Threshold Signature에 관한 연구

홍 성민, 윤 현수

대전광역시 유성구 구성동 373-1
한국과학기술원 전산학과
{smhong,hyoon}@camars.kaist.ac.kr

Study on Threshold Signature

Seong-Min Hong, Hyunsoo Yoon
Computer Science Dept., KAIST
373-1 Kusong-Dong Yusong-Gu Taejeon

요 약

n명의 사용자들 중 k명 이상이 모이면 시스템의 비밀정보가 재구성 되는 시스템을 threshold cryptosystem 이라고 한다. 1979년 Adi Shamir에 의해 threshold 개념이 제안된 이래로, 그 현실적인 효용성때문에 많은 연구들이 이루어져 왔다. Threshold cryptosystem을 구성하기 위해서는 n명의 사용자들이 각자의 share를 가지고 있어서 그 중 k명 이상이 자신의 share를 밝히면 시스템의 비밀정보가 재구성된다. 그러나, 시스템의 비밀정보가 재구성된 후에는 각 사용자들의 share를 재분배 해야 한다. 이러한 특성은 반복적으로 전자서명과 같은 비밀연산을 수행해야 하는 응용에서는 비현실적이 된다. 이러한 경우에 적용시키기 위한 암호시스템이 threshold signature이다. (t,n)-threshold signature 시스템에서는 n명의 사용자들 중 k명 이상이 자신의 share를 이용한 부분서명을 수행하면 전체 시스템 서명이 생성된다. 그러나, 시스템의 비밀정보는 재구성되지 않으므로 아무도 알 수 없고, 따라서 재분배 할 필요가 없다. 본 논문에서는 이러한 threshold signature 시스템에 대한 연구결과들을 살펴본다.

제 1 절 서론

사회 전반에 걸쳐 컴퓨터 시스템의 이용이 증가하고 정보 전송 수단이 발전함에 따라 정보처리능력이 증대되고 정보의 이동이 많아지고 있다. 이에 따라 현대 사회는 정보화사회라고 불리울 만큼 정보의 효율적 이용이 중요하게 인식되고 있다. 정보의 효용가치가 높아짐에 따라 정보를 보호하는 문제가 심각한 문제로 제기되고 있다. 만약 지정된 수신자 이외에는 결코 정보를 획득할 수 없는 전송수단이 존재하고, 허가받지 않은 사용자는 절대 접근할 수 없는 컴퓨터 시스템이 존재한다면 정보를 보호하기 위한 별도의 노력은 필요치 않을 것이다. 그러나 그러한 컴퓨터 시스템이나 정보전송 수단은 현재 존재하지 않으며, 따라서 이미 저장되었거나 현재 전송중인 정보는 누구에게나 노출되어 있다고 보아야 한다. 따라서, 정보의 보호를 위한 가장 효율적인 방법은 암호시스템(cryptosystem)을 구성하여, 정보를 알아볼 수 없는 형태로 바꾸어 저장하거나 전송하는 것이다.

정보보호의 기능 이외에도 전자현금, 전자투표 등 암호시스템이 필수적인 응용분야들이 있고, 그러한 것들 중 하나가 threshold cryptosystem이다. 핵미사일 발사 프로그램의 예를 들어 보면, 어떤 미친 사람이 발사프로그램을 작동시키는 것을 원치 않는다. 핵미사일의 오발을 막기 위해 5명의 장군들 중 최소한 세명이 모여야 발사시킬 수 있도록 하고싶다면, 기계적인 장치를 동원하여 세 개의 슬롯 모두에 키가 채워져야 작동할 수 있도록 만들고 각 장군들에게 키를 하나씩 나누어 주면 된다. 그러나, 훨씬 복잡한 상황이 일어날 수 있다. 예를 들면, 2명의 장군과 3명의 대령이 모여야 발사할 수 있도록 하고싶거나, 5명의 대령이 모여도 발사 될 수 있도록 하고싶을 수 있다.

훨씬 더 복잡한 상황까지도 수학적 방법으로 해결할 수 있는 것이, threshold scheme이라고 불리는, secret sharing 기법이다. 가장 간단한 것은 어떤 비밀 정보를 n개의 조각으로 나누고 - shadow 또는 share 라고 불린다 - 그들 중 어느것이든 t개가 모이면 원래의 메세지가 복원되도록 하는 것이다. (t,n)-threshold scheme 이라고 부른다. 일반적인 threshold scheme은 훨씬 많은 일들을 할 수 있는데, 상상가능한 어떠한 sharing 시나리오도 모델링될 수 있다 [1].

