Turbo Codes Can Be Asymptotically Information-Theoretically Secure

Xiao Ma
Department of ECE, Sun Yat-sen University
Guangzhou, GD 510275, China
E-mail: maxiao@mail.sysu.edu.cn

***Abstract*—This paper shows that a turbo-coded communication system can be made secure with a little bit of complexity cost. The classical permutation ciphers are revisited and analyzed. Firstly, the ideal stream permutation ciphers are shown to be asymptotically information-theoretically secure in the sense that the channel from plaintext to ciphertext has a vanished capacity, while the practical stream permutation ciphers are shown to be more secure than the classical stream ciphers in terms of protecting keys. Secondly, a necessary condition to break down a block permutation cipher is derived, which is then utilized to guarantee the computational security of a modified block permutation cipher. Thirdly, turbo ciphers (turbo-like codes with private interleavers) are proposed and analyzed.**

I. Introduction

Applications of error-correcting codes to cryptosystems began in 1978 when McEliece introduced a public-key cryptosystem [1] (which is called McEliece scheme in the literature) based on the fact that the decoding problem for a general linear code is an NP-hard problem [2]. Since then many public- and private-cryptosystems have been introduced, for example, see [3], [4], [5], etc. All there schemes are based on algebraic codes. In this paper, we will show that turbo codes [6] can be made secure with a little bit of complexity cost by combing with the classical permutation ciphers. The classical turbo code consists of two recursive and systematic convolutional codes concatenated parallelly by an interleaver π. The codeword from the turbo encoder can be written as $\underline{c} = (\underline{u}, \underline{p}^{(0)}, \underline{p}^{(1)})$, where $\underline{u}$ is the systematic bits, $\underline{p}^{(0)}$ is the parity check bits from the first encoder, and $\underline{p}^{(1)}$ is the parity check bits from the second encoder. Assume that $\underline{c}$ is transmitted over an insecure channel where an eavesdropper (called Oscar) exists. Assume that Oscar knows the turbo encoder except the interleaver π (called a *key*). After receiving $\underline{r} = (\underline{x}, \underline{y}^{(0)}, \underline{y}^{(1)})$, a noisy version of $\underline{c}$, what can Oscar do for estimating the data sequence $\underline{u}$? The maximum likelihood estimator is to solve such a problem as

$$\max_{\underline{u}'} \Pr(\underline{r}|\underline{u}') = \max_{\pi' \in \mathcal{S}} \max_{\underline{u}'} \Pr(\underline{r}|\underline{u}', \pi'), \tag{1}$$

where $\mathcal{S}$ is a subset of interleavers that contains the key π and has been determined by Oscar. The inner optimization can be solved approximately by turbo decoding algorithms. But, for the outer optimization, we make an assumption that *no efficient algorithms exist.* That is, we assume that the complexity of the outer optimization in (1) is of order $|\mathcal{S}|$. Hereafter, the cardinality of a finite set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. Even if Oscar solves the problem, there is not a shred of evidence that

$$\arg_{\underline{u}'} \max_{\pi' \in \mathcal{S}} \max_{\underline{u}'} \Pr(\underline{r}|\underline{u}', \pi') = \arg \max_{\underline{u}'} \Pr(\underline{r}|\underline{u}', \pi). \tag{2}$$

Now we may conclude that, for Oscar to *break* the system especially when the channel is noisy, what Oscar must do is to find the key π or reduce the size of $\mathcal{S}$ to be as small as possible based on $(\underline{u}, \underline{c})$ pairs (or other information) that he has collected (or selected) up to the present time. So what we need to do is to prevent Oscar from getting what he needs to attack the key.

The main results of this paper are outlined as follows. Firstly, we show that it requires at least $\lceil \log_2 L \rceil$ plaintext-ciphertext pairs to recover a private interleaver of size L. Motivated by this necessary condition, we propose a method to amend the block permutation cipher to be computationally secure. Secondly, we show that the encryption in a stream permutation cipher serves as a "channel" linking plaintext and ciphertext. We also show that this channel has an asymptotically vanished capacity. This implies that, asymptotically, Oscar can not get any useful information from ciphertexts. Compared with the classical stream ciphers, the stream permutation ciphers may have even higher security in terms of protecting the keystream. But, it is not secure in terms of protecting plaintexts having extremely low (or extremely high) weights. Finally, turbo codes with private interleavers, called *turbo ciphers*, are proposed, which can be considered as secret error-correcting codes [9]. Turbo ciphers provide both data security and data reliability, which may find applications in wireless communications.

