通信情報保護學會誌
第2卷, 第1號, 1992. 3

암호의 운영 방식

이 임 영*

1. 서 론

컴퓨터와 통신이 결합한 정보화 사회의 발전과 더불어, 데이타 보호 및 인증문제의 중요성은 아무리 강조하여도 지나치지 않을 것이다. 이러한 문제의 해결방법으로 암호에 관한 연구는 활발히 진행되고 있는 상황이다.

암호는 크게 블럭 암호와 스트림 암호로 나눌 수 있다. 블럭 암호(Block Cipher)는, 데이타를 블럭 단위로 암호화 및 복호화하는 방식이며, 스트림 암호(Stream Cipher)는, 1문자 혹은 1비트씩 서로 다른 키로 다른 문자 혹은 비트로 변환하는 방식이다.

본고에서는 실제적으로 암호화 방식이나 암호 장치에 이용되고 있는 암호의 이용모드에 관하여, 데이타 출력의 관점으로부터 블럭 암호와 스트림 암호로 나누어 간단한 해설을 하기로 한다.

2. 블럭 암호와 스트림 암호

블럭 암호는 M을 평문 메세지로 나타낼 때 M을 같은 크기를 갖는 연속한 블럭 M_1, M_2,…로 분할하여 각 M_i를 동일 키 K로써 암호화하는 방식이다. 즉,

$$E_k(M)=E_k(M_1)E_k(M_2)\cdots$$

이와 같이 적당한 길이의 문자열(블럭)을 같은 키로 다른 문자열(블럭)로 변환하는 블럭 암호의 대표적인 예는 표 1과 같다.

표 1. 블럭 암호의 예

블럭 암호	블럭 사이즈
Transposition with period d	d characters
Simple Substitution	1 character
Homophonic Substitution	1 character
Playfair	2 characters
Hill with d×d matrix	d characters
DES	64 bits
Exponentiation mod n	$\log_2{}^n$ bits
Knapsacks of length n	n bits

스트림 암호는 메세지 M을 연속한 문자 혹은 비트 m_1, m_2…로 분할하여 m_i의 암호화에는 키

* 한국전자통신연구소 선임연구원

스트림 $k=k_1, k_2, \cdots$의 i번째의 요소인 k_i를 사용하는 방식이다. 즉,

$$E_k(M)=E_{k1}(m_1)E_{k2}(m_2)\cdots$$

스트림 암호는 사용하는 키 스트림 형태에 따라 주기성 암호(Periodic Cipher)와 비주기성 암호(Nonperiodic Cipher)로 나누어진다. Rotor Machine이나 Hagelin Machine으로 만든 암호는 주기 스트림 암호이고, Vernam암호(One-time pad)나 Running Key 암호는 비주기 스트림 암호이다.

또한 스트림 암호는 동기식(Synchronous Stream Cipher)과 자동 동기식(Self-synchronous Stream Cipher) 방식으로 나눌 수 있으며, 그 대표적인 예는 표 2와 같다.

표 2. 스트림 암호의 예

	스트림 암호	주 기
동	Vigenère with period d	d
	Rotor machine with t rotors	26^t
	Hagelin machines with t wheels, each having p_i pins	$p_1p_2\cdots p_t$
기	Running Key	—
	Vernam	—
	LFSR with n-bit register	2^n
식	OFB mode with DES	2^{64}
	Counter method with DES	2^{64}
자 동	Autokey Cipher	
동기식	CFB mode	

3. 블럭 암호

평문 및 암호문의 블럭 사이즈를 각각 n비트라 할 때, 평문 블럭에 하나의 암호문 블럭을 대응시키는 암호 변환의 수는 $2^n!$ 가지수가 있다. 따라서 키의 총수는 $2^n!$ 개 있고, 키의 최대 사이즈는 $\log 2^n!$ 비트이다.

블럭 사이즈가 작으면 주어진 키에 대응한 평문과 암호문 블럭의 쌍을 모두 기억 장치에 저장 시킬 수 있고, Exhaustive attack으로 암호문으로 부터 평문을 탐색당하는 위험이 있다. 또한 평문의 언어적 특징이 암호문 블럭의 통계적 특징(문자의 출현 빈도)으로 나타나기 쉽고 이경우 문자의 출현 빈도에 의한 암호 해독 방법으로 해독될 위험이 있다.

