情報保護學會論文誌
第 22 卷 第 3 號, 2012. 6

소셜 네트워크에서 프라이버시를 보호하는 효율적인 거리기반 접근제어*

정 상 임,† 김 동 민, 정 익 래‡
고려대학교 정보보호대학원

Efficient Hop-based Access Control for Private Social Networks*

Sang Im Jung,† Dong Min Kim, Ik Rae Jeong‡
Graduate School of Information Security, Korea University

요 약

싸이월드, 페이스북과 같은 소셜 네트워킹 서비스는 개인적인 데이터를 지인들과 공유하는데 매우 유용하다. 이들은 대부분 중앙 집중형 서버를 기반으로 하는데, 이 같은 시스템은 사용자들의 모든 통신 내역이 서버에게 노출된다는 단점을 가진다. 이러한 문제점을 개선하기 위해서 p2p 시스템에 착안한 분산된 소셜 네트워킹 서비스와 그 안에서 타인의 데이터에 접근하는 것을 제어하는 연구가 진행 중이다. 기존의 접근제어 기법에서는 신뢰하는 제 3기관이 필요하거나 프로토콜에 참여하는 모든 사용자들이 온라인 상태여야 하고, 사용자들이 분산된 방식으로 구축한 소셜 네트워크가 서버에게 노출되는 단점이 존재했다. M. Atallah 등은 처음으로 암호학적인 키 관리 기법을 활용해서 기존의 기법들이 지닌 문제점을 모두 해결했지만, 제안된 기법이 매우 비효율적이라는 한계가 있었다. 본 논문에서는 이 기법이 가진 비효율을 분석하고, 키 관리 기법이 아닌 대칭키 기반의 환형(circular) 암호를 최초로 적용하여 효율적인 접근제어 기법을 제안한다. 제안하는 기법은 온라인 상에 구축된 소셜 네트워크를 통해서 사용자 데이터에 대한 접근을 분산된 방식으로 제어하고, 서버가 그 네트워크를 추론할 수 없도록 기존의 기법보다 향상된 효율성과 안전성을 제공한다.

ABSTRACT

Because people usually establish their online social network based on their offline relationship, the social networks (i.e., the graph of friendship relationships) are often used to share contents. Mobile devices let it easier in these days, but it also increases the privacy risk such as access control of shared data and relationship exposure to untrusted server. To control the access on encrypted data and protect relationship from the server, M. Atallah et al. proposed a hop-based scheme in 2009. Their scheme assumed a distributed environment such as p2p, and each user in it shares encrypted data on their social network. On the other hand, it is very inefficient to keep their relationship private, so we propose an improved scheme. In this paper, among encrypted contents and relationships, some authenticated users can only access the data in distributed way. For this, we adopt 'circular-secure symmetric encryption' first. Proposed scheme guarantees the improved security and efficiency compared to the previous work.

Keywords: Social Network, Access Control, Privacy

접수일(2011년 11월 8일), 게재확정일(2011년 11월 18일)
* 이 연구에 참여한 연구자(의 일부)는 '2단계BK21사업'의 지원비를 받았음

† 주저자, green86@daum.net
‡ 교신저자, irjeong@korea.ac.kr

I. 서 론

최근 페이스북과 같은 소셜 네트워킹 서비스(Social Networking Services, SNS)를 이용하여 지인들과 개인적인 데이터를 공유하는 사용자들이 증가하고 있다. 이 서비스는 중앙 집중형 웹 서버를 기반으로 하기 때문에 사용자들의 모든 통신 내역이 서버에 노출된다. 이러한 취약점으로부터 사용자 프라이버시를 보호하고 이전과 동일한 서비스를 제공하기 위해서는 공유할 데이터와 네트워크 구성도(network topology)에 대한 접근제어가 필요하다. 신뢰하는 사용자만 자신의 데이터와 연결정보를 열람할 수 있어야 하고, 권한이 없는 사용자와 신뢰할 수 없는 서비스 제공자는 그것들을 열람할 수 없어야 한다. 이는 효율성을 고려하여 대칭키 기반의 데이터 암호화를 적용할 수 있겠지만 암/복호화에 쓰이는 키를 여럿이 공유해야 한다는 점에서 키 관리의 문제점이 발생한다.

사용자들이 그들의 신뢰 수준(trust level)에 따라 타인의 개인적인 데이터에 부분적으로 접근할 수 있도록, 본 논문에서 제안하는 기법은 다음과 같은 시스템을 고려한다. 주어진 시스템에서 두 사용자의 경로(path)는 소셜 네트워크에서 그들을 최단으로 잇는 사용자들의 순서 있는 집합(ordered set)이고, 그 집합 내 원소들의 총 개수를 경로의 길이(혹은 두 사용자가 떨어진 거리)로 정의한다. 또한, 두 사용자 간의 신뢰 수준은 그들의 거리에 반비례한다고 가정한다. 실제로 사용자의 관점에서 보았을 때, 자신의 친구를 친구의 친구보다 더 신뢰한다고 여길 것이므로 이 같은 가정은 매우 자연스럽다. 만약 두 사용자가 소셜 네트워크에서 연결되지 않았다면, 그들이 떨어진 거리는 ∞로 정의한다. 데이터 소유자(data owner)는 자신의 저장소에 암호화된 데이터를 추가/삭제할 수 있고, 이 때 사용되는 비밀키를 접근키라 칭한다. 데이터 소유자는 자신을 통해 접근할 수 있는 지인들의 목록과 그 접근키를 암호화해서 서버에 저장한다. 각 사용자들은 서버에 저장된 값을 복호화하면서 필요한 접근키를 유도한다. 이 유도과정은 데이터 소유자와의 거리만큼 반복적으로 수행되고, 데이터에 맞는 권한을 가진 사용자라면 유도한 접근키를 통해 원하는 데이터를 열람할 수 있다.