threshold scheme의 실제적인 효용성이 매우 크기때문에, threshold cryptosystem에 대한 연구가 많이 이루어져 왔다. 그러나, 일반적인 threshold cryptosystem이 적용되기 힘든 분야가 있다. (t,n)-threshold cryptosys-

tem을 이용해서 수표를 발행하는 은행의 예를 들어 보면, 수표를 발행 하기 위해서는 은행의 직인을 찍어야 하고, 은행의 직인을 찍기 위해서는 수표발행 권한을 지닌 - share를 지닌 - 사람들 중 t명이 모여야 한다. 그런데, 한번 수표를 발행하고 나면 은행의 비밀정보가 밝혀지기때문에, 은행의 비밀정보에 대한 share를 재분배 하는 작업을 다시 수행해야 한다.

이처럼 반복적으로 전자서명을 수행하는 응용분야에 이용될 수 있는 암호시스템이 threshold signature 시스템이다. 1991년 Desmedt와 Frankel에 의해 제안되어 많은 연구들이 있어왔다. Threshold signature 시스템을 이용하면, 각자의 share를 이용한 부분서명들만으로 시스템 전체 서명이 수행되도록 함으로써 시스템의 비밀정보가 노출되지 않는다.

본 연구에서는 이러한 threshold signature 시스템에 대한 기존의 연구들을 살펴보고, 이를 실제로 구현하기 위해서 주의해야 할 사항들에 대해 조사한다.

제 2 절 관련연구

본 절에서는 공개키 암호시스템과 secret sharing 기법에 대해 살펴 본다.

2.1 공개키 암호시스템

암호시스템은 수 천년 전부터 사용되어 왔으나, 최근에 이르기까지 그 형태가 모두 동일하였다[2]. 즉, 평문(plaintext)을 알아볼 수 없는 암호문(ciphertext)으로 변형시키는 데에 사용되는 키(key)와, 암호문을 다시 알아볼 수 있는 형태의 복호문으로 바꾸는 데에 사용되는 키가 같았다. 이러한 방식의 암호시스템을 대칭적 암호시스템(symmetric cryptosystem) 또는 단일키 암호시스템(one-key cryptosystem)이라고 한다. 대칭적 암호시스템은 두 가지의 치명적인 문제점을 안고 있다. 하나는 정보를 비밀리에 교환하고자 하는 사람들은 키를 공유해야 하기때문에, 별도의 안전한 정보전송 통로를 필요로 한다는 점이다. 대칭적 암호시스템의 또 다른 문제점은 전자서명(digital signature)을 효과적으로 구현할 수 없다는 점이다. 전자서명이란, 실생활에서 사용되는 서명의 기능을 전자문서에 적용시킨 개념으로서, 문서 작성자의 정체와 문서의 무결성을 증명할 수 있는 표시(signature)를 말한다.

이러한 문제들을 해결하기 위해서, 1976년에 Diffie와 Hellman이 공개키 암호시스템(public-key cryptosystem)을 제안하였다[3]. 공개키 암호시스템은 암호화(encryption) 시와 복호화(decryption) 시에 사용하는 키를 따로 두어, 암호화 시에 사용하는 키(공개키)를 모든 사람들에게 공개하고, 복호화 시에 사용하는 키(개인키)는 비밀리에 보관하는 방식의 암호시스템을 말한다. 예를 들면, 갑이라는 사람이 을이라는 사람에게 정보를 비밀리에 전달하기 위해서는 누구나 알고있는 을의 공개키(public-key)로 정보를 암호화하여 전송한다. 그러면, 을은 자신만이 알고있는 개인키(private-key)로 갑이 보낸 암호문을 해독하여 정보를 얻을 수 있다. 따라서, 정보를 공유하고자 하는 사람들이 키를 공유할 필요가 없으므로 별도의 안전한 정보전송 통로를 필요로 하지 않는다. 또한 공개키 암호시스템을 사용하면 전자서명을 효과적으로 구현할 수 있다[3, 4, 5]. 공개키 암호시스템의 이러한 두 가지 특성들은 기존의 커다란 과제들을 해결한 획기적인 것이었다. 따라서, 공개키 암호시스템이 제안된 이래로 많은 연구들이 이루어져 왔다[6].

2.2 Secret Sharing

본 절에서는 threshold scheme이라고 불리는, secret sharing 기법에 대해 설명한다. 어떤 비밀 정보를 n개의 조각으로 나누고 - shadow 또는 share 라고 불린다 - 그들 중 어느것이든 t개가 모이면 원래의 메세지가 복원되도록 하는 것이다. (t, n)−threshold scheme 이라고 부른다.