한편 블럭 암호에서는 블럭내의 1비트의 전송 에러가 복호화 후에 동일 블럭내의 거의 모든 비트에 영향을 미친다. 따라서 블럭 사이즈가 너무 크면 전송 에러 전파(Error Propagation)의 관점에서 바람직하지 않으며, 블럭 사이즈는 이상의 점들을 고려하여 결정되어진다.

그리고 블럭 암호는 다음과 같은 고유의 문제점이 있다.

(1) 평문 전체의 사이즈가 블럭 사이즈보다 작은 경우나 블럭 사이즈의 정수배가 아닐 경우 padding (블럭의 남는 부분에 dummy bit를 넣는 것) 조작을 한다. 이 padding에 의해 평문의 내용에 치우침이 생겨 암호해독의 위험성이 생긴다.

(2) 동일의 암호키를 사용함으로써 동일의 평문에 대하여 동일의 암호문이 생겨나, 기지평문 공격(Known plaintext attack)에 약하다.

(3) 암호문에 대하여 능동적 부정행위, 즉 블럭 단위로 수정될 위험성이 있다.

위와 같은 문제점을 해결하기 위하여 chaining 기법이 고안되었고, Feistel은 블럭 chaining 기법을 사용함으로써 블럭 암호가 암호해독에 대하여 보다 강력해지는 것을 밝혔다. 이것은 각 평문 블럭 M_i를 암호화할 때 이전의 암호문 블럭 C_i 중, 일부분(비트)을 M_i의 일부로 M_i의 미사용비트 위치에 넣는 형태로 블럭을 계속해서 연결하는 방법이다. Kent는 같은 방식으로 일련번호를 사용하는 것을 제안하였다. 이러한 방법들은 CFB모드(Cipher FeedBack)의 스트림 암호와 같이 불법의 삽입,

삭제, 변경을 막는 것이다. 다만 CFB모드와 다른 점은 블럭 chaining의 경우는 각 블럭의 일부분이 유효한 메세지 비트가 되지 않는다는 점이다. 이러한 결점을 보완한 것이 CBC(Cipher Block Chaining)모드이다.

－CBC 모드

CBC 모드는 1개의 암호문 블럭 전체를 그대로 레지스터에 feedback하여 다음의 평문 블럭과 EXOR한 후 암호화하는 방식이다. 따라서 i번째의 평문 블럭 M_i를 암호화 및 복호화하는 방법을 수식으로 표현하면 다음과 같다.

$$C_i = E_k(M_i + C_{i-1}) \quad 단\ C_0 = I_0$$

$$D_k(C_i) + C_{i-1} = D_k(E_k(M_i + C_{i-1})) + C_{i-1}$$
$$= (M_i + C_{i-1}) + C_{i-1}$$
$$= M_i$$

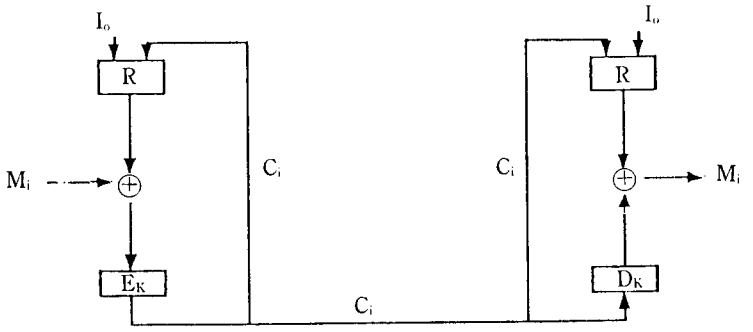


그림 1. CBC 모드

암호문 블럭 C_i는 M_i와 직전의 암호문 블럭 C_{i-1}로부터 계산하기 때문에 전송에러의 영향은 최대 2블럭에 미친다. 또 각 C_i는 그 이전의 전 암호문 블럭에 함수적으로 종속하여져 있기 때문에, 평문이 갖고 있는 통계적 특성은 암호문 전체에 확산되어져 암호해독이 어렵게 된다. CBC모드의 최후의 암호문 블럭 C_n은 CFB모드와 같이 checksum(검사용 합계)으로 사용할 수 있다.