II. Permutation Ciphers

A. Terminology

Let $\mathcal{A}$ be a finite set consisting of $|\mathcal{A}|$ symbols. Let $\underline{x} = (x_0, x_1, \cdots, x_{L-1})$ be a sequence in $\mathcal{A}^L$. A block interleaver of size L is a device that accepts $\underline{x}$ as input and delivers $\underline{y} = (y_0, y_1, \cdots, y_{L-1})$ as output such that $y_{\pi(i)} = x_i$, where $\pi : i \to \pi(i)$ is a *one-to-one* mapping from the index set $\mathcal{I} = \{0, 1, \cdots, L-1\}$ *onto* itself. For simplicity, the relation of $\underline{y}$ to $\underline{x}$ is written as $\underline{y} = \pi(\underline{x})$. An interleaver is uniquely determined by the index-mapping, which is also called a *permutation* in group theory. Consider symmetric group $\mathcal{S}_L$ consisting of all $L!$ different permutations of length L. A *uniform interleaver* U is a random variable which takes a particular permutation in

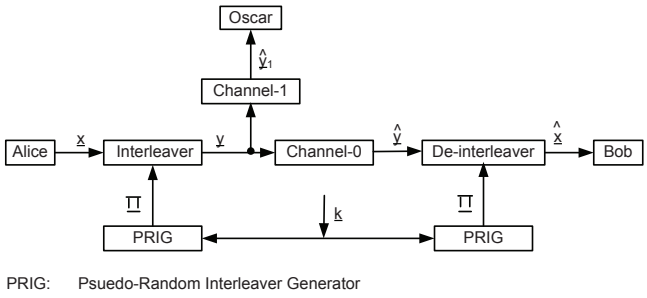

Fig. 1. A permutation cipher system with an adversary.

$\mathcal{S}_L$ with probability $1/L!$. In words, a uniform interleaver U is a probabilistic device that works as a particular permutation $\pi \in \mathcal{S}_L$ with probability $1/L!$.

As shown in Fig. 1, two parties, called Alice and Bob as in the literature, want to communicate with each other through an insecure channel, where a potential opponent, called Oscar, exists. Let $\underline{X} = (\underline{x}^{(0)}, \underline{x}^{(1)}, \ldots, \underline{x}^{(t)}, \cdots)$ be a sequence (finite or infinite), where $\underline{x}^{(t)} \in \mathcal{A}^L$ is referred to as a *plaintext* at time t. Let $\underline{\Pi} = (\pi_0, \pi_1, \cdots, \pi_t, \cdots)$ be a sequence of permutations from $\mathcal{S}_L$. Let $\underline{Y} = (\underline{y}^{(0)}, \underline{y}^{(1)}, \cdots, \underline{y}^{(t)}, \cdots)$, where $\underline{y}^{(t)} = \pi_t(\underline{x}^{(t)})$ is referred to as a *ciphertext* at time t. For Alice to send Bob a plaintext $\underline{x}^{(t)}$ at time t, she sends $\underline{y}^{(t)}$. Upon receiving $\hat{\underline{y}}^{(t)}$ from the channel-0, Bob does the inversion $\hat{\underline{x}}^{(t)} = \pi_t^{-1}(\hat{\underline{y}}^{(t)})$ to recover the plaintext. Note that the error probability $\Pr\{\hat{\underline{x}}^{(t)} \neq \underline{x}^{(t)}\}$ may not be zero due to the existence of the channel-0. Assume that the sequence of permutations $\underline{\Pi}$ is pseudo-randomly generated using an algorithm driven by a binary string $\underline{k}$ of length L_k. The binary string $\underline{k}$ is referred to as a *key*, which is randomly chosen by Alice and Bob while kept away from Oscar. We assume that the channel-1 is noiseless in the following analysis.

