DBpia

논문 99-24-5A-16 한국통신학회논문지 '99-5 Vol.24 No.5

채널 오류 확산을 줄인 ZS 동기 알고리듬

정회원 이 훈 재*, 문 상 재**

A ZS Synchronization Algorithm with a Small Channel Error Propagation

Hon-Jae Lee*, Sang Jae Moon** *Regular Members*

요 약

동기식 스트림 암호 시스템 적용을 위한 ZS 동기 알고리듬이 다수 제안되었지만, 그 중에서 ZS-2 알고리듬은 에러 확산이 많은 단점이 있다. 그러나 ZS-2 알고리듬에 대하여 블록 대체 비트 수를 최소화 시킬 경우 에러 확산을 줄일 수 있으며, 본 논문에서는 이러한 방법으로 개선하여 n=8일 때 에러 확산율이 평균 18.7% 감축되는 새로운 ZS-3 알고리듬을 제안한다.

ABSTRACT

Among Zero Suppression(ZS) algorithms for synchronous stream cipher system, a ZS-2 has a weakness of channel error propagation. By minimizing substituted bits in block substitution, it will be improved. In this paper we propose a ZS-3 algorithm which decreases a mean error propagation about 18.7% to ZS-2's at n=8.

Ⅰ. 서 론

동기식 스트림 암호는 이진 키 수열(keystream)의 PN-특성이 양호하여 출력 암호문에 "1"과 "0"이 균일 분포된다[1-3]. 이 경우 수신 데이타에는 연속 "0"이 나타날 수 있으며, 이러한 현상은 평문 통신에서는 없었던 새로운 문제를 유발할 수 있다. 이러한 문제를 해결하여 암호화 후에도 필요에 따라 k(>2) 비트 이하로 연속 "0"을 억제하는 것이 ZS 동기 방식이다. 암호 통신에 적용된 ZS 방식은 블록 검출 방식과 직렬 검출 방식이 있으며, ZS-1 알고리듬[4]은 블록 검출 방식이고, ZS-2 알고리듬[4]은 직렬 검출 방식이다. 이 중에서 ZS-2 알고리듬은 하드웨어 구현이 용이한 반면 채널 오류에 취약하여 오류 확산이 커지는 단점이 있다.

본 논문에서는 n=8일 때 ZS-2에 비하여 에러 확산이 평균 18.7% 개선되는 새로운 알고리듬 ZS-3을 제안한다. 제안 방식은 T1 회선등[5-7]과 같은 제한된 채널 특성을 갖는 회선에서 암호화와 병행하여 사용시 수신 클럭 불안정 문제를 해결할 수 있기 때문에 Daemen등[5]이 지적한 잦은 재동기로 인한 비도 저하를 막을 수 있다.

Ⅱ. 알고리듬 제안

ZS 알고리듬에 사용될 변수 k는 채널에서 최대로 허용되는 연속 "0" 비트수이며, 알고리듬에서 처리하는 블록 크기 $n = \lceil (k+1)/2 \rceil$이다($\lceil x \rceil$는 x를 넘지 않는 최대 정수). 그리고 ZS 알고리듬은 기능상으로 연속 "0"을 검출하는 검출부와 블록 크기 만큼 다른값으로 대체시키는 대체부로 나누어진다. 검출부는 ZS 입력단에 연속 "0"을 검출하는 것으로서 검출 판단 시점을 블록 크기 만큼 건너 뛰면서 검출하는 블록(block)방식과 1 비트씩 직렬로 검출 판단하는 직렬(serial)방식으로 나누어진다. 또

* 경운대학교 컴퓨터공학과(hjlee@Kyungwoon.ac.kr)
** 경북대학교 공과대학 전기전자공학부
논문번호 : 97342-0924, 접수일자 : 1997년 9월 24일

www.dbpia.co.kr

한 대체부는 연속 "0"이 검출되었을 때 다른 블록으로 대체시키는 것으로서 블록 크기의 정수배만큼 대체하는 블록 대체와 일부분만을 대체하는 부분 대체로 나누어진다. ZS-2 알고리듬은 직렬 검출/블록 대체방식이고, 제안될 ZS-3 알고리듬은 직렬 검출/부분 대체 방식이다.

