space of dimension k over the finite field $GF(2)$. This vector space has an orthogonal space of dimension $n-k$ and any element $(\mathbf{h_{c,0}}, \mathbf{h_{c,1}}, \ldots, \mathbf{h_{c,n-1}})$ in this space satisfies the property: $\sum_{j=0}^{n-1} \mathbf{g}_j^{(i)} * \mathbf{h_{c,j}} = 0\ \forall i \in [0, k-1]$. $(\mathbf{h_{c,0}}, \mathbf{h_{c,1}}, \ldots, \mathbf{h_{c,n-1}})$ can therefore be "translated" into a "dual word". Suppose $\mathbf{h_{c,j}} = (h_{c,j}^0, h_{c,j}^1, \ldots, h_{c,j}^{\tilde{m}-1})$ where $h_{c,j}^i = (0 \text{ or } 1)$. The binary vector

$$\mathbf{h_c} = \left(h_{c,0}^{\tilde{m}-1}, \ldots, h_{c,n-1}^{\tilde{m}-1}, \ldots, h_{c,0}^0, \ldots, h_{c,n-1}^0\right)$$

of length $n \times \tilde{m}$ will be the corresponding dual word.

After the dual word is obtained, the rest of the steps for reconstruction of the feedback polynomial and initial state of the LFSR are the same as those used for the linear block code. The only difference is that the received bit sequence is not divided into blocks. In fact, the dual word will be orthogonal to any segment of $n \times \tilde{m}$ bits in the coded sequence, when the starting offset of the $n \times \tilde{m}$ bits is n or a multiple of n. An example of the dot product of the dual word of a convolutional code with the received bit sequence is shown in Fig. 6.

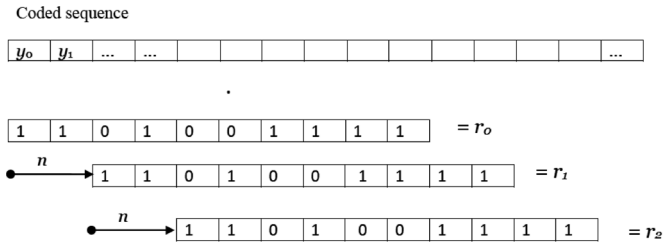


Fig. 6. Dot product of a dual word of a convolutional code with the received bit sequence.

In Fig. 6, the convolutional code is a (2,1,5) convolutional code with generator matrix [11011 11001]. It is found that the dual word of the convolutional code is 1101001111. As shown in Fig. 6, r_t is generated by making a dot product of the dual word with 10 bits in the coded sequence at time index t. For every increase of the time index t, the starting offset of the 10 bits will be increased by $n = 2$ bits. To see the effect of the proposed algorithm clearer, it is used to reconstruct some feedback polynomials of LFSR in synchronous scramblers placed after different convolutional codes with optimum distance spectrum [18]. The multiples detected and the number of bits required by the proposed algorithm are shown in Table IV. The number of bits required by Cluzeau's algorithm are also shown in Table IV for comparison. The setting of parameters for the simulation are the same as before.

From Table IV, it can be observed that the reduction of the number of bits required to do the reconstruction is very significant. This is because firstly, as described previously, the bias existing in the bit sequence after the sequence has passed through a convolutional encoder is very low, and consequently N is very big according to (9). Secondly, for convolutional code, the value of n is usually very small (<10), and consequently N_c is small according to (32). Therefore, the proposed scheme is the most suitable for convolutional code as the number of bits required by it to do the reconstruction is very small.

V. Reconstruction of Scrambler When Channel Noise is Present

In the previous sections, it is assumed that the channel is noiseless, i.e., there is no error in the received bit sequence. In practical situations, there is usually noise in the channel and some of the received bits will be wrong, as shown in Fig. 7. When channel errors are present, the dual words are no longer completely orthogonal to the received encoded bit sequence and the scheme proposed in Section IV cannot be applied directly.

