情報保護學會論文誌
第15券 第1號, 2005. 2

그룹 환경을 위한 안전한 인덱스 검색 스킴*

박 현 아,† 변 진 욱, 이 현 숙, 이 동 훈‡
고려대학교 정보보호대학원

Secure Index Searching Schemes for Groups

Hyun-A Park,† Jin-Uk Byun, Hyun-Suk Lee, Dong-Hun Lee‡
CIST(center for information security technology) in Korea University

요 약

안전한 인덱스 검색 프로토콜은 검색어에 대한 트랩도어를 사용하여 암호화된 문서의 인덱스를 검색하는 것이다. 그것은 비 신뢰적인 서버 관리자에게 검색어를 드러내지 않고 검색하여 그 결과 외엔 문서에 관한 어떤 정보도 알 수 없게 하는 것이다.[1,2,3] 무수히 많은 안전한 검색 프로토콜들이 제안되어져 왔으나, 이것들은 단지 개인 사용자와 서버와의 검색 과정만을 고려하였다. 그러나 실제로 계층적인 부서가 많은 기업이나 그룹과 같은 조직체들은 그룹의 공유 문서에 대한 검색 환경이 더 많이 요구되어진다. 따라서 이 논문에서는 그룹 키가 새로이 갱신되었을 때 기존 문서의 재 암호화 없이도 검색 가능한 안전한 인덱스 검색 프로토콜을 제안한다.

ABSTRACT

A secure index search protocol let us search the index of encrypted documents using the trapdoor for a keyword. It enables an untrusted server to learn nothing more than the search result about the documents without revealing the keyword. A lot of secure search protocols have been suggested but they only considered the search between a single-user and a server. In real organizations such as government offices or enterprises where exist many hierarchical departments, the search system for groups is arisen more often. In this paper, we construct secure index search protocols for hierarchical group settings without re-encryption of the old encrypted documents when group keys are re-keyed newly.

Keywords : *Index search, privacy, trapdoor, keyword, encrypted documents*

Ⅰ. 서 론

기업이나 공공기관 등의 모든 부서들은 문서를 서버에 저장해 놓고 수시로 열람하게 되는데, 이 때 다른 부서나 외부인이 보아서는 안될 회사의 기밀사항을 저장해야 하는 상황이 많이 발생하고 이런 문서를 저장하기 위해서는 암호화가 필요하다. 그리고 이렇게 암호화된 문서는 오직 그 부서만의 허락된 구성원에 한하여 열람이 가능해야 하며 서버 관리자들은 그러한 기밀문서의 열람이 불가능해야 한다.

안전한 인덱스 검색 프로토콜은 검색어에 대한 트랩도어를 사용하여 암호화된 문서의 인덱스를 검색하는 것이다. 트랩도어는 사용자의 비밀 키가 있어야 만이 생성될 수 있기 때문에, 그것은 비 신뢰적인 서버 관리자에게 검색어를 드러내지 않고 검색하여 그 결과 외엔 문서에 관한 어떤 정보도 알 수 없게 하는 것이다.[1-3]

접수일 : 2004년 12월 6일 ; 채택일 : 2005년 2월 2일
* 이 논문은 2003년도 한국학술진흥재단의 지원(KRF-2003-041-D00500)에 의하여 연구 되었습니다.
† 주저자, kokokzi@cist.korea.ac.kr
‡ 교신저자, donghlee@korea.ac.kr

현재까지 암호화된 문서를 검색하는 스킴은 많이 제안되어져 왔으나, 아직까지 사용자가 그룹인 환경에 대한 연구는 없었다.[1-8] 하지만 현실적인 측면에서는 개인의 사적인 비밀문서 보다는 이런 그룹의 공유문서를 암호화하여 서버에 저장해 두고 외부나 내부에서 검색하는 환경이 더 필요하다. 관공서나 기업과 같이 많은 하위 그룹들을 갖는 그룹들은 그들 그룹의 공유문서나 그 멤버들의 사적인 비밀문서를 암호화하여 서버에 저장한다. 따라서 그런 계층적인 그룹의 멤버는 다중 사용자로서, 전체 그룹의 멤버일 뿐만 아니라 그 하위 그룹의 멤버도 되며, 동시에 개인의 사적인 목적을 위해 검색 시스템을 사용하는 개인 사용자도 되는 것이다.

그런데 사용자가 그룹이라는 것은 문서 검색 시스템에 사용되는 키가 그룹 키라는 것을 의미한다. 하지만 이 그룹 키는 안전성을 위해 구성원 변화가 일어나면 그때마다 키가 갱신되어야 한다. 그렇다면 이 키 갱신이 일어날 때마다 그전에 암호화되었던 모든 문서들을 다시 재 암호화해야 하는 것일까? 그렇다면 이것은 실로 엄청난 계산량을 요할 것이며, 이것이 바로 암호화된 문서 검색 시스템의 사용자를 그룹으로 확장했을 때의 가장 큰 문제점이라 할 수 있겠다. 따라서 그룹 키가 갱신될 때마다 그전 문서를 다시 재암호화 하는 과정 없이 이전 문서까지 검색 가능하게 하는 스킴을 제안하고 그러한 환경에서 비 신뢰적인 서버에 대한 사용자의 프라이버시를 보장하고자 한다.

1.1 관련 연구 및 공헌도

아주 최근에 Goh.가 제안한 안전한 인덱스 검색이 있다. 그는 안전한 인덱스 검색(secure index search)을 정의하고, adaptive chosen keyword attack(ind-cka)에 대해 semantic security를 제공하는 보안(security) 모델을 제시했다. 뿐만 아니라, 슈도 랜덤 함수와 블룸 필터를 사용하여 'Z- IDX'라는 ind-cka에 안전한 인덱스 검색을 설계했다.[2]

또 다른 연구로, Boneh et al.은 공개키 시스템을 사용하는 키워드 검색 스킴을 제안했다. 그들은 키워드를 검색하는 공개키 암호화에 대한 개념을 정의하고 두 가지 스킴을 설계했다.[7] 그리고 Chang과 Mitzenmacher는 pre-built dictionary를 이용한 두 가지 인덱스 스킴을 제안했다.[3]

그러나 위의 프로토콜 모두는 양자간의 1:1 검색 환경(서버와 사용자)에 관한 것이었다. 즉 암호화되어 저장된 문서에 대해 복호화 키를 가지고 있는 사람은 오로지 한 명 뿐이었다. 따라서 이 논문에서는 검색 환경을 1:n의 그룹으로 확장하는 새로운 스킴을 제안하고자 한다. 이 그룹은 많은 하위 그룹을 가지는 계층적인 그룹이며, 이런 계층적인 그룹의 세션키가 갱신되어도 기존의 암호화된 문서를 재암호화 하지 않고 검색 가능한 스킴이다. 다음은 제안한 프로토콜들의 장점을 정리한 것이다.

