通信情報保護學會誌
第6卷 第1號, 1996. 3

ISO/IEC JTC1/SC27의 국제표준소개 (11) : ISO/IEC IS 10118-2 정보기술 - 보안기술 - 해쉬함수, 제 2 부 : n-비트 블럭 암호 알고리즘을 이용한 해쉬함수

(Information technology - Security techniques - Hash-functions - Part 2 : Hash-functions using an n-bit block cipher algorithm)

이 필 중*

요 약

작년 6월호에 ISO/IEC 10118의 제 1 부인 개론이 소개되었고 이번호에는 바로 표준이 된 제 2 부를 소개한다. 이 과제는 1990년에 CD(Committee Draft), 1992년에 DIS(Draft IS)가 되었고, 1994년에 IS(International Standard)가 되었으며 1998년에 1차 검토가 있을 예정이다.

1. 범 위 [Scope]

ISO/IEC 10118 제 2 부는 n-비트 블럭 암호 알고리즘을 사용하는 해쉬함수를 상술한다. 이 해쉬함수는 그러한 알고리즘이 이미 구현된 환경에서 구현되기가 적당하다. [This part of ISO/IEC 10118 specifies hash-functions which make use of an n-bit block cipher algorithm. They are therefore suitable for an environment in which such an algorithm is already implemented.]

두 가지 형태의 해쉬함수가 상술된다. 첫 번째는 사용된 알고리즘의 블럭 길이, n 보다 작거나 같은 길이의 해쉬코드를 생성하는 형태이고 두 번째는 2n 보다 작거나 같은 길이의 해쉬코드를 생성하는 형태이다. [Two types of hash-functions are specified. The first provides hash-codes of length smaller than or equal to n, where n is the block-length of the algorithm which is used. The second provides hash-codes of length less than or equal to 2n.]

2. 용어 정의 [Definitions]

ISO/IEC 10118의 제 1 부에서 소개된 용어에 더하여 제 2 부에서는 다음의 용어 정의가

* 포항공과대학 전자전기공학과

적용된다 : [For the purposes of this part of ISO / IEC 10118, and in addition to those presented in part 1, the following definition from ISO / IEC 10116 applies:]

2.1 n-비트 블럭 암호 알고리즘 : 평문블럭과 암호문블럭의 길이가 n 인 블럭 암호 알고리즘 [n-bit block cipher algorithm : A Block cipher algorithm with the property that plaintext blocks and ciphertext blocks are n bits in length.]

3. 참고 문헌 [Normative reference]

아래의 표준 ISO / IEC 10116은 본문의 참고문헌으로서 ISO / IEC 10118의 제 2 부에서 필요한 내용들을 포함하고 있다. 아래의 표준은 본 표준이 발표될 당시에는 타당했다. 모든 표준들은 개정되므로, ISO / IEC 10118의 제 2 부를 사용하려는 단체들은 아래에 명시되어 있는 표준들의 최신 개정본을 찾아 보아야 할 것이다. ISO / IEC의 구성원들은 최신의 국제표준을 계속 관리한다. [The following standard contains provisions which, through reference in this text, constitute provisions of this part of ISO / IEC 10118. At the time of publication, the edition indicated was valid. All standards are subject to revision and parties to agreements based on this part of ISO / IEC 10118 are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.]

[ISO / IEC 10116 : 1991, Information Technology-Security techniques-Modes of operation for an n-bit block cipher algorithm]

4. 기호와 약어 [Symbols and abbreviations]

ISO / IEC 10118의 제 1 부에서 소개된 것들에 더해 제 2 부에서는 다음의 기호와 약어가 적용된다 : [Throughout this part of ISO / IEC 10118 and in addition to those presented in part 1, the following symbols and abbrebiations apply:]

e	n-비트 블럭 암호 알고리즘 (ISO / IEC 10116 참고) [n-bit block cipher algorithm(see ISO / IEC 10116)]
K	알고리즘 e에 쓰이는 키 (ISO / IEC 10116 참고) [Key for the algorithm e (see ISO / IEC 10116)]
eK	알고리즘 e와 키 K를 사용한 암호화과정 (ISO / IEC 10116 참고) [Operation of encipherment using the algorithm e and the key K (see ISO / IEC 10116)]
u or u'	하나의 n-비트 블럭의 알고리즘 e에 대한 키로의 변환 [Transformation of one n-bit block into a key for the algorithm e]
$T_{[left]}$	- n 이 짝수일땐 블럭 T의 가장 왼쪽 n/2 개 비트로 구성된 스트링 - n 이 홀수일땐 블럭 T의 가장 왼쪽 (n+1)/2 개 비트로 구성된 스트링 [- when n is even, the string composed of the n / 2 leftmost bits of the block T -

when n is odd, the string composed of the (n+1)/2 leftmost bits of the block T]