Suppose the channel is modelled as a binary symmetric channel (BSC). The probabilities that the channel error e is equal to 1 and 0 are $\Pr(e = 1) = p = 0.5 - \delta$ and $\Pr(e = 0) = 1 - p = 0.5 + \delta$ respectively. Let

TABLE IV
SIMULATION RESULTS FOR RECONSTRUCTION OF SCRAMBLER PLACED AFTER CONVOLUTIONAL CODES

Code type	Feedback polynomial	Detected multiples	N	N_c
Convolutional (2,1,5) code	$x^8+x^4+x^3+x^2+1$	$x^{42}+x^{20}+1$, $x^{50}+x^2+1$	7.3e15	200
	$x^9+x^6+x^4+x^3+1$	$x^{72}+x^{38}+1$, $x^{84}+x^{10}+1$		268
	$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{46}+x^{14}+1$, $x^{130}+x^{40}+1$		360
Convolutional (3,1,5) code	$x^8+x^4+x^3+x^2+1$	$x^{33}+x^{15}+1$, $x^{120}+x^9+1$	5.8e16	540
	$x^9+x^6+x^4+x^3+1$	$x^{132}+x^{123}+1$, $x^{177}+x^{165}+1$		681
	$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{165}+x^{66}+1$, $x^{228}+x^{135}+1$		834
Convolutional (4,1,5) code	$x^8+x^4+x^3+x^2+1$	$x^{84}+x^{40}+1$, $x^{100}+x^4+1$	1.9e17	600
	$x^9+x^6+x^4+x^3+1$	$x^{144}+x^{76}+1$, $x^{168}+x^{20}+1$		872
	$x^{10}+x^6+x^5+x^3+x^2+x+1$	$x^{92}+x^{28}+1$, $x^{260}+x^{80}+1$		1240

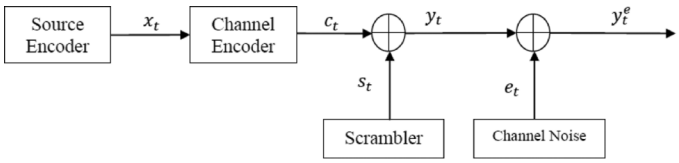


Fig. 7. Chain of scrambler, channel encoder, and channel.

the n-tuple channel errors at time index t be denoted by $\mathbf{e_t} = (e_{nt}, e_{nt+1}, \ldots, e_{(n+1)t-1})$; the n-tuple received codeword with errors, $\mathbf{y_t^e}$, is given by

$$\mathbf{y_t^e} = \mathbf{y_t} \oplus \mathbf{e_t}. \tag{43}$$

Since $\mathbf{y_t} = \mathbf{c_t} \oplus \mathbf{s_t}$, the dot product of the dual word $\mathbf{h_0}$ with the received bit sequence is given by

$$r_t^e = \mathbf{y_t^e} \cdot \mathbf{h_0}^T = \mathbf{c_t} \cdot \mathbf{h_0}^T \oplus \mathbf{s_t} \cdot \mathbf{h_0}^T \oplus \mathbf{e_t} \cdot \mathbf{h_0}^T. \tag{44}$$

According to the property of the dual word, we have $\mathbf{c_t} \cdot \mathbf{h_0}^T = 0$; therefore,

$$r_t^e = (\mathbf{s_t} \oplus \mathbf{e_t}) \cdot \mathbf{h_0}^T, \tag{45}$$

i.e.,

$$\begin{aligned} r_0^e &= (s_0 \oplus e_0) \cdot h_{0,0} \oplus (s_1 \oplus e_1) \cdot h_{0,1} \oplus \\ &\quad \cdots \oplus (s_{n-1} \oplus e_{n-1}) \cdot h_{0,n-1} \\ r_1^e &= (s_n \oplus e_n) \cdot h_{0,0} \oplus (s_{n+1} \oplus e_{n+1}) \cdot h_{0,1} \oplus \\ &\quad \cdots \oplus (s_{2n-1} \oplus e_{2n-1}) \cdot h_{0,n-1} \\ &\vdots \end{aligned} \tag{46}$$

Proposition 2: Suppose $\tilde{z}_t^e = r_t^e \oplus r_{t-i_1}^e \oplus r_{t-i_2}^e \oplus \cdots \oplus r_{t-i_{d-1}}^e$ $(t \geq i_{d-1})$. When $1 + X^{ni_1} + X^{ni_2} + \cdots + X^{ni_{d-1}}$ is not a multiple of the feedback polynomial $P(X)$, $\Pr(\tilde{z}_t^e = 1) = 1/2$. When $1 + X^{ni_1} + X^{ni_2} + \cdots + X^{ni_{d-1}}$ is a multiple of $P(X)$, $\Pr(\tilde{z}_t^e = 1) \leq 1/2[1 - (2\delta)^{wd}]$, where w is the weight of the dual word and $\delta = 0.5 - p$ (p is the channel crossover probability).

