익명성 메커니즘: 그룹 서명과 추적 가능한 서명

서울대학교 최승걸 · 박근수*

1. 서 론

우리는 대부분 어떤 시설이나 서비스를 정기적으로 이용하고 있다. 주차장이나 헬스클럽을 이용하는 것이 그 예가 될 수 있을 것이다. 어떤 고객이 그러한 서비스를 이용하기 위해서는, 서비스를 사용할 수 있는 자격이 있음을 서비스 제공자에게 인증을 받는 절차가 필요하다. 그럼 이제 이 인증 시스템이 자동화되어 있다고 가정해 보자. 어떤 인증 방법을 사용하는 것이 좋을까? 아마도 서비스 제공자가 고객에게 인증 카드를 발급하면 될 것이다. 인증카드는 크게 두 가지 종류로 구분될 수 있을 것이다.

- 고객의 신원이 포함된 인증카드
- 고객의 신원이 포함되지 않은 인증카드

고객의 신원이 포함된 인증카드의 경우에는, 인증이 일어날 때 서비스 제공자가 고객의 신원을 파악할 수 있다. 하지만 서비스를 이용할 때 반드시 신원을 알려줘야 하는가? 이는 사생활 보호의 관점에서 그리 바람직하지 못하다.

고객의 신원이 포함되지 않은 인증카드의 경우에는 사생활 보호의 입장에서 앞의 경우보다 바람직하다고 할 수 있다. 그런데, 고객의 신원을 반드시 알아야 할 때가 때때로 발생하기도 한다. 만약 무인 주차장에서 살인 사건이 발생했다면, 주차장에 누가 있었는지를 아는 것이 사생활 보호보다 더 중요한 일이 된다.

기존의 인증 방법은 고객의 신원이 항상 노출되거나, 항상 노출되지 않는 양 극단 중에 하나를 택함으로서, 다양한 경우에 유연하게 대처하지 못하는 경직성을 가지고 있다. 본고에서는 보다 유연한 2-단계 인증 방법을 제공할 수 있는 그룹 서명과 추적 가능한 서명에 대해서 설명한다. 또한 이 서명 방법이 사용될 수 있는 다양한 응용에 대해서도 알아보도록 하겠다.

2. 그룹 서명

그룹 서명은 Chaum과 Van Heyst[14]가 제안한 서명 방법으로 다음과 같은 특징을 가지고 있다.

- 그룹의 멤버만이 서명을 할 수 있다.
- 서명을 받은 사람은 서명을 통해 서명자가 그룹의 멤버라는 사실은 알 수 있으나, 정확히 누구인지는 알 수 없다.
- 특별한 경우에, 그룹 관리자는 서명을 개봉해서 누가 서명을 했는지 찾아낼 수 있다.

위의 특징을 살펴보면 서론의 예에서 제기된 주차장 문제를 그룹 서명을 통해서 보다 유연하게 해결할 수 있음을 알 수 있다.

그럼 이제 그룹 서명이 어떻게 구성되는지 살펴보도록 한다. 본고에서는 Bellare, Shi, Zhang[7]의 모델에 따라 설명하겠다.

2.1 구성자

그룹 서명은 다음과 같은 요소들로 구성된다.

- 멤버(member): 그룹에 가입하여 발급자를 통해 서명키를 얻는다. 서명키를 이용하여 그룹 서명을 할 수 있다.
- 발급자(issuer): 사용자가 그룹에 가입할 때 서명키를 획득할 수 있도록 해준다. 발급자는 서명키의 일부 정보를 알 수 있다. 이때 얻어낸 일부 정보는 등록 테이블에 저장된다.
- 개봉자(opener): 분쟁이 발생했을 경우, 해당 서명을 개봉해서 서명자를 찾아낼 수 있다. 발급자와 개봉자를 합쳐서 그룹 관리자라고 통칭하기도 한다.
- 등록 테이블(reg): 사용자가 가입할 때, 발급자는 등록 테이블에 그 사용자의 서명키의 일부 정보를 이 등록 테이블에 저장한다. 개봉자가 추후에 서명을 개봉하여 서명자를 찾아낼 때 이 테이블을 참고한다.

* 종신회원

2.2 키

그룹 서명은 다음과 같은 키들로 이루어져 있다.

- 그룹 공개키(gpk): 이 키는 그룹 서명의 구성원들과 구성원 이외에 서명을 받은 사람들 모두에게 공개된 키이다. 특히, 서명을 받은 사람들이 이 공개키를 이용해서 서명의 적법성을 판단하게 된다.
- 발급자 비밀키(ik): 발급자는 발급자 비밀키를 이용해서 사용자가 가입했을 때, 그 사용자가 서명키를 획득할 수 있도록 도와준다.
- 개봉자 비밀키(ok): 개봉자는 개봉자 비밀키를 이용해서, 분쟁이 생긴 그룹 서명에 대해 그 서명의 주인이 누구인지 밝혀낼 수 있다.
- 사용자 공개키(upk[i]): 각 멤버는 공개키와 비밀키 쌍을 만든 다음, 자신의 공개키는 공개한다.
- 사용자 비밀키(usk[i]): 각 멤버는 공개키와 비밀키 쌍을 만든 다음, 자신의 비밀키는 보관한다.
- 서명키(gsk[i]): 멤버가 발급자를 통해 얻는 키이다. 그룹의 멤버는 이 키를 이용하여 그룹 서명을 할 수 있다.