제안하는 기법은 위와 같이 정의된 시스템 내에서 다음과 같은 성질을 모두 만족한다. 데이터 소유자로부터 같은 거리에 놓인 사용자들은 동일한 접근키를 유도해야 하고, 신뢰 수준이 사전에 정의된 최소 수준보다 낮은 사용자는 데이터 소유자의 키 중 어떠한 값도 유도할 수 없어야 한다. 서버는 사용자들의 키를 전혀 알 수 없고, 저장하고 있는 소셜 네트워크의 전체 구성도도 알 수 없어야 한다.

그동안 p2p 기반의 분산된 소셜 네트워킹 서비스에서 신뢰할 수 있는 사용자만 데이터에 접근할 수 있도록 여러 가지 방식이 제안되었다. 준동형(homomorphic) 암호기술, 속성 기반 암호기술 등을 이용하여 인맥정보를 암호화하고, 권한이 있는 사용자만 복호화하여 접근키를 얻는 연구들이 그 예이다. 하지만 이 기법들은 그것에 활용된 준동형/속성 기반 암호기술들이 비효율적이라는 단점 때문에, 전체적인 기법의 효율성도 여전히 문제로 남아있다. 본 논문에서 제안하는 기법은 이 문제를 해결하기 위해 대칭키 기반의 환형 암호시스템을 적용한다. 본 논문의 가장 큰 공헌은 환형 암호를 적용함으로 인해서, 접근키와 인맥정보를 한꺼번에 암호화하고 서버로부터 인맥정보도 안전하게 보호한다는 점이다. 접근키와 인맥정보를 함께 다뤄서 기법의 효율성을 개선했고, 이전 연구들에서 해결하지 못했던 '인맥정보의 보호'라는 공개 과제(open problem)를 해결했다.

본 논문의 구성은 다음과 같다. 2장에서 관련된 연구들을 알아보고, 3장에서 제안하는 기법을 소개하고 4장에서 이것을 분석한다. 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구

2.1 소셜 네트워크에서의 접근제어

초기의 접근제어[1]는 사용자의 신뢰 수준과 관계 유형을 기반으로 믿을 수 있는 제 3기관을 통해 데이터로의 접근을 제어한다. 하지만 믿을 수 있는 제 3기관은 현실적으로 존재하기 어렵고 공격의 집중대상이 되는 병목현상을 초래할 수 있기 때문에, 이후부터 이러한 문제점을 개선한 기법들이 제안되었다. 이어서 제안된 기법[2,3]은 모두 제 3기관이 필요 없지만, 프로토콜을 수행하는 모든 사용자들이 동시에 온라인 상태여야 한다는 단점을 지니고 있다. 또 다른 기법[4]은 공개키 암호 프로토콜을 이용해서 이전과 동일한 기능을 제공하는 기법을 설계했다. 이 프로토콜에서 데이터 소유자는 자신과 데이터 요청자 사이의 거리를 기준으로 접근여부를 결정할 수 있지만, 각 신뢰관계

에 해당하는 가중치가 두 사용자의 경로에 놓인 중간 노드에게 모두 드러나는 단점이 존재한다. 관련된 다른 기법[5]은 이러한 단점을 보완한 접근제어를 제공했으나 여전히 신뢰할 수 있는 제 3기관을 필요로 한다.

2.2 M. Atallah 등의 접근제어 기법[6]

2009년에 M. Atallah 등은 위에서 언급한 모든 단점을 해결한 최초의 기법을 제안했다[6]. 이 접근제어 기법은 같은 해에 발표된 키 관리 기법[7]을 활용하여 초기화, 질의, 응답 알고리즘을 설계했고, 신뢰할 수 있는 제 3기관이 필요 없는 분산된 방식을 채택했다. 또한, 서버가 최소한의 기능만 수행하고, 프로토콜을 수행하는 모든 사용자들이 동시에 로그인 상태여야 하는 문제도 해결했다. 더불어 키 유도 과정에서 서버가 주어진 소셜 네트워크를 추론할 수 있는 문제를 해결했지만, 그 해결방안이 매우 비효율적이라는 새로운 문제점이 제기되었다. 본 논문의 아이디어는 이 문제점을 개선시키려는 데에서 출발했다. M.Atallah 등의 기법[6]은 접근제어를 분산된 방식으로 해결하기 위해서 모든 사용자들이 자신과 연관된 소셜 네트워크를 기반으로 자신만의 접근 그래프를 생성하고, 이를 암호화해서 서버에 저장한다. 이렇게 모아진 암호화된 접근 그래프는 서버에서 모두 통합되고, 이 통합된 정보는 다른 사용자의 접근키를 유도하려 할 때 사용된다. 접근하고자 하는 사용자가 자신으로부터 지정된 거리 내에 존재하지 않으면 접근키를 유도할 수 없다.

2.3 대칭키 기반의 안전한 환형 암호시스템[8]

LWE(Learning a linear function With Errors) 문제는 Lattice 기반 암호시스템에서 최근 수 년 동안 활발하게 암호학적인 환경에 적용된 난제이다. LPN(Learning Parity with Noise) 문제는 LWE 문제의 특수한 경우로, LWE문제와 더불어 양자 컴퓨팅 하에서도 현실적인 시간 내에 풀 수 있는 알고리즘이 없다고 알려진 난제이다.