직렬 검출의 편의상 연속되는 이웃 블록간에 겹쳐지도록 평문 블록, 키 수열 블록, 암호문 블록, 복호 평문 블록 및 **0** 벡터를 다음과 같이 정의한다.

i번째 평문 블록 $\mathbf{P}_i$: $(p_i, p_{i-1}, \cdots, p_{i-n+1})$

i번째 키 수열 블록 $\mathbf{K}_i$: $(k_i, k_{i-1}, \cdots, k_{i-n+1})$

i번째 암호문 블록 $\mathbf{C}_i$: $(c_i, c_{i-1}, \cdots, c_{i-n+1})$

i번째 복호 평문 블록 $\mathbf{Q}_i$: $(q_i, q_{i-1}, \cdots, q_{i-n+1})$

"0" 비트 벡터 **0** : (0, 0, ..., 0)

(가정)

1) 암호 시스템에서 임의로 잉여 비트를 삽입 또는 삭제할 수 없다.(CODEC과 MODEM 중간 시스템에서 클럭 rate의 증감이 어려움)

2) 평문에서 k 비트(k=2n-1 또는 k=2n) 이하로 연속 "0"이 억제된다.

3) 키 수열 발생기는 암호학적으로 충분한 비도를 갖는다.

1. ZS-2 알고리듬 분석

ZS-2 알고리듬 : 그림 1 참조

(송신) 1) $p_i \oplus k_i$ 암호문과 p_i가 비트 크기로 n단 이동 레지스터에 1비트씩 입력된다.

2) $\mathbf{P}_i$블록과 $\mathbf{P}_i \oplus \mathbf{K}_i$블록이 각각 **0**인지 검사한다.

3) $\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i \neq \mathbf{0}$인 경우($\mathbf{P}_i \neq \mathbf{K}_i$) : $c_{i-n} = p_{i-n} \oplus k_{i-n}$를 1비트를 정상 출력시킨다.

$\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$인 경우($\mathbf{P}_i = \mathbf{K}_i$) : $\mathbf{C}_i = \mathbf{P}_i$의 n비트 블록을 대체 출력시킨다.

$\mathbf{P}_i = \mathbf{0}$ 경우($\mathbf{P}_i \oplus \mathbf{K}_i$와 무관): $\mathbf{C}_{i-n} = \mathbf{P}_{i-n}$, $\mathbf{C}_i = \mathbf{P}_i$, $\mathbf{C}_{i+n} = \mathbf{P}_{i+n}$ 연속 3블록 3n 비트를 대체 출력시킨다.

(수신) 1) $c_i \oplus k_i$ 복호문과 c_i가 비트 크기로 n단 이동 레지스터에 1비트씩 입력된다.

2) $\mathbf{C}_i$블록과 $\mathbf{C}_i \oplus \mathbf{K}_i$블록이 각각 **0** 인지 검사한다.

3) $\mathbf{C}_i \neq \mathbf{0}$, $\mathbf{C}_i \oplus \mathbf{K}_i \neq \mathbf{0}$인 경우($\mathbf{C}_i \neq \mathbf{K}_i$): $q_{i-n} = c_{i-n} \oplus k_{i-n}$를 1비트를 정상 출력시킨다.

$\mathbf{C}_i \neq \mathbf{0}$, $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{0}$인 경우($\mathbf{C}_i = \mathbf{K}_i$): $\mathbf{Q}_i = \mathbf{C}_i$의 n비트 블록을 대체 출력시킨다.

$\mathbf{C}_i = \mathbf{0}$ 경우($\mathbf{C}_i \oplus \mathbf{K}_i$와 무관): $\mathbf{Q}_{i-n} = \mathbf{C}_{i-n}$, $\mathbf{Q}_i = \mathbf{C}_i$, $\mathbf{Q}_{i+n} = \mathbf{C}_{i+n}$ 연속 3블록(3n 비트)를 대체 출력시킨다.

상기 알고리듬에서 p_i와 $\mathbf{P}_i$는 구분되어야 하는데 p_i는 1 비트 평문 비트를 말하며, $\mathbf{P}_i$는 n 비트의 평문 벡터($p_i, p_{i-1}, \cdots, p_{i-n+1}$)를 의미한다. ZS-2 알고리듬은 직렬 검출 방식이기 때문에 ZS-1 알고리듬에 비해서 하드웨어 부담이 감소되고 구현이 용이하다. 그러나 블록 대체된 부분에 채널 오류가 발생되면 오류가 확산되는 문제점을 안고 있다.