Secret sharing에는 여러 종류가 있다. 첫째, 속임수를 시도하려는 사용자를 가정하는 secret sharing 기법이 있다. Threshold scheme을 깨트리는 방법들이 많이 있는데, 그러한 시나리오들 중 몇개만 예를 들어보면 다음과 같다: 엉터리 shadow를 내놓으면, 누가 그랬는지조차 알 길이 없고, 거짓상황 조작으로 shadow들을 내놓게 만들 수 있으며, 실제상황에서 가장 늦게 shadow를 내놓음으로써 계산할 시간과 정보를 얻을 수 있다. 둘째, 기본적인 threshold scheme과 같으나, 어느누구도 전체 비밀정보를 알지 못하게 하는 secret sharing without trent 기법이 있다. 세째, sharing a secret without revealing the shares 기법이 있다. 기본적인 방법에서는 비밀정보를 재구성하기 위해 모두 모이게 되면 각자의 share들이 드러나게 되므로, 다시 사용할 수 없게 된다. 따라서 필요하게 되는 방법이다. 요점은 비밀정보를 드러내지 않음으로써 재사용이 가능하게 하는 방법이다. 또한 비밀정보를 다루는 trusted processor가 필요치 않게 된다. 네째, verifiable secret sharing 기법이 있다. 이 방법에서는 share를 가지고 있는 모두가, 전체 비밀정보를 재구성해보지 않고, 각자 개인적으로 자신이 가지고 있는 share가 유효한 것인지를 확인 할 수 있다. 그리고, 비밀정보가 재구성되는 것을 막을 수 있는 장치가 덧붙여진 secret-sharing schemes with prevention 기법과, share를 지니고 있던 구성원을 제외시키고싶을때, 전체 시스템을 새로 만들지 않아도 되도록 하는 secret-sharing schemes with disenrollment 시스템이 있다.

secret sharing에 사용될 수 있는 기법들은 처음 Adi Shamir가 제안한 LaGrange interpolating polynomial scheme을 비롯해 George Blakley의 Vector Scheme, Asmuth-Bloom의 방법, Karnin-Greene-Hellman 방법 등이 있다. 본 절에서는 LaGrange Interpolating Polynomial Scheme에 대해 간략하게 설명한다.

1. prime number p를 선택한다. $p >$ # of possible shadow, and $p >$ largest possible secret.
2. $m-1$ degree의 임의의 polynomial을 만든다. 모든 계수들은 p보다 작아야 한다. polynomial의 상수항이 secret이 된다.
3. 만들어진 polynomial에 n개의 다른 점들의 좌표를 계산한다. (1부터 n까지 대입) coefficient가 알려지지 않게 주의...
4. 각 점을 n명에게 각각 알려준다.

5. n명중 m명 이상만 모이면 과정 2.에서 만든 polynomial을 다시 만들어 secret M을 알아 낼 수 있다. linear equation들만 풀면 된다.

2.3 주로 사용되는 암호 기법들

RSA 서명자는 매우 큰 소수 p, q를 구하고 $N = pq$를 계산한다. 그리고나서, $\phi(N)(= (p-1)(q-1))$과 서로 소인 임의의 정수 e를 고른 후에 $ed \equiv 1 \bmod \phi(N)$이 성립하는 d를 찾는다. 이러한 초기 설정이 끝나면, 서명자는 N, e를 공개하고, d는 자신만이 아는 개인키의 역할을 하게 된다. 임의의 메시지 M에 대한 서명은 $M^d \bmod N$이 된다.

ElGamal Z_p(,where p는 소수)상의 원시원소(primitive element)를 α로 놓는다. p,α는 모든 사용자들이 알고 있고 공통으로 사용하는 시스템 인수(parameter)들이다. 서명자는 $\alpha^a \equiv \beta \bmod p$ 인 a와 β를 선택하고, β는 공개된다. 임의의 메세지 x에 대한 서명은 (γ, δ)이다. $\gamma = \alpha^k \bmod p$이고 $\delta = (x - a\gamma)k^{-1} \bmod (p-1)$이다. 서명에 대한 확인은 $\beta^\gamma \gamma^\delta \equiv \alpha^x (\bmod p)$인지 아닌지로 수행한다.

DSS DSS(Digital Signature Standard)는 ElGamal 서명의 변종으로서 미국의 전자서명 표준이다. ElGamal 서명의 변형에 대한 동기를 살펴본다. 우선, 전자서명은 암호알고리즘과는 다르게 꽤 오랜시간동안 효력을 지녀야 하므로 보다 큰 안전성을 요구한다. ElGamal 서명의 p의 크기가 최소 512비트는 되어야 하고, 많은 사람들이 1024비트는 되어야 한다고 주장한다. 그러나, p의 크기가 512비트만 되도 서명의 크기가 1024비트가 된다. DSS는 이 점은 개선하여, 160비트의 메세지에 서명하여 320비트의 서명이 생성되도록 하였다. 그러나 계산은 512비트 modulus p에 대해 수행된다.

