通信情報保護學會誌
第2卷 第3號, 1992. 9

암호기와 부호기의 적정배열에 관한 연구†

A Study on the Optimal Arrangement of Encryptor and Encoder

박창섭* · 이수연*

1. 개 요

디지탈 통신채널을 통한 메시지의 효율적인 송수신에 장애가 되는 요인은 통신채널상에 존재하는 잡음(noise)으로 인한 메시지의 손상과 불법적인 메시지의 도청에 의한 귀중한 데이타의 노출이다. 채널잡음으로부터 메시지의 신뢰성(reliability)을 증대시키기 위해서 오류정정부호(error-correcting code)를 이용한 부호기를 통해 메시지를 부호화시키고, 메시지의 보안성(security)을 위해 DES와 같은 블럭 암호체계를 통해 메시지를 암호화시키고 있다. 특히, 채널잡음의 존재하에서 암호문의 정확한 decryption을 위해서는 오류정정부호의 사용은 필수적이다. 암호문의 1비트에 가해진 채널잡음의 효과는 decryption시, DES와 같은 블럭 암호체계가 지니는 확산(diffusion) 성질에 의한 오류의 파급(error propagation)으로 무작위(random)한 변화를 평서문에 야기시킨다.

본 연구에서는 메시지의 신뢰성 및 보안성이 동시에 요구되어지는 상황에서 부호기(encoder)와 암호기(encryptor)의 배열순서 변화에 따른 채널잡음의 효과를 복호기(decoder)의 복호화 오류율(decoding error rate) 및 메시지 오류율 그리고 정보율(information rate)의 측면에서 비교해 보고 능동적인 공격에 의한 인위적인 메시지의 변조 및 삽입의 검출 가능성을 부호화 과정의 블럭형성(blocking) 방식을 통해 타진해 본다.

분석의 도구로는 암호화를 위해 DES와 같은 블럭 암호체계를 CBC(cipher block chaining) 방식으로 운영하고, 부호화에는 선형부호(linear code)를 이용한다. 오류수정 방법은 수신측에서 오류의 검출 및 수정이 행하여지는 전방오류수정(forward error correction) 방식을 택한다. 분석의 범위를 제한하는 의미에서 다음을 가정한다. 첫째, 부호화에 사용되는 부호는 비밀사항이 아니다. 둘째, 암호화 과정에서 데이타의 팽창은 없고 세째, 사용 암호체계는 안전하다고 가정한다.

2절에서는 암호화 이후에 부호화가 이루어지는 경우, 3절에서는 암호화 이전에 부호화가 이루어지는 경우, 마지막으로 4절에서는 암호화 이전과 이후에 각각 부호화가 동시에 이루어지는 다단계 부호화를 살펴본다.

† 본 연구는 한국전자통신연구소의 지원금에 의한 연구결과임
* 단국대학교 전자계산학과

2. 암호화후 부호화

블럭형성 방식

그림 1. 암호화후 부호화

그림 1에서 보는 바와 같이, 이 경우는 암호화가 이루어진 이후에 부호화가 이루어진다. m개의 k비트의 평서문을 DES와 같은 블럭암호체계를 통해 CBC(cipher block chaining) 방식에 의해 암호화 시켜 m개의 k비트의 암호문을 생성한다. 물론, 암호화과정에 있어서 데이타의 팽창은 없는 것으로 가정한다. 다시 각각의 k비트의 암호문이 [n, k, $d \geq 2t+1$] 선형부호(linear code)에 의해 부호화(encoding) 되어 m개의 n비트의 코드워드(codeword)를 만든다($n=k+r$).

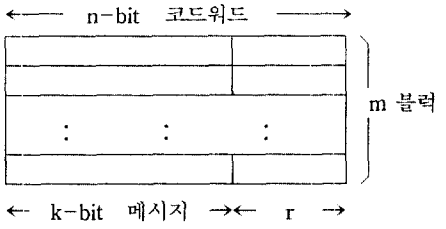


그림 2. 암호화후 부호화 블럭형성방식

신뢰성 분석

채널오류가 직접 암호문에 발생하는 것이 아니기 때문에 사용암호체계의 확산(diffusion) 성질에 의한 오류의 파급효과는 초래하지 않는다. 단지, 복호화(decoding) 과정에서 검출되지 못한 채널오류만이 그러한 변화를 야기 시킨다. 결국, 발생된 채널오류가 복호기에 걸리지 않을 확률은 복호화 오류율(decoding error rate)에 의해 표현되어지는데 p를 채널비트 오류율이라 할 때 한개의 코드워드에 대한 복호화 오류율 P_{de}는,

