한국산학기술학회논문지
Vol. 11, No. 5, pp. 1693-1701, 2010

계층 구조에서의 해쉬 체인과 랜덤난수의 논리 연산을 이용한 그룹키 생성 기법

김현철[1*], 이영구[2], 김정재[2], 이광형[3]
[1]한국과학기술정보연구원 정보화전략팀, [2]숭실대학교 컴퓨터학과, [3]서일대학 인터넷정보과

Group Key Generation Scheme using Logical Operation of HashChain and Random Number in Hierarchy Structures

Hyun-Chul Kim[1*], Young-Gu Lee[2], Jung-Jae Kim[2] and Kwang-Hyung Lee[3]
[1]**Korea Institute of Science and Technology Information**
[2]**Division of Computer Science, Soongsil University**
[3]**Division of Internet Information, Seoil College**

요 약 본 논문에서는 조직내 상하 역할 구분이 명확한 계층 구조 환경에서의 효율적인 그룹키 생성을 위한 요구조건을 제시하고 이를 만족하는 그룹키 생성 기법을 제안한다. 제안하는 기법은 일방향성의 해쉬 체인을 이용해 생성한 계층식별자와 접근권한에 따라 랜덤하게 생성하여 관리되는 그룹식별자와의 논리합 연산을 통해 키를 생성한다. 이를 통해, 기존의 정적 그룹키 생성 기법의 상위그룹 사용자로 하여금 과도한 키 생성 정보 소유 문제를 해결함과 동시에 하위그룹 사용자로 하여금 상위그룹 사용자의 키 정보를 유추할 수 없도록 하였다. 또한, 기존의 슈퍼 그룹키 생성 기법, 다중 계층 그룹키 생성 기법과의 실험을 통한 비교분석 결과 생성되는 전체키의 수, 사용자 소유 키의 수, 정보 암복호화에 사용되는 키의 수, 키 자체 확장성 등과 같은 여러 항목에서 기존 기법과 동일하거나 우수함을 확인할 수 있었다.

Abstract In this paper, requirements of efficient group key creation in multiple hierarchy structure environment with clear distinction of hierarchical roles within organizations are explained and the method of creating a group key that satisfies such requirements is proposed. The proposed method creates the group key through logical sum operation of hierarchy identifier created using uni-directional hash chain and group identifier randomly created according to the access right. The problem of excessive possession of key information by upper group users in the existing static group key creation technique was resolved. At the same time, lower group users were prevented from deducing key information of upper group users. In addition, as a result of comparative analysis performed with an experiment on existing super group key creation technique and multiple hierarchy group key method, the proposed method was found to be equivalent or superior to existing method in terms of various items including the total number of keys created, the number of keys possessed by users, the number of keys used for encoding and decoding of information, and expandability of keys.

Key Words : Group Key, Hash Chain, Hierarchy Structures, Random Number, Cryptograph

1. 서론

최근 일부 거대 기업의 고객정보 유출 사고를 계기로 내부정보 유출에 따른 부작용이 심각한 사회적 문제로 대두되고 있다. 이러한 사고로부터 조직내 중요 정보를 보호하고 유지하기 위해서는 접근을 요청하는 요청자가 등록된 사용자임을 증명하기 위한 인증기술, 부여된 권한에 부합된 정보에만 접근하도록 하는 접근제어 기술 그

*교신저자 : 김현철(dmzpolice@kisti.re.kr)
접수일 10년 01월 07일 수정일 10년 05월 06일 게재확정일 10년 05월 13일

리고 해당정보에 기밀성을 유지하기 위한 암호화 및 키 생성 기술이 요구된다[1-3]. 특히, 계층 및 그룹간에 역할이 명확하게 구분되어 운영되며 동일 그룹내 모든 구성원들이 공용키를 이용하여 정보에 기밀을 유지하는 조직에서는 그룹 구성원 변동에 따른 그룹키 생성이 우선적으로 요구된다.

현재 대표적인 그룹키 생성 기법으로는 그룹 구성원의 수를 사전에 미리 예상하여 그에 따른 키를 생성한 후 그룹 구성원에 변동이 발생할 때마다 생성한 키를 분배하는 정적 그룹키 생성 기법과 그룹 구성원의 변동에 따라 새로운 키를 생성하여 분배하는 동적 그룹키 생성 기법이 있다[9].

정적 그룹키 생성 기법은 구성원 변동에 따른 키 생성 및 갱신 등의 복잡한 처리 과정을 실시간으로 처리하지 않는다는 점에서 동적 그룹키 생성 기법에 비해 효율적이다. 그러나 정확한 그룹 구성원의 수를 미리 예측하기 어려우며, 사전에 키를 생성하여 보관한다는 점에서 키 관리상에 문제가 발생한다. 또한 그룹 구성원에 여러 변동 사유 중에서 탈퇴만을 고려했다는 점에서도 한계가 있다[9]. 이러한 정적 그룹키 생성 기법에는 슈퍼 그룹키 생성 기법[5,11], 다중 계층 그룹키 생성 기법[11]등이 대표적이다.

