Zero Suppression 알고리듬들의 여러 가지 특성 비교

(Comparisons of Various Properties for Zero Suppression Algorithms)

이 훈 재*, 박 영 호**
(Hoon-Jae Lee , Young-Ho Park)

요 약 동기식 스트림 암호 시스템 적용을 위한 ZS 동기 알고리듬이 다수 제안된 바 있지만, 그 중에서 ZS-1 알고리듬은 스트림 암호 시스템 구현상의 어려움이 있고, ZS-2 알고리듬은 에러 확산이 많으며, ZS-3는 회로가 복잡해지는 단점을 갖는다. 그리고 이들 알고리듬은 적용 분야를 주의하여 선택하여야 할 필요성이 있으며, 본 논문에서는 몇 가지 파라메터 변화에 따른 결과 값의 비교를 통하여 주어진 시스템에 적합한 알고리듬을 선택하는 방향을 제시하였다.

Abstract Among zero-suppression (ZS) algorithms proposed for synchronous stream cipher system, ZS-1 has the difficulty on the implementation of the stream cipher system, ZS-2 has a weakness of channel error propagation, and ZS-3 has the complexity on the hardware. Because each algorithm must be chosen carefully to the application system, in this paper, we propose a criterion of the system adapted from the simulated results on the parameter changed.

1. 서 론

일반적인 통신시스템은 수신 클럭 복구를 원활히 하기 위하여 스크램블러 등의 기법으로 출력 단의 연속 "0" 비트 수를 제한하거나 또는 B8ZS [4] 등과 같은 선로부호화(line coding)를 통하여 수신되는 "0" 레벨/"1"레벨에서의 레벨 변환(level transition)을 강화시키고 있다. 하지만 T1-carrier 시스템 등과 같은 현재 운영 중인 시스템 중에서는 스크램블링 기법으로도 해결이 어려울 뿐 아니라 선로부호화 레벨 변환이 불가능한 시스템이 존재하고 있다. 이러한 시스템 사양에 링크 암호화 형태의 동기식 스트림 암호 시스템을 적용시 송신단 암호문 출력에 과다한 연속 "0" 비트가 발생되는 현상이 나타나며, 이로 인한 시스템 성능 저하 뿐 아니라 시스템 호환성 문제를 야기 시킨다.

* 경운대학교 컴퓨터전자정보공학부(hjlee@kyungwoon.ac.kr),
** 상주대학교 전자전기공학부 (yhpark@sangju.ac.kr)

상기 문제점에 대한 새로운 해법인 ZS 알고리듬은 스트림 암호 시스템의 송신단에서 나타날 수 있는 과다한 연속 "0"의 문제를 스크램블링 기법이나 B8ZS 등과는 달리 암호시스템 자체에서 해결하고 있다. 암호 통신에 적용된 ZS 방식은 블록 검출 방식과 직렬 검출 방식이 있으며, ZS-1 알고리듬 [8]은 블록 검출 방식이고, ZS-2 및 ZS-3 알고리듬 [9]은 직렬 검출 방식이다. 하지만 이들은 여러 가지 기준에 대하여 시스템 적용상 몇가지 문제점을 안고 있다. 특히, 오류확산문제와 블록 동기 추가로 인한 시스템 설계의 복잡성, 비트 지연, 구현 용이성 및 적용 가능한 시스템 등의 여러 특성에 상당한 차이를 보이고 있기 때문에 이들에 대한 성능을 분석하여 시스템에 적합한 알고리듬의 선택할 필요성이 있다.

본 논문에서는 적용 분야에 따라 세 가지 알고리듬을 주의하여 선택할 필요성이 있기 때문에 오류확산, 블록 동기의 필요성, 비트 지연, 구현 용이성, 적용 시스템 등 몇 가지 파라메터를 선정한 후 이들 파라메터 변화에 따른 결과

를 비교하고, 이러한 비교 분석을 통하여 주어진 시스템에 적합한 알고리듬을 선택하는 방향을 제시하고자 한다.