이와 같은 CBC 모드는 송신측에서는 암호문의 feedback, 수신측에서는 암호문의 feedforward로서 특징지을 수 있다. 1비트의 암호문의 에러가 수신한 평문에 미치는 영향을 그림 2에서 나타내었다.

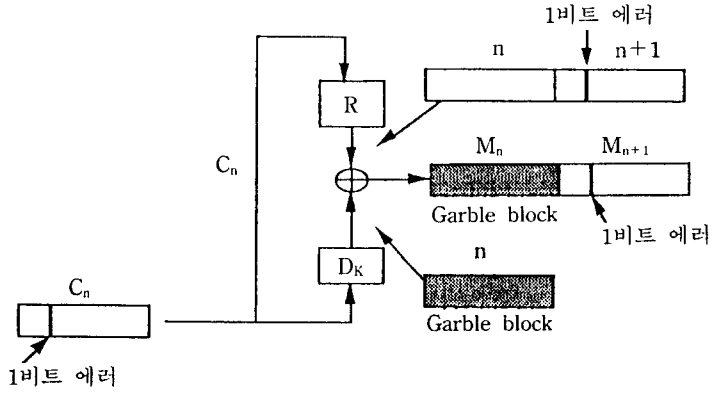


그림 2. CBC 모드에서의 1비트 에러

그 영향은 먼저 수신측의 암호 알고리즘의 입력을 변경하는 것이다. 암호 알고리즘의 뛰어난 특성에 의해 알고리즘에서의 출력은 랜덤하게 된다. 다음 블럭의 시점에서 암호문의 에러는 feedforward 경로의 레지스터에서 나타나 암호문 1개의 비트 에러는 평문 1비트 에러가 되어 제 2 블럭내의 대응하는 위치에 나타난다. 제 2 블럭에 이어지는 블럭은 암호문의 에러에 영향을 받지 않는다. 이런 점에서 CBC 모드는 자동 회복 기능을 가진다고 볼 수 있다.

비트 에러로부터 자동적으로 회복하는 것과는 대조적으로, 동기 에러의 경우 시스템은 회복이 어렵다. 즉 동기 에러의 경우는 전송중에 1비트가 상실 혹은 추가 되었을 때, 그 블럭위치는 1비트 만큼 비켜나 수신 시스템은 부정확한 데이타를 계속 생성하게 된다. CBC 모드에서는 블럭의 프레임의 형태를 잃는 것을 방지하여야 한다.

이와같이 전송상의 단순한 에러를 평문 출력의 에러로 변환하는 특성을 Error Extension 혹은 Garble Extension이라 한다.

IBM의 암호화 S/W인 IPS(Information Protection System ; 정보보호시스템)는 CBC 모드를 사용하고 있다.

4. 스트림 암호

스트림 암호의 키 스트림 생성 방식을 동기식(Synchronous)과 자동 동기식(Self-Synchronous)으로 나눌 수 있고, 동기식 스트림 암호에서는 키 스트림이 평문이나 암호문의 스트림과 독립으로 생성되고, 자동 동기식 스트림 암호는 키 스트림이 앞의 암호문 스트림에 의존하여 생성된다.

