情報保護學會論文誌
第17卷 第3號, 2007. 6

ID 기반 위임 네트워크의 성능 개선방안

윤 택 영,[1†] 박 영 호,[2‡] 정 상 태[3]
[1]고려대학교, [2]세종사이버대학교, [3]인하대학교

Improvement in efficiency on ID-based Delegation Network

Taek-Young Youn[1†], Young-Ho Park,[2‡] Sangtae Jeong[3]
[1]Korea University, [2]Sejong Cyber University, [3]Inha University

요 약

서명권한의 위임은 다양한 환경에서 요구되는 암호학적 서비스이다. Mambo 등은 서명권한 위임에 대한 해결방법으로 프록시 서명기법을 제안하였다. 프록시 서명기법이 제안된 후, 한 명의 서명자가 자신의 권한을 한 사람의 프록시 서명자에게 위임하는 기본적인 형태의 위임이 아니라 보다 일반적인 위임 구조를 제공하기 위한 프록시 서명기법들이 제안되었다. 모든 구성 가능한 위임 구조를 포괄할 수 있는 개념으로 위임 네트워크가 Aura에 의해 제안되었고, 이후 Chow 등은 ID 기반의 위임 네트워크를 제안하였다. E는 위임 네트워크에서 발생하는 위임의 개수라고 하고 N은 사용자의 개수라고 하자. 계산 복잡도의 관점에서 Chow 등의 위임 네트워크는 E번의 페어링 연산과 N번의 스칼라 곱셈 연산이 수행된다. 본 논문에서는 E번의 페어링 연산만으로 Chow 등이 제안한 것과 동일하게 동작하는 위임 네트워크를 제안한다. 또한 제안하는 위임 네트워크의 구성을 변형함으로써 N번의 페어링 연산이 요구되는 위임 네트워크를 구성한다.

ABSTRACT

Delegation of signing capability is a common practice in various applications. Mambo et al. proposed a proxy signatures as a solutions for delegation of signing capability. Proxy signatures allow a designated proxy signer to sign on behalf of an original signer. After the concept of proxy signature scheme is proposed, many variants are proposed to support more general delegation setting. To capture all possible delegation structures, the concept of delegation network was proposed by Aura. ID-based cryptography, which is suited for flexible environment, is desirable to construct a delegation network. Chow et al proposed an ID-based delegation network. In the computational point of view, their solution requires E pairing operations and N elliptic curve scalar multiplications where E and N are the number of edges and nodes in a delegation structure, respectively. In this paper, we proposed an efficient ID-based delegation network which requires only E pairing operations. Moreover, we can design a modified delegation network that requires only N pairing operations.

Keywords : *Delegation Network, Proxy Signature, ID-Based Cryptosystem.*

접수일: 2006년 10월 9일; 채택일: 2007년 1월 29일
* 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2004-042-D00159)
† 주저자, taekyoung@cist.korea.ac.kr
‡ 교신저자, youngho@sjcu.ac.kr

Ⅰ. 서 론

서명권한의 위임은 다양한 환경에서 유용하게 사용된다. 예를 들어, 회사의 관리자는 자신이 출장이나 여

행 등의 이유로 자리를 비울 경우에 자신의 서명 능력을 비서등과 같이 믿을 수 있는 사람에게 위임할 수 있다. 서명 능력을 위임받은 사람에게 어떤 문서가 주어지면, 관리자를 대신해서 서명을 수행할 수 있다. 즉, 관리자의 공백을 서명의 위임을 통해 채울 수 있게 된다. 또한, 관리자들은 자신의 일 중에서 중요성이 낮아 다른 사람이 처리해도 되는 사안에 대한 서명을 부하직원 등에게 서명 권리의 위임을 통해 해결할 수 있다. 이 경우에는 관리자의 작업 부담을 줄일 수 있다는 장점을 제공할 수 있다. 그 외에도 많은 기능들이 서명권한의 위임을 통해 제공된다[8]. 이와 같이 유용한 서명권한의 위임을 제공하는 방법으로 Mambo 등에 의해 프록시 서명기법이 제안되었다[8]. 프록시 서명기법은 원래 서명자의 서명권한을 프록시 서명자에게 위임함으로써 원래 서명자 대신 프록시 서명자가 서명 값을 생성할 수 있도록 하는 도구이다. 처음 제안된 프록시 서명에서는 한 명의 서명자가 한 명의 프록시 서명자에게 서명권한을 위임할 수 있도록 구성되어 있다. 그러나 그와 같은 간단한 형태의 권리 위임 형태는 어떤 환경에서는 충분하지 않을 수도 있다. 그 이유로 다음과 같은 환경을 고려해볼 수 있다.

- 한 명의 프록시 서명자에게 서명권한을 위임하는 경우에는 프록시 서명자가 위임받은 권한을 오용하거나 남용할 수 있다. 이런 문제를 해결하기 위해서는 서명권한을 한 명의 프록시에게 위임하는 것보다 여러 명에게 나누어 위임하는 형태로 구성되는 것이 바람직하다.
- 위임을 하고자 하는 서명자가 여러 명인 환경을 고려할 수 있다. 즉, 한 명의 프록시에게 다수의 사용자가 자신의 서명권한을 위임하는 형태이다. 예를 들어, 집단의 구성원들이 한 명의 대표를 뽑고, 집단의 의견을 대표를 통해 나타내고자 하는 경우가 그러하다.
- 조직사회에서는 계층적인 구조가 일반적이기 때문에 연속된 형태의 서명권한의 위임이 요구된다. O라는 사용자가 P라는 사용자에게 서명권한을 위임하고, P는 자신이 위임받은 권한을 P'에게 위임하기를 원할 수 있다. 예를 들어, 어떤 기업에서 사장이 자신의 서명권한을 부사장에게 위임하고, 부사장은 자신이 위임 받은 서명권한을 어떤 부서의 부장에게 위임하고자 할 수 있다. 회사에서 서명이 생성되어야 하는 문서는 많이 있고, 각 문서에 대한 중요성은 다르기 때문에 연속된 형태의 서명권한의 위임을 구성함으로써 일을 계층적으로 분산시킬 수 있다. 이와 같은 환경을 일반적으로 해결하기 위해서는 여러 계층을 갖는 구조를 제공하는 연속된 형태의 서명권한의 위임이 요구된다.