소수 p와 q를 선택하는데, p는 512비트의 크기이고 q는 $p-1$을 나누는 160비트의 크기를 지닌다. modulo p에 대한 $q-th$ root를 α로 놓는다. p,α는 모든 사용자들이 알고 있고 공통으로 사용하는 시스템 인수(parameter)들이다. 서명자는 $\alpha^a \equiv \beta \bmod p$ 인 a와 β를 선택하고, β는 공개된다. 임의의 메세지 x에 대한 서명은 (γ, δ)이다. $\gamma = (\alpha^k \bmod p) \bmod q$이고, $\delta = (x + a\gamma)k^{-1} \bmod q$이다. 서명검증은 $(\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$인지 아닌지로 수행한다. $e_1 = x\delta^{-1} \bmod q$이고, $e_2 = \gamma\delta^{-1} \bmod q$이다.

Lagrange interpolation formula t개의 ordered pair $(x_1, y_1), (x_2, y_2), \cdots, (x_t, y_t)$ 를 포함하는 $t-1$차를 넘지 않는 다항식은 다음과 같은 식으로 표현될 수 있고, 이를 Lagrange interpolation formula라고 한다.

$$a(x) = \sum_{j=1}^{t}(y_{i_j}) \prod_{1 \le k \le t, k \ne j} \frac{x - x_{i_k}}{x_{i_j} - x_{i_k}}.$$

위 식의 기능적인 의미는 t개의 ordered pair를 알고 있을 때, 그것들을 포함하는 다항식을 구하는 것이다. 그러나, threshold scheme들에서는 다항식 자체를 구할 필요가 없고, 단지 $a(0)$(, 즉 x가 0일때의 다항식 값, 또는 다항식의 상수항 값)만을 알면 되므로 조금 더 간단한 방법을 사용한다. 그리고, 그 방법이 threshold scheme들의 기본 골격이 된다.

다항식의 상수항을 K라고 놓으면(이것이 그룹 전체 키가 되기도 한다.) Lagrange interpolation formula는 다음과 같다.

$$K = \sum_{j=1}^{t}(y_{i_j}) \prod_{1 \le k \le t, k \ne j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}.$$

x_{i_j}들은 공개된 정보들이고, y_{i_j}들은 share들이므로, t개의 share들이 모이면 위 식을 계산할 수 있다.

Secret Sharing of Sums and Products SDC 없이 secret에 대한 share들을 분배할 때 필요한 방법을 설명한다. S.Langford가 [11]에서 정리한 것을 요약한다. secret sharing에서 secret의 값은 share들의 값에 의해 결정되지만, secret 자체는 share들이 생성될 때 명시적으로(explicitly) 계산되지 않는다. Threshold signature에서는 그룹의 private key가 직접적으로 이용되는 일이 없으므로 명시적으로 계산할 필요가 없다.

Shamir의 scheme은 묵시적으로(implicitly) secret을 생성하는 데 이용될 수 있다. 두 개의 숫자의 합에 대한 share들을 생성할 때, 아무도 그 합을 알지 못하게 하고자 하는 그룹을 가정한다. A는 secret S_A에 대한 sharing polynomial $f_A(x)$를 생성하고, 마찬가지로, B는 S_B에 대해 $f_B(x)$를 생성하여 그 결과로 생성되는 share들을 분배한다. 그러면, $f(x) = f_A(x) + f_B(x)$는 secret $f(0) = f_A(0) + f_B(0) = S_A + S_B$를 갖는 새로운 다항식이 된다. $f_A(i)$와 $f_B(i)$를 받은 사용자 i는 두 share들을 더함으로써 새로운 다항식에 대한 share $f_A(i) + f_B(i)$를 갖게 된다. 이러한 scheme은 여러개의 다항식으로도 확장된다.

이러한 다항식을 지수로 사용하면 서로 곱하는 share를 생성하게 된다. 소수 q와 $GF(q)$에서 $order$가 n인 원소 β가 주어졌을 때, 다항식 $f(x)$를 생성하고 secret을 $\beta^{f(0)}$으로 정의할 수 있다. 보통의 linear interpolation formula를 지수로 취함으로써 secret이 재구성될 수 있다.

제 3 절 연구흐름 분석

Threshold cryptosystem이나 secret sharing scheme에 대해서는 많은 논문들이 존재한다. 그러나, threshold signature를 직접적으로 다루고 있는 논문들은 다음과 같은 12개로 뽑아볼 수 있다. 각각의 논문의 개괄적인 contribution을 정리한다.

