韓國通信情報保護學會誌
第1卷, 第1號, 1991. 4

컴퓨터 通信網에서의 暗號키 生成, 分配, 그리고 管理方式

廉 興 烈*

1. 서 론

컴퓨터의 보급이 확대됨에 따라 지형적으로 멀리 분산되어 있는 하드웨어 및 소프트웨어 자원의 효율적 이용을 목적으로 한 컴퓨터 통신망(computer communication network)이 도입되었다. 이에 따라 악의적이고 불법적인 사용자가 컴퓨터 통신망을 통해 통신되는 도중의 정보에 또는 컴퓨터에 저장되어 있는 정보에 고의적으로 접근하여 이를 악용할 가능성이 있다는 새로운 문제점을 야기하였다.

이러한 문제점을 해결하는 가장 효율적이고 경제적인 방법은 암호기법(cryptography)을 이용하여 정보를 보호하는 것이다[1,2]. 지금까지 개발된 암호시스템은 암호화 및 복호시에 동일한 암호키를 사용하는 대칭형 암호시스템(symmetric cryptographic system)과 서로 다른 암호키를 사용하는 비대칭형 암호시스템(asymmetric cryptographic system) 등으로 대별될 수 있다[2,4,5].

전자는 DES(data encryption standard) 및 FDEA (fast data encipherment algorithm) 등이 있고[3], 후자는 Merkle-Hellman 공개키 암호시스템(public key cryptographic system)과 RSA(Rivest-Shamir-Adleman) 암호시스템 등이 있다[4,5]. 암호시스템의 보안성은 암호키의 보안에 커다란 영향을 받으므로 암호키의 생성, 분배, 그리고 관리는 정보의 보호를 목적으로 한 암호학 분야 중 매우 중요한 연구 분야가 되었다.

본고에서는 먼저 대칭형 암호시스템과 비대칭형 암호시스템에 대한 암호키 생성 및 저장기법을 소개하고, 어떻게 암호화 및 복호화를 수행하는 통신노드까지 효율적으로 생성된 암호키를 분배하는 기법을 분석하며, 컴퓨터 통신망에서 통신 보안(COMSEC ; communication security) 및 파일 보안(FILESEC ; file security)을 위한 안전한 암호키 분배, 암호키 변경(updata) 그리고 분배된 암호키 인증(authentication) 등을 포함하는 암호키 관리 기법을 분석 및 제안한다.

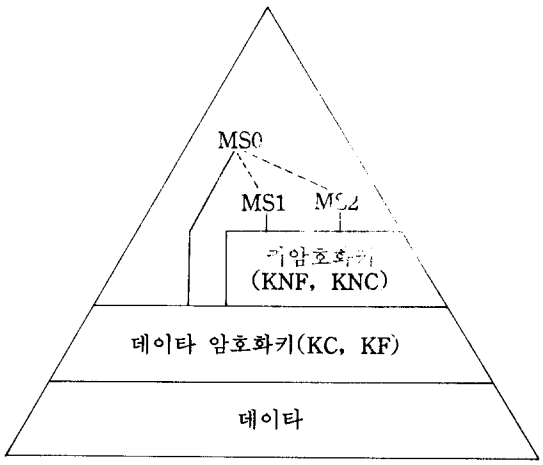
2. 암호키의 생성, 분배, 그리고 관리방식

2.1. 암호키의 정의[6]

컴퓨터 통신망은 일반적으로 자국의 호스트 및 터미날들과 상대방의 호스트 및 터미날들로 구성되어 있다. 컴퓨터 시스템의 보안을 위한 암호시스템은 하드웨어와 소프트웨어의 결합으로 구성되어 실제로 암호 알고리즘을 수행하는 암호장치

* 정회원, 順天鄕大學校 工科大學 電子工學科

(CF : cryptographic facility), 암호키 생성 및 분배, 갱신 등의 기능을 수행하는 암호키 관리 프로그램(key manager), 그리고 생성된 암호키를 적당한 암호키로 암호화하여 보관하는 암호키 저장장치(CKDS : cryptographic key data set)로 구성되어 있다.



여기서, KC : 1차 통신용 암호화키
KF : 1차 파일용 암호화키
KNC : 2차 통신용 키암호화키
KNF : 2차 파일용 키암호화키

그림 1. 암호키 계위

공개키 암호시스템에서의 암호키는 암호(encryption)에 필요한 공개키(public key)와 복호화(decryption)에 요구되는 비밀키(secret key)로 구분된다. 공개키는 키 분배센타(key distribution center)에 등록, 공개하는 암호화용 암호키이고, 비밀키는 자기 자신만이 간직하고 있는 복호용 암호키이다. 대칭형 암호시스템을 이용한 컴퓨터 통신을 위한 암호키는 그림 1과 같이 마스터키(master key), 키 암호화키(key encrypting key), 데이타 암호화키(data encrypting key) 등으로 계층화되어 있다[6].

데이타 암호화키는 데이타를 암호하기 위한 1차 암호키이며, 키 암호화키는 다시 데이타 암호화키를 보호하기 위한 2차 암호키이다. 마스터키는 컴퓨터 통신망에서 통신 및 파일보안 등과 같은 서로 다른 용도의 암호 시스템간의 독립성(independence)과 고립성(isolation)을 보장하기 위해 호스트 마스터키(MS0), 1차 및 2차 변형 마스터키인 1차 변형 마스터키(MS1), 2차 변형 마스터키(MS2)로 구분되며[1], 주로 1차 및 2차 암호화키를 보호하기 위한 암호키이다.

동상적으로 MS1과 MS2는 각각 MS0의 몇개의 비트를 반전하여 생성한다. MS0는 데이타 암호화키를 보호하기 위한 암호화키로 MS1과 MS2는 통신보안 및 파일보안을 위한 키암호화키를 보호하기 위한 암호키로서 이용된다. MS0는 CF내에 저장되어 있으며, CF에 일단 MS0가 입력되고 나면 MS1과 MS2는 자동적으로 CF내에서 생성된다. 데이타 암호화키는 다시 응용분야에 따라 1차 통신용 암호화키(primary communication key : KC, KS)와 1차 파일보호용 암호화키(primary file key : KF)로 나누어지며, 키 암호화키는 2차 통신용 암호화키(secondary communication key : KNC)와 2차 파일보호용 암호화키(secondary file key : KNF)로 구분되어 이용되고 있다.