위에 언급된 환경들을 해결하기 위한 것으로 multi-proxy signature[13,9], threshold proxy signature [01,5,12, 11,6], proxy multi-signatures[7], multi-proxy multisignatures[4] 와 같은 다양한 종류의 프록시 서명기법이 제안되었다. 그러나 [2]에서 지적되었듯이, 기존의 기법들은 충분히 일반적이고 유동적이지 못했다. 즉, 모든 가능한 서명권한의 위임 형태를 제공하지 못한다. 가장 일반적이고 유동적인 위임 모델은 Aura에 의해 제안된 위임 네트워크[3]이고, ID 기반의 위임 네트워크가 [2]에서 제안되었다. [2]에서 제안된 위임 네트워크는 앞에서 언급된 모든 위임의 형태를 제공할 수 있을 정도로 일반적이고 유동적인 구조를 갖고 있다. 위임 네트워크가 N개의 노드(node)와 E개의 에지(edge)로 구성되어 있다고 하자. 노드의 개수는 위임에 참여하는 사용자의 수를 의미하고 에지의 개수는 사용자 사이에 발생하는 서명권한 위임의 개수를 의미한다. 이와 같은 가정에서, 기존의 ID 기반 위임 네트워크는 E번의 페어링 연산과 N번의 스칼라 곱셈을 사용한다. 한 명의 사용자가 두 명 이상의 사용자에게 서명권한을 위임할 수 있으므로 일반적으로 E는 N보다 크다고 할 수 있다. 또한 서로 서명권한을 위임하는 경우가 많아질수록 E가 증가하는 것이므로 연산양이 증가하게 된다. 그러므로 위임 네트워크의 복잡도가 증가할수록 [2]에서 제안된 위임 네트워크는 많은 연산을 요구하게 된다.

본 논문에서는 효율적인 ID 기반의 위임 네트워크를 제안한다. [2]에서 제안된 위임 네트워크는 E개의 에지를 검증하기 위해 E번의 페어링 연산이 수행되고, N개의 노드를 검증하기 위해 N번의 스칼라 곱셈이 수행되었다. 본 논문에서 제안하는 위임 네트워크는 E번의 페어링만 수행한다. 즉, N번의 스칼라 곱셈을 적게 사용한다. 또한 제안하는 위임 네트워크의 구성을 약간 변형함으로써 N번의 페어링만 사용하는 위임 네트워크를 구성한다. N은 위임 네트워크에 참여하는 사용자수 이므로 네트워크의 복잡도가 증가하더라도 연산양이 증가하지 않는다. 그러므로 본 논문에서 제안하는 위임 네트워크와 제안하는 것은 변형된 것은 기존의 위임 네트워크보다 매우 효율적이다.

Ⅱ. 기본내용

2.1 Bilinear Maps

$(G_1,+)$와 $(G_2,\times)$를 위수가 q인 두 순환군이라고 하자. bilinear 페어링은 $e: G_1 \times G_1 \rightarrow G_2$로 정의되고 다음과 같은 특성을 만족한다.

1. Bilinearity: 모든 $P,Q,R \in G_1$에 대해 $e(P+Q,R)=e(P,R)e(Q,R)$와 $e(P,Q+R)=e(P,Q)e(P,R)$이 만족한다.
2. Non-degeneracy: $e(P,Q)\neq 1$을 만족하는 두 포인트 $P,Q \in G_1$가 존재한다.
3. Computability: 모든 $P,Q \in G_1$에 대해서 $e(P,Q)$를 계산하는 효율적인 알고리즘이 존재한다.

위의 특성을 만족하는 bilinear map을 admissible bilinear map 이라고 한다. P를 G_1의 생성원이라고 하고, $a,b,c \in GF(q)$라고 하자. 암호학적으로 다음과 같은 문제들이 관심의 대상이 된다.

정의 1. (CDHP: Computational Diffie-Hellman Problem) 주어진 G_1의 생성원 P에 대해 (aP,bP)가 주어지는 경우에, abP를 계산하는 문제를 CDHP라고 한다.

정의 2. (DDHP: Decisional Diffie-Hellman Problem) 주어진 G_1의 생성원 P에 대해 (aP,bP,cP)가 주어지는 경우에, $ab=c \bmod q$ 인지 확인하는 문제를 DDHP라고 한다.

2.2 위임 네트워크에 대한 안전성 모델

결과적으로 얘기하자면, 제안하는 위임 네트워크의 안전성은 ROM(random oracle model) 가정에서 CDHP를 푸는 것과 동치이다. 본 논문에서는 위임 네트워크에서 생성하는 서명을 위조하는 공격 알고리즘 F에서 CDHP를 푸는 알고리즘 A를 구성하는 형태로 제안하는 위임 네트워크의 안전성을 보인다. F의 공격능력을 활용하기 위해서는 F의 공격환경을 구성해주어야 하므로, 증명에서는 다음과 같은 F의 질의에 대한 응답을 구성해주어야 한다. 본 절에서는 제안하는 위임 네트워크에서 사용하는 설정으로 기반으로 설명하도록 한다.