- □ **그룹 환경에서의 인덱스 검색.** 1:n(서버: 사용자) 즉, 사용자를 그룹으로 확장하여 그룹멤버가 공유문서를 안전하게 검색하고 다운 받을 수 있게 한다.
- □ **접근 권한.** 전체그룹, 소그룹, 개인이 열람할 수 있는 문서의 구분을 두고, 각자의 문서 검색용 비밀키를 사용하여 그룹별 접근 권한을 부여한다.
- □ **효율성.** 그룹키 갱신이 있어도 이미 저장된 문서의 재암호화 없이 그전 문서까지 검색 가능하게 한다.
- □ **프라이버시 보장.** 비신뢰적인 서버에 대해 사용자의 프라이버시를 보장한다.
- □ **전방향 안전성.** 탈퇴한 멤버에 대해 전방향 안전성을 보장한다.

1.2 논문의 구성

먼저 2장에서는 제안한 프로토콜의 모태가 되는 Z-IDX 인덱스 검색 스킴에 대해 간단히 알아보고, 3장에서는 사용자를 그룹으로 확장한 새로 제안한 스킴 2가지를 소개한다. 그리고 4장에서는 새로운 스킴과 프라이버시에 대해 살펴보고 5장에서는 제안한 프로토콜의 몇 가지 특성에 대해 논하고 마지막으로 6장에서 결론을 맺는다.

II. Z-IDX

Z-IDX는 슈도 랜덤 함수와 블룸 필터를 가지고 인덱스를 생성해서 암호화된 문서를 검색하는 IND-CKA(index-chosen keyword attack)에 안전한 효율적인 스킴이다.

2.1 Background-블룸필터(Bloom Filters)

해당 집합의 멤버쉽 체크를 위한 공간 효율적인 데이터 구조를 구축하는 것이 주 목적이다. n개의 원소로 구성된 집합을 $S=\{s_1,...,s_n\}$라 하자. 블룸 필터는 m 비트들의 배열로 구성되며, r개의 독립적인 해쉬 함수 $h_1,...,h_r$를 가지고(여기서 $h_i:\{0,1\}^* \rightarrow [1,m]$ for $i \in [1,r]$), 초기엔 모두 0으로 셋팅 되어진다. 집합 S의 각 원소 $s \in S$ 에 대하여 $h_1(s),..h_r(s)$의 위치에 있는 배열의 비트들은 1로 셋팅 되어진다.

만일 어떤 요소 a가 집합 S에 속하는가를 알려면, $h_1(a),..h_r(a)$의 위치에 있는 비트들을 체크한다. 체크된 비트들이 모두 1이면 그것은 집합의 요소라고 여겨질 수 있다.

이해를 돕기 위해 간단한 예를 들어본다.

m = 5 (비트 수), r= 2 (해쉬 함수의 개수):
$h_1(x)=x \mod 5$
$h_2(x)=(2x+3) \mod 5$
Bloom Filter $S[1::5]$를 초기화하고, 그리고 나서 9와 11을 집어 넣어라:

	$h_1(x)$	$h_2(x)$	*Bloom Filter S*				
초기화			0	0	0	0	0
9를 넣었을 때	4	1	0	1	0	0	1
11를 넣었을 때	1	0	1	1	0	0	1

그런 다음 어떤 멤버쉽 질의를 시도한다.[2,9,10]

	$h_1(x)$	$h_2(x)$	결과
15를 질의	0	3	S 안에 없다.
11 질의	1	0	S 안에 있다.

2.2 알고리즘

Z-IDX는 다음과 같은 중요 알고리즘 4가지를 가진다.

1) *Keygen*(s)

security parameter로 s가 주어지면 슈도 랜덤 함수 $f:\{0,1\}^n * \{0,1\}^s \rightarrow \{0,1\}^s$를 선택하고, 마스터 키 $K_{priv}=(k_1,...,k_r) \in_R \{0,1\}^{sr}$를 생성한다.

2) *Trapdoor*(K_{priv}, w)

마스터 키 $K_{priv}=(k_1,...,k_r) \in \{0,1\}^{sr}$과 단어 w가 주어진다면, 단어 w에 대한 트랩도어로 $T_w=(f(w,k_1),...,f(w,k_r)) \in 0,1^{sr}$를 출력한다. 즉, 마스터 키로 단어를 암호화 하는 것이다.

3) *BuildIndex*(D, K_{priv})

입력으로 문서 $D=$(고유한 식별자(이름) $D_{id}=\{0,1\}^n$와 단어 목록 $(w_0,...,w_t) \in \{0,1\}^{nt}$)와 마스터 키 $K_{priv}=(k_1,...,k_r) \in \{0,1\}^{sr}$이 주어진다. 단어 w에 대해 다음을 계산한다.

□ 트랩도어 생성:
$(x_1=f(w,k_1),...,x_r=f(w,k_r)) \in 0,1^{sr}$
: 검색어를 마스터 키로 암호화한다.
□ Codeword 생성:
$(y_1,y_2,...,y_r)=$
$(f(D_i,x_1),f(D_i,x_2),...,f(D_i,x_r))$
즉, 트랩도어를 그것이 속한 문서의 식별자(identifier)와 함께 암호화한다.
□ D_i의 BF에 codeword $(y_1,y_2,...,y_r)$를 집어 넣는다.
□ D_i의 단어목록에 있는 모든 단어들에 대해 위의 세 과정을 반복 수행하여 블룸 필터 BF를 구성한다.
□ $I_{D_i}=(D_i,BF)$를 D_i에 대한 인덱스로 출력한다.