일반적으로 데이타 암호화키는 MS0로 암호화되어 E_{MS0}(KC or KF)의 형태로, 키암호화키는 MS1 또는 MS2로 암호화되어 E_{MS1}(KNC or KNF) 또는 E_{MS2}(KNC or KNF) 형태로 컴퓨터내의 암호키 저장장치에 보관되어 있다가 필요한 경우 이용된다. 호스트의 CF는 일반적으로 마스터키 저장 레지스터와 암호 및 복호 기능을 수행하는 암호 및 복호부로 구성된다. 컴퓨터의 호스트에서 키 관리 및 데이타 보호를 위해 요구되는 기본 암호 동작은 입력변수(input parameter)와 출력변수(output parameter) 영역을 갖으며, 이는 SMK(set master key), EMK(enciper under master key), RFMK (reenciper from master key to other key), RTMK (reenciper from other key to master key), ECPH (encipher data), 그리고 DCPH(decipher data) 등의 6가지로 정의되며, 이들은 CF내에서 수행된다.

SMK는 CF내의 마스터키 저장 레지스터에 MS0를 저장하는 기본 동작, EMK는 암호키(K)를 MS0

로 암호화하는 기본 동작, RFMK는 $E_{MS1}(KN)$와 $E_{MS0}(K)$를 입력받아 $E_{KN}(K)$를 출력하는 기본동작, RTMK는 $E_{MS2}(KN)$과 $E_{KN}(K)$를 입력받아 $E_{MS0}(K)$를 출력하는 기본동작, ECPH는 $E_{MS0}(K)$와 M를 입력받아 $E_K(M)$를 출력하는 기본동작, 그리고 DCPH는 $E_{MS0}(K)$와 $E_K(M)$를 입력받아 M을 출력하는 기본동작이다.

(1) SMK ; {MS0}
(2) EMK ; $\{K\} \rightarrow E_{MS0}(K)$
(3) RFMK ; $\{E_{MS1}(KN), E_{MS0}(K)\} \rightarrow E_{KN}(K)$
(4) RTMK ; $\{E_{MS2}(KN), E_{KN}(K)\} \rightarrow E_{MS0}(K)$
(5) ECPH ; $\{E_{MS0}(K), M\} \rightarrow E_K(M)$
(6) DCPH ; $\{E_{MS0}(K), E_K(M)\} \rightarrow M$

상기의 암호 기본 동작은 CF내에서 수행되며, 암호키를 MS0, MS1, MS2로 암호화 하여 입출력하는 것은 투명한 형태의 암호키가 CF 밖에서 존재하는 것을 방지하기 위해서이다.

터미날에서 키 관리 및 데이타 보호를 위해 요구되는 기본 암호동작은 입력 변수(input parameter)와 출력변수(output parameter) 영역을 가지며, 이는 LKD(load key direct), WMK(write master key), DECK(decipher key), ENC(encipher data), 그리고 DEC(decipher data) 등의 5가지로 정의되며, 이들은 터미날의 CF에서 수행된다. 터미날 CF는 크게 터미날 마스터키 저장 레지스터, 동작키 저장 레지스터, 암호기능을 수행하는 암호 및 복호부로 구성된다. LKD는 터미날 CF내의 동작 암호키 저장 레지스터에 K를 저장하는 기본 동작, WMK는 KN을 터미날 CF내의 마스터키 저장 레지스터에 터미날 마스터키를 저장하는 암호동작, DECK는 $E_{KN}(K)$를 입력받아 K를 동작 암호키 저장 레지스터에 입력하는 기본 동작, ENC는 M를 입력받아 $E_K(M)$를 출력하는 기본동작, 그리고 DEC는 $E_K(M)$를 입력받아 M를 출력하는 기본동작이다.

(1) LKD ; {K}
(2) WMK ; {KN}
(3) DECK ; $\{E_{KN}(K)\}$
(4) ENC ; $\{M\} \rightarrow E_K(M)$
(5) DEC ; $\{E_K(M)\} \rightarrow M$

상기의 기본 동작은 터미날 CF에서 수행된다.

2.2 암호키 생성 및 저장[8]

암호키의 생성은 암호시스템이 요구하는 암호키들을 생성하는 과정이다. 대칭형 암호시스템의 암호키는 그림 1과 같이 마스터키로서 MS0/MS1/MS2, 키 암호화키, 그리고 데이타 암호화키 등이 있다. 키 암호화키는 데이타 암호화키를 보호하기 위한 암호키로서 암호시스템이 초기화되는 과정에 생성되며, 수 주 또는 수 개월의 수명을 지닌 비교적 오랜시간 동안 변하지 않는 암호키이다.