$$P_{de}=1-\sum_{j=0}^{t}\binom{n}{j}\cdot p^j\cdot(1-p)^{n-j} \qquad (1)$$

m개의 블럭에 대한 복호화 오류율, 즉 메시지 오류율 P_{me}는,

$$P_{me}=1-(1-P_{de})^m \qquad (2)$$

암호화후 부호화의 신뢰성(reliability) 분석을 위해서 Hamming Bound를 이용해 n, k, 그리고 r(=n−k)의 값을 선정하였다. Hamming Bound는 t개의 오류를 수정하기 위해서는 패러티 검사 비트의 갯수가 적어도 r개 이상 있어야 한다는 것을 의미하는 하계(lower bound)의 개념이다.

$$\text{Hamming Bound} \qquad 2^{n-k}\geq\sum_{j=0}^{t}\binom{n}{j} \qquad (3)$$

그림 3은 채널 비트 오류율(channel bit error rate) p가 10^{-4}, 사용 부호의 최소거리(minimum distance)가 3, 그리고 메시지의 길이 $M=m\cdot k$이 1024 비트일 때, k의 길이를 16, 32, 그리고 64로 변화시킴에 따른 메시지 오류율과 정보율(information rate)의 변화를 나타내고 있다. 그림에서 보는 바와 같이 코드워드의 길이가 길어짐에 따라 정보율은 증가하지만, 메시지 오류율은 적은 양이지만 감소하는 추세를 보인다. p의 값이 10^{-5}, 10^{-6}, 10^{-7}, 그리고 10^{-8}에서도 같은 현상이 나타난다. 결국, 최소거리가 일정할 때 코드워드의 길이가 긴 부호를 사용함에 따라서 메시지의 신뢰성은 증가되고 정보율 역시 향상된다.

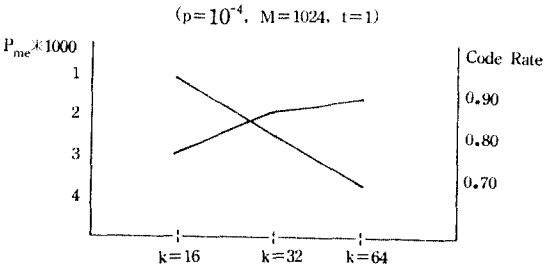


그림 3. k값에 따른 정보율 및 메시지 오류율

그림 4는 M=1024, k=64인 경우, 사용 부호의 최소 거리를 3에서 5로 증가 시켰을 때 메시지 오류율을 비교한 것이다. 정보율은 0.90에서 0.84로 하락하지만 당연히 메시지 오류율은 감소 한다.

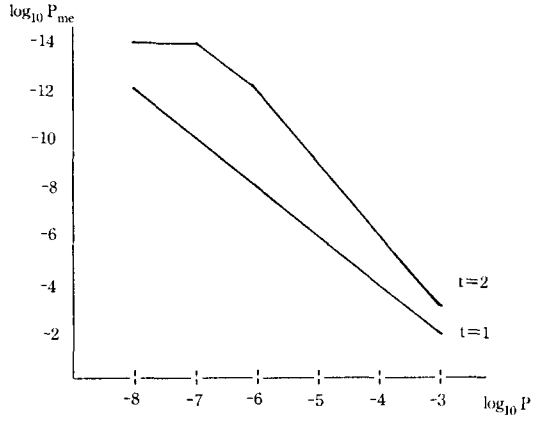


그림 4. channel BER에 따른 메시지오류율

메시지의 인증

암호화후 부호화의 경우, 부호화 과정은 비밀사항이 아니기 때문에 수신측의 복호화 과정에서 검출되지 않는 능동적인 메시지의 삽입 및 변조 공격의 가능성이 존재한다. 결국은 CBC(cipher block chaining)으로 운영되는 암호화 과정에서 메시지 인증기능이 요구되어지는데, MDC(manipulation detection code)가 보편적으로 이용되어지고 있다.

