의사 랜덤치환을 이용한 다중레벨 키분배

김 주 석[†] · 신 원[††] · 이 경 현[†]

요 약

대칭키 관리 시스템에서 계층적 구조를 지닌 다중 사용자 그룹에 대한 새로운 키관리 방안을 제안한다. 제안된 방식은 의사 랜덤치환에 의해 생성되는 트랩도어 일방향 치환을 이용하며, 구현시 시간과 공간적 측면에서 유리하기 때문에, 완전순서 집합과 부분순서가 있는 집합으로 구성되는 다단계 계층적 구조에 사용 가능하다. 또한, 다른 제안 방식과 비교하여 성능을 분석하고, 제안 방식이 키생성 시간과 키저장 크기에서 보다 효율적인 것을 보인다.

A Multilevel Key Distribution using Pseudo-random Permutations

Ju-Seog Kim[†] · Weon Shin[††] · Kyung-Hyune Rhee[†]

ABSTRACT

We propose a new key management scheme for multiuser group which is classified as hierarchical structure (sometimes it is called a multilevel security hierarchy) in the symmetric key cryptosystem. The proposed scheme is based on the trapdoor one-way permutations which are generated by the pseudo-random permutation algorithm, and it is available for multilevel hierarchical structure composed of a totally ordered set and a partially ordered set, since it has advantage for time and storage from an implemental point of view. Moreover, we obtain a performance analysis by comparing with the other scheme, and show that the proposed scheme is very efficient for computing time of key generation and memory size of key storage.

1. 서 론

현대 사회는 사회 활동 범위가 점점 복잡해지고 산업이 고도로 발전됨에 따라 대량의 정보를 처리하여 신속 정확하게 전달할 수 있는 정보화 시대를 맞이하고 있다. 한편 미래의 정보 산업화 사회에 있어서 큰 특징 중의 하나는 통신 네트워크의 급속한 확대 보급과 이를 통한 정보의 전달이다. 이러한 통신 네트워크에 있어서 정보통신 기술의 정상적이고 올바른 이용과 관리를 위하여 컴퓨터 통신의 안전성 보장을 위한 다양한 메커니즘과 서비스들이 필요하게 되었다. 이러한 메커니즘 중 가장 중요한 고려사항으로는 네트워크 상에서 전송되는 데이터를 암호화하고 복호화하는 암호화 기법과 이 암호화 기법에 사용된 키의 관리 문제[4, 5, 11, 14]이다. 대규모의 네트워크에서 누구나 자유스럽게 종점간 암호에 의해 비밀 통신을 수행하기 위해서는 매우 많은 사용자들 모두와 비밀키를 공유해야 하는 어려움이 있다.

실제 환경에서 키관리는 암호의 가장 어려운 부분이며 그 키의 비밀을 유지하는 것은 더욱 어렵다. 또한 키관리는 안전한 암호 체계를 구현하는데 고려해

※본 연구는 1996년도 부경대학교 기성회 연구비 지원에 의한 결과임.

† 준 회 원:부경대학교 전산정보학과 석사과정
†† 준 회 원:부경대학교 전자계산학과 석사과정
††† 정 회 원:부경대학교 전자계산학과 조교수

논문접수:1997년 5월 28일, 심사완료:1997년 8월 22일

야 할 가장 중요한 부분이다.

키계층은 각 호스트 노드에서 키층으로 구성되어 있다. 즉, 하나의 마스터 키와 두 번째 레벨에서의 키 암호화키, 세 번째 레벨에서의 데이터 암호화키들이다.

키관리의 주목적은 대칭 혹은 비대칭적인 암호 메커니즘에 사용되도록 암호키 자료를 처리하기 위해서 안전한 절차를 규정하는 것으로서 사용자 등록, 키발생, 키저장, 키삭제를 포함한다. 이들 키의 안전관리는 시스템에 암호함수 구성시 가장 중요한 요소의 하나이기 때문에 아주 정교한 보안 개념조차도 키관리가 허약하면 쓸모가 없게 된다. 따라서 본 논문에서는 정부, 외교 그리고 군, 그 외에도 사업이나 사용 부분의 다른 분야 등에서 흔히 발생하게 되는 계층적 구조가 존재하는 다단계 보안 시스템에서의 효율적인 키관리 방안을 제시하고자 한다[2]. 예를 들면 고도의 신중을 요하는 정보를 포함하는 데이터베이스의 관리나 산업 비밀의 보호 등에 적용가능하다.