4) *SearchIndex*(T_w, ID)

입력으로 문서 D_{id}에 대한 트랩도어 $T_w=(x_1,...,x_r)$와 인덱스 $I_{D_i}=(D_i,BF)$가 주어지면, D_{id}에 있는 w에 대한 codeword를 계산한다. w에 대한 codeword $(y_1,y_2,...,y_r)$로 BF 테스트를 실시하여 r개의 위치 모두가 1을 포함하면 1을 출력하고 그렇지 않으면 0을 출력한다.[2]

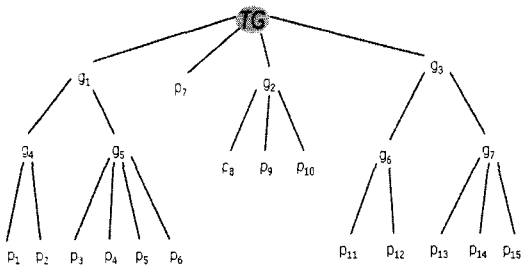


그림 1. 그룹의 조직 구성도

Ⅲ. 그룹을 위한 안전한 인덱스 검색의 설계

이 장에서는 계층적인 그룹에서의 안전한 인덱스 검색 스킴을 설계한다. 그림 1은 계층적 그룹의 예를 보여준다. 기업 또는 공공기관 및 단체 등 어떠한 조직체를 대상으로 하여, 전체 그룹(TG) 밑에 하위 그룹($g_1,g_2,...,g_7$)을 두고, 그 그룹들에 속하는 멤버들($p_1,p_2,...,p_{15}$)로 구성된 하나의 거대한 집합체를 환경으로 설정한다.

본 스킴에선 그룹 관리자(GC, group controller)를 둔다. 이 그룹 관리자는 각 그룹들의 커뮤니케이션을 위한 그룹 세션키와 암호화된 문서 검색을 위한 검색용 키 모두를 관리한다. 그리고 문서 검색용 키들은 문서 암호화 키와 인덱스 생성용 키로 이루어져 있다. 이처럼 검색용 키들이 독립적으로 구분되어 사용되어지기 때문에 문서의 압축 및 암호화를 용이하게 할 수 있어서 서버의 데이터베이스를 효율적으로 사용할 수 있다.[2] 그 뿐만 아니라, 이 검색용 키와 커뮤니케이션을 위한 그룹의 세션키나 개인의 비밀키 역시 안전성의 측면에서 구분되어 사용되어진다.

본 논문에서는 2가지 스킴을 제안한다. 하나는 SIS-D(secure index search in a direct way)로 서버와 사용자간에 직접적인 검색이 이뤄지는 경우이고, 다른 하나는 SIS-C(secure index search via a GC)로 GC를 통해서만이 문서 검색이 가능한 경우이다. 그리고 이들 각각은 다시 그룹 멤버로서 비밀 공유 문서를 검색하는 경우와 개인 사용자로서 개인의 비밀 문서를 검색하는 경우로 나누어져 총 4개의 스킴이 소개 된다: SIS-GD (secure index search for a group user in a direct way), SIS-PD(secure index search for a private user in a direct way), SIS-GC(secure index search for a group user passing through a group controller), SIS-PC(secure index search for a private user passing through a group controller).

Z-IDX의 네 가지 알고리즘을 근간으로 하여 제안하는 프로토콜은 총 6단계로 구성된다: 키 생성 단계(Key Generation Stage), 트랩도어 생성단계(Trapdoor Generation Stage), 인덱스 생성 단계(Index Building Stage), 데이터 생성 단계(Data Building Stage), 업로딩 단계(Uploading Stage), 인덱스 검색 단계(Index Searching Stage).

3.1 표기법 (notation)

- □ k_itg: 세션 i의 전체 그룹의 세션키.
- □ K_iD_{TG}: 세션 i의 전체 그룹의 문서 암호화 키.
- □ K_iI_{TG}: 세션 i의 전체 그룹의 인덱스 생성용 키. BF를 구성하는 r개의 해쉬 함수에 대한 입력값을 생성하기 위해 인덱스 생성용 키는 Z-IDX의 마스터 키처럼 r개의 키들 $K_iI_{TG}=(tgk^i_1,..,tgk^i_r)$로 구성되어 있다.
- □ k_ig_j: 세션 i에서 하위 그룹 g_j의 그룹 세션키
- □ K_iD_{Gj}: 세션 i에서 하위 그룹 g_j의 문서 암호화 키
- □ $K_iI_{Gj}=(g_jk^i_1,g_jk^i_2,....g_jk^i_r)$: 세션 i에서 하위 그룹 g_j의 인덱스 생성용 키.
- □ kp_i : 개인 사용자 p_i의 비밀 개인키
- □ KDp_i : 개인 사용자 p_i의 문서 암호화 키
- □ $KIp_1=(p_ik_1,p_ik_2,...,p_ik_r)$: 개인 사용자 p_i의 인덱스 생성용 키
- □ $T_{w_i}=(x^q_1,x^q_2,...x^q_r)$: w_j에 대한 q번째 세션의 트랩도어
- □ $(y^q_1,..y^q_r)=(f(i,x^q_1),..,f(i,x^q_r))$: w_j에 대한 q번째 세션의 codeword
- □ D_i: 식별자(identifier)가 i인 문서
- □ I_{D_i}: 식별자(identifier)가 i인 문서 D_i의 인덱스

3.2 SIS-D

SIS-GD에서 GC는 오로지 키 관리만을 주관하

표 1. 각 그룹별 세션마다 갱신되는 키 대응표

그룹	그룹의 세션키	문서 암호용 키	인덱스 생성용 키
TG	$k_1tg \rightarrow k_2tg$ $\rightarrow k_3tg....$	$K_1D_{TG} \rightarrow K_2D_{TG}$ $\rightarrow K_3D_{TG},...$	$K_1I_{TG} \rightarrow K_2I_{TG}$ $\rightarrow K_3I_{TG},...$
g_1	$k_1g_1 \rightarrow k_2g_1$ $\rightarrow k_3g_1,...$	$K_1D_{G1} \rightarrow K_2D_{G1}$ $\rightarrow K_3D_{G1},...$	$K_1I_{G1} \rightarrow K_2I_{G1}$ $\rightarrow K_3I_{G1},...$
g_2	$k_1g_2 \rightarrow k_2g_2$ $\rightarrow k_3g_2,...$	$K_1D_{G2} \rightarrow K_2D_{G2}$ $\rightarrow K_3D_{G2},...$	$K_1I_{G2} \rightarrow K_2I_{G2}$ $\rightarrow K_3I_{G2},...$
:	:	:	:

며 문서 검색 시스템용 키들은 일 방향 해쉬 함수 키 체인으로 생성된다. 그리고 이해를 돕기 위해 모든 단계들은 그룹 g_1의 멤버 p_1이 두 번째 세션으로 변동되었을 때를 가정한다.