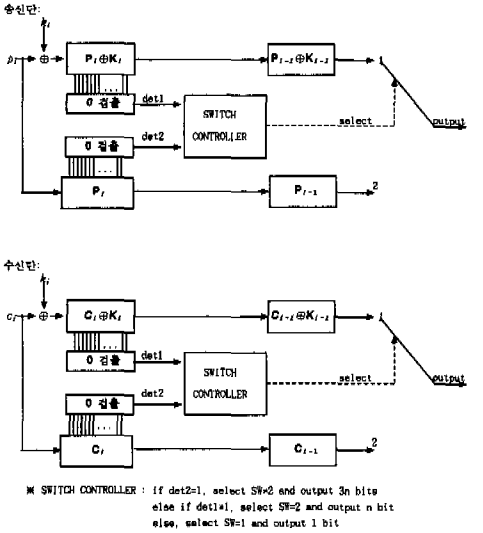


그림 1. ZS-2 알고리듬

2. ZS-3 알고리듬 제안

ZS-3 알고리즘은 그림 2와 같이 ZS-2 알고리듬을 개선하여 평문 3블록 연속 대체시에 직전 블록의 앞부분 일부와 직후 블록의 뒷부분 일부를 대체에서 제외시킴으로서 오류 확산을 최소화 시킬 수 있는 직렬 검출/부분 대체 방식이다.

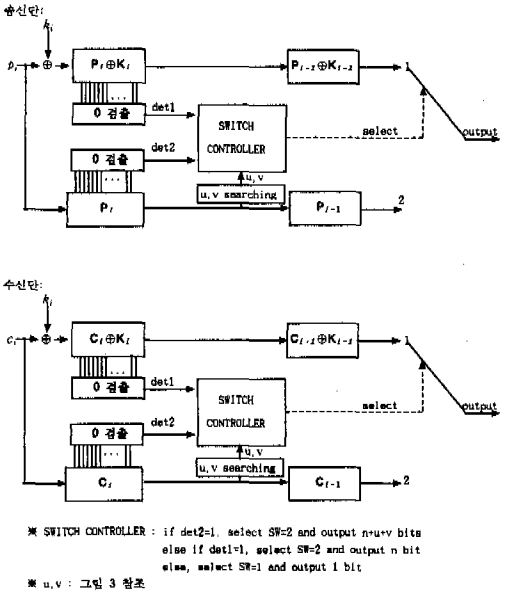


그림 2. ZS-3 알고리즘

www.dbpia.co.kr

ZS-3 알고리즘 : 그림 2 및 그림 3 참조

(송신) 1) $p_i \oplus k_i$ 암호문과 p_i가 비트 크기로 n단 이동 레지스터에 1비트씩 입력된다.

2) $\mathbf{P}_i$블록과 $\mathbf{P}_i \oplus \mathbf{K}_i$블록이 각각 0인지 검사한다.

3) $\mathbf{P}_i \neq 0$, $\mathbf{P}_i \oplus \mathbf{K}_i \neq 0$인 경우($\mathbf{P}_i \neq \mathbf{K}_i$) : $c_{i-n} = p_{i-n} \oplus k_{i-n}$를 1비트를 정상 출력시킨다.

$\mathbf{P}_i \neq 0$, $\mathbf{P}_i \oplus \mathbf{K}_i = 0$인 경우($\mathbf{P}_i = \mathbf{K}_i$) : $\mathbf{C}_i = \mathbf{P}_i$의 n비트 블록을 대체 출력시킨다.

$\mathbf{P}_i = 0$ 경우($\mathbf{P}_i \oplus \mathbf{K}_i$와 무관): $\mathbf{C}_{i-n} = \mathbf{P'}_{i-n}$, $\mathbf{C}_i = \mathbf{P}_i$, $\mathbf{C}_{i+n} = \mathbf{P'}_{i+n}$ 연속 3블록 n+u+v 비트를 대체 출력시킨다.