한 개 이상의 메시지 블럭을 암호화 시킬 경우 ECB(electronic codebook) 분석에 대처하기 위하여 CBC(cipher block chaining) 방식이 일반적으로 사용되어지고 있다. 하지만 평서문에 어떠한 redundancy도 존재하지 않을 경우, 암호문에 오류가 발생했는지 아닌지를 정당한 수신자가 판단하기 위해서 MDC(manipulation detection code)를 마지막 블럭으로 부착하여 송신되어진다. 즉, $E_k(P_i+C_{i-1})=C_i$, $0\leq i\leq n$에서 P_i는 평서문, C_i는 암호문, 이때 P_0=IV(initialization vector), P_n이 MDC를 포함한다. MDC는 평서문을 k비트씩 끊어서 exclusive-or한 값이다. 결국, MDC는 m개의 단순 패러티 검사부호(simple parity-check code)를 사용하여 얻게 된다.

MDC 방식은 능동적 공격에 의한 2개의 동일한 메시지의 삽입시 서로 상쇄되어 MDC에 의해 검출되어지지 않는다. 이것은 MDC의 생성방식이 단순 exclusive-or 이어서 선형(linearity)의 성질을 가지기 때문이다.

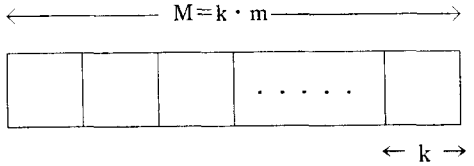


그림 5. MDC 블럭형성방식

P_n이 MDC일 경우 ($P_1+P_2+\cdots P_{n-1}=P_n$), 2개의 같은 메시지 C_x, C_x가 C_i와 C_{i+1} 사이에 삽입되었을 때, 수신측에서는 다음과 같은 decryption 과정이 행해진다.

$$
\begin{aligned}
&\vdots\\
&D(C_i) + C_{i-1} = P_i\\
&D(C_x) + C_i = P_{x1} \longrightarrow ①\\
&D(C_x) + C_x = P_{x2} \longrightarrow ②\\
&D(C_{i+1}) + C_x = P_{x3} \longrightarrow ③\\
&D(C_{i+2}) + C_{i+1} = P_{i+2}\\
&\vdots
\end{aligned}
$$

위의 경우에, ①의 $D(C_x)$와 ②의 C_x가, 그리고 ②의 $D(Cx)$와 ③의 C_x가 함께 서로 상쇄되어진다. 결국은, MDC에 의해서는 그러한 유형의 메시지의 삽입이 검출되어 지지 않는다.

이러한 메시지의 삽입을 수신측에서 검출하기 위한 방안으로 PBC(plaintext block chaining)방식을 새로이 제안한다. 그림 6에서 처럼 운영방식은 다음과 같다.

$$C_i=E_k(P_{i-1}+P_i),\quad 1\leq i\leq n \tag{4}$$
$$P_i=D_k(C_i)+P_{i-1} \tag{5}$$

이때, $P_0=P_n$=IV(initialization vector)에 해당하고 첫째 블럭 C_1과 마지막 블럭 C_n은 다음과 같다.

$$C_1=E_k(IV+P_1) \tag{6}$$
$$C_n=E_k(P_{n-1}+IV) \tag{7}$$

C_n을 decryption할 경우, 얻어지는 것은 $D_k(C_n)+P_{n-1}$=IV이다. CBC의 경우에는 오류의 파급이 오류가 발생한 그 다음 블럭까지만 제한적으로 영향을 미치지만 PBC에서는 체인방식에 암호문 대신 평서