2009년 B. Applebaum 등은 이러한 LPN 문제의 어려움으로 안전성을 제공하는 대칭키 기반의 효율적인 환형 암호시스템을 설계했고, 그것의 안전성을 이론적으로 증명했다.

III. 제안하는 기법

제안하는 기법은 형성된 소셜 네트워크를 권한 없는 다른 사용자들과 서버로부터 보호하며 사용자가 키 유도에 필요한 공개정보만 내려 받을 수 있도록 키에 대한 접근제어를 수행한다. 우선 개괄적인 알고리즘을 간단한 예제로 살펴본 뒤에, 구체적인 기법을 소개한다.

3.1 예제

이 예제는 제안하는 기법의 아이디어를 간략히 소개하기 위한 것이다. 각각의 사용자보다는 서버의 관점에서 전체 알고리즘이 어떻게 수행되는지에 초점을 맞춰 개괄적인 아이디어를 살펴본다. 우선 예제에 사용될 기호들을 〔표 1〕과 같이 정리한다.

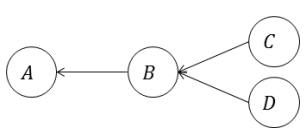
서버는 구성된 소셜 네트워크 $G=(V,E)$를 접근 그래프 $G'=(V',E')$로 전환한다. 신뢰 그래프 G에서

〔표 1〕 기법에 사용될 기호와 정의

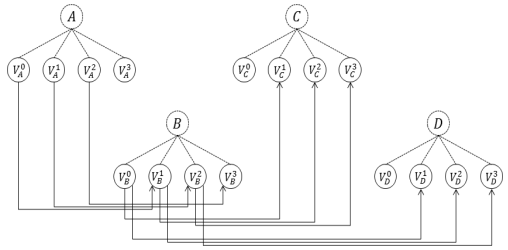
기호	의미
$G=(V,E)$	소셜 네트워크 그래프, V는 노드(사용자)들의 집합, E는 연결선(친분관계)들의 집합
$depth_G(u,v)$	그래프 G에서 노드 u에서 v까지 이르는 거리(hop)
L	데이터 소유자가 자신의 데이터를 공유할 최대 거리, 사용자 u는 $depth_G(u,v) \le L$인 사용자 v에게만 접근을 허용
$G'=(V',E')$	G로부터 유도된 접근 그래프, V'는 노드들의 집합, E'는 연결선들의 집합
$G_u'=(V_u',E_u')$	$G' \subseteq G'$에서 u와 직접 관련된 노드와 연결선들로 이루어진 그래프
V_u^d	사용자 u로부터 d만큼 떨어진 사용자를 위해 생성된 u의 접근노드
k_u^d	사용자 u로부터 d만큼 떨어진 사용자에게 허용할 u의 d번째 데이터 접근키
y_u^d	사용자 u의 d번째 키 k_u^d로 접근할 수 있는 ID와 해당되는 접근키를 k_u^d로 암호화한 공개 값
pub	서버가 저장하고 있는 전체 사용자들의 공개 값 $\{y_u^d : 0 \le d \le L-1, u \in V\}$

노드 $u \in V$는 실제 사용자 u를 의미하고, 연결선 $(u,v) \in E$은 사용자 u가 사용자 v를 신뢰함을 나타낸다. 접근 그래프 G'에서 노드 $V_u^d \in V'$는 사용자 가 자신으로부터 거리가 d만큼 떨어진 사용자에게 부여할 접근정보를 나타낸다. 또한 그래프 G'의 연결선 $(V_v^d, V_u^{d+1}) \in E'$은 사용자 $v \in V$의 d번째 접근키 k_v^d로 사용자 $u \in V$의 $d+1$번째 접근키 k_u^{d+1}를 유도할 수 있음을 의미한다. 따라서 소셜 네트워크를 구성하는 임의의 사용자 $u \in V$는 접근 그래프 G'에서 $L+1$개의 노드 쌍 $\{V_u^d\}_{d=0}^{L}$를 가지고, 소셜 네트워크를 구성하는 임의의 연결선 $(u,v) \in E$은 접근 그래프 G'에서 연결선 쌍 $\{(V_v^d, V_u^{d+1}) \in E'\}_{d=0}^{L-1}$을 가진다. 각 그래프 G, G'에서 연결선이 가지는 상이한 의미 때문에, 두 그래프 내의 연결선 방향은 반대이다. 사용자 u가 v를 신뢰한다면 자신의 데이터에 접근하도록 허용할 것이기 때문에, v는 u의 접근키를 얻을 수 있도록 v에서 u로의 연결선이 생성된다. 또한 키 유도 과정이 거리를 증가시키면서 반복적으로 수행되기 때문에, 각 연결선의 도착노드는 시작노드보다 하나 더 큰 거리 $depth$를 가져야 한다. 신뢰 그래프 G에서 $depth_G(u,v) \le L$인 두 사용자는 접근 그래프 G'에서 임의의 두 사용자 사이에 경로(path)가 존재하고, 이를 통해 해당하는 접근키를 유도한다.