본 논문에서는 정적 그룹키 생성 기법에 문제를 해결함과 동시에 다중 계층 환경에 적용하기 위해 필요한 그룹키 생성 조건을 제시하고 이를 만족시키는 그룹키 생성 기법을 제안한다. 제안하는 기법은 일방향성의 해쉬체인 알고리즘을 이용해 생성한 계층식별자(계층 구분)와 랜덤난수를 이용해 생성한 그룹식별자(동일 계층내 서로 다른 그룹 식별)의 논리적 연산을 통해 효율적인 키 생성을 수행함으로써 정적 그룹키 관리 기법의 상위그룹 사용자에 하위그룹키 생성정보 소유 문제 및 하위그룹 사용자에 상위그룹 키 생성 정보 유추 문제를 해결한다.

본 논문은 다음과 같은 구성을 통해 전개해 나가고자 한다. 2장은 본 논문의 정당성을 증명하기 위한 선행연구로써 정적 그룹키 생성 기법의 일종인 슈퍼 그룹키 생성 기법과 다중 계층 그룹키 생성 기법의 특징 및 장·단점을 분석하고 다중 계층 구조 환경에서의 그룹키 생성을 위해 필요한 요구 조건에 대하여 기술한다. 3장에서는 2장에서 도출된 문제점을 해결하고 제시된 요구조건을 만족시킬 수 있는 그룹키 생성 기법에 대하여 기술한다. 4장에서는 제안 알고리즘에 성능 평가를 위한 시스템 구현 및 기존의 슈퍼 그룹키 생성 기법 그리고 다중 계층 그룹키 생성 기법과의 실험을 통한 비교분석 결과를 기술하였다. 마지막으로 5장에서는 결론을 맺는다.

2. 관련연구

2.1 슈퍼 그룹키 생성 기법

슈퍼 그룹키 생성 기법[11]은 그룹의 접근권한 계층에 따라 그룹키를 차등 생성한 후 정보의 가장 중요한 부분을 최상위 계층에 그룹키로 먼저 암호화 한 후 그 다음으로 중요한 부분을 다음 계층에 그룹키로 차등 암호화해 가는 방식으로 하위 계층 그룹에 속한 사용자는 상위 계층에 정보를 확인할 수 없는 장점을 지닌다. 그러나 상위 계층으로 갈수록 하위 계층에 그룹키 정보를 모두 소유해야 한다는 점에서 키 생성 및 키 교환의 어려움이 존재한다. 또한, 정책 기반의 암호화가 아닌 단순 기밀 정도에 따라 암호화를 수행한다는 한계를 지니고 있다[6,10].

2.2 다중 계층 그룹키 생성 기법

다중 계층 그룹키 생성 기법[11]은 기존에 슈퍼 그룹키 생성 기법의 문제점을 해결하기 위해 제안된 방법으로 사전에 각각의 그룹 계층에 맞는 키를 생성하고 접근권한에 따라 필요한 부분만을 암호화하여 전달하는 모델이다[12]. 이 모델은 슈퍼 암호화 모델에 단점인 중복 암호화에 따른 연산량 문제를 해결한다. 그러나 이 모델 역시 슈퍼 암호화 모델과 마찬가지로 상위그룹 사용자는 하위그룹의 모든 키생성 정보들을 소유해야 하는 문제를 지닌다[13,14].

2.3 그룹키 생성을 위한 요구조건

조직내 상하 역할 구분이 명확한 다중 계층 구조 환경에서의 조직내 중요 정보에 대한 기밀성을 유지하기 위해서는 다음 조건들과 같은 요구사항들을 고려하여야한다.

조건 1 : 그룹 내 모든 사용자 U_n은 동일한 권한 A를 가지는 사용자 U_n'와 하나의 서브 그룹 UG_n을 구성하며 이러한 그룹은 최소 한 개부터 다수 개가 존재할 수 있다.

$UG_n = \{U_1, U_2, \cdots \cdots, U_{n-1}, U_n\}$ $(U_1 \cdots U_n = A)$

$\{UG_n, UG_{n+1}, \cdots \quad \cdots \quad \cdots\}$ $(\sum_{n=1}^{n=\infty})$

조건 2 : 조건 1에서 각 사용자그룹 UG_n은 접근관계를 바탕으로 다른 사용자 그룹 UG_n'에 대하여 관계 "≤"에 의해 부분적으로 정렬되며 그룹 UG_n에 속한 사용자 U_n 역시 다른 사용자그룹 UG_n'를 구성하는 사용자 U_n에 대하여 그

룹 관계 “≤”에 따라 부분적으로 정렬된다.

$\{UG_n, UG_{n+1}, \cdots\} (\sum_{n=1}^{n=\infty}, (UG_n \le UG_{n+1}))$ 일 때

$UG_nA \le UG_{n+1}A$

$\therefore UG_nA\{U_1 \cdots U_n\} \le UG_{n+1}A\{U_1 \cdots U_n\}$
$(UG_n \cdots UG_{n+1} = A)$

조건 3 : 사용자 그룹과 마찬가지로 모든 데이터 D_n은 자신에게 접근할 수 있는 사용자 그룹 UG_n이 동일한 다른 데이터 D_n'와 하나의 데이터 서브그룹 G_n을 구성할 수 있으며 이러한 데이터 그룹은 최소 한 개부터 다수 개가 존재 할 수 있다.