Proof: For linear block codes, r_t^e can be written as

$$\begin{aligned} r_t^e = {} & (s_{nt} \oplus e_{nt}) \cdot h_{0,0} \oplus (s_{nt+1} \oplus e_{nt+1}) \cdot h_{0,1} \oplus \\ & \cdots \oplus \left(s_{n(t+1)-1} \oplus e_{n(t+1)-1}\right) \cdot h_{0,n-1}. \end{aligned} \tag{47}$$

Similarly,

$$\begin{aligned} r_{t-i_1}^e = {} & \left(s_{n(t-i_1)} \oplus e_{n(t-i_1)}\right) \\ & \cdot h_{0,0} \oplus \left(s_{n(t-i_1)+1} \oplus e_{n(t-i_1)+1}\right) \\ & \cdot h_{0,1} \oplus \cdots \oplus \left(s_{n(t-i_1+1)-1} \oplus e_{n(t-i_1+1)-1}\right) \\ & \cdot h_{0,n-1}, \\ & \vdots \\ r_{t-i_{d-1}}^e = {} & \left(s_{n(t-i_{d-1})} \oplus e_{n(t-i_{d-1})}\right) \\ & \cdot h_{0,0} \oplus \left(s_{n(t-i_{d-1})+1} \oplus e_{n(t-i_{d-1})+1}\right) \\ & \cdot h_{0,1} \oplus \cdots \oplus \left(s_{n(t-i_{d-1}+1)-1} \oplus e_{n(t-i_{d-1}+1)-1}\right) \\ & \cdot h_{0,n-1}. \end{aligned} \tag{48}$$

Therefore,

$$\begin{aligned} \tilde{z}_t^e = {} & r_t^e \oplus r_{t-i_1}^e \oplus r_{t-i_2}^e \oplus \cdots \oplus r_{t-i_{d-1}}^e \\ = {} & \left(s_{nt} \oplus s_{nt-ni_1} \oplus \cdots \oplus s_{nt-ni_{d-1}}\right) \cdot h_{0,0} \\ & \oplus \left(e_{nt} \oplus e_{nt-ni_1} \oplus \cdots \oplus e_{nt-ni_{d-1}}\right) \cdot h_{0,0} \\ & \oplus \left(s_{nt+1} \oplus s_{nt+1-ni_1} \oplus \cdots \oplus s_{nt+1-ni_{d-1}}\right) \cdot h_{0,1} \\ & \oplus \left(e_{nt+1} \oplus e_{nt+1-ni_1} \oplus \cdots \oplus e_{nt+1-ni_{d-1}}\right) \cdot h_{0,1} \\ & \vdots \\ & \oplus \left(s_{n(t+1)-1} \oplus \cdots \oplus s_{n(t+1)-1-ni_{d-1}}\right) \cdot h_{0,n-1} \\ & \oplus \left(e_{n(t+1)-1} \oplus \cdots \oplus e_{n(t+1)-1-ni_{d-1}}\right) \cdot h_{0,n-1}. \end{aligned} \tag{49}$$

According to the property of the LFSR, when $1 + X^{ni_1} + X^{ni_2} + \cdots + X^{ni_{d-1}}$ is not a multiple of $P(X)$, and as $\Pr(s_t = 1) = 1/2$, it is apparent that $\Pr(\tilde{z}_t^e = 1) = 1/2$. When $1 + X^{ni_1} + X^{ni_2} + \cdots + X^{ni_{d-1}}$ is a multiple of $P(X)$, $s_k \oplus s_{k-ni_1} \oplus \cdots \oplus s_{k-ni_{d-1}} = 0$ for any $k \geq ni_{d-1}$ and we have

$$\begin{aligned} \tilde{z}_t^e = {} & \left(e_{nt} \oplus e_{nt-ni_1} \oplus \cdots \oplus e_{nt-ni_{d-1}}\right) \cdot h_{0,0} \\ & \oplus \left(e_{nt+1} \oplus \cdots \oplus e_{nt+1-ni_{d-1}}\right) \cdot h_{0,1} \oplus \cdots \\ & \oplus \left(e_{n(t+1)-1} \oplus \cdots \oplus e_{n(t+1)-1-ni_{d-1}}\right) \cdot h_{0,n-1}. \end{aligned} \tag{50}$$

