通信情報保護學會誌
第3卷 第3號, 1993.9

ISO/IEC JTC1/SC27의 국제표준소개 (2) : ISO/IEC DIS 9797

정보기술 – 보안기술 – 블럭 암호화 알고리즘을 사용하여 만든 암호학적 검산 함수를 이용한 데이타 무결성 기법

(Information Technology – Security techniques – Data integrity mechanism using a cryptographic check function employing a block cipher algorithm)

이 필 중*

ISO/IEC JTC1/SC27의 국제표준소개의 첫번째 편으로 DIS 9797을 소개한다. 이것은 1989년에 1차로 국제표준이 되었다가 결함이 발견되어 다시 작업이 시작되어 1992년 10월 SC27 정기총회에서 CD(Committee Draft)에서 수정후 DIS(Draft International Standard) 수준으로 올리기로 하여 1993년 3월 DIS로서 표결에 부쳐진 문서이다. 이해를 돕기 위하여 국문뒤에 원문을 덧붙였다. 번역이 적절하지 않거나 더 좋은 표현이 있으면 역자에게 알려주면 참고하여 추후의 수정본에 반영하겠다.

서 론

이 표준에서 명세하고있는 기법은 n비트의 데이타 블럭과 m비트의 검산값을 그리고 추가적인 채워 넣기 방법의 명시 이외에는 ISO 8731-1, ISO 9807에 그리고 ANSI X9.9 표준에서 사용된 기법과 유사하다. (The mechanism specified in this International Standard is similar to that used in ISO 8731-1, ISO 9807 and in the ANSI X9.9 standard, except that it is defined in terms of an algorithm using n-bit data blocks and an m-bit check value, and that an additional padding method is specified.)

ISO 8731-1, ANSI X9.9과 ANSI X9.19에서 설명된 암호학적 검산값의 계산은 이 국제표준으로서 n=64, m=32, 4.1절에 명시된 채워 넣기 방법 1, 그리고 DEA(ANSI X3.92 : 1981 참조)을 사용하는 특별한 경우이다. (The calculation of cryptographic check values as described in ISO 8731-1, ANSI X9.9 and ANSI X9.19 is a subset of this International Standard when n=64 and m=32, the padding method 1 specified in clause 4.1 is used, and when DEA(see ANSI X3.92 : 1981) is used.)

* 포항공과대학 전자전기공학과

1. 범위 (Scope)

이 국제표준은 데이타가 비공인된 방법으로 변경되었는지를 검출하는 데이타 무결성 기법(data integrity mechanism)으로 사용되어질 수 있는 한가지 방법을 규정한다. 이 방법에서는 m-비트의 암호학적 검산값(m-bit cryptographic check value)을 계산하기 위하여 n-비트의 블럭 암호화 알고리즘(n-bit block cipher algorithm) 및 암호화 키(cryptographic key)(줄여서 "키"라고 부르겠음)를 사용한다. 데이타의 무결성의 정도(degree of integrity of data)는 키의 길이가 얼마나 긴가, 키의 보안이 얼마나 잘되어 있는가, 검산값의 길이 m이 얼마나 긴가, 그리고 암호화 알고리즘이 얼마나 안전한가 등에 달려 있다. (This International Standard specifies a method of using a key and an n-bit block cipher algorithm to calculate an m-bit cryptographic check value that can be used as a data integrity mechanism to detect that data has not been altered in an unauthorised manner. The degree of integrity of the data is dependent on the key length and its secrecy, on the nature of the cryptographic algorithm, and on m, the length of the check value.)

2. 용어 (Terminology)

이 국제표준에서는 암호학적 검산값을 메세지 인증코드(Message Authentication Code), 줄여서 MAC, 이라 칭한다. (This International Standard refers to the cryptographic check value as a Message Authentication Code (MAC).)

3. 요구조건 (Requirements)

MAC의 길이 m은 암호화 알고리즘에서 사용되는 블럭의 길이 n보다 작거나 같아야 한다. 계산의 결과와 선택적 처리(optional process)의 결과인 블럭들의 길이는 암호화 알고리즘에서 사용되는 블럭과 같은 n의 길이를 갖는다. MAC의 길이 m이 n보다 작은 경우 최종 n-비트 블럭의 맨 좌측부터 m개의 비트들을 선택하여 MAC을 구성한다. (The length (m) of the MAC will be less than or equal to the block length (n). The result of the calculation and of any optional process is an information block of length n. The MAC is the m leftmost bits of the final n-bit block.)