2. ZS 알고리듬들

일반적으로 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 통신에서와 달리 새로운 문제를 유발한다. 이를 해결하여 암호화 후에도 필요에 따라 k(>2) 비트 이하로 연속 "0"을 억제하는 것이 바로 ZS 동기 방식이다.

ZS 알고리듬에 사용될 변수 k는 채널에서 최대로 허용되는 연속 "0" 비트 수이며, 알고리듬에서 처리하는 블록 크기 $n = \lceil (k+1)/2 \rceil$ 이다. 여기에서 $\lceil x \rceil$ 는 x를 넘지 않는 최대 정수를 말한다. ZS 알고리듬은 출력단에서의 연속 "0"을 검출하기 위한 검출부 (detection part)와 검출된 연속 "0" 값 (블록 크기, n)을 다른 값으로 대체시키는 대체부 (substitution part)로 나누어진다. 문제는 연속 "0"이 검출되었을 때 이를 대체시킬 새로운 값을 송신부와 수신부에서 통신 redundancy 비트의 추가 없이 어떻게 동일한 값으로 설정할 것인가에 달려있다. 일반적으로 생각할 수 있는 방법은 특정 패턴을 이용하는 방법이지만, 이 경우에는 송신문에서 특정패턴에 대한 제약성이 생기게 되고, 특히 약정된 패턴이 암호문에서 자연 발생될 가능성 때문에 방법상의 문제점이 노출된다. ZS 알고리듬에서의 아이디어는 '송수신단에서 서로 reference 될 수 있는 특정 패턴을 어떻게 결정하느냐?'에 달려있으며, 제안된 ZS 알고리듬 들에서는 연속 "0"이 발생된 시점의 키 수열 블록을 활용하는 방법만이 유일한 해법임이 밝혀진 바 있다[9].

암호문 출력단의 연속 "0"을 검사하는 부분을 검출부라 하는데, 이는 연속 "0"에 대한 검사가 블록 단위로 이동하면서 수행되는 블록 검출방식 (block detection)과 같은 블록 단위이지만 1 비트씩 직렬로 이동하면서 조사되는 직렬 검출방식 (serial detection)으로 대별 된다. 대체부는 연속 "0"이 검출되었을 때 다른 블록을 대체하는 것이며, 블록 크기의 정수배만큼 대체되는 블록 대체와 일부분만 대체되는 부분 대체로 나눌 수 있다. ZS-1 알고리듬은 블록 검출/블록 대체 방식이고, ZS-2 알고리듬은 직렬 검출/블록 대체방식이며, ZS-3 알고리듬은 직렬 검출/부분 대체 방식이다.

2.1 블록 검출 알고리듬

블록 검출의 편의상 모든 벡터들을 블록 단위로 다음과 같이 나눌 수 있다. i번째 n 비트 평문벡터 $\mathbf{P}_i$, i번째 n 비트 난수열 블록 $\mathbf{K}_i$, i번째 n 비트 암호문 블록 $\mathbf{C}_i$, i번째 n 비트 복호평문블록 $\mathbf{Q}_i$라 하고 다음과 같이 정의한다. 그리고 n 비트 0 벡터 $\mathbf{0}$, 블록크기 $n = \lceil (k+1)/2 \rceil$ 이고, $\lceil x \rceil$ 는 x를 넘지않는 최대 정수이다.

- $\mathbf{P}_i$: $(p_{in}, p_{in+1}, \cdots, p_{in+n-1})$
- $\mathbf{K}_i$: $(k_{in}, k_{in+1}, \cdots, k_{in+n-1})$
- $\mathbf{C}_i$: $(c_{in}, c_{in+1}, \cdots, c_{in+n-1})$
- $\mathbf{Q}_i$: $(q_{in}, q_{in+1}, \cdots, q_{in+n-1})$

ZS-1 알고리듬 [8]을 위해서 다음과 같은 가정을 설정한다.

(1) 송신단 암호 시스템에서 잉여비트를 삽입 또는 삭제할 수 없다.