2.3 연산

그룹 서명이 지원하는 연산은 아래와 같다.

- 그룹키 생성(GKg): 신뢰할 수 있는 기관이 이 연산을 수행한다. 이 연산을 통해 그룹 공개키(gpk), 발급자 비밀키(ik), 개봉자 비밀키(ok)가 만들어진다.
- 사용자키 생성(UKg): 각 사용자는 이 연산을 수행하여 공개키와 비밀키 쌍(upk[i], usk[i])을 만들어 낼 수 있다.
- 가입, 발급 (Join, Iss): 멤버는 가입연산을 수행하고, 키 발급자는 발급연산을 수행한다. 이 연산이 수행되면 멤버는 서명키를 획득하게 되며, 등록 테이블에 서명키의 일부가 저장된다.
- 그룹 서명(GSig): 그룹 멤버 i는 자신의 서명키(gsk[i])를 이용하여 이 연산을 수행함으로써 그룹 서명을 만들어낸다.
- 그룹 서명 검증(GVf): 그룹 서명을 받은 사람은 그룹 공개키(gpk)를 가지고 이 연산을 수행하여 받은 서명이 적법한지를 판단하게 된다.
- 개봉(Open): 개봉자만이 이 연산을 수행할 수 있다. 개봉자는 비밀키(ok)와 등록 테이블(reg)을 이용하여 이 연산을 수행하며, 주어진 서명의 서명자를 찾아낼 수 있다. 이 연산의 수행결과는 (j, τ) 이다. j는 그룹 멤버를 나타내는 인덱스이며, τ는 서명자가 j인 근거이다. τ는 개봉자가 거짓말을 할 수 없도록 하는 역할을 한다.
- 심의(Judge): 개봉연산을 통해서 나온 결과 (j, τ)를 보고 개봉자가 찾아낸 멤버 j가 정말 서명의 주인이 맞는지 확인하는 연산이다. 그룹 공개키(gpk)와 멤버 j의 사용자 공개키(upk[j])를 참조하여 연산이 수행된다.

3. 그룹 서명의 응용

3.1 조직 내의 구성원에 대한 익명성

그룹이 조직되고, 그룹 멤버가 조직 내의 구성원이 되는 그룹 서명을 생각해 볼 수 있다. 이 경우 그룹 멤버는 그룹 서명을 통해서 자신이 어떤 조직의 일원임을 증명할 수 있다. 한편, 서명을 받은 사람은 그 서명을 통해 조직내 구성원이 서명했다는 사실만을 알 뿐, 실제 누가 서명을 했는지는 확인할 수 없다. 이와 같은 조직 내 구성원에 대한 익명성을 이용해서 다음과 같은 응용을 생각해 볼 수 있다[4,10].

- 익명 계약: 계약서에 서명할 때 익명으로 서명할 수 있다. 이를 통해 조직원들의 자율성을 제고할 수 있을 것이다. 물론 문제가 생길 경우에는 그 서명을 개봉해볼 수 있기 때문에 책임소재의 문제도 해결할 수 있다.
- 정부, 군 보도자료: 익명의 제보나 보도 자료는 신뢰할 수 없는 경우가 많다. 하지만 그룹 서명을 이용하면 보도 자료의 신뢰성을 높일 수 있다.

익명 계약의 시나리오가 구체적으로 그룹 서명에 어떻게 적용되는지 알아보자. 먼저, 신뢰할 수 있는 조직 내의 기관(예: 조직 내 심의 기관)이 그룹키 생성 연산을 수행하여 그룹 공개키, 발급자 비밀키, 개봉자 비밀키를 생성한 후, 발급자를 담당하는 기관과 개봉자를 담당하는 기관에게 각각 발급자 비밀키와 개봉자 비밀키를 분배한다. 조직에 새로운 직원이 들어오면, 신입 직원과 발급자 기관은 가입/발급(Join/Iss) 연산을 수행한다. 그 결과 신입 직원은 서명키를 얻을 수 있으며, 발급자 기관은 등록 테이블에 서명키의 일부 정보를 저장하게 된다. 조직 내의 직원이 익명으로 다른 조직과 계약을 맺고 싶을 경우에는 그룹 서명(GSig) 연산을 통해 서명을 할 수 있다. 타 조직에서는 계약의 적법성을 그룹 서명검증(GVf) 연산으로 확인할 수 있을 것이다. 만약 조직내 구성원이 불법적인 계약을 해서 문제가 발생할 경우에는 개봉자 기관이 등록 테이블과 개봉자 비밀키를 이용하여 개봉(Open)연산을 수행할 것이다. 그 결과 계약을 행한 직원이 누구인지 밝혀낼 것이고(밝혀진 직원을 j라고 하자), 조직 내의 심의 기관에서는 서명자가

j라고 개봉자가 제시한 근거 τ를 가지고 심의(Judge) 연산을 수행하여, j가 정말 서명자인지 확인한 다음 적절한 조치를 취할 수 있을 것이다.