4. MAC 알고리즘의 수행 (MAC calculation)

4.1. 채워 넣기(padding)와 블럭 나누기 (blocking)

MAC 산출시 아래의 두가지 채워 넣기 방법 중 한가지를 선택해야 한다. 어떤 방법을 선택 하는가에 대한 것은 이 국제표준의 범위에서 벗어난다. (The generation of a MAC requires the selection of one of two padding methods. The way in which the selection is made is beyond the scope of this International Standard.)

주 : 확인자가 확인하고자 하는 데이타의 길이를 알고있지 않는다면 채워넣기 방법 2를 사용하여야만 한다. 왜냐하면 방법 2는 (데이타가 전송 혹은 저장 중 변하지 않았다면) 확인자가 채워넣은 비트 '0'들의 갯수를 정확히 알 수가 있는 반면, 방법 1은 그렇지 못하기 때문이다. (NOTE : If the length of data is not known by a verifier then padding method 2 should be used, since it permits a verifier to detect the addition or deletion of trailing '0' bits.)

[역자 주 : 반면 데이타의 길이가 n의 배수로 고정되어 있다면 방법 1을 사용하는 것이 유리하다. 왜냐하면 필요없는 채워 넣기가 없어지기 때문이다. 그 이외의 경우는 사용자가 어느 방식을 선택하든 무방하다.]

방법 1. (Method 1)

MAC을 계산할 데이타의 길이(비트 단위로)가

n의 정수배가 되지 않으면 n의 정수배가 되도록 필요한 갯수만큼 (최소한의) 비트 '0'들을 채워 넣는다. (The data for which the MAC is to be calculated shall be appended with as few (possibly none) '0' bits as necessary to obtain a data string whose length (in bits) is an integer multiple of n.)

방법 2. (Method 2)

MAC을 계산할 데이타의 끝에 한개의 비트 '1'을 채워 넣은 후 한 비트 늘어난 데이타의 길이(비트 단위로)가 n의 정수배가 되지 않으면 n의 정수배가 되도록 필요한 갯수 만큼 (최소한의) 비트 '0'들을 채워 넣는다. (The data for which the MAC is to be calculated shall be appended with a single '1' bit. The resulting data shall then be appended with as few (possibly none) '0' bits as necessary to obtain a data string whose length (in bits) is an integer multiple on n.)

채워 넣기 결과의 데이타는 q개의 n-비트 블럭 (D_1, D_2, …, D_q)으로 나누어진다. 선택된 채워 넣기 방법에 따라 원래의 데이타에 채워 넣어진 비트는 단지 MAC을 계산하고 확인하는데만 사용된다. (만일 있다면) 채워넣은 비트는 데이타와 함께 전송되거나 저장될 필요는 없다. 사용된 채워 넣기 방법을 알고 있는 확인자는 그 채워넣은 비트가 무엇이었는지를 알 수가 있기 때문이다. (The resulting data is divided into n-bit blocks (D_1, D_2, …, D_q). The bits which are padded to the original data, according to the chosen padding method, are only used for calculating and verifying the MAC. If the data is to be transmitted the padding bits (if any) need not be transmitted with the data. The verifier shall know if they are transmitted.)

4.2. 암호화 키 (The cryptographic key)

MAC을 산출할 때 암호화에 사용되는 키는 무작위로 또는 무작위에 가깝게 발생되어야만 한다. 만일 같은 알고리즘이 비밀성을 위하여 메시지를 암호화하는데 사용된다면, MAC을 계산하는데 사용되는 키는 암호화하는데 사용되는 키와는 달라야만 한다. (The key should be randomly or pseudo-randomly generated. If the same algorithm is used for encipherment of the message, the key used for the calculation of the MAC should be different from that used for encipherment.)