1. "How to share a secret", Adi Shamir, CACM, 1979
   Secret sharing 개념을 최초로 제안한 논문으로서, Lagrange interpolation 을 이용해 RSA 기반의 secret sharing scheme을 제안한다 [1].

2. "Shared generation of authenticators and signatures" Yvo Desmedt and Yair Frankel, CRYPTO'91 pp.457-
   처음으로 제안된 threshold signature scheme으로서, l명에게 secret share를 나누어 주고 k명이 모이면 RSA sigature를 생성할 수 있다 [7].

3. "Remark on the threshold RSA signature scheme", Chuan-Ming Li, Tzonelih Hwang, Narn-Yih Lee,

CRYPTO'93, pp.413-
Y.Desmedt와 Y.Frankel이 91년에 제안한 threshold RSA signature scheme의 '공모에 의한 공격'에의 취약점에 대해 연구한 논문이다. k 명 이상이 모이면 시스템 비밀이 노출될 수 있음을 보인다 [8].

4. "(t,n) threshold signature schemes based on discrete logarithm", Chuan-Ming Li, Tzonelih Hwang, Narn-Yih Lee, Eurocrypt'94, pp.191-
기존의 방법들(하나는 Y.Desmedt등이 제안한 거고, 다른 하나는 Harn이 제안한 방법인데 reference를 구하기 어려운 보도듣도 못한 conference 논문을 reference랍시고 적어놓았음)이 가진 문제점은 t명 이상이 작당을 하면 system secret을 알아낼 수 있다는 것임. 이러한 문제를 해결하는 새로운 방법의 threshold scheme 두 개를 제안하는데, 하나는 SDC(shared distribution center)를 이용하는 방법이고, 다른 하나는 이용하지 않는 방법이다. 이 논문에서 제안하는 방법은 서명과 서명자들의 정체성이 연결되어 서명자들의 정체가 드러난다 [9].

5. "Group-oriented (t,n) threshold digital signature scheme and digital signature scheme and digital multisignature", L.Harn, IEE Proc.-Computers and Digital Techniques, Vol.141, No.5, Septermber 1994, pp.307-313
discrete logarithm 문제에 기반한 threshold signature scheme을 제안한 논문. 시기적으로 앞의 논문보다 일찍인 것으로 생각되며, 최초의 RSA 기반 threshold signature scheme과 마찬가지로 k명 이상이 모이면 시스템 비밀이 드러난다. [10].

6. "Threshold DSS signatures without a trusted party", Susan K.Langford, CRYPTO'95, pp.397-
기존의 대부분의 threshold signauture scheme들은 믿을만한 제3자를 필요로 했었다. Harn이 4에서 제안한 방법과, 3에서 C.Li 등이 제안한 방법들은 discrete logarithm을 기반으로 하고, 믿을만한 제3자를 필요로 하지 않는다. 이러한 특성을 DSS signature 에 적용시켜 (2,l) threshold signature 를 만들고, (t,l) threshold로 확장시키는 두가지 방법을 제안한다 [11].

7. "Robust threshold DSS signatures", Rosario Gennaro, Stanislaw Jareckl, Hugo Krawczyk, and Tal Rabin, Eurocrypt'96, pp.354-
CRYPTO'95에 실린 5번 논문에 대한 확장 논문으로, Robust 개념 또한 도입되었다. Robust 개념은 서명 생성시에 악의의 서명자들이 서명생성에 훼방을 놓아도, 즉 잘못된 부분서명(partial signature)을 생성하여도, 전체 서명생성을 올바로 수행할 수 있음을 말한다 [12].

8. "Weaknesses in some threshold cryptosystems", Susan K.Langford, CRYPTO'96, pp.74-
threshold cryptosystem은 신뢰할만한 제3자를 필요로 하느냐, 그렇지 않느냐에 따라 2 종류로 나뉜다. 이 논문에서는 신뢰할한한 제3자를 이용하지 않는 threshold cryptosystem의 보안 헛점(security weakness)들을 짚어본다. Auscrypto'92에 발표된 Harn과 Yang의 group-oriented undeniable signature가 2-out-of-n정도의 보안정도밖에 제공하지 못함을 보인다. 또한, discrete logarithm에 기반한 시스템들이 key generation protocol에 헛점을 가지고 있음을 보인다. 마지막으로, Laih와 Harn이 Asiacrypt'91에서 발표한 generalized threshold cryptosystem이 특정 access structure들에서 안전하지 못함을 보인다. [13].