B. Block Permutation Ciphers

The system described above and shown in Fig. 1 is called a *block permutation cipher* if $\underline{\pi}_t = \pi$ for all $t \geq 0$. Oscar attempts to find $\underline{k}$, to find π, or to decrypt a ciphertext $\underline{y}^{(t)}$ at time t. Note that finding $\underline{k}$ is the most difficult task, whereas finding $\underline{x}^{(t)}$ is the least difficult task. There are several cases.

1) *Ciphertext only attack:*
Oscar can observe ciphertexts. Assume that $\mathcal{A}$ consists of English letters and that plaintexts are meaningful English texts. For small L_k, Oscar may use exhaustive search to see which plaintext is more meaningful. For large L_k, Oscar may use statistical analysis to find the plaintext [10][11]. Now assume that plaintexts are first encoded into binary bits by a known algorithm and then *bit-interleaved*. That is, assume that $\underline{x} \in \{0,1\}^L$. Then it is not obvious how to find the plaintext using statistical analysis. In the case when ASCII codes are used, for example, there will be 35 binary sequences of length 7 that may represent the letter "E" as well as the letter "F", which, definitely, leads to many problems when computing frequencies and higher-order statistics. Hereafter, we will assume that plaintexts are selected from $\{0,1\}^L$.
2) *Known plaintext attack:*
Oscar has collected $N \geq \lceil \log_2 L \rceil$ plaintext-ciphertext pairs. Then, under the assumption that the plaintexts are uniformly distributed over $\{0,1\}^L$, the probability that Oscar can determine the interleaver is
$$\frac{L!}{2^{NL}} \begin{pmatrix} 2^N \\ L \end{pmatrix} = \prod_{0 \leq i \leq L-1} \left(1 - \frac{i}{2^N}\right), \qquad (3)$$
which approaches 1 as N becomes large. But if $N < \lceil \log_2 L \rceil$, Oscar may not be able to determine the interleaver even using exhaustive search. Consider a simple example with $L = 3$. If Oscar knows $(0,1,1) \to (0,1,1)$ and $(0,1,0) \to (0,0,1)$, then he can determine that π is the transposition (1, 2). But if he only knows $(0,1,1) \to (0,1,1)$, after exhaustive search, he will find that there are two possible permutations that are indistinguishable when operating on the ciphertext $(0,1,1)$. So it is impossible for Oscar to determine the private key for this example.
3) *Chosen plaintext attack:*
Oscar is able to choose plaintexts and observe the corresponding ciphertexts. Then Oscar may choose as plaintexts the rows from the following matrix of size $\lceil \log_2 L \rceil \times L$,
$$M_x = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 1 & 0 & \cdots \\ 0 & 1 & 0 & 1 & 0 & \cdots \end{pmatrix}, \qquad (4)$$
where the i-th column is the binary representation of the index $i (0 \leq i \leq L-1)$.
4) *Chosen ciphertext attack:* Oscar is able to choose ciphertexts and observe the corresponding plaintexts. This case is equivalent to *chosen plaintext attack.*

In summary, we have

Proposition 1: Suppose that Oscar has obtained (in any way) N pairs of plaintext and ciphertext, denoted by $(\underline{x}^{(i)}, \underline{y}^{(i)})$, $0 \leq i \leq N-1$. Then a necessary condition for Oscar to find a *unique* permutation π satisfying that $\underline{y}^{(i)} = \pi(\underline{x}^{(i)})$ for all $0 \leq i \leq N-1$ is that the matrices
$$M_x = \begin{pmatrix} \underline{x}^{(0)} \\ \underline{x}^{(1)} \\ \vdots \\ \underline{x}^{(N-1)} \end{pmatrix} \text{ or } M_y = \begin{pmatrix} \underline{y}^{(0)} \\ \underline{y}^{(1)} \\ \vdots \\ \underline{y}^{(N-1)} \end{pmatrix} \qquad (5)$$
have distinct columns. Hence, it is necessarily required that $N \geq \lceil \log_2 L \rceil$.