4.3. 초기 단계 (The initial stage)

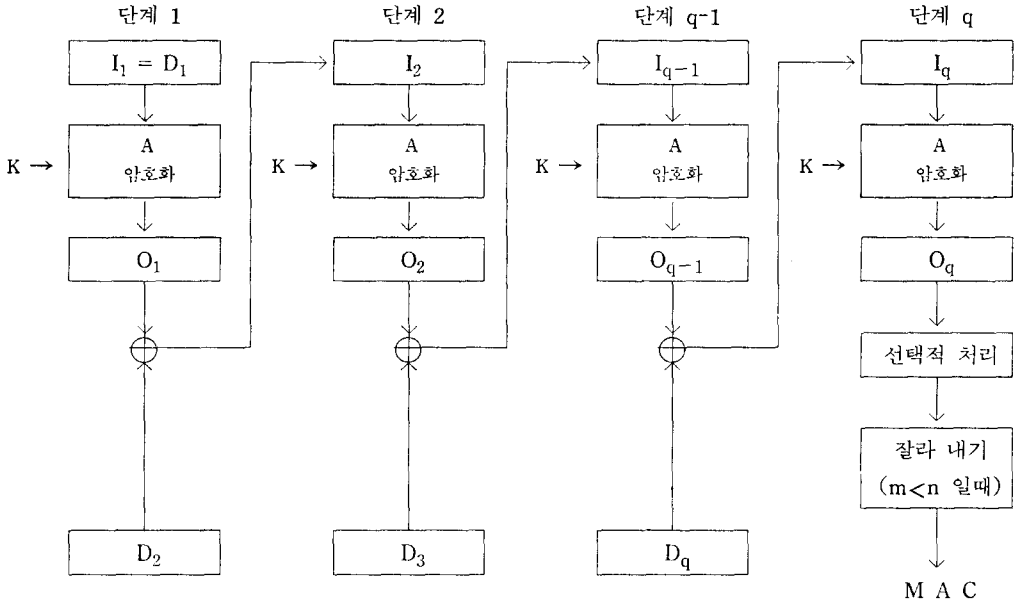
MAC은 그림 1에 나타낸 것과 같이 계산된다. (The MAC is calculated as illustrated in figure 1.)

암호화 알고리즘의 입력 레지스터 I_1은 첫번째 데이타 블럭 D_1으로 초기화된다. 입력 데이타 I_1은 알고리즘 A로 넘겨지며, 이 알고리즘은 키 K를 사용하여 그 결과 n비트를 출력 레지스터 O_1에 입력시킨다. (The input register is initialized with the first block (D_1). This input data I_1 is passed through the algorithm (A), which uses a key (K) to produce n bits in the output register (O_1).)

[역자주 : 암호화 알고리즘의 입력과 출력도 데이타 블럭의 길이와 마찬가지로 각 n비트이다.]

4.4. 후속 단계들 (Subsequent stages)

그 다음 n비트의 데이타 블럭 D_2는 출력 레지스터 O_1의 비트와 비트끼리의 배타적 논리합(exclusive or) 연산($0\oplus0=1\oplus1=0$, $0\oplus1=1\oplus0=1$)이 되며 그 결과는 그 다음 단계의 입력 레지스터 I_2로 옮겨진다. 입력 레지스터 I_2의 내용은 알고리즘 A로 넘겨지며, 이 알고리즘은 키 K를 사용하여 그 결과 n비트를 출력 레지스터 O_2에 입력시킨다. 이 동작이 모든 블럭이 처리될 때까지 계속된다. 그 결과는 최종 출력 블럭 O_q가 된다. (The next n bits of data (D_2) are bitwise exclusive or'ed with the n bits of the output reister (O_1) and the result is loaded into the input register of the next stage (I_2). The contents of the input register (I_2) is passed through the algorithm (A), which uses the key (K) to



설명(Legend) : I_i 입력(input block) i = 1, 2, ···, q
A n-비트 암호화 알고리즘 (n-bit block cipher algorithm)
O_i 출력(output block) i = 1, 2, ···, q
K 키 (key)
D_i 데이타 블럭 (data block) i = 1, 2, ···, q
⊕ 배타적 논리합 (exclusive or)

그림 1. MAC 계산 (The MAC calculation)

produce n bits in the output register (O_2). This operation continues until all blocks have been processed. The result will be the final output block (O_q).)