In (50), $\tilde{z}_t^e$ is a modulo 2 summation of wd channel errors e, where w is the weight of the dual word. Similar to (14), it can be derived that

$$\begin{aligned} \Pr\left(\tilde{z}_t^e = 1\right) &= \sum_{l=1,3,\ldots}^{wd} \binom{wd}{l} \left(\frac{1}{2} - \delta\right)^l \left(\frac{1}{2} + \delta\right)^{wd-l} \\ &= \frac{1}{2}\left[1 - (2\delta)^{wd}\right]. \end{aligned} \tag{51}$$

For convolutional codes, similarly, $\tilde{z}_t^e$ is a modulo 2 summation of wd channel errors e. However, according to Fig. 6, some of the channel errors might be overlapped; therefore, we have

$$\Pr\left(\tilde{z}_t^e = 1\right) \le \frac{1}{2}\left[1 - (2\delta)^{wd}\right]. \tag{52}$$

□

Suppose $\tilde{Z}_e = \sum_{t=i_{d-1}+1}^{i_{d-1}+N_r^e} \tilde{z}_t^e$, where N_r^e is the number of bits in $\mathbf{r}$ required for the reconstruction when noise is present. According to Proposition 2 and the scheme described in Section IV, when $Q(X) = 1 + X^{ni_1} + X^{ni_2} + \cdots + X^{ni_{d-1}}$ is not a multiple of $P(X)$, $\tilde{Z}_e$ is Gaussian distributed with the mean value 0 and variance N_r^e. Similar to the derivation of the distribution of Z [5], when $Q(X)$ is a multiple of $P(X)$, it can be derived that $\tilde{Z}_e$ is Gaussian distributed with the mean value $\mu_e = N_r^e(2\delta)^{wd}$ and variance $\sigma_e^2 \le N_r^e[1 + d((2\delta)^{2w} - (2\delta)^{2wd})]$. Therefore, the algorithm proposed in Section IV can still be used with a minor change in Step 4, i.e., a threshold T_e can be used to determine whether $Q(X)$ is a multiple of the feedback polynomial. Similar to Cluzeau's algorithm described in Section II, when the false-alarm probability P_f and the nondetection probability P_n are given, the threshold T_e can be determined by

$$T_e = \frac{a_e^2 + a_e b_e \sqrt{1 + d\left((2\delta)^{2w} - (2\delta)^{2wd}\right)}}{(2\delta)^{wd}} \tag{53}$$

where

$$a_e = \Phi^{-1}(1 - P_f) = \frac{T_e}{\sqrt{N_r^e}} \tag{54}$$

and

$$-b_e = \Phi^{-1}(P_n) = \frac{T_e - \mu_e}{\sigma_e}. \tag{55}$$

From (54) and (55), it can be derived that the total number of bits N_c^e used in the reconstruction is given by

$$\begin{aligned} N_c^e &= n\left(i_{d-1} + N_r^e\right) \\ &= n\left(i_{d-1} + \frac{\left(a_e + b_e\sqrt{1 + d\left((2\delta)^{2w} - (2\delta)^{2wd}\right)}\right)^2}{(2\delta)^{2wd}}\right). \end{aligned} \tag{56}$$

In Figs. 8 and 9, the numbers of bits required for reconstruction when channel noise is present are shown for different error correcting codes and channel error probabilities. It is assumed that $d = 3$, $P_f = 10^{-7}$ and $P_n = 10^{-5}$. The feedback polynomial is assumed to be $x^8 + x^4 + x^3 + x^2 + 1$.

From Figs. 8 and 9, it can be observed that the number of bits required to do the reconstruction when channel noise is present is larger, as compared with that required in a noiseless condition. The larger the channel error probability, the larger the number of bits required to do the reconstruction. Another factor which affects the number of bits for the reconstruction is the dual word weight w. Obviously, with the increase of w, the number of bits required will increase accordingly, especially when the channel error probability is large. Therefore, for the same error correcting code, the dual word of minimum weight w is the best choice for the reconstruction.