본 논문은 암호 표기법에 기초한 다단계 보안문제 해결 제시를 위해 랜덤치환을 이용한 다중 레벨 키관리 방안을 제시한다.

2장에서는 일반적인 키관리 방안을 살펴보고, 키계층, 키생성, 키분배, 키공유, 키유지 관리에 대해 간략하게 소개한다. 3장에서는 기존에 알려진 다중 레벨 키관리 방안으로서 완전순서 집합(Totally Ordered Sets)과 부분순서가 있는 집합(Arbitrary Posets) 그리고 지수법에 근거한 대칭 암호 알고리즘을 살펴보고, 4장에서는 랜덤치환(Random Permutation)을 이용한 키관리 방안인 다중 레벨 키관리 방안을 제시한다. 마지막으로, 5장에서는 이들 다중 레벨 키관리의 방식별 성능을 비교 평가 분석하며 응용 사례에 대해서 간단히 언급한다.

2. 키관리 방안

통신망의 보안을 유지하기 위하여 흔히 사용되는 키계층은 각 호스트 노드에서 키층으로 구성되며 하나의 마스터키와 두 번째 레벨에서의 키 암호화키, 세번째 레벨에서의 데이터 암호화키 등으로 나누어진다[4, 5, 11].

마스터키는 키계층의 최상위층의 키로서 시스템의 모든 세션키와 터미널키를 저장 관리하는데 사용되며, 터미널키는 중간키로서 세션키를 암호화하여 터미널로 전송할 때 쓰이는 키암호화키이다. 세션키는 최하위의 키로서 통신하려는 시간 즉, 세션을 통한 데이터와 인증메세지를 암호화하는데 사용되며 다른 키와는 달리 키의 생명 주기가 짧아 빈번히 생성되는 특징을 갖고 있으며, 또한 키들은 링크의 한쪽 끝에서 의사랜덤 알고리즘에 의해 생성되어 링크의 상대방으로 안전하게 전송된다.

키의 생성에서 인위적으로 생성하는 것은 적절하지 못하고 랜덤하게 생성하는 것이 이상적이다. 키를 랜덤하게 생성하는 방법은 랜덤 생성기(Random Bit Generator)나 의사랜덤생성기(Pseudo Random Generator)를 이용하는 방법이 있다[4, 5, 14].

키분배 방식에서 대칭키 분배 시스템은 송신자는 암호화키를 이용한 암호알고리즘으로 평문으로부터 암호문를 생성시켜서 전송하며, 수신자는 복호알고리즘으로 복호화키를 이용하여 암호문에서 복호화한 후 평문을 얻는다. 대칭키 암호화시스템은 암호화 및 복호화가 동일한 키를 사용하므로 키가 사전에 두 통신자 사이에 설정되어야 하며 보안이 유지되어야 한다.

비대칭키 분배 시스템은 송수신자는 비대칭키 암호알고리즘을 이용한 키분배 프로토콜을 수행하여 실제 암호 통신에 사용할 수 있는 세션키를 갖는다. 각 가입자는 공개키 P와 비밀키 S만을 갖게 되는데 가입자 A는 (P_A, S_A)를 갖는다. P_A는 가입자 모두에게 공개되고 P_A, S_A는 특정의 관계식에 의해 생성된다[4, 8, 14].

키의 생성과 분배를 제외한 키 관리요소는 키활성, 키저장, 키교체, 키삭제을 위한 절차가 있고 이중 키교체에 대한 사항은 키의 사용기한(life cycle)에 의존하며 구현상 키분배와 결부시켜 고려하여야 한다.

