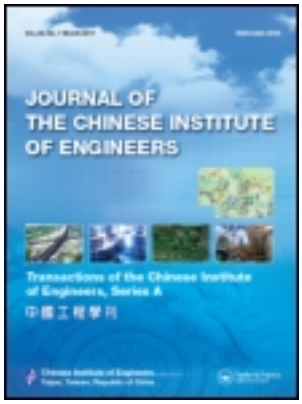


Journal of the Chinese Institute of Engineers
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcie20

An error-correcting stream cipher design with state-hopping architecture

Chih-Hsu Yen [a] & Bing-Fei Wu [b]

[a] Department of Electrical and Control Engineering , National Chiao Tung University , Hsinchu, Taiwan 300, R.O.C

[b] Department of Electrical and Control Engineering , National Chiao Tung University , Hsinchu, Taiwan 300, R.O.C Phone: 886-3-5131538 Fax: 886-3-5131538 E-mail:
Published online: 04 Mar 2011.

To cite this article: Chih-Hsu Yen & Bing-Fei Wu (2005) An error-correcting stream cipher design with state-hopping architecture, Journal of the Chinese Institute of Engineers, 28:1, 9-16, DOI: 10.1080/02533839.2005.9670968

To link to this article: http://dx.doi.org/10.1080/02533839.2005.9670968

AN ERROR-CORRECTING STREAM CIPHER DESIGN WITH STATE-HOPPING ARCHITECTURE

Chih-Hsu Yen and Bing-Fei Wu*

ABSTRACT

A new architecture of stream cipher based on state-hopping shift registers and a *pseudorandom vector generator* (PRVG) is introduced. The proposed stream cipher merges secrecy coding and channel coding into one processing step. It could be either a pure cryptosystem or a secrecy-channel coding by demand. Considered as cryptography, the PRVG generates pseudo random vectors which are treated as keystreams setting up the encryption scheme. This is different from the general concept in stream ciphers, state-hopping shift registers do not generate a pseudo random sequence but act as substitutions on plaintexts. Viewed as channel coding, the state-hopping shift registers play as the ones in convolution code and the PRVG generates a sequence of pseudorandom vector to determine the Trellis diagram. If the system acts as a pure cryptosystem, the decoding scheme is exactly the inverse of the encryption scheme. When the error-correcting ability is chosen, a modified sequential decoding is proposed.

Key Words: cryptography, stream cipher, error correction, channel coding, encryption.

I. INTRODUCTION

The great pace of internet, wireless services and multimedia services have led to an increasing demand for efficient, secure, and reliable digital data-transmission systems. In the general case of a transmitter, the data path follows the order: compression, encryption, and error control coding. For achieving efficient implementation and fast coding, hybrid systems which merge heterogeneous systems, e.g., source-channel coding, sourcesecrecy coding, or secrecy-channel coding, have been developed.

For source-secrecy coding, the partial encryption scheme is proposed by Cheng and Li (2000) to decrease processing time. There are several studies on sourcechannel coding (SCC) (Bystrom and Modestino, 2000; Ho and Kahn, 1996; Modestino *et al.*, 1981) which provide each priority class of information with a distinct data-resilience level, then processing time is lowered by coding significant data only. A technique for secure and reliable transmission of information was introduced by Denis and Kinsner (1999). It required two distinct coding systems to realize resilience and security.

McEliece (1978) presented a public-key cryptosystem based on t-error correcting Goppa code. The main idea is to add a random error vector with Hamming distance $t'<t$ to the encoded message before transmission. (Rao and Nam, 1987; Rao and Nam, 1989) proposed a similar approach, a private-key cryptosystem based on algebraic code. These two schemes execute secrecy coding and channel coding in one step. There are two definitions of secrecy-channel coding defined by Hwang and Rao (1990), the *Joint Encryption and Error Correction* (JEEC) scheme and the *Secret Error-Correcting Code* (SECC) scheme. However, the SECC scheme is attacked by Zeng *et al*. (2001) with a known-plaintext attack.

A new secrecy-channel coding is presented in this paper. The proposed stream cipher can be either a pure cryptosystem or secrecy-channel coding by demand. The design of the encryption scheme is based on the shift registers and the PRVG. In general, the

*Corresponding author. (Tel: 886-3-5131538; Fax: 886-3-5712385; Email: bwu@cssp.cn.nctu.edu.tw)

The authors are with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C.

contents of the registers and system parameters are initialized by a private key. Then, it becomes possible to encrypt the same plaintext into different ciphertexts by distinct private keys. There are various well-known attacks. The linear feedback shift registers (LFSRs), based on stream ciphers, are susceptible to various versions of the correlation attack (Meier and Staffebach, 1989; Siegenthaler, 1985; Zhang, 2000). When the pure cryptosystem is chosen, our decryption scheme is just the inverse of the encryption scheme. In secrecy-channel coding, the encryption scheme is a maximum likelihood decoding. Additionally, a new architecture, the shift register with state hopping, named *state-hopping shift register* (SHSR), is proposed here as secure core.