$DG_n=(D_1,D_2, \cdots \cdots D_{n-1}, D_n), (D_1 \cdots D_n = UG_n)$

$\{DG_n, DG_{n+1}, \cdots \cdots \cdots\} (\sum_{n=1}^{n=\infty})$

조건 4 : 조건 3의 각 데이터그룹 DG_n은 접근 관계를 바탕으로 다른 데이터그룹 DG_n'에 대하여 관계 “≤”에 의해 부분적으로 정렬되며 그룹 DG_n에 속한 데이터 D_n 역시 다른 데이터그룹 DG_n'를 구성하는 데이터 D_n에 대하여 그룹 관계 “≤”에 따라 부분적으로 정렬된다.

$\{DG_n, DG_{n+1}, \cdots\} (\sum_{n=1}^{n=\infty}, (DG_n \le DG_{n+1} \le UG_n))$ 일 때 $DG_nA \le DG_{n+1}A \le UG_nA$

$\therefore DG_nA\{D_1 \cdots D_n\} \le DG_{n+1}A\{D_1 \cdots D_n\} \le UG_nA\{U_1 \cdots U_n\}$

조건 5 : 모든 그룹에 키는 유일해야 한다.

위의 조건을 통해 통합 조건 6을 유추할 수 있다.

조건 6 : 사용자그룹 UG_n과 데이터그룹 DG_n을 가지는 통합그룹 TG_n이 존재하고 또다른 사용자 UG_n'와 데이터그룹 DG_n'를 가지는 또 다른 통합그룹 TG_n'가 존재 할 때 그룹 TG_n이 또 다른 그룹 TG_n'에 부분 집합관계에 놓인다면 TG_n'의 사용자 그룹 UG_n'를 구성하는 모든 사용자 U는 그룹 TG_n에 속한 사용자가 접근할 수 있는 데이터에 접근할 수 있는 사용자 집합의 부분집합 관계 “≤”를 형성한다.

$TG_n = \{UG_n, DG_n\}, TG_n' = \{UG_n', DG_n'\}$
$G_n = \{DG_n \le UG_n\}, G_n' = \{DG_n' \le UG_n'\}$
$TG_n \le TG_n'$ 라면, $DG_n \le DG_n', UG_n \le UG_n'$

$\therefore TG_n\{DG_n\} \le TG_n'\{UG_n'\}$

결 론 : 관계 “≤”를 가지는 여러 개의 그룹이 존재할 때 모든 그룹은 서로 다른 그룹키 K를 소유하여야 하고, 상위그룹 TG_n'의 사용자그룹 UG_n'를 구성하는 모든 사용자 U는 부분 집합 관계를 가지는 하위그룹 TG_n에 대한 그룹키를 소유하거나 생성할 수 있어야 하며, 하위그룹 TG_n을 구성하는 사용자그룹 UG_n에 사용자 U는 상위그룹 TG_n'의 그룹키를 소유하거나 생성해서는 안된다.

$TG_n = \{UG_n, DG_n, K_n\}, TG_n' = \{UG_n', DG_n', K_n'\}$
$(K_n \ne K_n')$이고 $TG_n \le TG_n'$ 라면
$DG_n \le DG_n', UG_n \le UG_n', K_n \le K_n'$

$\therefore TG_n\{K_n\} \le TG_n'\{UG_n'\{U_1 \cdots U_n\}\}$

즉 그룹 TG_n'에 속한 사용자 U는 자신의 그룹키 K_n'와 그룹 TG_n의 그룹키 K_n을 소유하거나 생성할 수 있어야 하며 그룹 TG_n의 사용자 U는 자신의 그룹키 K_n만을 소유하면 모든 조건을 만족 시킬 수 있다.

$TG_n = \{K_n\}, TG_n' = \{K_n, K_n'\} (G_n \le G_n')$
$(K_n \ne K_n')$

3. 제안하는 시스템

3.1 그룹키 생성 알고리즘

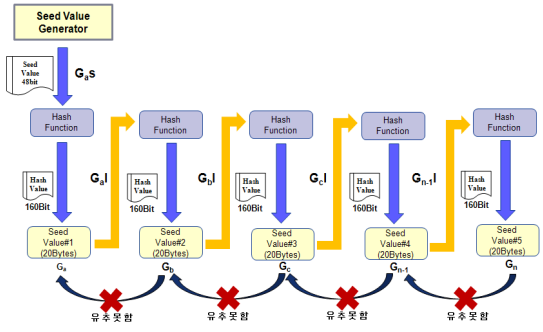
2.3절에서 결론으로 제시한 조건 중에서 “상위그룹 사용자는 하위그룹 키를 소유하거나 생성할 수 있어야 한다.”라는 조건과 “모든 키는 고유해야 한다.”라는 조건을 만족시키기 위해 본 논문에서는 해쉬 체인 키 알고리즘을 통해 생성한 계층식별자와 키 서버에서 생성하여 그룹별로 부여한 그룹식별자의 논리합 연산을 통해 키를

생성함으로써 위의 조건을 모두 만족 시킬 수 있는 다중 계층 구조 환경에서의 해쉬 체인과 랜덤난수의 논리합 연산을 이용하여 효율적인 그룹키 생성 기법을 제안한다.