Proof: It is omitted. ■

Motivated by the above necessary condition, we propose the following method to amend the block permutation cipher to be computationally secure. Let $f(D) = 1 + f_1 D +$

$\cdots + f_{\nu-1}D^{\nu-1} + f_\nu D^\nu$ be a primitive polynomial [12] of degree ν over $\mathbb{F}_2$, which is known to Oscar. Let $\underline{x} = (x_\nu, x_{\nu+1}, \cdots, x_{L-1})$ be the plaintext of length $L-\nu$ to be transmitted. The encrypter works as follows.

S1. Add random padding $\underline{p}$ of length ν to $\underline{x}$, that is, $\tilde{\underline{x}} \triangleq (\underline{p}||\underline{x})$.
S2. Calculate $u_t = \tilde{x}_t + \sum_{1\le i\le \nu} f_i u_{t-i}$ for $0 \le t \le L-1$, where $u_t = 0$ for $t < 0$.
S3. Check whether or not the Hamming weight of $\underline{u}$ (denoted by $W_H(\underline{u})$) is greater than a preset threshold w_{min}. If not, go to S1; otherwise go to next step.
S4. Calculate the ciphertext $\underline{y} = \pi(\underline{u})$.

It can be seen that $\underline{u}$ is randomly varied even if the plaintext $\underline{x}$ is fixed. At the receiver, an estimation of $\underline{u}$ (denoted by $\hat{\underline{u}}$) is first obtained by $\hat{\underline{u}} = \pi^{-1}(\underline{\hat{y}})$. The decrypter then checks if the Hamming weight of $\hat{\underline{u}}$ is greater than w_{min}. If not, report a failure and/or just randomly inverse some digits to meet this requirement. Finally, the plaintext is calculated by $\hat{x}_t = \hat{u}_t + \sum_{1\le i\le \nu} f_i \hat{u}_{t-i}$ for $\nu \le t \le L-1$. We have the following remarks.

- The information rate loss is negligible if ν/L is small.
- If $\hat{\underline{u}} = \underline{u}$, we have $\hat{\underline{x}} = \underline{x}$. Therefore the frame-error-rate will not be worsened. Also note that one digit error occurring in $\hat{\underline{u}}$ affects at most ν bits of $\hat{\underline{x}}$.
- The parameter ν is chosen to be large (say $\nu > 80$), while w_{min} is chosen to make $\binom{L}{w_{min}}$ to be large. As an example, for $L = 4096$ and $w_{min} = 5$, $\binom{L}{w_{min}} > 2^{53}$.

We assume that only $\underline{x}$ and/or $\underline{y}$ can be reached by Oscar. That is, Oscar can neither observe nor construct the intermediate sequence $\underline{u}$. How much effort is it required to find a pair $(\underline{u}, \underline{y})$? One possible way is to search over the random padding $\underline{p}$ to see if $\tilde{\underline{x}}$ can generate a sequence $\underline{u}$ of the same weight as $\underline{y}$. The other possible way is to search over π to see if $\pi^{-1}(\underline{y})$ generates the correct $\underline{x}$. The former has complexity of order 2^ν, while the latter has complexity of order at least $\binom{L}{w_{min}}$.

C. Stream Permutation Ciphers

We have seen that, to break down a block permutation cipher, one needs at least $\lceil \log_2 L \rceil$ plaintext-ciphertext pairs. This also suggests us that the security of permutation ciphers can be guaranteed by (pseudo-)randomly changing the interleavers every frame. The resulting system is called a *stream permutation cipher*. That is, the key stream $\underline{\Pi} = (\pi_0, \pi_1, \cdots, \pi_t, \cdots)$ shown in Fig. 1 is assumed to be a pseudo-random sequence and have an extremely large period. If this is the case, Oscar can only have one plaintext-ciphertext pair for a given permutation π_t. Therefore, Oscar may not be able to determine the interleaver π_t even using exhaustive search.