Section II defines the representation of the stream cipher and the symbols used throughout this manuscript, and the invertibility of the proposed system is also explained. The significant functions of our proposed scheme are separately described in Section III. Encryption schemes with and without the error correction ability, named *inverse SHSR* (iSHSR) and *state-hopping sequential algorithm* (SHS), are shown in Section IV. The simulation and discussion of our approach are presented in Section V. The conclusions are given in Section VI.

II. PRELIMINARY

Definition 1: A stream cipher is a tuple ($\mathcal{P}$, $\mathcal{C}$, $\mathcal{K}$, $\mathcal{L}$, $\mathcal{F}$, $\mathcal{E}$, $\mathcal{D}$), where the following conditions are satisfied (Stinson, 1995): $\mathcal{P}$ is a finite set of possible plaintexts. $\mathcal{C}$ is a finite set of possible ciphertexts. $\mathcal{K}$, the keyspace, is a finite set of possible keys. $\mathcal{L}$ is a finite set called the keystream alphabet. $\mathcal{F}=(f_1, f_2, \cdots)$ is the keystream generator. For $i>1$, f_i: $\mathcal{K}\times\mathcal{P}_{i-1}\rightarrow\mathcal{L}$. For each $z\in\mathcal{L}$, there is an encryption rule $E_z\in\mathcal{E}$ and a corresponding decryption rule $D_z\in\mathcal{D}$. $E_z : \mathcal{P}\rightarrow\mathcal{C}$ and $D_z : \mathcal{C}\rightarrow\mathcal{P}$ are functions such that $D_z(E_z(x))=x$ for every plaintext $x\in\mathcal{P}$.

Define $\mathbb{Z}_l^{m\times n}$ as a set of $m\times n$ matrixes whose entry belongs to the set {0, 1, 2, ..., l–1}. Our stream cipher ($\mathcal{P}$, $\mathcal{C}$, $\mathcal{K}$, $\mathcal{L}$, $\mathcal{F}$, $\mathcal{E}$, $\mathcal{D}$) is proposed, where $\mathcal{P}=\mathbb{Z}_2^{N_p\times 1}$, $\mathcal{C}=\mathbb{Z}_2^{N_C\times 1}$, $\mathcal{K}=\mathbb{Z}_2^{1\times N_k}$, $\mathcal{L}=\mathbb{Z}_M^{N\times 1}$, $\mathcal{F}$ is a PRVG by using the matrix method (Niederreiter, 1992), $\mathcal{E}$ is an N_p-SHSR, and $\mathcal{D}$ is an N_p-iSHSR or an SHS algorithm.

The description of the notations is shown below. N_p and N_c are the bit length of plaintext $P_i\in\mathcal{P}$ and of ciphertext $C_i\in\mathcal{C}$, respectively. The length of the private key $k\in\mathcal{K}$ is N_k bits. M is the modulus used by PRVG and N is the dimension of PRVG. When the error correction ability is chosen for reliable transmission, N_c is greater than N_p, because of the redundancy caused by the channel encoder; otherwise, N_c equals N_p. Instead of 7-tuple presentation

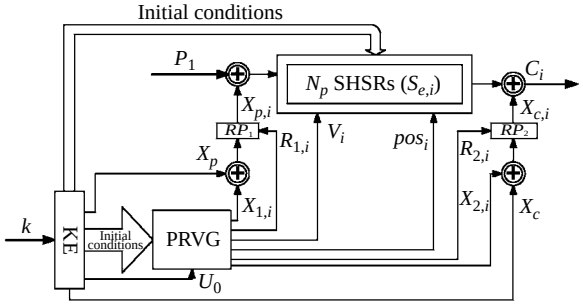


Fig. 1 Block diagram of the encryption scheme

of the system in Def. 1, the proposed stream cipher will be represented in a simple form, a 4-tuple (N_c, N_p, m, M) throughout the whole manuscript, where m is the number of registers in one SHSR.

The block diagram of the encryption scheme is shown in Fig. 1. The system has three major parts, Key Expansion (KE), N-dimensional PRVG, and N_p SHSRs. The KE enlarges the private keyspace, KE : $\mathcal{K}\rightarrow\tilde{\mathcal{K}}$, where $\tilde{\mathcal{K}}=\mathbb{Z}_2^{1\times N_{k_e}}$, to get the expansion key $k_e\in\tilde{\mathcal{K}}$ of bit length N_{k_e} required for initializing the system. In Fig. 1, exclusive-or pairs {$\boldsymbol{X_p}$, $\boldsymbol{X_c}$}, initial conditions of PRVG, and initial states of N_p SHSRs are initialized by $\tilde{k}\in\tilde{\mathcal{K}}$, a key expanded from k by KE. The PRVG determines the random permutation (RP) by giving $\boldsymbol{R_{1,i}}$ and $\boldsymbol{R_{2,i}}$, exclusive-or pairs {$\boldsymbol{X_{1,i}}$, $\boldsymbol{X_{2,i}}$}, and transition of of state diagram of SHSRs by giving V_i and pos_i. The SHSRs take as a bit-substitution function $S_{e,i}$. For each $z_i\in\mathcal{L}$, there is a corresponding $E_{z_i}\in\mathcal{E}$, consisting of $\boldsymbol{X_{p,i}}$, $\boldsymbol{X_{c,i}}$ and $S_{e,i}$, such that E_{z_i}: $\mathcal{P}\rightarrow\mathcal{C}$.