(2) 모든 i($i \geq 0$)에 대하여 $\mathbf{P}_i \neq \mathbf{0}$이다.

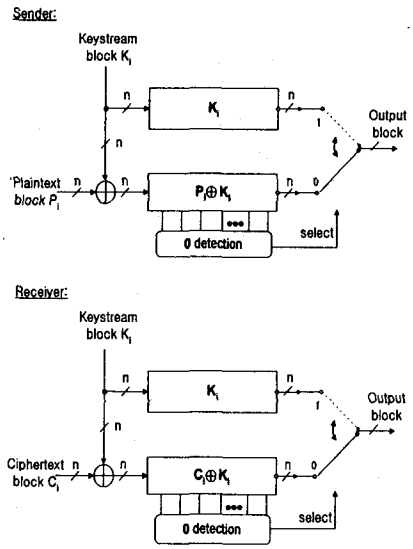
(3) 난수열 발생기는 높은 비도를 갖는다.

상기의 가정하에 송신단에서의 ZS-1 알고리듬 동작은 다음과 같다.

(1) $\mathbf{P}_i \oplus \mathbf{K}_i$ 연산된 암호문블록과 $\mathbf{K}_i$를 각각 n단 이동레지스터에 입력시킨다.

(2) $\mathbf{P}_i \oplus \mathbf{K}_i$ 연산된 암호문블록이 $\mathbf{0}$인지 검사한다.

(3) $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$ 일 경우에는 $\mathbf{C}_i = \mathbf{K}_i$를 출력시킨다. 이 외의 경우에는 $\mathbf{C}_i = \mathbf{P}_i \oplus \mathbf{K}_i$를 출력시킨다.



<그림 1>. ZS-1 알고리듬

<Fig. 1>. Zero suppression-1 algorithm.

또한 수신단에서의 동작은 다음과 같다.

(1) $C_i \oplus K_i$ 연산된 복호문블록과 K_i를 각각 n단 이동레지스터에 입력시킨다.

(2) $C_i \oplus K_i$ 연산된 복호문블록이 0인지 검사한다.

(3) $C_i \oplus K_i = 0$일 경우에는 $Q_i = K_i$를 출력시킨다. 이 외의 경우에는 $Q_i = C_i \oplus K_i$를 출력시킨다.

대체방법에 대하여 좀 더 깊이 고찰하면, 오직 키 수열 블록의 대체만이 본 알고리즘을 완전하게 구성할 수 있음을 알 수 있다. 즉, 출력단에서 $P_i \oplus K_i = 0$의 검출시 임의 벡터 $C_i = R(\neq K_i)$를 대체할 경우 수신단에서는 R을 검출해서 역대체해야 하는데, 이 과정에서 암호문 벡터 $R(=P_i \oplus K_i)$인지 대체된 값 $C_i = R$인지 구별이 불가능해진다.

2.2 직렬 검출 알고리듬

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 P_i :

$$(p_i, p_{i-1}, \cdots, p_{i-n+1})$$

- i번째 키 수열 블록 K_i :

$$(k_i, k_{i-1}, \cdots, k_{i-n+1})$$

- i번째 암호문 블록 C_i :

$$(c_i, c_{i-1}, \cdots, c_{i-n+1})$$

- i번째 복호 평문 블록 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.

ZS-2 알고리듬 [9]은 다음과 같다. (그림 2 참조)

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

2) P_i블록과 $P_i \oplus K_i$블록이 각각 0인지 검사한다.

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

$P_i \neq 0$, $P_i \oplus K_i = 0$인 경우($P_i = K_i$) : $C_i = P_i$의 n비트 블록을 대체 출력시킨다.

$P_i = 0$ 경우($P_i \oplus K_i$와 무관): $C_{i-n} = P_{i-n}$, $C_i = P_i$, $C_{i+n} = P_{i+n}$ 연속 3블록 3n 비트를 대체 출력시킨다.