3. 다중 레벨 키관리 방안

본 장에서는 네트워크상에서 다수의 키 관리 대상 사용자들을 보안등급(security class)에 따라 사용자 집합으로 분리하고 각 사용자 집합간에 계층적인 구조가 주어지는 경우로 모델링하여 분석하고자 한다[2]. 서로 배반인 사용자 집합들을 $U_1, U_2, \cdots, U_n$이라 하자 각 첨자 $i(1 \le i \le n)$은 각 사용자 집합에 부여된 보안등급이라 두면, 클래스 집합 $S=\{U_1, U_2, \cdots, U_n\}$에

이항관계인 부분순서(예를 들어 $\leq$)를 다음과 같이 부여하여 부분순서가 있는 집합 $(S, \leq)$를 정의할 수 있다. 즉, 만약 $U_i \leq U_j$이면 집합 U_i의 사용자들은 보안 등급에서 집합 U_j의 사용자들보다 낮거나 같다. 이것은 실례로써 집합 U_j의 사용자들은 집합 U_i의 사용자들의 정보에 대한 접근 및 컴퓨터 등의 사용 권한이 부여되지만, 그 반대 (U_i가 U_j에 대한 관계)의 경우는 성립하지 않는다는 의미이다. 또한 집합 S상의 부분순서 관계에 의해 집합 U_i의 사용자들은 $m \leq i$를 만족하는 모든 집합 U_m 사용자들의 정보에 대한 접근 및 컴퓨터 등의 사용권한이 부여된다. 예를 들어 다음 (그림 1), (그림 2)와 같은 완전순서와 부분순서가 있는 집합 $(S, \leq)$ $(n=9)$의 Hasse 다이어그램을 고려하면 S내의 각 사용자 클래스가 유지해야 할 키들은 다음과 같다.

편의상 각 U_i클래스가 지녀야할 키를 K_i라 하면

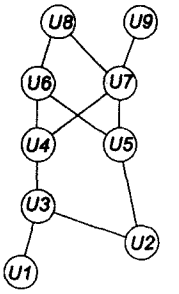
(1) 완전순서 집합의 경우

클래스	키
U_1	K_1
U_2	K_1, K_2
U_3	K_1, K_2, K_3
⋮	
U_9	$K_1, K_2, K_3, \cdots, K_9$

(그림 1) 완전순서 집합의 Hasse 다이어그램
(Fig. 1) Hasse diagrram of totally ordered sets

(2) 부분순서가 있는 집합의 경우

클래스	키
U_1	K_1
U_2	K_2
U_3	K_1, K_2, K_3
U_4	K_1, K_2, K_3, K_4
U_5	K_2, K_5
U_6	$K_1, K_2, K_3, K_4, K_5, K_6$
U_7	$K_1, K_2, K_3, K_4, K_5, K_7$
U_8	$K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8$
U_9	$K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_9$



(그림 2) 부분순서가 있는 집합의 Hasse 다이어그램
(Fig. 2) Hasse diagram of arbitrary posets

따라서 각 클래스의 사용자들은 많은 수의 키를 보관유지해야 하면 최악의 경우 모든 i에 대해 $U_i \leq U_j$인 클래스 U_j의 사용자는 n개의 키를 유지해야 한다. 특히, 이렇게 유지해야할 키의 개수는 클래스의 개수가 증가함에 따라 선형적으로 증가한다.

3.1 Totally Ordered Sets

부분순서가 있는 집합의 가장 간단한 경우는 그 집합이 완전순서일 때 발생한다. 즉, 사용자 집합이 $U_1 \leq U_2 \leq \cdots \leq U_n$일 때이다. 만약, 함수 f가 일방향이고 특히 f함수를 m번 합성한 함수 f^m가 모든 양의 정수에 대해 일방향이라 가정하자. 그리고 K_n이 랜덤하게 선택되면 다음과 같이 정의할 수 있다.

$$K_{n-1} = f(K_n)$$
$$K_{n-2} = f(K_{n-2}) = f(f(K_n)) = f^2(K_n)$$
$$\vdots$$
$$K_1 = f(K_2)$$

따라서, $g_{ij} = f^{j-i}$이다.

f의 양의 멱승들은 일방향이므로 i가 j보다 크지 않은 경우만 g_{ij}가 계산될 수 있다.

3.2 Arbitrary Posets

부분순서가 있는 집합에 대한 해법은 다음과 같은 형태를 고려한다. 정수 t_i를 각 클래스 U_i에 t_j가 t_i를 나누면 $U_i \le U_j$를 만족하도록 할당한다.