예를 들어, 〔그림 1〕과 같은 소셜 네트워크를 사용하고 $L=3$이라 할 때, 생성되는 접근 그래프는 〔그림 2〕와 같다. 점선으로 표현한 부분은 접근 그래프의 가시성을 높이기 위한 것이고, 실선으로 그려진 노드와 연결선이 실제 접근 그래프를 구성한다. 사용자 B의 경우, 기반으로 하는 환형 암호화〔8〕의 암호화 알고리즘 $CSEenc_{k_B^d}(\bullet)$을 이용해서 B의 각 노드 V_B^d에 키 k_B^d와 공개 값 $y_B^d = CSEenc_{k_B^d}((C, k_C^{d+1}) \| (D, k_D^{d+1}))$를 할당한다. 이 상태에서 사용자 A는 자신의 키 $\{k_A^d\}_{d=0}^{2}$로 $\{y_A^d\}_{d=0}^{2}$을 복호화해서 $\{k_B^d\}_{d=1}^{3}$를 유도할 수 있고, 같은 방식으로 $\{k_C^d, k_D^d\}_{d=2}^{3}$를 다시 얻을 수 있다. 이 유도과정에서는 두 노드간의 연결 경로가 있어야 가능한 것으로, $depth$가 증가할수록 유도할 수



〔그림 1〕 간단한 소셜 네트워크의 방향성 있는 그래프 G



〔그림 2〕 신뢰 그래프 G로부터 생성한 접근 그래프 G'

있는 키가 하나 씩 줄어들어 사전에 정의된 L 범주 내의 데이터만 접근 가능하다. 주어진 접근 그래프에서 V_A^0에서 V_C^1로 이어지는 경로가 존재하지 않으므로, A는 C의 1번째 키를 얻을 수 없고, 이와 같은 방식으로 각자의 신뢰수준에 맞는 키만 유도할 수 있다.

3.2 소셜 네트워크에서 프라이버시를 보호하는 효율적인 거리기반 접근제어

제안하는 기법은 초기화 단계(Setup Phase), 키 유도 단계(Derive Phase), 그리고 동적인 소셜 네트워크를 지원하는 확장된 단계(Extended Phase)로 구성된다. 각 단계 별로 다음과 같은 알고리즘이 사용된다.

1) 초기화 단계(Setup Phase) : $Setup$
2) 키 유도 단계(Derive Phase) ; $Derive$
3) 확장된 단계(Extended Phase) : $Offer/Accept$, $Revoc/Delete$

위의 알고리즘에 내부적으로 쓰이는 함수들은 다음과 같다.

- $Authenticate(u, pwd_u)$: 주어진 사용자 u가 자신의 비밀번호 pwd_u로 인증 받는다.
- $Send(u, v, M)$: 사용자 u가 사용자 v에게 메시지 M을 전달한다.
- $Renew(y, P)$: 집합 P의 원소 y를 갱신한다.
- $CSEkeygen(1^n)/CSEenc_{k_{node}^d}(\bullet)/CSEdec_{k_{node}^d}(\bullet)$: 환형 암호시스템의 키 생성 및 암/복호화 알고리즘
- $friend$; 사용자 v와 접근키 k_v^{d+1}의 쌍 (v, k_v^{d+1})들이 접합되어있는 문자열
- $nodeset/keyset$: $friend$에서 사용자들만 모아놓은 집합($nodeset$)과 접근키만 모아놓은 집합($keyset$)

〔표 2〕 초기화 알고리즘

$Setup(G_{user}, L, 1^n)$	
1:	$V_{user}' = \varnothing$
2:	$E_{user}' = \varnothing$
3:	$FOR\ \ all\ v \in V_{user} \setminus \{user\}\ DO$
4:	$\quad FOR\ \ d=1\ to\ L\ DO$
5:	$\quad\quad V_{user}' = V_{user}' \cup \{V_v^d\}$
6:	$\quad END FOR$
7:	$END FOR$
8:	$FOR\ \ d=0\ to\ L\ DO$
9:	$\quad V_{user}' = V_{user}' \cup \{V_{user}^d\}$
10:	$\quad k_{user}^d = CSEkeygen(1^n)$
11:	$END FOR$
12:	$FOR\ \ all\ (v, user) \in E_{user}\ \ DO$
13:	$\quad FOR\ \ d=0\ to\ L-1\ \ DO$
14:	$\quad\quad E_{user}' = E_{user}' \cup \{(V_{user}^d, V_v^{d+1})\}$
15:	$\quad END FOR$
16:	$END FOR$
17:	$FOR\ \ all\ V_{user}^d \in V_{user}'\ \ DO$
18:	$\quad IF\ \ (V_{user}^d, V_v^{d+1}) \notin E_{user}'\ \ THEN$
19:	$\quad\quad friend = NULL$
20:	$\quad ELSE\ FOR\ \ (V_{user}^d, V_v^{d+1}) \in E_{user}'\ \ DO$
21:	$\quad\quad friend = friend \| (v, k_v^{d+1})$
22:	$\quad\quad END FOR$
23:	$\quad END IF$
24:	$\quad y_{user}^d = CSEenc_{k_{user}^d}(friend)$
25:	$\quad pub_{user} = pub_{user} \cup y_{user}^d$
26:	$END FOR$
27:	$Send(user, server, pub_{user})$

3.2.1. 초기화 단계(Setup Phase)

사용자 A는 자신과 관련된 L수준의 접근 그래프 G_A를 생성하고, 그 그래프의 모든 노드 $\{V_A^d\}_{d=0}^L$에 적절한 키 $\{k_A^d\}_{d=0}^L$를 할당한다. 또한 자신을 신뢰하는 친구의 키도 암호화해서 노드에 할당한 뒤 서버에 공개 값을 업로드 한다. 구체적인 알고리즘은 〔표 2〕로 정의한다.