The decryption scheme is simply obtained by reversing the procedure of the encryption scheme and substituting $S_{d,i}$ into $S_{e,i}$.

1. The Invertibility (without Error Correction)

Given the ith plaintext $\boldsymbol{P_i}\in\mathcal{P}$, according to Fig. 1, the encryption E_z and decryption D_z are the following:

$$E_z: \mathcal{P}\rightarrow\mathcal{C},\ \boldsymbol{C_i}=E_{z_i}(\boldsymbol{P_i})$$
$$=S_{e,i}(\boldsymbol{P_i}\oplus\boldsymbol{X_{p,i}})\oplus\boldsymbol{X_{c,i}} \quad (1)$$

$$D_z: \mathcal{C}\rightarrow\mathcal{P},\ \boldsymbol{P_i}=D_{z_i}(\boldsymbol{C_i})$$
$$=S_{d,i}(\boldsymbol{C_i}\oplus\boldsymbol{X_{c,i}})\oplus\boldsymbol{X_{p,i}} \quad (2)$$

At the ith time, PRVG generates one subkey z_i and $\boldsymbol{X_{p,i}}$; $\boldsymbol{X_{c,i}}\in\mathbb{Z}_2^{N_p\times 1}$ are calculated from z_i. Both $S_{e,i}$ and $S_{d,i}$ are functions of pos_i, V_i, and the past data. Assuming $\boldsymbol{P_i}\in\mathbb{Z}_2^{N_p\times 1}$, the system is invertible if

$$S_{d,i}(S_{e,i}(\boldsymbol{P_i}))=\boldsymbol{P_i}=[p_{i,0}\ \ p_{i,1}\ \cdots\ p_i,\ N_{p-1}]^T \quad (3)$$

Because an SHSR of degree m is a linear combination of m memories in shift registers whose content is assignable, for N_p independent SHSRs, $S_{d,i}$ will equal $S_{e,i}$ when three arguments are correctly provided to $S_{d,i}$ and $S_{e,i}$. Thus we can use $\boldsymbol{S_i}$ to represent $S_{e,i}$ and $S_{d,i}$. In an N_p-SHSR, the $\boldsymbol{S_i}$ should be treated as N_p subsystems, $[s_{i,0}\ \ s_{i,1}\ \cdots\ s_{i,\ N_{p-1}}]$, and each $s_{i,n}$: $\mathbb{Z}_2^{1\times 1}\rightarrow\mathbb{Z}_2^{1\times 1}$. Given $p_{i,n}\in\mathbb{Z}_2^{1\times 1}$, $s_{i,n}$ acts as follows:

$$c_{i,n}=s_{i,n}(\boldsymbol{p_i})=(\boldsymbol{A_n}\otimes\boldsymbol{B_{i,n}})\oplus\boldsymbol{p_{i,n}} \quad (4)$$

$$\boldsymbol{A_n}=[a_{n,1}\ \ a_{n,2}\ \cdots\ a_{n,m}]$$

$$\boldsymbol{B_{i,n}}=[b_{n,1}\ \ b_{n,2}\ \cdots\ b_{n,m}]^T$$

where $\boldsymbol{A_n}\in\mathbb{Z}_2^{1\times m}$ is the coefficient vector of function of $\boldsymbol{B_{i,n}}\in\mathbb{Z}_2^{1\times N}$, contents in SHSR at the ith time. And $\boldsymbol{p_n}(x)=1+a_{n,1}x+a_{n,2}x^2+\cdots+a_{n,m}x^m$ is the polynomial representation of the nth SHSR. $\boldsymbol{A_n}$ is constant for a given system, and $\boldsymbol{B_{i,n}}$ is modified by randomly changing two entries at most for each $p_{i,n}$. Consider a subsystem $s_{i,n}$ in (3), at the ith time, then

$$\begin{aligned}s_{i,n}(s_{i,n}(p_{i,n}))&=s_{i,n}((\boldsymbol{A_n}\otimes\boldsymbol{B_{i,n}})\oplus p_{i,n})\\&=(\boldsymbol{A_n}\otimes\boldsymbol{B_{i,n}})\oplus((\boldsymbol{A_n}\otimes\boldsymbol{B_{i,n}})\oplus p_{i,n})\\&=p_{i,n}\end{aligned} \quad (5)$$

Substituting (5) into (3), it yields $S_i(\boldsymbol{P_i})=[s_{i,0}(p_{i,0})\ \ s_{i,1}(p_{i,1})\ \cdots\ s_{i,N_p}(p_{i,\ N_{p-1}})]$. According to (5), $S_i(S_i(\boldsymbol{P_i}))=\boldsymbol{P_i}$, the cryptosystem is invertible, i.e., $P=D_k(E_k(\boldsymbol{P}))$.