이와같은 할당의 예는 U_i 노드에 기록된 t_i값과 함께 (그림 3)에 나타나 있다. 또한, $f_{mk} = f_m * f_k$(여기서 m과 k는 정수이고 $*$는 합성을 나타낸다)로서 일방향 함수의 그룹 $\{f_m\}$을 정의한다.

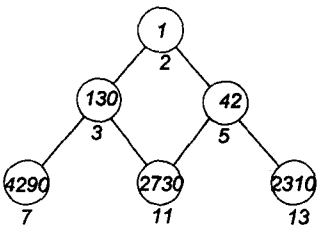
랜덤한 K_0을 선택하고 $K_i = f_{t_i}(K_0)$를 정의한다. 만약, $U_i \le U_j$이면 어떤 정수 k에 대해 $t_i = k \cdot t_j$이고 $K_i = f_{t_i}(K_0) = f_k * f_{t_j}(K_0) = f_k(K_j)$이다. 즉, K_i는 K_j와 $g_{ij} = f_{t_i/t_j}$로부터 계산된다.

이러한 계산은 $U_i \le U_j$인 경우에만 가능하다. 예를들어 키 센터는 두 개의 큰 소수 p와 q를 선택하고 $M = p \cdot q$를 계산한 후 이를 공개한다. 다음으로 $f_m(K) = K^m \pmod M$을 계산한다.

상위 클래스에서 하위 클래스의 키 생성은 다음과 같이 구할 수 있다.

$$K_i = K_0^{t_i} = [K_0^{t_j}]^{t_i/t_j} = [K_j]^{t_i/t_j} \pmod M$$

한편 t_i에 대한 할당은 다음과 같다. 각 계층 U_i는 서로 다른 소수 p_i와 $t_i = \prod_{U_j \le U_i} p_j$를 할당한다. 구체적인 p_i와 t_i의 값 할당은 (그림 3)에 한 예로써 주어졌다.



(그림 3) 클래스 U_i에 할당된 소수 p_i와 t_i
p_i : 노드아래 값, t_i : 노드안의 값
(Fig. 3) Assigned prime p_i, t_i to class U_i
p_i : value under node, t_i : value in node

3.3 지수법에 근거한 대칭암호 알고리즘

3.2에서 기술한 키 관리 시스템에 적절히 동작하는 암호알고리즘은 다음의 모듈러 지수 연산이다[2, 8].

키 센터는 소수 N을 선택하고 공개한다. 각 계층 U_i는 $N-1$에 서로 소인 비밀수인 b_i를 받는다. 암호화될 모든 대상은 N보다 더 작은 정수 x로 표현되는 각각의 블록으로 나누어진다. 암호화와 복호화 절차는 다음과 같이 수행된다.

$$x' = (x)^{a_i} \bmod N \text{ and } x = (x')^{b_i} \bmod N$$

여기서 a_i는 모든 i에 대해

$$a_i \cdot b_i = 1 \bmod (N-1)$$

이 되도록 선택된다.

암호알고리즘은 주어진 N에 대해 대칭적이며, a_i와 b_i는 서로 쉽게 계산된다. 이러한 방식은 만약 K_i가 $N-1$와 서로 소라면 $b_i = K_i$로써 3.2의 키 관리시스템에 적용되어질 수 있다.

4. 랜덤치환을 이용한 다중레벨 키관리방안

4.1 랜덤치환 발생 알고리즘

이진 난수발생기로는 주로 원시다항식에 근거한 LFSR(Linear Feedback Shift Register)을 이용하여 왔고, 일반적인 이진 난수발생기로 선형합동법(Linear Congruential Method)을 많이 이용하여 왔다[6, 7]. 그러나, LFSR과 선형 합동법을 이용한 방법들은 난수의 통계적 특성은 우수하지만 선형적인 성질 때문에 암호 수열로는 사용이 불가능하다. 이에 Akl과 Meijer[1, 12]는 선형 합동법이 제공하는 주기성과 난수의 통계적 특성을 보존하는 랜덤치환 발생기를 제안하였다. 크기 m의 임의의 치환과 0과 $m!-1$ 사이의 정수가 일대일로 대응된다는 성질을 이용하여, 선형 합동법에 의해 발생되는 난수에 대응되는 치환을 결합하여 랜덤치환을 발생하는 것을 제안하였다. 그들은 소프트웨어로 구현이 용이하고 고속 랜덤 치환이 가능하도록