3.2.1 SIS-GD

1) 키 생성 단계(Key Generation Stage)

먼저, GC는 각 그룹에 대한 그룹 세션키와 문서 암호용 키, 인덱스 생성용 키를 만들어서 표 1과 같은 키 대응표를 가지고 있는다.

표 1은 하위 그룹 g_1의 경우, 세션이 세션 1에서 세션2로 변경되면 그룹의 세션키는 그룹키 프로토콜에 의해 $k_1g_1 \rightarrow k_2g_1$으로 변하고, 문서 검색 시스템에 이용되는 문서 암호용 키와 인덱스 생성용 키도 $K_1D_{G1} \rightarrow K_2D_{G1}$과 $K_1I_{G1} \rightarrow K_2I_{G1}$으로 각각 변함을 보여준다.

그러나 문서 검색용 키는 다음과 같은 일방향 해쉬 함수 키체인으로 형성된다. 우선 security parameter s가 주어지고 마지막 키의 값 K_qD_{G1}과 K_qI_{G1}를 랜덤하게 선택한다(만약 키 체인의 길이를 q라고 한다면). 그것을 랜덤하게 선택한 일방향 해쉬 함수에 계속 적용하여 다른 키들을 계산해 나간다. : $K_iI_{G1} = h^r(K_{i+1}I_{G1})$, $K_iD_{G1} = h(K_{i+1}D_{G1})$, 여기서 $i \in [1, q-1]$이고, h^r은 일방향 함수 h가 r개의 각 요소에 독립적으로 적용되어짐을 의미한다. 이러한 방법으로 우리는 모든 검색키들을 역으로 생성해 나갈 수 있다.

$$K_iI_{G1} = \begin{cases} K_qI_{G1} = (g_1k_1^q, g_1k_2^q, ..., g_1k_r^q) \in_R \{0,1\}^{sr} \\ K_{q-1}I_{G1} = h^r(K_qI_{G1}) = (g_1k_1^{q-1}, g_1k_2^{q-1}, ..., g_1k_r^{q-1}) \\ K_{q-2}I_{G1} = h^r(K_{q-1}I_{G1}) = (g_1k_1^{q-2}, g_1k_2^{q-2}, ..., g_1k_r^{q-2}) \\ ..... \\ K_1I_{G1} = h^r(K_2I_{G1}) = (g_1k_1^1, g_1k_2^1, ..., g_1k_r^1) \end{cases}$$

$$K_iD_{G1} = \begin{cases} K_qD_{G1} \in_R \{0,1\}^s \\ K_{q-1}D_{G1} = h(K_qD_{G1}), \\ K_{q-2}D_{G1} = h(K_{q-1}D_{G1}) \\ ..... \\ K_1D_{G1} = h(K_2D_{G1}) \end{cases}$$

여기서 일방향 함수 h의 역할은 굉장히 중요하다. h의 일방향성에 의해 탈퇴자는 그룹을 떠난 후의 새로운 키들은 알 수 없으나 새로 가입한 멤버는 현재 키만 알고 있으면 해쉬 함수를 계속 적용시켜 그 이전의 모든 키들을 쉽게 계산해 낼 수 있다. 따라서 탈퇴자에 대한 전방향 안전성(forward secrecy)이 보장됨과 동시에 이전 문서들을 재 암호화 하지 않고서도 모든 문서를 검색 가능하게 하는 것이다.

예로, 세션이 세션 1에서 세션 2로 변경되었을 때 하위 그룹 g_1의 멤버 p_1은 갱신된 그룹키 k_2g_1을 g_1의 그룹키 프로토콜의 방법으로 전송받으며, GC는 세션2에서 사용될 이미 만들어져 있는 K_2D_{G1}과 K_2I_{G1}을 세션 2의 그룹키 k_2g_1으로 암호화하여 그룹 g_1의 각 구성원에게 전송한다.

2) 트랩도어 생성단계(Trapdoor Generation Stage)

사용자 p_1은 슈도 랜덤 함수 $f: \{0,1\}^n * \{0,1\}^s \rightarrow \{0,1\}^s$를 랜덤하게 선택한다. 문서 D_i는 그 문서에 대한 고유한 식별자(identifier) $i = \{0,1\}^n$와 단어 목록 $(w_0, ..., w_t) \in \{0,1\}^{nt}$으로 이루어져 있다. 사용자 p_1은 슈도 랜덤 함수 f의 입력값으로 인덱스 생성용 키 $K_2I_{G1} = (g_1k_1^2, g_1k_2^2, ...g_1k_r^2)$와 단어 w_j를 가지고 w_j에 대한 트랩도어 $T_{w_j} = \left(f(w_1, g_1k_1^2), ...f(w_1, g_1k_r^2)\right) = (x_1, .., x_r)$를 계산한다. 즉, 인덱스 생성용 키를 가지고 검색어들(단어 목록)을 암호화 하는 것이다.

3) 인덱스 생성 단계(Index Building Stage)

사용자 p_1은 문서 D_i와 2)에서 생성한 트랩도어 $T_{w_j} = \left(f(w_j, g_1k_1^2), f(w_j, g_1k_2^2), ...f(w_j, g_1k_r^2)\right) = (x_1, x_2, ..., x_r)$를 가지고, 단어 w_j에 대해 다음 과정을 수행한다.

① Codeword: 사용자는 문서 D_i에 있는 w_j에 대한 트랩도어 T_{w_j}를 가지고 codeword를 계산한다. $(y_1,y_2,...,y_r)=(f(i,x_1),f(i,x_2),...,f(i,x_r))$즉, 사용자는 검색어 w_j가 포함된 문서 D_i의 식별자 i와 함께 트랩도어 T_{w_j}를 슈도랜덤 함수 f의 입력값으로 하여 암호화하는 것이다.

② 문서 D_i에 대한 BF 구성: D_i의 BF에 codeword $(y_1,y_2,...,y_r)$를 집어 넣어 테스트한다. 모든 w_j에 대해 이 과정을 반복하여 문서 D_i에 대한 BF를 구성한다.