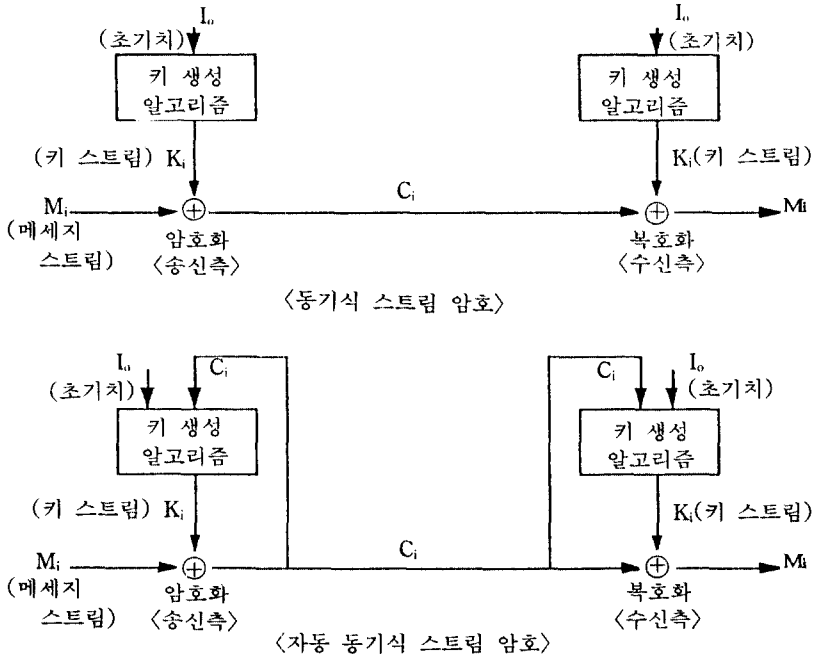


그림 3. 동기식 스트림 암호와 자동 동기식 스트림 암호

동기식 스트림 암호에서의 키 스트림은 메세지 스트림과는 독립적으로 생성된다. 따라서 전송 중에 암호문 1비트가 다른 비트로 변화하더라도 동기는 어긋나지 않으나, 암호문 1비트가 소멸하면 그 비트 이후의 모든 암호문에 대하여 동기가 어긋나 에러가 전파된다. 동기를 다시 맞추기 위해서는 송신측과 수신측에서 암호 통신 개시의 프로토콜에 맞추어 다시 새로운 초기치 부터 키 계열을 생성하여야 한다.

한편 자동 동기식 스트림 암호(키가 1비트 전까지의 암호문에 의존한다고 하자)에서는 전송 중에 암호문 1비트가 다른 비트로 변화하던지 1비트가 소멸하면 그 에러는 그 후의 1비트에 전파한다. 그러나 그 후 바른 암호문이 1비트 만큼 수신측에 도착하면 다시 자동적으로 동기가 맞추어져 올바른 복호화가 행하여진다. 이와같이 1비트의 전송 에러가 후속의 1비트에만 영향을 미치기 때문에 자동 동기식이라 한다. 그리고 키의 각 문자가 선행하는 전 메세지 스트림에 함수적으로 종속하기 때문에 비주기적이다.

즉, 동기식 스트림 암호는 chaining이 없는 스트림 암호이고, 자동 동기식 스트림 암호는 chaining이 있는 스트림 암호이다.

4. 1 동기식 스트림 암호

Chaining이 없는 동기식 스트림 암호는 키 스트림이 평문이나 암호문 스트림과 독립적으로 생성되는 스트림 암호이다. 수신측에서도 같은 키 스트림을 재생성시키기 위해서는 생성 알고리즘을 미리 결정해야 한다. 이러한 동기식 스트림 암호는 그림 3의 키 생성장치에 있어서 키 생성 알고리즘의 성질에 의해 선형 궤환 방식(Linear Feedback Method)과 비선형 궤환 방식(Non-linear Feedback Method)으로 나누어진다.

4. 1. 1 선형 궤환 방식

선형 궤환 방식을 실현하는 간단한 동기식 스트림 암호로서 LFSR(Linear Feedback Shift Register) 방식이 있다. 구체적인 구성에 대하여서는 본 학회지 창간호(1991. 4. pp. 45~53)를 참조하길 바란다.

4. 1. 2 비선형 궤환 방식

선형 궤환 방식은 안전성에 있어서 문제점이 있기 때문에 키 생성 알고리즘은 비선형이 좋다. 비선형 성질을 가지는 암호 알고리즘(예를 들어 DES)을 키 생성 알고리즘으로 사용하여, feedback을 행하지 않는 Counter 방식과 feedback을 행하는 OFB모드에 관하여 논하기로 한다.

－Counter 방식

Diffie와 Hellman은 Counter 방식을 제안하였다. E_k의 출력 결과를 반복하여 E_k에 입력하여 순환시키는 대신에, 단순한 Counter를 사용하여 입력 블럭을 연속적으로 생성하는 방식이다.