$T_{[right]}$ - n이 짝수일땐 블럭 T의 가장 오른쪽 n/2 개 비트로 구성된 스트링 - n 이 홀수일땐 블럭 T의 가장 오른쪽 (n-1)/2 개 비트로 구성된 스트링 [- when n is even, the string composed of the n/2 rightmost bits of the block T - when n is odd, the string composed of the (n-1)/2 rightmost bits of the block T]

본문에서 "최상위비트"와 "최하위비트"라는 용어는 비트 스트링을 수치값으로 생각한 것으로 블럭의 가장 왼쪽 비트가 최상위 비트가 된다. [In contexts where the terms "most significant bit" and "least significant bit" have a meaning, e.g. where strings of bits are treated as numerical values, then the leftmost bits of a block shall be the most significant.]

5. 요구 사항 [Requirements]

ISO/IEC 10118 제 2 부의 해쉬함수를 사용할 때 선택해야 할 사항들은 [Users who wish to use a hash-function from this part of ISO/IEC 10118 shall select]

n-비트 블럭 암호 알고리즘 e ; [- an n-bit block cipher algorithm e;]

한 개(두 개)의 변환 u (와 u') ; [- one (two) transformation(s) u (and u');]

한 개(두 개)의 초기값 IV (와 IV') ; [- one (two) inintializing value(s) IV (and IV');]

덧붙이기 방법 ; [- a padding method;]

$H(L_H)$의 길이. [- the length of $H(L_H)$.]

선택의 예는 부록 A 에 있다. 사용된 n-비트 블럭 암호 알고리즘은 ISO/IEC 10118 제 2 부에서는 상술하지 않으며 ISO/IEC 9799에서 정의된 것처럼 the Register of Cryptographic Algorithms 또는 그외에서 선택할 수 있다. 더욱이 사용된 알고리즘의 암호학적 특성이 해쉬함수에 결함을 가져올 수 있음을 고려해야 한다. [An example of such a selection is presented in annex A. The n-bit block cipher algorithm to be used is not specified in this part of ISO/IEC 10118 and may be selected from the Register of Cryptographic Algorithms, as defined in ISO/IEC 9799, or from another source. Nonetheless, it should be taken into consideration that a cryptographic property of the algorithm used may introduce some weakness into the resulting hash-function.]

두가지 형태의 해쉬함수는 한 개의 변환 u 또는 두 개의 변환 u, u'를 이용하나, 그 구체적인 것은 사용된 알고리즘에 따라 다르므로 ISO/IEC 제 2 부에서는 설명하지 않는다. 이 알고리즘이 the Register of Cryptographic Algorithms에서 선택되고, 변환 u, u'가 정의되면 사용자는 그것들을 가급적 사용하기를 권해진다. [The two types of hash-functions make use of either one transformation, called u, or two transformations, called u and u', which are not specified in this part of ISO/IEC 10118, as they depend on the algorithm used. If this algorithm has been selected from the Register of Cryptographic

Algorithms, and the transformations u and u' are defined, users are encouraged to use them.]

6. single 길이 해쉬코드 생성 해쉬 함수 [Hash-functions providing a single length hash-code]

6.1 개요 [General]

이 절에서 상술된 해쉬함수는 n 보다 작거나 같은 길이의 해쉬코드를 만든다. [The hash-functions which are specified in this clause provide hash-codes of length LH, where LH is less than or equal to n.]

사용된 변환 u 는 하나의 출력 블럭을 알고리즘 e에 대한 적절한 LK-비트 키로 변환시킨다. u의 구체적 내용은 ISO/IEC 10118 제 2 부의 범위를 벗어난다. [One transformation denoted by u is used, the purpose of which is to transform an output block into a suitable LK-bit key for the algorithm e. The specification of u is beyond the scope of this part of ISO/IEC 10118.]