③ 인덱스 생성: $I_{D_i}=(i \mid BF)$를 D_i에 대한 인덱스로 출력한다. 즉, 이 BF가 식별자가 i인 문서 D_i의 인덱스가 되는 것이다.

4) 데이터 생성 단계(Data Building Stage) 및 업로딩 단계(Uploading Stage)

사용자 p_1은 문서 $D=\{D_1,D_2,..,D_i,...,D_n\}$를 세션 2의 문서 암호용 키 K_2D_{G1}로 암호화하여 식별자 i와 연결한 후 생성된 $\{I_D\}$와 함께 서버에게 보낸다. 각 인덱스와 해당 문서를 매칭 시키지는 않는다.

$$C=[\{E_{K_2D_{G1}}(D_i) \mid i\},\{I_{D_i}=(i \mid BF)\}]$$

서버는 암호문 C를 받아 저장한다.

5) 인덱스 검색 단계(Index Searching Stage)

사용자 p_1은 세션 2의 인덱스 생성용 키 K_2I_{G1}와 세션 1의 인덱스 생성용 키 $K_1I_{G1}=h^r(K_2I_{G1})$으로 검색어 w에 대한 트랩도어 T_{w_2}, T_{w_1}을 계산한다. 이것은 검색어 w가 세션 1에서 만든 문서에 포함될 수도 있고 세션 2에 만든 문서에 있을 수도 있기 때문이다. 즉, 누적된 세션의 횟수만큼 트랩도어를 생성한다. 이것은 현재 세션의 키만 저장하고 있으면 일방향 해쉬 함수 키체인을 계속 적용하여 그 이전 키들을 모두 복구해낼 수 있으므로 가능하게 된다. 이렇게 만들어진 트랩도어를 한번에 서버에게 전송한다. 서버는 받은 $T_{w_2}=(x_1^2,x_2^2,...x_r^2)$, $T_{w_1}=(x_1^1,x_2^1,...x_r^1)$와 저장하고 있던 인덱스들 $I_{D_i}=(i \mid BF)$의 각 식별자 i를 가지고, D_i 안에 들어있는 w에 대한 codeword들을 계산한다.

$$(y_1^2,y_2^2,...y_r^2)=(f(i,x_1^2),f(i,x_2^2),...,f(i,x_r^2))$$
$$(y_1^1,y_2^1,...y_r^1)=(f(i,x_1^1),f(i,x_2^1),...,f(i,x_r^1))$$

그리고 나서 서버는 BF 테스트를 수행한다. 만약 w에 대한 codeword $(y_1^2,y_2^2,...y_r^2)c$와 $((y_1^1,y_2^1,...y_r^1)$에 대해 BF 테스트를 통과하는 i가 있다면, 그 i에 해당하는 문서 $\{E_{K_2D_{G1}}(D_i)\}$와 $\{E_{K_1D_{G1}}(D_i)\}$를 그룹 멤버 p_1에게 보낸다.

사용자 p_1은 전송받은 $\{E_{K_2D_{G1}}(D_i)\}$와 $\{E_{K_1D_{G1}}(D_i)\}$를 복호화 하여 필요한 문서들을 얻게 된다.

$$\{D_{K_2D_{G1}}(E_{K_2D_{G1}}(D_i))\},\ \{D_{K_1D_{G1}}(E_{K_1D_{G1}}(D_i))\}$$
; 여기서 $K_1D_{G1}=h(K_2D_{G1})$

만일 q번째 세션이라면, 사용자는 q개의 트랩도어를 생성해야 하고, 사용자는 다운받은 문서들에 대해 평문이 나올 때까지 키체인을 계속 적용해서 복호화해야 한다. SIS-GD의 전체적인 검색 과정은 그림 2와 같다.

3.2.2 SIS-PD

SIS-PD에서 사용자 p_1은 GC도 알고 있는 개인

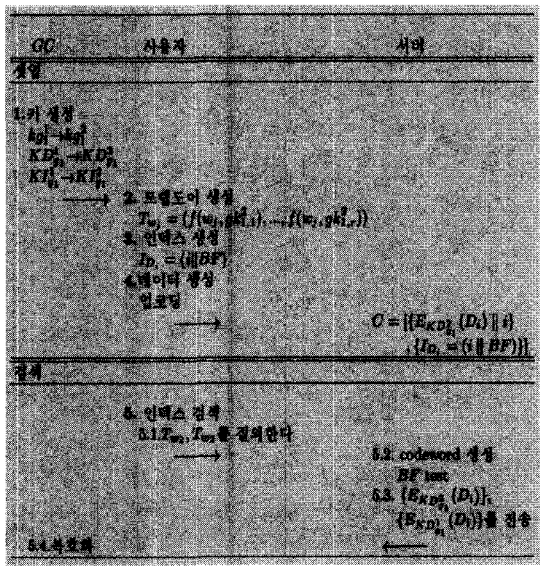


그림 2. SIS-GD

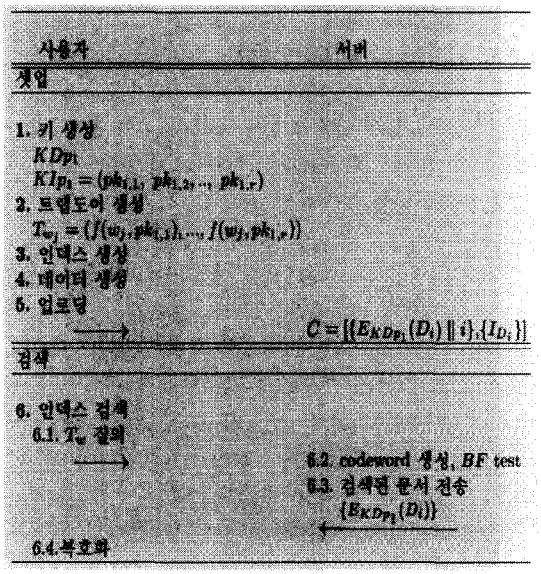


그림 3. SIS-PD

키 kp_1과 GC가 알지 못하는 문서 암호화 키 KDp_1과 인덱스 생성용 키 KIp_1을 가진다. 왜냐하면 사적인 목적으로 검색 시스템을 사용하는 개인 사용자는 GC마저도 검색 과정을 아는 것을 원하지 않기 때문이다. 검색 단계는 그림 3과 같다.