2. The Invertibility (with Error Correction)

In the case with error correction, the encryption scheme is similar to (1) except for the relationship of N_p and N_c. Given a system (N_p, N_c, m, M), the encryption is E_z: $\mathcal{P}\rightarrow\mathcal{C}$, and the substitution function of N_p-SHSR is $S_{e,i}$: $\mathbb{Z}_2^{N_p\times 1}\rightarrow\mathbb{Z}_2^{N_c\times 1}$. Given the ith plaintext $\boldsymbol{P_i}\in P$, $\boldsymbol{P_i}$ is encrypted as $C_i=E_{z_i}(\boldsymbol{P_i})=S_{e,i}(\boldsymbol{P_i}\oplus\boldsymbol{X_{p,i}})\oplus\boldsymbol{X_{c,i}}$, and each matrix $\boldsymbol{A_n}$ in $S_{e,i}$ is

$$\boldsymbol{A}_n=\begin{bmatrix}a_{0,1}&a_{0,2}&\cdots&a_{0,m}\\ \vdots&\vdots&\ddots&\vdots\\ a_{r_n,1}&a_{r_n,2}&\cdots&a_{r_n,m}\end{bmatrix},$$

where $\frac{1}{r_n}$ is the coding rate of each SHSR, and $\sum_{n=0}^{N_p-1}\frac{1}{r_n}=N_c$. Then each $s_{i,n}$ in $S_{e,i}$ is a mapping from $\mathbb{Z}_2^{1\times 1}$ to $\mathbb{Z}_2^{r_n\times 1}$. The decryption scheme is just an Maximum Likelihood Decoder (MLD).

III. DESCRIPTIONS OF MAIN FUNCTIONS

Each block in Fig. 1 is addressed in this section. The processing flow of this scheme can be divided into two parts: key processing and data processing. In the key processing part, only the KE function is involved. In data processing, PRVG generates the pseudorandom vectors or keystream based on the initial conditions given by k_e, and SHSRs substitute plaintexts.

1. Pseudorandom Vector Generator

The task in PRVG is to produce a sequence of independent and identically distributed random vectors. In our approach, the matrix method (Niederreiter, 1992) is adopted to produce pseudorandom vectors. One important duty in PRVG is to vary the state transition of SHSRs that will make it hard for intruders to attack the system by predicting the state trajectory.

PRVG in our proposed system is depicted in the following.

$$\boldsymbol{X_{n+1}}=(\boldsymbol{G}\cdot\boldsymbol{X_n}+\boldsymbol{U_j})\ mod\ M, \quad (6)$$

where $\boldsymbol{G}\in\mathbb{Z}_2^{N\times N}$, $\boldsymbol{X}\in\mathbb{Z}_M^{N\times 1}$,

$\boldsymbol{U}\in\mathbb{Z}_M^{N\times 1}$, and M is a prime

This system may have the maximum period M^N-1 for some $\boldsymbol{G}$ and M (Stinson, 1995). In order to get a longer period than M^N-1, a simple method is to add a control term which starts off once the period of the system (6) is detected. Because (6) has the maximum period with some $\boldsymbol{G}$ and M, each vector $\boldsymbol{X_n}\in\mathbb{Z}_M^{N\times 1}$ is a periodic point. The dynamics of the system (6) can be changed by altering the control term $\boldsymbol{U_j}=[u_{0,j},\ u_{1,j},\ u_{2,j},\ u_{3,j}]^T$, where j indexes how many times the period occurs.

Because the PRVG sequence generated by (6) is a uniform distribution vector and has the maximum length N^M-1, each $\boldsymbol{X_n}\in\mathbb{Z}_M^{N\times 1}$ is periodic. Herewith the period can be checked by discovering the repetition of $\boldsymbol{X_0}$. When the period is detected, the control term $\boldsymbol{U_j}=[u_{0,j}\ u_{1,j}\ u_{2,j}\ u_{3,j}]^T$ ($\boldsymbol{U_0}$ is given by k_e) is computed below:

$$\boldsymbol{U_{j+1}}=\begin{bmatrix}2&0&0&0\\0&4&0&0\\0&0&2&0\\0&0&0&4\end{bmatrix}\otimes_M U_j \quad (7)$$

where $\otimes_M$ is modulo-M multiplication. A period-check mechanism can dramatically increase the period. Since the modulus M is prime, the set $\{u_{i,0},\ u_{i,1},$

…, $u_{i,M-1}$} is a multiplicative group. With altering the control term in (6), the period will grow into $(N^M-1)^2$. The functions controlled by PRVG are itemized as follows: (1) State transition or substitution functions ($S_{e,i}$ and $S_{d,i}$). (2) Random permutations ($\boldsymbol{R_{1,i}}$ and $\boldsymbol{R_{2,i}}$). (3) Exclusive-or pair {$\boldsymbol{X_p}$, $\boldsymbol{X_c}$}.

2. Expansion Function

The expansion function is used to yield not only k_e but also $\boldsymbol{R_{1,i}} \in Z_2^{N_p \times 1}$ and $\boldsymbol{R_{2,i}} \in Z_2^{N_c \times 1}$. If the bit length of PRVG's output is less than N_{RP} bits required to set the permutation, then $\boldsymbol{R_{1,i}}$ and $\boldsymbol{R_{2,i}}$ are created through expansion function with pseudorandom vectors as inputs.

Assume the input of bit length l_i and output of bit length l_o. The expansion function can be implemented by the following steps.

Step 1: Segment the input into $v=\left\lceil \frac{l_i}{8} \right\rceil$ blocks of which block size is 8 bits. If l_i is not an 8-multiple number, then 0s are attached to the LSB of the input.

