고속 동기식 스트림 암호에서의 ZS 동기 방식 개선

이 훈 재†

요 약

동기식 스트림 암호에 적용하기 위한 여러 가지 Zero suppression(ZS) 알고리즘 중에서 ZS-2 알고리즘은 블록동기기능 제거, 구현 용이성 등 여러 가지 좋은 특성을 보여주고 있다. 하지만, 이 방법은 채널 오류 확산 측면에서 취약점을 보이고 있다. 따라서, 본 논문에서는 열악한 잡음 채널에서 오류 확률에 따른 성능을 개선하기 위하여 ZS-2에서 실행했던 대체 블록에서의 대체 비트 수를 최소화시키는 새로운 방법을 제안하였다. 결과적으로, 제안된 ZS-3 기법은 $n=8$에서 평균 오류 확산을 ZS-2의 값 보다 18.7% 떨어뜨리는 좋은 특성을 나타냄을 확인하였다.

An Improved ZS Algorithm for High-Speed Synchronous Stream Ciphers

Hoon-Jae Lee†

ABSTRACT

Among the various zero suppression (ZS) algorithms used in a for synchronous stream cipher system, a ZS-2 exhibits certain good properties, including the omission of the block synchronization, easy implementation, etc., yet also a weakness in channel error propagation. Accordingly, This paper proposes a new method by minimizing the bit-wide substitution in the substitution blocks of ZS-2 to improve the degenerated error property in a noisy channel. As a result, the proposed ZS-3 algorithm can decrease the mean error propagation by about 18.7% over that of ZS-2 at $n=8$.

키워드 : Zero Suppression, 동기(Synchronization), 스트림 암호(Stream Cipher), 랜덤 수열(Random Sequence), 키수열(Keystream)

1. Introduction

In a synchronous stream cipher [1-3], ciphertext bits are randomly distributed because they are the exclusive-ored value of a '0'-'1' balanced keystream sequence and plaintext. In this case, long consecutive '0' sequences can occur, which creates a number of communication problems [4-5]. As a solution, several zero suppression algorithms [4-5] have been proposed, which first suppress k or more zeros between successive ones in the ciphertext at the sender end and then completely recover the original message at the receiver end of a synchronous stream cipher system.

ZS algorithms can be divided into serial detection algorithms and block detection algorithms. The former detects an all-zero input block (00 ⋯ 0) using the interval of a bit serial slide relative to the block size containing previous bits in the buffer, whereas the latter detects an all-zero input block by checking on the block interval relative to the block size. As such, ZS-1 [4] uses block detection and ZS-2 uses serial detection.

Among zero suppression (ZS) algorithms for a synchronous stream cipher system, ZS-2 has various good properties, including the omission of block synchronization, easy implementation, etc., yet a weakness in channel error propagation. Accordingly, this paper proposes a new method by minimizing the bit-wide substitution in the substitution blocks of ZS-2 to improve the degenerated error property in a noisy channel. As a result, a ZS-3 algorithm is proposed which can decrease the mean error propagation to that of ZS-2 at $n=8$. Consequently, when the proposed ZS algorithm is applied to a special channel (for example, a T1-carrier system [6]), the system performances for clock recovery are improved, and deterioration in cryptographic security from frequent resynchronizations by Daeman[7] is prevented.

2. Zero Suppression Algorithms

A ZS algorithm consists of a detection part, which detects

† 정 회 원 : 동서대학교 인터넷공학부 정보네트워크공학전공 교수
논문접수 : 2001년 9월 12일, 심사완료 : 2002년 4월 2일

n-bit zeros (blocks of n consecutive zeros), and a substitution part, which then substitutes n-bit zeros with a non-zero block. Detection methods can be divided into serial detection, which detects an all-zero input block (00 ⋯ 0) by checking on the bit-by-bit serial interval relative to the block size containing previous bits in the buffer, and block detection, which detects an all-zero input block by checking on the block interval relative to the block size. Substitution methods can be divided into current block substitution and mixing block substitution, which substitute three or more blocks (preceding blocks, present block, and next blocks) with non-zero random blocks.