6.2 해쉬화 과정 [Hashing operation]

e를 n-비트 블럭 암호 알고리즘, IV를 길이가 n인 초기값이라 한다. IV 는 미리 설정된 집합에서 선택되나 그 구체적인 것은 ISO/IEC 제 2 부의 범위를 벗어난다. [Let e be an n-bit block cipher algorithm and IV be an initializing value of length n. IV shall be selected from a prescribed set of fixed values, the specification of which is beyond the scope of this part of ISO/IEC 10118.]

데이타 D의 해쉬코드 H는 네 단계를 거쳐 계산된다. [The hash-code H of the data D is calculated in four steps.]

6.2.1 단계 1 (분할) [Step 1 (splitting)]

데이타 D를 n-bit 블럭들 D_1, D_2, ...로 나눈다. [The data D are split into n-bit blocks D_1, D_2, ...]

주의 - 마지막 블럭은 완전하지 않을 수 있다(즉, 길이가 n 보다 작을 수 있다). [NOTE - The last block may be incomplete(i.e., its length may be less than n).]

6.2.2 단계 2 (덧붙이기) [Step 2 (padding)]

마지막 블럭의 길이가 n이 되도록 데이타를 덧붙인다. 덧붙이는 방법은 ISO/IEC 10118 제 2 부의 영역을 벗어난다. 실제 예는 ISO/IEC 10118 제 1 부의 부록 B에 있다. [The data are padded in order to ensure that the last block has length n. The padding method is beyond the scope of this part of ISO/IEC 10118. Examples of such a method are presented in annex B of part 1 of ISO/IEC 10118.]

6.2.3 단계 3 (반복) [Step 3 (iteration)]

D_1, D_2, ..., D_q는 덧붙이기를 한 후의 n-비트 블럭들이다. H_0를 IV로 설정한다. 블럭 H_1, H_2, ... H_q는 다음 과정을, i를 1에서 q까지에 대하여, 반복적으로 수행하여 계산된다: [Let D_1, D_2, ..., D_q be the n-bit blocks of the data after padding. Set H_0 equal to IV. The output blocks H_1, H_2, ... H_q are calculated

iteratively in the following way, for i from 1 to q:]

$$K_i = u(H_{i-1})$$
$$H_i = eK_i(D_i) \oplus D_i$$

단계 3을 그림 1에 나타냈다. [Step 3 is shown in figure 1.]

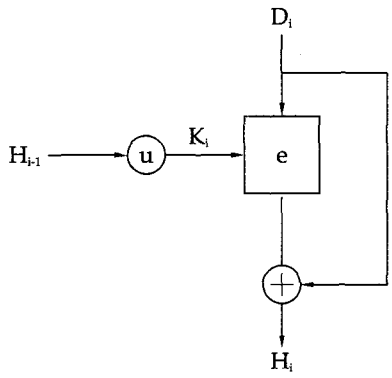

Figure 1 - Iteration of single length hashing operation

6.2.4 단계 4 (자르기) [Step 4 (truncation)]

해쉬코드 H는 마지막 출력 블럭 H_q의 가장 왼쪽 LH 비트들로 생성한다. [The hash-code H is derived by taking the leftmost LH bits of the final output block H_q.]

7. double 길이 해쉬코드 생성 해쉬 함수 [Hash-functions providing a double length hash-code]

7.1 개요 [General]

이 절에서 상술된 해쉬함수는 2n보다 작거나 같은 길이의 해수코드 L_H를 생성한다. [The hash-functions which are specified in this clause provide hash-codes of length L_H, where L_H is less than or equal to 2n.]

사용된 두 개의 변환 u, u' 각각은 하나의 출력 블럭을 알고리즘 e에 대한 적절한 LK-비트 키로 변환시킨다. u, u'의 내용은 ISO/IEC 10118 제 2 부의 범위를 벗어난다. 그러나, u, u' 의 선택은 해쉬함수의 보안에 중요하다. 구체적으로, Ki 와 K'i 값이 항상 다르게 u와 u'를 선택한다. [Two transformations denoted by u and u' are used, the purpose of which is to transform an output block into a suitable LK-bit key for the algorithm e. The specificaion of u and u' is beyond the scope of this part of ISO/IEC 10118. However, it should be taken into consideration that the selection of u and u' is important for the security of the hash-function. In particular, u and u' should be chosen such that Ki and K'i are always distinct.]