3.1.1 계층식별자 생성

해쉬 체인 알고리즘은 역함수가 존재하지 않는 일방향성의 특성을 가지는 해쉬 함수를 이용하여 체인 형태의 키를 생성하는 것으로 상위 계층의 값을 통해 하위 계층의 값들을 도출해 낼 수 있으나 그 역계산은 불가능한 키 생성 알고리즘이다. 이러한 해쉬 체인 알고리즘을 통해 생성된 그룹키는 상위 계층 사용자가 하위 계층 사용자의 키를 모두 소유해야 하는 슈퍼 그룹키, 다중 레벨 그룹키 생성 모델의 문제를 해결한다. 그러나 동일한 입력에 대해 동일한 결과값을 생성하는 해쉬 함수에 특성으로 인해 동일 계층 다른 권한을 가지는 그룹에 대해 동일한 그룹키가 생성된다는 문제가 존재한다. 따라서 본 논문에서는 이러한 해쉬 체인 알고리즘의 특성을 고려해 해쉬 체인 알고리즘을 이용하여 그룹키를 생성하는 것이 아닌 단지 그룹 계층을 구분 짓는 계층식별자를 생성하는 용도로만 사용한다.

그림 1은 해쉬 체인 알고리즘을 이용한 계층식별자 생성 과정으로써 최상위그룹 사용자는 그림 1에서와 같이 SEED값 생성기를 통해 생성된 초기 SEED값을 입력으로 160비트 크기의 계층식별자를 생성한다. 또한 자신의 계층식별자를 입력으로 자신의 서브그룹에 대한 식별자를 생성할 수도 있다. 이러한 일련의 연속된 과정을 통해 최상위그룹 사용자는 자신보다 하위에 있는 모든 그룹의 대한 식별자를 생성할 수 있게 된다.



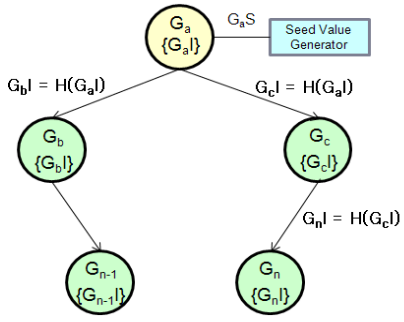
[그림 1] 계층식별자 생성 과정

예를 들어 최상위그룹 G_a가 있다고 가정했을 때 최상위그룹 사용자 U는 SEED값 생성기를 통해 부여받은 초기 SEED값에 대하여 해쉬 함수를 적용한 계층식별자 G_aI를 생성할 수 있으며 생성된 G_aI를 입력으로 차상위그룹 G_b에 대한 계층식별자 G_bI를 생성할 수 있다. 이와 같은 과정을 반복해 최하위그룹인 G_n에 계층식별자 G_nI 까지 생성할 수 있게 된다. 이러한 해쉬 체인 알고리즘을 이용해 계층식별자를 생성하는 과정은 다음과 같다.

$$H(G_aS) \rightarrow G_aI,\ H(G_aI) \rightarrow G_bI,\ \cdots,\ H(G_{n-1}I) \rightarrow G_nI$$

$$\therefore\ UG_a = \{G_aI, G_bI, G_{n-1}I, G_nI\},\ UG_n = \{G_nI\}\ (G_n \leq G_a)$$

위와 같은 방법으로 생성된 계층식별자는 계층적인 연결 관계를 가지며 그림 2의 방향성 비 순환 그래프로 표현될 수 있다.



[그림 2] 계층식별자 방향성 비 순환 그래프

3.1.2 그룹키 생성

그룹키 생성 알고리즘은 다음의 전제를 고려한다.

전 제 1 : 모든 그룹은 하나의 계층식별자와 하나의 그룹식별자를 가진다.

전 제 2 : 계층식별자와 그룹식별자는 해당 그룹의 사용자 이외에는 비공개 정보이며 그룹식 별자는 그림 3과 같이 접근권한에 따라 적절하게 생성되어 관리되어야 한다.

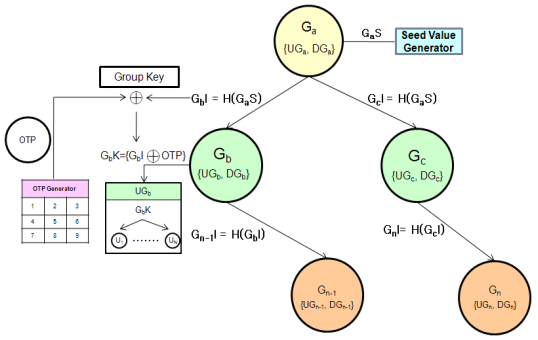
OTP REPOGITORY					
그룹 / OTP	G_a	G_b	·····	G_{n-1}	G_n
G_aOTP	A	-	-	-	-
G_bOTP	A	A	-	-	-
:	A	A	A	-	-
G_{n-1}OTP	A	A	A	A	-
G_nOTP	A	A	A	A	A

[그림 3] 그룹식별자 접근권한 매트릭스

전 제 3 : 계층식별자와 그룹식별자의 논리적 연산을 통해 생성된 그룹키 중 일부가 ASCII 코드 값과 충돌이 발생할 위험이 존재한다. 따라서 이러한 충돌을 방지하기 위한 별도의 키 셋 정의가 필요하며, 키 셋은 중복의 위험을 최소화할 수 있을 만큼 충분한 값을 가져야한다.

