情報保護學會論文誌
第12券 第6號, 2002. 12

서명 능력을 제한하는 전자 서명 스킴

황 정 연*, 이 동 훈*, 임 종 인*

Digital Signature Schemes with Restriction on Signing Capability

Jung Yeon Hwang*, Dong Hoon Lee*, Jong In Lim*

요 약

실제적인 환경에서는 서명자의 능력이 제한되어져야 하는 경우가 있다. 그룹 서명 스킴(Group Signature Scheme)에서 그룹의 구성원은 그룹 내에서 자신의 직위에 따라 정해진 횟수까지 서명을 생성하도록 허락되어 질 수 있다. 또는 대리 서명 스킴(proxy signature scheme)에서 본래의 서명자는 대리 서명자(proxy signer)에게 그 서명자를 대신하여 일정한 횟수까지만 서명을 생성하도록 서명능력을 제한하고 싶을 것이다. 본 논문에서는 전자서명 스킴 설정단계에서 미리 정해진 값 c에 대해 서명자의 서명횟수를 c번까지만 제한하는 서명 스킴(c-times 서명스킴)을 제안한다. 최초로 c-times 서명 스킴의 형식적인 정의를 기술하고 일반적인 서명 스킴을 c-times 서명 스킴으로 변환시키는 방법을 소개한다. 그리고 특정한 예로서 (t, n)-임계 기법(Threshold Technique)인 Feldman의 증명가능한 (t, n)-비밀분산 기법을 사용하여 DSA (Digital Signature Algorithm)에 기반한 c-times 서명 스킴인 ^{r}DSA를 제시하고 이 스킴의 안전성을 증명한다. 여기에 적용된 변환방법은 다른 ElGamal형태의 전자서명 스킴에 적용되어질 수 있다.

ABSTRACT

In some practical circumstances, the ability of a signer should be restricted. In group signature schemes, a group member of a group may be allowed to generate signatures up to a certain number of times according to his/her position in the group. In proxy signature schemes, an original signer may want to allow a proxy signer to generate a certain number of signatures on behalf of the original signer. In the paper, we present signature schemes, called c-times signature schemes, that restrict the signing ability of a signer up to c times for pre-defined value c at set-up. The notion of c-times signature schemes are formally defined, and generic transformation from a signature scheme to a c-times signature scheme is suggested. The proposed scheme has a self-enforcement property such that if a signer generates c+1 or more signatures, his/her signature is forged. As a specific example, we present a secure c-times signature scheme ^{r}DSA based on the DSA (Digital Signature Algorithm) by using a threshold scheme. Our transformation can be applied to other ElGamal-like signature schemes as well.

Keyword : *Digital Signature Scheme, Threshold Scheme, Verifiable Secret Sharing*

1. 서 론

1.1 배경

전자 서명은 현대암호의 가장 근본적이고 유용한 창조 가운데 하나이다. Diffie-Hellman 논문[1]에서 트랩도어(trap-door) 함수 모델에 근거한 공개키 암호 시스템을 통하여 전자서명이 가능함이 알려진 이후로 다양한 서명 스킴들이 제안되어져 왔다. 전자서명 스킴은 이전에 생성했던 비밀키와 공개키를

* 고려대학교 정보보호 대학원(videmot@cist.korea.ac.kr),(donghlee, jilim@korea.ac.kr)

가진 사용자가 주어진 메시지에 대해 전자서명을 생성하도록 허락해 준다. 사용자의 공개키를 알고 있는 어느 누구도 서명을 확인할 수 있는 반면 비밀키에 대한 지식 없이 사용자의 서명을 위조하는 것(즉, 이 사용자에 의해 이전에 서명되지 않은 메시지에 서명을 생성하는 것)은 계산적으로 불가능하다.

일반적으로 전자서명은 사용자의 서명 능력에 따라 두 가지 형태로 분류되어진다. 첫 번째 형태의 스킴에서 사용자는 서명 능력에 어떤 제한도 없다. RSA시스템[2]이 최초로 그런 형태로 구현되어진 스킴이다. 두 번째 형태는 생성될 서명의 횟수에 제한이 있는 경우이다. 이런 형태의 서명 스킴에서는 일단 서명의 스킴이 설정되어진 후 서명자는 단지 정해진 횟수인 c번의 서명만 할 수 있다. 이런 형태의 전자서명 스킴의 예로는 일회용 전자서명[3,4]의 일반화된 스킴을 들 수 있다. 일회용 전자서명은 최대 하나의 메시지만 서명할 수 있는 전자서명 스킴이다. 만일 그렇지 않다면 서명은 위조될 수 있다. 일회용 전자서명의 예로는 Merkle의 일회용 전자서명과 Goldwasser, Micali, 그리고 Rivest등이 제안한 일회용 전자서명[8]등이 있다. 전자는 충돌회피 해쉬 함수에 기반하고 있으며 후자는 claw-free permutation의 쌍에 기반하고 있다. 이 일회용 전자서명 스킴들은 인증 트리(authentication tree)와 결합되어 다중 서명을 할 수 있는 서명 스킴으로 구성될 수 있다.

1.1 연구의 동기

여기서 우리는 서명 능력에 제한이 있는 전자서명 스킴들을 c-times 전자서명이라 부르기로 한다. c-times 전자서명은 서명자의 서명 능력이 제한되어져야 하는 상황에 이용되어 진다. 예를 들어 그룹 서명 스킴[5]에서 그룹의 구성원은 그룹 내의 그의 직위 또는 위치에 따라 어떤 정해진 횟수의 서명만 생성하도록 허락되어진다. 이것은 매우 자연스러운 제한이다. 그룹은 계층적인 구조로 구성되어지며 높은 위치의 구성원은 그룹에 대해 낮은 위치의 구성원보다 더 많은 대표성을 가지기 때문이다. 다른 예로는 대리 서명 스킴[6]을 생각해 볼 수 있다. 대리 서명이란 본래의 서명자가 그의 서명 능력을 어떤 대리 서명자에게 위임하는 전자서명 스킴이다. 대리 서명자는 본래의 서명자를 대신하여 서명을 생성한다. 이 경우 본래의 서명자는 제한된 서명 능력을 위임함으로서 대리 서명자를 통제하고 싶을 것이다. c-times 전자서명 스킴은 일상적인 전자서명 시스템을 유지시키면서 서명 능력의 일시적인 제한을 위해서도 사용 가능하다. 이사회나 법원이 불법적인 행동으로 의심받는 일반 서명자의 서명 능력을 제한시킬 필요가 있는 경우 일상적인 전자서명 스킴을 임시적으로 c-times 전자서명 스킴으로 변환시킬 것이다. 만일 사용자가 무죄인 것이 증명되어지면 그 사용자는 제한이 철회됨으로써 예전의 서명 시스템을 다시 사용하도록 허락되어 진다. c-times 전자서명 스킴은 희소성 또는 제한의 가치가 필요한 다양한 상황에 응용되어진다.

본 논문에서는 일반적인 일 방향 함수에 근거한 유한 번 사용 가능한 전자서명 스킴 이외에 DSA 시스템과 같은 트랩도어 일 방향 함수에 근거한 스킴들을 살펴본다. 더욱 폭넓은 응용을 위해 다양한 안전성의 가정(인수분해 문제[2] 또는 이산 대수 문제[7])에 기반하는 스킴의 연구가 필요할 것이다.