3.2 Identity Escrow

Identity escrow[21]는 Kilian과 Petrank가 제안한 시스템으로, 익명으로 인증을 하고 문제가 되는 행동을 했을 경우에 신원을 밝혀내는 시스템을 말한다. 서론에서 소개했던 예가 identity escrow의 대표적인 경우라고 할 수 있겠다. 이 시스템은 인증 시에 인터랙션이 일어난다는 점 이외에는 그룹 서명과 매우 유사하다. 이 시스템은 그룹 서명을 이용하여 구현할 수 있다[21].

이 외에도 인터넷 서비스를 이용한다든지, 정기권을 가지고 요금소를 통과하는 등의 예를 생각해 볼 수 있다. 또한, 익명 컴퓨터 채팅 서비스도 가능하다. 익명으로 채팅 방에 들어갈 수 있고(익명 인증), 이 익명성을 가지고 마음껏 채팅을 할 수 있지만, 그 방에서 문제가 있는 행동(예: 마약거래)을 하는 사람의 신원을 밝혀내서 탈퇴조치를 시킬 수 있을 것이다.

3.3 입찰

그룹 서명의 익명성은 입찰 과정[2]에도 이용될 수 있을 것이다. 입찰에 참여하는 회사들이 하나의 그룹을 이룬다. 각 회사는 입찰 가격을 그룹 서명으로 서명하여 제출한다. 낙찰을 결정하는 기관에서는, 그룹 서명의 익명성 때문에 제시된 입찰 가격만을 알 수 있을 뿐, 어떤 회사가 어떤 입찰 가격을 제출했는지 알 수가 없다. 따라서 입찰 가격을 가지고 공정하게 낙찰 여부를 판단하게 된다. 또한 낙찰이 결정된 후에도, 서명을 개봉하여 낙찰된 회사만 알 수 있기 때문에 다른 회사가 어떤 가격을 제시했는지는 알 수가 없다.

3.4 자동차 안전 통신 시스템

자동차에 단거리 발신기를 장착하여 근거리에 있는 자동차끼리 서로 통신할 수 있도록 하는 시스템을 생각해 볼 수 있다[9]. 자동차에 장착된 발신기는 자동차의 상태 정보를 주위의 차들에게 발신한다. 예를 들어, 자동차가 급제동을 하는 경우에, 이 사실이 발신기를 통해 주위의 모든 차에게 알려진다. 이런 시스템이 도입되면 자동차 사고를 줄이는데 많은 도움이 될 수 있을 것이다.

문제는 자동차가 아닌 다른 장소에서 거짓 상태 정보를 발신함으로써 통신 시스템의 혼란을 꾀하는 악의적인 사람이다. 이를 막기 위해서는 자동차 안에, 복제할 수 없는 칩을 집어넣어서, 발신기의 상태 정보 메시지가 이 칩을 통해 디지털 서명이 되도록 하면 될 것이다. 하지만 단순히 디지털 서명을 하는 경우에는 자동차 번호가 노출될 수 있다. 발신기가 내보내는 메시지는 자동차의 상태나 위치 등의 내용이 담겨 있기 때문에 결국 누가 어디에 있는지에 대한 정보가 노출되는 결과를 낳는다. 이는 사생활 침해의 관점에서 바람직하지 못하다. 더군다나 모든 자동차에 대한 공개키를 저장할 공간도 없다. 그룹 서명을 이용하면 이런 문제를 해결할 수 있을 것이다.

3.5 증명서 전달 시스템(Credential Transfer System)

이 시스템[11,13]에서는 여러 서비스 제공자가 존재하고, 각 서비스 제공자는 사용자를 가명(pseudonym)으로만 알고 있다. 동일한 사용자의 여러 가명은 서로 연관되지 않는다. 서비스 제공자는 증명서(credential)를 발행할 수 있고, 이 증명서는 다른 서비스 제공자가 참고하여 서비스 제공 여부를 판단할 수 있다. 시스템에 따라, 사용자는 받은 증명서를 한번 또는 여러 번 사용할 수 있다. 서비스 제공자들이 연계해서 사용자의 이용 패턴 등의 프로파일을 수집할 수 없도록 사용자는 서비스 제공자에 접속할 때마다 각기 다른 가명을 가지고 접속한다. 그런 다음, 증명서를 가지고 있다는 사실만을 그 서비스 제공자에게 알려줌으로써 (증명서를 보여주지는 않는다) 해당 서비스를 사용한다.