4.5. 선택적 처리 (Optional Process)

최종 출력 블럭 O_q은 MAC의 보안성을 강화하기 위해 선택적 처리를 받을 수도 있다. 선택적 처리는 (만약 사용된다면) 표준 부록 A에 규정된 것들 중 한가지 이어야만 한다. (The final output block (O_q) may be subjected to optional processing to increase the strength of the MAC. The optional process (if used) shall be selected from those specified in normative Annex A.)

4.6. MAC (The MAC)

MAC의 길이 m이 n보다 작다면 최종 출력 블럭의 n비트 중 맨 좌측부터 m비트로 MAC을 형성한다. (The m leftmost bits of the final n-bit block form the MAC.)

주 : n과 같은 m을 선택하는 것은 MAC에 사용된 키 K가 무엇이었는가를 모든 가능성이 있는 키 K의 값을 다 대입해서 찾아내는 공격(exhaustive key search attack)을 당하기 쉽다. 그 공격에 걸리는 시간은 사용된 암호화 알고리즘의 강도와 관계된다. 부록 A의 'A.1' 절에 규정된 선택적 처리를 사용함으로써 이 위협을 줄일 수 있다. (NOTE : Choosing m equal to n may make the MAC key susceptible to exhaustive search attack, the search

which will be dependent upon the strength of the cipher algorithm being employed. Use of the optional process specified in clause 'A.1' of Annex A reduces this threat.)

(표준) 부록 A : 선택적 처리
(Annex A(normative) : Optional Process)

A.1. 선택적 처리 1 (Optional Process 1)

아래의 처리 순서는 송신자와 수신자 사이에 미리 정의되어 합의된 뒤 사용될 수 있는 선택적 처리(4.5절 참조)를 규정하고 있다. 이 선택적 처리는 키를 다 찾아보는 공격(exhaustive key search attack)과 평문을 선택하여 하는 공격(chosen plaintext attack)에 대한 MAC의 내성을 증가 시킨다. (The following process specifies an optional process (see clause 4.5) which may be used in accordance with a pre-defined agreement between sender and receiver. This optional process increases the strength of the MAC with respect to exhaustive key search and chosen plaintext attacks.)

이 선택적 처리에서는 두개의 암호화 키가 사용되며, K와 K_1으로 표시된다. (In this optional process two cryptographic keys are used, which are denoted by (K) and (K_1).)

n 비트 블럭 O_q는 4.4절 및 4.5절에 규정된 처리 순서에 의해 키 K를 사용하여 먼저 만들어 진다. (The n-bit block (O_q) is first generated using by (K) in the procedure specified in clauses 4.4 and 4.5.)

두가지의 추가된 단계는 아래와 같다(그림 2 참조) (Two additional steps shall then be followed (see figure 2)) :

a) 키 K_1를 사용하여 출력 O_q를 복호화한 뒤 출력 O'_q를 얻는다. (decipher the output (O_q) using key (K_1) to obtain (O'_q)) ;

b) 키 K를 사용하여 출력 O_q를 암호화한 뒤 출력 O''_q를 얻는다. (encipher the output (O'_q) using key (K) to obtain (O''_q)).

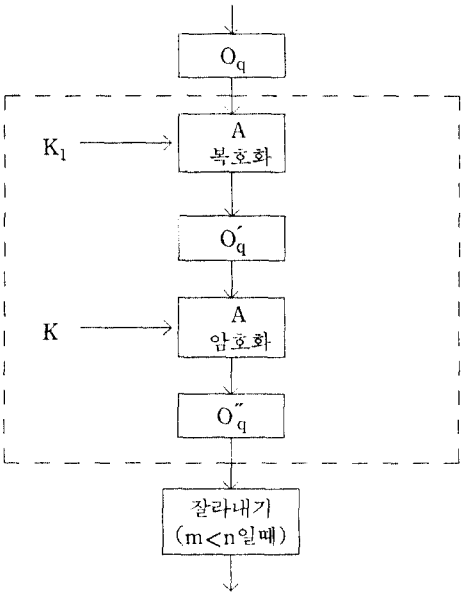


그림 2. 선택적 처리 1
(Figure 2. Optional process 1)

이것으로 선택적 처리는 완성된다. MAC은 4.6절에서 규정한 대로 얻는다. (This completes the optional process. The MAC is obtained as specified in clause 4.6.)