Step 2: The v blocks, $\boldsymbol{\Lambda}=\{\lambda_0, \lambda_1, \cdots, \lambda_{v-1}\}$, can be at most grouped into 8 subsets $\boldsymbol{\Lambda}_n=\{\lambda_\mu | \mu \equiv n \bmod 8\}$, where n is an element of the set {0, 1, …, 7}.

Step 3: Other v blocks, $\{\tilde{\lambda}_0, \tilde{\lambda}_1, \cdots, \tilde{\lambda}_{v-1}\}$, can be obtained by circularly left shifting by n of each entry in set $\boldsymbol{\Lambda}_n$.

Step 4: Extend $8v$ bits obtained in Step 3 to l_o bits by appending 0s as the LSB of the new block set $\tilde{\boldsymbol{\Lambda}}$.

Step 5: Set TEMP=$\tilde{\boldsymbol{\Lambda}}$. The output is obtained as below:

```
for μ=1 : (l_o−8v)
   Λ̃=Λ̃<<μ
; circular left shift of Λ̃ by v.
   TEMP = TEMP⊕Λ̃
end
output =TEMP
```

If l_i and l_o are smaller than 8, then the block size can be reduced as 4 bits. The illustrative representation of this algorithm is shown in Fig. 2.

Key expansion (KE) in Fig. 1 expands the original key to meet the requirement of the initialization process. Assume that the key length of k and k_e are N_k and N_{k_e}, respectively, and $N_k+N_p+N_c=N_{k_e}$. KE is an expansion function with $l_i=N_k$, $l_o=N_{k_e}$, and the private key k as input.

3. Random Permutation

The permutation maps the input $x=[x_0 \ x_1 \ \cdots \ x_{n-1}]$ into the output $y=[y_0 \ y_1 \ \cdots \ y_{n-1}]$, and the mapping is determined by the $\boldsymbol{R_{1,i}}$ and $\boldsymbol{R_{2,i}}$ of bit length

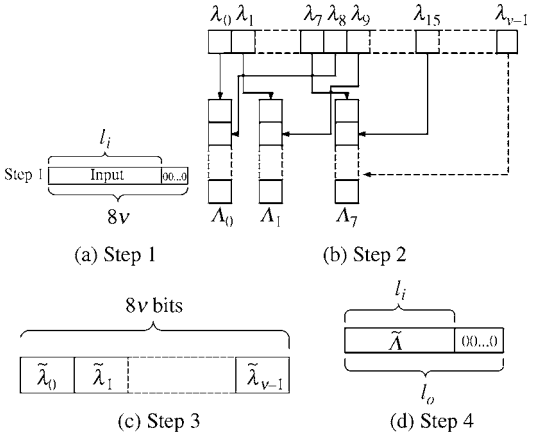


Fig. 2 The illustrative representation of Expansion Algorithm

N_{RP}. For a permutation box which is obtained by given $\lceil \log_2 n \rceil$ bits as position indexes, each input bit possibly appears on each output bit. Assume v_l is the decimal presentation of the lth index, then the output y_l is $y_l=x_{v_l} \bmod n$. Hence, for an n-bit random permutation function, the total bits to define the mapping are $n\lceil \log_2 n \rceil$.

N_{RP} required by $\boldsymbol{R_{1,i}}$ and $\boldsymbol{R_{2,i}}$ is calculated by

$$N_{RP}=N_P \cdot \lfloor \log_2 N_P \rfloor + N_c \cdot \lfloor \log_2 N_c \rfloor \quad (8)$$

Because N_{RP} bits needed by RP are probably larger than the bits that PRVG can provide, the expansion function with output bits of PRVG as input and $l_o=N_{RP}$ in the previous subsection can solve this problem.

4. State-Hopping Shift Register

These shift registers are used as substitution functions. All shift registers are formed by distinct primitive polynomials over GF(2) with degree m. There is a new concept introduced into the shift-register structure. In Fig. 3, the state is changed by not only shifting the content in registers but also the value of $\boldsymbol{V_i}$. Besides the plaintext $\boldsymbol{P_i}=[p_{i,0}, p_{i,1}, \cdots, p_{i,N_{p-1}}]$, $\boldsymbol{V_i}=[v_{i,0}, v_{i,1}, \cdots, v_{i,N_{p-1}}]$ and $pos_i \in ZZ_m^{1\times 1}$ are the inputs of SHSRs, where $p_{i,n}, v_{i,n} \in ZZ_2^{1\times 1}$.

Given the nth SHSR with the primitive polynomial $p(x)=1+a_1x+\cdots+a_{m-1}x^{m-1}+x^m$, then the output can be obtained by (5). The content of the SHSR, $[b_{n,1} \ b_{n,2} \ \cdots \ b_{n,m}]$, is changed by $p_{i,n}$ and $v_{i,n}$ sequentially. After $b_{1,n}=p_{i,n}$, the value $v_{i,n}$ is assigned to SHSR by the rule $b_{pos_i}=v_{i,n}$. It can be adumbrated that the transition of state diagram is determined by V_i and pos_i obtained from PRVG.