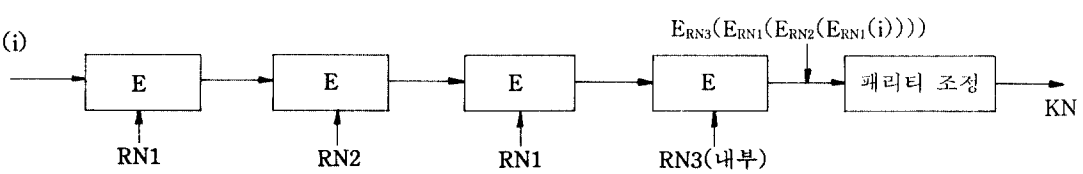


그림 2. 전형적인 키암호화키 생성절차

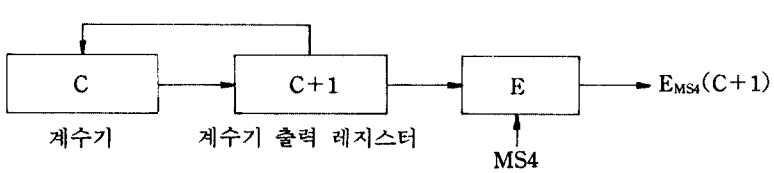


그림 3. 데이타 암호화키 생성 절차

데이타 암호화키는 시스템이 정상적인 동작중에 동적(dynamic-generated key)으로 생성되며, 통신 보안을 위한 1차통신용 암호키는 수분 또는 수시간 동안에만 존속되며 파일보안을 위한 1차 파일용 암호키는 파일이 존재하는 동안에 존재한다. 호스트 마스터키는 암호키를 총괄적으로 관리하는 암호키로서, 이의 생성은 관리자의 전화번호나 생년월일 등에 기초하여 생성되면 않되고, 동전던지기나 주사위 던지기 등의 랜덤 프로세스에 의해 생성되어야 한다. 암호시스템에는 다수의 키 암호화키가 존재해야 하므로 이들중 하나가 알려지더라도 나머지 키 암호화키는 알려지지 않는 암호키 생성 알고리즘을 설정해야 한다. 키 암호화키 생성을 위한 대표적인 알고리즘은 DES 암호 알고리즘의 암호 및 복호화 알고리즘, 외부에서 랜덤프로세스 과정의 결과인 임의의 시퀀스(RN1, RN2), 그리고 시스템 내부의 TOD(time of date) 클럭에 기초한 시퀀스(RN3)를 이용한 그림 2와 같은 과정을 통해 생성될 수 있다. 여기서 i는 생성되는 암호키의 일련번호이다. 즉 4개의 DES 암호화 과정과 1개의 패리티 조정과정으로 구성되어 있다. 상기의 과정을 여러번 반복하면 여러개의 키 암호화키를 생성할 수 있다.

데이타 암호화키는 pseudo-random number 생성기에 의해서 생성되며 데이타 암호화키 값 자체가 투명하게 나타남으로 인해 악의의 침입자에 의해 암호키가 탈로날 가능성을 방지하기 위하여 pseudo-random number 생성기 출력을 $E_{MS0}(k)$로 간주한다. 데이타 암호화키의 대표적인 생성과정은 그림 3과 같이 64비트 계수기와 데이타 암호화키 생성을 위한 전용 4차 변형 마스터키(MS4)를 이용하여 생성될 수 있다. 새로운 데이타 암호화키가 생성될 때마다 계수기의 내용은 하나씩 증가되며 계수기의 출력은 MS4에 의해 암호화되어 이 출력이 $E_{MS0}(KS)$ 형태의 데이타 암호화키가 된다.

비대칭형 암호시스템중 대표적인 암호시스템인 RSA 암호시스템과 MH(Merkle-Hellman) 암호시스템을 들 수 있다. 이중 RSA 암호시스템의 암호키는 다음과 같은 과정을 통해 생성된다[4].

1. 큰 값의 소수 p와 q를 선택하여 N(=pq)를 계산한다.

2. (p−1)과 (q−1)의 최소공배수(least common multiple) L을 계산하여 L과 서로소(relatively prime) 관계인 e를 먼저 임의로 정한다.

3. ed=1 (mod L)의 해를 구하여 d를 구한다.

4. e를 공개키로 공개하고 d를 비밀키로 보관한다.

그리고 MH 암호시스템의 암호키는 다음과 같은 과정을 통해 생성된다[5].

1. $W_i > \sum_{j=1}^{i-1} W_j$을 만족하도록 길이가 n인 벡타 $\mathbf{W}=(W_1, W_2, \cdots, W_n)$를 결정한다.

2. $q > \sum_{j=1}^{n} W_j$이면서 소수인 q를 임의로 결정하고, 그리고 r을 임의로 선택한다.

3. 다음의 방정식을 이용하여 공개 키 $\mathbf{K}=(k_1, k_2, \cdots, k_n)$을 구한다.

$$k_i = W_i \cdot r \pmod{q}, \quad \text{for } i=1, \cdots, n \quad \cdots\cdots (1)$$

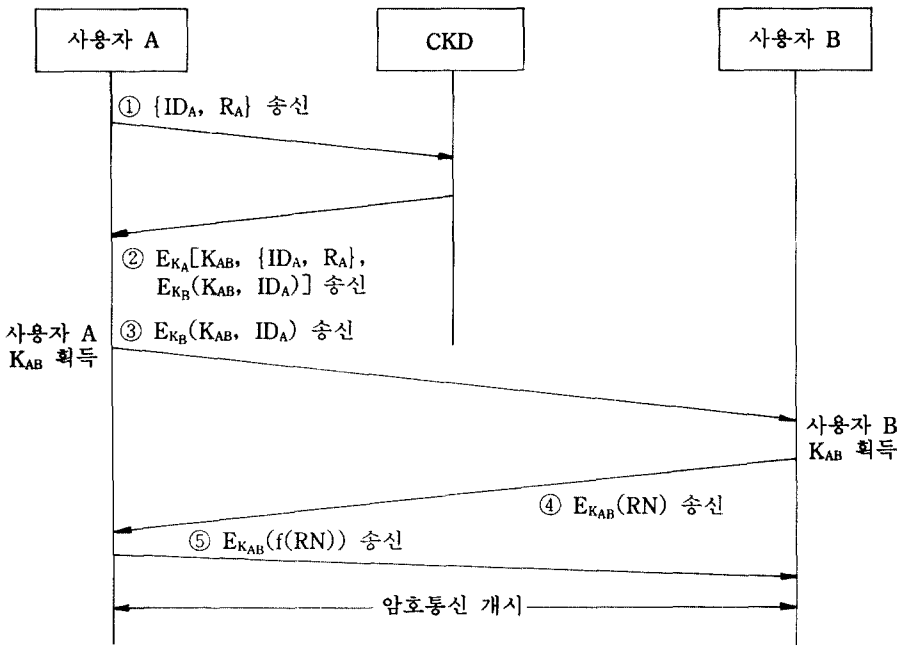
4. 생성된 **K**를 공개키로서 공개하고 **W**을 비밀키로 보관하다.

이밖에 공개키 암호시스템도 독특한 각각의 암호키 생성 알고리즘을 이용하여 암호키를 생성한다.

2.3. 암호키 분배[4,5,9]

암호키 분배방식에는 집중형 암호키 분배(CKD : centralized key distribution) 방식과 분산형 암호키 분배(distributed key distribution) 방식이 있다. 집중형 암호키 분배 방식은 하나의 암호키 분배 센타(key distribution center)에서 여러 컴퓨터에 필요한 암호키를 분배하는 방식이며, 분산형 암호키 분배 방식은 각각의 컴퓨터 호스트가 키 분배 센타를 갖추고 암호키 분배를 독립적으로 수행하는 암호키 분배 방식이다. 집중형 분배 방식은 그림 4와 같은 과정을 통해 암호키가 분배된다. 여기서 사용자 A에 의해 통신이 개시되고 CKD와 각각의 사용자간의 통신을 위한 2차 암호키는 이미 설정되어 있는 것으로 가정한다.