A.2. 선택적 처리 2 (Optional Process 2)

아래의 처리 순서는 송신자와 수신자 사이에 미리 정의되어 합의된 뒤 사용될 수 있는 선택적 처리(4.5절 참조)를 규정하고 있다. 이 선택적 처리는 키를 다 찾아보는 공격(exhaustive key search attack)과 평문을 선택하여 하는 공격(chosen plaintext attack)에 대한 MAC의 내성을 증가시킨다. (The following procedure specifies an optinal process (see clause 4.5) which may be used in accordance with a pre-defined agreement between sender and receiver. This optional process increases the strength of the MAC with respect to exhaustive key search and chosen plaintext attacks.)

이 선택적 처리에서는 두개의 암호화 키가 사용되며, K와 K_1으로 표시된다. 여기서 K_1은 K로 부터 얻을 수도 있다. (In this optional process two cryptographic keys are used, which are denoted by (K) and (K_1), where (K_1) may be derived from (K).)

주 : K로 부터 K_1을 구하는 방법의 한 예는 첫번째 4개의 비트는 보수(complement)를 취하고 매번 다음의 4개의 비트는 K에 있는 것을 그대로 취하는 것을 되풀이 함으로써 K_1을 만든다. (NOTE : An example of how to derive (K_1) from (K) is to complement alternate blocks of four bits of (K) commencing with the first four bits.)

n비트 블럭 O_q는 4.4절 및 4.5절에 규정된 처리 순서에 의해 키 K를 사용하여 먼저 만들어 진다. (The n-bit block (O_q) is first generated using key (K) in the procedure specified in clauses 4.4 and 4.5.)

한개의 추가된 단계는 아래와 같다(그림 3 참조). (An additional step shall then be followed (see figure 3))

a) 키 K_1를 사용하여 출력 O_q를 복호화하여 출력 O_q'를 얻는다. (encipher the output (O_q) using key (K_1) to obtain (O_q').)

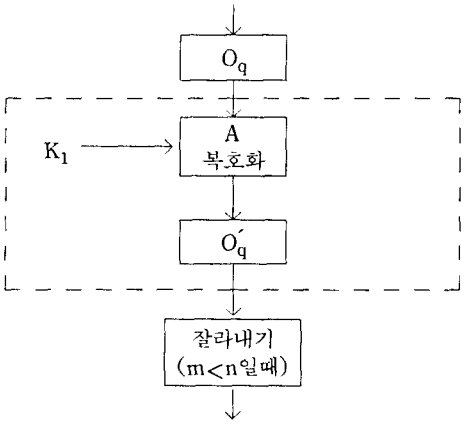


그림 3. 선택적 처리 2
(Figure 3. Optional process 2)

이것으로 선택적 처리는 완성된다. MAC은 4.6절에서 규정한 대로 얻는다. (This completes the optional process. The MAC is obtained as specified in clause 4.6.)

(참고) 부록 B : 참고문헌
(Annex B(informative) : Bibliography)

[1] ISO 7498-2 : 1989, Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2 : Security Architecture.

[2] ISO 8731-1 : 1987, Banking - Approved algorithms for message authentication - Part 1 : DEA.

[3] ISO 9807 : 1991, Banking and related financial services - Requirements for message authentication (retail).

[4] ISO/IEC 9979 : 1991, Data cryptographic techniques - Procedures for the registration of cryptographic algorithms.

[5] ANSI X3.92 : 1981, Data Encryption Algorithm.

[6] ANSI X9.9 : 1986, Financial Institution Message Authentication (Wholesale).

[7] ANSI X9.19 : 1986, Financial Institution Retail Message Authentication.