1.3 논문에서 제시한 결과

본 논문의 목적은 유한 번 사용가능한 전자서명 스킴의 개념에 대해 이야기하는 것이다. 본 논문은 우선 일회용 전자서명의 일반화된 경우를 포함하는 c-times 전자서명의 형식적인 정의들과 새로운 안전성 모델에 관해 기술한다. c-times 전자서명의 안전성 모델에서 새로운 형태의 공격모델, ^{c}ATK가 요구되어진다. 이 스킴에서 공격자는 단지 c 또는 그 이하의 유효한 서명 값들만 이용하도록 허락되어진다. 이것은 보통의 공격 모델과는 매우 다른 형태이다. 즉, 주어진 스킴의 보안 상수 k에 관해 다항식(polynomial)번의 질문을 허용하는 일반적인 모델에서와는 다르게 미리 정해진 고정된 상수 c번의 질문만을 허용하므로 질문 횟수에 상수번의 제한이 있는 제한된 공격자 모델이 필요하게 된다.

또한, 본 논문에서는 보통의 전자서명을 c-times 전자서명으로 변환시키는 일반적인 방법을 제시한다. 이 방법은 보통의 전자서명과 증명 가능한 (c+1, c+1)-비밀 분산 기법의 합성이다. 이 방법은 제 3의 신뢰기관을 요구하지 않는다. 다시 말해, 서명자가 c+1번 또는 그 이상의 서명을 생성하면 서명자의 비밀키가 노출되어 매우 큰 잠재적 손실에 직면하게 되는 자체적으로 강화된(self-enforcing) 서명 스킴이므로 서명자는 c+1번 또는 그 이상의 서

명을 하지 못하게 된다.

마지막으로, 본 논문에서는 c-times 전자서명 스킴의 특정한 예로 DSA(Digital signature standard)와 Feldman의 증명 가능한 (t, n)-비밀 분산기법에 기반한 ^{c}DSA를 제시한다.

1.4 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 c-times 전자서명을 정의하고 안전성에 관한 새로운 공격모델을 기술한다. 3장에서는 증명 가능한 (t, n)-비밀분산 기법을 형식적으로 기술한다. 4장에서는 c-times 전자서명에 관한 일반적인 구성을 제시하고 이의 안전성을 제시한다. 5장에서는 DSA와 Feldman의 증명 가능한 (t, n)-비밀분산 기법에 기반한 c-times 전자서명의 특정한 예를 제시한다. 그리고 5장에서는 결론을 내린다.

II. 정 의

여기서는 c-times 전자서명의 개념을 형식적으로 기술한다. 다음 정의 중 일부분에 대해서는 세부적으로 Glodwasser, Micali, Rivest가 제시한 논문[8]을 참조하도록 한다.

2.1 기호

- $n \leftarrow_R N$은 집합 N으로부터 원소 n이 임의로(randomly) 선택되어짐을 나타낸다.
- 만일 모든 $c>0$에 대해 $\epsilon(n)<1/n^c$을 만족하는 $n_c>0$가 존재한다면 함수 $\epsilon(n)$은 무시할 수 있는 양이다 라고 말한다.

2.2 전자서명

일반적으로, 전자서명 스킴 S은 3가지 다항식 시간 알고리즘인 키 생성 알고리즘, 서명생성과 서명확인 알고리즘에 의하여 정의된다.

키 생성 알고리즘 KG는 입력으로 1^k(k는 보안상수)을 받고 키 쌍 (sk, pk)을 출력하는 임의성을 갖는 알고리즘이다. 여기서 sk는 서명 생성을 위한 비밀키이고 pk는 서명 확인을 위한 공개키이다.

서명 생성 알고리즘 Sig는 비밀키 sk와 메시지 m을 입력으로 받아 서명 σ를 생성하는 임의성을 갖는 알고리즘이다. 메시지 m은 관련된 메시지 공간 M에 속한다.

서명확인 알고리즘 Ver은 서명 값 s와 공개키 pk를 입력으로 받아 1 또는 0을 되돌려준다. 1을 결과값으로 얻는 경우 서명 s는 메시지 m에 대해 유효하다고 얘기한다.

서명 스킴의 안전성 정의는 [8]의 정의를 이용하기로 한다. 여기서의 안전성 목적은 위조 불가능성이며 UF로 나타낸다. 공격 모델은 선택 메시지 공격이며 CMA로 나타낸다. 공격자 F는 서명생성 알고리즘에 대한 오라클 접근을 적합하게 하도록 허용되어진다. 즉 적합하게 선택된 메시지에 대해 오라클 질문을 할 수 있도록 허용된다. 서명 스킴 S을 깨는 공격자의 이점(advantage) $Adv_{S,F}^{UF-CMA}$은 이전에 허용된 오라클 질문과 같지 않은 메시지에 대해 유효한 서명을 얻을 확률로 정의된다. 만일 보안 상수 k에 대해 다항식 시간의 운영 시간을 갖는 모든 공격자 F에 대해 $Adv_{S,F}^{UF-CMA}$가 무시할 수 있는 양이라면 전자 서명 스킴은 적합하게 선택된 메시지 공격에 대해 안전하다고 한다.

2.3 c-times 전자서명

우선, c-times 전자서명의 개념을 형식적으로 정의하기 전에 '어떻게 서명자의 서명 능력을 제한시킬 수 있는가'에 관해 논의해 보자. 사실상 이 개념은 서명의 횟수가 증가함에 따라서 서명 스킴의 생명이 줄어든다는 의미에서 음성적(negative or maximal) 임계 기법이 된다. 특정한 횟수가 스킴의 생사에 관한 임계점이 된다.

이것을 실현하는 방법에는 여러 가지가 있다. 첫째, 제 3의 신뢰기관을 이용하는 방법이다. (공개키 인증서를 이용하는 방법을 포함할 수 있다.) 둘째, 자체적으로 강화된(Self-enforcing) 특성을 갖는 스킴을 구성하는 방법이다. 위의 첫 번째 방법은 쉽게 서명 횟수의 제한을 가할 수 있는 방법이다. 사실상 유효한 서명의 횟수에 대한 합법적인 한계가 있는 상황에서 초과된 서명 값들 그 자체는 제한이 위반되었다는 증거를 제시해 준다. 하지만 보다 엄격히 서명능력을 제한하기 위하여 초과된 서명행위에 대한 어떤 '벌칙'이 필요해지며 본 논문에서는 정책적인 면이 강한 첫 번째 방법 이외에 자체적으로 강화된 스킴을 구성하여 위의 목적을 기술적으로 실현하는 방법에 대해 생각해 보기로 한다.

이런 의미에서 다음에서는 두 번째 방법의 구성을 고려해 보자. 자체적으로 강화된 서명 스킴을 구성하기 위해서는 최대 임계 기법을 스킴의 안전성과 연계하여 구성할 수 있다. 즉, 특정한 횟수의 서명 후에 주어진 서명 스킴의 안전성을 사라지게 하는 것이다. 다시 말해, 서명 위조가 가능해진다는 뜻이다. 서명 위조는 서명자에게 불법적인 행위(서명횟수의 위반)에 대해 매우 큰 손실을 안겨 줄 잠재적인 벌칙을 제공해 주게 되므로 서명 능력에 제한을 주는 방법으로 적합하게 쓰일 수 있게 된다.

이처럼, 서명 능력을 제한시키기 위해 c-times 서명 스킴은 서명 위조 알고리즘을 자체적으로 포함하고 있어야 한다. 서명 위조 알고리즘은 공개된 c+1개 이상의 서명 값들, 이에 대응하는 메시지들과 공개키를 입력으로 받아 합법적으로 서명을 위조할 수 있다.

