ELSEVIER

Signal Processing 82 (2002) 259–265

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Parallel stream cipher for secure high-speed communications

Hoonjae Lee[a,*], Sangjae Moon[b]

[a] *Department of Computer Engineering, Kyungwoon University, Induk-ri 55, Sandong-Myun, Kumi, Kyungbuk 730-850, South Korea*

[b] *School of Electronic and Electrical Engineering, Kyungpook National University, 1370, Sankyuk-Dong, Taegu 702-701, South Korea*

Received 28 April 2000; received in revised form 9 July 2001; accepted 7 November 2001

Abstract

Due to ongoing improvements in high-speed communications, the speed of data encryption must also increase. Accordingly, this paper proposes an PS-LFSR with an $m(\geqslant 2)$-times faster shifting during one clock interval and a parallel stream cipher that is faster by paralleling many similar keystream generators using the PS-LFSRs. Finally, an m-parallel SUM-BSG with 8-parallel for detail is proposed as a design example of the proposed parallel stream cipher. When compared with a conventional stream cipher, the properties of the proposed cipher exhibited the same crypto-degree with m-times faster processing. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Parallel stream cipher; Period; Linear complexity; Randomness; Summation generator

1. Introduction

Due to ongoing improvements in high-speed communications, the speed of data encryption must also increase. Cryptography is the only known practical method for protecting information transmitted through communication networks that use land lines, communication satellites, and microwave facilities. Cryptographic methods can be divided into block ciphers, stream ciphers, and public-key cryptosystems [4,8]. There are four application modes of block ciphers: the ECB (electronic codebook) mode, CFB (cipher feedback) mode, CBC (cipher-block chaining) mode, and OFB (output feedback) mode [8]. The ECB mode outputs ciphertext blocks from plaintext blocks via a complex transformation controlled by a secret key. The CFB mode autonomously establishes communication synchronization using feedback from the ciphertext to the input block. The CBC mode is useful for a general-purpose block-oriented transmission or for authentication with block-chaining. The OFB mode is similar to a stream cipher, in which a block cipher generates random sequence blocks from an initial value block [8]. However, all four modes have weaknesses in their application to an erroneous channel as in, for example, a wireless channel. In an erroneous channel, a one-bit error in a ciphertext will propagate to many blocks of recovered plaintext in the receiver. In the ECB mode, a one-bit channel error in a ciphertext will propagate to the full range of the recovered plaintext block in the receiver. Accordingly, a channel with a 10^{-6} BER (bit error rate) will be degraded to a channel with a 10^{-4} ($\approx 128 \times 10^{-6}$) BER if a block cipher with a 128-bit block size is applied. In terms of error propagation, cases using the CFB and CBC modes will be more seriously affected than those

* Corresponding author. Tel.: +82-546-479-1222; fax: +82-546-479-1029.

E-mail address: hjlee@kyungwoon.ac.kr (H. Lee).

PII: S0165-1684(01)00180-3

using the ECB mode. In contrast, the OFB mode offers a unique solution to the block cipher problem, however, it needs a faster encryption speed. For example, a DES [7,8] with 16 rounds will generally output 64 bits in 16 system-clock intervals, therefore, the concept of repetition (round) decreases the processing speed.

Public-key cryptosystems are not useful for data-encryption because of their slow processing rate and the problem of bit-error propagation as in the ECB mode. Stream ciphers exhibit good properties including no error propagation, security levels properly selectable according to certain security criteria, and a higher processing ability than block ciphers, however, new high-speed communication systems are requiring faster data encryption.

This paper focuses on the following three problems in designing a cryptosystem: security, fast enciphering/deciphering, and error propagation persistence in channels including mobile communication. As a result, a parallel stream cipher is proposed that combines the strengths of stream and block ciphers, that is, the security and freedom from error propagation of a stream cipher and the parallel processing ability of a block cipher. Normally, all LFSRs in a stream cipher shift/output 1-bit for one clock-time interval, whereas, in the proposed cipher the LFSRs are elevated to a high-speed type, PS-LFSRs, which shifts/output $m(\geqslant 2)$-bits for one clock interval. Plus, as an improved version of the (single) nonlinear combine function, an m-parallel nonlinear combine function (general type) is introduced, which generates m-bit keystream sequences for the proposed parallel stream cipher. Finally, an m-parallel SUM-BSG is presented as a design example, arranged with many Rueppel's summation generators [1] in parallel and $m=8$ for details. Its performance is analyzed in terms of cryptographic security and the processing speed compared with a conventional stream cipher.