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

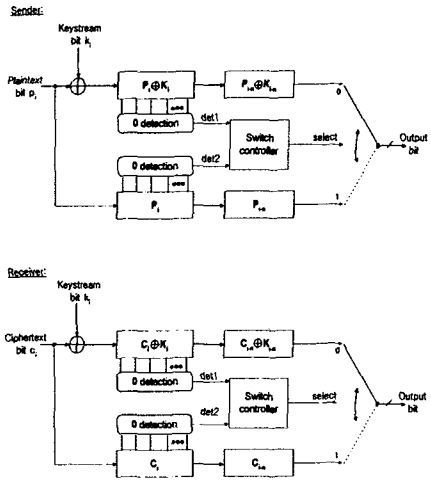
2) C_i블록과 $C_i \oplus K_i$블록이 각각 0인지 검사한다.

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

$C_i \neq 0$, $C_i \oplus K_i = 0$인 경우($C_i = K_i$): $Q_i = C_i$의 n비트 블록을 대체 출력시킨다.

$C_i = 0$ 경우($C_i \oplus K_i$와 무관): $Q_{i-n} = C_{i-n}$, $Q_i = C_i$, $Q_{i+n} = C_{i+n}$ 연속 3블록(3n 비트)를 대체 출력시킨다.

상기 알고리듬에서 p_i와 P_i는 구분되어야 하는데 p_i는 1 비트 평문 비트를 말하며, P_i는 n 비트의 평문 벡터 $(p_i, p_{i-1}, \cdots, p_{i-n+1})$를 의미한다.



※ Switch controller :
if det2=1, select SW=1 and output $3n$ bits
else if det1=1, select SW=1 and output n bit
else, select SW=0 and output 1 bit

<그림 2>. ZS-2 알고리듬
<Fig. 2> ZS-2 algorithm

ZS-2 알고리듬은 블록 대체된 부분에 채널오류가 발생되면 오류가 확산되는 단점이 있지만, $\mathbf{P}_i \neq \mathbf{0}$의 가정이 필요없어 T1급 PCM회선등에 적용시 블록동기를 일치시켜야 하는 하드웨어 부담이 감소되므로 실현이 용이하다. 그러나 블록 대체된 부분에 채널 오류가 발생되면 오류가 확산되는 문제점을 안고 있다.

ZS-3 알고리즘 [9]은 <그림 2>와 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 방식이며, 다음과 같다. (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 $\mathbf{0}$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.

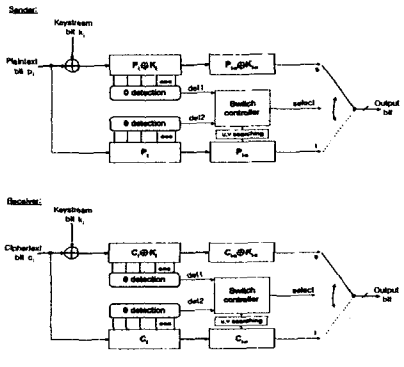
$\mathbf{P}_i \neq \mathbf{0}$, $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$인 경우($\mathbf{P}_i = \mathbf{K}_i$) : $\mathbf{C}_i = \mathbf{P}_i$의 n비트 블록을 대체 출력시킨다.

$\mathbf{P}_i = \mathbf{0}$ 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 $n+u+v$ 비트를 대체 출력시킨다.
여기서, $\mathbf{P'}_{i-n}$ 블록이란 <그림 3>과 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 $\mathbf{0}$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 $n+u+v$ 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.



※ 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

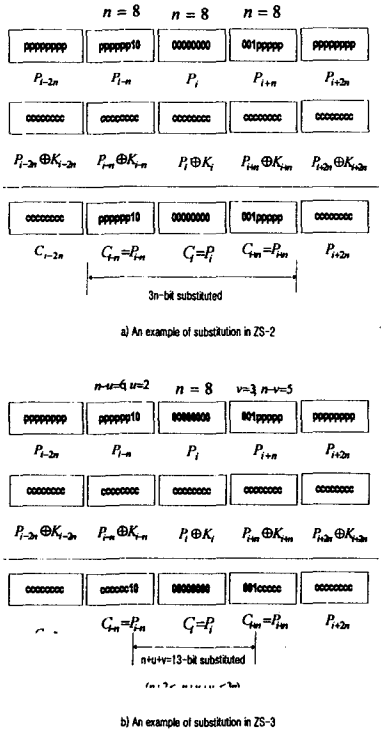
※ u, v : refer to fig. 4.