$$Y = X + C \bmod m!$$

를 난수 발생 수식으로 사용하였다. 한편 이 결과를 확장하여

$$Y = AX + C \bmod m! \text{ (A와 C는 상수)}$$

에서 발생되는 난수와 일대일로 대응하는 치환 계산 과정을 벡터와 상삼각 행렬의 곱으로 변환하여 고속으로 계산하는 알고리즘을 제안하고, 이 알고리즘의 출력 치환을 n개 결합하여 치환을 발생하는 랜덤치환 발생기를 설계한 결과가 있었다[13].

4.2 Trapdoor 일방향 함수

Trapdoor는 trapdoor를 알고있는 사람이 일상의 보안 접근 절차를 통하지 않고 액세스를 획득할 수 있도록 하는 프로그램의 어떤 비밀 진입점이다[4, 9, 14].

Trapdoor는 어떤 일련의 특별한 입력을 인식하는 코드 또는 어떤 사용자 ID로부터 수행되거나 나타날 것 같지 않은 일련의 사건들에 의하여 발생되는 코드이다. trapdoor 일방향 함수는 비밀 트랩문을 가진 특별한 형태의 일방향성 함수이다. 한쪽 방향으로 계산하는 것은 쉽지만 다른 반대쪽 방향으로 계산하는 것은 아주 어렵다. 그러나 만약 누군가가 그 비밀을 알고 있다면 반대쪽 방향으로의 함수도 쉽게 계산할 수 있다. 즉 $f(x)$를 주어진 x로 계산하는 것은 쉽지만, x를 주어진 $f(x)$로 계산하는 것은 계산량적으로 매우 어렵다. 그러나, 주어진 $f(x)$에 어떤 비밀 정보 y가 있다면 x를 계산하는 것은 매우 쉽다. 대표적인 trapdoor 일방향 함수로는 소인수 분해를 들 수 있다[8].

지금, p와 q를 소수로 하고 n을 그 곱인 $p \cdot q$라 하자, 이때, $C=[M^e] \bmod n$라는 함수를 생각한다. M으로부터 그 멱승의 나머지 $C=[M^e] \bmod n$을 계산하기 위해서는 p, q를 알 수 있다고는 할 수 없다. p, q를 구하기 위해서는 소인수 분해가 필요하다. 그런데 p와 q가 매우 큰 정수인 경우, $n=p, q$를 소인수 분해하는 것은 매우 어렵다. p와 q를 모를 경우, 멱승의 역을 계산해서 M을 구하는 것은 매우 어렵게된다. 따라서, p와 q를 비밀로 하여 두면 나머지 계산의 멱승은 일방향성을 가지고 있다고 해도 좋을 것이다. 더우기, 이 경우 p와 q를 알고 있다면 그 일방향성은 무너지게 된다. 이와 같은 함수를 trapdoor 일방향 함수라고 부른다.

또다른 예로써 난수 발생알고리즘에 의한 치환발생을 들 수 있다[12, 13]. 난수 발생에 의해 일대일 대응된 랜덤치환을 연속적으로 두개 발생하고, 발생된 두 개의 치환을 결합시켜 생성된 치환으로부터 이 치환에 일대일 대응되는 난수를 구하는 문제를 고려하면 원래 초기에 사용된 난수를 찾는 문제는 trapdoor 일방향성이 존재하게 된다. 두개의 랜덤치환의 결합으로 생성된 치환과 이에 대응되는 난수 사이에는 또 다른 하나의 trapdoor 일방향성 함수의 성질이 성립된다.

4.3 랜덤치환을 이용한 다중레벨 키관리 방안

사이즈가 N인 임의의 치환과 0과 $N!-1$ 사이의 정수가 일대일 대응된다는 성질을 이용해서 선형합동법에 의해 발생되는 난수에 대응되는 치환들을 결합하여 랜덤치환을 발생하는 방법을 고려한다[10]. 사용되는 난수 발생 알고리즘은 구현이 용이한 선형 합동법 $X_{m+1}=X_m+Q \bmod N!$를 이용한다.