다음은 c-times 전자서명을 형식적으로 기술한다.

[정의 1] c-times 전자 서명 스킴

c-times 전자 서명 스킴 ${}^{c}S$은 4가지 알고리즘의 쌍이다: ${}^{c}S=({}^{c}KG, {}^{c}Sig, {}^{c}Ver, Forge_{c+1})$.

- 키 생성 알고리즘. ${}^{c}KG$.
  키 생성 알고리즘 ${}^{c}KG$는 입력으로 1^{k} (k는 보안 상수)을 받고 키 쌍 (sk, pk)을 출력하는 임의성(random)을 갖는 다항식 시간 알고리즘이다. 여기서 sk는 서명 생성을 위한 비밀키이고 pk는 서명 확인을 위한 공개키이다.
- 서명 생성 알고리즘. ${}^{c}Sig$.
  서명 생성 알고리즘 ${}^{c}Sig$는 비밀키 sk와 메시지 m을 입력으로 받아 서명 ${}^{c}\sigma$를 생성하는 임의성을 갖는 다항식 시간 알고리즘이다. 메시지 m은 관련된 메시지 공간 M에 속하고 ${}^{c}\sigma \in \{0,1\}^{*}$이다.
- 서명 확인 알고리즘. ${}^{c}Ver$.
  서명확인 알고리즘 ${}^{c}Ver$은 서명 값 ${}^{c}\sigma$와 공개키 pk을 입력으로 받아 1 또는 0을 되돌려 주는 결정성(deterministic)을 갖는 다항식 시간 알고리즘이다. 1을 결과 값으로 얻는 경우 서명 ${}^{c}\sigma$는 메시지 m에 대해 유효하다고 얘기한다.
- 서명 위조 알고리즘. $Forge_{c+1}$.
  c+1개의 유효한 전자서명 ${}^{c}\sigma_i$, 이에 대응하는 메시지들 m_i, 그리고 공개키 pk를 입력으로 받아 이전에 서명되지 않은 메시지 m'에 대해 새로운 서명을 ${}^{c}\sigma'$을 얻는 결정성을 갖는 다항식 시간 알고리즘이다. 즉,

$$Forge_{c+1}(\{m_i, {}^{c}\sigma_i | 1 \le i \le c+1\}, pk, m') \rightarrow {}^{c}\sigma'.$$

어떤 모델에서는 위의 서명 위조 알고리즘이 확장된 모습으로 다루어질 필요가 있는 경우도 있다. 예를 들어, 서명 위조 알고리즘은 다른 서명 스킴에 대한 위조 알고리즘으로 대체되어 사용될 수 있을 것이다. 예를 들어 대리 서명 스킴에서 대리 서명자는 대리 서명을 생성할 때마다 자신의 또 다른 고유한 비밀키 값을 서명 생성과 함께 쉐어의 일부분으로 분배하는 경우이다. 이 경우 대리 서명자의 능력 제한은 매우 강하게 작용한다.

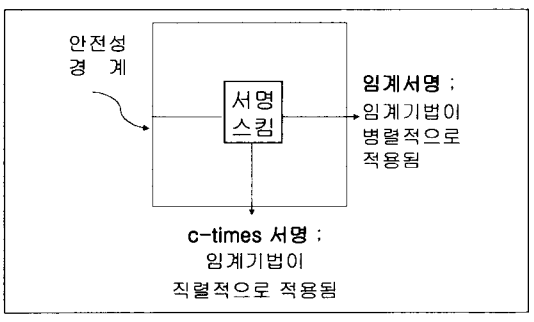
[정의 2] c-times 전자서명의 완전성.

만일 $\forall (sk, pk) \in {}^{c}G(1^{k})$, $m \in M$,

그리고 $\sigma \in {}^{c}Sig_{sk}(M)$에 대하여,

$Forge_{c+1}(\{m_i, {}^{c}\sigma_i | 1 \le i \le c+1\}, pk, m')$가 매우 높은 확률로 성공적인 서명 위조를 한다면, c-times 전자서명 스킴 ${}^{c}S$는 완전하다고 얘기한다.

임계 전자 서명 스킴[9](Threshold signature scheme) 에서는 기반이 되는 서명 스킴에 임계 기법이 병렬적으로 적용되어지는데 반하여 c-times 전자서명에서는 임계기법이 직렬적으로 적용된다. 임계 전자 서명이란 Boyd[10], Desmedt[11], 그리고 Desmedt 와 Frankel[12]에 의하여 소개되어 임계 암호론으로 알려진 일반적인 접근의 한 부분이다. 임계 서명 스킴에서 비밀키는 n명의 참가자에게 분산되어져 있다. 주어진 메시지에 대해 유효한 서명을 생성하기 위해서는 병렬적으로 각 참가자가 그 메시지에 대한 자신의 서명을 생성한 다음 이 후에 완전한 서명이 되도록 개별적인 서명 조각이 결합되어진다. 그러나 c-times 서명 스킴에서는 서명 작업이 직렬적인 모습으로 c번 이루어지게 된다.



〔그림 1〕 전자 서명의 임계성에 대한 일반적인 개념

2.4 c-times 전자서명의 안전성

일반적으로 서명 스킴의 안전성은 두 가지에 관한 문제가 중점적으로 논의된다: 안전성 목적(security goal)과 공격 모델(attack model). 본 논문에서 안전성 목적은 가장 일반적인 개념인 서명 위조 불가능성을 다루기로 하고 이것을 UF로 나타내기로 약속한다. 보통의 전자 서명 스킴에서 보안 상수에 관한 다항식에 의해 제한된 개수의 오라클 질문을 허용하는 공격 모델과는 다르게, c-times 전자 서명 스킴의 공격 모델은 공격자가 서명 오라클에게 미리 정해진 횟수의 질문만을 하도록 허용함으로써 이용할 수 있는 정보의 양에 제한이 있는 모델을 구성하게 된다. 예를 들어, 알려진 메시지 공격에 있어서 미리 정해진 수 c개 이하의 메시지들의 서명 값들이 공격자에게 주어진다. 또한 (적합한) 선택 메시지 공격에서는 공격자가 서명 오라클에 많아야 c번의 질문만을 할 수 있도록 허용되어 진다. 이처럼 c-time 전자서명에서는 새로운 공격 모델인 제한된 공격 모델이 필요함을 알 수 있다. 여기서는 이 공격 모델을 ${}^{c}ATK$으로 나타내기로 한다. 보통의 ATK의 관점에서 안전한 서명 스킴은 ${}^{c}ATK$에 대해서도 안전함을 쉽게 알 수 있다.

사실, c-times 전자서명에서 c값은 주어진 시스템의 안전성을 위하여 기본적으로 주어진 스킴의 보안상수에 관한 다항식에 의하여 제한되어야 한다. 이 후에서는 c에 대한 이런 가정을 계속 사용하기로 약속한다.

다음에서는 c-times 전자서명 스킴 ${}^{c}S=({}^{c}KG, {}^{c}Sig, {}^{c}Ver, Forge_{c+1})$에 대한 공격에서 공격자를 운영하는 실험을 형식적으로 나타낸다.

실험 $\mathrm{Exp}_{{}^{c}S,F}^{UF-{}^{c}CMA}$

$(sk, pk) \leftarrow_R {}^{c}KG$

$(m', {}^{c}\sigma) \leftarrow F^{{}^{c}Sig_{sk}(\cdot)}(pk)$

만일 $Ver(m', {}^{c}\sigma)=1$이고 m'이 이전의 오라클 질문과 같지 않으면 1을 되돌려주고 아니면 0을 되돌려 준다.