그림 4에서 알 수 있듯이 통신을 개시할 의도를 갖는 사용자 A는 자기 자신의 주소 ID_A(identifi-



여기서, K_{AB} : 사용자 A와 사용자 B간의 데이타 보호를 위한 데이타 암호키
ID_A : A의 인식 주소(identification address)
R_A : 사용자 A의 요구정보
K_A : 사용자 A와 CKD간의 2차 통신용 암호키
K_B : 사용자 B와 CKD간의 2차 통신용 암호키

그림 4. 대칭형 암호시스템을 위한 암호키 문제

cation name of user A)와 B와 통신을 원한다는 내용의 R_A을 투명하게 CKD에 전송한다.

CKD는 A와 B간의 통신용 세션키(K_{AB}), [ID_A, R_A], 그리고 사용자 B의 2차 통신용 암호키를 암호화되어 있는 $E_{K_B}(K_{AB}, ID_A)$을 사용자 A의 2차 암호화 키(K_A)로 암호화하여 사용자 A에게 송신한다. 사용자 A는 K_A로 K_{AB}와 [ID_A, R_A]을 복구하여 수신 [ID_A, R_A]가 송신 [ID_A, R_A]와 일치하는가를 검사한 후, 같을 경우 $E_{K_B}[K_{AB}, ID_A]$를 B로 송신한다. $E_{K_B}[K_{AB}, ID_A]$를 수신한 사용자 B는 K_{AB}와 ID_A를 구하여 midnight attack[1] 등과 같은 공격을 피하기 위하여 인증과정을 개시한다.

사용자 B는 random number(RN)를 생성한 후 이를 K_{AB}로 암호화한 $E_{K_{AB}}(RN)$을 생성하여 사용자 A에 전송한다. 사용자 A는 미리 정해진 함수(f)에 의해 f(RN)을 수행한 후 K_{AB}로 비화하여 사용자 B로 전송한다. 사용자 B는 f(RN)과 자기가 계산한 f(RN)과 동일한가를 검사하여 같으면 사용자 A를 인증하고 K_{AB}를 이용하여 암호통신을 개시한다. 한편, 분산형 암호키 분배방식은 각각의 호스트 컴퓨터들에게 암호키 분배권한을 배분하는 방식으로 n개의 통신노드가 있을 경우 각각의 통신노드는 $n(n-1)/2$개의 2차 통신용 암호키를 저장하고 있다가 때때로 타국의 암호키 요구 변경에 의해 이를 바꾸어야 하므로 많은 양의 메모리와 복잡한 암호키 관리 프로그램이 요구된다. 이와 같은 많은 암호가 보관을 방지하기 위하여 계층적으로 호스트를 나누는 방법 또한 이용된다.

비대칭형 암호시스템은 암호키가 공개키와 비밀키로 분리되어 있으므로, 비대칭형 암호시스템의

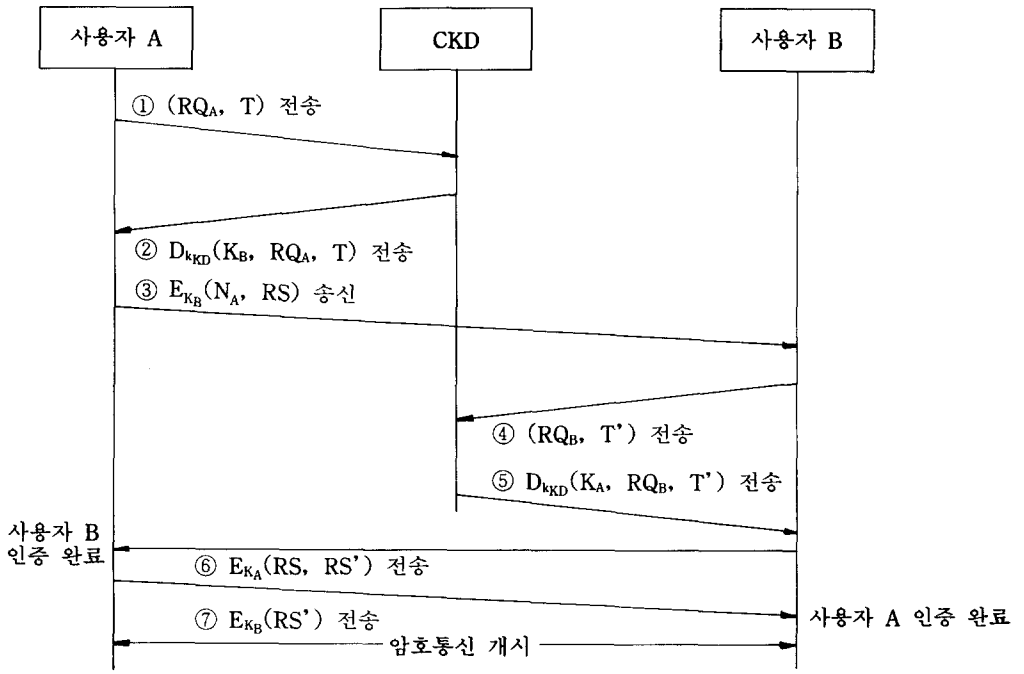


그림 5. 비대칭 암호시스템을 위한 암호키 분배

암호키 분배의 경우 공개키를 공개된 채널을 통해 상대에게 전송되므로, 암호키 분배는 대칭형 암호시스템의 경우와는 달리 간단하지만 분배된 암호키를 인증해야 한다는 새로운 과정을 통해 완수된다. 컴퓨터 통신망에서의 암호키 저장 장소인 암호키 디렉토리(KD : key directory)는 암호키 인증을 위하여 공개키를 보관, 갱신, 분배, 유지보수하는 기능을 수행한다.

사용자 A의 암호키 쌍을 (K_A, k_A), 사용자 B의 암호키 쌍을 (K_B, k_B), 그리고 KD의 암호키 쌍을 (K_{KD}, k_{KD})라고 하고 모든 사용자가 KD의 공개키 K_{KD}를 알고 있고 모든 사용자의 공개키들이 KD에 저장되어 있다면 암호키 분배는 사용자가 암호키를 요구할 경우 개시된다.