2.2.1 ID 해쉬 오라클에 대한 질의 (Hash Queries on Oracle for Identity)

- F가 사용자 U_i의 ID_i에 대한 공개키를 문의하면, $H_1(ID_i)=Q_i$ 를 만족하는 공개키 Q_i를 생성한다.

2.2.2 메시지 해쉬 오라클에 대한 질의 (Hash Queries on Oracle for Message)

- F가 메시지 m에 대한 해쉬 값을 요구하면, $H_2(m)=M$ 을 만족하는 해쉬값 M를 생성해준다.

2.2.3 비밀키 생성 질의 (Private Key Extraction Queries)

- F가 사용자 U_i의 비밀키를 요구하면, ID_i에 대응되는 비밀키 d_i를 생성해준다. $d_i=sQ_i$를 만족하는 값이고, s는 PKG(Private Key Generator)의 비밀키이다. 이 경우에는 비밀키 생성 질의 이전에 ID_i에 대한 해쉬 질의가 수행되었다고 가정한다.

2.2.4 프록시 서명키 생성 질의 (Proxy Signing Key Generation Queries)

- F가 사용자 U_i의 ID_i에 대한 프록시 서명키를 요구하면 해당되는 프록시 서명키 SK_i를 생성한다.

2.2.5 최종 서명 질의 (Final Signing Qeries)

- F가 어떤 위임 네트워크에서 생성된 최종 서명을 요구하면 위임 네트워크에 대응되는 서명을 생성해준다.

제안하는 위임 네트워크에서는 프록시 서명키 생성 질의와 최종 서명 질의가 아래에서 진술될 각 사용자의 서명 생성 질의의 연속으로 구성된다. 그러므로 증명에서는 각 사용자의 서명 생성 질의에 대한 응답을 구성해줌으로써 프록시 서명키 생성 질의와 최종 서명 질의에 대한 응답으로 대체한다.

2.2.6 서명 질의 (Signing Queries)

- F가 사용자 U_i의 m에 대한 서명을 요구하면 검증과정을 통과하는 서명을 생성해준다. 이 경우에는 서명 생성 질의 이전에 ID_i에 대한 해쉬 질의와 메시지 m에 대한 해쉬 질의가 수행된 것으로 가정한다.

제안하는 위임 네트워크의 안전성은 [14,2]에서와 마찬가지로 가장 강한 공격자를 고려한다. [14,2]에서는 공격자가 공격하고자 하는 대상의 것을 제외한 모든 비밀키(프록시 서명키)를 알고 있는 것으로 가정한다. 위임 네트워크에 대한 공격자 F는 다음과 같이 정의된

다. 위임 네트워크에 대한 안전성 모델은 [2]에서 언급된 것을 인용하였으며, 본 논문에서는 해당 내용을 정의로 구성한 것이다.

정의 3. 위임 네트워크에 대한 위조 공격자 F는 아래의 조건을 만족하는 서명 $(\sigma, ID_1, ..., ID_N m_1, ..., m_E)$을 생성하는 위조 공격자로 정의된다. 이 경우, 위임 네트워크의 구조는 N개의 노드와 E개의 에지로 구성되어 있다고 가정하는데 N은 위임에 참여하는 사용자의 수를 의미하고 E는 각 사용자간의 위임이 수행된 회수를 의미한다.

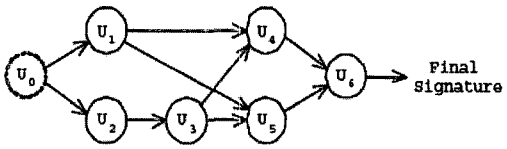
1. 서명은 검증과정은 통과한다.
2. F가 위조한 서명에는 비밀키 생성 질의가 수행되지 않은 사용자 U_i에 대한 서명이 포함되어 있다.
3. F가 생성한 서명은 최종 서명 질의에 문의되지 않은 값이어야 하고, 특히 비밀키 생성 질의가 수행되지 않은 ID_i에 대해서, 서명에 포함된 (ID_i, m_j)는 서명 질의에 문의되지 않은 것이어야 한다.

직관적으로, 위임 네트워크는 F가 의미 있는 서명을 생성할 확률이 매우 낮은 경우에 안전하다고 할 수 있다. 본 논문에서는 앞에서 언급된 것처럼 공격하고자 하는 사용자의 것을 제외한 모든 비밀키를 알고 있는 공격자 F를 가정한다. 정의 3에 정의된 공격자에 대한 위임 네트워크의 안전성은 다음과 같이 정의된다.

정의 4. F를 위임 네트워크에 대한 $(q_{H_1}, q_{H_2}, q_S, q_E)$-위조자 라고 하자. 여기서 qH1, qH2, qS, qE 는 각각 ID에 대한 해쉬 질의의 개수, 메시지에 대한 해쉬 질의의 개수, 서명 질의의 개수 그리고 비밀키 생성 질의의 개수이다. 이때, F의 AdvF는 공격자 F가 정의 3에서 열거된 조건을 만족하는 의미 있는 서명을 생성하는 확률을 의미한다. 위임 네트워크는 것은 공격자 F에 대해 충분히 큰 k에 대해서 다음과 같은 조건을 만족하는 함수 $negl(k)$이 있는 경우에 안전하다고 한다:
$Adv_F^{dnf}(q_{H_1}, q_{H_2}, q_S, q_E) < negl(k)$.

Ⅲ. 제안하는 위임 네트워크

본 절에서는 새로운 ID 기반의 위임 네트워크를 제안한다. 편의상 다음과 같은 기호를 사용하도록 한다.