F는 서명 오라클 ${}^{c}Sig_{sk}(\cdot)$에 대해 많아야 c번까지 접근할 수 있는 공격자이다. ${}^{c}S$의 공격에 대한 공격자의 '이점'은 다음과 같이 보안상수 k에 관한 함수 Adv로 정의된다.

$$Adv_{{}^{c}S,F}^{UF-{}^{c}CMA}(k) = \Pr[\mathrm{Exp}_{{}^{c}S,F}^{UF-{}^{c}CMA}=1],$$

위의 식에서 확률 값은 실험에서 만들어진 모든 임의의 선택에 대해 얻어진 값이다.

만일 보안 상수 k에 관해 운영시간이 다항식인 모든 공격자에 대해 함수 $Adv_{{}^{c}S,F}^{UF-{}^{c}CMA}(k)$가 무시할 수 있는 양이라면 c-times 전자 서명 스킴 ${}^{c}S$는 $UF-{}^{c}CMA$의 관점에서 안전하다고 말한다.

Ⅲ. 증명 가능한 (t, n)-비밀분산 기법

(t, n)-비밀분산 기법은 비밀을 분배하기 원하는 비밀 분배자와 이 비밀 값의 쉐어(share)를 받는 n명의 쉐어 소유자들로 구성된다. 그리고 비밀 값의 복원은 이 후 그들 가운데 단지 t명 이상의 그룹만이 할 수 있도록 허락되어진다. 증명 가능한 비밀분산(VSS) 기법은 부정한 비밀 분배자의 문제를 극복하기 위해서 Chor, Goldwasser 그리고 Micali에 의해 최초로 제안되었다.[13] VSS 스킴에서 쉐어의 소유자들은 자신의 쉐어에 대한 유효성을 증명할 수 있다. 만일 쉐어의 소유자들이 비밀 분배자 또는 서로의 상호작용 없이 그들의 쉐어의 유효성을 증명할 수 있다면 주어진 스킴 VSS이 비상호작용(non-interactive) 성질이 있는 스킴이라고 말한다.

3.1 증명 가능한 (t, n)-비밀분산 기법

U를 쉐어 소유자들의 집합 $\{P_1, \ldots, P_n\}$이라고 하자. 그리고 Γ_t를 다음과 같이 정의된 접근 구조라 가정하자.

$$\Gamma_t := \{A | A \in 2^{\{1,\ldots,n\}} \wedge |A| \geq t\}$$

여기서 A는 $\{1, \ldots, n\}$의 한 부분집합이며 A의 원소의 개수는 t 이상이다. 따라서 위의 접근구조로부터 ((t, n)-임계 기법의 특성상) t 명 이상의 쉐어 소유자들 P_{i_m} $(1 \leq m \leq t)$의 모임에 의해서만 비밀 값에 대한 접근, 즉 비밀 값의 복구가 허용됨을 알 수 있다. 증명 가능한 (t, n)-비밀분산 기법 $(1 \leq t \leq n)$ 은 4개의 다항식 시간 알고리즘의 쌍 $(PG, Share, Verify, Recover)$으로 구성된다.

- 공개 및 비밀 파라미터 생성 알고리즘.

PG는 입력으로 1^{λ}(λ는 보안 상수)을 받고 공개 및 비밀 파라미터 (sec, pp)를 출력하는 임의성(random)

을 갖는 다항식 시간 알고리즘이다. 여기서 sec는 참가자들에게 분배될 비밀값이고 pp는 쉐어의 정당성을 확인하기 위한 공개 파라미터이다.

- 쉐어 생성 및 분배 알고리즘.
  *Share*는 비밀값 sec를 입력으로 받아 n개의 쉐어 값들 $\{s_1, \ldots s_n\}$을 생성하는 임의성을 갖는 다항식 시간 알고리즘이다. 세부적으로 표현하면 $\forall i$ $(1 \le i \le n)$, $u \leftarrow_R \{0,1\}^*$, $s_i \leftarrow share(\sec, u)$이다. 앞으로 한 개의 쉐어를 계산하는 표현은 다음과 같이 $s_i \leftarrow share(\sec, u)$ 간단히 표현하기로 한다. (경우에 따라서는 u값이 s_i에 포함되어진다.)
  이 후, 비밀 분배자는 각 쉐어를 n명의 참가자들 P_i에게 안전하게 분배한다.
- 쉐어 검증 알고리즘.
  *Verify*는 쉐어 값 s_i와 공개 파라미터 pp을 입력으로 받아 '참' 또는 '거짓'을 되돌려 주는 결정성(deterministic)을 갖는 다항식 시간 알고리즘이다. '참'을 결과 값으로 얻는 경우 쉐어 s_i는 유효하다고 얘기한다.
- 비밀 값 복구 알고리즘.
  *Recover*는 c+1개의 유효한 쉐어들의 모임을 입력으로 받아 최초의 비밀값 sec를 복구해주는 결정성을 갖는 다항식 시간 알고리즘이다.

3.2 증명 가능한 (t, n)-비밀분산 기법의 안전성

여기서는 증명 가능한 (t, n)-비밀분산 기법의 안전성에 대해 기술한다.

만일 (t, n)-비밀분산 기법이 다음의 성질들을 만족하면 각각 정확성과 증명가능성을 갖는다고 말한다.

- (정확성) $Share(\sec) = (s_1, \ldots, s_n)$이면
  $\forall i \in A(\in \Gamma_t)$, $Verify(s_i) = \text{true}$ 이다.
- (증명가능성) $\forall A \in \Gamma_t : \forall i \in A$,
  $Verify(s_i) = \text{true}$ 이면
  $Recover(\{s_i | i \in A\}) = \sec$ 이다.

(t, n)-비밀분산 기법이 안전하다는 뜻은, 직관적으로, $\forall A' \notin \Gamma_t$에 대해 집합 $\{s_i | i \in A'\}$으로부터 비밀값을 복구하는 것은 계산적으로 불가능해야 함을 의미한다. 다시 말해, 접근 구조에 의해 정의된 정당한 쉐어의 모임에 의해서만 비밀 값이 복구되어진다는 의미이다. 이런 비밀값 복구 불가능에 대한 안전성 목적을 $URec$으로 나타내기로 한다. 여기서 다항식 시간 공격자 F는 공개 파라미터 pp를 입력으로 받아 쉐어 생성 오라클 $share(\sec, \cdot)$에 많아야 $t-1$번까지 질문을 할 수 있다. (이것을 CSA(선택 쉐어 공격)로 나타내기로 한다.) 다시 말해, 공격자는 $t-1$개의 유효한 쉐어를 얻을 수 있다.

다음에서는 (t, n)-비밀분산 기법 (PG, $Share$, $Verify$, $Recover$)에 대한 공격에서 공격자를 운영하는 실험을 형식적으로 나타낸다.

$$\begin{aligned} &\text{실험 } \mathrm{Exp}^{URec-CSA}_{(t,n)-VSS,F} \\ &\quad (\sec, pp) \leftarrow_R PG \\ &\quad s' \leftarrow F^{share(\sec,\cdot)}(pp) \end{aligned}$$

만일 $s' = \sec$ 이면 1을 되돌려주고 아니면 0을 되돌린다.

$(t, n) - VSS$의 공격에 대한 공격자의 '이점'은 다음과 같이 보안상수 k에 관한 함수 Adv로 정의된다.