사용자 A와 사용자 B간의 암호키 분배는 다음과 같은 절차로 수행된다. 사용자 A는 통신하고 싶다는 정보의 RQ_A와 현재의 시간변수 T를 KD로 전송한다. 이를 수신한 KD는 자신의 비밀키 k_{KD}로 암호화한 사용자 B의 공개키(K_B), RQ_A및 T를 {D$_{kKD}$(K_B, RQ_A, T)} 형태로 사용자 A에 전송한다. 사용자 A는 KD의 공개키를 알고 있으므로 K_B, RQ_A및 T를 구할 수 있다. K_B를 안 사용자 A는 사용자 A의 이름(N_A)와 random number RS를 K_B로 암호화한 $E_{K_B}(N_A, RS)$를 사용자 B로 전송한다. 사용자 A로부터 수신된 암호키를 인증하기 위하여 사용자 B는 KD로 사용자 A와 통신하고 싶다는 정보의 RQ_B와 T'을 투명한 형태로 송신한다. KD는 자신의 비밀키(k_{KD})로 {K_A, RQ_B, T'}를 암호화하여 사용자 B로 전송한다. 사용자 B는 K_{KD}를 이용하여 {K_A, RQ_B, T'}를 구한후, 송수신 {RQ_B, T'}가 같으면 사용자 A의 인증된 공개키(K_A)의 획득을 완료한다.

사용자 B는 임의로 선택된 RS'을 선택하여 K_A 암호화한 E_{K_A}(RS, RS')을 사용자 A로 전송한다.

사용자 A는 송신 RS와 수신 RS를 비교하여 같으면 사용자 B가 인증된 것으로 간주한 후, 사용자 B의 공개키(K_B)로 RS'을 암호화하여 E_{K_B}(RS')을 사용자 A로 전송한다.

사용자 B는 RS'를 복구하여 송신 RS'와 수신 RS'를 비교하여 같으면 사용자 A를 인증한 것으로 간주한다. 위와 같은 방식에서는 인증된 암호키를 분배하고 통신 주체간의 통신을 개시해 주는 암호키 분배센타가 반드시 통신망에 존재해야 하며, 암호키 분배센타는 이를 위한 하드웨어와 소프트웨어로 구성되어 있어야 한다.

2.4 암호키 관리[1, 7, 10]

암호키 관리는 관리 프로그램에 의해 수행되며, 이는 통신망에서 요구하는 암호키를 생성하여 저장하고 있다가 망에서 요구하는 암호키로 변환하며, 마스터키 변경시 저장되어 있는 키암호화키를 새로운 마스터키로 암호화한 키암호화키로 변경하고 또 다른 암호키를 입력, 변경, 그리고 삭제하는 기능을 주로 수행한다.

컴퓨터 통신망에서의 암호키 관리는 암호 알고리즘의 종류, 암호 알고리즘의 응용분야, 통신망의 구조에 따라 달라진다. 일반적으로 통신망은 하나의 호스트와 여러개의 터미날들로 구성되어 통신 및 파일보안을 수행하는 단일통신망(single domain communication network)과 2개 이상의 호스트와 터미날들로 구성되어 있는 다중 통신망(multiple domain communication network)으로 구분된다.

본 절에서는 대칭형 암호시스템중의 대표적인 알고리즘인 DES를 이용하는 다중통신망(multiple domain communication network)에서의 암호키 관리에 대해 주로 기술한다.

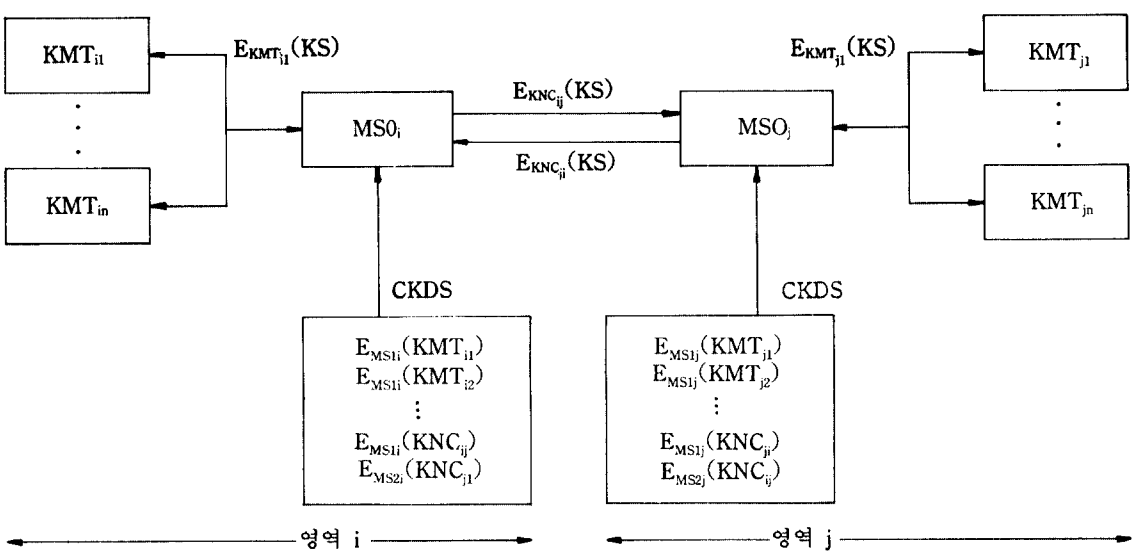


그림 6. 통신보안을 위한 다중통신망에서의 암호키 관리

다중 통신망에서의 통신보안을 위한 암호키는 데이타를 보호하는 데이타 암호키, KS와 KF를 보호하기 위한 2차 키암호화키[KNC$_{ij}$: 영역 i에서 영역 j로 전송되는 데이타 암호키 보호용, KNC$_{ji}$: 영역 j에서 영역 i로 전송되는 KS보호용], 터미날로 전송되는 KS를 보호하기 위한 영역 i에서의 터미날 마스터키(KMT$_{ik}$, k=1, 2…, n)와 영역 j에서의 터미날 마스터키(KMT$_{jk}$, k=1, …, n), KMT$_{ik}$, KMT$_{jk}$, KNC$_{ij}$, KNC$_{ji}$등을 암호화하여 보관하기 위한 영역 i에서의 MS0$_i$/MS1$_i$/MS2$_i$, 그리고 영역 j에서의 MS0$_j$/MS1$_j$/MS2$_j$등이 있다. 터미날 마스터키는 MS1으로 암호화되어,