(참고) 부록 C : 예
(Annex C(informative) : Examples)

이 부록은 DEA(ANSI X3.92 참조)를 사용하여 MAC을 발생시키는 예를 채워 넣기 방법 1과 2 그리고 선택적 처리 방법 1과 2에 대하여 보였다. 평문은 7비트의 ASCII코드로 "Now◆is◆the◆time◆for◆all◆"과 "Now◆is◆the◆time◆for◆it◆"이다. 여기서 "◆"는 공백을 뜻한다. 첫번째의 평문은 채워 넣기 방법 1이 사용된다면 채워 넣는 것이 전혀 필요없게 된다. 키는 0123456789ABCDEF이다. 선택적 처리 1에서 사용되는 키 K_1은 FEDCBA9876543210로 선택되어졌고, 한편 선택적

처리 2에서 사용되는 키 K_1은 부록 'A.2'의 주에 따라 만들어졌다. (This Annex presents examples of the generation of a MAC employing the DEA (see ANSI X3.92) for padding methods 1 and 2 as well as optional processes 1 and 2. The plaintexts are the 7-bit ASCII codes for "Now◆is◆the◆time◆for◆all◆" and "Now◆is◆the◆time◆for◆it◆", where "◆" denotes a blank. The first plaintext does not require any padding if padding method 1 is chosen. The key (K) is 0123456789ABCDEF. The key (K_1) in optional process 1 was chosen to be FEDCBA9876543210, while the key (K_1) in optional process 2 was derived according to the note in annex 'A.2'.)

C.1. 채워 넣기 방법 1 (Padding method 1)

예 1 (Example 1) : Now◆is◆the◆time◆for◆all◆

key (K)	01	23	45	67	89	AB	CD	EF
D_1	4E	6F	77	20	69	73	20	74
D_2	68	65	20	74	69	6D	65	20
D_3	66	6F	72	20	61	6C	6C	20
$I_1=D_1$	4E	6F	77	20	69	73	20	74
O_1	3F	A4	0E	8A	98	4D	48	15
$I_2=O_1 \oplus D_2$	57	C1	2E	FE	F1	20	2D	35
O_2	0B	2E	73	F8	8D	C5	85	6A
$I_3=O_2 \oplus D_3$	6D	41	01	D8	EC	A9	E9	4A
O_3	70	A3	06	40	CC	76	DD	8B

만일 선택적 처리가 사용되지 않는다면 MAC은 O_3의 맨 왼쪽의 m개의 비트들로 구성된다. (If no optional process is used the MAC consists of the m leftmost bits of (O_3).)

선택적 처리 1 (Optional process 1)

key (K_1)	FE	DC	BA	98	76	54	32	10
O'_3	B4	8D	36	ED	7A	D5	69	4F
O''_3	A1	C7	2E	74	EA	3F	A9	B6

MAC은 O''_3의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O''_3).)

선택적 처리 2 (Optional process 2)

key (K_1)	F1	D3	B5	97	79	5B	3D	1F
O'_3	10	F9	BC	67	A0	3C	D5	D8

MAC은 O'_3의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O'_3).)

예 2 (Example 2) : Now◆is◆the◆time◆for◆it

key (K)	01	23	45	67	89	AB	CD	EF
D_1	4E	6F	77	20	69	73	20	74
D_2	68	65	20	74	69	6D	65	20
D_3	66	6F	72	20	69	74	00	00
$I_1=D_1$	4E	6F	77	20	69	73	20	74
O_1	3F	A4	0E	8A	98	4D	48	15
$I_2=O_1 \oplus D_2$	57	C1	2E	FE	F1	20	2D	35
O_2	0B	2E	73	F8	8D	C5	85	6A
$I_3=O_2 \oplus D_3$	6D	41	01	D8	E4	B1	85	6A
O_3	E4	5B	3A	D2	B7	CC	08	56

만일 선택적 처리가 사용되지 않는다면 MAC은 O_3의 맨 왼쪽의 m개의 비트들로 구성된다. (If no optional process is used the MAC consists of the m leftmost bits of (O_3).)

선택적 처리 1 (Optional process 1)

key (K_1)	FE	DC	BA	98	76	54	32	10
O_3	32	8A	C7	8B	A1	CA	0B	3F
O''_3	2E	2B	14	28	CC	78	25	4F

MAC은 O''_3의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O''_3).)