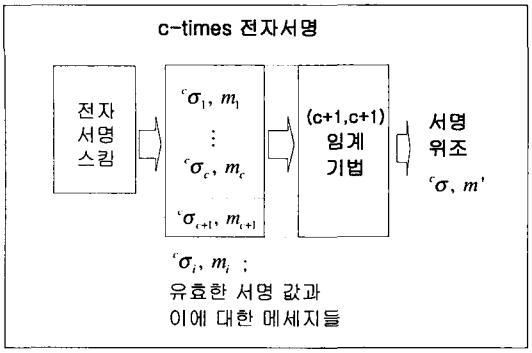
$$Adv^{URec-CSA}_{(t,n)-VSS,F}(k) = \Pr[\mathrm{Exp}^{URec-CSA}_{(t,n)-VSS,F} = 1]$$

위에서 정의된 확률 값은 실험에서 만들어진 모든 임의의 선택에 대해 얻어진 값이다. 만일 보안 상수 k에 관해 운영시간이 다항식인 모든 공격자에 대해 $Adv^{URec-CSA}_{(t,n)-VSS,F}(k)$가 무시할 수 있는 양이라면 $(t, n) - VSS$는 $URec - CSA$의 관점에서 안전하다고 말한다.

Ⅳ. c-times 전자서명의 일반적인 구성

여기서는 주어진 임의의 전자서명 스킴을 c-times 전자서명으로 변환시키는 방법을 살펴보기로 한다. 서명자의 능력에 대한 제한을 얻는 방법은, 비형식적으로 말하면, 공개된 서명 값들, 이에 대응되는 메시지들 그리고 공개키에 대해 임계기법을 적용시키는 것이다([그림 2] 참조).

본 논문에서 제시할 방법을 구체적으로 말하면, 비밀분산 기법을 메시지에 적용시키는 것이다. 즉, 서명될 메시지는 서명자의 비밀키를 분산시키는 데에 이용되며 따라서 c-times 전자서명 스킴의 서명 값은 주어진 메시지에 대한 서명 값과 비밀키의 쉐어로 구성된 쌍이 된다. 생성된 서명의 횟수의 한계는 주어진 서명 값 내의 비밀 값의 쉐어를 사용하여 VSS의 증명 알고리즘을 통하여 공개적으로 증명되



〔그림 2〕 c-times 전자서명의 구성 방식

어진다. 서명자가 서명을 생성할 때마다 비밀키의 쉐어 하나가 동시에 생성되며 이것은 서명이 한번 생성되었음을 공개적으로 나타내어 준다. 만일 정해진 수 이상의 유효한 쉐어가 모이면 서명자의 비밀키 정보가 *VSS*의 복구 알고리즘에 의하여 노출되게 된다.

4.1 일반적인 구성 방법

여기서는 c-times 전자서명 스킴의 일반적인 구성방법을 제시하고 이의 안전성을 증명한다.

4.1.1 스킴의 구성

먼저, 서명 스킴 $S=(G, Sig, Ver)$, 랜덤함수 Ran, 그리고 증명가능한 $(c+1, c+1)$-비밀분산 스킴 $VSS=(PG, Share, Verify, Recover)$이 주어졌다고 가정하자. c-times 전자서명 스킴 $^cS=(^cKG, ^cSig, ^cVer, Forge_{c+1})$은 다음과 같이 구성할 수 있다.

- 키 생성 알고리즘, ^cG.
  주어진 입력 1^k에 대하여(여기서 k는 보안상수이다.), 알고리즘 ^cG는 서명 스킴 S의 키 생성 알고리즘 G을 이용하여 주어진 서명 스킴을 위한 비밀키와 공개키 쌍 (sk, pk)을 생성한다. 그리고 다음에는 비밀분산 스킴 VSS의 키 생성 알고리즘 PG를 이용하여 쉐어의 유효성을 검증하는데 필요한 공개 파라미터 pp를 생성하고 비밀값은 비밀키 sk로 정의한다. c-times 전자서명의 공개키는 (pk, pp)이며 비밀키는 sk이다.
- 서명 생성 알고리즘, ^cSig.
  비밀키 sk와 메시지 m을 입력으로 받아 다음과 같이 σ와 nt를 계산한다.
  $\sigma = Sig(sk, m)$, $nt = Sihare(sk, Ran(m))$. c-times 전자서명 값은 $^c\sigma=(\sigma, nt)$이다.
- 서명 확인 알고리즘, ^cVer.
  공개키 sk, 메시지 m, 그리고 이에 대응하는 서명 $^c\sigma=(\sigma, nt)$를 입력으로 받아 다음과 같이 두 가지 알고리즘을 이용하여 유효성을 검증한다. 만일 $Ver(\sigma, m)=1$과 $Verify(pp, nt, Ran(m))=1$이 면 유효한 서명 값이 된다.
- 서명 위조 알고리즘, $Forge_{c+1}$.
  서명 위조 알고리즘은 VSS의 $Recover$ 알고리즘을 이용하여 $c+1$개의 유효한 서명 값 $^c\sigma=(\sigma, nt)$들과 이에 대응하는 메시지들을 입력으로 받아 비밀키 값 sk를 계산한다.

$$Recover(I)\rightarrow sk,$$
$$I=\{(m_i, {}^c\sigma_i) | 1\le i\le c+1, (m_i, {}^c\sigma_i)\neq(m_j, {}^c\sigma_j)\}.$$

제시한 c-times 전자서명의 안전성을 위하여, 주어진 메시지 m과 쉐어 값 nt를 독립적으로 만들기 위하여 랜덤 함수가 필요하다. 일반적으로, 주어진 전자서명이 메시지 해쉬 함수를 사용하는 경우, 이 해쉬 함수는 그 특성상 랜덤 함수의 역할을 하기에 충분하다.

쉐어 값을 계산하기 위해 제시한 함수은 메시지의 서명 값이 아닌 메시지에 대한 함수이다. 이것은 주어진 서명 스킴에 의존하지 않으므로 후자가 더 일반적임을 알 수 있다. 그리고, 서명과 쉐어를 동시에 계산할 수 있다.

[정리 1]
위에서 제시한 c-times 전자서명은 정의 2의 완전성을 만족한다.

증명은 직접적이다. 주어진 c+1개의 서명 값과 이에 대응하는 메시지 m에 대하여 비밀키 sk의 유효한 c+1개의 쉐어들을 얻을 수 있으며 충돌 쌍이 발생할 확률은 매우 작으며 서로 다른 c+1개의 쉐어들에 대해 $Recover$알고리즘을 통하여 서명자의 비밀키 sk를 복구할 수 있다.

4.1.2 제시한 c-times 서명스키의 안전성

만일 안전하지 않은 $(c+1, c+1)$- VSS 또는 랜덤 함수 또는 전자서명 스킴 S가 사용된다면 제시한 스킴이 안전하지 않다는 것은 자명하다. 다음 정리는 ^cS

의 안전성이 이 스킴을 구성하기 위해 주어진 서명 스킴 S, 랜덤 함수 Ran, $(c+1,c+1)$- VSS의 안전성보다 약하지 않음을 보여준다.

[정리 2]

만일 서명 스킴 S, 랜덤 함수 Ran, $(c+1,c+1)$- VSS이 안전하다면 위의 제시한 c-times 서명 스킴 ^{c}S은 안전하다.