3.2.2. 키 유도 단계(Derive Phase)

사용자들은 자신이 원하는 데이터에 접근하기 위해서 데이터 소유자의 키를 유도한다. 타깃으로 하는 키를 유도하기 위해 거리를 하나씩 증가시키면서 경로에 놓인 공개 값을 모두 복호화한다. 〔표 3〕은 키 유도 알고리즘을 구체적으로 정의한다.

〔표 3〕 키 유도 알고리즘

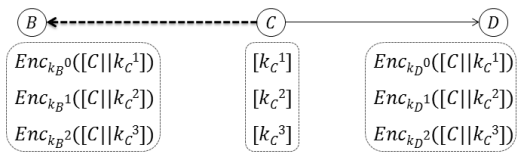
$Derive(pub, snode, ctr, dnode, 1^n)$	
1:	$FOR\ \ d=ctr\ to\ L-1\ DO$
2:	$\quad y_{snode}^d \leftarrow pub$
3:	$\quad (nodeset, keyset) = (nodeset, keyset) \cup CSEdec_{k_{snode}^d}(y_{snode}^d)$
4:	$END FOR$
5:	$IF\ \ dnode \in nodeset\ \ THEN$
6:	$\quad RETURN\ \{k_{dnode}^d\}_{d=ctr+1}^L \subset keyset$
7:	$ELSE\ \ FOR\ all\ w \in nodeset\ DO$
8:	$\quad Derive(pub, w, ctr+1, dnode, 1^n)$
9:	$\quad END FOR$
10:	$END IF$

3.2.3. 확장된 단계(Extened Phase)

소셜 네트워크는 신뢰관계의 수립과 제거가 수시로 발생하므로 이러한 동적인 네트워크를 반드시 지원해야 한다. 첫째로, 새로운 관계가 생성되었을 때 접근 그래프의 수정을 다루는 알고리즘은 $Offer$과 $Accept$이다. 소스노드 $snode$가 $dnode$를 신뢰한다면 자신의 0번째를 제외한 접근키를 안전한 채널을 통해 전달($Offer$)하고, $dnode$는 받은 키 값을 포함시켜 자신의 공개 값을 갱신($Accept$)한다. 둘째로, 알고리즘 $Revoc$은 신뢰관계를 삭제했을 때, 그로인한 모든 과정을 다룬다. 〔그림 3〕는 사용자 B, C, D로 구성된 간단한 신뢰 그래프에서 연결선 $(B, C) \in E$의 삭제로 변경되어야 할 값들을 나타낸다.

사용자 C가 B, D와 맺고 있는 신뢰관계로 인해 B, D는 각자 C의 키 쌍을 암호화해서 서버의 pub에 저장하고 있다. 이 때 사용자 C가 더 이상 B를 신뢰하지 않는다면, 즉 다시 말해서 연결선 $(B, C) \in E$가 삭제된다면, 다음과 같은 과정이 이뤄져야 한다.

1) B는 자신의 공개 값에서 C와 관련된 정보를 제



〔그림 3〕 3명의 사용자로 구성된 신뢰 그래프에서 연결선 $(B, C) \in E$ 의 삭제

〔표 4〕 질의/응답($Offer/Accept$) 알고리즘

	$Offer(snode, pwd_{snode}, dnode, \{k_{snode}^d\}_{d=1}^{L})$
1:	$Authenticate(snode, pwd_{snode})$
2:	$Send(snode, dnode, \{k_{snode}^d\}_{d=1}^{L})$
	$Accept(snode, G_{dnode}', pwd_{dnode}, dnode, \{k_{snode}^d\}_{d=1}^{L})$
1:	$Authenticate(dnode, pwd_{dnode})$
2:	$FOR \quad d=0 \ to \ L-1 \ \ do$
3:	$V_{dnode}' = V_{dnode}' \cup \{V_{snode}^{d+1}\}$
4:	$E_{dnode}' = E_{dnode}' \cup \{V_{dnode}^{d}, V_{snode}^{d+1}\}$
5:	$friend = \{\ \}$
6:	$IF \quad snode \in V_{dnode}' \quad THEN$
7:	$friend = CSEdec_{k_{dnode}^d}(y_{dnode}^d)$
8:	$Renew(k_{snode}^{d+1}, friend)$
9:	$ELSE \quad friend = CSEdec_{k_{dnode}^d}(y_{dnode}^d) \| (snode, k_{snode}^{d+1})$
10:	$END IF$
11:	$y_{dnode}^d = CSEenc_{k_{dnode}^d}(friend)$
12:	$Renew(y_{dnode}^d, pub_{dnode})$
13:	$END FOR$

〔표 5〕 차단/삭제($Revoc/Delete$) 알고리즘

	$Revoc((snode, dnode), V_{snode})$
1:	$Send(snode, dnode, sign_{snode}(bye))$
2:	$FOR \quad d=1 \ to \ L \ DO$
3:	$k_{user}^d = CSEkeygen(1^n)$
4:	$END FOR$
5:	$FOR \quad v \in V_{snode} \setminus \{dnode\} \ \ DO$
6:	$Offer(snode, pwd_{snode}, v, \{k_{snode}^d\}_{d=1}^{L})$
7:	$END FOR$
	$Delete((snode, dnode), sign_{snode}(bye))$
1:	$Verify(sign_{snode}(bye))$
2:	$FOR \quad d=0 \ to \ L-1 \ \ do$
3:	$friend = CSEdec_{k_{dnode}^d}(y_{dnode}^d) \setminus (snode, k_{snode}^{d+1})$
4:	$y_{dnode}^d = CSEenc_{k_{dnode}^d}(friend)$
5:	$Renew(y_{dnode}^d, pub_{dnode})$
6:	$END FOR$
1:	$FOR \quad v \in V_{snode} \setminus \{dnode\} \ \ DO$
2:	$Accept(snode, G_v', pwd_v, v, \{k_{snode}^d\}_{d=1}^{L})$
3:	$END FOR$

거해야 한다.