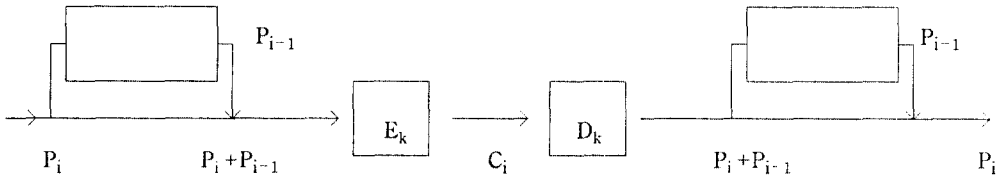


그림 6. PBC 운영방식

문이 사용되어지기 때문에 오류의 파급효과가 마지막 블럭까지 확산되어진다. 마지막 블럭의 decryption 과정에서 기대되어지는 평서문은 항상 IV=P_n이기 때문에 IV와의 해밍거리(Hamming distance)가 결국은 채널오류의 영향을 받은 암호문의 확산(diffusion)된 양을 표시할 뿐만 아니라, 해밍거리가 0이 아니라면 자동적으로 오류 또는 변조를 검출할 수도 있게 된다. 평서문간에 체인이 형성되어지기 때문에 보안적인 측면 역시 CBC와 마찬가지로 electronic codebook 공격에 대처할 수 있다. 특히, MDC 방식에서는 검출할 수 없었던 불법적인 메시지의 삽입은 쉽게 마지막 decryption 과정에서 검출 되어진다.

3. 암호화전 부호화

오류의 파급효과

전방 오류 수정(forward error correction)을 수행하는 암호화전 부호화의 경우에는 다음과 같은 문제점이 존재한다. 그림 7에서와 같이 부호화 이후에 암호화가 이루어지기 때문에 채널 오류의 영향을 받은 암호문의 decryption 결과는 사용 암호체계가 지니는 확산성질에 의한 오류의 파급(error propagation)으로 무작위의 변화를 평서문에 야기시킨다.

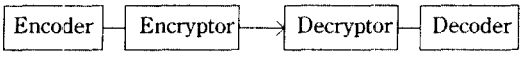


그림 7. 암호화후 부호화

그림 8은 64비트 평서문을 DES 프로그램을 이용하여 암호화 시킨후, 암호문의 서로 다른 위치에 1비트씩 변화를 가한 후에 decryption 한 결과를 원래의 평서문과 해밍거리를 비교한 도표이다. 오류의 파급효과로 인하여 거의 50% 정도의 차이를 나타내고 있다.

결국, 복호화(decoding) 과정에서는 한개의 코드워드에 약 50% 정도의 오류가 발생한 것을 수정해야 하는데 Singleten bound에 의하면 그러한 오류수정능력을 지니는 부호는 존재하지 않는다.

KEY : 3 1 3 2 3 3 3 4 3 5 3 6 3 7 3 8

	암 호 문	평 서 문	해밍거리
	94D4436BC3B5B693	6162636465666768	
1.	14D4436BC3B5B693	57F03980BA9B2483	38
2.	54D4436BC3B5B693	459A08BAF4E7E8AC	31
3.	74D4436BC3B5B693	C000DDE7F6501BEF	32
4.	64D4436BC3B5B693	B5789A32AEE3DDB9	35
5.	6CD4436BC3B5B693	FE01EACA6DB6C9CC	30
6.	68D4436BC3B5B693	A69D89A480239D30	37
7.	6AD4436BC3B5B693	17F2958347BF0313	35
8.	68D4436BC3B5B693	46CA09D51C2C30C0	31
평균 해밍거리			33.5

그림 8. 암호문 1비트 변화에 따른 평서문간의 해밍거리

Singleton Bound : $d \le n-k+1$ (8)

$d \ge 2 \cdot t+1$ (9)

$t \le \lfloor (n-k)/2 \rfloor$ (10)

d : 최소거리, n : 코드워드 길이,
k : 메시지 길이, t : 오류갯수

더구나, CBC(cipher block chaining) 방식에서는 채널오류의 영향을 받은 그 다음 블럭까지도 오류의 효과가 파급되어진다. 이러한 문제점을 우회하기 위해서 즉, 오류의 파급효과를 여러 코드워드에

분산시켜주기 위해서는 burst-error 수정에 이용되는 code interleaving 방식을 사용해야 한다.

블럭형성 방식(code interleaving)

Code interleaving을 사용할 경우에는 실제로 암호화 과정에서 CBC 방식을 사용할 필요가 없다. 체인방식(chaining)의 효과가 code interleaving에 의해 대체되어진다.

M(=k · m) 비트의 메시지가 먼저 m개의 k비트의 메시지로 나누어지고, 각각의 k 비트의 메시지에 대하여 [n, k, d ≥ 2t+1] 부호에 의해 부호화(encoding)가 행하여져 그림 9에서와 같이 m개의 n비트의 코드워드를 생성한다. 이때 암호문의 길이를 k라고 할 때, 다음의 제약조건이 주어진다.