7.2 해쉬화 과정 [Hashing operation]

e를 n-비트 블럭 암호 알고리즘, IV와 IV'를 길이가 n인 초기값들이라 한다. IV와 IV'는 IV는 미리 설정된 집합에서 선택되나 그 구체적인 것은 ISO/IEC 제 2 부의 범위를 벗어난다. 더욱이 u(IV)와 u'(IV') 이 다르도록 IV 와 IV' 이 선택되어야 한다. [Let e be an n-bit block cipher algorithm, IV and IV' be two initializing values each of length n. IV and IV' shall be selected from a prescribed set of fixed values, the specification of which is beyond the scope of this part of ISO/IEC 10118. Moreover, IV and IV' shall be selected so that u(IV) and u'(IV') are distinct.]

데이타 D의 해쉬코드 H는 네 단계를 거쳐 계산된다. [The hash-code H of the data D is calculated in four steps.]

7.2.1 단계 1 (분할) [Step 1 (splitting)]

데이타 D를 블록들 n-bit D_1, D_2, ... 로 나눈다. [The data D are split into n-bit blocks D_1, D_2, ...]

주의 - 마지막 블럭은 완전하지 않을 수 있다(즉, 길이가 n 보다 작을 수 있다). [NOTE - The last block may be incomplete (i.e., its length may be less than n).]

7.2.2 단계 2 (덧붙이기) [Step 2 (padding)

마지막 블럭의 길이가 n이 되도록 데이타를 덧붙인다. 덧붙이는 방법은 ISO/IEC 10118 제 2 부의 범위를 벗어난다. 실제 예는 ISO/IEC 10118 제 1 부의 부록 B에 있다. [The data are padded in order to ensure that last block has length n. The padding method is beyond the scope of this part of ISO/IEC 10118. Examples of such a method are presented in annex B of part 1 of ISO/IEC 10118.]

7.2.3 단계 3 (반복)[Step 3 (iteration)]

D_1, D_2, ..., D_q는 덧붙기를 한 후 n-비트 블럭들이다. H_0와 H'_0를 각각 IV와 IV'로 설정한다. 출력 블럭 H_1, H_2, ...H_q와 H'_1, H'_2, ..., H'_q는 다음 과정을, i를 1에서 q까지에 대하여, 반복적으로 수행하여 계산된다: [Let D_1, D_2, ..., D_q be the n-bit blocks of the data after padding. Set H_0 and H'_0 equal to IV and IV' respectively. The output blocks H_1, H_2, ..., H_q and H'_1, H'_2, ..., H'_q are calculated iteratively in the following way, for i from 1 to q:]

$$K_i = u(H_{i-1}) \text{ an } K'_i = u'(H'_{i-1})$$
$$T_i = eK_i(D_i) \oplus D_i \text{ and } T'_i$$
$$= eK'_i(D_i) \oplus D_i$$
$$H_i = T_{i[left]} \,||\, T'_{i[right]} \text{ and } H'_i$$
$$= T'_{i[left]} \,||\, T_{i[right]}$$

단계 3을 그림 2에 나타냈다. [Step 3 is shown in figure 2.]

7.2.4 단계 4 (자르기) [Step 4 (truncation)]

L_H가 짝수일 때, 해쉬코드는 H_q의 $L_H/2$개 가장 왼쪽 비트들과 H'_q의 $L_H/2$개 가장 왼쪽 비트들을 연접하여 생성한다. L_H가 홀수일 때, 해쉬코드는 H_q의 $(L_H+1)/2$개 가장 왼쪽 비트들과 H'_q의 $(L_H-1)/2$개 왼쪽 비트들을 연접하여 생성한다. [In case L_H is even, the hash-code is the concatenation of the $L_H/2$ leftmost bits of H_q and the $L_H/2$ leftmost bits H'_q. In case L_H is odd, the hash-code is the concatenation of the $(L_H+1)/2$ leftmost bits of H_q and the $(L_H-1)/2$ leftmost bits H'_q.]

부록 A [Annex A]

(참고) [(informative)]

DEA의 사용 [Use of DEA]

A.1 개요 [General]

이 부록은 ISO/IEC 10118의 제 2 부에서 상술된 해쉬화과정과 관련하여 DEA(ANSI X3.92)를 이용하는 방법을 설명한다. DEA는