IV. THE DECRYPTION SCHEME

The structure of the decryption scheme is

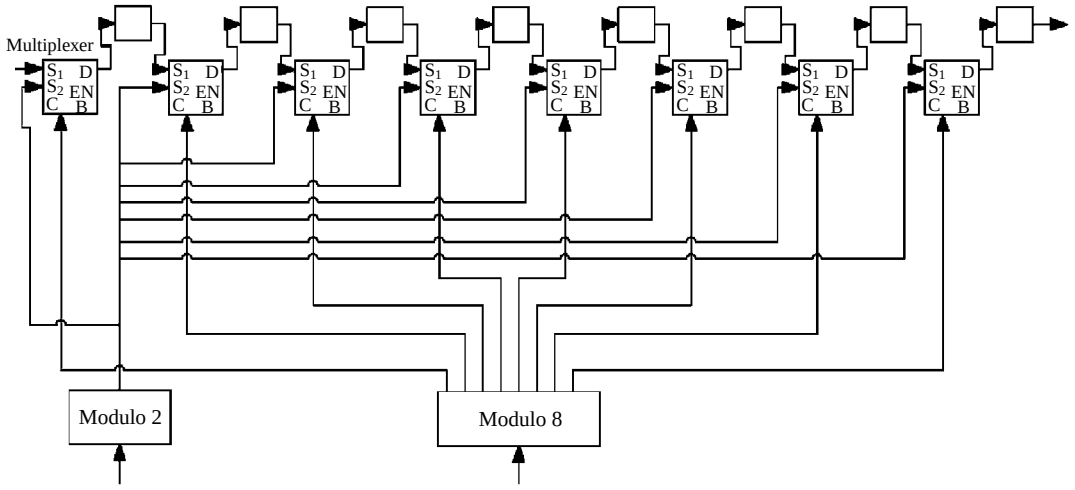


Fig. 3 The shift register with arbitrary bit assignment

similar to the encryption scheme in Fig. 1, except for the design of the SHSRs block. This block can be designed with or without channel coding. Without error correction ability, i.e. $N_p=N_c$, the decoder is similar to the encoder.

When the channel coding is enabled, i.e., $N_p<N_c$, the decoding algorithm of convolution code must be supposed to adapt. The Viterbi algorithm and sequential decoding (Lin, 1983) are two frequently used techniques. Due to the intrinsic characteristic of SHSRs, the Viterbi algorithm is hard to modify for decoding. But the sequential algorithm is suitable for SHSR-based stream ciphers. Because the sequential algorithm decoding is based on the code tree, there will be no problems in decoding by sequential algorithm. An SHSR is a nonlinear error coding, hence there must be a corresponding decoder for each SHSR.

Sequential Decoding with State Hopping

There are several algorithms of sequential decoding, e.g., stack algorithm, Fano algorithm, generalized stack algorithm, and multiple stack algorithm (Lin, 1983). For simplicity, the stack algorithm (Lin, 1983) is adopted.

Step 1: Load the stack with the origin node in the tree, whose metric is taken to be zero.
Step 2: Compute the metric of the successors of the top path in the stack.
Step 3: Delete the top path from the stack.
Step 4: Insert the new paths in the stack, and rearrange the stack in the order of decreasing metric values.
Step 5: If the top path in the stack ends at a terminal node in the tree, stop. Otherwise , return to Step 2.

A revised state algorithm is proposed here. At Step 2, the two metrics are computed by adding the previous metric stored in the stack and the current metric obtained by comparing the outputs of shift registers and received signals. In general, the outputs of shift registers are dependent on the current state, but not for the proposed system. In convolution code, there is only one state diagram, i.e., the state is changed only by the inputs of the shift registers. So the state can be obtained from the decoding path in the stack when computing the metric in Step 2.

One bit of information is still lacking in the SHS algorithm. Besides the inputs of SHSRs, the determination of the next state must have knowledge about the two values, pos_i and V_i in Fig. 1. Hence the history of pos_i and V_i have to be recorded.

V. SIMULATION

The experimental results of a (4,8,8,977) system are illustrated in this section. Following the design described earlier, we can obtain that the private key length N_k is 96 bits, the expanded key length N_{k_e} is 108 bits, the dimension N of PRVG is 4, and N_{RP} is 7×2+7×3=35. The polynomial matrixes of SHSRs are selected as $\boldsymbol{A_0}$={0*x*1*CF*, 0*x*1*F*5}, $\boldsymbol{A_1}$={0*x*18*D*, 0*x*14*D*}, $\boldsymbol{A_2}$={0*x*12*D*, 0*x*1*C*3}, and $\boldsymbol{A_3}$={0*x*1*E*7, 0*x*11*D*}. Because of N_p=4 and N_c=8 in the system, the secrecychannel coding scheme is selected. Each subsystem $s_{e,i}$ is an error control code with coding rate 1/2. Note that the code rate is not obtained directly from the division $\frac{N_p}{N_c}$.

Security Analysis

The secrecy-channel coding has an inborn

Table 1 Probability distribution in 4 tests with zero-input, one-input, different keys and different data

Outcome	Test 1	Test 2	Test 3	Test 4
0	0.49362	0.50325	0.49363	0.50054
1	0.50638	0.49676	0.50637	0.49946

advantage in security. The error correction is impossible for lack of knowledge about the private key, hence the channel noise will create a more secure channel than the one provided by general systems which do secrecy coding and channel coding in two steps. Noises are removable for a cryptanalyst in a conventional system, since the scheme of the channel decoder is known. Certainly, the security of the secrecy-channel system can not rely on the channel noise.