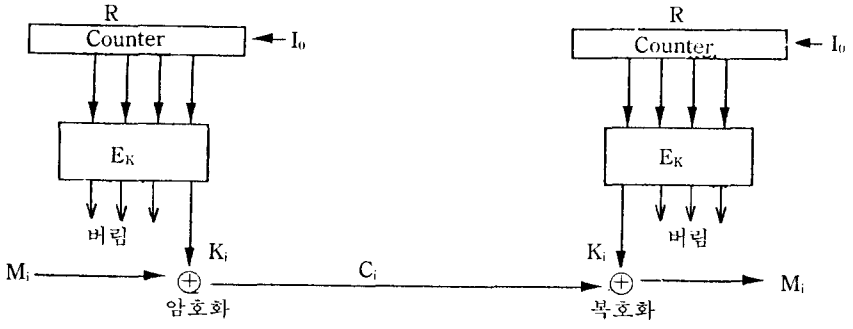


그림 4. Counter 방식

이 방식에서는 Counter를 I_0+i-1로 세트하여 i번째의 키 문자 K_i를 생성하기 때문에 K_i에 선행하는 i−1개의 키 문자를 만드는 순서가 불필요하다. 이것은 대단히 편리한 특징으로서 access file의 i번째 문자에 직접 access할 경우 편리하다. 이것을 OFB모드로 실행하려고 하면 선행의 i−1개의 키 문자를 계산하여야 한다.

−OFB(Output FeedBack) 모드

Feedback레지스터 R(64비트)의 내용을 암호 알고리즘 E_k(암호화 키 K는 고정)의 입력으로써 사용한다. 초기치 I_0는 최초에 레지스터 R에 둔다.

암호 알고리즘의 출력 $Y_i=E_k(R_i)$를 레지스터 R에 feedback시키면서, Y_i의 좌측(LSD측)의 j비트(i≦j≦64)를 키 K_i로써 꺼낸다. 즉 j비트 단위의 키로써 EXOR의 암호화/복호화를 행하는 스트림 암호이다. FIPS(Federal Information Processing Standards)에서는 feedback하는 비트수를 64비트로 고정하지 않고 m비트(1≦m≦64)로 일반화하여 규정하고 있다. 즉 난수키 생성기의 출력 Y의 모든 것이 아니고, Y의 좌측 m비트를 레지스터 R의 우측 m비트에 feedback시킨다. 이때 레지스터 R은 m비트의 왼쪽 shift를 행한다. 그림 5는 j비트의 문자와 j비트폭의 feedback경로로 동작하는 OFB 모드를 나타낸 것이다. 이것은 feedback의 출발 위치를 빼면 모든 점에서 CFB 모드와 유사하다.

회선상의 2개의 단말에서 난수 생성기의 동기를 맞추는 것은 중요하다. 만일 동기가 어긋나면 수신측의 평문 출력은 랜덤하게 된다. OFB 모드에서는 문자가 첨가 혹은 분실되면 회복 못하게 되므로, OFB모드를 사용하는 시스템은 이러한 경우가 발생하였을때는 재차 동기를 맞추어야 한다. 이것은 새로운 초기치로 재개시하는 것과 같다.