2. Parallel stream cipher

2.1. General requirements of a stream cipher

The following requirements are assumed necessary for cryptosystems [10]:

(1) *Error propagation*: The error propagation due to encryption/decryption should be minimal.
(2) *Redundant information*: The insertion of redundant information bits should be minimal.
(3) *Cryptographic security*: The number of secret keys should be large enough so an exhaustive key search attack is impossible.
(4) *Simplicity of implementation*: The encryption/decryption system should be realizable with software or hardware.
(5) *Performance speed*: The encryption/decryption should be performable at speeds ranging from T1 rate (1.544 Mbps) up to many Gbps.

For a secure stream cipher, the keystream should be unpredictable and subsequent keystreams should not be able to be anticipated from previous ones. The following are necessary conditions for the unpredictability of a keystream [7,10]:

(1) *Long period*: A keystream should have a long period.
(2) *Large linear complexity*: Large linear complexity implies that it is impractical (infeasible) to use the equivalent LFSR to predict the keystream output sequences.
(3) *Randomness*: A large linear complexity does not imply randomness. The statistical property of the keystream should be the same as an ideal random source.
(4) *Proper order of correlation immunity*: A nonlinear combining function F is called a kth order correlation immune when any $k(\leqslant N)$ combinations $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ $(1 \leqslant i_1, i_2, \ldots, i_k \leqslant N)$ of all N-input-bits $x_1, x_2, \ldots, x_N$, on function F are uncorrelated with the output of F.

2.2. Proposed PS-LFSR

Parallel-structured/-shifting LFSRs (PS-LFSRs) for use as the basic element of the parallel stream cipher are proposed as shown in Fig. 1. An PS-LFSR can answer the question 'how can an LFSR be shifted by m-bits within one clock interval?' For a parallel structure, there are an n-stage PS-LFSR on the right in Fig. 1(b) and an $(m-1)$-stage LBUF which stores temporally a lot bits of the shifted-out on the left in the figure. Each m-bit block of the n-stage PS-LFSR shifts left by system clock and the m feedback paths are independently XORed based on each combination of the feedback taps, thereafter the results can be simultaneously shifted to the rightmost of the LFSR.

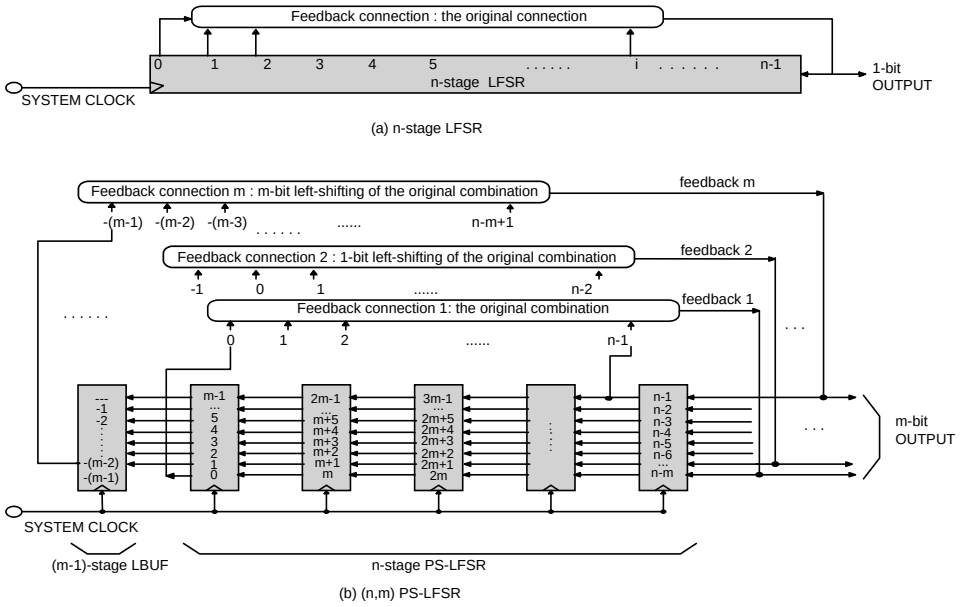

Fig. 1. LFSR and proposed PS-LFSR.