In Table 1, the probabilities of 0 and 1 are addressed for the four cases. The input in test 1 is a zero vector with length 10^5 bits, and is a 10^5-bit vector of 1 in test 2. There are 10^3 patterns in test 3, each pattern is a 10^3-bit vector with Hamming distance 1. In test 4, the 10^3 zero bits are encrypted in 96 distinct keys $\{k_0, k_1, \cdots, k_{95}\}$, the Hamming distance between k_0 and k_j is 1, where $1 \le j \le 95$.

According to the numerical data shown in Table 1, the probability approximates 0.5 for each case. For security or for randomness, this is a good phenomenon. It is not helpful for error control coding, since the error control ability is determined by the shortest Hamming weight of the designed code. Table 1 also shows that no matter what the distance is calculated 0 or 1, the Hamming distance is not long enough for constituting a good code for error-control coding.

We also use NIST's statistical test suite (Rukhin *et al*., 2000) to verify the randomness of our system. We generate 40 sequences of 10^5 bits and run the 11 tests, frequency test, block-frequency test, cumulative sums test, runs test, long-run test, rank test, discrete fourier transform test, non-overlapping template matching test, serial test, Lempel-Ziv test, and linear complexity test. The results are shown in Table 2. The table has 3 columns: column 1 is the name of test, column 2 is the P-value that arises via the application of chi-square test, column 3 is the ratio of sequences that passed the test. In Table 3 rows 1-11 are the distributions of P-values of the given 40 sequences, where C1 to C10 separately correspond to 10 equal bins obtained by dividing a unit interval. Each row in Table 2 and 3 is a single test. The test program transformed each result into an identical index named P-value. High P-value means that the sequence provides high randomness. In general, if the P-value is greater than 0.01, we can conclude that the sequence is random. As Table 2 shows, our system has high P-values, above 0.5 in most tests, and high a passing ratio. From the testing results, we assert the randomness of our system.

Table 2 The experimental results of our system obtained by NIST's statistical test suite

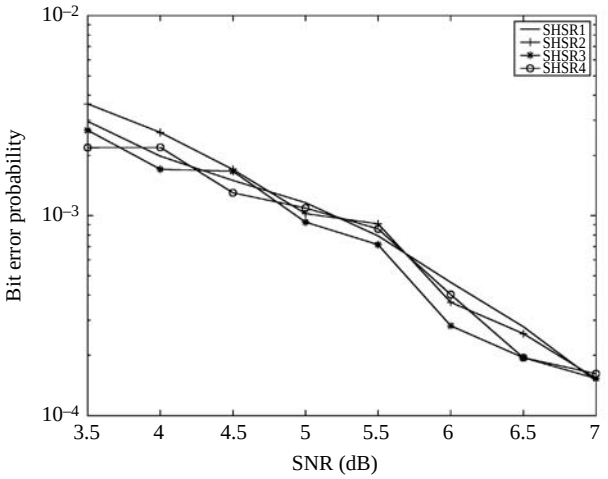
Statistical Test	P-Value	Ratio
1. Frequency	0.834308	0.9750
2. Block-Frequency	0.834308	1.0000
3. Cusum	0.350485	0.9750
4. Runs	0.739918	1.0000
5. Long-Run	0.637119	1.0000
6. Rank	0.437274	1.0000
7. FFT	0.484646	1.0000
8. Aperiodic	0.999438	1.0000
9. Serial	0.834308	0.9750
10. Lempel-Ziv	0.242986	0.9750
11. L. Complexity	0.834308	1.0000

Table 3 The experimental results of our system obtained by NIST's statistical test suite

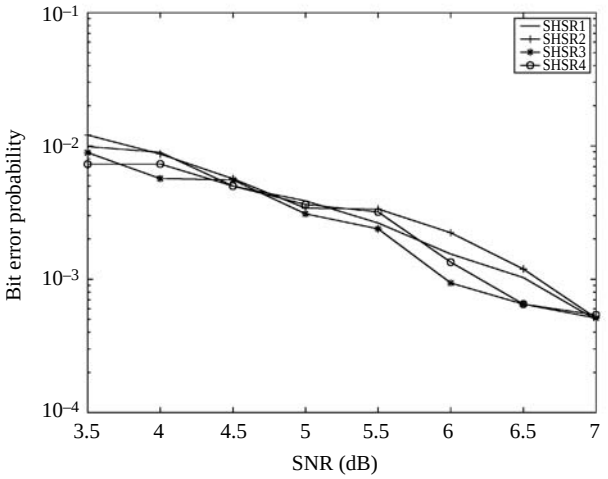
Test	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Test 1	6	3	5	3	4	2	5	2	6	4
Test 2	4	4	6	4	3	4	5	3	1	5
Test 3	5	6	7	2	5	1	5	5	1	3
Test 4	2	6	5	4	3	4	4	2	3	7
Test 5	3	2	3	5	2	6	2	6	6	5
Test 6	2	5	5	8	6	3	4	2	3	2
Test 7	1	1	4	5	3	4	4	5	6	7
Test 8	5	4	4	5	3	4	4	3	4	4
Test 9	6	3	5	3	4	2	5	2	6	4
Test 10	9	3	4	4	1	3	3	2	5	6
Test 11	3	3	3	3	5	7	2	5	5	4

Performance of Error Correction Ability

Figure 4 is the error probability of this code over an AWGN channel. The system parameters are the same in both cases, the only difference is the changeability of the 1st register in SHSRs. From the simulation result, it's obvious that, under the channel-coding sense, the coding performance in case 1, in which the 1st register is unchangeable, is better than the one in case 2, in which the 1st register is changeable. Comparing pure channel coding, our system has normal performance at low SNR, but lower performance at high SNR. This is caused by the noise-like state transition of SHSRs.