여기서, $\mathbf{P'}_{i-n}$ 블록이란 그림 3과 같이 처음 n-u 비트($\mathbf{P}_{i-n}$ 블록에서 최후 "1"이 나오기직전 까지의 비트수)는 $\mathbf{P}_{i-n} \oplus \mathbf{K}_{i-n}$ 암호문 블록의 처음 n-u 비트를 그대로 두고 나중 u비트는 "1"을 포함한 $\mathbf{P}_{i-n}$의 후미 u 비트로 부분 대체시킨 블록을 말하며, $\mathbf{P'}_{i+n}$ 블록은 뒷부분 n-v 비트($\mathbf{P}_{i+n}$블록에서 최초 "1"이 나온 이후의 비트수)는 $\mathbf{P}_{i+n} \oplus \mathbf{K}_{i+n}$ 암호문 블록의 뒷부분 n-v 비트를 그대로 두고 처음 v 비트는 "1"을 포함한 $\mathbf{P}_{i+n}$의 처음 v 비트로 부분 대체시킨 블록을 말한다.

(수신) 1) $c_i \oplus k_i$ 복호문과 c_i가 비트 크기로 n단 이동 레지스터에 1비트씩 입력된다.

2) $\mathbf{C}_i$블록과 $\mathbf{C}_i \oplus \mathbf{K}_i$블록이 각각 0인지 검사한다.

3) $\mathbf{C}_i \neq 0$, $\mathbf{C}_i \oplus \mathbf{K}_i \neq 0$인 경우($\mathbf{C}_i \neq \mathbf{K}_i$): $q_{i-n} = c_{i-n} \oplus k_{i-n}$를 1비트를 정상 출력시킨다.

$\mathbf{C}_i \neq 0$, $\mathbf{C}_i \oplus \mathbf{K}_i = 0$인 경우($\mathbf{C}_i = \mathbf{K}_i$): $\mathbf{Q}_i = \mathbf{C}_i$의 n비트 블록을 대체 출력시킨다.

$\mathbf{C}_i = 0$ 경우($\mathbf{C}_i \oplus \mathbf{K}_i$와 무관): $\mathbf{Q}_{i-n} = \mathbf{C'}_{i-n}$, $\mathbf{Q}_i = \mathbf{C}_i$, $\mathbf{Q}_{i+n} = \mathbf{C'}_{i+n}$ 연속 3블록 n+u+v 비트를 대체 출력시킨다.

여기서, $\mathbf{C'}_{i-n}$ 블록이란 처음 n-u 비트($\mathbf{C}_{i-n}$블록에서 최후 "1"이 나오기 직전까지의 비트수)는 $\mathbf{C}_{i-n} \oplus \mathbf{K}_{i-n}$ 복호문 블록의 처음 n-u 비트를 그대로 두고 나중 u비트는 "1"을 포함한 $\mathbf{C}_{i-n}$의 후미 u 비트로 부분 대체시킨 블록을 말하며, $\mathbf{C'}_{i+n}$ 블록은 뒷부분 n-v 비트($\mathbf{C}_{i+n}$블록에서 최초 "1"이 나온 이후의 비트수)는 $\mathbf{C}_{i+n} \oplus \mathbf{K}_{i+n}$ 복호문 블록의 뒷부분 n-v 비트를 그대로 두고 처음 v 비트는 "1"을 포함한 $\mathbf{C}_{i+n}$의 처음 v 비트로 부분 대체시킨 블록을 말한다.

[정리 1] 평문에서 k-비트(k=2n-1 또는 k=2n) 이하로 연속 "0"이 억제된다는 가정하에서 ZS-3 알고리즘을 동기식 스트림 암호에 적용할 경우 송신단 암호문 출력에 역시 k-비트 이하로 연속 "0"이 억제되며, 채널 오류가 없을 경우 수신단에서 평문이 완벽하게 복호된다.

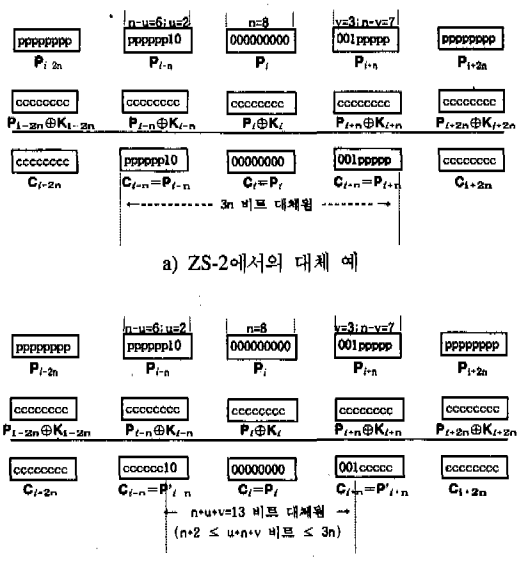


그림 3. 송신단 부분 대체의 예(n=8,u=2,v=3)

(증명) 채널 오류가 없을 경우 수신 평문의 복호 상태는 다음과 같이 완벽하게 복호된다.