2) C는 B가 더 이상 자신의 데이터에 접근할 수 없게 자신의 키 쌍을 갱신해야 하고, D에게 알린다.
3) D는 자신의 공개 값이 바뀐 C의 키 쌍을 포함하도록 갱신한다.

C가 직접 신뢰하는 친구들만이 C의 키 쌍을 가지고 있기 때문에, 그 값이 수정되어도 D와 같은 사용자들만 연산이 필요하고 그 외의 사용자들은 아무런 영향을 받지 않으므로 제안하는 기법은 동적인 네트워크를 효율적으로 지원한다. 확장된 단계에 쓰일 구체적인 알고리즘은 〔표 4〕,〔표 5〕와 같다.

IV. 분석

제안하는 기법은 2009년에 제안된 기법〔6〕을 개선하였고, 제시한 공개문제(open problem)를 해결하였기 때문에, 효율성 및 안전성 분석은 〔6〕을 바탕으로 한다. 〔6〕은 기존의 접근제어가 가지고 있던 문제(프로토콜에 참여하는 사용자들이 동시에 온라인이어야 하고, 타인의 연결정보가 노출되는 점)들을 최초로 해결하였지만 효율성 측면에서 새로운 문제점이 발생했다. 본 논문에서는 이러한 문제점을 효율적으로 해결하였고, 본 절을 통해서 안전성과 효율성이 개선되었음을 보인다.

4.1 효율성 분석

제안하는 기법은 소셜 네트워크에서 분산된 접근제어를 수행하는 관련 연구들 중 가장 효율적이다.

첫째로, 서버에 질의하여 받아오는 데이터의 양이다. 제안하는 기법에서 각각의 사용자는 자신이 원하는 접근키를 얻기 위해 최대 L반경 내에 있는 사용자의 공개정보만 서버로부터 받고, 나머지 키 유도 과정은 자신의 기기 내에서 수행한다. 이는 $Derive$알고리즘이 호출될 때 마다 서버가 사용자에게 전체 pub을 중복해서 반환해야 했던 기존의 기법보다 매우 효율적이다. 아래의 〔표 6〕를 보면, 서버와의 통신량이 줄어든 것을 확인할 수 있다. E, V, L은 기법에서 사용한 것과 같은 의미이며, n은 맺을 수 있는 최대 친구 수를, $|V|$는 집합 V의 원소 수를 뜻한다. L은 시스템에서 정하는 상수이지만, 소셜 네트워크의 작은 세상 현상(small world phenomenon) 에 의해서 네트워크에서 임의로 선택된 두 사용자 간의 평균 거리가 평

〔표 6〕 〔6〕과의 비교 분석

	〔6〕	제안하는 기법
서버에 저장될 데이터의 양	$O(\|V\|L+\|E\|L)$	$O(\|V\|L+\|E\|L)$
경로 탐색을 위한 동일성 연산 횟수	$O(\|V\|L+\|E\|L)$	$O(\|V\|L+\|E\|L)$
키 유도를 위한 복호화 횟수	$O(L)$	$O(n^{L-1})$
서버와의 통신량	$O(\|V\|L+\|E\|L)$	$O(n^{L-1}L)$
소셜 그래프 추론공격	가능	불가능

균적으로 $\log(|V|)$임을 알 수 있다. 따라서 실제적으로 시스템에서 사용할 값은 데이터 접근을 허용할 사용자 범위가 전체 네트워크에서 차지하는 비율인 $c(0<c\leq 1)$에 대해서 $L\approx c \bullet \log(|V|)$일 것이고, 〔표 6〕의 $O(n^{L-1}L)$는 페이스북의 전체 사용자 수가 5억 이상임을 감안하면 굉장히 향상된 복잡도임을 알 수 있다.

둘째로, 필요한 변수의 양과 그것의 관리가 훨씬 효율적이다. 기존의 기법〔6〕은 키 관리 기법을 적용시키기 위해서 키 유도에 쓸 키와 데이터 암호화에 쓸 키를 따로 관리하고, 개인을 식별할 공개 값을 ID 외에 별도로 설정하여 pub에 저장하는 번거로움이 존재했다. 제안하는 기법은 데이터 암호화에 사용할 키와 키 유도에 사용할 키를 일원화시키고 이에 따른 안전성 문제를 해결하기 위해 환형 암호화 기법을 적용시켰다. 이렇게 함으로 인해서 별도로 관리하던 개인 식별 값을 없애고, 연결정보만 pub에 저장하여 연산의 효율성과 저장량을 동시에 향상시켰다.

셋째로, 연결정보 암호화 후에도 암호화 전의 기능을 무리 없이 제공한다는 점이다. 기존의 기법〔6〕은 서버가 네트워크를 추론해 낼 수 없게 하기 위해서 연결정보를 암호화했지만, 이와 동시에 접근 가능한 데이터들이 직접 연관된 사용자의 것으로 제한되어 암호화 전에 제공하던 기능을 수행할 수 없다. 따라서 〔표 6〕의 키 유도를 위한 복호화 횟수는 이 같은 기능을 원활히 제공하기 위해 반드시 필요한 부하이고, 이 값을 알고리즘의 구조를 개선하여 최적화 하는 것이 최선일 것이다. 제안하는 기법은 기본적으로 필요한 복잡도인 $O(n^{L-1})$로 복호화를 수행하고, 이를 더욱 최적화시키는 것을 추후 연구 과제로 남긴다.