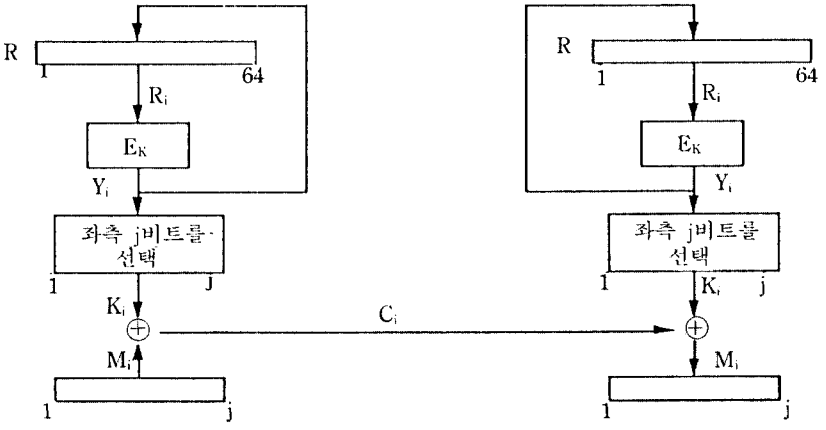


그림 5. OFB 모드

4. 2 자동 동기식 스트림 암호

4. 2. 1 선형 궤환 방식

−Antokey 암호

Antokey 암호는 암호화할 메세지로 부터 키를 도출하는 방식이다. Vigenère의 제 1 암호에서는 키는 priming key를 첫번째의 k_1으로써, 그 후는 평문 메세지 $M=m_1m_2\cdots$를 부가하여 만든다. 즉 i번째 키 문자 (i>1)는 $k_i=m_{i-1}$로써 주어진다.

(예) Priming key D를 사용하여, 평문 RENAIS-

SANCE를 암호화하면 다음과 같이 된다.

M=RENAISSANCE
K=DRENAISSANC
$E_k(M)$=UVRNIAKSNPG

Vigenère의 제 2 암호에서는 키는 priming key 뒤에 암호문 문자를 부가하여 만든다. 즉 $k_i=c_{i-1}$ (i>1)로써 주어진다.

(예) Priming key D를 사용하여 평문 RENAISSANCE를 암호화하면 다음과 같다.

M=RENAISSANCE
K=DUYLLTLDDQS
$E_k(M)$=UYLLTLDDQSW

이러한 것은 지금의 기준으로 보면 강한 암호라고 볼 수 없으나, Vigenère이 발견한 "비반복성의 키 스트림을 암호화하는 메세지 자체로부터 도출한다."라고 하는 발상은 암호의 발전에 큰 기여를 하였다.

Vigenère의 제 2 암호는 각 키문자가 선행하는 암호문 문자로부터 산출된다는 의미에서 일종의 자동 동기식 스트림 암호라 할 수 있다. 그러나 이러한 Vigenère 시스템의 약점은 암호문 중에 키가 존재하기 때문에, 암호문으로 부터 키를 얻을 수 있다는 점이다. 이점을 해결하는 간단한 방법으로써 키를 도출할 때 복수의 암호문자를 비선형의 암호 알고리즘에 feedback 입력하는 방법이다. 이러한 기법이 CFB 모드이다.

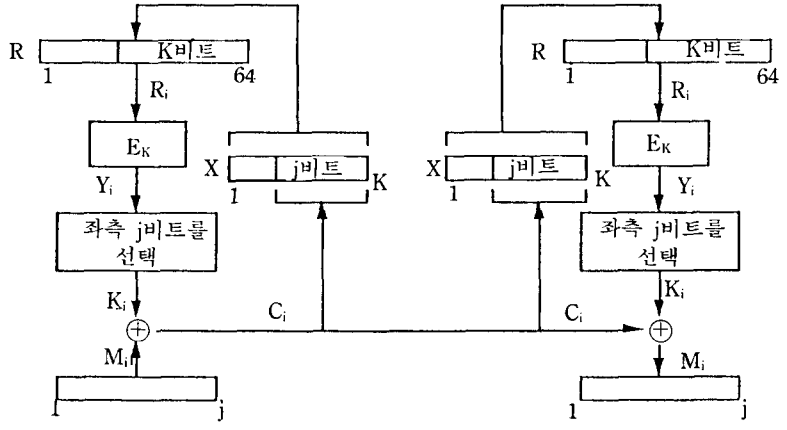


그림 6. CFB 모드

4. 2. 2 비선형 궤환 방식

-CFB(Cipher FeedBack) 모드

Feedback레지스트 R은 shift 레지스트로서 움직여, 처음에 I_o로 초기화한다. 암호문 문자 C_i를 생성하면 곧 레지스트 R의 좌단에 shift-in하여 우단의 1문자를 shift-out하여 버린다. 이 R_o의 값을 입력하여 E_k(암호 알고리즘)을 계산하고, 출력 결과 중 최하위 문자를 다음 단계의 키 문자로 한다.

CBC 모드가 전체의 블럭을 조작하는 것에 대하여 CFB 모드는 1회에 1개의 문자를 조작하고 문자의 길이 m을 설계의 파라메터로 선택할 수 있고, 이것을 m비트 CFB 모드라 한다. 문자의 길이는 최소

1비트의 경우 1비트 CFB 모드가 되며, 일반적으로 m의 값은 64비트까지 사용할 수 있다. 보통 통신 시스템에서 사용하는 문자의 크기에 의해 m을 선택한다.