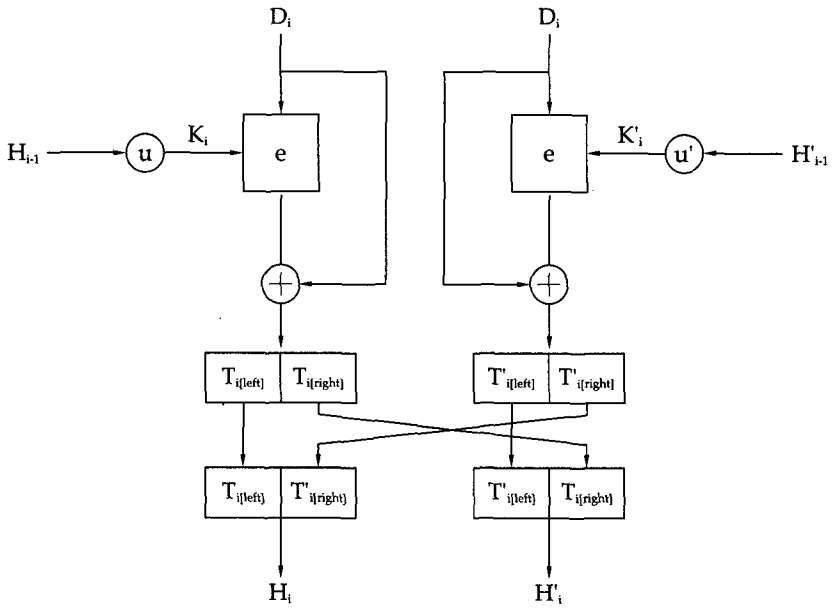

Figure 2 - Iteration of double length hashing operation

또한 DES란 이름으로 알려져 있다. 이러한 방법들은 [3]이나 [4]에 기술되었다(부록 C 참고). [This annex presents a way of using the DEA(ANSI X3.92) in conjuction with hashing operations specified in this part of ISO/IEC 10118. The DEA is also known under the name DES. These methods have been described in [3] or [4](see annex C).]

DEA에 쓰이는 인수들은 n = 64, L_K = 56 이다. [The parameters for DEA are n = 64 and L_K = 56.]

A.2 Clause 6.2 해쉬화 과정 [hashing operation]

IV는 16진수로 표시된 '5252525252525252' 로 설정한다. [IV should be equal to '5252525252525252' (in hexadecimal notation).]

변환 u는 다음과 같다. X = $x_1x_2...x_{64}$ 를 64-비트 스트링 X의 이진표현식이라 하자. 그러면 Y = u(X)는 X에서 x_8, x_{16}, x_{24}, x_{32}, x_{40}, x_{48}, x_{56}, x_{64} 비트들을 없앤 후 x_2와 x_3를 '10'로 설정한 값이다. 결과는 다음과 같다 : Y = x_1'10'$x_4x_5x_6x_7x_9x_{10}...x_{63}$. [The transformation u should be chosen as follows. Let X = $x_1x_2...x_{64}$ be the binary decomposition of a 64-bit string X. Then Y = u(X) is the string obtained after removing the bits x_8, x_{16}, x_{24}, x_{32}, x_{40}, x_{48}, x_{56}, x_{64} of X and forcing the bits x_2 and x_3 to the values '10'. The result is : Y = x_1'10'$x_4x_5x_6x_7x_9x_{10}...x_{63}$.]

주의 - 결과는 해쉬함수이나 충돌회피성을 만족하지 않을 수 있다. [NOTE - The resulting function is believed to be a hash-function but may not be collision-resistant.]

A.3 Clause 7.2 해쉬화 과정 [hashing operation]

IV는 A.3절과 동일하다. [IV should be the same as in clause A.3.

IV'는 16진수로 표시된 '2525252525252525'로 설정한다. [IV' should be equal to '2525252525252525' (in hexadecimal notation).

변환 u는 A.3 절과 동일하고 변환 u'는 다음과 같다. $X = x_1x_2...x_{64}$를 64-비트 스트링 X의 이진표현식이라 하자. 그러면 $Y = u'(X)$는 X에서 x_8, x_{16}, x_{24}, x_{32}, x_{40}, x_{48}, x_{56}, x_{64} 비트들을 없앤 후 x_2와 x_3를 '01'로 설정한 값이다. 결과는 다음과 같다 : $Y = x_1'01'x_4x_5x_6x_7x_9x_{10}...x_{63}$. [The transformation u should be the same as in clause A.3 and the transformation u' should be chosen as follows. Let $X = x_1x_2...x_{64}$ be the binary decomposition of a 64-bit string X. Then $Y = u'(X)$ is the string obtained after removing the bits x_8, x_{16}, x_{24}, x_{32}, x_{40}, x_{48}, x_{56}, x_{64} of X and forcing the bits x_2 and x_3 to the values '01'. The result is : $Y = x_1'01'x_4x_5x_6x_7x_9x_{10}...x_{63}$.]