3.3 SIS-C

2.2의 SIS-D는 서버와 사용자간의 직접적인 검색 환경으로 세션 횟수가 누적됨에 따라 사용자의 계산량이 많아진다. 따라서 그런 세션 횟수와 상관없이 사용자의 계산량을 줄일 수 있는 방안으로 GC를 키 관리자의 역할만이 아닌 문서 검색시 반드시 거쳐야 하는 경유지로 설정하고, TTP 및 믹스넷의 기능을 부여하여 프라이버시 보장을 강화한다.

3.3.1 SIS-GC

GC는 각 그룹에 대해 표 2와 같은 키 대응표를 유지한다.

세션이 세션 1에서 세션 2로 변경되어 소그룹 g_1의 그룹키가 $k_1g_1 \rightarrow k_2g_1$이 되더라도 GC가 가진 그룹 g_1에 해당하는 문서 암호용 키 KD_{G1}과 인덱스 생성용 키 KI_{G1}은 변함없다. 갱신된 그룹 세션키는 그룹키 프로토콜에 의해 각 사용자에게 전달되고, 그 세션키로 사용자는 데이터를 암호화하여 GC에게

표 2. 각 그룹의 세션키에 대해 GC가 가지는 문서 검색 시스템 키

그룹	그룹의 세션키	GC의 문서 암호용 키	GC의 인덱스 생성용 키
TG	$k_1tg \rightarrow k_2tg \rightarrow k_3tg....$	KD_{TG}	KI_{TG}
g_1	$k_1g_1 \rightarrow k_2g_1 \rightarrow k_3g_1, ...$	KD_{G1}	KI_{G1}
g_2	$k_1g_2 \rightarrow k_2g_2 \rightarrow k_3g_2, ...$	KD_{G2}	KI_{G2}
:	:	:	:

전송하면 GC는 그것을 복호화한 후 다시 그 그룹에 해당하는 문서 검색용 키로 암호화하여 서버에게 질의한다. 즉, GC가 사용자를 대신하여 문서 검색 과정을 수행하는 것이다.

세션 2에서 p_1이 자신이 속한 소그룹 g_1의 공유 문서를 저장하고 검색하려고 할 때 그림 4와 같은 단계를 거친다.

3.3.2 SIS-PC

SIS-GC에서 서버는 사용자가 어떤 검색어로 어떤 문서를 검색하는지 전혀 알 수 없지만 사용자의 역할을 대신하는 GC는 이 모든 것을 알게 된다. 물

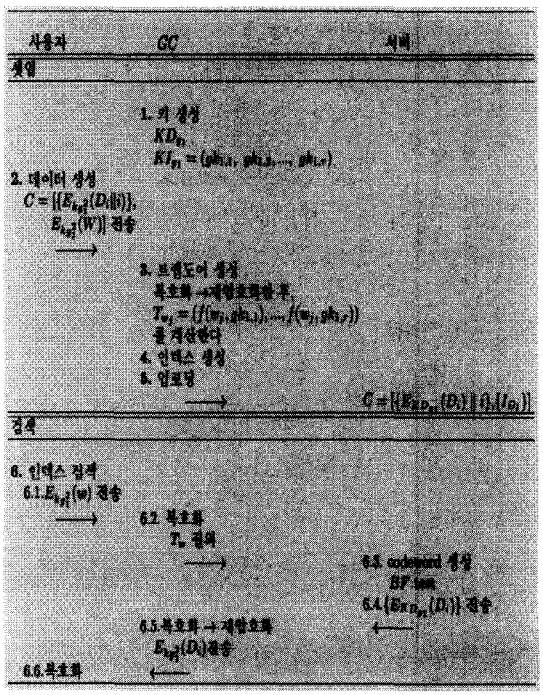


그림 4. SIS-GC

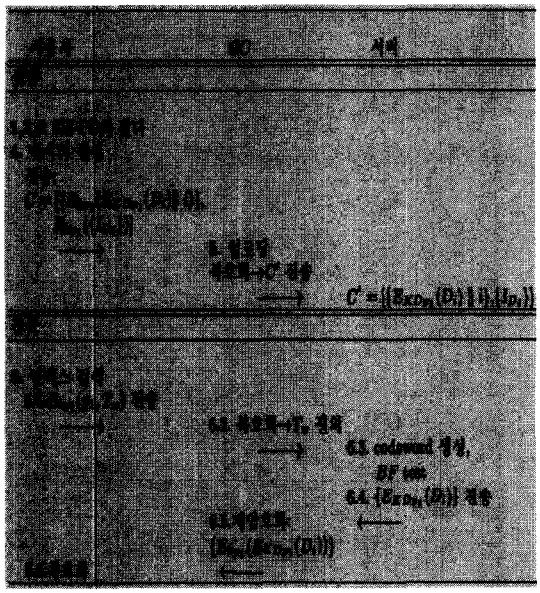


그림 5. SIS-PC

론 GC가 TTP의 역할을 하므로 문제시 될 것은 없지만, 개인의 비밀문서를 저장/검색하는 개인 사용자는 GC마저도 알기를 원하지 않는다. 따라서 개인키 kp_1 외의 GC가 알지 못하는 별도의 문서 검색 시스템용 키를 두어 모든 과정을 개인 본인이 수행하고 GC는 단지 거쳐가는 경유지로만 이용한다. 만약 이때 p_1이 GC를 경유하지 않고 서버에 직접 질의한다면, 서버는 p_1이 다운받는 문서가 개인의 비밀문서임을 알 수 있다. 이는 접근 방법 프라이버시(access pattern privacy)를 만족시키지 않으므로, SIS-PC의 개인 사용자는 반드시 GC를 경유하도록 한다.[2] 검색 과정은 그림 5에 나타나 있다.

Ⅳ. 프라이버시

그룹 환경에서의 안전한 검색 스킴으로서 공유문서 및 개인의 비밀 문서 모두를 포함하는 제안 프로토콜의 프라이버시는 다음과 같은 사항을 만족해야 한다.

- □ 비신뢰적인 서버는, 누가 어떤 검색어로 어떠한 내용의 문서를 다운 받았는지 알 수 없어야 한다.
- □ 검색어와 리턴하는 문서와의 관계에서 서버는 아무것도 알아낼 수 없다.
- □ 리턴하는 문서가 비밀문서인지 공유문서인지 서버는 알수 없다.

SIS-C에서 GC는 그룹의 문서 검색에 관한 모든 것을 알게 되나, GC를 제외하면 프라이버시는 거의 완벽하게 구현될 수 있다.