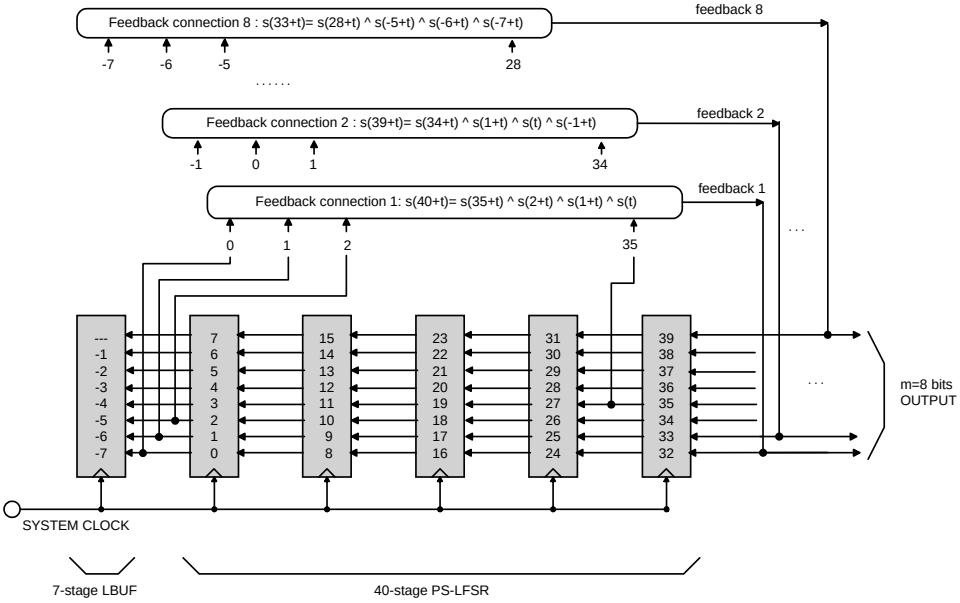

Fig. 2. ($n = 40,\ m = 8$) PS-LFSR as an example.

In this case, the first path (feedback 1 in the figure) is computed by using the original feedback connection function (from primitive polynomial), the second path (feedback 2) by using the 1-bit left shifted function of the original combination, and the third path (feedback 3) by using the 1-bit left shifted of the second combination, and so on. We depict an example of a 40-stage, 8-parallel PS-LFSR in Fig. 2.

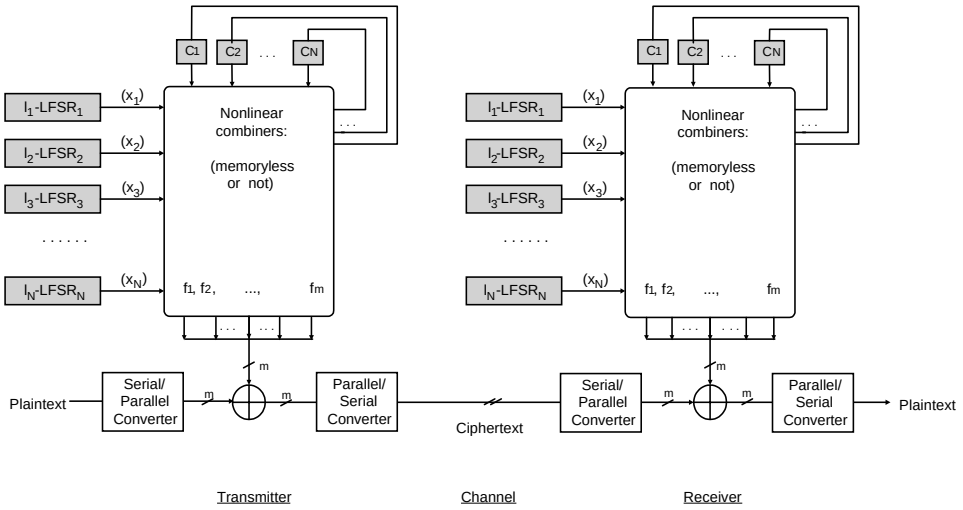

Fig. 3. Proposed parallel stream cipher.