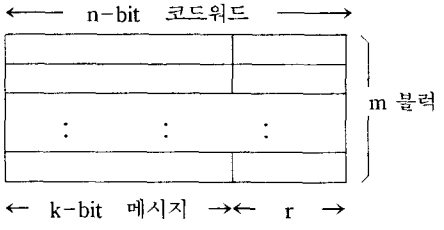


그림 9. 암호화전 부호화의 초기 블럭형성방식

$$m \geq k \qquad (11)$$

m개의 코드워드에 대해 암호화가 이루어지기 전에 m · n 비트를 다음과 같이 interleaving하여 재배치한다. 그림 10에서와 같이 왼쪽 그림의 첫째 행(column) 부터 k 비트씩, 순서대로 오른쪽 그림의 첫째 열(row), 그리고 둘째 열로 채워진다. 채워지지 않는 마지막 열은 0으로 채운다. 이때,

$$m \cdot n = k \cdot w \qquad (12)$$

interleaving에 의해 새로이 배열된 블럭들이 한개의 블럭(row)씩, 즉 k 비트씩, 암호화 되어진다. code interleaving을 사용할 경우에는 CBC 방식으로 운영할 필요가 없다. ECB 방식으로 운영을 해도 각각의 암호문은 메시지의 순서대로 암호화 되어진 것이 아니라, m개의 코드워드 중에서 각각의 코드워드의 특정 위치의 비트들을 모아서 형성되어지기 때문에, 두 개의 암호문 블럭이 동일하다 할지라도 ECB 분석은 불가능하다.

m ≥ k인 이유는 한개의 암호문에 발생한 오류의 효과를 모든 코드워드에 균등히 분산시키기 위해서이다.

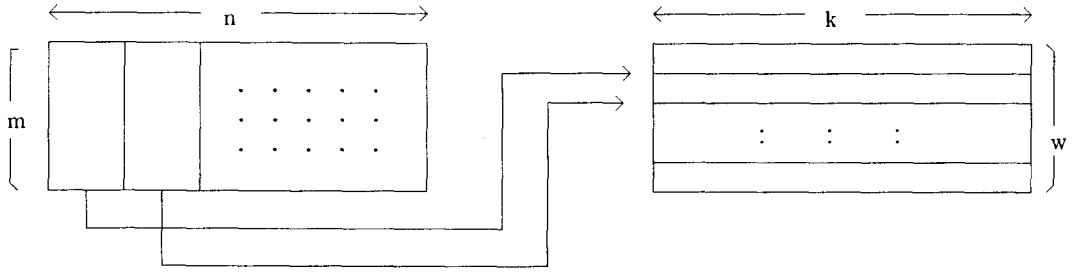


그림 10. 암호화전 부호화의 code interleaving 방식

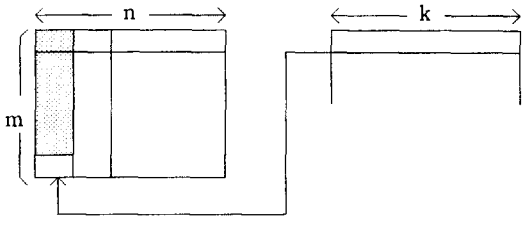


그림 11-1. 암호화된 부호화의 복호과정(m ≥ k)

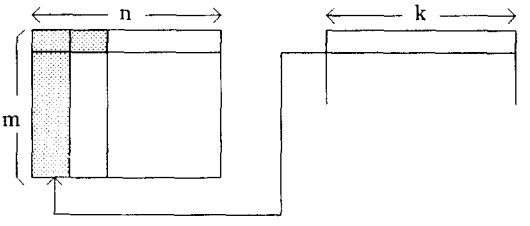


그림 11-2. 암호화된 부호화의 복호과정(m < k)

그림 11-1에서 보는 바와 같이 만약, 오른쪽의 첫째 열(row)이 오류의 영향을 받았을 때, decryption 후에 무작위의 변화를 해당 평서문에 야기시킨다. Code interleaving의 반대작업이 행해진 이후에는 그림 11-1의 왼쪽과 같이 배열 되어진다. 이때, 복호화는 열(row) 단위로 행해지기 때문에 각각의 열 즉, 코드워드에 대하여 단 한개의 오류가 발생한 것을 수정하기만 하면된다. 하지만 m<k이면 그림 11-2와 같이 그 이상의 오류를 수정해야만 한다.