It is easily to see that the permutation cipher does not provide perfect secrecy [7]. Especially, the all-zero sequence and the all-one sequence cannot convey any secret information. However, we will show that the stream permutation cipher is asymptotically information-theoretically secure. Assume that the key stream π_t is sampled independently from a uniform

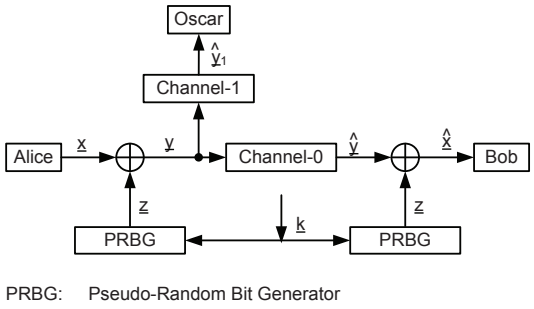

Fig. 2. The classical stream cipher.

interleaver. As mentioned by Shannon [7], the encryption serves as a channel that transforms a plaintext into a ciphertext according to the following probability law

$$\Pr(\underline{y}|\underline{x}) = \begin{cases} \frac{1}{\binom{L}{w}}, & W_H(\underline{y}) = W_H(\underline{x}) = w \\ 0, & otherwise \end{cases}. \quad (6)$$

We have the following theorem.

Theorem 1: The capacity of the channel that links $\underline{X}$ and $\underline{Y}$ is $\log(L+1)$.

Proof: Let $p(\underline{x})$ be an arbitrary distribution over $\{0,1\}^L$. Define $p_w = \Pr\{W_H(\underline{X}) = w\}$ and $q_w = \Pr\{W_H(\underline{Y}) = w\}$. Then $p_w = q_w$ and $p(\underline{y}) = \frac{p_w}{\binom{L}{w}}$ where $w = W_H(\underline{y})$. Therefore,

$$\begin{aligned} H(\underline{Y}) &= \sum_{0\le w\le L} p_w \left(\log \binom{L}{w} - \log p_w \right) \\ H(\underline{Y}|\underline{X}) &= \sum_{0\le w\le L} p_w \log \binom{L}{w} \\ I(\underline{X};\underline{Y}) &= H(\underline{Y}) - H(\underline{Y}|\underline{X}) \\ &= -\sum_{0\le w\le L} p_w \log p_w \le \log(L+1). \end{aligned}$$

■

This theorem shows that the maximum information provided by a ciphertext is $\log(L+1)$ bits, which actually corresponds to the weight of the plaintext. In other words, the average leaking information $\log(L+1)/L$ approaches zero as $L \to \infty$. When the plaintexts are uniformly distributed, a typical ciphertext provides essentially no useful information. However, if the plaintexts take only those sequences with different weights, the cipher is then collapsed down.

Corollary 1: $\lim_{L\to\infty} \frac{1}{L} H(\underline{X}|\underline{Y}) = \lim_{L\to\infty} \frac{1}{L} H(\underline{X}) \triangleq H(X)$.

Proof: From $I(\underline{X};\underline{Y}) = H(\underline{X}) - H(\underline{X}|\underline{Y})$, we have $H(\underline{X}) - \log(L+1) \le H(\underline{X}|\underline{Y}) \le H(\underline{X})$, which completes the proof by dividing L and taking limitation. ■

The above corollary means that the asymptotic equivocation is equal to the entropy of the source, and hence the system is said to be asymptotically information-theoretically secure [8].

In practice, the assumption of the independent uniformly distribution of the key stream is unrealistic. In the sequel, we assume that the key stream is a sequence of pseudo-random interleaves and show that the stream permutation

cipher system as shown in Fig. 1 is more secure (against key-recovery attacks) than the classical stream cipher system as shown in Fig. 2. To break these two stream ciphers, Oscar must be able to predict the output (called *keystream*) from the Pseudo-Random Bit Generator (PRBG) or the Pseudo-Random Interleaver Generator (PRIG). Assume that Oscar has collected (or chosen) T consecutive plaintexts $\underline{X} = (\underline{x}^{(0)}, \underline{x}^{(1)}, \cdots, \underline{x}^{(T-1)})$, while the corresponding ciphertexts are $\underline{Y} = (\underline{y}^{(0)}, \underline{y}^{(1)}, \cdots, \underline{y}^{(T-1)})$ for the classical stream cipher and $\underline{Y}' = (\underline{y}'^{(0)}, \underline{y}'^{(1)}, \cdots, \underline{y}'^{(T-1)})$ for the stream permutation cipher.