As a result, the processing speed of the PS-LFSR is m-times faster than that of a normal LFSR where the clock only shifts 1-bit left. Moreover, a PS-LFSR has the same cryptographic security in terms of randomness, period, and linear complexity as a conventional LFSR, because it simultaneously generates m-bit outputs while m-bit shifting and each output is only used once. In addition, recent VLSI technology facilitates the implementation of a PS-LFSR, in contrast to the increased complexity of the hardware implementation.

2.3. Proposed parallel stream cipher

Unlike a conventional keystream generator, the proposed parallel stream cipher generates $m(\leqslant N)$ independent sequences from nonlinear combine functions $(f_1, f_2, \ldots, f_m)$ via N LFSRs, as in Fig. 3,and each (m) sequence enciphers (XORs) from a plaintext block to a ciphertext block in parallel. This makes the proposed cipher m-times faster than a conventional stream cipher in spite of the increased complexity in the hardware implementation. The proposed cipher also retains the channel quality level in the BER using channel error propagation without additional equipment. If required, it can prevent a correlation attack with the use of a correlation immune function with memory bits [3,6,9,11].

In Fig. 4, m-parallel nonlinear combine functions $(f_1, f_2, \ldots, f_m)$ are proposed as a generalized model, which use m-bit memories $(c_1, c_2, \ldots, c_m)$ and PS-LFSRs (Fig. 1) in place of LFSRs. All the LFSRs must have different lengths and that are pair-wise co-prime: $\gcd(l_i, l_j) = 1$ for all $1 \leqslant i, j, (i \neq j) \leqslant N$.

Each function f_i $(i = 1, 2, \ldots, m)$ in an algebraic normal form (ANF) is defined as follows:

$$
\begin{aligned}
& f_i(x_{1i}, x_{2i}, \ldots, x_{Ni}, c_{i1}, c_{i2}, \ldots, c_{iM_i}) \\
& = a_{i0} + \left(\sum_{j=1}^{N} a_{ij} x_{ji} + \sum_{j=N+1}^{N+m} a'_{ij} c_{ij} \right) \\
& \quad + \left(\sum_{j,k} a_{ijk} x_{ji} x_{ki} + \sum_{j,k} a'_{ijk} c_{ij} c_{ik} + \sum_{j,k} a''_{ijk} x_{ji} c_{ik} \right) \\
& \quad + \cdots + a_{ijk..N+i} x_{ji} x_{ki} .. x_{ti} c_{i1} c_{i2} .. c_{iMi}, \qquad (1)
\end{aligned}
$$

where, x_{jk} is the kth output sequence of the parallel m-bit on LFSR_j, c_{jk} $(1 \leqslant j, k \leqslant m)$ is the kth memory sequence of the jth function, $a_{ij}, a'_{ij}, a_{ijk}, a'_{ijk}, a''_{ijk}, \ldots, a_{ijk..N+i} \in [0, 1]$ are all binary coefficients, and M_j is the number of memories used in the jth function f_j.

Each $f_i(x_{1i}, x_{2i}, \ldots, x_{Ni}, c_{i1}, c_{i2}, \ldots, c_{iM_i})$, $i = 1, 2, \ldots, m$, is required to fulfill the conditions in Section 2.1.