이런 시스템은 투표권을 행사한다거나, 의료 서비스를 받는다거나, 자동차를 빌리거나 하는 등의 용도에 적용될 수 있을 것이다. 이 시스템은 그룹 서명으로 구현될 수 있음이 알려져 있다[11].

3.6 기타

그룹 서명의 익명성은 여러 곳에서 이용될 수 있는 매우 매력적인 특징이다. 따라서 앞에서 든 예들 이외에도 많은 응용을 생각해 볼 수 있다.

투표 시스템에서 그룹 서명의 익명성을 이용하여 투표자의 신원을 숨길 수 있을 것이다. 전자 화폐 시스템에서 그룹 서명을 이용하여 화폐를 발급한 은행이나 전자 화폐를 소비하는 소비자의 신원을 숨길 수 있을 것이다[18]. 또한 신뢰 가능한 컴퓨팅[27]에서도 그룹 서명의 쓰임새를 찾을 수 있다. 신뢰 가능한 컴퓨팅은 remote attestation이라는 기능을 가지고 있는데, 이 기능은 데스크톱 PC가 원격의 상대방에게 어떤 소프트웨어를 수행하고 있는지 증명하는 기능이다. 이 때 익명성을 보장하기 위해서 그룹 서명을 사용할 수 있다[17 section 2.2].

4. 추적 가능한 서명

4.1 배경

추적 가능한 서명[19]이 최근에 Kiayias, Tsiounis, Yung에 의해 제안되었다. 이 서명 방법은 그룹 서명에 연산을 추가함으로써, 더 많은 익명성을 얻고자 한다. 다음과 같은 익명 트랜잭션 시스템을 생각해 보자.

- 그림 1과 같이 서비스를 제공하는 기관이 여러 곳으로 나뉘어 있고, 각 트랜잭션 내역은 해당 서비스 기관에 로그 형태로 남게 된다. 사용자는 그룹 서명을 이용하여 익명 트랜잭션을 수행할 수 있다. 서비스 기관에서는 그룹 서명의 익명성 때문에 어떤 사용자가 서비스를 받고 있는지 알지 못한다.

위와 같은 익명 트랜잭션 시스템에서, 법적으로 문제가 있는 트랙잭션이 발생해서 (그림 1의 경우 서비스 기관 2에서 발생), 그 트랜잭션을 요구한 악의적인 사용자를 찾아할 일이 생겼다. 이 경우 개봉자 기관에서 트랜잭션에 담긴 서명에 대해 개봉 연산을 수행하여 악의적인 사용자를 찾아낼 수 있을 것이다. 여기서 더 나아가서, 악의적인 멤버의 모든 트랜잭션을 무효화시킬 필요가 있는 경우를 생각해볼 수 있을 것이다.

이 경우 그룹 서명이 제시해 줄 수 있는 답은 그리 만족스럽지 않다. 아마도 모든 트랜잭션에 담겨있는 서명에 대해서 개봉 연산을 수행하는 것이 유일한 해결책일 것이다. 하지만, 그렇게 되면 두 가지 문제점이 발생한다. 첫째로 각 트랜잭션에 대해 개봉 연산을 하게 되면, 그 결과로 각 트랜잭션을 요구한 사용자의 신원이 밝혀진다. 이는 그룹 서명의 근본적인 취지인 사생활 보호와 익명성을 완전히 파괴하게 되는 셈이다. 둘째로, 설사 익명성의 파괴를 감수하고 모든 트랜잭션에 대해 개봉 연산을 수행을 한다 하더라도 상당한 시간이 소요된다. 그룹 멤버의 수를 n이라고 하면, 개봉 연산은 등록테이블(reg)을 차례로 뒤져서 서명자를 찾아야 하기 때문에 O(n)의 시간이 걸린다. 따라서 트랜잭션의 개수가 m일 경우 모든 트랜잭션에 대해 개봉연산을 수행하는 데는 O(mn)의 시간이 걸린다. 이는 n과 m에 비례하여 시간이 걸리기 때문에 규모가변성(scalability)이 떨어진다. 이와 같은 문제를 해결하기 위해서 추적 가능한 서명이 제안되었다. 그렇다면, 추적 가능한 서명이 이 문제를 어떻게 해결할 수 있는지 살펴보자. 먼저 추적 가능한 서명의 구성자와 연산에 대해서 알아보도록 한다. 키의 종류는 그룹 서명과 동일하므로 생략한다.

4.2 구성자

추적 가능한 서명은 멤버, 발급자, 개봉자, 등록테이블, 추적자(tracer)로 구성된다. 즉, 그룹 서명에다가 추적자가 추가된 형태이다. 추적자는 다수가 존재할 수 있다. 각 추적자들은 트랜잭션에 담겨진 서명에 대해, 그 서명이 특정 멤버의 서명인지 확인하는 일을 한다.