In practical situations, the number of bits available for reconstruction is usually limited. In that case, the false-alarm probability or the nondetection probability will be affected. Suppose the number of bits in $\mathbf{r}$ available for reconstruction is $\bar{N}_r^e$ and the false-alarm probability is determined in advance, i.e., a_e is determined in advance. The threshold $\bar{T}_e$ is then given by

$$\bar{T}_e = a_e \cdot \sqrt{\bar{N}_r^e} \tag{57}$$

and the nondetection probability $\bar{P}_n$ can then be calculated by

$$\begin{aligned} \bar{P}_n &= \Phi\left(\frac{\bar{T}_e - \mu_e}{\sigma_e}\right) \\ &\approx \Phi\left(\frac{a_e \cdot \sqrt{\bar{N}_r^e} - \bar{N}_r^e(2\delta)^{wd}}{\sqrt{\bar{N}_r^e\left[1 + d\left((2\delta)^{2w} - (2\delta)^{2wd}\right)\right]}}\right). \end{aligned} \tag{58}$$

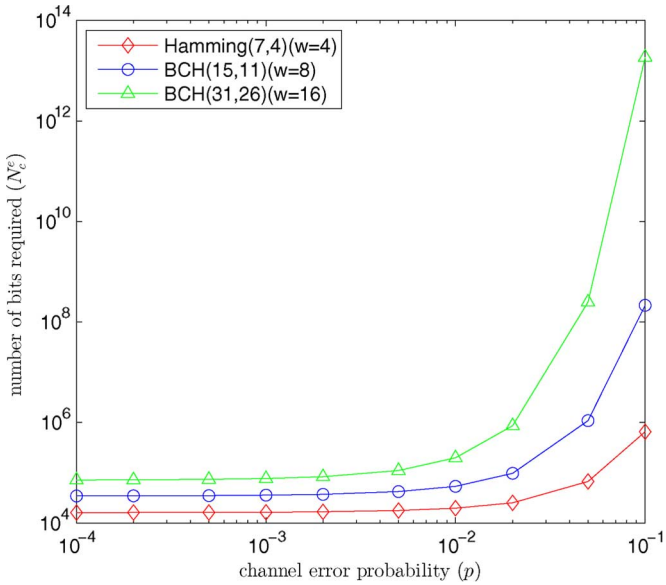


Fig. 8. Number of bits required for reconstruction when linear block codes are used and channel noise is present.

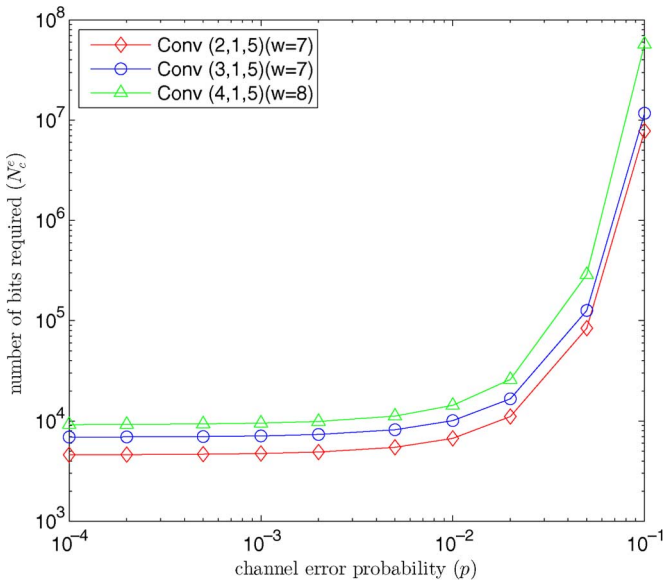


Fig. 9. Number of bits required for reconstruction when convolutional codes are used and channel noise is present.