신뢰성 분석

t개의 오류를 수정할 수 있는 부호를 사용할 경우 interleaving 되어진 w개의 암호문 블럭 중에서 t개의 암호문 블럭에 발생한 오류를 수정할 수 있다. 메시지 오류율 P_{me}는,

$$P_{me}=1-\sum_{j=0}^{t}\binom{n}{j}\cdot(1-(1-p)^m)^j\cdot((1-p)^m)^{n-j} \quad (13)$$

암호화후 부호화의 경우와 마찬가지로 해밍 계(Hamming bound)를 이용하여 n, k 그리고 r의 값을 선정한다. 그림 12는 M=1024, n=38, k=32, m=32, t=1일때, 그림 13은 t=2일때의 암호화전 부호화와 암호화후 부호화의 메시지 오류율을 비교한 것이다. 어떠한 경우에도 암호화후 부호화가 더 높은 신뢰성을 보여준다.

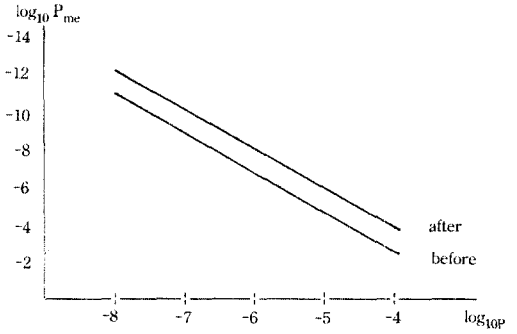


그림 12. 암호화전 및 암호화후 부호화의 메시지 오류율, t=1.

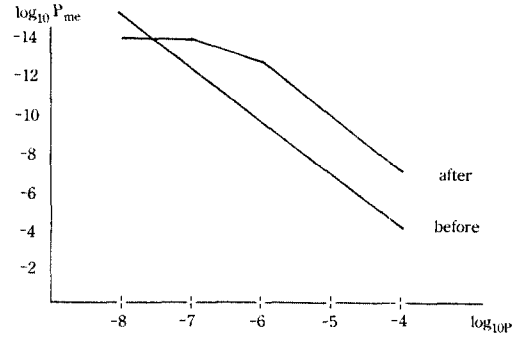


그림 13. 암호화전 및 암호화후 부호화의 메시지 오류율, t=2.

4. 다단계 부호화

블럭형성 방식

그림 14에서 보는 것처럼, 다단계 부호화는 암호화전 부호화와 암호화후 부호화를 통합한 것이라 볼 수 있다.

그림 15에서 처럼 $[n_1, k_1]$ 부호를 이용한 1단계 부호화를 행하고, code interleaving에 의해 w개의 k_2 비트 블럭을 형성한 다음, k_2 비트씩 암호화를 시킨다. $[n_2, k_2]$ 부호를 이용한 2단계 부호화가 마지막으로 행해진다.