- □ 검색어를 트랩도어로 전송한다.: 어떤 검색어를 질의했는지 알수 없다.
- □ GC 즉, 믹스넷을 통과: SIS-C에서 믹스넷은 다음과 같은 세 단계를 거친다.

  ① 암/복호화 단계(encryption/decryption phase): 입력 데이터가 암호화 또는 복호화 되며, 주소지 정보는 제거되거나 암호화 되어진다.
  ② 배치 생성 단계(batch generation phase): 입력 데이터는 배치 단위로 처리된다.
  ③ 혼합 단계(mixing phase): 입력 데이터는 슈도 랜덤 치환 함수에 의해 순서가 바뀌어진다.

  입/출력되는 암호문은 위와 같은 믹스넷을 지나면서 형태와 순서가 변하게 된다. 그뿐 아니라 주소지 정보도 제거되거나 암호화되어져 서버에 전달되므로 서버는 누가 어떤 검색어를 질의하여 어떤 문서를 다운로드 받는지 전혀 알수 없다.[6,11]

- □ 암호화된 문서: 서버는 GC에게 검색 결과인 암호화된 문서를 전달하므로 서버는 이 또한 누구에게 어떠한 내용의 문서를 전송했는지 전혀 알 수 없다.

따라서 GC를 통과하는 SIS-C는 송신자 및 수신자의 익명성을 보장할 수 있고 추적 불가능하다. 뿐만 아니라 어떤 검색어에 대해 어떤 내용의 문서를 다운받았는지 알 수 없으며, 그것이 공유문서인지, 비밀문서인지도 알 수 없다. 단지 어떤 사람이 문서 검색 시스템을 이용해 몇 개의 문서를 다운받았는지만을 알 뿐이다.

SIS-D는 GC를 통과하지 않는 서버와 사용자간의 직접적인 검색 환경이지만, 이 역시도 트랩도어를 이용하여 암호화된 문서를 다운 받으므로, 서버는 그 사용자가 어떤 검색어에 대해 어떠한 문서를

내려 받았는지 알 수 없다. 따라서 SIS-D도 문서 검색 환경에서의 프라이버시는 보장된다고 말할 수 있다.

V. 제안 프로토콜의 특징

5.1 전방향 안전성(forward secrecy)

사용자를 그룹으로 확장한 본 스킴의 안전성 문제는 역시 구성원 변화에 의한 세션 변동에 기인한다. 보통 그룹의 키 갱신시 요구되는 조건은 전방 안전성과 후방 안전성(backward secrecy)이다.

탈퇴한 사람에 있어 전방 안전성은 두 스킴 모두 만족한다. 주어진 그룹키 프로토콜에 의해 SIS-C에선 갱신된 세션키를 탈퇴자는 알 수 없어 질의시 GC에게 인증받지 못하므로 전방 안전성이 보장되고, SIS-D에선 GC에게 갱신된 키를 전송받지 못한 탈퇴자는 일방향 해쉬 함수 키체인의 일방향성에 의해 갱신된 세션키를 알 수 없어 탈퇴 후의 세션에 대한 트랩도어는 생성할 수 없으므로 이후의 문서 검색은 불가능하다.

반면 새로 가입한 사람에게 후방 안정성을 제공하지는 않는다. 그러나 우리가 설정한 기업이나 기타 단체에서의 요구 사항은 후방 안전성이 없어서 이전 문서 검색을 가능하게 해야 하는 것이다. 예를 들어 신입사원이나 새로이 발령받은 사람일 경우 그 이전 문서의 검색이 모두 가능하여 그를 참고로 연속적인 업무를 수행해야 하기 때문이다.

5.2 SIS-D의 키체인과 SIS-C의 GC

제안한 프로토콜은 그룹 환경을 위한 최초의 검색 스킴이기 때문에 비교할 만한 기존의 스킴은 없다. SIS-D와 SIS-C를 비교해 보면, 이들의 차이점은 SIS-D는 키체인을 사용한 한명의 사용자와 서버간의 직접적인 검색이고 SIS-C는 GC를 경유한다는 데서 비롯된다.

SIS-D에서 사용자는 하나의 단어를 검색하려면 누적된 세션의 횟수만큼 트랩도어를 생성해야 하고, 다운 받은 문서에 대해 평문이 나올 때까지 키체인을 계속 적용하여 복호화해야 한다. 이는 저장되었던 문서 전체를 다시 재암호화 하는 것에 비하면 상당히 효율적이지만, 멤버쉽 체인지가 잦은 다이나믹한 모바일 환경에선 부적합하다. 그러나 앞서 처음에 가정한 환경은 모바일 환경이 아니라 기업이나 공공기관 같은 조직체이며, 이런 조직체들은 그들의 문서를 분기/반기/년간 등 기간별로 관리한다. 따라서 키체인의 길이를 적당하게 선택하기만 한다면, SIS-D는 그런 그룹의 멤버들이 기간별 문서 검색을 하는데 아주 효율적인 스킴이라고 할 수 있다.

SIS-C는 세션 변동에 관계없이 고정된 키 값을 GC가 가지고 있으므로 사용자 측면의 계산량은 훨씬 적으나 반드시 GC를 경유해야 하므로 전송횟수가 증가하는 단점을 지닌다. 또 이로 인해 GC의 부하가 커지는 면이 있기도 하나, 그대신 GC에게 TTP 및 믹스넷의 기능을 부여함으로써 문서 검색 환경에서의 프라이버시를 보다 강화시킬 수 있다.

5.3 응용 시나리오

앞서 분석한 특성을 바탕으로 본 스킴의 적용 가능한 환경을 살펴본다. 처음 설정한 환경처럼 이것은 계열사 및 많은 하위 부서들로 구성된 기업이나 많은 세부 조직들로 구성된 관공서나 병원 같은 조직체에 적합한 스킴이다.