ZS-1 is efficient for block processing systems yet not so useful in a stream cipher system, because a stream cipher works on the bit-operation principle (exclusive-ored), whereas ZS-1 is a block operation in a block detection method. For bit-serial detection the block variables are defined to increase the bit-wise as follows :

Let :

- $\mathbf{P}_i$ ($p_i, p_{i-1}, \cdots, p_{i-n+1}$) be the ith n-bit plaintext vector block
- $\mathbf{K}_i$ ($k_i, k_{i-1}, \cdots, k_{i-n+1}$) the ith n-bit keystream block
- $\mathbf{C}_i$ ($c_i, c_{i-1}, \cdots, c_{i-n+1}$) the ith n-bit ciphertext block
- $\mathbf{Q}_i$ ($q_i, q_{i-1}, \cdots, q_{i-n+1}$) the ith n-bit recovered plaintext block at the receiver
- $\mathbf{0}$ (0, 0, ⋯, 0) the n-bit 0 vector ($i > 2n$).

The block size is

$$n = \left\lceil \frac{k+1}{2} \right\rceil$$

where $\lceil x \rceil$ denotes the maximum integer which is not over x.

ZS-2 can be described based on the following assumptions and algorithm.

2.1 Assumptions

1. A redundant bit insertion or deletion at the sender is not permitted. (This is because it is difficult to control the clock rate in an intermediated synchronous stream cipher system.)
2. It is not permitted to present k-bit consecutive zeros in plaintext (checking in serial time, not block intervals).
3. Cryptographically strong keystream sequences must be used in the system.

2.2 ZS-2 Algorithm

Sender :

1. Put plaintext bit p_i and ciphertext bit $p_i \oplus k_i$ in two n-stage shift registers respectively, and shift four registers ($\mathbf{P}_i$, $\mathbf{P}_{i-n}$, $\mathbf{P}_i \oplus \mathbf{K}_i$, $\mathbf{P}_{i-n} \oplus \mathbf{K}_{i-n}$).
2. Check two n-bit vector $\mathbf{P}_i = \mathbf{0}$ or $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$.
3. If $\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i \neq \mathbf{0}$, the output is the previously buffered 1-bit ciphertext,

$$c_{i-2n} = p_{i-2n} \oplus k_{i-2n}$$

otherwise if $\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$, the output is the previously buffered n-bit vector,

$$\mathbf{P}_{i-n} (= p_{i-n}, p_{i-n-1}, p_{i-n-2}, \cdots, p_{i-2n+1})$$

otherwise if $\mathbf{P}_i = \mathbf{0}$, the output is three n-bit vectors,

$$\mathbf{P}_{i-n},\ \mathbf{P}_i \text{ and } \mathbf{P}_{i+n}$$

Receiver :

1. Put ciphertext bit c_i and plaintext bit $c_i \oplus k_i$ in two n-stage shift registers respectively, and shift four registers ($\mathbf{C}_i$, $\mathbf{C}_{i-n}$, $\mathbf{C}_i \oplus \mathbf{K}_i$, $\mathbf{C}_{i-n} \oplus \mathbf{K}_{i-n}$).
2. Check two n-bit vector $\mathbf{C}_i = \mathbf{0}$ or $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{0}$.
3. If $\mathbf{C}_i \neq \mathbf{0}$, $\mathbf{C}_i \oplus \mathbf{K}_i \neq \mathbf{0}$, the output is the previously buffered 1-bit plaintext,

$$p_{i-2n} = c_{i-2n} \oplus k_{i-2n}$$

otherwise if $\mathbf{C}_i \neq \mathbf{0}$, $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{0}$, the output is the previously buffered n-bit vector,

$$\mathbf{C}_{i-n} = \mathbf{P}_{i-n} (= p_{i-n}, p_{i-n-1}, p_{i-n-2}, \cdots, p_{i-2n+1})$$

otherwise if $\mathbf{C}_i = \mathbf{0}$, the output is three n-bit vectors,

$$\mathbf{C}_{i-n} = \mathbf{P}_{i-n},\ \mathbf{C}_i = \mathbf{P}_i \text{ and } \mathbf{C}_{i+n} = \mathbf{P}_{i+n}.$$

Theorem 1. *Under the condition that it is not permitted to present k-bit consecutive zeros for a plaintext in a synchronous stream cipher, the ZS-2 algorithm limits the output of consecutive k-bit zeros at the sender and then (excluding channel error) accurately recovers the plaintext at the receiver.*

Proof.