For the classical stream cipher system, Oscar can easily obtain the keystream by calculating $\underline{Z} = \underline{X} + \underline{Y} \bmod 2$. Since the output from the PRBG is not truly random, Oscar may derive some relations from $\underline{Z}$ that can be used to predict the future keystream.

For the stream permutation cipher system, Oscar knows $\underline{y}'^{(t)} = \pi_t(\underline{x}^{(t)})$. But it should be very difficult for him to know π_t exactly. This is because, to know π_t, Oscar must observe several input-output pairs for π_t, as discussed in Section II-B. So Oscar must know when π_t appears in the keystream and choose (or collect) as many as required plaintext-ciphertext pairs corresponding to the specific π_t. Therefore he needs more resources for the stream permutation system than what he needs for the classical stream cipher system.

Compared with the classical stream ciphers, the stream permutation ciphers have higher complexity and cause an undesired delay, which, however, will not be issues when implemented together with turbo-like codes.

III. Turbo Ciphers

A. Secret Error-Correcting Codes

The concept of *secret error-correcting code (SECC)* was first introduced by Hwang and Rao [9]. The SECC scheme provides both data security and data reliability by using a key-controlled error-correcting code. For comparison, the conventional secure communication system and the secure communication system using a SECC are shown in Fig. 3 (a) and (b), respectively.

The conventional system shown in Fig. 3 (a) works as follows. Alice encrypts a binary plaintext $\underline{x}$ using an encryption algorithm and a private key from the key generator. The ciphertext $\underline{y}$ is then put into the channel encoder (including modulation). The resulting signal sequence, denoted by $\underline{s}$, is then transmitted. Bob observes a noisy version of $\underline{s}$ through the channel-0, denoted by $\underline{r}$. Then Bob performs the channel demodulation/decoding algorithms to get an estimation of $\underline{y}$, denoted by $\hat{\underline{y}}$. Finally, Bob decrypts $\hat{\underline{y}}$ into $\hat{\underline{x}}$. The probability that $\hat{\underline{x}} \neq \underline{x}$ is the same as the probability that $\hat{\underline{y}} \neq \underline{y}$. However, one bit error in $\underline{y}$ may cause many errors in $\underline{x}$. At the same time, Oscar observes a noisy version of $\underline{s}$ through the channel-1, denoted by $\underline{r}_1$. Since Oscar knows the channel encoder, he is assumed to be able to perform demodulation/decoding algorithms to get $\hat{\underline{y}}_1$. If the channel-1 is not too bad, it is reasonable to assume that $\hat{\underline{y}}_1 = \underline{y}$. Therefore, traditionally, Oscar is assumed to know the ciphertext $\underline{y}$. Oscar attempts to find the key or the plaintext $\underline{x}$ based on what he has collected up to the present time, including plaintext-ciphertext pairs.

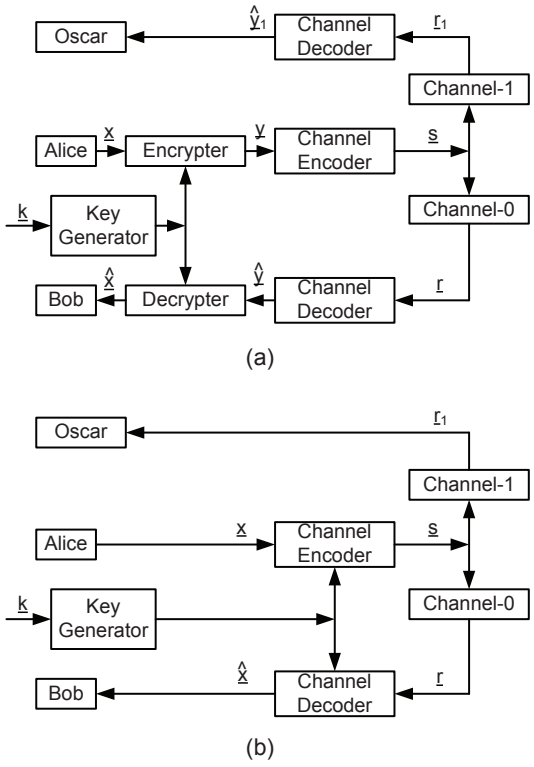

Fig. 3. (a) The conventional secure communication system. (b) The secure communication system using a secret error-correcting code.