본 논문에서 제안하는 그룹키는 해쉬 체인을 이용해 생성한 계층식별자와 그룹별로 부여된 그룹식별자의 논리합 연산을 통해 생성되며, 그룹식별자는 접근통제 매트릭스에 의해 접근권한이 있는 그룹사용자에게만 적절히 공개되어야 한다. 이때 그룹식별자는 해당 그룹의 계층을 구분 짓는 계층식별자와 달리 생성되는 그룹키의 고유성을 만족시키기 위한 요소로 사용된다.

그림 4는 본 논문에서 제안하는 그룹키 생성 과정으로써 최상위 계층식별자 G_aS를 가지며 사용자 그룹 UG_a와 데이터 그룹 DG_a로 구성된 최상위그룹 G_a가 존재 할 때 최하위그룹 G_n에 키를 구하고자 한다면 다음과 같은 절차를 거치게 된다.



[그림 4] 그룹키 생성 과정

과 정 1 : 그룹 G_a의 사용자는 자신의 초기 SEED값 G_aS에 대하여 해쉬 체인 함수 H()를 이용하여 그룹 G_n에 계층식별자인 G_nI를 생성한다.

$G_aS \rightarrow H(G_aS) \rightarrow G_cI$, $G_cI \rightarrow H(G_cI) \rightarrow G_nI$ ($G_n \leq G_c \leq G_a$)

표 1은 초기 SEED VALUE 123456에 대하여 해쉬 체인을 수행하여 생성한 그룹 G_n의 계층식별자이다.

[표 1] 생성된 계층식별자

Group	SEED NAME	SEED VALUE	HASH VALUE
G_a	G_aS	123456	aD3927test
G_c	G_cI	aD3927test	DmZ78BYnce
G_n	G_nI	DmZ78BYnce	KMZk38TYxI

과 정 2 : 접근통제 매트릭스를 통해 그룹 G_a의 권한을 확인 한 후 그룹 G_n과 같은 경로상의 상위그룹일 경우 G_n의 그룹식별자를 가져온다. 표 2는 사전에 생성해 놓은 그룹 식별자이다.

[표 2] 그룹식별자

Group	그룹식별자
G_a	1dtea15nke
G_c	e4df4d3mYT
G_n	h45yk2DuUc

과 정 3 : 키 생성에 사용될 키 셋 테이블을 생성한다. 표 3은 랜덤하게 생성한 키 셋 테이블을 보여주고 있다.

[표 3] 키 셋 테이블 생성

	000	001	010	011	100	101	110	111
000	a	x	Q	w	1	R	2	s
001	C	i	U	B	r	P	g	V
010	/n	K	o	h	J	W	q	3
011	b	y	A	T	m	I	F	t
100	X	j	z	0	L	S	f	M
101	_	D	c	Z	E	O	4	u
110	k	8	n	7	6	l	5	p
111	9	Y	G	d	v	H	e	N

과 정 4 : 생성된 그룹 G_n의 SEED값 "KMZk38TYxI"에 대해 임의적으로 생성한 키 셋 테이블을 표 4와 같이 생성한 후 표 5와 같은 이진정보를 생성한다.

[표 4] 그룹 G_n의 SEED 값 테이블

	1	2	3	4	5	6	7	8
1	a	x	Q	w	1	R	2	s
2	C	i	U	B	r	P	g	V
3	/n	K	o	h	J	W	q	3
4	b	y	A	T	m	I	F	t
5	X	j	z	0	L	S	f	M
6	_	D	c	Z	E	O	4	u
7	k	8	n	7	6	l	5	p
8	9	Y	G	d	v	H	e	N

[표 5] 그룹 G_n의 이진 SEED값

K→(3,2) (010,001)	M→(5,8) (100,111)	Z→(6,4) (101,011)
k→(7,1) (110,000)	3→(3,8) (010,111)	8→(7,2) (110,001)
T→(4,4) (011,011)	Y→(8,2) (111,001)	x→(1,2) (000,001)
I→(7,6) (110,101)		

과 정 5 : 생성된 그룹 G_n의 그룹식별자 "h45yk2DuUc"에 대해 임의적으로 생성한 키 셋 테이블을 표 6과 같이 생성한 후 표 7과 같은 이진 정보를 생성한다.

[표 6] 그룹 G_n의 그룹식별자 값 테이블

	1	2	3	4	5	6	7	8
1	a	x	Q	w	l	R	2	s
2	C	i	U	B	r	P	g	V
3	/n	K	o	h	J	W	q	3
4	b	y	A	T	m	I	F	t
5	X	j	z	0	L	S	f	M
6	_	D	c	Z	E	O	4	u
7	k	8	n	7	6	l	5	p
8	9	Y	G	d	v	H	e	N

[표 7] 그룹 G_n의 이진 그룹식별자 값

h→(3,4) (010,011)	4→(6,7) (101,110)	5→(7,7) (110,110)
y→(4,2) (011,001)	k→(7,1) (110,000)	2→(1,7) (000,110)
D→(6,2) (101,001)	u→(6,8) (101,111)	U→(2,3) (001,010)
c→(6,3) (101,010)		

과 정 6 : 과정 4와 과정 5를 통해 생성한 계층식별자와 그룹식별자에 대해 표 8과 같이 논리합 연산을 수행한 후 그룹키를 생성 한다.