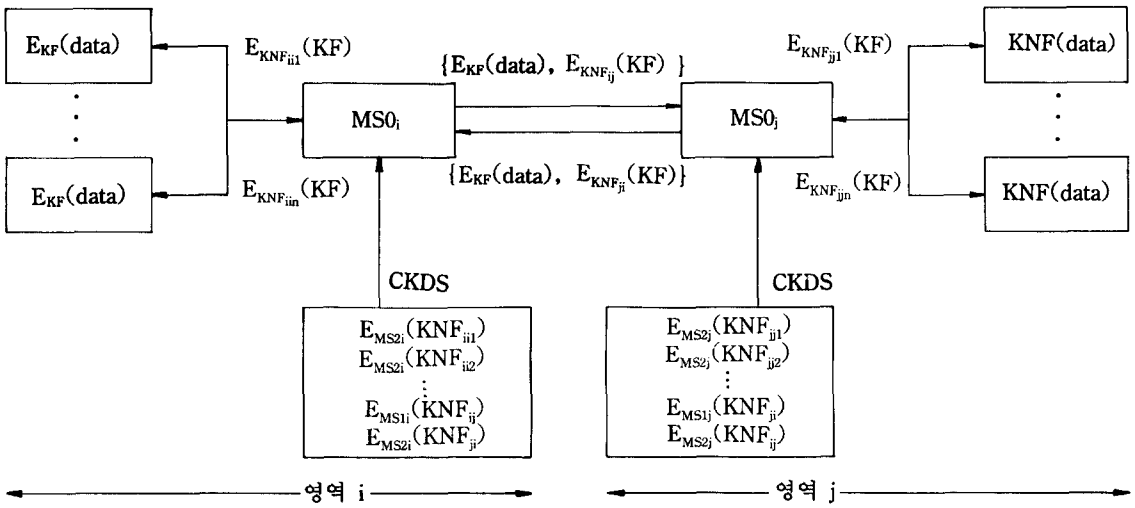


그림 7. 파일 보안을 위한 다중 통신망에서의 암호키 관리

자기 영역에서 상대방 영역으로의 2차 통신용 키 암호화키는 MS1으로 암호화되어, 상대방 영역에서 자기 영역으로의 2차 통신용 키암호화키는 MS2로 암호화되어 CKDS(cryptographic key data set)에 저장된다. 이와 같은 형태로 2차 및 1차 암호키를 암호화하여 저장하는 이유는 통신보안 및 파일보안간의 독립성 및 고립성 보장과 암호키의 유일 방향특성(uni-directional property)을 보장하기 위함이다[1].

그림 6과같은 다중 통신망에서의 1차 데이타 암호화키(KS) 생성 및 분배를 위한 과정은 다음과 같다. 통신을 개시하려고 하는 호스트가 영역 i일 경우 영역 i의 호스트는 random nurmber generator를 이용하여 RN을 생성하고 KS가 투명한 값으로 알려지는 것을 방지하기 위하여 RN을 E_{MS0i}(KS)로 간주한다. E_{MS0i}(KS)는 ECPH와 DCPH 기본 암호화 명령의 입력변수로 이용되며, 이는 데이타를 암호화 또는 복호화할 때 이용된다. domain i의 암호키 관리 프로그램은 [RFMK : {E_{KS1_i}(KNC_{ij}), E_{MS0_i}(KS) } → $E_{KNC_{ij}}$(KS)]를 이용하여 $E_{KNC_{ij}}$(KS)를 생성하여 영역 j로 전송하고, [RFMK : {E_{MS1_i}(KMT_{ik}), E_{MS0_i}(KS) }→$E_{KMT_{ik}}$(KS) for k=1, 2, … , n]을 이용하여 $E_{KMT_{ik}}$(KS)을 생성하여 자신의 k번째 터미날로 전송한다. $E_{KNC_{ij}}$(KS)를 수신한 영역 j의 호스트는[RTMK : {E_{MS2_j}(KNC_{ij}), $E_{KNC_{ij}}$(KS) }→E_{MS0_j}(KS)]를 이용하여 E_{KM0_j}(KS)를 구한다. E_{KM0_j}(KS)는 데이타를 암호화 및 복호화하는데 이용된다. E_{MS0_j}(KS)를 얻은 호스트 j는 [RFMK : {E_{MS1_j}(KMT_{jk}), E_{MS0_j}(KS)} → $E_{KMT_{jk}}$(KS)]를 이용하여 $E_{KMT_{jk}}$(KS)를 생성하여 자신의 k번째 터미날로 전송한다. 영역 i의 k번째 터미날은 [DECK : {$E_{KMT_{ik}}$(KS)]을 이용하여 KS를 동작 키 저장 레지스터(working key storage)에 저장하며, 영역 j의 k번째 터미날도 마찬가지 동작을 함으로서 영역 i의 터미날 k, 호스트 i, 호스트 j 및 영역 j의 터미날 k는 KS를 공유한다.

따라서, 통신보안을 위해서 영역 i에서는 KMT_{ik}및 KNC_{ij}는 MS1i로, KNC_{ji}는 $MS2_i$로 암호화되어 저장하며, 영역 j에서는 KMT_{jk}와 KNC_{ji}를 $MS1_j$로 KNC_{ij}는 $MS2_j$로 암호화하여 CKDS에 저장한다.

다중 통신망에서의 파일보안을 위한 암호키는 그림 7에서 보는바와 같이 실제 파일을 저장하기 위한 1차 파일키(KF), 영역 i에서 영역 j로 전송되는 KF를 보호하기 위한 2차 파일용 암호키(KNF_{ij}), 영역 j에서 영역 i로 전송되는 KS를 보호하기 위한 2차 파일용 암호키 KNF_{ji}, 호스트 i에서 생성되어 사용되는 1차 파일용 암호키(KF)를 보호하기 위한 KNF_{ii}, 호스트 j에서 생성되어 사용되는 1차 파일용

암호키(KF)를 보호하기 위한 KNF_{jj}등이 있다. KNF_{ij}는 영역 i에서는 $E_{MS1_i}(KNF_{ij})$ 형태로 영역 j에서는 $E_{MS2_i}(KNF_{ij})$ 형태로 각각의 CKDS에 저장되며, KNF_{ji}는 영역 i에서 $E_{MS2_i}(KNF_{ji})$ 형태로 영역 j에서 $E_{MS2_i}(KNF_{ji})$ 형태로 CKDS에 저장되어 있다.