9. "How to Share a Function Securely", Alfredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung ACM STOC'94, pp.522-533
이 논문은 function sharing 개념을 세움으로써, threshold signature scheme을 일반화 하고, RSA를 이에 맞게 변형시켜 sharing 가능한 RSA로 만들었다 [14].

10. "On the Risk of Disruption in Several Multiparty Signature Schemes", Markus Michels and Patrick Horster, Asiacrypt'96, pp.334-345
threshold signature scheme들에 대한 몇가지 공격에 대한 논문이다 [15].

11. "Robust and Efficient Sharing of RSA Functions", Rosario Gennaro, Stanistaw Jarecki, Hugo Krawczyk, and Tal Rabin, Crypto'96, pp.158-172
RSA에 기반한 threshold signature scheme을 robust하게 만든 것이다. 기반이 되는 scheme은 [14]로서, [12] 의 robust 개념을 구현하기 위해, undeniable signature와 verifiable secret sharing scheme을 이용한다 [16]

12. "New ElGamal Type Threshold Digital Signature Scheme", Choonsik PARK and Kaoru KUROSAWA, IEICE Trans. Fundamentals. Vol.E79-A, No.1, January, 1996, pp.86-93
discrete log 문제에 기반한 threshold signature scheme을 제안한 논문이다 [17].

threshold signature에서 issue가 되었던 연구방향들을 살펴본다. 처음에 Desmedt와 Frankel이 threshold signautre를 제안할 때, 세 개의 알고리즘을 제안하였다. RSA를 기반으로 하는것, probably secure, 그리고 unconditionally secure threshold signautre algorithm이었다. 그 중 RSA를 기반으로 하는 threshold signature는 1993년 Chuan-Ming Li 등에 의해 약점이 발견되었다. 그 약점은 (k,l) threshold signature에서 k명 이상이 작당을 하면 system secret을 발견하게 되어, 서명위조, 서명부인 등이 가능하게 된다. 물론, $k-1$명은 그렇게 할 수 없다. 이러한 약점을 해결하기 위해 Chuan-Ming Li 등이 discret logarithm에 기반한 threshold signature를 1994년에 발표하였다. 이 때, share distribution center(SDC)를 이용하는 방법과 그렇지 않은 방법 두 개를 발표하였다. 1995년에는 DSS(Digital Signature Standard)를 기반으로 하여, SDC를 필요로 하지 않는 threshold signature 기법이 제안되었고, 1996년에 개선되었다. Crypto'96에서 Susan Lanford는 discret logarithm을 기반으로 하는 threshold cyrptosystem에서의 헛점을 보였다.

이러한 논문들을 분야별로 정리해 보면 크게 두 부분으로 나눌 수 있다. 하나는 RSA를 기반으로 하는 시스템이고, 다른 하나는 discrete log 문제에 기반한 시스템이다.

1. RSA 기반:
   [7], [8], [14], 그리고 [16]이 여기에 속한다. 이들 중에서 [8]은 공격논문이며 나머지 셋은 시스템을 제안하는 논문이다.

2. discrete log 기반:
   [9], [10], [11], [12], [17], 그리고 [13]이 여기에 속한다. 이 중에서 [13]은 공격논문이며 나머지는 시스템을 제안하는 논문이다. discrete log 문제에 기반하는 시스템을 제안하는 논문들은 다시 두 부분으로 나뉜다.

   - ElGamal 기반: [9], [10], [17].
   - DSS 기반: [11], [12].

제 4 절 Paractical Consideration

지금까지 살펴본 기존의 threshold signature system들을 실제로 사용하기 위해서 고려해야 할 사항들에 대해 살펴본다.

기존의 threshold signature system들에서는 구성멤버들의 변동을 크게 고려하지 않고 있다. 예를 들어 (t,n)-threshold signature system에서 1명의 멤버를 축출(disenrollment)하고 싶을때, 가장 간단한 방법은 그 1명의 멤버를 제외한 다른 멤버들로만 구성해서 share distribution protocol을 다시 수행하는 것이다. 그러나, 그렇게 하면 그 그룹의 공개키가 바뀌게 되고 이 사실이 그룹 밖의 사람들을 포함한 다른 모든 사람들에게 적시에 알려져야 한다. 따라서, 적절한 대책이 필요하다.