[표 8] 그룹 G_n의 그룹키 값

번호	키 요소	키 요소 이진 값	연산 결과	값
1	K XOR h	(010001 ⊕ 010011)	010011	h
2	M XOR 4	(100111 ⊕ 101110)	101111	u
3	Z XOR 5	(101011 ⊕ 110110)	111111	N
4	k XOR y	(110000 ⊕ 011001)	111001	Y
5	3 XOR k	(010111 ⊕ 110000)	110111	p
6	8 XOR 2	(110001 ⊕ 000110)	110111	p
7	T XOR D	(011011 ⊕ 101001)	111011	d
8	Y XOR u	(111001 ⊕ 101111)	111111	N
9	x XOR U	(000001 ⊕ 001010)	001011	B
10	I XOR c	(110101 ⊕ 101010)	111111	N

4. 실험 및 비교분석

4.1 실험 환경

제안하는 기법의 성능 평가를 위한 시스템은 Visual C# 2008과 MS-SQL 2008을 이용하여 구현하였고 정보 전송을 위한 통신 프로토콜은 단순객체접근프로토콜(SOAP : Simple Object Access Protocol)을 사용하였다. 전송되는 키 값의 기밀성을 확보하기 위하여 RSA 공개키 알고리즘과 AES 대칭키 알고리즘을 각각 적용하였다. 마지막으로 실험은 Intel(R) Core2 Quad Q6600 2.40GHz, 2048M RAM의 PC에서 MS-Windows XP Professional Service Pack 3 운영체제 하에서 수행하였다.

4.2 실험 개요

키 생성 부분에 대한 실험을 위해 해당 그룹의 권한은 Top_Secret, Secret, Confidential, Sensitive, Public으로만 한정하였으며 계층식별자는 SHA-1 해쉬 함수를 이용해 체인 형식으로 20바이트 크기로 생성하였다. 또한 그룹식별자는 영문대문자, 영문소문자, 숫자, 특수문자의 랜덤 조합을 통해 생성하였으며 그룹키는 계층식별자와 그룹식별자를 이진수로 변환한 후 논리합 연산을 통해 20바이트 크기로 생성하였다.

키 생성 부분에 대한 실험에서 가장 중요한 요소는 이론적으로는 불가능하지만 중복키가 발생하지 않도록 해야 하며, 이를 위해 데이터베이스에 중복된 키가 입력되지 않도록 설계 하였으며 키 생성에 대한 실험은 다음과 같이 각각의 요소에 대해 중점을 두고 실험을 진행하였다.

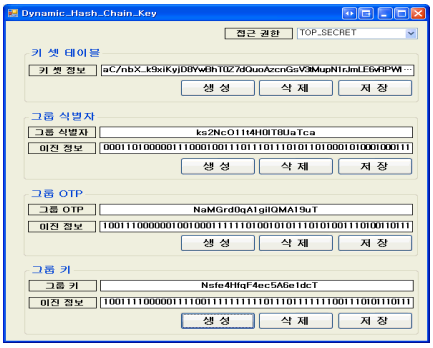
첫 째 : 제안된 알고리즘에 입각해 키 셋 테이블이 정확하게 형성되는지 그리고 생성한 키 셋 테이블에 대한 충돌이 발생하는지에 대한 실험을 지속적으로 수행하였다.

둘 째 : SHA-1 해쉬 함수 체인을 이용해 생성한 계층식별자 값에 대한 중복값이 생성되는지 여부를 확인하기 위한 실험을 지속적으로 수행하였다.

셋 째 : 랜덤하게 생성한 20바이트 크기의 그룹식별자에 대한 충돌 실험을 지속적으로 수행하였다.

넷 째 : 계층식별자와 그룹식별자에 논리합 연산을 통해 생성된 그룹키에 대한 충돌이 발생하는지에 대하여 실험을 수행하였다.

그림 5는 제안한 알고리즘에 따라 키를 생성해주는 키 생성기 이다.



[그림 5] 키 생성기

4.3 키 생성에 따른 주요 항목 비교분석

본 절에서는 슈퍼 그룹키 생성 기법, 다중 계층 그룹기 생성 기법 그리고 제안하는 기법에 대하여 비교분석한 결과를 기술한다. 비교분석 항목으로는 생성되는 총 키의 수, 사용자 소유 키의 수, 정보 암·복호화에 사용되는 키의 수, 키 자체의 대한 확장성, 동일 계층 다른 권한을 가지는 정보에 대한 키 생성 여부 등 다섯 가지 항목을 선정하였다.

먼저 하나의 정보에 대하여 생성되는 총 키의 수를 비교분석한 결과 슈퍼 그룹키 생성, 다중 계층 그룹키 생성, 그리고 제안하는 기법 모두 정보에 접근 할 수 있는 권한 설정 수에 따라 키 생성량이 달라지는 특징을 보였다. 즉, 정보의 접근권한 설정 수가 n이라면 생성되는 키의 수는 최소 한 개에서 접근권한의 수 n만큼 생성된다. 그러나 일반적으로 접근권한의 수가 적절한 수준에서 생성됨에 따라 생성되는 키의 수는 많지 않은 편이다.