The secure communication system using a SECC shown in Fig. 3 (b) is different. The channel encoder (as well as the decoder) is secret, which can be easily configured using the private key $\underline{k}$. Alice puts directly the plaintext $\underline{x}$ into the secret channel encoder, resulting in a signal sequence $\underline{s}$. For this system, $\underline{s}$ can be considered as the ciphertext. Bob observes a noisy version of $\underline{s}$ through the channel-0, denoted by $\underline{r}$. Then Bob performs the channel demodulation/decoding algorithms to get an estimation of $\underline{x}$, denoted by $\hat{\underline{x}}$. For Oscar, he observes a noisy version of $\underline{s}$, denoted by $\underline{r}_1$. Depending upon the channel-1, the probability that $\underline{r}_1 \neq \underline{s}$ may be non-neglected. If this is the case, Oscar can not even get an error-free ciphertext, a distinguished feature over the traditional secure system.

In the sequel, we assume that the secret channel encoder consists of a turbo code with private interleavers. The structure of the channel encoder/decoder is known to Oscar with the exception of the key to generate the interleaver. Note that the original turbo code can not be used here directly because Oscar may make hard decisions on systematic bits or decode only the first code to get $\hat{\underline{u}}$ without knowing π. The estimation $\hat{\underline{u}}$ may have erroneous bits but still contains lots of secrets especially when the bit-error-rate is far less than $1/2$. To avoid such attacks, the plaintext is transmitted in its interleaved version. That is, a codeword $\underline{c} = (\pi(\underline{u}), \underline{p}^{(0)}, \underline{p}^{(1)})$ consists of the interleaved version of plaintext, the parity check bits from the

first encoder and the parity check bits from the second one.

B. Stream Turbo Ciphers

Assume that the private key $\underline{k}$ is utilized to drive a PRIG, which outputs a sequence of interleavers. Since the interleavers are pseudo-randomly changed for every frame, we call such a system *a stream turbo cipher*.

As discussed in Section II-C, the stream turbo cipher will be at least as secure as the traditional stream cipher in terms of protecting the keystream even if the channel-1 is noiseless.But it is not secure in terms of protecting plaintexts with small or relatively large Hamming weights when the channel-1 is noiseless. A possible attack is as follows. Suppose that the error-free $\pi(\underline{u})$ has Hamming weight w. Then Oscar may encode each binary sequence with weight w to check if the parity checks are correctly produced. The complexity of such an exhaustive search is of order $\binom{L}{w}$. Therefore, plaintexts with extremely low weights (or extremely high weights) are not protected well. If $L = 4096$, then there are about 2^{53} sequences which have weight 5. So we may conclude that plaintexts having weight $5 \leq w \leq 4091$ for $L = 4096$ cannot be recovered efficiently. But it is easy to decrypt ciphertexts having lower weights, say, 1 or 0. However, this attack works only when the channel-1 is error-free.

C. Block Turbo Ciphers

Assume that a random interleaver is generated using a public algorithm driven by the private-key $\underline{k}$ and fixed "forever". Since the interleaver is unchanged for many frames, we call such a system *a block turbo cipher*. As discussed in Section II-B, the block turbo cipher is insecure against known-plaintext attacks provided that the channel-1 is noiseless. To make the block turbo cipher secure when the channel-1 is noiseless, we may turn to the method proposed in Section II-B which has a rate loss ν/L.

IV. Further Remarks and Conclusions

We have proposed turbo ciphers, which provide both data security and data reliability and may find applications in wireless communications. The ideal stream turbo ciphers can be asymptotically information-theoretically secure, while the practical stream turbo ciphers are even more secure than the classical stream ciphers against recovering the key-stream.