(i) If $\mathbf{P}_i \neq \mathbf{0}$ and $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$, then $c_{i-n} = p_{i-n} \oplus k_{i-n}$ is transmitted to the receiver. As a result, the receiver output is $q_{i-n} = c_{i-n} \oplus k_{i-n} = (p_{i-n} \oplus k_{i-n}) \oplus k_{i-n} =$

$= p_{i-n}$ and the plaintext is completely recovered.

(ii) When detected, only $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$ ($\mathbf{P}_i \neq \mathbf{0}$ and $\mathbf{P}_i = \mathbf{K}_i$), and then $\mathbf{C}_{i-1} = \mathbf{P}_{i-1} \oplus \mathbf{K}_{i-1}$ and $\mathbf{C}_i = \mathbf{P}_i$ are transmitted to the receiver. As such, the receiver detects $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{P}_i \oplus \mathbf{P}_i = \mathbf{0}$ and then outputs $\mathbf{Q}_{i-1} = \mathbf{C}_{i-1} \oplus \mathbf{K}_{i-1} = \mathbf{P}_{i-1} \oplus \mathbf{K}_{i-1} \oplus \mathbf{K}_{i-1} = \mathbf{P}_{i-1}$ and $\mathbf{Q}_{i-1} = \mathbf{C}_i = \mathbf{P}_i$ and the plaintext is completely recovered.

(iii) When detected, $\mathbf{P}_i = \mathbf{0}$, $\mathbf{C}_{i-1} = \mathbf{P}_{i-1}$, $\mathbf{C}_i = \mathbf{P}_i$, $\mathbf{C}_{i+1} = \mathbf{P}_{i+1}$ are transmitted to the receiver. Then the receiver detects $\mathbf{C}_i = \mathbf{0}$ and outputs $\mathbf{Q}_{i-1} = \mathbf{C}_{i-1} = \mathbf{P}_{i-1}$, $\mathbf{Q}_i = \mathbf{C}_i = \mathbf{P}_i$, and $\mathbf{Q}_{i+1} = \mathbf{C}_{i+1} = \mathbf{P}_{i+1}$, such that the plaintext is completely recovered.

(iv) When detected, $\mathbf{P}_{i-1} = \mathbf{0}$ and $\mathbf{P}_i = \mathbf{0}$, ZS-2 outputs $\mathbf{C}_{i-1} = \mathbf{P}_{i-1}$ and $\mathbf{C}_i = \mathbf{P}_i$, therefore, the output of ZS-2 at the sender cannot permit k ($= 2n-1$ or $2n$)-bit consecutive zeros based on the assumption that,

$$n = \left\lceil \frac{k+1}{2} \right\rceil$$

and k is a positive odd or even number (selected from the system characteristics).

2.3 New ZS-3 algorithm

ZS-2 has various good properties, including the omission of block synchronization, easy implementation, etc., yet a weakness in channel error propagation. Accordingly, this study minimized the bit-wide substitution in the substitution blocks in an attempt to improve the degenerated error property in a noisy channel. The proposed algorithms at the sender and the receiver are as follows:

Sender (see Figs. 1 and 2):

1. Put plaintext bit p_i and ciphertext bit $p_i \oplus k_i$ in two n-stage shift registers respectively, and shift four registers ($\mathbf{P}_i$, $\mathbf{P}_{i-n}$, $\mathbf{P}_i \oplus \mathbf{K}_i$, $\mathbf{P}_{i-n} \oplus \mathbf{K}_{i-n}$) (Fig. 1).
2. Check two n-bit vector $\mathbf{P}_i = \mathbf{0}$ or $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$.
3. If $\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i \neq \mathbf{0}$, the output is the previously buffered 1-bit ciphertext,

$$c_{i-2n} = p_{i-2n} \oplus k_{i-2n}$$

otherwise if $\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$, the output is the previously buffered n-bit vector,

$$\mathbf{P}_{i-n} (= p_{i-n}, p_{i-n-1}, p_{i-n-2}, \cdots, p_{i-2n+1})$$

otherwise if $\mathbf{P}_i = \mathbf{0}$, the output is three vectors, $\mathbf{P'}_{i-n}$, $\mathbf{P}_i$ and $\mathbf{P'}_{i+n}$, totally $(u+n+v)$-bit substituted.