CFB 모드에서는 전송 에러가 feedback 그룹에 영향을 준다. 전송 중의 암호문에서 1문자가 변하면 수신측의 shift 레지스트가 송신측의 레지스트와 동기가 맞지 않게 되어, 그 이후의 암호문은 에러 문자의 영향이 shift-out할 때까지 올바르게 복호되지 못한다. 만일 블럭별 문자 수를 n이라 하면 송수신측의 레지스트는 n사이클 후 동기가 맞아지기 때문에 에러는 최대 n문자에 영향을 미쳐 그 후에 다시 정상으로 돌아온다.

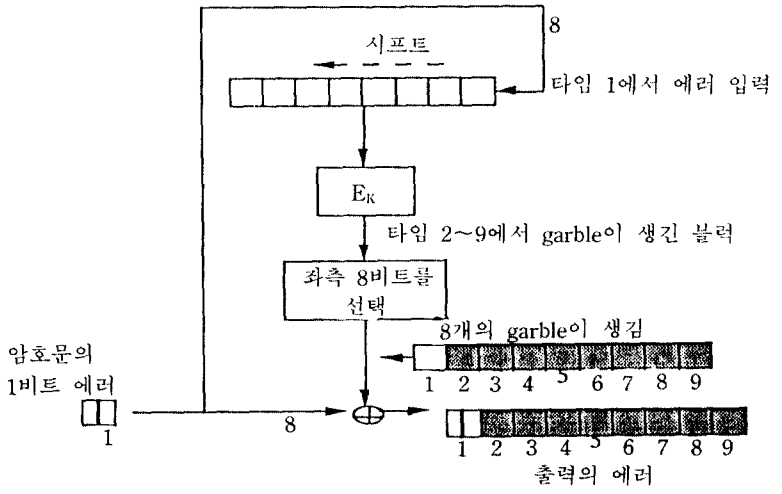


그림 7. 8비트 CFB 모드에서 1비트 에러

그림 7은 8비트 CFB 모드에서 1비트의 전송 에러의 효과를 나타낸 것이다. 먼저 그 문자의 평문 출력 중의 대응하는 비트를 변경하는 것이다. 에러는 또 shift 레지스터에 들어가 shift 과정에서 레지스터의 좌측으로부터 에러를 낼 때까지 그곳에 남아, 그동안 출력의 값을 변경시킨다. 그림 7에서 8비트 CFB 모드의 경우 단일 비트 에러가 평문 문자 9비트에 영향을 미친다.

CFB 모드에서는 Counter 방식과 마찬가지로 access 파일의 데이타에 직접 access가 가능하다. 따라서 i번째의 암호문 문자 C_i를 복호하기 위해서는 feedback 레지스트에 가까운 n개의 암호문 문자 C_{i-n}, …, C_{i-1}을 돌려 feedback loop를 한 사이클 실행함으로써 K_i를 얻을 수 있다.

5. 결 론

본고에서는 암호의 이용 모드에 대하여 블럭 암호와 동기식 스트림 암호 및 자동 동기식 스트림 암호로 나누어 고찰하였다.

동기식 스트림암호에서는 평문의 내용이 동일하더라도 블럭이 다르면, 암호화 키가 다르기 때문에 암호문 탐색에 의한 해독에 대하여서는 방어할 수

있다. 암호문중에 외부로부터의 삭제 혹은 삽입등의 경우 동기가 어긋나기 때문에, 허위의 암호문을 삽입하던지 암호문의 일부를 삭제하는 등의 공격을 방지할 수 있다.

동기식 스트림암호의 또 하나의 이점으로는 에러를 파급하지 않는다는 것이다. 즉, 어떤 문자에 에러가 생기더라도 후속의 문자에는 영향을 주지 않는 점이다.