1) 블록 대체 없는 경우($\mathbf{P}_i \neq 0$ 및 $\mathbf{P}_i \oplus \mathbf{K}_i \neq 0$) : 송신단에서는 $\mathbf{P}_i \neq 0$ 및 $\mathbf{P}_i \oplus \mathbf{K}_i \neq 0$ 이므로 $c_i = p_i \oplus k_i$의 1비트 암호문이 송신되고, 수신단에서는 $\mathbf{C}_i \neq 0$ 및 $\mathbf{C}_i \oplus \mathbf{K}_i \neq 0$ 이므로 $q_i = c_i \oplus k_i = p_i \oplus k_i \oplus k_i = p_i$로 1비트 평문이 정상 복호된다.

2) 1 블록만 대체 있는 경우($\mathbf{P}_i \neq 0$ 및 $\mathbf{P}_i \oplus \mathbf{K}_i = 0$) : 송신단에서는 $\mathbf{P}_i \neq 0$ 및 $\mathbf{P}_i \oplus \mathbf{K}_i = 0$ 이므로 $\mathbf{C}_i = \mathbf{P}_i$의 n-비트 블록이 송신되고, 수신단에서는 $\mathbf{C}_i \neq 0$ 및 $\mathbf{C}_i \oplus \mathbf{K}_i = 0$ 이므로 $\mathbf{Q}_i = \mathbf{C}_i = \mathbf{P}_i$로 n-비트 블록이 정상 복호된다.

3) 3 블록 대체 있는 경우($\mathbf{P}_i = 0$, $\mathbf{P}_i \oplus \mathbf{K}_i$는 무관함) : 송신단에서는 $\mathbf{P}_i = 0$ 이므로 $\mathbf{C}_{i-n} = \mathbf{P'}_{i-n}$, $\mathbf{C}_i = \mathbf{P}_i$, $\mathbf{C}_{i+n} = \mathbf{P'}_{i+n}$의 n-비트 3 블록이 송신되고, 수신단에서는 $\mathbf{C}_i = 0$ 이므로 $\mathbf{Q}_{i-n} = \mathbf{C'}_{i-n} = \mathbf{P}_{i-n}$, $\mathbf{Q}_i = \mathbf{C}_i = \mathbf{P}_i$, $\mathbf{Q}_{i+n} = \mathbf{C'}_{i+n} = \mathbf{P}_{i+n}$로 각각 n-비트 3 블록이 정상 복호된다.

결국 ZS-3는 ZS-2에서 발생되는 오류 확산 3n비트를 이보다 작은 n+u+v비트로 감축시킨 개선 알고리듬이다.

Ⅲ. 비트 오류 확산 비교 분석

채널에서의 비트 오류율(channel bit error rate)을

www.dbpia.co.kr

B라 둘 때 ZS-1과 ZS-2 알고리듬 적용시의 비트 오류율은 참고문헌[4]에서 알 수 있으며, ZS-3 알고리듬에서도 동일한 조건으로 이 값을 구한 다음 기존 방식과 비교한다.

1. ZS-1 알고리듬의 비트 오류율[4]

$$P_E(ZS\text{-}1) = (n)\ 2^{-n}[1-(1-B)^n] + B \qquad (1)$$

2. ZS-2 알고리듬의 비트 오류율[4]

$$P_E(ZS\text{-}2) = (n)\ 2^{-(n-2)}[1-(1-B)^n] + B \qquad (2)$$

3. ZS-3 알고리듬의 비트 오류율

ZS-3의 오류 특성은 ZS-2의 오류 특성과 비슷하며, 다만 3블록 대체시 직전 블록을 u비트 부분 대체시키고 직후 블록을 v비트 부분 대체시키는 차이가 있다. 비트오류율은 u,v값이 각 블록마다 다르기 때문에 가변적이며, 평문의 "0"와 "1"의 비율이 비슷(랜덤)할 경우 부분 대체 블록은 평균 2비트 정도 대체(u=2, v=2)가 일어난다고 판단할 수 있다. 충분히 큰 n(≥6)에 대해서 P_{M1}는 송신 1블록 대체시의 미검출 확률, P_{M3}는 송신 3블록 대체시의 미검출 확률, P_{F1}는 채널오류로 수신 1블록 대체될 오검출 확률, P_{F3}는 채널 오류로 수신 3블록 대체될 오검출 확률이라 둘때 전체 비트 오류율 P_E는 다음과 같이 계산된다.