4.3 연산

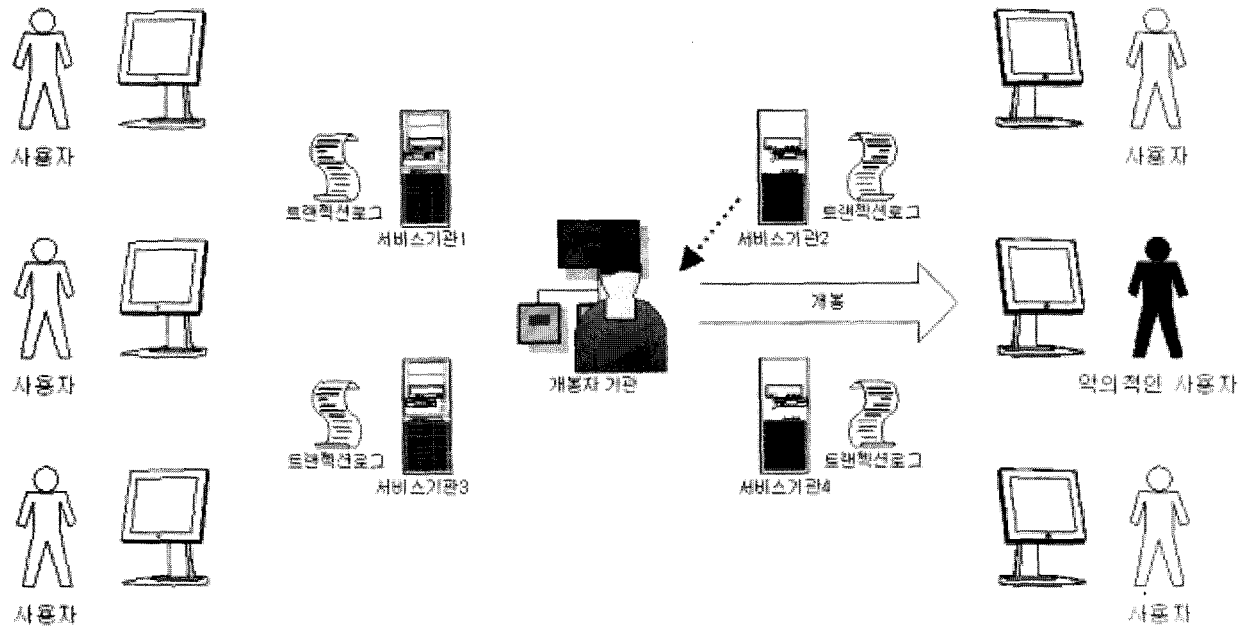


그림 1 익명 트랜잭션 시스템: 사용자들은 그룹 서명으로 트랜잭션 서비스를 받을 권리가 있음을 증명하기 때문에, 서비스 기관에서는 정확히 어떤 사용자가 트랜잭션을 요구하는지 알지 못한다. 악의적인 사용자가 익명성을 무기로 문제가 있는 트랜잭션을 요구할 경우, (이 그림에서는 서비스기관 2에서 발생) 이 트랜잭션에 대해 개봉자 기관이 개봉 연산을 수행하여 악의적인 사용자를 찾아낸다.

추적 가능한 서명은 2.3에 제시된 그룹 서명의 연산들과 그에 더하여 아래의 연산을 제공한다.

- 노출(Reveal): 개봉자가 분쟁이 생긴 서명을 개봉하여 악의적인 멤버를 찾아낸 후, 이 연산을 수행한다. 이 연산은 등록 테이블을 참고하여 해당 멤버의 서명키의 일부를 리턴한다. 이 정보는 다수의 추적자들에게 전달된다.
- 추적(Trace): 다수의 추적자들은 자신이 맡은 트랜잭션들에 대해 이 연산을 수행한다. 이 연산은 주어진 서명이 해당 멤버가 서명한 것인지 여부를 리턴한다.
- 소유권 주장(Claim): 서명자는 자신이 만든 서명에 대해 소유권을 주장할 때 이 연산을 수행한다. 이 연산은 해당 서명의 소유권을 가지고 있음을 주장하는 근거를 리턴한다.
- 소유권 주장 검증(Claim_Verify): 서명과 소유권 주장의 결과를 입력으로 받아서 정말 해당 멤버가 서명을 만들었는지 검증할 때 이 연산을 사용한다.

5. 추적 가능한 서명의 응용

5.1 익명 트랜잭션 시스템

4.1에서 소개했던 시나리오를 추적 가능한 서명을 이용하여 해결할 수 있다. 먼저 악의적인 멤버의 신원을 알아낸다. 여기까지는 그림 1과 동일하게 동작한다. 그런 다음, 그림 2와 같이 개봉자 기관이 노출 연산을 수행하여 등록 테이블에서 서명키 정보의 일부를 얻어낸다. 이 정보는 다수의 추적자들에게 전달되며, 다수의 추적자들은 자신들이 맡은 트랜잭션들에 대해 추적 연산을 수행하여, 악의적인 멤버가 수행한 트랜잭션을 찾아낸다.