선택적 처리 2 (Optional process 2)

key (K_1)	F1	D3	B5	97	79	5B	3D	1F
O_3'	21	5E	9C	E6	D9	1B	C7	FB

MAC은 O_3'의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O_3').)

C.2. 채워 넣기 방법 2 (Padding method 2)

예 1 (Example 1) : Now◆is◆the◆time◆for◆all◆

key (K)	01	23	45	67	89	AB	CD	EF
D_1	4E	6F	77	20	69	73	20	74
D_2	68	65	20	74	69	6D	65	20
D_3	66	6F	72	20	61	6C	6C	20
D_4	80	00	00	00	00	00	00	00
$I_1=D_1$	4E	6F	77	20	69	73	20	74
O_1	3F	A4	0E	8A	98	4D	48	15
$I_2=O_1\oplus D_2$	57	C1	2E	FE	F1	20	2D	35
O_2	0B	2E	73	F8	8D	C5	85	6A
$I_3=O_2\oplus D_3$	6D	41	01	D8	EC	A9	E9	4A
O_3	70	A3	06	40	CC	76	DD	8B
$I_4=O_3\oplus D_4$	F0	A3	06	40	CC	76	DD	8B
O_4	10	E1	F0	F1	08	34	1B	6D

만일 선택적 처리가 사용되지 않는다면 MAC은 O_4의 맨 왼쪽의 m개의 비트들로 구성된다. (If no optional process is used the MAC consists of the m leftmost bits of (O_4).)

선택적 처리 1 (Optional process 1)

key (K_1)	FE	DC	BA	98	76	54	32	10
O_4'	79	53	7F	EE	18	CF	18	93
O''_4	E9	08	62	30	CA	3B	E7	96

MAC은 O''_4의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O''_4).)

선택적 처리 2 (Optional process 2)

key (K_1)	F1	D3	B5	97	79	5B	3D	1F
O_4'	BE	7C	2A	B7	D3	6B	F5	B7

MAC은 O_4'의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O_4').)

예 2 (Example 2) : Now◆is◆the◆time◆for◆it

key (K)	01	23	45	67	89	AB	CD	EF
D_1	4E	6F	77	20	69	73	20	74
D_2	68	65	20	74	69	6D	65	20
D_3	66	6F	72	20	69	74	80	00
$I_1=D_1$	4E	6F	77	20	69	73	20	74
O_1	3F	A4	0E	8A	98	4D	48	15
$I_2=O_1\oplus D_2$	57	C1	2E	FE	F1	20	2D	35
O_2	0B	2E	73	F8	8D	C5	85	6A
$I_3=O_2\oplus D_3$	6D	41	01	D8	E4	B1	05	6A
O_3	A9	24	C7	21	36	14	92	11

만일 선택적 처리가 사용되지 않는다면 MAC은 O_3의 맨 왼쪽의 m개의 비트들로 구성된다. (If no optional process is used the MAC consists of the m leftmost bits of (O_3).)

선택적 처리 1 (Optional process 1)

key (K_1)	FE	DC	BA	98	76	54	32	10
O_3'	7A	71	AF	2F	5D	15	40	A7
O''_3	5A	69	2C	E6	4F	40	41	45

MAC은 O''_3의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O''_3).)

선택적 처리 2 (Optional process 2)

key (K_1)	F1	D3	B5	97	79	5B	3D	1F
O_3'	17	36	AC	1A	61	63	0E	FB

MAC은 O_3'의 맨 왼쪽의 m개의 비트들로 구성된다. (The MAC consists of the m leftmost bits of (O_3').)

□ 著者紹介

李 弼 中 (종신회원, 국제이사)

1951년 12월생
1974년 2월 서울대학교 전자공학과 학사
1977년 2월 서울대학교 전자공학과 석사
1982년 6월 U.C.L.A. System Science, Engineer
1985년 6월 U.C.L.A. Electrical Engineering, Ph.D.
1980년 6월~1985년 8월 : Jet Propulsion Laboratory, Senior Engineer
1985년 8월~1990년 2월 : Bell Communications Research, M.T.S
1990년 2월~현재 : 포항공과대학 전자전기공학과, 부교수.