4.2 안전성 분석

본 절에서는 제안하는 기법의 안전성을 분석한다. 〔정리 1〕은 정적인 신뢰 그래프 G로 이뤄진 접근 그래프 G'이 효과적으로 비 인가된 사용자의 접근을 막을 수 있는지 분석한다. 이를 기반으로 〔정리 2〕는 동적인 그래프 G, G'에 대해서도 접근제어가 이뤄지는지 분석한다. 이 때, 중요한 점은 소셜 네트워크는 오프라인 인맥이 반영되기 때문에 대부분의 사용자들은 주어진 프로토콜을 변형하지 않는 반-정직(semi-honest)한 공격자로 가정한다는 점이다. 따라서 분석도 이러한 공격자를 가정하고 이뤄진다. 또한 서버는 암호화된 연결정보를 저장하고 일종의 공개 게시판처럼 자신이 저장한 데이터를 사용자들과 공유한다. 사용자들이 오프라인 상태에서 공개게시판에 접근하여 필요한 공개 값을 다운받고 권한이 있는 값만 복호화 할 수 있다는 점에서, 서버는 특정 사용자가 어떤 사용자의 정보를 다운받는지 추적할 수 없다. 따라서 이후 제시될 〔정리 1〕과 〔정리 2〕는 사용자 입장의 공격자를 가정하고 안전성을 분석한다. 더불어 제안하는 기법은 기존의 기법〔6〕이 보장하는 안전성을 동일한 수준으로 보장하고, 주어진 소셜 네트워크를 서버가 추론할 수 없도록 하여 사용자의 프라이버시를 서버로부터 안전하게 보호한다.

정리 1. 초기화 알고리즘 $Setup(G_{user}, L, 1^n)$을 통해 정적인 신뢰 그래프 G로부터 생성된 접근 그래프를 G'라 할 때, 임의의 두 사용자 $u, v \in G$가 $depth_G(u,v) \leq d$를 만족하면 $V_v^0 \in G'$와 $V_u^{depth_G(u,v)} \in G'$를 잇는 경로가 반드시 존재하고 역도 성립한다.

증명. $l = depth_G(u,v) \leq d$인 사용자 u, v가 있다고 가정하면, 신뢰 그래프 G내에 두 사용자 u, v를 잇는 또 다른 $d-1$명의 사용자들로 구성된 경로 $u, x_1, x_2, \cdots, x_{l-2}, x_{l-1}, v$가 존재한다. 따라서 G로부터 생성된 접근 그래프 G'내에 경로 $V_v^0, V_{x_{l-1}}^1, V_{x_{l-2}}^2, \cdots, V_{x_2}^{l-2}, V_{x_1}^{l-1}, V_u^l$가 존재해야 한다.

모든 연결선은 $Setup$알고리즘의 12~16줄에 의해서 생성되고, $(V_v^0, V_{x_{l-1}}^1) \in E'$는 $Setup(G_v, L, 1^n)$를 통해서 pub에 존재한다. 같은 방식으로 $1 \leq i \leq l-2$에 대해서 $(V_{x_{l-i}}^i, V_{x_{l-1-i}}^{i+1}) \in E'$는 $Setup(G_{x_{l-i}}, L, 1^n)$로 생성되고, $(V_{x_1}^{l-1}, V_u^l) \in E'$는 $Setup(G_{x_1}, L, 1^n)$에 의해서 pub에 존재한다. 따라서 임의의 두 사용자 $u, v \in G$

가 $depth_G(u,v) \le d$를 만족하면 $V_v^0 \in G'$와 $V_u^{depth_G(u,v)} \in G'$를 잇는 경로가 반드시 존재한다.

역에 대해서도 살펴보면, 우선 그래프 G'내에 경로 $V_v^{i_0}, V_{x_{l-1}}^{i_1}, V_{x_{l-2}}^{i_2}, \cdots, V_{x_2}^{i_{l-2}}, V_{x_1}^{i_{l-1}}, V_u^{i_l}$가 존재한다고 가정하자. 접근 그래프 내의 모든 연결선은 *Setup*알고리즘의 12~16줄에 의해서 생성되므로, 반드시 $i_j = i_{j-1}+1$이다. 그러므로 $i_l = i_0 + l \ (0 \le i_0)$이고, $i_l \le d$에 의해서 $depth_G(u,v) = i_l - i_0 = l \le d$이다. 따라서 주어진 접근 그래프 G'내에 경로 $V_v^{i_0}, V_{x_{l-1}}^{i_1}, V_{x_{l-2}}^{i_2}, \cdots, V_{x_2}^{i_{l-2}}, V_{x_1}^{i_{l-1}}, V_u^{i_l}$가 존재하면 $depth_G(u,v) \le d$인 사용자 u, v가 반드시 신뢰 그래프 G에 존재한다.

정리 2. 알고리즘 $Setup(G_{user}, L, 1^n)$, $Offer(snode, pwd_{snode}, dnode, \{k_{snode}^d\}_{d=1}^L)$, $Accept(snode, G_{dnode}{}', pwd_{dnode}, dnode, \{k_{snode}^d\}_{d=1}^L)$을 통해 동적인 신뢰 그래프 G로부터 생성된 접근 그래프를 G'라 할 때, 임의의 두 사용자 $u, v \in G$가 $depth_G(u,v) \le d$를 만족하면 $V_v^0 \in G'$와 $V_u^{depth_G(u,v)} \in G'$를 잇는 경로가 반드시 존재하고 역도 성립한다.