Where block $\mathbf{P'}_{i-n}$ (fig. 2) consists of the first $(n-u)$-bit subblock of the ciphertext block, $\mathbf{P}_{i-n} \oplus \mathbf{K}_{i-n}$, corresponding to the start bit to the last '1' in block $\mathbf{P}_{i-n}$ and the last (remained) u-bit subblock of the plaintext block, $\mathbf{P}_{i-n}$, contained with the boundary '1'. Plus block $\mathbf{P'}_{i+n}$ consists of the last $(n-v)$-bit subblock of the ciphertext block, $\mathbf{P}_{i+n}$ $\mathbf{K}_{i+n}$, corresponding to after the first '1' to the end of block $\mathbf{P}_{i+n}$ and the first (remained) v-bit subblock of the plaintext block, $\mathbf{P}_{i+n}$, contained with the boundary '1'.

Receiver (see Figs. 1 and 2) :

1. Put ciphertext bit c_i and plaintext bit $c_i \oplus k_i$ in two n-stage shift registers respectively, and shift four registers ($\mathbf{C}_i$, $\mathbf{C}_{i-n}$, $\mathbf{C}_i \oplus \mathbf{K}_i$, $\mathbf{C}_{i-n} \oplus \mathbf{K}_{i-n}$) (Fig. 1).
2. Check two n-bit vector $\mathbf{C}_i = \mathbf{0}$ or $\mathbf{C}_i \oplus \mathbf{K}_i = 0$.
3. If $\mathbf{C}_i \neq \mathbf{0}$, $\mathbf{C}_i \oplus \mathbf{K}_i \neq \mathbf{0}$, the output is the previously buffered 1-bit plaintext,

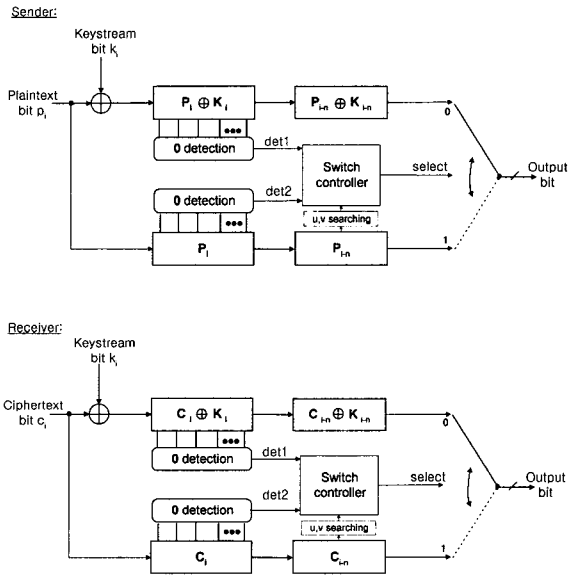
$$p_{i-2n} = c_{i-2n} \oplus k_{i-2n}$$

otherwise if $\mathbf{C}_i \neq \mathbf{0}$, $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{0}$, the output is the previously buffered n-bit vector,

$$\mathbf{C}_{i-n} = \mathbf{P}_{i-n} (= p_{i-n}, p_{i-n-1}, p_{i-n-2}, \cdots, p_{i-2n+1})$$

otherwise if $\mathbf{C}_i = \mathbf{0}$, the output is three vectors, $\mathbf{C'}_{i-n}$, $\mathbf{C}_i$ and $\mathbf{C'}_{i+n}$, totally $(u+n+v)$-bit substituted.