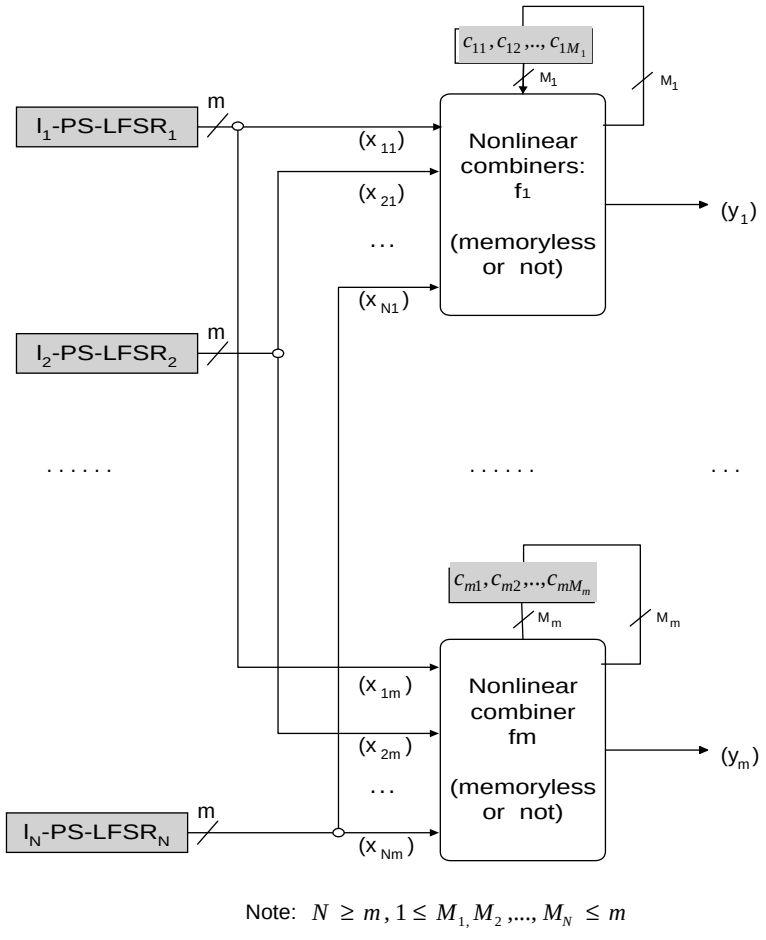

Fig. 4. Generalized m-parallel nonlinear combiners.

As an example, an m-parallel summation generator is proposed with memories (called "m-parallel SUM-BSG") in Fig. 5.In this figure, m number of SUM-BSGs [1,6] used is configured in parallel and all the LFSRs used are the same type of PS-LFSR. Each PS-LFSR (simply LFSR) generates m-output sequences: PS-LFSR$_1$ generates $(x_{11}, x_{12}, \ldots, x_{1m})$ sequences for a clock, PS-LFSR$_2$ generates $(x_{21}, x_{22}, \ldots, x_{2m})$ sequences, and the same method PS-LFSR$_m$ generates $(x_{m1}, x_{m2}, \ldots, x_{mm})$ sequences. For the next clock PS-LFSR$_1$ generates $(x_{1,m+1}, x_{1,m+2}, \ldots, x_{1,2m})$ sequences, and so on. Therefore, (x_{11}) sequences are as follows: $x_{11}, x_{1,m+1}, x_{1,2m+1}, x_{1,3m+1}, \ldots$.

The properties of the output sequence y_i of the ith SUM-BSG are as follows:

$$(y_i) = \{(x_{1i}) \oplus \cdots \oplus (x_{mi})\} \oplus \{(c_{i1}) \oplus \cdots \oplus (c_{iM})\}, \quad (2)$$

where (y_i) represents the ith output sequences of SUM-BSG$_i$ $(i = 1, 2, \ldots, m)$, (x_{1i}) the ith output sequences of LFSR$_1$, (x_{2i}) the ith output sequences of LFSR$_2$, (x_{mi}) the ith output sequences of LFSR$_m$, and (c_{ij}) the jth carry sequences of the ith function.