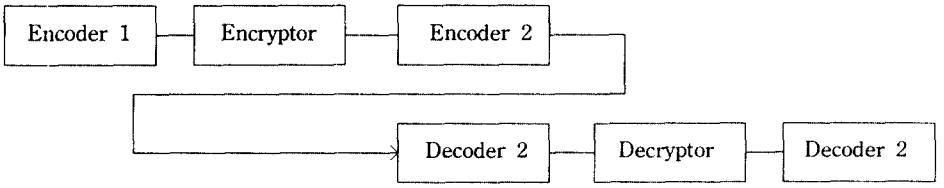


그림 14. 다단계 부호화

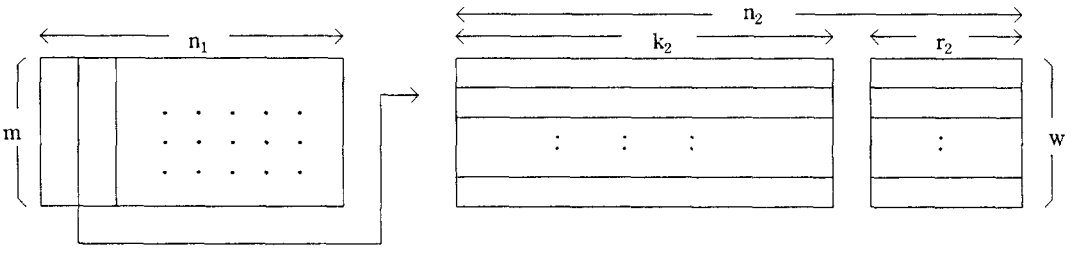


그림 15. 다단계 부호화의 블럭형성방식

신뢰성 분석

다단계 부호화의 메시지 오류율을 계산하기 위해서는 출력비트 오류율(output bit error rate) p′의 개념을 도입해야 한다. 복호기 2를 빠져나온 오류가 복호기 1에서 다시 검출, 수정되기 때문에 복호기 2의 메시지 오류율은 오류율 p를 이용해 계산되지만, 복호기 1의 메시지 오류율은 복호기 2를 빠져나온 오류의 확률 즉, 출력비트 오류율 p′를 이용해 계산되어진다.

복호기 1에는 $[n_1, k_1, d_1 \geq 2 \cdot t_1+1]$ 선형부호가 사용되고, 부호기 2에는 $[n_2, k_2, d_2 \geq 2 \cdot t_2+1]$ 선형부호가 사용되어질때, 복호기 2의 한개의 코드워드 블럭에 대한 복호화 오류율 P_{de}는

$$P_{de}=\sum_{j=t2+1}^{n2} \binom{n2}{j} \cdot p^j \cdot (1-p)^{n2-j} \quad (14)$$

복호기의 복호화 오류(decoding error)를 범했다는 것은 결국, 복호화 과정에서 오류를 수정한게 아니라 오류를 더 추가시켰다는 것이다. 실제로 j개의 오류가 발생했을 경우에 A(j)를, 복호화시킨 결과 나타난 오류의 갯수라고 할때

$$d_2 \leq A(j) \leq j+t_2 \quad (15)$$

기껏해야 복호기 2는 d_2-j개의 오류를 추가하여 d_2개의 오류를 생성하고, 최악의 경우 t_2개의 오류를 수정하려는 과정에서 t_2개의 오류를 추가시켜 $j+t_2$개의 오류를 생성시킨다. 이때, 복호기 2의 출력비트 오류율 p′는

$$p'=\frac{1}{n_2}\sum_{j=t2+1}^{n2} A(j) \cdot \binom{n2}{j} \cdot p^j \cdot (1-p)^{n2-j} \quad (16)$$

복호기 1의 메시지 오류율 P_{me}는,

$$P_{me}=1-\sum_{j=0}^{t1} \binom{n1}{j} \cdot (1-(1-p')^m)^j \cdot ((1-p')^m)^{n_1-j} \quad (17)$$

그림 16은 $M=1024$, $n_1=n_2=38$, $k_1=k_2=32$, $t_1=t_2=1$일때, p가 10^{-4}에서 10^{-8}까지 변함에 따른 메시지 오류율의 변화를 보여 주고 있다. 다단계 부호화의 경우가 가장 신뢰성이 높은 방법이라는 것을 알 수 있다. 암호화전 부호화와 암호화후 부호화의 정보율은 공히 0.84이나 다단계 부호화의 경우에는 정보율이 0.71로 감소한다. 그림 17은 서로 다른 오류 수정 능력을 지닌 부호를 부호기 1과 2에 사용할 경우 어디에 보다 많은 오류 수정 능력을 가진 부호를 배치해야 신뢰성이 더 높은 방식인지를 보여준다. $M=1024$일때, $[n_1=38, k_1=32, d_1=3]$ 그리고 $[n_2=42, k_2=32, d_2=5]$인 경우와 $[n_1=42, k_1=32, d_1=5]$ 그리고 $[n_2=38, k_2=32, d_2=3]$을 비교한 결과, 부호기 2에 오류 수정 능력이 큰 부호를 배치하여야 신뢰성이 더 증대된다.