[그림 1] 위임 네트워크의 예제

- $E_{A,B}$: 사용자 U_A와 U_B를 연결하는 에지
- AE_A: 사용자 U_A의 모든 상위 에지들의 집합
- PE_A: 사용자 U_A에 직접 연결된 모든 상위 에지들의 집합
- AN_A: 사용자 U_A의 모든 상위 노드들의 집합
- PN_A: 사용자 U_A에 직접 연결된 모든 상위 노드들의 집합
- $M_A = \{m_{i,j} | (i,j) \in AE_A\}$, $R_A = \{R_{i,j} | (i,j) \in AE_A\}$

기호에 대한 이해를 돕기 위해 예를 들어 살펴보도록 하자. $m_{i,j}$는 U_i에서 U_j로의 위임 조건을 나타내는 메시지라고 하고, $R_{i,j}$는 U_i에서 U_j로의 위임을 수행하는 과정에서 U_i이 사용하는 난수로 생성한 값이라고 하자. $E_{2,3}$는 U_2에서 U_3로 연결된 에지이고, $AE_5 = E_{2,3}$,, $E_{1,5}, E_{3,5}$, $PE_5 = E_{1,5}, E_{3,5}$, $AN_5 = U_1, U_2, U_3$, $AN_5 = U_1, U_3$이다. 그리고 $M_5 = m_{2,3}, m_{1,5}, m_{3,5}$, $R_5 = R_{2,3}, R_{1,5}, R_{3,5}$이다.

3.1 위임 네트워크의 구성

본 소절에서는 제안하는 위임 네트워크의 구성을 살펴보도록 하자.

3.1.1 시스템 구성 및 키 생성

이 단계에서 PKG는 다음과 같은 과정의 수행으로 시스템을 구축한다.

1. 위수가 q인 두 그룹 G_1과 G_2를 생성한다.
2. G_1의 생성원 P를 선택하고 임의의 난수 s를 선택한 뒤 $Q = sP$를 계산한다.
3. 두 개의 암호학적 해쉬함수 $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_1 \times \{0,1\}^* \rightarrow G_1$를 생성하고 $e : G_1 \times G_1 \rightarrow G_2$는 bilinear map 이라고 하자.

공개 시스템 변수는 $(G_1, G_2, P, Q, H_1, H_2, e)$이고, 대응되는 비밀 시스템 변수는 s이다. 사용자 U_i가 자신의 비밀키를 요청하면 PKG는 $d_i = sQ_i$를 계산해서 U_i에게 안전하게 전송한다. 이때 $Q_i = H_1(U_i)$이다. U_i는 자신이

받은 비밀키를 다음의 관계식을 확인해 봄으로써 검증할 수 있다: $e(d_i,P)=e(Q_i,Q)$. 검증식이 만족하면 U_i는 d_i를 비밀키로 사용한다.

3.1.2 계층적인 프록시 서명키 생성(U_A-> U_B 위임)

초기값으로 W_O는 null string으로 설정하고 $SK_O=d_o$로 설정한다. 이때 $d_O=sQ_O$이고, Q_O는 PKG에 대한 공개키로 가정되므로 $Q_O=P$이고 $d_O=sP=Q$이다. 계층적인 프록시 서명키 생성은 다음과 같은 과정으로 수행된다.

1. U_A는 $r_{A,B}$를 선택하고 $R_{A,B}=r_{A,B}P$를 계산한다.
2. U_A는 $W_{A,B}=H_2(\sum_{i\in PN_A} W_{i,A}\|ID_A\|ID_B\|m_{A,B})$와 $r_{A,B}$ $W_{A,B}$를 계산한다. $m_{A,B}$는 U_A에서 U_B로의 위임 조건을 기술한 보증서(warrant)이다.
3. U_A는 $S_{A,B}=SK_A+r_{A,B}W_{A,B}$에 대해서 $(S_{A,B},M_A,$ $m_{A,B},R_A,R_{A,B})$를 U_B에게 전송한다.
4. U_B는 아래의 조건을 확인하고 조건이 만족할 경우 $S_{A,B}$로 자신의 프록시 서명키를 생성한다

$$e(P,S_{A,B})=e(P,SK_{A,B}+r_{A,B}W_{A,B})$$
$$=e(P,\sum_{(i,j)\in AE_B}(d_i+r_{i,j}W_{i,j})+d_A+r_{A,B}W_{A,B})$$
$$=e(Q,\sum_{i\in AN_B}(n_iQ_i)+Q_A)\prod_{(i,j)\in AE_B}e(R_{i,j},W_{i,j})$$

U_B의 프록시 서명키는 $SK_B=S_{A,B}+d_B$로 계산되어진다. 여기서 n_i는 $E_{i,j}\in PE_A$를 만족하는 j의 개수다.

3.1.3 최종 서명 생성 (U_R에 의한 서명 생성)

U_R은 다음과 같은 과정으로 위임 네트워크에 대한 최종 서명을 다음과 같이 생성한다.

1. U_R은 난수 r을 선택하고 $R=rP$를 계산한다.
2. U_R 은 주어진 메시지 m에 대해서 $M=H_2(\sum_{i\in PN_R} W_{i,R}\|ID_R\|m)$, $\sigma=SK_R+rM$을 계산한다.
3. U_R은 서명을 받는 대상에게 (σ,M_R,m,R_R,R)을 전송한다.

3.1.4 최종 서명 검증

서명은 다음의 검증식을 통과하면 합당한 것으로 받아들여 진다.

$$e(P,\sigma)=e(P,SK_R+rM)$$
$$=e(P,\sum_{(i,j)\in AE_R}(d_i+r_{i,j}W_{i,j})+d_R+rM)$$
$$=e(Q,\sum_{i\in AN_R}(n_iQ_i)+Q_R)e(R,M)\prod_{(i,j)\in AE_R}e(R_{i,j},W_{i,j})$$

n_i는 $E_{i,j}\in PE_R$를 만족하는 j의 개수이다.