두 번째로 사용자가 보유해야 되는 키 개수에 대한 비교분석 결과로써 슈퍼 그룹키 생성의 경우 암호화 필드 수에 따라 사용자가 소유해야 하는 키의 수가 달라지는 특성이 존재하며, 4.2절 실험 개요에서 제시한 최상위 권한을 가진 사용자를 기준으로 할 때 최대 다섯 개의 키를 소유하게 된다.

슈퍼 그룹키 생성 기법과 마찬가지로 다중 계층 그룹키 기법 또한 암호화 필드 수에 따라 사용자가 소유해야 하는 키의 수가 달라지는 특성을 지니고 있다. 즉, 슈퍼 그룹키 생성 기법과 다중 계층 그룹키 기법 모두 사용자 보유 키의 개수는 최소 한 개부터 암호화 필드 수 만큼 보유하여야 함을 확인할 수 있다.

그러나 제안하는 기법의 경우 최상위 사용자라 할지라도 사용자가 속한 그룹의 키만 보유하면 다른 하위그룹의 키를 유추 할 수 있는 고유 특성으로 인해 사용자는 단지 자신의 그룹키 하나만을 소유함을 확인 할 수 있다. 즉, 슈퍼 그룹키 생성 기법, 다중 계층 그룹 기법과 달리 접근권한에 따른 의존성이 존재하지 않음을 확인할 수 있다.

세 번째로 정보 암·복호화에 사용되는 키의 수에 대한 비교분석 결과로써 슈퍼 그룹키 기법과 다중 계층 그룹키 기법은 암호 필드의 수에 따라 암·복호화에 사용되는 키의 수가 달라진다. 특히, 암호화 필드의 수는 사용자에게 설정된 접근권한 계층과 동일하다는 점에서 사용자가 소유해야 되는 키 개수와 정보 암·복호화에 필요한 키의 수가 동일한 특성을 지닌다. 제안하는 그룹키 생성 기법의 경우 사용자가 소유해야 하는 키의 수와 정보 암·복호화에 사용되는 키의 수가 동일한 특성을 지닌다.

네 번째로 키 자체에 대한 확장성을 비교분석한 결과이다. 이 항목은 사용자가 소유하고 있는 키를 이용하여 또 다른 키를 생성할 수 있는가에 대한 항목으로써 슈퍼 그룹키 생성 기법과 다중 계층 그룹키 생성 기법은 사전에 정의된 형식에 따라 그룹키를 생성하기 때문에 키 자체에 대한 확장성을 제공할 수 없다. 그러나 제안하는 기법의 경우 사용자가 소유한 키가 하위그룹에 키를 생성하기 위한 입력 값으로 이용된다는 점에서 키 자체에 대한 확장성을 지원한다.

마지막으로 동일 계층 다른 권한을 가지는 그룹에 대한 키 생성 여부를 비교분석한 결과이다. 슈퍼 그룹키 생성 기법과 다중 계층 그룹키 생성 기법은 하나의 정보에 대해 여러 개의 그룹을 지정할 수 있지만 동일 계층 다른 권한에 대해서는 고려하지 않는다. 그러나 제안하는 기법은 해쉬 체인을 통해 계층식별자를 생성한 후 랜덤하게 생성된 그룹식별자와의 논리합 연산을 통해 키를 생성하기 때문에 동일 계층 다른 권한에 따른 지원이 가능하다. 표 9는 키 생성에 따른 주요 항목에 대한 비교분석 결과를 도표화 한 것이다.

[표 9] 키 생성에 따른 주요 항목 비교

	슈퍼 그룹키	다중레벨 그룹키	제안하는 기법
생성되는 총키의 수	$\sum_{K=1}^{K=N}$	$\sum_{K=1}^{K=N}$	$\sum_{K=1}^{K=N}$
	N= 접근 권한 수		
사용자 소유키의 수	$\sum_{K=1}^{K=N}$	$\sum_{K=1}^{K=N}$	1
	N = 접근 권한 레벨		
암·복호화에 사용되는 키의 수	$\sum_{K=1}^{K=N}$	$\sum_{K=1}^{K=N}$	1
	N = 암호 필드 수		
키자체 확장성	없음	없음	있음
동일레벨 다른권한	미지원	미지원	지원

5. 결론

본 논문에서는 조직 구성원에 이동이 빈번한 다중 계층 환경에서의 효율적인 그룹키 생성을 위한 해쉬 체인과 랜덤난수의 논리합 연산을 이용한 그룹키 생성 기법을 제안하였으며 제안하는 알고리즘에 입각하여 시스템을 구현하였고, 성능 평가를 위해 기존의 슈퍼 그룹키 생성 기법, 다중 계층 그룹키 생성 기법과의 키 생성에 따른 주요 항목에 대한 실험 및 비교분석을 수행하였다.