자동 동기식 스트림 암호에서는 메세지 스트림중 서로 다른 문자는 키 스트림의 서로 다른 문자로 암호화 되기 때문에 암호문 탐색등의 침입을 막을 수 있다. 또한 암호문에 어떤 조작을 하면 반드시 키 스트림이 변하기 때문에 인증등에 대한 침입에 대하여서도 강한 이점이 있다. 암호문의 최종 블럭은 메세지 전체에 함수적으로 종속되어 있기 때문에 메세지 전체의 검사 합계로서의 역할을 한다.

NBS(National Bureau of Standard ; 미국 상무성 표준국)와 ISO(International Organization of Standardization ; 국제 표준화 기구)는 DES 등의 블럭 암호 이용 모드로서 4종류를 규정하고 있다. 이용 모드는 64비트 및 64비트 미만의 암호 알고리즘에 적용된다. 기본이 되는 모드는 ECB 모드(Electronic Code Book)로서 암호 알고리즘을 그대로 반복하여 이용하는 모드이다. 확장한 모드로서 CBC 모드, CFB 모드, OFB 모드가 있다. 최근의 DES-LSI에서는 이 4종류의 모드가 LSI 내부에 내장되어 외부에서 모드 선택 변환이 가능한 것이 있다.

표 3은 이러한 모드들의 성질을 정리한 것이다.

표 3. 각종 모드의 성질

각종 모드	암호화 데이타의 성질	암 호 화 처리 단위	에러 파급효과*	적용 분야
ECB	평문 데이타 대응	64비트	64비트	초기치, 키의 암호화
CBC	과거의 암호문과 평문에 영향을 받는 Chaining 암호	64비트	65비트	메세지 인증
CFB	과거의 암호문과 평문에 영향을 받는 Chaining 암호	K비트 (1≤K≤64)	65비트	일반 데이타 암호화
OFB	과거의 암호문에 의존치 않는 bit by bit의 암호	K비트 (1≤K≤64)	1비트	음성 데이터 암호화 잡음이 많은 통신로의 암호통신

* : 통신로상에서 1비트 에러가 생겼을 때 수신측의 영향을 나타낸 것임.

참 고 문 헌

1. Davies, D.W. and Price, W.L., Security for Computer Network, John Wiley and Sons(1984)

2. Diffie, W. and Hellman, M., "New Directions in Cryptography," IEEE Trans. Inf. Theory, IT-22, 6, pp.644-654(Nov. 1976)

3. Diffie, W. and Hellman, M., "Privacy and Authentication ; An Introduction to Cryptography," Proc. IEEE Vol. 67, pp.397-427(Mar. 1979)

4. Denning, D.E., Cryptography and Data Privacy, Addison-Wesley Pub(1982)

5. Feistel, H., "Cryptography and Computer Privacy," Sci. Am. Vol. 228(5) pp.15-23(May 1973)

6. FIPS "Data Encryption Standard," National Bureau of Standard, Federal Information Processing Standards Publications, 46(Jan. 1976)

7. Hellman, M.E., "On DES-based Synchronous Encryption," Dept. of Electrical Eng., Stanford Univ.(1980)

8. Kahn, D., The Codebreaker, Macmillan Co., New York(1967)

9. Kent, S.T., "Encryption-Based Protection Protocols for Interactive User-Computer Communication," MIT/LCS/TR-162, MIT Lab. for Computer Networks, D. Reidel, pp.239-259(1978)

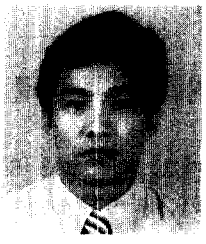
10. 한국전자통신연구소, 현대암호학(1991)

11. 한국통신정보보호학회, 통신정보보호학회지 창간호, pp.45-53(1991. 4)

12. Meyer, C.H. and Matyas, S.M., Cryptography : A New Dimension in Computer Data Security, John Wiely and Sons

13. Savage, J.E., "Some Simple Self-synchronizing Digital Data Scramblers," Bell System Tech. J., pp.448-487(Feb. 1967)

□ 著者紹介



李 壬 永(正會員)

1958年 5月 12日生
弘益大學校 工科大學 電子工學科(學士)
日本 大阪大學 大學院 通信工學科(碩士, 博士)
現在 : 韓國電子通信研究所 勤務
關心分野 : 暗號理論, 符號理論, 情報理論