<그림 3>. ZS-3 알고리듬
<Fig. 3>. ZS-3 algorithm.

3. 성능 비교 분석

채널에서의 비트 오류율(channel bit error rate)을 B라 둘 때 각각의 ZS 알고리듬에 대한 통신망에서의 전체 비트 오류율은 다음과 같다 [8-9].

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



<그림 4>. 대체방법의 비교 예(n=8,u=2,v=3)
< Fig. 4>. Examples of comparison of the substitution (n=8,u=2,v=3)

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

$$P_E(ZS\text{-}3) = [(n/2)+1][2^{-(n-2)}][1-(1-B)^n] + B \qquad (3)$$

$B=10^{-5}$이라는 가정 하에 ZS 알고리즘의 전체 비트 오류율을 n에 따라 시뮬레이션한 결과는 <표 1>과 같다. 3가지 모두 전체 비트 오류율은 n에 따라 단조 감소되어 채널 비트 오류율(BER)에 접근되므로 n값 (또는 k값)을 크게 선택할수록 좋은 오류 특성을 얻을 수 있다. 특별히 k=15, n=8인 T1 전송 시스템에서 전체 비트 오류율은 표 2와 같아지며, 동일한 조건으로 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.

ZS 알고리즘의 적용이 필요한 통신망에서는 여러 가지 통신 특성에 따라 ZS 알고리즘이 선택되어지겠지만, 여기서는 3가지 ZS 알고리즘의 특성을 <표 3>과 같이 비교 검토하였다. 검토 결과 블록 검출 방식인 ZS-1은 블록 동기 필요성으로 인하여 하드웨어적인 복잡성이 증대되기 때문에 소프트웨어적인 구현이 용이하였으며, 이를 적용할 수 있는 시스템으로는 Eurocom D/1 15ch/30ch CVSD multipler 시스템 등을 들 수 있다. 그리고 직렬 검출 방식인 ZS-2는 블록 동기가 불필요하고 하드웨어적인 구현이 용이한 반면, ZS-1의 경우보다는 채널 오류 특성에 취약하며, ZS-3는 이러한 문제점을 보완하여 설계되었기 때문에 암호 시스템 출력에서의 제로 특성을 완화시키는데 좋은 해결책이라고 볼 수 있다. 이들 시스템은 북미형 T-1 carrier 시스템이나 유럽형 E-1 carrier 시스템에 적용될 수 있다.

4. 결 론

본 논문에서는 적용 분야에 따라 이들 알고리듬을 주의하여 선택할 필요성이 있기 때문에 블록 동기, 비트 지연, 오류 확산 및 암호비도수준 저하 등의 몇 가지 파라메터 변화에 따른 결과 분석을 통하여 주어진 시스템에 적합한 알고리듬을 선택하는 방향을 제시하였다. 분석 결과 ZS-2를 개선시킨 ZS-3에서는 대체 블록의 비트 수를 최

<표 1>. n 가변에 따른 ZS 알고리즘들의 비트 오류율 (BER=10-5)

※ 비트 오류 개선율(%)는 ZS-2에 대한 ZS-3의 개선 비율임.