Property 1. If $\gcd(l_i, l_j) = 1$ $(1 \leqslant i, j \leqslant m, i \neq j)$, is relatively prime, and the all LFSRs used have a

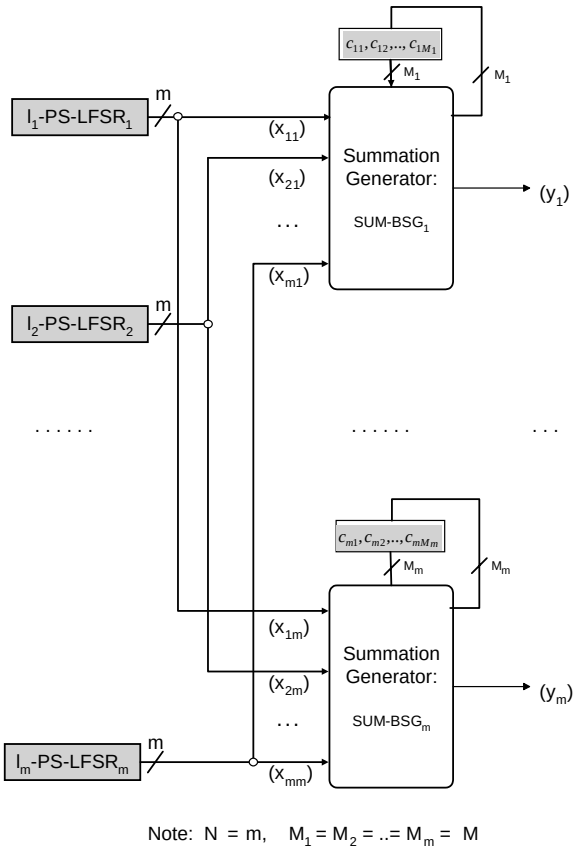

Fig. 5. Proposed m-parallel SUM-BSG.

non-null initial value, then each SUM-BSG$_i$ will have the following properties [1,7]:

(1) *Period*: $P_i = \prod_{j=1}^{m} (2^{l_j} - 1)$.
(2) *Randomness*: good.
(3) *Linear complexity*: $LC_i \approx P_i$.
(4) The order of the correlation immunity of the function $f_i : K_i = m - 1$.

An SUM-BSG includes a maximum period, good randomness, near maximal linear complexity, and maximum order of correlation immunity, as in Property 1.

An 8-parallel summation generator with a 3-bit carry (called "8-parallel SUM$_{11}$-BSG", with an 11-input in total) in detail is also proposed, which can operate a real-sum from an 11-input $(x_{1i}, x_{2i}, \ldots, x_{8i}, c_{i3}, c_{i2}, c_{i1})$ and then convert the decimal (summed) to binary.

The primitive polynomials in the proposed generator are generated by Ref. [5].

$$g_1(x) = x^{19} + x^9 + x^6 + x^3 + x^2 + x + 1,$$
$$g_2(x) = x^{23} + x^{12} + x^6 + x^3 + x^2 + x + 1,$$
$$g_3(x) = x^{29} + x^{11} + x^7 + x^3 + x^2 + x + 1,$$
$$g_4(x) = x^{31} + x^3 + 1,$$
$$g_5(x) = x^{37} + x^{18} + x^2 + x + 1,$$
$$g_6(x) = x^{41} + x^7 + x^4 + x^3 + x^2 + x + 1,$$
$$g_7(x) = x^{43} + x^{16} + x^4 + x^3 + x^2 + x + 1,$$
$$g_8(x) = x^{47} + x^{14} + x^4 + x^3 + x^2 + x + 1.$$

Property 2. If $\gcd(l_i, l_j) = 1$, $(1 \leqslant i, j \leqslant m,\ i \neq j)$ and all the LFSRs used have non-null initial values, then the ith SUM$_{11}$–BSG$_i$ of the proposed 8-parallel SUM$_{11}$-BSG will have the following properties:

(1) *Period*:

$$P_i = (2^{19} - 1)(2^{23} - 1)(2^{29} - 1)(2^{31} - 1) \times (2^{37} - 1)(2^{41} - 1)(2^{43} - 1)(2^{47} - 1) \approx 2^{270} \approx 10^{81}, \quad i = 1, 2, \ldots, 8.$$

(2) *Randomness*: good [6,7].
(3) *Linear complexity*: $LC_i \approx P_i,\ i = 1, 2, \ldots, 8$.
(4) The order of the correlation immunity of the function: $K_i = m - 1 = 7,\ i = 1, 2, \ldots, 8$.
(5) The ciphering speed is $m = 8$ times faster than that of the original application in a stream cipher.
(6) The complexity of the number of gates used in the hardware is approximately 2-times (upper-limited by m times) more complex than that of the conventional SUM$_{11}$-BSG (Refer to Table 1, similar concluded in [2].).