The block turbo ciphers are very different from the conventional block ciphers. Traditionally, an encryption is usually defined as a key-dependent transformation from $\mathbb{F}_2^L$ to $\mathbb{F}_2^L$. For the block turbo ciphers, the encryption is a key-dependent transformation from $\mathbb{F}_2^{(L-\nu)}$ to $\mathbb{F}_2^{L'}$, where $L-\nu$ is the length of the binary plaintext and L' is the length of the ciphertext to be transmitted over the physical channel. Compared with the existing block ciphers, the block turbo ciphers have the following features.

1) One plaintext can have thousands of bits. Therefore, we can choose $\nu > 100$ without sacrificing too much information rate. This makes block turbo ciphers secure against Struik-Tilburg attacks [13] and Meijers-Tilburg attacks [14].
2) One plaintext can results in lots of different ciphertexts. This makes it unclear how to implement differential attacks [15] on block turbo ciphers.
3) Ciphertexts can be real vectors when AWGN exists in the channel for Oscar. This makes it unclear how to implement linear attacks [16] on block ciphers.
4) Changing few bits in a ciphertext may not affect the delivered plaintext. This could be a useful property against possible chosen-ciphertext attacks. Actually, Oscar is not able to choose error-free ciphertexts because he does not know what sequences are legal codewords.

In addition, it appears that differential attacks and linear attacks are only applicable to iterated cryptosystems based on iterating a weaker round function [17]. It is desired to develop new attacks on the block turbo ciphers. The intended attacks should focus on obtaining *error-free* input-output pairs for the private interleaver. To conclude this paper, we remind readers of a fact that a randomly-chosen interleaver performs almost as well as an optimized deterministic interleaver in the low-SNR region especially when the data length is large.

References

[1] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Rep. 42-44. Jet Propulsion Laboratory, CA*, pp. 114–116, Jan. and Feb. 1978.

[2] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384–386, May 1978.

[3] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems of Control and Information Theory*, vol. 15, pp. 157–166, Feb. 1986.

[4] T. R. N. Rao and K. H. Nam, "Private-key algebraic-code encryptions," *IEEE Trans. Inform. Theory*, vol. 35, pp. 829–833, July 1989.

[5] X.-M. Wang, "Digital signature scheme based on error-correcting codes," *Electronics Letters*, vol. 26, pp. 898–899, June 1990.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[7] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, 1948.

[8] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, pp. 1355–1387, 1975.

[9] T. Hwang and T. R. N. Rao, "Secret error-correcting codes (SECC)," *Advances in Cryptology-CRYPTO'88*, LNCS 403, pp. 540–563, 1990.

[10] D. R. Stinson, *Cryptography: Theory and Practice*. New York: Chapman & Hall/CRC, 2002.

[11] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*. Upper Saddle River, NJ: Prentice-Hall, Inc., 2002.

[12] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. Cambridge University Press, 1994.

[13] R. Struik and J. van Tilburg, "The Rao-Nam scheme is insecure against a chosen-plaintext attack," in *Advances in Cryptology–CRYPTO'87*, pp. 445–457, New York: Springer-Verlag, 1987.

[14] J. Meijers and J. van Tilburg, "On the Rao-Nam private-key cryptosystem using linear codes," in *Proc. IEEE Int. Symp. Inform. Theory*, (Budapest, Hungary), June 1991.

[15] E. Biham and A.Shamir, "Differential cryptanalysis of DES-like cryptosystems (extended abstract)," *Advances in Cryptology-Crypto'90*. Springer-Verlag, pp.2-21, 1991.

[16] M. Matsui, "Linear cryptanalysis method for DES cipher," *Advances in Cryptology-EUROCRYPT'93*, Spinger-Verlag, LNCS 765, pp.386–397.

[17] D. Wagner, "Towards a unifying view of block cipher cryptanalysis," *Fast Software Encryption: 11-th international workshop, FSE 2004* (Delhi, India), pp.16–33, Feb. 2004.