P_{M1} = {송신 1 블록 대체될 확률} x {채널 오류로 블록 미검출될 확률}
x {블록내 평균 오류 확산 비트 수}
= $\{2^{-n}\}$ x $\{1-(1-B)^n\}$ x {n/2} = $(n)\ 2^{-(n+1)}[1-(1-B)^n]$

P_{M3} = {송신 3 블록 대체될 확률} x {채널 오류로 블록 미검출될 확률}
x {블록내 평균 오류 확산 비트 수}
= $\{[1+(4/n)]\ (2^{-n})\}$ x $\{1-(1-B)^n\}$ x {n/2} = $[1+(4/n)]\cdot P_{M1}$

P_{F1} = {비대체된 블록에 채널 오류 발생 확률} x {수신 1 블록 대체로 오검출될 확률} x {블록내 평균 오류 확산 비트 수}
= $\{1-(1-B)^n\}$ x $\{2^{-n}\}$ x {n/2} = $(n)\ 2^{-(n+1)}[1-(1-B)^n]$

P_{F3} = {비대체된 블록에 채널 오류 발생 확률} x {수신 3 블록 대체로 오검출될 확률} x {블록내 평균 오류 확산 비트 수}
= $\{1+(4/n)\ (2^{-n})\}$ x $\{1-(1-B)^n\}$ x {3n/2} = $[1+(4/n)]\cdot P_{M1}$

$P_E = P_{M1} + P_{M3} + P_{F1} + P_{F3} + B = [4+(8/n)]\cdot P_{M1} + B$
$= [(n/2)+1][2^{-(n-2)}][1-(1-B)^n] + B \qquad (3)$

4. 비트 오류 확산 비교

$B=10^{-5}$에서 ZS 알고리즘의 전체 비트 오류율을 n에 따라 구한 값은 표 1 및 그림 4와 같다. 3가지 모두 전체 비트 오류율은 n에 따라 단조 감소되어 채널 비트 오류율(BER)에 접근되므로 n값(k값)을 크게 선택할수록 좋은 오류 특성을 얻을 수 있다.

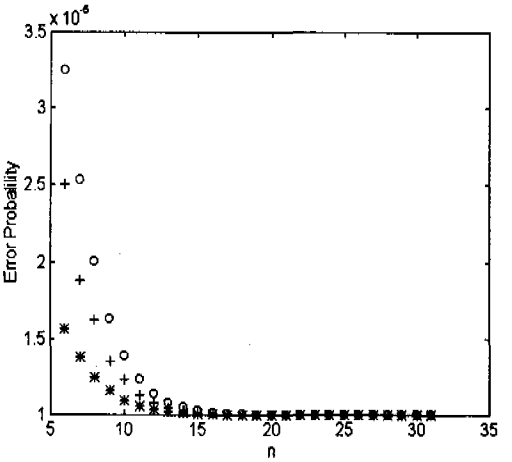


그림 4. n 가변에 따른 ZS 알고리즘들의 비트 오류 특성 (BER=10^{-5})

표 1. n 가변에 따른 ZS 알고리즘들의 비트 오류율 (BER=10^{-5})

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

특별히 k=15, n=8인 T1 전송 시스템에서 전체 비트 오류율은 표 2와 같아지며, 동일한 조건으로

www.dbpia.co.kr

비교할 때 BER 대비 ZS-1은 평균 1.25배, ZS-2는 2배의 증가가 있음을 알 수 있다. 그러나 ZS-2와 동일한 대체 방법을 쓰면서도 비트 오류 확산을 최소화시킨 ZS-3에서는 n=8일 때 1.625배의 증가에 그치기 때문에 ZS-2 방식과 비교할 경우 평균 18.7%(표 1 참조)의 개선이 있음을 알 수 있다.