4.3.1 Trapdoor 일방향 치환 발생

랜덤치환 발생 알고리즘은 선형 합동법 $X_{m+1}=X_m+Q \bmod N!$에 의해 발생된 연속된 2개의 난수 X_1, X_2에 대응되는 치환 P_1, P_2를 결합하여 새로운 치환 $P_1'=P_1 \circ P_2$를 생성한다. 즉,

$$X_1=X_0+Q \bmod N! \leftrightarrow P_1$$
$$X_2=X_1+Q \bmod N! \leftrightarrow P_2$$
$$P_1'=P_1 \circ P_2$$

한편, 치환 P_1'에 대응되는 난수 X_1'의 발생은 쉽지만 이 X_1'로부터 치환 P_1에 대응되는 난수 X_1 또는 치환 P_2에 대응되는 X_2를 찾는 것은 계산량적으로 어려운 점에 있어 trapdoor 일방향성이 있으며 이러한 성질은 효율적인 계층적 구조의 키 관리 방안에 적용할 수 있다.

4.3.2 사용자 클래스가 완전순서 집합인 경우

사용자 클래스가 $U_1 \le U_2 \le \cdots \le U_n$으로 순서지워져 있을 때 최상위 클래스 U_1의 키를 K_1(임의의 난수)이라 하면, 하위 클래스의 키(K_2, K_3, …, K_n)은 다음과 같이 발생된다.

U_2의 키 $X_1=K_1+Q \bmod N! \leftrightarrow P_1$
$X_2=X_1+Q \bmod N! \leftrightarrow P_2$
$P_2'=P_1 \circ P_2 \leftrightarrow K_2$

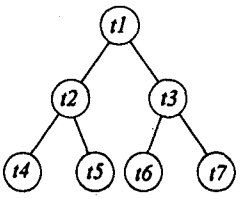
U_3의 키 $X_1=K_2+Q \bmod N! \leftrightarrow P_1$

$$X_2 = X_1 + Q \bmod N! \leftrightarrow P_2$$
$$P_3' = P_1 \circ P_2 \leftrightarrow K_3$$

같은 방법으로 모든 하위 클래스의 키 $K_4, K_5, \cdots, K_n$을 결정할 수 있다.

4.3.3 사용자 클래스가 트리형태의 부분순서가 있는 집합인 경우

사용자 클래스들이 부분순서가 있는 집합인 경우들 중 편의상 이진 트리의 형태인 경우만 고려하도록 한다.



(그림 4) 이진트리 형태의 부분순서가 있는 집합
(Fig. 4) Arbitrary ordered set in binary tree

일반성을 잃지 않도록 (그림 4)인 경우의 완전 이진 트리를 가정한다($n=7$). 즉,

$$U_7 \le U_3 \le U_1$$
$$U_6 \le U_3 \le U_1$$
$$U_5 \le U_2 \le U_1$$
$$U_4 \le U_2 \le U_1$$

앞에서와 마찬가지로 최상위 클래스 U_1의 키를 K_1(임의 난수)라 하면, 하위 클래스 중 좌측노드 클래스 U_2의 키 K_2는 다음과 같이 생성된다.

$$X_1 = K_1 + Q \bmod N! \leftrightarrow P_1$$
$$X_2 = X_1 + Q \bmod N! \leftrightarrow P_2$$
$$P_2' = P_1 \circ P_2 \leftrightarrow K_2$$

우측노드 클래스 U_3의 키 K_3는

$$X_1 = K_1 + Q \bmod N! \leftrightarrow P_1$$
$$X_2 = X_1 + Q \bmod N! \leftrightarrow P_2$$
$$X_3 = X_2 + Q \bmod N! \leftrightarrow P_3$$
$$P_3' = P_1 \circ P_2 \circ P_3 \leftrightarrow K_3$$

같은 방법으로 클래스 U_4의 키 K_4는 K_2를 이용하여 위의 좌측노드에 행해졌던 방법을 그대로 적용하면 구할 수 있다.