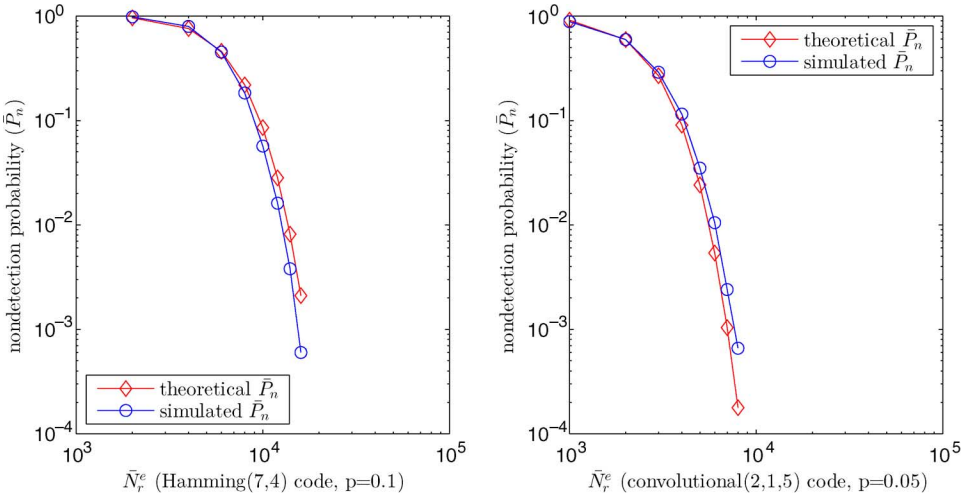


Fig. 10. Nondetection probabilities versus the number of bits available for reconstruction.

In Fig. 10, the nondetection probabilities versus different number of bits available for reconstruction are plotted. It is assumed that $d = 3$, $P_f = 10^{-7}$ and the feedback polynomial is $x^8 + x^4 + x^3 + x^2 + 1$.

For recovering the initial state of the LFSR when noise is present, some known techniques, such as those proposed in [20], [21], can be used.

VI. Conclusion

In this paper, the problem of reconstruction of the LFSR in a linear scrambler placed after a channel encoder is studied. The existing algorithm, i.e., Cluzeau's algorithm, is very promising in reconstructing the feedback polynomial based on the assumption that the source bits are biasedly distributed. However, after passing through a channel encoder, the bias (relative numbers of 1 s and 0 s) in the bit sequence drops, especially when a convolutional code is used, and the number of bits required by Cluzeau's algorithm will become exorbitantly large. In this paper, a new scheme which, instead of relying on the bias in the bit sequence, uses the orthogonality between the dual words and codewords generated by the channel encoder is studied. Our analysis shows that by using this proposed scheme, the feedback polynomial can be reconstructed much faster, as the number of bits required to do the reconstruction is reduced greatly, especially when convolutional codes are used as the error correcting codes. When channel noise is added, the above scheme can still be used to perform reconstruction, as long as the number of bits used to do the reconstruction is increased accordingly. It is noted that the larger the channel error probability, the larger the number of bits required to do the reconstruction.

Based on the above results, it is clear that scrambling the source bits before applying the FEC offers better protection against scrambler reconstruction when all else being equal.

Secondly, it has been shown that for a linear block code, the bias of the binary bits stream before scrambling can be approximated by the product of the bias of the source bits and the code rate (16). For convolutional encoder, the resultant bias is much lower (20). However, using dual words of the encoder, our results show that a convolutional code-linear scrambler pair is a much weaker pair compared with a linear block code-linear scrambler pair. This is because any shift of a multiple of n bits of a dual word is orthogonal to the coded sequence, and for most practical convolutional code, n is typically a small number.

The work presented in this paper is focused on determining the scrambler polynomial assuming dual word is known and word synchronization has been achieved *a priori*. A more challenging reconstruction problem would be to reconstruct both the code and the scrambler at the same time. One possible solution to this problem is to incorporate a scheme which recovers the code's length and achieves synchronization without considering the scrambler, such as schemes proposed in [10], [11] into the scheme proposed in this paper. For example, for a short linear block code or a convolutional code, an exhaustive search can be used to test all possible dual words and generate all possible $\mathbf{r}$. Obviously, after applying the scheme proposed in Section IV-A to $\mathbf{r}$, in noiseless case, only the $\mathbf{r}$ generated by the correct dual word will lead to two different distributions of Z as shown in Fig. 5. In a noisy condition, the situation is similar. For longer block codes, more sophisticated schemes need to be used for recovering both the code and the scrambler at the same time. Finally, the weight of the dual word plays a key part in the reconstruction, as low weight dual words are easier to be found and in noisy condition, low weight dual words lead to fewer bits required for the reconstruction. Therefore, one might consider using error correcting codes which do not have low weight dual words. How to find such codes is also an interesting topic for future work.