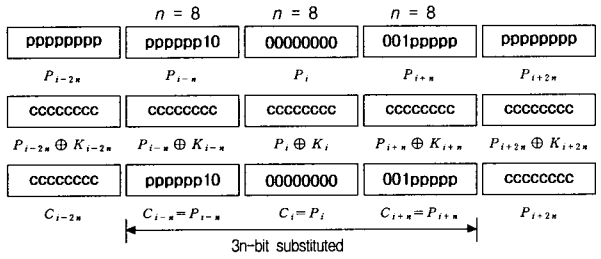
Where block $\mathbf{C'}_{i-n}$ (fig. 2) consists of the first $(n-u)$-bit subblock of the plaintext block, $\mathbf{C}_{i-n} \oplus \mathbf{K}_{i-n}$, corresponding



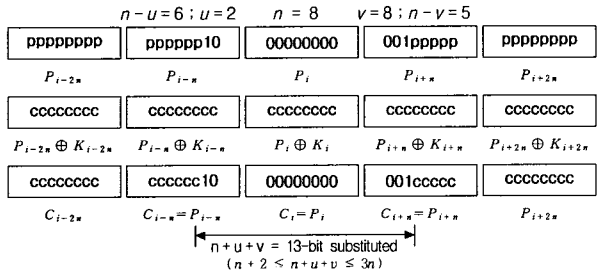
※ Switch controller : if det2 = 1, select SW = 1 and output n+u+v bits
else if det1 = 1, select SW = 1 and output n bit
else, select SW = 0 and output 1 bit
※ u, v searching : refer to (fig. 2).

(Fig.1) ZS-3 algorithm.

to the start bit to the last '1' in block $\mathbf{C}_{i-n}$ and the last (remained) u-bit subblock of the ciphertext block, $\mathbf{C}_{i-n}$, contained with the boundary '1'. And block $\mathbf{C'}_{i+n}$ consists of the last $(n-v)$-bit subblock of the plaintext block, $\mathbf{C}_{i+n} \oplus \mathbf{K}_{i+n}$, corresponding to after the first '1' to the end of block $\mathbf{C}_{i+n}$ and the first (remained) v-bit subblock of the ciphertext block, $\mathbf{C}_{i+n}$, contained with the boundary '1'.



(a) An example of substitution in ZS-2



(b) An example of substituted in zs-3

(Fig. 2) Examples of substitution comparison (n=8,u=2,v=3).

Theorem 2. Under the condition that it is not permitted to present k-bit consecutive zeros for a plaintext in a synchronous stream cipher, the ZS-3 algorithm limits the output of consecutive k-bit zeros at the sender and then (excluding channel error) accurately recovers the plaintext at the receiver.

Proof.

(i) If $\mathbf{P}_i \neq \mathbf{0}$ and $\mathbf{P}_i \oplus \mathbf{K}_i \neq \mathbf{0}$, then $c_i = p_i \oplus k_i$ is transmitted to the receiver. As a result, the receiver output is $q_i = c_i \oplus k_i = (p_i \oplus k_i) \oplus k_i = p_i$, and the plaintext bit is completely recovered.

(ii) When detected, only $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$ ($\mathbf{P}_i \neq \mathbf{0}$ and $\mathbf{P}_i = \mathbf{K}_i$), and then $\mathbf{C}_i = \mathbf{P}_i$ are transmitted to the receiver. As such the receiver detects $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{P}_i \oplus \mathbf{P}_i = \mathbf{0}$ and then outputs $\mathbf{Q}_i = \mathbf{C}_i = \mathbf{P}_i$, such that the plaintext block is completely recovered.

(iii) When detected, $\mathbf{P}_i = \mathbf{0}$, three plaintext blocks (two partial), $\mathbf{C}_{i-n} = \mathbf{P'}_{i-n}$, $\mathbf{C}_i = \mathbf{P}_i$ and $\mathbf{C}_{i+n} = \mathbf{P'}_{i+n}$, are transmitted to the receiver. Then the receiver detects $\mathbf{C}_i = \mathbf{0}$ and outputs $\mathbf{Q}_{i-n} = \mathbf{C'}_{i-n} = \mathbf{P}_{i-n}$, $\mathbf{Q}_i = \mathbf{C}_i = \mathbf{P}_i$, and $\mathbf{Q}_{i+n} = \mathbf{C'}_{i+n} = \mathbf{P}_{i+n}$, as such the three plaintext blocks are completely recovered.