먼저 입력값의 연관성을 기반으로 연속적인 키를 생성하는 해쉬 체인 알고리즘을 통해 그룹의 계층을 구분짓는 계층식별자를 생성함으로써 기존의 상위 권한 소유자로 하여금 모든 하위 권한에 키 정보를 소유해야 하는 키 생성 문제를 해결하였으며, 계층식별자와 랜덤하게 생성한 그룹식별자의 조합을 통해 동일 계층 다른 권한을 가지는 그룹에 대한 고유키를 생성함으로써 조직 구성원 이동에 따른 동적 변화에 유연하게 대처 할 수 있도록 설계하였다. 성능 부분에 있어서도 4.3절에서 제시한 기존의 그룹키 생성 기법과의 실험을 통한 비교분석 결과에서 확인 할 수 있듯이 제안하는 기법은 생성되는 전체키의 수, 사용자 소유키의 수, 정보 암복호화에 사용되는 키의 수, 키 자체에 확장성 등과 같은 여러 항목에서 기존 기법보다 우수함을 확인 할 수 있었다. 향후에는 조직 구성원에 이동이 빈번한 접근제어 환경에서의 효율적인 전자문서 관리를 지원하기 위한 그룹키 응용연구를 진행하고자 한다.

참고문헌

[1] 김현철, 이창수, 이경석, 전문석, “인증서를 이용한 보안성이 강화된 일회용 패스워드 검증 시스템의 설계” 한국통신학회 논문지, Vol.34 No. 4, pp.435-441, 2009.

[2] 이성운, 김현성, 유기영, “패스워드 기반의 효율적인 키 교환 프로토콜”, 한국정보과학회논문지, 정보통신 제31권 제4호, pp.347-352, 2004.

[3] 최은정, 김찬오, 송주석, “공개키 암호 기법을 이용한 패스워드 기반의 원거지 사용자 인증 프로토콜”, 한국정보과학회논문지, 정보통신 제30권 제1호, pp.75-80, 2003.

[4] 김희열, 이윤호, 박용수, 윤현수, “접근제어를 위한 반응적 방식의 그룹키 관리 기법”, 한국정보과학회논문지, 제34권제11호, pp.589-598, 2007.

[5] Donald E.Eastlake Ⅲ, and Kityy Niles, Secure XML : The New Synax for Signatures and Encryption, Pearson Education, 2003.

[6] John Linn, “Trust Models and Management in Public Key Infrastructures”, Technical Notes and Reports of RSA Laboratories, 2000.

[7] Jen-Chiun Lin, Feipei Lai, and Hung-Chang Lee, "Efficient Group Key Management Protocol with One-Way Key Derivation", Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary, 2005.

[8] Gildas Avoine, Philippe Oechslin, “A Scalable and Provably Secure Hash-Based RFID Protocol”, Third IEEE International Conference on Pervasive Computing, pp.110-114, 2005.

[9] 김대엽, 허미숙, 주학수, "효율적인 Batch 처리를 위한 그룹키 관리 기술", 한국정보보호학회논문지, v.18,no.5, pp.189-193, 2008.

[10] John Linn, “Trust Models and Management in Public Key Infrastructures”, Technical Notes and Reports of RSA Laboratories, 2000.

[11] Takeshi Imamura, Blair Dillaway, EdSimon, “XML Encryption Syntax and Processing”, W3C Recommendation, 2002.

[12] 김수희, “의료 환경에 적용 가능한 웹서비스 보안 및 키 관리 기술 연구”, 세종대학교대학원석사학위논문, 2007.

[13] 김진성, “멀티미디어 컨텐츠에 대한 등급별 보안”, 경상대학교대학원박사학위논문, 2007.

[14] 이성은, 장홍종, 박인재, 한선영, “다중 암호화 기법을 활용한 하이브리드 스마트카드 구현”, 정보보호학회논문지, 제13권 2호, pp.81-89, 2003.

[15] Yih-Chun Hu, Markus Jakobsson and Adrian Perrig, "Efficient Constructions for One-Way Hash Chains" LNCS 3531, pp. 423-441, 2005.

김 현 철(Hyun-Chul Kim) **[종신회원]**

- 2003년 2월 : 인제대학교 정보컴퓨터학부 (공학사)
- 2005년 2월 : 경원대학교 전자계산학과 (공학석사)
- 2009년 2월 : 숭실대학교 컴퓨터학과 (공학박사)
- 2009년 5월 ~ 현재 : 한국학기술정보연구원 정보화전략팀 선임연구원

<관심분야>
정보보안 정책 및 전략, 접근제어, DRM

이 영 구(Yoyng-Gu Lee) **[정회원]**

- 2003년 2월 : 숭실대학교 전자계산원 (공학사)
- 2006년 2월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2007년 3월 ~ 현재 : 숭실대학교 컴퓨터학과 박사과정

<관심분야>
인터넷 보안, PKI, DRM

김 정 재(Jung-Jae Kim) **[정회원]**

- 1995년 2월 : 영동대학교 컴퓨터공학과 (공학사)
- 1999년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학석사)
- 2005년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)

<관심분야>
멀티미디어 보안, DRM, RFID

이 광 형(Kwang-Hyoung Lee) **[종신회원]**

- 1998년 2월 : 광주대학교 컴퓨터공학과 (공학사)
- 2002년 2월 : 숭실대학교 일반대학원 컴퓨터공학과 (공학석사)
- 2005년 2월 : 숭실대학교 일반대학원 컴퓨터공학과(공학박사)
- 2005년 3월 ~ 현재 : 서일대학교수

<관심분야>
USN, RFID, 영상보안, 홈네트워크