3.2 제안하는 위임 네트워크의 안전성

정리 1. F를 위임 네트워크에 대한 $(\epsilon,q_{H_1},q_{H_2},q_S,q_E)$-위조자 라고 하자. 여기서 q_{H_1}, q_{H_2}, q_S, q_E 는 각각 ID에 대한 해쉬 질의의 개수, 메시지에 대한 해쉬 질의의 개수, 서명 질의의 개수 그리고 비밀키 생성 질의의 개수이다. 그러면 F를 통해서 $\epsilon'\approx\epsilon/e^2q_S$의 확률로 CDHP를 푸는 A를 구성할 수 있다.

증명) 의미 있는 위임 네트워크의 서명을 위조하는 공격자 F가 있다고 가정하자. 그러면 F의 공격능력을 사용하여 CDHP를 푸는 알고리즘 A를 구성할 수 있다. F를 통해서 A를 구성하는 방법으로 증명이 수행된다. (P,aP,bP)를 CDHP에 대한 문제로 주어진 순서쌍이라고 하자. A가 F의 공격능력을 활용하기 위해서는 공격환경을 구성해주어야 한다. 우선 A는 (P,aP)를 위임 네트워크의 시스템 변수에서 공개키로 설정한다. F의 질의를 일관성 있게 처리하고 중복을 피하기 위해 A는 두 개의 해쉬 리스트 H_1-list와 H_2-list를 관리한다. 두 리스트는 초기에 공집합으로 설정된다. A는 F의 질의를 다음과 같이 처리한다.

3.2.1 ID 해쉬 오라클에 대한 질의 (Hash Queries on Oracle for Identity)

- 동일한 질의에 대해서는 동일한 값으로 응답한다. A는 ID_i에 대한 질의에 다음과 같이 응답한다.
  1. ID_i가 이전에 질의된 것이면 A는 H_1-list에서 $(ID_i,k_i,Q_i,c_{1,i})$를 찾아서 Q_i를 반환한다.
  2. 이전에 질의된 값이 아니면 난수 $c_{1,i}\in\{0,1\}$를 선택한다. 이때 $\Pr[c_{1,i}=0]=\delta_1$이다. $c_{1,i}=0$이면 A는 난수 k_i를 생성하고 $Q_i=k_iP$를 계산한다.

$c_{1,i}=1$이면 난수 k_i를 생성하고 $Q_i=k_i(bP)$를 계산한다.

3. A는 새로 생성된 순서쌍 $(ID_i,k_i,Q_i,c_{1,i})$를 H_1-list에 추가하고 F에게 Q_i를 반환한다.

3.2.2 메시지 해쉬 오라클에 대한 질의 (Hash Queries on Oracle for Message)

- H_1-list와 마찬가지로 H_2-list를 관리한다. A는 (W_i,m_i)에 대한 F의 질의를 다음과 같이 처리한다. m_i는 위임을 하는 사용자와 위임 받는 사용자의 아이디 정보를 포함하고 있기 때문에 질의된 순서쌍에 대한 서명을 생성할 서명자의 아이디 ID_j를 예측할 수 있다.

1. ID_j가 기존의 H_1 해쉬 오라클에 대한 질의가 아니면 ID 해쉬 오라클에 대한 질의로 처리한다.
2. (W_i,m_i)가 이전에 질의된 것이면 H_2-list에서 순서쌍 $(W_i,m_i,l_i,M_i,c_{2,i})$을 복원한다.
3. 이전에 질의된 것이 아니면 난수 $c_{2,i}\in\{0,1\}$를 선택한다. 이때 $\Pr[c_{2,i}=0]=\delta_2$이다. $c_{2,i}=0$이면 난수 l_i를 생성하고 $M_i=l_iP$를 계산한다. $c_{2,i}=1$이면 난수 l_i를 생성하고 $M_i=l_i(bP)$를 계산한다.
4. A는 새로 생성된 순서쌍 $(W_i,m_i,l_i,M_i,c_{2,i})$를 H_2-list에 추가하고 F에게 M_i를 반환한다.

3.2.3 비밀키 생성 질의 (Private Key Extraction Queries)

- F가 ID_i에 대한 비밀키를 요구하면 A는 H_1-list에서 ID_i에 대응되는 순서쌍 $(ID_i,k_i,Q_i,c_{1,i})$를 복원한다. $c_{1,i}=0$이면 $d_i=aQ_i=a(k_iP)=k_i(aP)$를 비밀키로 반환하고 $c_{1,i}=1$이면 시뮬레이션을 중단한다.

3.2.4 서명 질의 (Signing Queries)

- F가 ID_i의 (W_j,m_j)에 대한 서명을 요구하면 A는 H_1-list와 H_1-list에서 ID_i와 (W_j,m_j)에 대응되는 순서쌍 $(ID_i,k_i,Q_i,c_{1,i})$와 $(W_j,m_j,l_j,M_j,c_{2,j})$를 복원한다. A는 F의 질의를 다음과 같이 처리한다.

1. $c_{1,i}=0$이면 A는 난수 r을 선택하고 ID_i의 (W_j,m_j)에 대한 서명을 (σ,R,M_j)로 생성하고 F에게 (σ,R,M_j)를 서명으로 반환한다. 이때 $\sigma=k_i(aP)+rM_j$이다.
2. $c_{1,i}=1$이고 $c_{2,i}=1$이면 A는 난수 r을 선택하고 $\sigma=rl_j(bP)$와 $R=rP-l_j^{-1}k_i(aP)$를 계산한다. A는 (σ,R,M_j)를 ID_i의 (W_j,m_j)에 대한 서명으로 반환한다.
3. 그 외의 경우에는 A는 시뮬레이션을 중단한다.