(iv) When detected, $\mathbf{P}_{i-n} = \mathbf{0}$ and $\mathbf{P}_i = \mathbf{0}$, ZS-3 outputs $\mathbf{C}_{i-n} = \mathbf{P}_{i-n}$ and $\mathbf{C}_i = \mathbf{P}_i$, therefore, the output of ZS-3 at the sender cannot permit k (= $2n-1$ or $2n$)-bit consecutive zeros based on the assumption that,

$$n = \left\lceil \frac{k+1}{2} \right\rceil$$

and k is a positive odd or even number (selected from system characteristics).

Because the ZS-3 algorithm, an improved version of ZS-2 in serial detection/partial block substitution, includes partial-bit substitution of the front and rear blocks of all zero-blocks, it minimizes the bit-error propagation in the channels. Based on experimental results, ZS-3 can reduce the bit error propagation from $3n$-bit to $(u+n+v)$-bit, $u+n+v \leq 3n$.

3. Analysis

Occasionally there will be an n-bit error propagation due to a channel bit error from the substituted block. The total error rate of ZS-1 or ZS-2 [4-5] can be simulated as follows.

$$P_E(ZS-1) = (n)(2^{-n})\{1-(1-B)^n\} + B \quad (1)$$

$$P_E(ZS-2) = (n)(2^{-(n-2)})\{1-(1-B)^n\} + B \quad (2)$$

, where B is BER (Bit Error Rate in channel).

The error propagation property of ZS-3 is similar to that of ZS-2, however, it also differs in that the first/last block is substituted by u or v ($\leq n$) bits instead of n bits in a triple block substitution. The bit error propagation rate is varied based on the parameters (u, v, etc.). Let P_{M1} be the probability of missing the detection of a substituted block when a single block substitution occurs at the sender, P_{M3} be the probability of missing the detection of a substituted block when a triple block substitution occurs at the sender, P_{F1} be the probability of a false-detection of (similar) unsent single substitution blocks at the sender, and P_{F3} be the probability of a false-detection of (similar) unsent triple substitution blocks at the sender. For a large n (≥ 6) the total bit error propagation rate, P_E, can be computed by the following :

P_{M1} = {the probability that the substituted single blocks occurred at the sender} × {the missing probability that the substituted single blocks were not detected by a channel error} × {the mean number of error propagation bits from a channel error in the block}

$= (2^{-n})\{1-(1-B)^n\}(n/2)$

$= (n)(2^{-(n+1)})\{1-(1-B)^n\}$

P_{M3} = {the probability that the triple block substitutions occurred at the sender} × {the missing probability that the substituted triple blocks were not detected by a channel error} × {the mean number of error propagation bits from a channel error in the blocks}

$= \{(1+(4/n))(2^{-n})\}\{1-(1-B)^n\}(n/2)$

$= \{1+(4/n)\}P_{M1}$

P_{F1} = {the false detection probability that the substituted single blocks occurred due to an unsent channel error at the sender} × {the mean number of propagation bits from a channel error in the block}

$= \{1-(1-B)^n)(2^{-n})\}(n/2)$

$= (n)(2^{-(n+1)})\{1-(1-B)^n\}$

P_{F3} = {the false detection probability that the substituted triple blocks occurred due to an unsent channel error at the sender} × {the mean number of the propagation bits from a channel error in the blocks}

$= \{1+(4/n))(2^{-n})\}\{1-(1-B)^n\}(3n/2)$

$= \{1+(4/n)\}3P_{M1}$

$P_E = P_{M1}+P_{M3}+P_{F1}+P_{F3}+B$

$= \{4+(8/n)\}P_{M1}+B$

$= \{(n/2)+1\}\{2^{-(n-2)}\}\{1-(1-B)^n\}+B$

〈Table 1〉 Total error rate of ZS for variable n (at BER = 10^{-5})