주의 - 결과는 해쉬함수라고 믿어진다. 또한 고정된 키로 2^{55}번의 DES 암호화를 수행했을 때 계산상 실행 불가능한 충돌회피 해쉬함수로 믿어진다. [4] 참고. [NOTE - The resulting function is believed to be a hash-function. It is also believed to be a collision-resistant hash-function in environments where performing 2^{55} DES encipherment operations with a fixed key is deemed to be computationally infeasible. See also [4].]

부록B [Annex B]
(참고) [(Informative)]

예 [Examples]

B.1 개요 [General]

이 부록은 ISO/IEC 10118 제 2 부 부록 A에 설명된 해쉬코드계산과 ISO/IEC 10118 제 2 부 부록 B에 설명된 덧붙이기 방법에 대한 예를 보인다. [This annex gives examples for the computation of a hash-code using the hash-functions specified in annex A of this part of ISO/IEC 10118 and the padding methods specified in annex B of ISO/IEC 10118.]

데이타스트링은 "Now__is__the__time__for__all__"에 대한 parity 비트 없는 16진수로 표시된 7비트 ASCII 코드이다("__"는 여백을 의미) : [The data string is the 7-bit ASCII code(no parity) for "Now__is__the__time__for__all__", where "__" denotes a blank, in hexadecimal notation:]

'4E6F772069732074686520746965206F7220616C6C20'

B.2 Clause A.2 single 길이 해쉬화 과정 [hashing operation(single length)]

덧붙이기 방법 1 [Padding method 1]

i	D_i	H_{j-1}	H_j
1	4E6F772069732074	5252525252525252	858A260F7391482D

2	68652074696D6520	858A260F7391482D	BDE06E66A0454081
3	666F7220616C6C20	BDE06E66A0454081	FF87B67E29BB87B1

덧붙이기 방법 2 [Padding method 2]

i	D_i	H_{j-1}	H_j
1	4E6F772069732074	5252525252525252	858A260F7391482D
2	68652074696D6520	858A260F7391482D	BDE06E66A0454081
3	666F7220616C6C20	BDE06E66A0454081	FF87B67E29BB87B1
4	8000000000000000	FF87B67E29BB87B1	D992E6CBDFD9BA81

B.3 Clause A.3 double 길이 해쉬화과정 [hashing operation(double length)]

덧붙이기 방법 1 [Padding method 1]

i	D_i	H_{j-1}	H'_j
1	4E6F772069732074	5252525252525252	2525252525252525
2	68652074696D6520	858A260FFD4873A8	49771DD37391482D
3	666F7220616C6C20	B002740352F7CF4F	CFE8087E1B93CCB2

i		H_{j-1}	H'_j
1		858A260FFD4873A8	49771DD37391482D
2		B002740352F7CF4F	CFE8087E1B93CCB2
3		42E50CD224BACEBA	760BDD2BD409281A

덧붙이기 방법 2 [Padding method 2]

i	D_i	H_{j-1}	H'_j
1	4E6F772069732074	5252525252525252	2525252525252525
2	68652074696D6520	858A260FFD4873A8	49771DD37391482D
3	666F7220616C6C20	B002740352F7CF4F	CFE8087E1B93CCB2
4	8000000000000000	42E50CD224BACEBA	760BDD2BD409281A

i		H_{j-1}	H'_j
1		858A260FFD4873A8	49771DD37391482D
2		B002740352F7CF4F	CFE8087E1B93CCB2
3		42E50CD224BACEBA	760BDD2BD409281A
4		2E4679B5ADD9CA75	35D87AFEAB33BEE2

부록C [Annex C]
(참고) [(informative)]
Bibliography

[1] ISO / IEC 9979 : 1991 Information processing - Data cryptographic techniques - ISO standard procedures for the registration of cryptographic algorithms.

[2] ANSI X3.92 - 1981 : American National Standard for Information Systems - Data Encryption Algorithm.

[3] S.M.Matyas : Key Processing with control vectors, J. of Cryptology, Vol. 3, n°2, 1991, pp. 113-136.

(본 원고를 정리하는데에 수고를 해 준 대학원생 김 용덕에게 감사를 표한다.)

□ 著者紹介

이 필 중(李 弼 中) 종신회원

1951년 12월 30일생
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