RSA에 대한 (t,n)-threshold signature system에서의 disenrollment를 살펴 본다. 그룹 전체의 비밀키를 d라고 할 때, SDC는 $f(0) = d - 1$인 $t-1$차 다항식 $f(x)$를 만들고, K_i를 각 멤버들에게 전달한다. * 서명생성 과정은 앞에 기술되 있으므로 생략한다. 이제 그룹멤버들 중 임의의 멤버를 축출하는 방법을 살펴본다. 축출해야 할 멤버를 k(, where $1 \le k \le n$)로 놓는다. 멤버 k는 이미 자신의 share를 알고 있으므로, 이를 축출하기 위해서는 현재의 share를 소용없게 만들어야 하는데, 가장 간단한 방법은 SDC가 그룹비밀키를 d'으로 바꾸고 †, $f'(0) = d' - 1$인 $f'(x)$를 만들어 k를 제외한 모든 멤버들에게 바뀐 share $K'_i = \frac{f'(x_i)/2}{\prod_{j \in A, j \neq i}} (\mathrm{mod} p'q')$ (, where $1 \le i \le n$)들을 전달하는 것이다. 그러나, 이 방법을 사용하면 그룹비밀키가 바뀌었기때문에 그룹공개키도 바뀌어야 한다. 그룹 공개키가 바뀌게 되면, 이 그룹의 서명을 필요로 하는 모든 다른 그룹 또는 개인들은 이 그룹의 공개키가 바뀌었다는 사실을 즉시 알 수 있어야 하고 해당하는 조처를 취해 주어야 한다. 이는 상당한 부담이 된다.

Discret log에 기반한 threshold signature system의 경우에도 마찬가지다. 우선, SDC가 있는 경우를 살펴본다. SDC는 $f(0)$가 그룹비밀키가 되는 다항식 $f(x)$를 만들고, $f(x_i) \bmod q$를 각 멤버 i의 share로 분배한다. x_i는 각 멤버들의 공개된 정보이다. $y = \alpha^{f(0)} \bmod p$가 그룹공개키가 되고, 각 멤버 i의 공개키는 $y_i = \alpha^{f(x_i)}$가 된다. 멤버 k를 축출하고자 할 때, SDC는 $f'(0) \neq f(0) \bmod q$인 $f'(x)$를 만들고 $f'(x_i) \bmod q$를 k를 제외한 모든 멤버들에게 나누어 준다. RSA의 경우와 마찬가지로 그룹공개키가 바뀌는 문제가 발생한다.

SDC가 없는 discret log 기반의 threshold signature system의 경우를 살펴본다. 각 멤버 i는 자신이 선택한 부분비밀키 z_i를 이용해서 $y_i = \alpha^{z_i} \bmod p$를 만들고, 그룹공개키는 $y = \prod_{i=1}^{n} y_i \bmod p$가 된다. 각 멤버 i는 자신의 비밀키에 대한 share를 만들어야 하는데 이는 $y_{i,j} = \alpha^{f_i(x_j)} \bmod p$가 되고, 여기서 $f_i(x)$는 멤버 i가 생성한 $t-1$차 다항식으로서 $f_i(0)$는 z_i이다. 여기서 멤버 k를 축출하고자 하면, 가장 간단한 방법은 위의 그룹공개키 생성과정과 그에 대한 share 분배 프로토콜을 다시 수행하는 것이다. 그러나, 위의 다른 시스템들과 마찬가지로 공개키가 바뀌는 문제가 발생한다.

우선, SDC가 멤버 k를 축출하기 위해 다항식 $f'(x)$를 만들 때, $f'(0) = d - 1$이 되도록 할 수 있다. 그러면, 비밀키 d가 바뀌지 않았으므로 공개키가 바뀔 필요가 없다. 각 멤버가 가지게 되는 새로운 share K'_i만 새로 분배하면 된다. 하지만, 앞에서도 지적했듯이 t명 이상이 축출되고 나면 축출된 멤버들끼리만으로도 서명이 가능해진다. 한번에 k명이 축출되지 않더라도 오랜 기간에 걸쳐 축출된 멤버들의 수가 k 이상이 될 수 있고, 축출된 멤버가 k명이 되지 않더라도 축출된 멤버들의 수만큼 시스템의 안전성이 약화되었다고 볼 수 있다. 예를들면, 악의를 지닌 l명이 축출된 $k-l$명의 share들을 댓가를 지불하고 구하게 될 수 있다.

다음으로, SDC가 $g(0) = 0$인 $t-1$차 다항식 $g(x)$를 구해서, 축출하고자 하는 멤버 k를 제외한 모든 멤버 i들에게 $g(x_i)$를 이용한 새로운 share $L_i = \frac{g(x_i)/2}{\prod_{j \in A, j \neq i}} (\mathrm{mod} p'q')$들을 전달해 주는 방법이 있다. 멤버 k를 제외한 각 멤버 i들은 $K'_i = K_i + L_i$를 계산해서 새로운 share로 가지게 있게 된다. $f(0) + g(0) = d - 1$로 변함이 없으므로 t개의 새로운 share들이 모이면 서명수행이 변함없이 가능해지고, 기존의 share K_i는 K'_i와 함께 사용할 수 없으므로 멤버 k는 서명수행능력을 잃게되어 결과적으로 축출된 셈이 된다. 그러나, 이 방법은 전혀다른 새로운 share를 분배하는 앞의 방법과 차이점이 없으므로 security가 약해지는 같은 문제점을 지니게 된다.