위에 구성된 것과 A가 수행하였을 때 성공적으로 F의 공격환경을 구성하는 확률을 계산해보자. 우선 A가 모든 비밀키 생성 질의와 서명 질의에 대해 성공적으로 응답할 확률은 각각 $\delta_1^{q_E}$와 $(1-(1-\delta_1)(1-\delta_2))^{q_S}$이다. 결과적으로 A가 공격 환경의 구성을 성공적으로 마칠 확률은 $\delta_1^{q_E}(1-(1-\delta_1)(1-\delta_2))^{q_S}$이다. CDHP문제를 풀기 위해서는 F가 생성한 유효한 위임 네트워크 서명에 비밀키 생성 질의가 수행되지 않은 아이디 ID_i의 서명 질의가 수행되지 않은 메시지 (W_j,m_j)에 대한 서명이 포함되어 있어야 하고, 이 경우 $coin_{2,j}=1$이어야 한다. F가 비밀키 생성 질의가 수행되지 않은 아이디 ID_i의 서명 질의가 수행되지 않은 메시지 (W_j,m_j)에 대한 서명을 포함하고 있는 위임 네트워크 서명을 생성할 확률이 ϵ라고 가정되어 있으므로 F가 $coin_{2,j}=1$인 조건까지 만족하는 서명을 포함하도록 위임 네트워크에 대한 서명을 생성할 확률은 $\epsilon(1-\delta_2)$이다. A가 CDHP를 풀기 위해서는 F의 공격환경을 성공적으로 구성해야 하고 F가 공격에 유용한 서명을 생성해야 한다. 그러므로 A는 ϵ'의 확률로 CDHP를 풀게 되고 $\epsilon'=\epsilon\delta_1^{q_E}(1-(1-\delta_1)(1-\delta_2))^{q_S}(1-\delta_2)$이다. $\delta_1=1-\frac{1}{q_E}$라고 하고 $\delta_2=1-\frac{1}{q_S}$라고 하면 ϵ'는 다음과 같이 표현된다:

$$\epsilon'=\epsilon(1-\frac{1}{q_E})^{q_E}(1-\frac{1}{q_Eq_S})^{q_S}\frac{1}{q_S}.$$

충분히 큰 x에 대해서 $(1-\frac{1}{x})^x\approx\frac{1}{e}$이므로 ϵ'은 다음과 같이 추정할 수 있다:

$$\epsilon'=\epsilon(1-\frac{1}{q_E})^{q_E}(1-\frac{1}{q_Eq_S})^{q_S}\frac{1}{q_S}\geq$$

$$\epsilon(1-\frac{1}{q_E})^{q_E}(1-\frac{1}{q_Eq_S})^{q_Eq_S}\frac{1}{q_S}\approx\epsilon\frac{1}{e^2q_S}$$

F가 CDHP를 풀 수 있는 형태의 위임 네트워크의 서명을 생성할 확률에 대하여 살펴보았다. ϵ'의 확률로 생성된 위임 네트워크의 서명이 CDHP를 풀 수 있는 형태로 생성되었을 때 CDHP를 푸는 과정을 살펴보자. F가

위임 네트워크의 서명으로 σ를 생성했다고 하자. ID_{i^*}에 대한 비밀키 생성 질의가 수행되지 않았다고 가정하고 ID_{i^*}의 (W_{j^*}, m_{j^*})에 대한 서명이 서명 질의로 수행되지 않았으며 $coin_{2,j^*}=1$이라고 하자. (이런 경우에 대한 위임 네트워크의 서명을 생성할 확률은 위에서 계산되어 있다.) 아래 식에서 $AN_F^* = AN_F/\{i^*\}$이라고 하자. σ는 검증식을 통과하는 서명이기 때문에 다음과 같은 식을 얻을 수 있다:

$$e(P,\sigma) = e(P, SK_F + r_F M)$$
$$= e(P, \sum_{(i,j)\in AE_F}(d_i + r_{i,j}W_{i,j}) + d_F + r_F M)$$
$$= e(Q, \sum_{i\in AN_F}(n_i Q_i) + Q_F)\, e(R_F, M) \prod_{(i,j)\in AE_F}(R_{i,j}, W_{i,j})$$
$$= e(P, a(\sum_{i\in AN_F^*}(n_i k_i) + k_{i^*} b + k_F)P)\, e(l_F R_F + \sum_{(i,j)\in AE_F}(l_{i,j}R_{i,j}), P)$$
$$= e(P, a(\sum_{i\in AN_F^*}(n_i k_i) + k_{i^*} b + k_F)P + l_F R_F + \sum_{(i,j)\in AE_F}(l_{i,j}R_{i,j}))$$

그러면 σ는 다음과 같이 표현할 수 있다.

$$\sigma = a(\sum_{i\in AN_F^*}(n_i k_i) + k_{i^*} b + k_F)P + l_F R_F + \sum_{(i,j)\in AE_F}(l_{i,j}R_{i,j})$$

모든 k, l, n은 A가 F의 공격 환경을 구성하면서 생성한 값이므로 알고 있는 값이고, 결과적으로 다음과 같은 식을 통해서 abP를 계산할 수 있다.

$$abP = k_{i^*}^{-1}(\sigma - a(\sum_{i\in AN_F^*}(n_i k_i) + k_F)P - l_F R_F - \sum_{(i,j)\in AE_F}(l_{i,j}R_{i,j}))$$