한편, KNF_{iik}(k=1, …, n)은 $E_{MS2_i}(KNF_{iik})$ 로, KNF_{ijk}는 $E_{MS2_j}(KNF_{ijk})$의 형태로 저장되어 있다. 호스트는 파일 보안을 위해 각 사용자마다 2차 파일키(KNF_{iik})를 할당하며, 각각의 파일의 파일헤더의 특정영역에는 $E_{KNF_{iik}}(KF)$를 저장하며, 데이타는 $E_{KF}(Data)$ 형태로 저장된다. 새로운 파일을 생성하여 영역 i에서 영역 j로 전송하려면 호스트 i는 random number generator를 이용하여 RN을 생성하고 KF가 투명한 값으로 노출되는 것을 방지하기 위하여 RN을 $E_{MS0_i}(KF)$로 간주한다.

호스트 i는 $E_{MS0_i}(KF)$과 [ECPH : {RN, Data} $\rightarrow E_{KF}(Data)$]를 이용하여 파일의 데이타를 암호화한다. 호스트 i는 [RFMK : {$E_{MS1_i}(KNF_{ij})$, $E_{MS0_i}(KF)$} $\rightarrow E_{KNF_{ij}}(KF)$]을 이용하여 $E_{KNF_{ij}}(KF)$를 생성하여 $E_{KNF_{ij}}(KF)$를 파일헤더에 입력하여, $E_{KF}(Data)$와 함께 영역 j로 전송한다. 영역 j의 호스트는 파일 헤더에서 $E_{KNF_{ij}}(KF)$을 꺼내어 [RTMK : {$E_{MS2_j}(KNFij)$, $E_{KNF_{ij}}(KF)$} $\rightarrow E_{MS0_j}(KF)$]을 이용하여 $E_{MS0_j}(KF)$를 생성하고 [DCPH : {$E_{MS0_j}(KF)$, $E_{KF}(Data) \rightarrow Data$]를 이용하여 투명한 파일의 데이타를 복구한다.

위와같은 과정 및 암호키 관리에 의해 영역 i에서 생성, 암호화된 파일을 영역 j에서 복호화하여 투명한 파일을 얻을 수 있다. 영역 i에서 이미 만들어진 파일은 $E_{KF}(Data)$와 파일헤더에 $E_{KNF_{ii}}(KF)$로 구성되어 있으므로 [{RTMK : {$E_{MS2_i}(KNFii)$, $E_{KNF_{ii}}(KF)$} $\rightarrow E_{MS0_i}(KF)$}]를 이용하여 $E_{MS0_i}(KF)$을 구하고, 다시 [RFMK : {$E_{MS1_i}(KNFji)$, $E_{MS0_i}(KF)$} $\rightarrow E_{KNF_{ij}}$]을 수행하여 $E_{KNF_{ij}}(KF)$를 구한 후, $E_{KNF_{ij}}(KF)$를 파일헤더의 $E_{KNF_{ii}}(KF)$와 치환하여 영역 j로 전송하며, 나머지 과정은 새로운 파일의 경우와 같으며, 이와 같은 과정을 통해 영역 i에서 만들어져 암호화되어 보관되어 있는 파일을 영역 j에서 복호할 수 있다. 결론적으로 호스트의 암호시스템은 하드웨어와 소프트웨어로 구성되어 암호 알고리즘을 수행하는 암호장치(crytographic facility), 키 암호화키를 생성하는 프로그램인 암호키 생성기(key generator), 그리고 1차 암호화키를 생성하고 CKDS에 접근하여 키변환 기능을 수행하는 키관리 프로그램으로 실현된다.

컴퓨터에서의 CKDS는 키 관리 프로그램이 암호키 관리를 위해 요구되는 모든 종류의 2차 암호

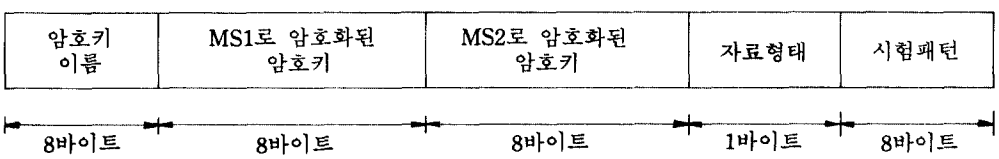


그림 8. CKDS 레코드 구조

키를 기억하는 기억장치로서, 2차 암호키는 MS1 또는 MS2로 암호화되어 저장되어 있고 CKDS의 생성 및 유지보수는 암호키 생성 프로그램에 의해 수행된다. CKDS 레코드 입력 명령은 entry의 부가(add), 갱신(update), 삭제(delete) 등이 있으며, CKDS record 구조는 일반적으로 그림 8과 같다.

그림 8의 CKDS 레코드는 CKDS내의 entry 위치 확인용으로 이용되는 암호키 이름부, MS1으로 암호화되어 저장되는 암호키 보관영역, MS2로 암호화되어 저장되는 암호키 보관영역, 자료형태부, 그리고 시험패턴부로 구성된다. 자료형태부는 암호키가 MS1으로 암호화되어 저장되어 있을 경우 형태 1으로 부호화되고, 암호키가 MS2로 암호화되어 저장되어 있을 경우 형태 2로 부호화되며, MS1과 MS2로 동시에 암호화되어 저장되어 있는 경우 형태 3으로 부호화된다. 즉, 앞에서 기술했듯이

KMT는 E_{MS1}(KMT) 형태로 저장되어 있으므로 형태 1이고, 2차 파일키 KNF는 E_{MS2}(KNF) 형태로 암호화 되어 저장되어 있으므로 형태 2이며, 통신 보안용 2차 암호키인 KNC_{ij}와 KNC_{ji}는 MS1과 MS2로 공히 암호화되어 저장되므로 형태 3이다. 시험 패턴부는 저장되어 있는 암호키를 인증하는데 이용된다. 따라서 컴퓨터의 호스트 시스템에 저장되어 있는 모든 암호키는 그림 8과 같은 구조의 CKDS에 저장된다.

컴퓨터 시스템에서 호스트 마스터 키를 변경해야 할 필요가 있을 경우, MS1과 MS2로 암호화되어 CKDS에 저장되어 있는 모든 키 암호화키를 새로운 호스트 마스터 키로 암호화하여 저장해야 한다. 만약 변경전 호스트 마스트 키를 MS0*, MS1*, MS2*라 하고, 새로운 호스트 마스터 키를 MS0, MS1, MS2라 하며, MS0가 호스트 시스템에서 SMK 기본 암호동작을 이용하여 CF에 저장되어 있고 x가 "0" 또는 "1"일 경우, E_{MSx}*(Key)를 E_{MSx}(Key)로 변경하는 절차는 다음과 같다.