(증명)

증명의 중심 생각은 랜덤 함수의 사용에 있다. 랜덤 오라클 모델하에서 Ran함수는 독립적인 랜덤성을 나타내며 이 조건하에서 주어진 스킴의 안전성을 기술한다. 비형식적으로 말하자면, 제시한 c-times 서명 스킴에 대한 서명 위조는 주어진 서명 스킴에 대한 서명 위조 또는 $(c+1,c+1)$- VSS 안전성을 깨는데 이용될 수 있다. 다음에서는 이 증명에 대해 형식적인 기술을 한다. 증명의 기법은 바로 위에서 언급한 논의와 같이 대우 명제를 이용하기로 한다.

먼저, 공격자 F'가 무시할 수 없을 확률 ε으로 ^{c}S의 $UF-{}^{c}CMA$ 안전성을 깬다고 가정한다. 그리고, 일반성을 잃지 않고, 공격자 F'는 ^{c}S에 대해 c개의 오라클 질문을 만든다고 가정한다. m_i를 공격자 F'이 서명 받기 원하는 i $(1\le i\le c)$번째 메시지라 하고 (σ_i, nt_i)를 이 메시지의 서명이라고 하자. 여기서 $\sigma_i \leftarrow Sig_{sk}(m_i)$ 이고 $nt_i \leftarrow Share(sk, Ran(m_i))$ 이다. (σ, nt)와 m을 공격자 F'이 출력하는 값이라고 가정한다. 여기서 $m\notin\{m_i|1\le i\le c\}$, (σ, nt)는 m에 대응하는 위조된 서명이다. 마지막으로 $\sigma\notin\{\sigma_i|1\le i\le c\}$의 사건을 Forged로 나타내기로 한다. 그러면 공격자 F'이 성공할 확률은 다음과 같음을 알 수 있다.

$$\varepsilon < \Pr(F' \text{ succeeds}) = \Pr(F' \text{ succeeds} \wedge Forged) + \Pr(F' \text{ succeeds} \wedge \neg Forged).$$

따라서 최소한 위의 확률 가운데 한 개는 $\varepsilon/2$ 보다 크거나 같다. 다음에서는 첫 번째 확률 값이 $\varepsilon/2$보다 크다면 S가 $UF-CMA$의 관점에서 안전하지 않음을 보이고 반면에 두 번째 확률 값이 $\varepsilon/2$보다 크다면 $(c+1,c+1)$- VSS가 $URec-CSA$관점에서 안전하지 않음을 보인다.

- 경우 1. $\Pr(A' \text{succeed} \wedge Forged) \ge \varepsilon/2$.

이 경우에는 주어진 공격자 F'를 이용하여 서명 스킴 S의 안전성을 깨는 공격자 F_{sig}을 구성한다. F_{sig}는 자체적으로 임의의 비밀 값 x와 공개 파라미터 pp를 선택한다. 다음에 F_{sig}는 F'의 메시지 m_i에 대한 질문을 먼저 자신의 서명 오라클 $Sig_{sk}(\cdot)$에게 요청하여 서명 σ_i을 받은 후 쉐어 생성 알고리즘을 이용하여 쉐어 값 nt_i을 생성하여 (σ_i, nt_i)를 되돌려 줌으로써 F'을 시뮬레이트(simulate)한다. 여기서 주목할 점은 x가 서명 생성키 sk와 다르다는 점이다. 사실, x는 비밀키 sk와는 독립적이다. 하지만 이런 환경은 Ran함수가 랜덤 오라클처럼 행동하는 특성 때문에 공격자 F'의 이점을 감소시키지 않음을 알 수 있다. 공격자 F'이 새로운 메시지 m에 대해 ^{c}S에 관한 위조된 서명 $^{c}\sigma=(\sigma, nt)$을 출력 했을 때 공격자 F_{sig}은 새로운 메시지 m에 대해 S에 관한 위조된 서명 σ을 출력하게 된다. 여기서 $Forged$의 사건이 발생했다는 사실과 σ가 서명 스킴 S에 관한 새로운 서명이 된다는 사실과 동치임을 주목하자. 따라서 공격자 F_{sig}은 최소한 $\varepsilon/2$(무시할 수 없는 양)의 확률로 성공하게 되며 이것은 S의 $UF-CMA$안전성에 관한 모순을 이끌어 준다.

- 경우 2. $\Pr(A' \text{succeeds} \wedge \neg Forged) \ge \varepsilon/2$.

이 경우에는 주어진 공격자 F'를 이용하여 $(c+1, c+1)$- VSS의 안전성을 깨는 공격자 F_{VSS}를 구성한다. 이 공격자 F_{VSS}는 어떤 c개의 유효한 쉐어에 대해서도 비밀 값을 복구할 수 있다. F_{VSS}는 자체적으로 서명 스킴 S의 임의의 비밀키, 공개키 쌍 (sk, pk)을 선택한다. 다음에 F_{VSS}는 F'의 메시지 m_i에 대한 질문을 먼저 자신의 쉐어 생성 오라클 $share(\text{sec}, Ran(m_i))$에게 요청하여 쉐어 값 nt_i을 얻고 서명 생성 알고리즘을 이용하여 서명 값을 생성함으로써 시뮬레이트(simulate)한다. 여기서 주목할 점은 비밀 값 x가 서명 생성키 sk와 다르다는 점이다. 사실, x는 비밀키 sk와는 독립적이다. 하지만 이런 환경은 Ran함수가 랜덤 오라클처럼 행동하는 특성 때문에 공격자 F'의 이점을 감소시키지 않음을 알 수 있다. 공격자 F'이 새로운 메시지 m에 대해 ^{c}S에 관한 위조된 서명 $^{c}\sigma=(\sigma, nt)$을 출력 했을 때 공격자 F_{VSS}는 VSS에 관한 쉐어 값 nt을 얻게 된다. 이 경우 사건은 nt가 이전의 오라클 질문으로 얻은 값들 중에 어떤 nt_i와 같은 경우와 같지 않은 경우로 나누어질 수 있고 전자의 사건을 $Equal$로 나타내고 후자를 $\neg Equal$으로

나타내기로 하자. 만일 사건 ¬*Equal*이 일어났을 경우 공격자 F_{VSS}는 주어진 공격자 F'의 성공 확률 $\varepsilon/2$로 비밀값을 복구하게 된다. 반면에 *Equal*이 일어난 경우 공격자 F_{VSS}는 실패하게 된다. 하지만 이 사건이 발생할 확률 δ은 *Ran*함수의 랜덤성 때문에 $c/2^{\lambda}$이다. (여기서 λ는 (c+1,c+1)- *VSS*의 보안 상수이다.) c값은 시스템 초기 설정단계에서 스킴의 안전성을 위해 보안상수에 제한되어지므로 이 값은 무시할 수 있을 정도의 양으로 가정할 수 있고 공격자 F_{VSS}의 성공확률은 $\varepsilon/2-\delta$(무시할 수 없는 양)가 된다. 따라서 이것은 (c+1, c+1)- *VSS*의 *URec- CSA*안전성에 관한 모순을 이끌어 준다. □

기존의 서명 스킴에 'c-times'의 제한이 사용될 때 마다, 비밀키에 대한 새로운 접근구조가 이용된다. 따라서 'c-times'의 중복 사용은 주어진 전자서명의 안전성을 손상 시키지 않는다.

V. c-times 전자서명 스킴의 예

이 장에서는 DSA(Digital Signature Standard)[13]와 Feldman의 증명 가능한 (t, n)-비밀분산 기법[15]에 기반한 c-times 전자서명 스킴을 기술한다.