마지막으로 우리들이 제안하는 방법을 살펴본다. 우선 SDC가 처음에 다항식 $f(x)$를 만들때, $f(0) = d$가 되도록 하는 대신 $f(0) = R \cdot d - 1$이 되도록 한다. 그리고, 나머지 과정은 기존의 방법과 같이 한다. 단지 서명생성시에 $s_{m,i,B}$들로부터 그룹서명을 생성할 때의 과정만 다

*RSA에 대한 기존의 모든 threshold signature system에서는 SDC가 반드시 필요하다. K_i는 $\frac{f(x_i)/2}{\prod_{j \in A, j \neq i}} (\mathrm{mod} p'q')$이다.

†그룹비밀키를 d'으로 바꾸지 않고 그대로 d를 사용해서 다항식만 바꿀 수도 있다. 그러나, 그러한 경우 t명이 축출되고나면 축출된 멤버들끼리 서명을 수행할 수 있다.

음과 같다: $s_m = (m\prod_{i\in B} s_{m,i,B})^{R^{-1}}$. 멤버 k를 축출하고자 할 때, $f'(0) = R'd - 1$인 $f'(x)$를 구해서 멤버 k를 제외한 모든 멤버 i들에게 $f'(i)$를 이용한 share K'_i를 전달해 준다. 이전의 두 가지 방법은 모두 축출된 멤버들이 모여서 악의의 행위를 할 수 있는 가능성을 남겨놓고 있다. 그러나 이 방법을 사용하면 SDC의 도움 없이는 서명을 수행할 수 없고 매번 R이 바뀌게 되므로 한 번 축출당한 멤버 k는 서명자로서의 권리를 다시는 행사할 수 없다. 또한 공개키가 바뀔 필요가 없으므로 막대한 비용을 초래하는 일을 막을 수 있다.

참고 서적

[1] A. Shamir, "How to share a secret," *Communication of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] D. Kahn, *The Codebreakers.* Macmillan Publishing, 1967.

[3] W.Diffie and M.E.Hellman, "New directions in cryptography," *IEEE Trans. Computers*, vol. IT-22, pp. 644–654, June 1976.

[4] R.L.Rivest, A.Shamir, and L.Adleman, "A method for obtaining digital signatures and public key crytosystems," *CACM*, vol. 21, pp. 120–126, 1978.

[5] T.ElGmal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469–472, 1985.

[6] W. Diffie, "The first ten years of public-key cryptography," in *Proceeding of The IEEE*, vol. 76,NO.5, pp. 560–576, May 1988.

[7] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures," in *Crypto'91*, pp. 457–469, 1991.

[8] T. H. Chuan-Ming Li and N.-Y. Lee, "Remark on the threshold rsa signature scheme," in *Crypto'93*, pp. 413–419, 1993.

[9] T. H. Chuan-Ming Li and N.-Y. Lee, "(t,n) threshold signature schemes based on discrete logarithm," in *Eurocrypt'94*, pp. 191–200, 1994.

[10] L. Harn, "Group-oriented (t,n) threshold digital signature scheme and digital signature scheme and digital multisignature," *IEE Proc.-Computers and Digital Techniques*, vol. 141, no. 5, pp. 307–313, 1994.

[11] S. K.Langford, "Threshold dss signatures without a trusted party," in *Crypto'95*, pp. 397–409, 1995.

[12] R. Gennaro, S. Jareckl, H. Krawczyk, and T. Rabin, "Robust threshold dss signatures," in *Eurocrypt'96*, pp. 354–371, 1996.

[13] S. K.Langford, "Weaknesses in some threshold cryptosystems," in *Eurocrypt'95*, pp. 74–82, 1996.

[14] A. D. Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," in *ACM STOC'94*, pp. 522–533, 1994.

[15] M. Michels and P. Horster, "On the risk of disruption in several multiparty signature," in *Asiacrypt'96*, pp. 334–345, 1996.

[16] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and efficient sharing of rsa functions," in *Crypto'96*, pp. 158–172, 1996.

[17] C. PARK and K. KUROSAWA, "New elgamal type threshold digital signature scheme," *IEICE Trans. Fundamentals*, vol. E79-A, pp. 86–93, Jan. 1996.