주목할 점은 이렇게 하면 앞에서 지적했던 문제들이 발생하지 않는다는 점이다. 먼저 익명성이 보존된다. 모든 트랜잭션에 대해서 개봉 연산을 하는 경우에는 그 트랜잭션을 어떤 사용자가 요구했는지가 알려지는데 반해, 각 트랜잭션에 대해 추적 연산을 수행하면 그 트랜잭션이 악의적인 멤버의 것인지 아닌지만 알 수 있다. 따라서 추적 연산 결과 어떤 트랜잭션이 악의적인 멤버가 요구한 것이 아닐 경우에는, 그 트랜잭션이 누구의 것인지 알 수 없다.

또한 추적 연산을 이용함으로써 규모가변성도 얻을

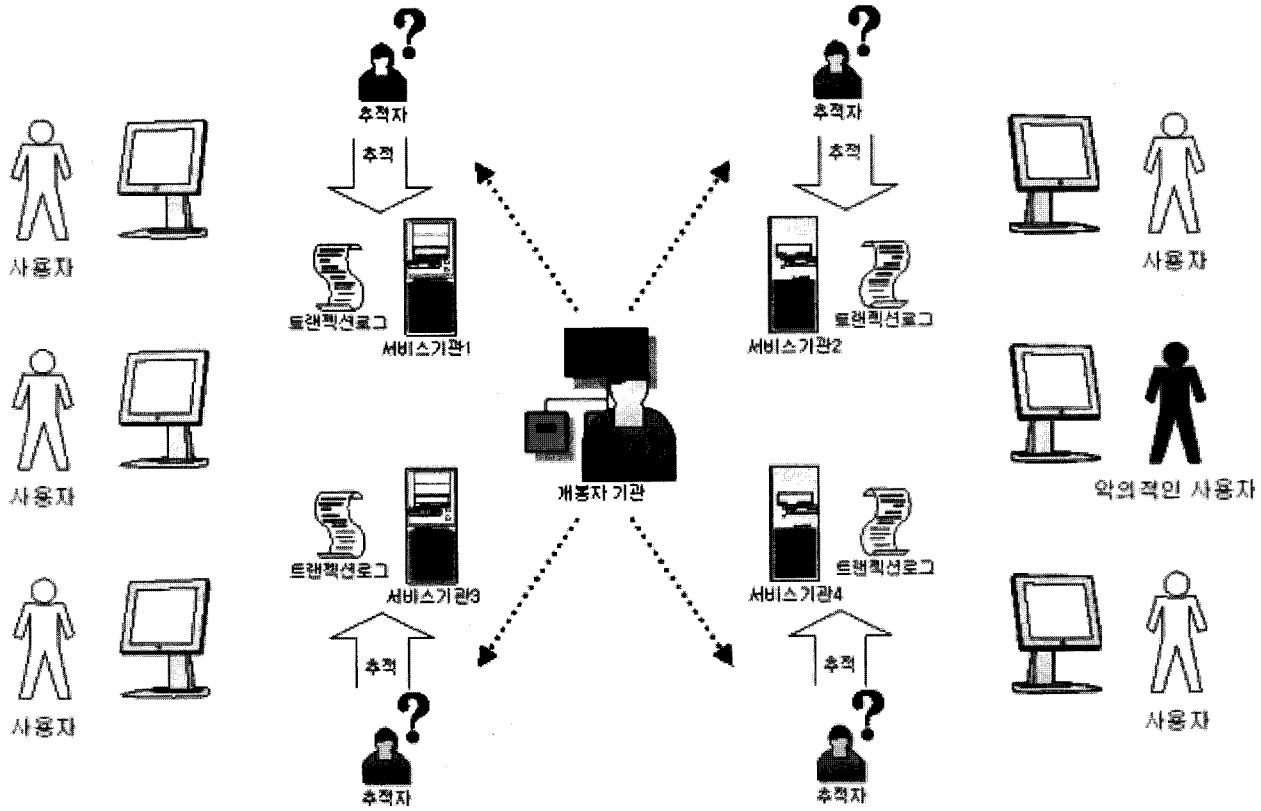


그림 2 익명 트랜잭션 시스템 : 그림 1과 같은 방법으로 악의적인 사용자를 알아낸 후, 악의적인 사용자가 요구한 모든 트랜잭션을 찾아내야 할 경우가 있다. 개봉자 기관은 노출 연산을 통해 등록 테이블에 저장된 악의적인 사용자의 서명키 일부를 추적자에게 전달한다. 추적자는 받은 정보를 가지고 각 트랜잭션들에 대해 추적 연산을 수행하여 트랜잭션이 악의적인 사용자가 요구한 것인지 아닌지를 판단한다.