3.3 제안하는 위임 네트워크의 변형

[2]에서는 [1]에서 제안된 효율적인 집합 서명(aggregate signature) 기법 등을 사용해서 위임 네트워크의 구성에서 저장 공간을 줄이거나 통신량을 줄이는 방법을 구성하는 문제를 남겨놓았다. 본 논문에서 제안하는 위임 네트워크에서 사용자 U_A는 U_B에게 서명권한을 위임하는 과정에서 난수 $r_{A,B}$를 생성하고 $R_{A,B} = r_{A,B}P$를 계산하여 사용하였다. [1]에서 제안된 집합 서명의 기법을 도입하면 다음과 같이 효율적으로 저장 공간과 통신량을 줄일 수 있다. U_A는 U_B에게 권한을 위임하는 과정에서 난수 $r_{A,B}$선택하는 것이 아니라 고정된 자신의 비밀키 r_A를 사용한다. r_A에 대응되는 공개키는 $R_A = r_A P$라고 하자. 본 논문에서 제안된 위임 네트워크에서는 각 사용자가 선택해서 생성한 난수 $R_{i,j}$의 집합을 서명으로 포함해야 했다. 그런데 각 사용자가 자신의 비밀키로 난수 사용하는 것을 대신하면 $R_{i,j}$의 집합이 서명에서 제외될 수 있으므로 $R_{i,j}$의 집합의 크기에 해당하는 저장 공간과 통신량을 줄일 수 있다. 서명에 포함되어야 하는 정보 중에서 위임의 조건을 포함하고 있는 메시지는 제외할 수 없기 때문에 $R_{i,j}$의 집합을 서명에서 제외하는 것은 저장 공간과 통신량의 측면에서 최적화된 상태이다. 그러나 이와 같은 구성은 각 사용자의 공개키를 인증하는 비용을 추가로 요구하기 때문에 효율적이지 않다.

[1]의 구성을 직접 사용하지는 않지만 설정을 제안하는 위임 네트워크에 도입함으로써 효율성의 증대를 기대할 수 있다. U_A는 U_B에게 권한을 위임하는 과정에서 난수 $r_{A,B}$선택하지 않고 고정된 비밀 정보 r_A을 사용한다. 즉, 각 사용자는 자신의 비밀 값을 하나씩 가지고 서명권한을 위임하는 과정에서 사용한다. 고정된 비밀 값을 사용하더라도, 각 사용자가 서명권한의 위임과정에서 ID키 $d_i = sQ_i$를 사용하기 때문에 서명은 인증 기능을 여전히 제공한다. 그러나 고정된 난수를 사용하는 경우에는 동일한 평문에 대한 서명이 동일하게 생성되는 단점이 있다. 이것은 Boneh 등이 제안한 집합 서명[1]에서도 가지고 있는 문제점이다. 그런데 위임 네트워크에서 서명되는 메시지는 권한의 위임 기간 등의 정보가 포함되어 있기 때문에 동일한 메시지에 대한 서명이 수행되는 경우를 배제할 수 있고, 이는 결정적인 서명을 사용하는 것이 문제를 야기하지 않는 다는 것을 의미한다. 이와 같이 변형한 위임 네트워크의 안전성은 정리 1과 [1]에서 사용된 증명 기법의 도입을 통해 쉽게 보일 수 있다. 증명은 거의 동일한 형태로 수행되므로 생략하도록 한다. 이처럼 각 사용자가 고정된 비밀 값과 ID키를 사용하여 서명권한의 위임을 수행하도록 위임 네트워크를 구성하면 $R_{i,j}$의 집합의 크기는 네트워크의 에지의 개수인 E에서 노드의 개수인 N로 줄어든다. 앞에 언급된 것처럼 일반적으로 E가 N보다 크기 때문에 해당되는 크기의 저장 공간 및 전송량의 효율성 증대를 기대할 수 있다. 또한 각 사용자가 생성한 서명이 동일한 난수로 계산되기 때문에 한 번에 검증 식에서 계산할 수 있게 되므로 제안하는 위임 네트워크에서 E번의 페어링 연산이 요구되었던 것과 달리 제안하는 것의 변형된 위임 네트워크는 N번의 페어링 연산으로 동일한 위임 네트워크를 구성할 수 있다.

〔표 1〕 기존의 위임 네트워크와 제안하는 위임 네트워크 그리고 그것의 변형된 위임 네트워크의 효율성 비교

	위임된 서명키 검증	프록시 서명키 생성	최종 서명 검증	서명의 길이
기존 결과	$(E_A+2)P+N_AM$	$3M$	$(E_F+2)P+N_FM$	$(E_A+1)Pt+N_FWt$
위임 네트워크 1	$(E_A+2)P$	$2M$	$(E_F+2)P$	$(E_A+1)Pt+N_FWt$
위임 네트워크 2	$(N_A+2)P$	$1M$	$(N_F+2)P$	$(N_A+1)Pt+N_FWt$

3.4 효율성

기존의 위임 네트워크〔2〕와 제안하는 위임 네트워크 및 그것의 변형들 간의 효율성 비교는 〔표 1〕에서 확인할 수 있다. 타원곡선에서의 페어링 연산과 스칼라 곱셈 연산을 각각 P, M 이라고 하자. 타원곡선 포인트의 비트크기와 보증서(warrant)의 비트사이즈를 각각 Pt와 Wt라고 하자. 사용자 U_A의 모든 상위 에지의 개수와 노드의 개수를 각각 E_A와 N_A라고 하자. 그러면 〔표 1〕과 같은 결과를 확인할 수 있다.