Since each SUM-BSG$_i$ function generates each output sequence using an independent method, the cryptographic properties of a single output sequence of the proposed generator are the same as those of a single SUM-BSG. Accordingly, the proposed generator guarantees a maximum period, near-maximum linear complexity, maximum order of correlation immunity, and randomness properties like the conventional generator. Therefore, the proposed parallel

Table 1
Comparison of similar generators

Items	SUM_{11}-BSG	8-parallel SUM_{11}-BSG ($M = 8$)
Period	10^{81}	10^{81}
Randomness	Random	Random
Linear complexity	Approximate to period	Approximate to period
Correlation immunity	7	7
Number of F/Fs used (F/F means flip/flop device)	270	398
Number of XOR gates used	42	336
Total number of gates used (if 1 F/F = 5 gates)	1392	2326 (1.67 times complex)
Processing rate ratio	1	8 ($M = 8$ times high)

generator is a secure high-performance generator with slightly more complex hardware.

3. Conclusion

This paper proposed a parallel stream cipher which combines the strengths of stream and block ciphers, that is, the security and freedom from propagation error of a stream cipher and the block or parallel processing ability of a block cipher. Generally, all LFSRs in a stream cipher shift/output 1-bit during one clock-time interval. This was improved with the use of parallel-structured type PS-LFSRs to m-bit shifting/outputting for one clock. In addition, m-parallel nonlinear combine functions (general type) were introduced that improve the nonlinear combine function, outputting 1-bit keystream sequences and generating m-bit keystream sequences for applying the proposed parallel stream cipher. Finally, an m-parallel SUM-BSG was presented as a design example, arranged with an m number of Rueppel's summation generators (SUM-BSGs) in parallel, and an 8-parallel SUM_{11}-BSG in detail. The design performance was analyzed in terms of cryptographic security and the processing speed compared with a conventional stream cipher. The results showed the same cryptographic security from the perspective of period, linear complexity, randomness, and correlation immunity, however, the performance was m-times faster.

References

[1] Hoonjae Lee, SangJae Moon, On an improved summation generator with 2-bit memory, Signal Processing 80 (1) (January 2000) 211–217.
[2] Hoon-jae Lee, Sang-jae Moon, On a high-speed implementation of LILI-128 stream cipher using FPGA/VHDL, Journal of Korea Inst. Inform. Security Cryptol. 11 (3) (June 2001) 23–32.
[3] W. Meier, O. Staffelbach, Correlation properties of combiners with memory in stream ciphers, J. Cryptol. 5 (1992) 67–86.
[4] A.J. Menezes, P.C. Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1997.
[5] B. Park, H. Choi, T. Chang, K. Kang, Period of sequences of primitive polynomials, Electron. Lett. 29 (4) (February 1993) 390–391.
[6] R.A. Rueppel, Correlation immunity and the summation generator, Advances in cryptology, Proceedings of CRYPTO'85, Santa Barbara, Aug. 18–22, 1985, pp. 260–272.
[7] R.A. Rueppel, Analysis and Design of Stream Ciphers, Springer, Berlin, 1986.
[8] B. Schneier, Applied Cryptography, 2nd Edition, Wiley, New York, 1996.
[9] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, IEEE Trans. Inform. Theory IT-30 (5) (September 1984) 776–780.
[10] M. Tatebayashi, N. Matsuzaki, D.B. Newman, A cryptosystem using digital signal processors for mobile communication, ICASSP'90, 1990, pp. 37.1.1–37.1.4.
[11] G.Z. Xiao, J.L. Massey, A spectral characterization of correlation—immune combining functions, IEEE Trans. Inform. Theory 34 (3) (May 1988) 569–571.