5.1 Digital Signature Standard(DSS)

Digital Signature Algorithm(DSA)은 미국의 표준 전자서명 알고리즘으로 채택된 ElGamal 전자서명[16] 기반의 전자서명 스킴이다. 다음에는 DSA에 대해 간략히 기술한다.

[DSA]

- 시스템 파라미터 : 시스템 구성을 위한 공개 파라미터는 다음과 같다.
  (1) p는 크기 n_1비트의 소수이다. q는 크기 n_2비트의 소수이고 $p-1$의 약수이다. (n_1과 n_2는 시스템의 안전성에 관계하는 보안 상수이다. 현재 n_1은 1024비트 n_2는 160비트면 안전한 것으로 여겨진다.)
  (2) g는 위수 q인 Z_p^*의 부분군의 생성원이다.

- 키 생성 알고리즘 :
  (1) 서명자의 비밀키 x는 1과 q 사이의 임의의 값이다.
  (2) 서명자의 공개키는 $y=g^x \pmod p$이다.

- 서명 생성 알고리즘 :
  (1) 주어진 메시지 m에 대하여 서명자는 난수 $k \in Z_q^*$를 선택하고 $r=((g^k) \bmod p) \bmod q$과 $s=k^{-1}(h(m)+xr) \bmod q$을 계산한다. 여기서 h는 해쉬 함수를 나타낸다.
  (2) 메시지 m에 대한 서명은 (r, s)이다.

- 서명 확인 알고리즘 :
  (1) 주어진 메시지 m에 대한 서명 (r, s)은 다음을 검증하여 공개적으로 증명된다.
  $$r=((g^{h(m)s^{-1}} y^{rs^{-1}}) \bmod p) \bmod q.$$

5.2 Feldman의 증명 가능한 (t, n)-비밀분산 기법

Feldman의 증명 가능한 (t, n)-비밀분산 기법은 Shamir의 (t, n)-비밀분산 기법[17]을 확장한 스킴이다. 다음에서는 t, n을 모두 $c+1$로 가정한다.

[Feldman의 증명 가능한 $(c+1, c+1)$-비밀분산 기법]

- 시스템 파라미터 : 시스템 구성을 위한 공개 파라미터는 다음과 같다.
  (1) p는 크기 n_1비트의 소수이다. q는 크기 n_2비트의 소수이고 $p-1$의 약수이다. (n_1과 n_2는 시스템의 안전성에 관계하는 보안 상수이다. 현재 n_1은 1024비트 n_2는 160비트면 안전한 것으로 여겨진다.)
  (2) g는 위수 q인 Z_p^*의 부분군의 생성원이다.

- 시스템 초기화 :
  (1) 비밀 값 x는 1과 q 사이의 임의의 값이다.
  (2) 비밀 분배자는 임의의 c차 다항식 $f(z)$을 선택한다.
  $$f(z)=a_0+a_1 z+\ldots+a_c z^c,\ f(0)=a_0=x$$
  (3) 검증식은 $F(z)=\prod_{i=0}^{c} b_i^{z^i}$이다.
  (4) $b_i=g^{a_i}$ $(0 \le i \le c)$은 공개된다.

- 비밀 분배 : 비밀 분배자 P는 참가자 P_j들에게 비밀값 x의 쉐어 $w_i=f(j)$를 보낸다.

- 쉐어의 정당성 확인 : 각 참가자 P_j는 쉐어 값이 검증식 $g^{w_j}=F(j)$을 만족하는지 확인하고 만일 성립하지 않으면 분배자에게 이 사실을 알린다.

- 비밀 복구 : 주어진 정당한 $c+1$개의 쉐어 값을 통해 c차 다항식 $f(z)$가 다시 구성되어지며 $f(0)=x$을 계산하여 비밀값이 복구되어진다.

  주어진 $\{(j, f(j))|1\le j\le c+1\}$에 대해,
  Lagarnge 교항 다항식을 이용하면
  $x=f(0)=\sum_{i=1}^{c+1} f(i) \prod_{j=1, j\neq i}^{c+1} \frac{0-j}{i-j} \bmod q$ 이다.

비밀값 x에 대해, c 또는 그 이하의 쉐어 값을 아는 공격자는 g^x으로부터 얻을 수 있는 정보 이상의 더 많은 정보를 얻지 못함을 쉽게 알 수 있다. Feldman의 증명 가능한 (t, n)-비밀분산 기법의 구체적인 안전성은 [15]를 참조하기로 한다.

5.3 DSA와 Feldman의 (t, n)-VSS에 기반한 c-times 전자서명 스킴

여기서는 DSA와 Feldman의 증명 가능한 (t, n)-비밀분산 기법에 기반한 c-times 전자서명 *^{c}DSA*을 제안한다. 키 생성 알고리즘에서 보통의 DSA의 서명 값을 확인하는 공개키 이외에 쉐어를 확인하기 위해 부가적인 공개키 값들이 생성된다. 서명 생성 알고리즘은 보통의 DSA 전자서명과 메시지의 해쉬 값을 c차 다항식에 넣어 얻은 함수 값을 서명 값으로 출력한다. 사실, 이 함수 값은 Feldman의 증명 가능한 (t, n)-비밀분산 기법을 통해 얻은 비밀키의 쉐어이고, 이 쉐어 값은 서명이 "1 번" 생성되었음을 나타내어 준다. 합법적인 서명 위조 알고리즘은 Lagarnge 교항 다항식을 이용한 비밀키 복구 알고리즘이 사용된다.

[*^{c}DSA* 스킴]

- 시스템 파라미터 : 시스템 구성을 위한 공개 파라미터는 다음과 같다.
  (1) p는 크기 n_1비트의 소수이다. q는 크기 n_2비트의 소수이고 $p-1$의 약수이다. (n_1과 n_2는 시스템의 안전성에 관계하는 보안 상수이다. 현재 n_1은 1024비트 n_2는 160비트면 안전한 것으로 여겨진다.)
  (2) g는 위수 q인 Z_p^*의 부분군의 생성원이다.

- 키 생성 알고리즘 :
  (1) x는 1과 q 사이의 임의의 값으로 서명자의 비밀키이다.
  (2) a_i $(1\le i\le c)$는 1과 q 사이의 임의의 값으로 서명자의 비밀 값이다.
  (2) 서명자의 공개키는 $y=g^x \pmod p$와 $b_i=g^{a_i} \pmod p$ $(1\le i\le c)$ 이다.

- 서명 생성 알고리즘 :
  (1) 주어진 메시지 m에 대하여 서명자는 난수 $k\in Z_q^*$를 선택하고 $r=((g^k) \bmod p) \bmod q$과 $s=k^{-1}(h(m)+xr) \bmod q$을 계산한다. 여기서 h는 해쉬 함수를 나타낸다.
  (2) $f(w)=x+a_1w+a_2w^2+\ldots+a_cw^c \bmod q$를 계산한다. 여기서 $w=h(m)$이다.
  (3) 메시지 m에 대한 서명은 $\sigma=((r, s), f(w))$이다.