n	P_E (ZS-1)	P_E (ZS-2)	P_E (ZS-3)	Improved Rate (%)
6	1.5632498×10^{-5}	3.2529991×10^{-5}	2.5019994×10^{-5}	23.0
7	1.3833208×10^{-5}	2.5332834×10^{-5}	1.8761619×10^{-5}	25.9
8	1.2503307×10^{-5}	2.0013229×10^{-5}	1.6258268×10^{-5}	18.7%
9	1.1584116×10^{-5}	1.6336465×10^{-5}	1.3520258×10^{-5}	17.3
10	1.0977844×10^{-5}	1.3911378×10^{-5}	1.2346827×10^{-5}	11.2
11	1.0591593×10^{-5}	1.2366372×10^{-5}	1.1290748×10^{-5}	8.7
12	1.0352020×10^{-5}	1.1408082×10^{-5}	1.0821381×10^{-5}	5.2
13	1.0206566×10^{-5}	1.0826266×10^{-5}	1.0444912×10^{-5}	3.6
14	1.0119783×10^{-5}	1.0479134×10^{-5}	1.0273791×10^{-5}	2.0
15	1.0068753×10^{-5}	1.0275012×10^{-5}	1.0146673×10^{-5}	1.3%
16	1.0039112×10^{-5}	1.0156450×10^{-5}	1.0088003×10^{-5}	0.6
20	1.0003819×10^{-5}	1.0015278×10^{-5}	1.0008403×10^{-5}	0.1
25	1.0000186×10^{-5}	1.0000746×10^{-5}	1.0000388×10^{-5}	0
31	1.0000004×10^{-5}	1.0000018×10^{-5}	1.0000009×10^{-5}	0

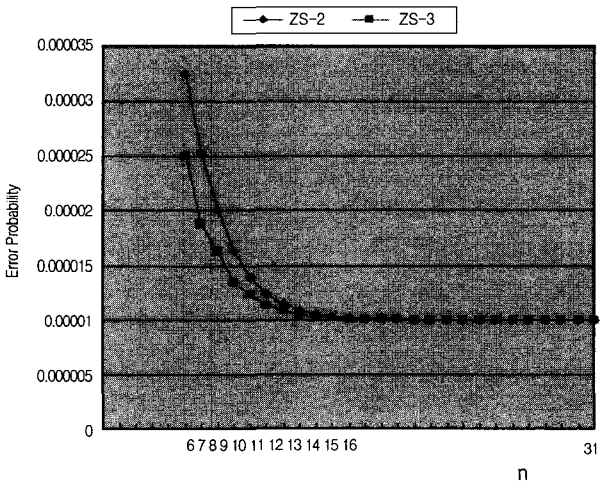
Based on the results of the computer simulation, the total bit error rates for the three types of ZS at B = 10-5 were as shown in <table 1>. In addition, the functions were super-decreasing by n and exhibited good channel bit error properties when selecting a large n.

〈Table 2〉 Total error rate of ZS for variable *BER* (k=15, n=8)

BER	P_E (ZS-1)	P_E (ZS-2)	P_E (ZS-3)
10^{-1}	1.1779790×10^{-1}	1.7119160×10^{-1}	1.4449475×10^{-1}
10^{-2}	1.2414228×10^{-2}	1.9656913×10^{-2}	1.6035570×10^{-2}
10^{-3}	1.2491268×10^{-3}	1.9965071×10^{-3}	1.6228169×10^{-3}
10^{-4}	1.2499125×10^{-4}	1.9996500×10^{-4}	1.6247813×10^{-4}
10^{-5}	1.2499912×10^{-5}	1.9999650×10^{-5}	1.6249781×10^{-5}
10^{-6}	1.2499991×10^{-6}	1.9999965×10^{-6}	1.6249978×10^{-6}
10^{-7}	1.2499999×10^{-7}	1.9999997×10^{-7}	1.6249998×10^{-7}
10^{-8}	1.2500000×10^{-8}	2.0000000×10^{-8}	1.6250000×10^{-8}
10^{-9}	1.2500000×10^{-9}	1.9999999×10^{-9}	1.6250000×10^{-9}
10^{-10}	$1.2500000 \times 10^{-10}$	$2.0000000 \times 10^{-10}$	$1.6250000 \times 10^{-10}$

<Table 3> is a summary of the current proposal. The ZS method reduces the occurrence of long consecutive-zero sequences in encrypted data used in a stream cipher. As a result, the receiver clock can be easily recovered, while maintaining the cryptographic security of the underlying stream cipher. ZS-1 uses a block detection method, whereas ZS-2 uses a precise serial block detection method. Therefore, ZS-1 requires block synchronization at the start and end of each block, while ZS-2 does not. In terms of hardware implementation, ZS-2 is simpler than ZS-1 because of the omission of block synchronization.