종합 병원을 예로 들어 보자. 그림 1을 참고로 TG는 거대한 최상위 그룹인 종합 병원을 나타내며, 그 하위 그룹인 $(g_1,g_2,\cdots,g_7)$는 내과 병동, 정신과 병동, 방사선과, 원무과, 식당 등 병원내 여러 부서를 나타내고, 그리고 그룹 멤버들$(p_1,p_2,\cdots,p_{15})$은 병원의 직원들을 나타낸다.

g_1을 정신과 병동으로 가정하면, 정신과 환자들의 의무 기록은 그 병동 의료진들을 제외하고 다른 어떤 사람에게도 보여져서는 안된다. 오직정신과 병동의 비밀키를 가진 정신과 의사와 간호사만이 그것을 기록하고 열람할 수 있다.*

그렇기 때문에 제안한 프로토콜의 접근 권한은 그 조직내 다른 하위 그룹 멤버 및 서버 관리자가 병원에 관한 어떤 정보를 누출하는 것을 방지한다. 특히 내부자에 의한 프라이버시 침해(예, 등록 정보가 보험 회사 및 금융 기관으로 유출되거나 성병이나 성형 수술력과 같은 민감한 의무 기록이 누출되어 사회적 이슈를 일으킨 경우 등)가 다소 빨리 증가하고 있는 점을 고려해 본다면, 우리의 새로운 스킴은 절대적으로 필요한 것이다. 따라서 권한을 부여 받지

* 원칙적으로 의무 기록은 담당 의료진외의 다른 사람이 열람할 수 없다. 의료법규 참고.

않은 사람이나 서버 관리자로부터 비밀 정보를 보호하기 위해서는 문서의 암호화뿐만 아니라, 엄격한 접근 통제가 필요하다고 보여진다.

또 병동에 새 의사가 한명 전입되어 온다면, 그는 본인의 할당 받은 환자의 과거력 파악을 위해서 이전 문서들을 참고로 해야 한다. 때문에 이전 문서의 재암호화 없이 모든 문서의 검색이 가능한 우리의 스킴은 이런 그룹 환경에 적합하다.

Ⅵ. 결 론

지금까지 암호화된 문서를 검색하는 환경에서 그 사용자를 그룹으로 확장하여 공유문서 및 개인의 비밀문서 모두를 검색하는 스킴을 살펴보았다.

이는 대그룹, 소그룹별, 개인별로 접근 제한 하에 검색 가능한 환경이며, 그룹 환경에서 그룹키가 갱신되어도 문서의 재암호화 없이 이전 문서까지 검색 가능하게 한다. 또, 검색어에 트랩도어를 사용하여 암호화된 문서를 다운받음으로써 사용자의 프라이버시를 보장할 수 있고, 특히 SIS-C에서 GC에 TTP로서의 믹스넷 기능을 하게 함으로써 완벽에 가까운 프라이버시를 보장할 수 있다.

하지만, SIS-C에서는 GC를 반드시 경유해야 한다는 점과 SIS-D는 검색시 생성해야 할 트랩도어의 수와 그로 인한 많은 계산량 때문에 세션 변동이 잦은 모바일 환경에 부적합하다는 단점이 있다. 따라서 GC없이 모바일 환경에도 적합한 그룹 환경을 위한 연구가 앞으로 계속 필요하다고 보여진다.

참 고 문 헌

[1] D. Song, D. Wagner, and A. Perrig. "Practical techniques for searches on encrypted data." In Proceedings of IEEE Symposium on Security and Privacy, pages 44-55. IEEE, May 2000.

[2] Eu-Jin Goh. "Secure Indexes." A early version of this paper first appeared on the Cryptology ePrint Archive on October 7th 2003. May 5, 2004

[3] Y.-C. Chang and M. Mitzenmacher. "Privacy preserving keyword searches on remote encrypted data." Cryptology ePrint Archive, Report 2004/051, Feb 2004.

[4] B. Waters, D. Balfanz, G. Durfee, and D. Smetters. "Building an encrypted and searchable audit log." In Proceedings of the 11th Network and Distributed System Security (NDSS) Symposium, pages 205-214. Internet Society (ISOC), Feb 2004.

[5] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. "Private information retrieval." Journal of the ACM, 45(6):965-981, Nov 1998.

[6] David Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms" In Communications of the ACM 4(2), February 1981.

[7] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. "Public-key encryption with keyword search." In C. Cachin, editor, Proceedings of Eurocrypt 2004, LNCS. Springer-Verlag, May 2004.

[8] S. Bellovin and W. Cheswick. "Privacy-enhanced searches using encrypted bloom filters." Cryptology ePrint Archive, Report 2004/022, Feb 2004.

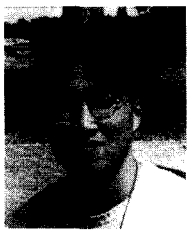
[9] B. Bloom. "Space/time trade-offs in hash coding with allowable errors." Communications of the ACM, 13(7): 422-426, Jul 1970.

[10] UC Berkeley|CS 170: Ecient Algorithms and Intractable Problems Handout 10 Lecturer: David Wagner February 27, 2003

[11] Markus Jakobsson and Ari Juels. "An Optimally Robust Hybrid Mix Network (Extended Abstract)" In the Proceedings of Principles of Distributed Computing - PODC '01, 2001.

〈著者紹介〉

박 현 아 (Park Hyun-a) 학생회원
2003년 2월: 고려대학교 수학과 졸업
2005년 2월: 고려대학교 정보보호대학원 석사
2005년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 암호프로토콜, 익명성 연구, DB 프라이버시



변 진 욱 (Byun Jin-wook) 학생회원
2001년 2월 : 고려대학교 전산학과 졸업
2003년 2월 : 고려대학교 정보보호대학원 석사
2003년 3월~현재 : 고려대학교 정보보호대학원 박사수료
〈관심분야〉 암호프로토콜, 키 교환, 익명성 연구, DB 보안

이 현 숙 (Rhee Hyun-sook) 학생회원
1998년 2월 : 단국대학교교 수학과 졸업
2000년 2월 : 단국대학교 수학과 석사
2001년 3월~현재 : 고려대학교 정보보호대학원 박사수료
〈관심분야〉 암호 프로토콜, 익명성 연구, DB 보안

이 동 훈 (Lee Dong-hoon) 정회원
1983년 8월 : 고려대학교 경제학사
1987년 12월 : Oklahoma University 전산학 석사
1992년 5월 : Oklahoma University 전산학 박사
1993년 3월~1997년 2월 : 고려대학교 전산학과 조교수
1997년 3월~2001년 2월 : 고려대학교 전산학과 부교수
2001년 2월~현재 : 고려대학교 정보보호대학원 부교수
〈관심분야〉 암호프로토콜, 암호이론, USN 이론, 키 교환, 익명성 연구, PET 기술