수 있다. 추적 연산은 등록 테이블(reg)을 참조해야 할 필요가 없이 단순한 계산만으로 수행할 수 있는 연산이다. 따라서 그룹 멤버의 수가 n일 경우 개봉 연산이 O(n) 시간이 걸리는데 비해, 추적 연산은 O(1) 시간이 걸린다. 그리고 추적자 여러 명이 동시에 트랜잭션들을 분담해서 추적 연산을 수행할 수 있다. 전체 트랜잭션의 수를 m, 추적자의 수를 t라고 했을 때, 악의적인 멤버가 요구한 트랜잭션을 모두 찾아낼 때 필요한 시간은 O(m/t)가 되므로, 그룹 서명 방법을 사용했을 때 걸리는 O(mn) 시간에 비해 훨씬 효율적이다.

5.2 익명 경매 시스템

추적 가능한 서명이 지원하는 소유권 주장, 소유권 주장 검증 연산을 이용하면 그룹 서명에 비해 좀더 효율적인 경매 시스템을 만들 수 있다. 낙찰된 멤버는 자신이 낙찰 받은 사람임을 소유권 주장 연산을 이용하여 주장할 수 있고, 이 주장을 소유권 주장 검증을 이용해서 확인할 수 있기 때문이다.

6. 영지식 증명

그룹 서명과 추적 가능한 서명이 제공하는 익명성은 영지식 증명을 통해서 구현된다. 영지식 증명이란 내가 어떤 사실을 알고 있는 것을 상대방에게 증명할 때, 그 사실 이외에는 아무것도 상대방에게 알려주지 않는 증명 방법을 이야기한다. 이는 암호학에서 매우 중요한 개념이므로, 영지식 증명에 대해 간략하게 소개하고자 한다. 본고에서는 Discrete Logarithm에 대한 영지식 증명 방법에 대해 간략히 설명한다.

$$7^2 \equiv 49 \pmod{71}$$
$$7^3 \equiv 59 \pmod{71}$$
$$7^4 \equiv 58 \pmod{71}$$
$$7^5 \equiv 51 \pmod{71}$$
$$\cdots$$

위의 그림은 7의 멱승을 나열해 놓은 것이다. 나머지 시스템에서도 어떤 수에 대한 로그 값을 구할 수 있다. 예를 들어, $\log_7 51 \pmod{71}$는 5이다. 이를 보통의 로그와 구별하기 위해 Discrete Logarithm이라고 이야기한다. 일반적으로 이 문제는 어려운 것으로 알려져 있다. 즉,

$$a^x = b \pmod{n}$$

에서 a, b, n을 알고 있을 때 x를 구하기가 어렵다고 알려져 있다. 여기서 x값을 알고 있다는 사실을 상대방에게 증명해야 한다고 생각해보자. 물론 x값을 알려주면 되지만 x값은 비밀로 간직해야 한다. 이 경우 영지식 증명 방법을 사용하게 된다. x값을 알고 있는 사람을 증명자(Prover), 증명자가 증명해야 할 상대방을 검증자(Verifier)라고 한다. 그럼 이제 증명 방법에 대해 알아보자. 표현을 간략하기 위해 이후로는 (mod n)을 생략하도록 하겠다.

- 증명자는 r값을 랜덤하게 선택해서 $B = a^r$을 검증자에게 보낸다.
- 검증자는 c값을 랜덤하게 선택해서 증명자에게 보낸다.
- 증명자는 $s = r + cx$를 계산해서 s값을 검증자에게 보낸다.
- $a^s = Bb^c$가 성립하면 검증자는 증명자가 x값을 알고 있다고 판단한다.

실제 그룹 서명이나 추적 가능한 서명에서는 다수의 복잡한 식을 만족하는 Discrete Logarithm 값에 대한 영지식 증명 방법을 이용하여 증명한다. 영지식 증명은 Fiat과 Shamir가 제안한 변환 방법[1,16]을 이용하여 서명으로 변환될 수 있다.

7. 최근 연구 동향

초기의 그룹 서명 방법들은 서명의 크기가 그룹 멤버의 수에 비례해서 증가하여 매우 비효율적이었다. 2000년에 Ateniese 등이 서명의 크기가 고정된 그룹 서명방법을 제안하면서 새로운 전기를 맞이했으며[2], 2003년과 2004년에는 그룹 서명과 추적 가능한 서명의 모델과 안전도의 개념이 정립되었다[7,8,19,20].

또한 Boneh 등이 타원 곡선 상의 Bilinear Pairing을 이용해서 서명의 크기를 혁신적으로 줄이는 데 성공했다[9]. 이전의 서명의 크기가 1000바이트가 넘는데 비하여, [9]의 서명 방법을 이용하면 200바이트 미만으로 서명의 크기를 줄일 수 있다.

8. 결 론

본 고에서는 익명성 메커니즘인 그룹 서명과 추적 가능한 서명에 대해 알아보았다. 이 서명 방법들은 서명자가 그룹에 속한다는 사실만 알려줄 뿐 실제 서명자의 신원을 숨길 수가 있기 때문에 그 응용분야가 많다.