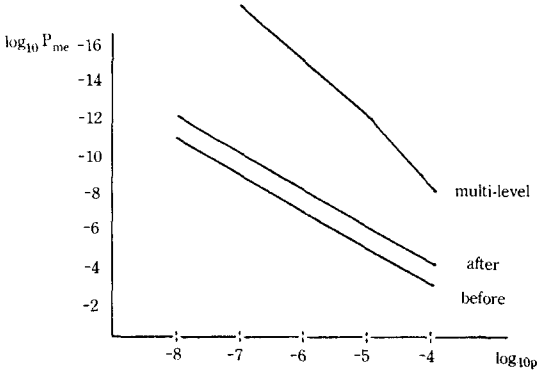


그림 16. 3가지 방식의 메시지 오류율 비교

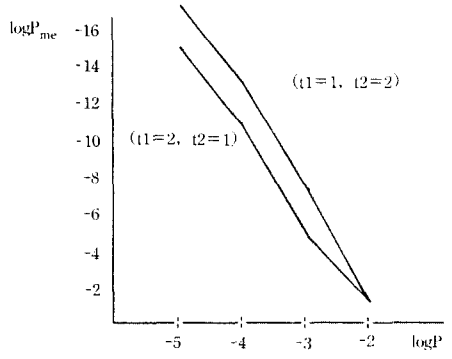


그림 17. 다단계 부호화의 부호 1과 부호 2의 최소거리 변화에 따른 메시지 오류율

5. 결 론

채널잡음에 대한 메시지의 신뢰성 측면을 고려할 경우, 다단계 부호화, 암호화후 부호화, 암호화전 부호화의 순으로 효율적이다. 하지만 다단계 부호화는 2단계의 서로 다른 부호의 사용으로 인한 데이타의 팽창이 크다. 특히, 다단계 부호화의 경우, 첫째 부호 보다 두번째 부호에 보다 큰 최소거리를 갖는 부호를 사용하는 것이 신뢰성 면에 있어서 더 뛰어나다.

암호화후 부호화의 경우, 능동적 공격에 의한 메시지의 삭제, 삽입은 부호화의 알고리즘 자체가 비밀사항이 아니기 때문에 메시지 자체에 충분한 redundancy가 존재하지 않는다면, 수신자에 의한 복호화 과정에서 노출되어 지지 않는다. 이러한 상황에서는 CBC(cipher block chaining) 보다는 새로이 제시되는 PBC(piaintext block chaining) 방식을 사용한다면 마지막 블럭 decryption 과정에서 기대되어지는 IV(initialization vector)를 통해서 그러한 메시지의 삽입, 삭제를 검출할 수 있게 된다.

암호화전 부호화와 다단계 부호화의 경우에는 능동적 공격에 의한 단순한 메시지의 삽입, 또는 삭제는 블럭형성(blocking) 방식 때문에 쉽게 검출 되어진다. w개의 블럭 중에서 어느 특정 블럭의 삭제와 동시에 임의의 블럭의 삽입은 블럭형성 방식에 의해 노출이 되지 않지만 암호화전 부호화에서 수행되어지는 code interleaving 방식에 의해 검출되어진다. 특히, 다단계 부호화의 경우, 암호화후 부호화의 복호기를 무사히 빠져나오는 임의의 대체된 블럭은 interleaving 되어진 암호문이기 때문에 decryption 되어진 이후의 그것에 해당하는 평서문의 각각의 비트들은 m개의 코드워드에 각각 속하는 비트여서 임의의 메시지의 단순 대체 역시 암호화전 부호화의 복호과정에 의해 검출되어진다.

참 고 문 헌

1. F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-correcting Codes*, North-Holland Publishing Company, 1977.

2. Seberry and Pieprzyk, *Cryptography*, Pretice Hall, 1990.

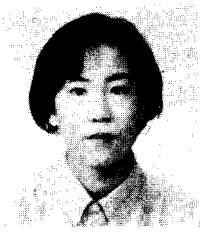
3. W.Diffie and M.D. Hellman, "Privacy and Authentication : An Introduction to Cryptology," Proc. of the IEEE, Vol.67, No.3, pp.397-427, Mar. 1979.

4. G.B. Agnew, "Cryptographic Systems Using Redundancy," IEEE Tr. on Inform. Theory, Vol. 36, No. 1, pp.31-39, Jan. 1990.

□ 著者紹介

박 창 섭

1958년생
연세대학교 경제학과 졸업
미국 Lehigh 대학 전자계산학 석사
미국 Lehigh 대학 전자계산학 박사
현재 단국대학교 전자계산학과 조교수
관심분야 : 암호이론 및 부호이론



이 수 연

1967년생
단국대학교 전자계산학과 졸업
현재 단국대학교 전산통계학과 대학원
관심분야 : 암호이론 및 부호이론