(a) Case 1: The 1st register of SHSRs is unchangeable



(b) Case 2: The 1st register of SHSRs is changeable

Fig. 4 The error probability in a system (4,8,8,977)

VI. CONCLUSIONS

The SECC and JEEC schemes presented by Rao *et al*. are block coding systems, but our proposed scheme, a new secrecy-channel scheme, is a stream coding system. Our architecture can be either a pure cryptosystem or a secrecy-channel coding system. The combination of secrecy-channel coding reduces the computation time and enjoys an extra benefit that the channel error will make it hard for intruders to attack. The proposed scheme is also flexible for design. Given a 4-tuple (N_p, N_c, m, M) for an application, following the design flow will get an appropriate system. The plaintexts/ciphertexts can be fast encrypted/decrypted by a pure cryptosystem and our system has strong statistical security. The error control ability depends on the polynomial of SHSRs, the output of PRVG and the decoding scheme. Comparing pure channel coding, our system has normal performance at low SNR, but lower performance at high SNR.

ACKNOWLEDGMENTS

This work was supported by National Science Council under Grant No. NCS90-2213-E-009-058.

REFERENCES

Bystrom, M., and Modestino, J. W., 2000, "Combined Sourcechannel Coding Schemes for Video Transmission over an Additive White Gaussian Noise Channel," *IEEE Journal on Selected Areas in Communcations*, Vol 18, No. 6, pp. 880-890.

Cheng, H., and Li, X., 2000, "Partial Encryption of Compressed Images and Videos," *IEEE Transactions on Signal Processing*, Vol. 48, No. 8, pp. 2439-2451.

Denis, A., and Kinsner, W., 1999, "Secure and Resilient Data Printed on Paper," *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, Shaw Conference Center, Edmonton, Alberta, Canada, Vol. 1, pp. 245-248.

Ho, K. P., and Kahn, J. M., 1996, "Transmission of Analog Signals Using Multicarrier Modulation: a Combined Source-Channel Coding Approach," *IEEE Transactions on Communications*, Vol. 44, No. 11, pp. 1432-1443.

Hwang, T., and Rao, T. R. N., 1990, "Secret Error-Correcting Codes (SECC)," *Advances in Cryptology-Crypto* '88, pp. 540-563, Springer-Verlag, New York, USA.

Lin, S., 1983, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, NJ, USA.

McEliece, R. J., 1978, "A Public Key Cryptosystem Based on Algebraic Coding Theory," *DSN Progress Report* 42-44, pp. 114-116, Jet Propulsion Laboratory, CA, USA.

Meier, W., and Staffebach, O., 1989, "Fast Correlation Attacks on Certain Stream Ciphers," *Journal of Cryptology*, Vol. 1, No.3, pp. 159-167.

Modestino, J. W., Daut, D. G., and Vickers, A. L., 1981, "Combined Source-channel Coding of Images Using the Block Cosine Transform," *IEEE Transactions on Communications*, Vol. IT-33, No. 9, pp. 827-837.

Niederreiter, H., 1992, *Random Number Generation and Quasi-Monte Carlo Methods*, Society of Industrial and Applied Mathematics, PA, USA.

Rao, T. R. N., and Nam, K., 1987, "Private-Key Algebraic Cryptosystems," *Advances in Cryptology-Crypto* '86, pp. 35-48, Springer-Verlag, London, UK.

Rao, T. R. N., and Nam, K., 1989, "Private-Key Algebraic-code Encryptions," *IEEE Transactions*

on Information Theory, Vol. IT-35, No. 4, pp. 829-833.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J.,and Vo, S., 2001, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *National Institute of Standards and Technology publication 800-22*.

Siegenthaler, T., 1985, "Decrypting a Class of Stream Ciphers Using Ciphertext Only," *IEEE Transactions on Computing*, Vol. C-34, No. 1, pp. 81-85.

Stinson, D. R., 1995, *Cryptography: Theory and Practice*, CRC press, New York, USA.

Zeng K., Yang C. H., and Rao, T.R.N., 2001, "Cryptanalysis of the Hwang-Rao Secret Error-Correcting Code Schemes," *Lecture Notes in Computer Science*, Vol. 2229, pp. 419-428.

Zhang, M., 2000, "Maximum Correlation Analysis of Nonlinear Combining Functions in Stream Ciphers," *Journal of Cryptology*, Vol. 13, No.3, pp. 301-314.

Manuscript Received: Mar. 14, 2003
Revision Received: May 10, 2004
and Accepted: Jun. 09, 2004