또한 이 서명 방법들의 토대를 이루는 영지식 증명 방법에 대해 간략히 소개하였고 최근의 연구 동향에 대해서도 알아보았다.

아직까지 실제적으로 그룹 서명이 적용되어 사용된

예는 많지 않지만, 앞으로 이 서명 방법들이 실제로 많이 사용될 것이라 예상한다.

참고문헌

[1] M. Abdalla, J. An, M. Bellare, and C. Namprepre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. EUROCRYPT 2002, LNCS, Springer.

[2] G. Ateniese, J.Camenish, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. Crypto 2000, LNCS, Springer.

[3] G. Ateniese and B. de Medeiros. Efficient Group Signatures without Trapdoors. ASIACRYPT, 2003, LNCS, Springer.

[4] G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. Financial Cryptography 1999.

[5] G. Ateniese, G. Tsudik, and D. Song. Quasi-efficient revocation of group signatures. Financial Cryptography 2002.

[6] M. Bellare, D. Micciancio, and B.Warinshci. Foundations of group signatures: Formal definitions, siimplified requirements, and a construction based on general assumptions. EUROCRYPT 2003, LNCS, Springer.

[7] M. Bellare, H. Shi, C. Zhang. Foundations of group signatures: The case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, http://eprint.iacr.org/.

[8] D. Boneh, B. Lynn, and H.Shacham. Short signatures from Weil paring. Asiacrypt 2001, volume 2248 of LNCS, Springer. Full paper: http://crypto.standard.edu/dabo/pubs.html.

[9] D. Boneh, X. Boyen,and H. Shacham. Short Group Signatures. Crypto 2004, LNCS, Springer.

[10] J. Camenisch. Efficient and generalized group signatures. EUROCRYPT 1997, LNCS, Springer.

[11] J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. EUROCRYPT 2001, LNCS, Springer.

[12] J. Camenisch and M Stadler. Efficient group signature schemes for large groups. CRYPTO 1997, LNCS, Springer.

[13] D. Chaum. Security without identification: Transactions systems to make big brother obsolete. Communications of the ACM, 28(10):1030-1044, 1985.

[14] D. Chaum and E. van Heyst. Group signatures. EUROCRYPT 1991, LNCS, Springer.

[15] L. Chen, TP. Pedersen. New group signature schemes. EUROCRYPT 1994, LNCS, Springer.

[16] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. Crypto 1986, LNCS, Springer.

[17] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: A virtual machine-based platform for trusted computing. SOSP 2003.

[18] A Lysyanskaya, Z Ramzan. Group blind digital signatures: A scalable solution to electronic cash. Financial Cryptography 1998.

[19] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable Signatures. EUROCRYPT 2004, LNCS, Springer.

[20] A. Kiayias and M. Yung. Group signatures: Efficient constructions and annymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076. http://eprint.iacr.org/.

[21] J Kilian, E Petrank. Identity escrow. CRYPTO 1998, LNCS, Springer

[22] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. IEICE Trans. Fundamentals, E84-A(5):1234-43, May 2001.

[23] L. Nguyen and R. Safavi-Naini. Efficient and Provably Secure Trapdoor-free Group Signatures Schemes from Bilinear Pairings. In Asiacrypt 2004. Available at: http://eprint.iacr.org/2004/104

[24] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 3(3): 361-396, 2000.

[25] K. Rubin and A. Silverberg. Supersingular Abelian varieties in cryptology. Crypto 2002, LNCS, Springer.

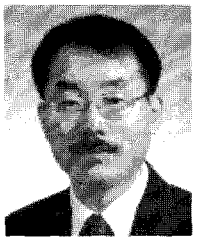
[26] C. Shnorr. Efficient signature generation by smart cards. Journal of Cryptology, 4(3):161-174, 1991

[27] Trusted Computing Group. Trusted Computing Platform Alliance (TCPA) Main Specification, 2003. Online: http://www.trustedcomputinggroup.org.

최 승 걸

1999 서울대학교 컴퓨터공학과(공학사)
2003~현재 서울대학교 전기,컴퓨터공학부 석사과정
관심분야 : 컴퓨터 보안, 암호학, 컴퓨터 이론
E-mail : sgchoi@theory.snu.ac.kr

박 근 수



1979~1983 서울대학교 컴퓨터공학과(학사)
1983~1985 서울대학교 컴퓨터공학과(석사)
1985~1991 미국 Columbia 대학교(박사)
1991. 11~1993. 8 영국 런던대학교 King's College 조교수
1995. 7~1995. 8 호주 Curtin 대학교 방문연구원
1993. 8~현재 서울대학교 컴퓨터공학부 교수
관심분야 : 컴퓨터이론, 생물정보학, 암호학
E-mail : kpark@theory.snu.ac.kr