1. [EMK : {MS1} → E_{MS0}(MS1)]을 이용하여 E_{MS0}(MS1)를 생성한다.

2. [ECPH : {E_{MS0}(MS1), MS1} → E_{MS1}(MS1), ECPH : {E_{MS0}(MS1), MS2} →E_{MS1}(MS2)]를 이용하여 E_{MS1}(MS1), E_{MS1}(MS2)를 생성한다.

3. [EMK : {MS2} → E_{MS0}(MS2)]을 이용하여 E_{MS0}(MS2)를 생성한다.

4. [ECPH : {E_{MS0}(MS2), MS1*} → E_{MS2}(MS1*), ECPH : {E_{MS0}(MS2), MS2*} → E_{MS2}(MS2*)]를 이용하여 E_{MS2}(MS1*)와 E_{MS2}(MS2*)를 생성한다.

5. 암호키 관리 프로그램은 CKDS에서 E_{MS1}*(α)와 E_{MS2}*(β)를 읽어서 [RTMK : {E_{MS2}(MS1*), E_{MS1}*(α)} → E_{MS0}(α), RFMK : {E_{MS1}(MS1), E_{MS0}(α)} → E_{MS1}(α)]를 이용하여 E_{MS1}(α)를 구한후 CKDS에 보관하고, [RTMK : {E_{MS2}(MS2*), E_{MS2}*(β)} → E_{MS0}(β), RFMK : {E_{MS1}(MS2), E_{MS0}(β)} →E_{MS2}(β)]를 이용하여 E_{MS2}(β)를 구한 후 CKDS에 보관한다.

따라서, 상기의 5단계를 통하여 암호키 관리 프로그램은 임의의 E_{MSx}*(Key)를 E_{MSx}(Key)로 변경할 수 있다.

암호키 관리 프로그램은 현재 CF내에 저장되어 있는 MS0가 자신이 의도했던 MS0'과 같음을 인증할 수 있어야 하며, 이는 다음과 같은 절차로 수행된다.

1. [EMK : {MS0'} → E_{MS0}(MS0')]을 이용하여 E_{MS0}(MS0')를 생성한다.

2. [DCPH : {E_{MS0}(MS0'), E_{MS0}(MS0')} → $D_{MS0'}$(E_{MS0}(MS0'))]를 수행한후, $D_{MS0'}$ (E_{MS0}(MS0'))를 v로 한다.

3. 만약 v=MS0'이면 CF내에 MS0는 자신이 의도했던 MS0'과 같다고 간주하고, 만약 v≠MS0'이면 CF내에 MS0는 자신이 의도했던 MS0'과 같다고 간주한다.

상기와 같은 방법으로 암호키 관리 프로그램은 호스트 마스터 키를 인증한다.

3. 결 론

컴퓨터 통신망에서 이용되는 암호키는 암호시스템의 보안성에 커다란 영향을 미치므로 암호키의 효율적이고 안전한 생성, 응용분야에 적합한 해당 암호키의 분배, 그리고 암호키의 생성, 분배, 변경 등을 총괄적으로 관리하는 암호키 관리는 컴퓨터 통신망에서 매우 중요한 역할을 수행한다.

본 고에서는 비대칭형 암호시스템 및 대칭형 암호시스템의 효율적인 암호키의 생성방식을 분석하고, 생성된 암호키의 효율적이고 안전한 암호키 분배기법을 기술하였으며, DES를 기본으로 한 다중 통신망에서의 통신보안과 파일 보안을 위한 여러 종류의 통신용 암호키 및 파일용 암호키 등의 1차 암호화키와 1차 암호화 키를 보호하기 위한 KNC, KNF 등의 2차 키 암호화키를 정의하여 이들을 이용하여 실현될 수 있는 다중 통신망에서의 보안통신을 위한 절차를 분명히 기술하였고, 키 암호화키를 암호화하여 저장하는 CKDS의 구조 및 입출력 명령문의 종류를 정의하였을 뿐만 아니라, 호스트 마스터 변경시 변경되어야 할 키 암호화키를 변경하는 절차를 분석하였고, 또 CF내에 저장되어 있는 MS0를 인증하는 기법을 소개하였다.

참 고 문 헌

1. C.H. Meyer, and S.M. Matyas. Cryptography : A New Dimension in Computer Data Security, John Wiley and Son, New York, 1982.

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

3. "Data Encryption Standard," National Bureau of Standard, Federal Information Processing Standards Publications, 46, 1977.

4. R.L. Rivest, A. Shamir, and L. Adleman, A New Method of Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM. pp.120-126, 1978.

5. R.C. Merkle, and M.E. Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks," IEEE Trans. Inf. Theory, IT-2, 5, pp.525-535, 1978.

6. J.K. Everton, "A Hierachical Basis for Encryption Key Management in Computer Communications Network," Trends and Applications 1978 : Distributed Processing, IEEE Computer Society, Long Beach CA. 1978.

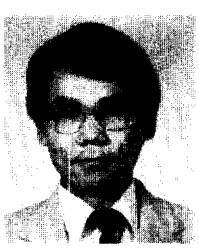
7. W.F. Ehrsam, S.M. Matyas, C.H. Meyer, and W.L. Tuchman, "A Cryptographic Key Management Scheme for Implementing the Data Encryption Standards," IBM System Journal, 17, No. 2, pp.106-125, 1978.

8. S.M. Matyas and C.H. Meyer, "Generation, Distribution, and Installation of Cryptographic Keys," IBM System Journal, 17, No. 2, pp.126-137, 1978.

9. J. Sebbery, J. Pieprzyk, Cryptography : An Introduction to Computer Security, Prentice Hall, New York, 1989.

10. G.L. Popek, and C.S. Kline, "Encryption Protocols, Public Key Algorithms, and Digital Sginatures in Computer Network," Academic Press, New York, pp.135-153, 1979.

□ 著者紹介



廉 興 烈(正會員)

1981년 漢陽大學校 電子工學科(學士)
1983년 漢陽大學校 大學院 電子工學科(碩士)
1990년 漢陽大學校 大學院 電子工學科(博士)
韓國電子通信研究所 先任研究員
現在 順天鄕大學校 工科大學 電子工學科 助教授