5. 성능비교 및 결론

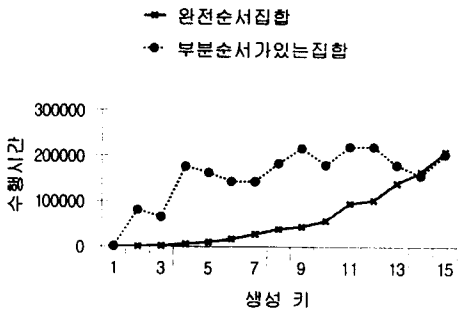
랜덤치환의 발생 및 랜덤치환에 대응되는 난수생성에 소요되는 컴퓨터 수행시간을 측정하였다. 기존 방식과의 비교를 위하여 완전순서 집합인 경우와 부분순서가 있는 집합에서 완전이진 트리의 경우로 나눈 후, IBM PC Pentium(100MHz) 상에서 15개의 키

〈표 1〉 두 방식의 비교(수행시간, 단위 : 초)
〈Table 1〉 Comparisons between two proposed methods (running time, second)

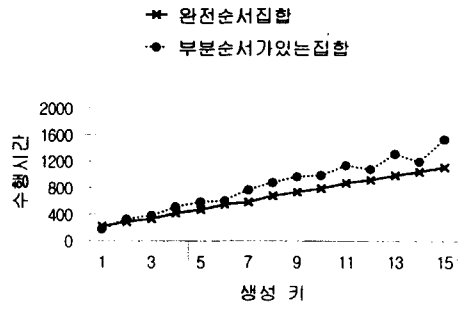
	완전순서 집합		부분순서가 있는 집합	
	Akl과 Taylor의 제안방식	본 논문의 제안방식	Akl과 Taylor의 제안방식	본 논문의 제안방식
Key 1	0.000242	0.000220	0.000240	0.000173
Key 2	0.000538	0.000286	0.080667	0.000330
Key 3	0.001727	0.000332	0.065333	0.000380
Key 4	0.006033	0.000418	0.176667	0.000513
Key 5	0.009800	0.000474	0.163333	0.000590
Key 6	0.016500	0.000548	0.143333	0.000603
Key 7	0.027333	0.000584	0.143333	0.000770
Key 8	0.038667	0.000680	0.183333	0.000880
Key 9	0.044000	0.000736	0.216667	0.000967
Key 10	0.057333	0.000790	0.180000	0.000990
Key 11	0.095333	0.000874	0.220000	0.001137
Key 12	0.102667	0.000922	0.220000	0.001080
Key 13	0.140000	0.000988	0.180000	0.001313
Key 14	0.165000	0.001044	0.156667	0.001193
Key 15	0.210000	0.001110	0.203333	0.001530
평 균	0.061011	0.000667	0.155527	0.000830

(Akl과 Taylor의 제안방식은 $K_0=19$, $M=3625207$를, 제안방식에서는 $X_0=19$, $Q=13181$, $N=10$를 사용하였다.)

를 생성시켜 그 수행시간을 측정하고, 평균값을 산출하였다. 〈표 1〉에서 볼 수 있는 것과 같이 제안방식이 Akl과 Taylor의 제안방식보다 완전순서 집합인 경우 평균 91.47배, 부분순서가 있는 집합인 경우 평균 187.38배 이상 빠름을 알 수 있다. 이들 두 방식의 비교를 그래프로 나타내면 (그림 5)와 (그림 6)으로 각각 주어지며, 특히 (그림 5)의 Akl과 Taylor의 제안방식에서 부분순서가 있는 집합의 경우 그래프가 불규칙하게 나타나는 이유는 트리가 왼쪽이 오른쪽보다 더 큰 소수가 할당된 완전이진트리로써 각 수행시간이 오른쪽으로 갈수록 증가하기 때문이다. 아울러 각 사용자 클래스 U_i가 유지해야할 정보의 크기는 Akl과 Taylor의 제안방식에서는 t_i와 자신의 키 K_i를 가지고 있어야 하는 반면에 본 논문의 제안방식은 X_i만을 가지고 있으면 되므로 소유해야 할 정보량의 측면에서 제안된 방식이 유리함을 알 수 있다.