〔표 1〕에서 위임 네트워크 1(DN1) 은 본 논문에서 제안하는 위임 네트워크를 의미하고 위임 네트워크 2(DN2)는 위임 네트워크 1을 변형함으로써 효율성을 개선한 것을 의미한다. 표에서 확인할 수 있듯이, 본 논문에서 제안하는 두 위임 네트워크는 기존의 것보다 효율적이다. DN1의 경우는 모든 연산 과정에서 기존 위임 네트워크보다 효율적이다. DN2는 연산 측면에서도 가장 효율적이고 서명의 길이가 짧기 때문에 전송량과 저장량의 측면에서도 효율적이다.

Ⅳ. 결 론

본 논문에서는 ID 기반의 효율적인 위임 네트워크를 제안하였다. 제안한 위임 네트워크는 기존의 결과보다 연산량의 측면에서 효율적이고, 제안한 것의 변형된 형태의 경우에는 연산량 뿐만 아니라 저장량, 통신량의 측면에서도 효율적이다. 특히, 〔2〕에서 〔1〕의 결과를 사용하여 효율성을 높이는 문제를 남겨놓았는데, 본 논문에서는 〔1〕의 결과를 부분적으로 도입하여 기존의 결과보다 효율적인 구성을 제안하였다.

참고문헌

〔1〕 Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings, volume 2656 of Lecture Notes in Computer Science, pages 416-432. Springer.

〔2〕 Sherman S.M. Chow, Richard W.C. Lui, Lucas C.K. Hui, and S.M. Yiu, *Identity Based Delegation Network*, Mycrypt 2005, LNCS 3715, pp.99-115, 2005.

〔3〕 Tuomas Aura, *On the Structure of Delegation Networks*, In PCSFW: Proceedings of the Eleventh Computer Security Foundations Workshop. IEEE Computer Society Press, 1998.

〔4〕 Shin-Jia Hwang, and Chiu-Chin Chen, *New Multi-Proxy Multi-Signature Schemes*, Applied Mathematics and Computation, 147(1):57-67, 2004.

〔5〕 Min-Shiang Hwang, Eric Jui-Lin Lu, and Iuon-Chang Lin, *A Practical (t,n)-Threshold Proxy Signature Scheme Based on the RSA Cryptosystem*, IEEE Transactions on Knowledge and Data Engineering, VOL.15, NO.6, November/December 2003.

〔6〕 Wen-Chung Kuo, and Ming-Yang Chen, *A Modified (t, n)-Threshold Proxy Signature Scheme Based on the RSA Cryptosystem*, Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05), 2005.

〔7〕 Xiangxue Li, Kefei Chen, Longjun Zhang, and

Shiqun Li, *Proxy Structured Multisignature Scheme from Bilinear Pairings*, ISPA 2004, LNCS 3358, pp. 705-714, 2004.

[8] M. Mambo, K. Usuda, and E. Okamoto, *Proxy Signatures for Delegating Signing Operation*, Proc. 3rd ACM Conference on Computer and Communications Security, 1996.

[9] So-Young Park, and Sang-Ho Lee, *Multi-proxy Signatures Based on Diffie-Hellman Problems Allowing Repeated Delegations*, HSI 2005, LNCS 3597, pp. 340-344, 2005.

[10] H.-M.Sun, N.-Y.Lee, and T.Hwang, *Threshold proxy signatures*, IEE Proc. Computers and Digital Techniques, Vol.146, No.5, September, 1999.

[11] Guilin Wang, Feng Bao, Jianying Zhou, and Robert H. Deng, Comments on "*A Practical (t,n) Threshold Proxy Signature Scheme Based on the RSA Cryptosystem*", IEEE Transactions on Knowledge and Data Engineering, VOL. 16, NO. 10, OCTOBER 2004.

[12] Qingshui Xue, and Zhenfu Cao, *A Threshold Proxy Signature Scheme Using Self-Certified Public Keys*, ISPA 2004, LNCS 3358, pp.715-724, 2004.

[13] Q. Xue and Z. Cao, *Improved of Multi-proxy Signature Scheme*, Proceeding of International Symposium on Communications and Information Technologies, pp.450-455, 2004.

[14] Jing Xu, Zhenfeng Zhang, and Dengguo Feng, *ID-Based Proxy Signature Using Bilinear Pairings*, Cryptology ePrint Archive, Report 2004/206, 2004. Available at http://eprint.iacr.org.

[15] Lijang Yi, Guoqiaiig Bai, and Guozhen Xiao, *Proxy multi-signature scheme: A new type of proxy signature scheme*, Electronics Letters, 16th, March 2000, Vol.36, No.6, 2000.

〈著者紹介〉

윤 택 영 (Taek-Young Youn) 정회원
2003년 2월: 고려대학교 수학과 이학학사
2005년 2월: 고려대학교 정보보호대학원 정보보호학과 공학석사
2005년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 박사과정
<관심분야> 암호 이론, 정보보호 이론, 암호 프로토콜, 부채널 공격

박 영 호 (Young-Ho Park) 정회원
1990년 2월: 고려대학교 수학과 이학사
1993년 2월: 고려대학교 수학과 이학석사
1997년 2월: 고려대학교 수학과 이학박사
2002년 3월~현재: 세종 사이버 대학교 부교수
<관심분야> 정수론, 공개키 암호, 암호 프로토콜, 부채널 공격

정 상 태 (Sangtae Jeong)
1989년 2월: 고려대학교 수학과 이학사
1991년 2월: 고려대학교 수학과 이학석사
1999년 : University of Texas at Austin 이학박사
2002년 9월~현재: 인하대학교 수학과 조교수
<관심분야> 정수론, 공개키 암호