표 2. BER 가변에 따른 ZS 알고리즘들의 비트 오류율(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}	$1.2500000 \times 10^{-10}$	$2.0000000 \times 10^{-10}$	$1.6250000 \times 10^{-10}$

ZS 알고리즘의 적용이 필요한 통신망에서는 여러 가지 통신 특성에 따라 ZS 알고리즘이 선택되어지겠지만, 여기서는 제안된 3가지 ZS 알고리즘의 특성을 표 3과 같이 비교 검토하였다.

표 3. 제안된 ZS 알고리즘 비교

비교 항목	ZS-1	ZS-2	ZS-3
전제 조건	$P_i \neq 0$	평문 통신에서 k 비트 이하로 연속 "0" 억제	평문 통신에서 k 비트 이하로 연속 "0" 억제
블록 동기	반드시 필요함	불필요	불필요
비트 지연	n 비트	2n 비트	2n 비트
오류 확산	확산 가능 (소)	확산 가능 (대)	확산 가능 (중)
n 선택	$n = \lceil (k+1)/2 \rceil$	$n = \lfloor (k+1)/2 \rfloor$	$n = \lceil (k+1)/2 \rceil$
구현 용이성	소프트웨어	하드웨어	하드웨어

* $\lceil x \rceil$: x를 넘지않는 최대 정수

Ⅳ. 결 론

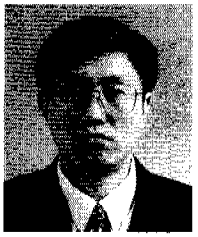
제안된 ZS-3 알고리듬은 ZS-2 알고리듬에 비하여 에러 확산을 줄이기 위하여 고안된 것이다. 분석 결과 ZS-2를 개선하여 블록 대체시 대체된 블록의 비트 수를 최소화 시킴으로서 에러 확산을 줄일 수 있었으며, n=8일 때 에러 확산은 평균적으로 18.7% 개선됨을 확인하였다. 3가지 알고리듬의 비교시 ZS-2와 ZS-3는 평문 블록에 대한 블록 동기를 일치시켜야 하는 어려움이 없기 때문에 하드웨어 구현이 용이하였으며, 3가지 방법 중에서 에러 확산이 개선된 ZS-3 방식이 가장 우수하다.

참 고 문 헌

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. CCITT Rec. G.703 : "Physical/Electrical Characteristics of Hierar-chical Digital Interface", CCITT red book, vol. III, 1985.
5. 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.
6. D. E. Dodds, L. R. Button and S. Pan, "Robust Frame Synchronization for Noisy PCM Systems," IEEE Trans. on Comm., vol. COM-33, no. 5, pp. 465-469, May 1985.
7. R. Maruta, "A Simple Firmware Realization of PCM Framing Systems," IEEE Trans. on Comm., vol. COM-28, no. 8, pp. 1228-1223, Aug. 1980.
8. 이훈재, 박봉주, 장병화, 문상재, 박영호, "T1 전송 시스템 보호를 위한 ZS 동기 알고리 듬," 한국통신정보보호학회 논문지 제7권 제2호, pp. 93-106, 1997년 6월.

www.dbpia.co.kr

이 훈 재(Hon-Jae Lee) 정회원



1985년 2월 : 경북대학교 공과대학 전자공학과(전자공학, 공학사)
1987년 2월 : 경북대학교 대학원 전자공학과(통신공학, 공학석사)
1987년 2월~1998년 1월 : 국방과학연구소 선임연구원
1993년 3월~1998년 2월 : 경북대학교 대학원(정보통신, 공학박사)
1998년 2월~현재 : 경운대학교 컴퓨터공학과(전임강사)
<주관심분야> 정보보호기술, 디지털 통신, 정보통신망

문 상 재(Sang-Jae Moon) 정회원

1972년 2월 : 서울대학교 공과대학 공업교육과(전자공학, 공학사)
1974년 2월 : 서울대학교 대학원 전자공학과(통신공학, 공학석사)
1984년 6월 : 미국 UCLA(통신공학, 공학박사)
1984년 6월~1985년 6월 : UCLA Postdoctor 근무
1984년 6월~1985년 6월 : OMNET 컨설턴트
1974년~현재 : 경북대학교 공과대학 전기전자공학부 교수
<주관심분야> 정보보호, 디지털 통신, 정보통신망

www.dbpia.co.kr