증명. 주어진 정리의 증명은 〔정리 1〕로부터 쉽게 유도된다. *Offer*와 *Accept*는 *Setup*의 일부분과 동일한 과정을 수행하므로, 정리 1에서 이뤄진 증명과 동일한 방식으로, 임의의 두 사용자 $u, v \in G$가 $depth_G(u,v) \le d$를 만족하면 $V_v^0 \in G'$와 $V_u^{depth_G(u,v)} \in G'$를 잇는 경로가 반드시 존재한다. 또한, 역으로 접근 그래프 G'내에 경로 $V_v^{i_0}, V_{x_{l-1}}^{i_1}, V_{x_{l-2}}^{i_2}, \cdots, V_{x_2}^{i_{l-2}}, V_{x_1}^{i_{l-1}}, V_u^{i_l}$가 존재하면 $depth_G(u,v) \le d$인 사용자 u, v가 반드시 신뢰 그래프 G에 존재한다.

제안하는 기법의 안전성 증명은 〔정리 2〕의 대우를 통해 완성된다. 〔정리 2〕의 대우는 신뢰 그래프 G에 대해서 $depth_G(u,v) > d$(경로가 존재하지 않는 경우는 $depth_G(u,v) = \infty$)인 임의의 두 사용자 u, v가 접근 그래프 G'에서 $V_v^0 \in G'$와 $V_u^{depth_G(u,v)} \in G'$를 잇는 경로를 가질 수 없다는 것이다. 제안하는 기법에서 접근키를 유도하는 구조상, 접근 그래프에서 경로가 존재하지 않으면 접근키를 유도할 수 없고 데이터에 접근하는 것도 불가능하다. 결론적으로, 적법한 거리내의 사용자만이 데이터에 접근 가능하다.

V. 결론

본 논문에서는 동적인 소셜 네트워크 환경에서 분산된 방식으로 프라이버시를 보호하는 효율적인 접근제어 기법을 제안하였다. 기존 연구에서 서버가 소셜 네트워크를 추론하지 못하도록 연결정보를 비효율적으로 중복 암호화하는 문제점을 해결하기 위해서, 본 논문에서는 키를 또 다른 키로 암호화하는 환형 암호화를 적용시켰다. 이 기법은 기존의 기법과 동일한 안전성으로 서버로부터 개인의 프라이버시를 보호하고 개선된 효율성을 보장한다. 다만, 서버가 두 사용자 간의 연결경로를 찾기 위해 검색하는 과정에서 너비우선 검색이 사용되는 단점이 존재하므로, 알고리즘 구조의 개선을 통해 검색 과정과 복호화 횟수를 최적화하는 것을 향후 연구 과제로 남긴다. 또한 소셜 네트워크만이 가지는 그래프 이론적 특성을 보안 메커니즘에 적용시켜 소셜 네트워크에 특화된 연구를 구축하는 것이 더 필요하다.

참고문헌

〔1〕 B. Carminati, E. Ferrari, and A. Perego, "Rule-based access control for social networks," On the Move to Meaningful Internet System, LNCS 4278, pp. 1734-1744, 2006.

〔2〕 B. Carminati, E. Ferrari, and A. Perego, "Private relationships in social networks," Proceedings of IEEE International Conference on Data Engineering Workshop, pp. 163-171, Apr. 2007.

〔3〕 B. Carminati and E. Ferrari, "Privacy-aware collaborative access control in web-based social networks," Data and Applications Security XXII, LNCS 5094, pp. 81-96, 2008.

〔4〕 J. Domingo-Ferrer, "A public-key protocol for social networks with private relationships," Modeling Decisions for Artificial Intelligence, LNAI 4617, pp. 373-379, 2007.

〔5〕 V. Alexandre, J. Domingo-Ferrer, S. Francesc, and G.N. Ursula, "Privacy homomorphisms for social networks with private relationships," Elsevier, Computer Networks, vol. 52, no. 15, pp. 3007-3016, Oct. 2008.

〔6〕 K.B. Frikken and P. Srinivas, "Key allocation schemes for private social networks," Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society, pp. 11-20, Nov. 2009.

〔7〕 M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and efficient key management for access hierarchies," ACM Transactions on Information and System Security, vol. 12, no. 3, pp. 18-43, Jan. 2009.

〔8〕 B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," Advanced in Cryptology, CRYPTO'09, LNCS 5677, pp. 595-618, 2009.

〈著者紹介〉

정 상 임 (Sang Im Jung) 학생회원
2010년 2월: 숙명여자대학교 수학과 졸업
2010년 3월~현재: 고려대학교 정보보호학과 석사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론

김 동 민 (Dong Min Kim) 학생회원
2009년 2월: 서울시립대학교 수학과 졸업
2011년 2월: 고려대학교 정보경영공학과 석사 졸업
2011년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론

정 익 래 (Ik Rae Jeong) 정회원
1998년 2월: 고려대학교 전산학과 학사 졸업
2000년 2월: 고려대학교 전산학과 석사 졸업
2004년 8월: 고려대학교 정보보호학과 박사 졸업
2006년 6월~2008년 2월: 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월~현재: 고려대학교 정보경영공학부 교수
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 암호, 암호 이론