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

<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 x 10^{-1}	1.7119160 x 10^{-1}	1.4449475 x 10^{-1}
10^{-2}	1.2414228 x 10^{-2}	1.9656913 x 10^{-2}	1.6035570 x 10^{-2}
10^{-3}	1.2491268 x 10^{-3}	1.9965071 x 10^{-3}	1.6228169 x 10^{-3}
10^{-4}	1.2499125 x 10^{-4}	1.9996500 x 10^{-4}	1.6247813 x 10^{-4}
10^{-5}	1.2499912 x 10^{-5}	1.9999650 x 10^{-5}	1.6249781 x 10^{-5}
10^{-6}	1.2499991 x 10^{-6}	1.9999965 x 10^{-6}	1.6249978 x 10^{-6}
10^{-7}	1.2499999 x 10^{-7}	1.9999997 x 10^{-7}	1.6249998 x 10^{-7}
10^{-8}	1.2500000 x 10^{-8}	2.0000000 x 10^{-8}	1.6250000 x 10^{-8}
10^{-9}	1.2500000 x 10^{-9}	1.9999999 x 10^{-9}	1.6250000 x 10^{-9}
10^{-10}	1.2500000 x 10^{-10}	2.0000000 x 10^{-10}	1.6250000 x 10^{-10}

<표 3> ZS 알고리즘 특성 비교

※ $\lceil x \rceil$: x를 넘지 않는 최대 정수

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 필요성	반드시 필요함	불필요함	불필요함
비트 지연 크기	n 비트	$2n$ 비트	$2n$ 비트
전체 비트 오류 확산	확산 (소) : $(n)2^{-n}[1-(1-B)^n] + B$	확산 (대) : $(n)2^{-(n-2)}[1-(1-B)^n] + B$	확산 (중) : $[(n/2)+1][2^{-(n-2)}][1-(1-B)^n] + B$
비도 수준	암호 알고리즘에 의하여 결정되며 본 알고리즘과는 무관함	암호 알고리즘에 의하여 결정되며 본 알고리즘과는 무관함	암호 알고리즘에 의하여 결정되며 본 알고리즘과는 무관함
n 선택	$n = \lceil (k+1)/2 \rceil$	$n = \lceil (k+1)/2 \rceil$	$n = \lceil (k+1)/2 \rceil$
적합한 구현방법	소프트웨어	하드웨어	하드웨어
적용 용이한 시스템	Eurocom D/1 15ch/30ch CVSD multipler 시스템	- T-1 carrier 시스템 - E-1 carrier 시스템 - Eurocom D/1 15ch/30ch CVSD multipler 시스템	

최소화시킴으로서 에러 확산을 줄일 수 있었으나, n 값이 15 이상일 때 에러 확산의 단점은 3가지 모두 무시될 정도로 낮았다. 또한 비도 요소 저하에 있어서는 3가지 알고리듬이 거의 무관함을 알 수 있었고, 블록 동기 측면에서는 ZS-1 알고리듬에서만 블록동기가 요구되고 ZS-2와 ZS-3에서는 평문 블록에 대한 블록 동기가 불필요하기 때문에 통신망에서 구현시에 ZS-1 방법은 소프트웨어 구현에 유리하며, 나머지 방법들은 하드웨어 구현이 유리함을 확인하였다.

참고 문헌

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

[2] Henk C.A. van Tilborg, Fundamentals of 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 Hierarchical 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] Hoonjae Lee and Sangjae Moon, "On ZS Synchronization Algorithm for Synchronous Stream Cipher," Applied Signal Processing (London) Vol. 5, No.4, pp.240-243, Dec. 1998.

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

이 훈 재 (Hoon-Jae Lee)
1985년 2월 : 경북대학교 전자공학과 졸업(학사)
1987년 2월 : 경북대학교 전자공학과 졸업(석사)
1998년 2월 : 경북대학교 전자공학과 졸업(박사)
1987년 2월~1998년 1월 : 국방과학연구소 선임연구원
1998년 2월~현재 : 경운대학교 컴퓨터전자정보공학부 조교수
<주관심분야> 암호이론, 네트워크보안, 디지털 통신

박 영 호 (Young-Ho Park)
1989년 2월 경북대학교 공과대학 전자공학과 (공학사)
1991년 2월 경북대학교 대학원 전자공학과 (공학석사)
1995년 2월 경북대학교 대학원 전자공학과 (공학박사)
1996년 3월 - 현재 상주대학교 전자전기공학부 조교수
<주관심분야> 정보보호, 컴퓨터 네트워크