- 서명 확인 알고리즘 :
  (1) 주어진 메시지 m에 대한 서명 $((r, s), f(w))$은 다음을 검증하여 공개적으로 그 유효성이 증명된다.

$$r=((g^{h(m)s^{-1}}y^{rs^{-1}}) \bmod p) \bmod q.$$
$$g^{f(w)}=yb_1^w b_2^{w^2}\ldots b_c^{w^c} \bmod p.$$

- 서명 위조 알고리즘 :
  (1) 주어진 유효한 $c+1$개의 서명 값에 대해, Lagrange 교항 다항식을 이용하여 서명자의 비밀키 값을 복구한다. 즉,

$$\{(\sigma_i, m_i)|1\le i\le c, \sigma_i=((r_i, s_i), f(h(m_i)))\},$$
$$x=f(0)=\sum_{i=1}^{c+1} f(h(m_i)) \prod_{j=1, j\neq i}^{c+1} \frac{0-h(m_j)}{h(m_i)-h(m_j)} \pmod q$$

  (2) 비밀키의 복구 후 서명 알고리즘을 사용하면 합법적인 서명 위조를 할 수 있다.

주어진 서명 값의 유효성을 검증하기 위해서는 서명의 확인 단계에서 c개의 베이스를 사용하는 법 지수승 연산이 필요하다. 이 경우 동시 지수승(simul-

taneous exponentiation) 연산법[18]을 이용하면 매우 효율적으로 값을 계산할 수 있다.

[따름 정리]

cDSA는 안전한 c-times 전자서명 스킴이다.

제시한 스킴이 정의 2의 완전성 성질을 만족함은 자명하다. DSA에 사용된 해쉬 함수의 충돌회피 성질을 이용하면 서명자가 주어진 서로 다른 메시지에 대해 같은 해쉬 값을 얻을 확률은 생일역설 원리를 이용하면 많아야 $c(c-1)/2q$이고 무시할 수 있을 정도의 양이기 때문이다. 랜덤 함수의 역할을 일반적으로 랜덤성을 갖는 것으로 여겨지는 해쉬 함수로 대치하면 4장의 증명 방법을 따르므로 *cDSA*의 안전성을 보장하게 된다.

Ⅵ. 결 론

c-times 전자서명 스킴은 서명자의 능력이 제한되어야 하는 상황에 필요한 유용한 도구이다. 본 논문에서는 서명자가 미리 정해진 횟수의 서명만을 할 수 있는 c-times 전자서명의 형식적인 정의를 하고 이에 대한 새로운 안전성 개념을 제시했다. 이 안전성 모델에서는 일반적인 서명 스킴과는 다르게 공격자가 어떤 정해진 횟수 이상의 오라클 질문을 할 수 없는 제한된 공격능력을 가진다.

c-times 전자서명의 특정한 예로 DSA와 Feldman의 증명 가능한 (t, n)-비밀분산 기법에 기반한 cDSA을 제안하였다. DSA에 근거한 특정한 예를 제시하였지만 기본적인 접근 방법은 이산대수를 이용하는 ElGamal 형태의 전자서명에 적용가능하다.

기존의 일반적인 일 방향 함수에 기반한 일회용 전자서명 또는 이의 다중(multiple) 사용 형태가 효율성에 장점이 있는 반면 본 논문에서 고려한 스킴은 응용성에 중점이 두어졌다. 예를 들어, 이산대수에 기반한 기존의 특정 목적을 달성하는 전자서명에 c-times 규칙을 적용하기 위해 제시된 방법이 이용될 수 있을 것이다.

제시된 스킴에서 앞으로 개선시켜야 할 문제는 서명의 횟수를 나타내는 c 값이 증가함에 따라 공개키 값이 선형적으로 증가한다는 것이다. 효율성을 위해 공개키 확장이 일어나지 않는 스킴의 연구가 필요하다.

참 고 문 헌

[1] W. Diffie and M.E. Hellman, "New directions in Cryptgraphy", *In IEEE Trans. Info. Theory*, volume IT-22, No. 6, pp. 644~654, November 1976.

[2] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem", *communications of the ACM*, Vol. 21, No. 2. pp. 120~126, 1978.2

[3] D. Bleichenbacher and U. M. mauer, "Directed acyclic graphs, one-way functions and digital signatures", In Crypto '94, Lecture Notes in Computer Science No. 839, *Springer-Verlag*, pp. 75-82, 1994.

[4] L. Lamport, "Constructing digital signatures from a one-way function", *Technical Report SRI Intl. CSL* 98, 1979.

[5] D. Chum, "Group Signatures", Advances in Cryptology-EUROCRYPT'91 Proceeding, *Springer-Verlag*, pp. 257~265, 1991.

[6] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature", *Proceedings of the 1995 Symposium on Cryptography and Information Security*, Inuyama, Japan, pp. B1.1.1-17, Jan 1995.

[7] C. P. Schnorr, "Efficient identification and signatures for smart cards", In Crypto '89, Lecture Notes in Computer Science No. 435, *Springer-Verlag*, pp. 239~252, 1990.

[8] S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks", *SIAM Journal of Computing*, Vol. 17, No. 2, pp. 281~308, April 1988.

[9] S. Langford, "Threshold DSS signatures without trusted party", *In Crypto'95, Lecture Notes in Computer Science No. 963, Springer-Verlag*, pp. 397~409, 1995.

[10] C. Boyd, "Digital Multisignayures", *In H. Baker and F. Piper, editor, Cryptography and Coding, Claredon Press*, pp. 241~246.

1986.
[11] Y. Desmedt, "Society and group oriented cryptography: A new concept", In Crypto '87, Lecture Notes in Computer Science No. 293, *Springer-Verlag*, pp. 120~127, 1988.
[12] Y. Desmedt and Y. Frankel, "Threshold cryptosystem", In Crypto'89, Lecture Notes in Computer Science No. 435, *Springer-Verlag*, pp. 307~315, 1990.
[13] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultanity in the presence of faults", *In Proceedings of 26th IEEE Symposium on the Foundations of Computer Science(FOCS)*, pp. 383~395, 1985.
[14] National Institute for Standard and Technology, *Digital Signature Standard (DSS) Technical Report 169*, August 30 1991.
[15] P. Feldman, "A Practical Scheme Non-Interactive Verifiable Secret Sharing", *In Proc. 28th IEEE Symp. on Foundations of Comp. Science*, pp. 427~437, 1987.
[16] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. Info. Theory, IT31*, 1985.
[17] A. Shamir, "How to Share a Secret", *Communications of the ACM*, pp. 612~613, 1979.
[18] B. Moller, "Algorithms for multi-exponentiation," In SAC2001, Lecture Notes in Computer Science No. 2259, *Springer-Verlag*, 2001.

〈著者紹介〉

황 정 연 (Jung Yeon Hwang) 학생회원
1999년 2월 : 고려대학교 수학과 학사
2001년 3월~현재 : 고려대학교 정보보호대학원 석사과정
〈관심분야〉 공개키 암호 알고리즘, 암호 프로토콜

이 동 훈 (Dong Hoon Lee) 정회원
1984년 2월 : 고려대학교 경제학과 학사
1987년 2월 : Oklahoma Univ. 전산학 석사
1992년 2월 : Oklahoma Univ. 전산학 박사
1993년 3월~현재 : 고려대학교 전산학과 정교수
2000년 3월~현재 : 고려대학교 정보보호대학원 교수
〈관심분야〉 암호이론, 암호 프로토콜, 정보이론

임 종 인 (JongIn Lim) 정회원
1980년 2월 : 고려대학교 수학과 학사
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 : 고려대학교 수학과 박사
1986년 3월~현재 : 고려대학교 수학과 정교수
1999년 2월~현재 : 고려대학교 자연과학대학 정교수, 고려대학교 정보보호 대학원 원장, 고려대학교 정보보호 기술센타장
〈관심분야〉 블록 암호 및 스트림 암호 분석 및 설계, 암호 프로토콜, 공개키 암호 알고리즘 분석, 스테가노 그래피