(그림 5) Akl과 Taylor의 제안방식(수행시간. 단위 : μs)
(Fig. 5) Akl and Taylor's scheme(running time, μs)



(그림 6) 본 논문의 제안방식(수행시간, 단위 : μs)
(Fig. 6) Proposed scheme(running time, μs)

제안된 방식은 대칭키 암호시스템의 키 관리 방안에 광범위하게 적용될 수 있으며, 특히 애널로그 스크램블링 기법을 이용하는 음성 보안 시스템 또는 TV한정 시스템 등[3]에서 위에서 발생된 치환을 직접 동작 키로서 사용할 수 있으며, 특히 계층적인 구조를 지닌 기관 등의 키관리에 매우 효율적으로 사용될 수 있는 방안이다.

참 고 문 헌

[1] S.G. Akl & H. Meijer, "A Fast Pseudo Random Permutation Generator with Application to Cryptology", Advanced in Cryptology: Proceedings of CRYPTO 84, pp. 260-275, 1985.

[2] S.G. Akl & P.D. Taylor, "Cryptographic solution to a multilevel security problem", Advanced in Cryptology: Proceedings of CRYPTO'84, pp. 237-248, 1985.

[3] H.J. Beker & F.C. Piper, Secure Speech Communications, Academic Press, 1984.

[4] D.E.R. Denning, "Cryptography and data Security, Addison-Wesley, Reading, Massachusetts (1982).

[5] Walter Fumy, "Key Management", Computer Security & Industrial Cryptography'91 LNCS Vol. 741, p. 132-145, 1991.

[6] G.D. Knott, "A Numbering System for Permutations of Combinations", Comm. of ACM, June Vol. 19, No. 6, pp. 355-356, 1976.

[7] D.E. Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, Addison-Wesely, 1981.

[8] R.L. Rivest, A. Shamir & L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", commun. ACM, Vol. 21, (1978).

[9] Bruce Schneier, Applied Cryptography, 2nd, John Wiley & Sons, p. 30, 1996.

[10] N. J. A. Sloane, "Encrypting by Random Rotations", Advances in Cryptology, Proceedings of CRYPTO 82, pp. 71-128, 1983.

[11] 현대 암호학, 한국전자통신연구소편, 1991.

[12] 이경현, "랜덤치환의 안전성과 통계적 검정", 한국통신정보보호학회지, Vol. 3, No. 4, pp. 63-70, 1993.

[13] 고승철, 이경현, "랜덤 치환 고속 발생기", Proceedings of 1st Workshop in Applied Mathematics, pp. 379-383, 1993.

[14] 류춘열, 이경현, 박지환 역, "암호이야기", 동영출판사, 1996.

김 주 석

1995년 밀양산업대학교 컴퓨터공학과 졸업(공학사)

1995년~현재 부경대학교 전산정보학과 석사과정

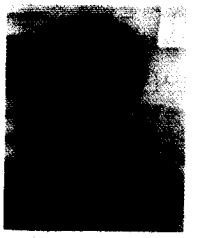
관심분야: 정보보호론, 컴퓨터 보안, 암호학

신 원

1996년 부경대학교 전자계산학과 졸업(이학사)

1996년~현재 부경대학교 전자계산학과 석사과정

관심분야: 네트워크 이론(성능분석), 광대역 통신망, 대기체계론, 정보보호론, 컴퓨터 보안, 암호학



이 경 현

1982년 경북대학교 사범대학 수학교육과 졸업(이학사)

1985년 한국과학기술원 응용수학과 졸업(이학석사)

1992년 한국과학기술원 수학과 졸업(이학박사)

1985년 2월~1991년 2월 한국 전자 통신 연구소 연구원

1991년 3월~1993년 2월 한국 전자 통신 연구소 선임연구원

1993년 3월~현재 부경대학교(구 부산수산대학교) 전임강사, 조교수

1995년 7월~1996년 7월 Univ. of Adelaide, 응용수학과, Australia 방문교수

관심분야: 네트워크 이론(성능분석), 광대역 통신망, 대기체계론, 정보보호론, 컴퓨터 보안, 암호학