In contrast, the proposed ZS-3 algorithm is a revised version of ZS-2, which although easily applied to a stream cipher, also includes a heavy error propagation property. In particular, when applying ZS-2 to a T1 carrier system at n = 8 or k = 15 the BER is approximately 2 times higher than that of ZS-1, however, under the same conditions ZS-3



(Fig. 3) Bit error rate of ZS algorithms for variable n (BER = 10^{-5})

produced a more acceptable BER that was only 1.625 times higher than that with ZS-1, that is, an 18.7% improvement on ZS-2 as shown in (fig. 3) (also <table 1>~<table 3>). Without ZS algorithms, the mean numbers of a keystream synchronization error resulting from excessive zeros in a synchronous stream cipher system are rapidly increased due to the number of transmitted data or by decreasing the k-parameter of the ZS algorithm. In contrast, this will be zero or null when applying a ZS algorithm.

〈Table 3〉 Comparison of ZS-1 through ZS-3

Items	ZS-1	ZS-2 and improved ZS-3
Detection method	Block detection : detects all-zero input blocks by checking block interval with block size	Serial detection: detects all-zero input blocks by checking bit-serial interval with block size
Block synchro-nization	Required	Not required
System clock recovery	When applying a synchronous stream cipher to a point-to-point link encryption in a T1-carrier system, there is a limit of 15 or more consecutive zeros in the ciphertext at the sender.	
Error propaga-tion property	Smaller than ZS-2 and ZS-3	ZS-2 is larger than ZS-3. ZS-3 produces an 18.7% improvement over ZS-2 at n = 8 or k = 15
Implemen-tation complexity	More complex than ZS-2 and ZS-3 (available for software implementation)	Simple and easy for hardware or software implementation

4. Conclusion

This paper proposed a new algorithm, ZS-3, as an updated version of ZS-2, which although easily applied to a stream cipher also includes a heavy error propagation property. Based on experimental results, the application of ZS-2 to a T1 carrier system at $n = 8$ or $k = 15$ increased the BER (bit error rate in channel) to approximately 2 times that with ZS- 1, however, under the same conditions the use ZS-3 produced a more acceptable BER that was only 1.625 times higher than that with ZS-1, that is, an 18.7% improvement on ZS-2. Furthermore, in terms of hardware implementation, ZS-3 (also ZS-2) is much simpler than ZS-1, because of the omission of block synchronization.

References

[1] H. J. Beker and F. C. Piper, *Cipher Systems : The Protection of Communications*, Orthwood Books, London, 1982.

[2] Henk C. A. van Tilborg, *An Introduction to Cryptology*, KLUWER ACADEMIC PUBLISHERS, Boston, etc., 1988.

[3] S. W. Golomb, Shift Register Sequences, Holden-Day, San Francisco, 1967.

[4] Hoonjae Lee and Sangjae Moon, "On ZS Synchronization Algorithm for Synchronous Stream Cipher," Applied Signal Processing (London) Vol.5, No.4, pp.240-243, 1998.

[5] Hoonjae Lee, Sangjae Moon, "A New ZS Algorithms for Synchronous Stream Cipher," appears in Applied Signal Processing (London), Vol.6, No.4, pp.177-181, 1999.

[6] CCITT Rec. G.703 : Physical/Electrical Characteristics of Hierarchical Digital Interface, CCITT red book, Vol.III, 1985.

[7] J. Daemen, R. Govaerts and J. Vandewalle : Resynchronization Weaknesses in Synchronous Stream Ciphers', *Advances in Cryptology - Eurocrypto'93*, Lecture Notes in Computer Science, No.765, Springer-Verlag, pp.159-167, 1994.

이 훈 재

e-mail : hjlee@dongseo.ac.kr

1985년 경북대학교 전자공학과 졸업(학사)

1987년 경북대학교 전자공학과 졸업(석사)

1998년 경북대학교 전자공학과 졸업(박사)

1987년~1998년 국방과학연구소 선임연구원

1998년~2002년 경운대학교 컴퓨터전자정보공학부 조교수

2002년~현재 동서대학교 인터넷공학부 정보네트워크공학전공 조교수

관심분야 : 정보보호, 네트워크보안, 정보통신/네트워크