References

[1] M. Cluzeau, "Reconstruction of a linear scrambler," *IEEE Trans. Computers*, vol. 56, no. 9, pp. 1283–1291, Sep. 2007.

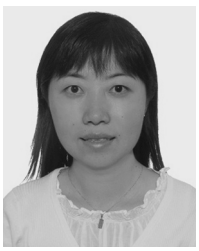
[2] X. Wu, S. N. Koh, and C. C. Chui, "Primitive polynomials for robust scramblers and stream ciphers against reverse engineering," in *Proc. IEEE ISIT*, Austin, TX, USA, Jun. 13–18, 2010, pp. 2473–2477.

[3] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[4] R. Gautier, G. Burel, J. Letessier, and O. Berder, "Blind estimation of scrambler offset using encoder redundancy," in *Proc. 36th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 3–6, 2002, vol. 1, pp. 626–630.

[5] 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.

[6] K. Umebayashi, S. Ishii, and R. Kohno, "Blind adaptive estimation of modulation scheme for software defined radio," in *Proc. PIMRC, 2000*, Sep. 18–21, 2000, vol. 1, pp. 43–47.
[7] H. Ishii, S. Kawamura, T. Suzuki, M. Kuroda, H. Hosoya, and H. Fujishima, "An adaptive receiver based on software defined radio techniques," in *Proc. 12th PIMRC*, USA, Sep. 2001, vol. 2, pp. 120–124.
[8] C. Han, A. Doufexi, S. Armour, K. H. Ng, and J. McGeehan, "Adaptive MIMO OFDMA for future generation cellular systems in realistic outdoor environment," in *Proc. IEEE VTC Spring*, May 2006, pp. 142–146.
[9] A. Valembois, "Detection and recognition of a binary linear code," *Discrete Appl. Math.*, vol. 111, no. 1–2, pp. 199–218, Jul. 2001.
[10] M. Cluzeau, "Block code reconstruction using iterative decoding techniques," in *Proc. IEEE ISIT*, Seattle, WA, USA, 2006, pp. 2269–2273.
[11] M. Cluzeau and M. Finiasz, "Recovering a code's length and synchronization from a noisy intercepted bitstream," in *Proc. IEEE ISIT*, Seoul, Korea, 2009, pp. 2737–2741.
[12] E. Filiol, "Reconstruction of convolutional encoder over $GF(q)$," in *Proc. Sixth IMA Conf. Cryptography and Coding*, 1997, no. 1355, pp. 100–110, Lecture Notes in Computer Science, Springer Verlag.
[13] 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.
[14] 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.
[15] J. Dingel and J. Hagenauer, "Parameter estimation of a convolutional encoder from noisy observations," in *Proc. IEEE ISIT*, Nice, France, 2007, pp. 1776–1780.
[16] M. Côte and N. Sendrier, "Reconstruction of convolutional codes from noisy observation," in *Proc. IEEE ISIT*, Seoul, Korea, Jun. 28–Jul. 3 2009, pp. 546–550.
[17] B. Sklar, *Digital Communications, Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[18] P. Frenger, P. Orten, and T. Ottosson, "Convolutional codes with optimum distance spectrum," *IEEE Commun. Lett.*, vol. 3, no. 11, pp. 317–319, Nov. 1999.
[19] E. Filiol, "Decimation attack of stream ciphers," in *Proc. INDOCRYPT 2000, LNCS 1977*, 2000, pp. 31–42, Springer Verlag.
[20] W. Meier and O. Staffelbach, "Fast correlation attack on stream ciphers," in *Proc. Advances in Cryptology (EUROCRYPT'88)*, 1988, vol. 330, pp. 301–314, Lecture Notes in Computer Science, Springer-Verlag.
[21] W. Meier and O. Staffelbach, "Fast correlation attack on certain stream ciphers," *J. Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.



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. 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.

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, USA, in 2001 and 2005, respectively, all in electrical engineering.

He is currently with TL@NTU, 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.

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, LA, USA. From Jun. 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.