通信情報保護學會誌
第4卷 第2號, 1994. 6

안전하고 고속적인 디지탈 서명을 위한 병렬 알고리즘 설계

서장원*, 문필주*, 방혜자**, 전문석***, 이철희****

요 약

본 논문은 예전의 방법들에서 일어나고 있는 수행 속도 문제점들을 향상하기 위하여 병렬 처리를 이용하여 난수를 발생하는 방법중에서 가장 보편적이고 빠른 방법으로 알려진 저차 합동 다항식에 기초한 새로운 고속 디지탈 서명방식에 대한 병렬 알고리즘을 제안한다. 새로운 디지탈 서명 방식은 비밀키로써 큰 소수 p,q를 이용하여, 공개 정보로써 $n=p^{2}*q$를 이용한다. 난수는 서명을 생성할때 이용되며, 서명을 검증하기 위하여 부등식을 이용하며 병렬 알고리즘을 이용하여 서명을 생성하는 처리속도의 향상된 성능을 위하여 전처리와 디지탈 서명을 구축하는 계산방법의 새로운 병렬 알고리즘을 작성하였다. 본 논문에서 새로 제안한 서명방식에 대한 병렬 알고리즘을 검증하고 비도를 산출할 것이며, 시뮬레이션을 통하여 예전의 방식들과 비교 분석한다.

본 논문은 공개키를 이용한 병렬 암호와 시스템과 신호 처리에 대한 병렬 알고리즘으로 응용될 수 있을 것이며, 병렬과 분산 처리 환경하에서 개발되는 정보서비스 특히 메세지 처리 시스템 서비스, 전자 교환 서비스 등의 디지탈 서명에 유용하게 이용될 수 있을 것이다.

1. 서 론

컴퓨터 기술 및 통신 기술의 발달은 고도의 정보통신망을 이룩하는데 지대한 공헌을 하였다. 고도의 정보통신망의 실현은 인류에게 유용한 정보를 신속하고 정확하게 처리하여 제공할 수 있다. 그러나 통신망을 통하여 전송되는 정보의 위조, 무단절취, 파괴의 위험성을 내포하고 있으므로 이러한 위험성을 해결하지 못한다면 개인은 물론 사회 전반에 커다란 영향을 미치게 된다. 따라서 이러한 정보 보안상의 문제 및 범죄 행위를 사전에 예방하기 위해서 출현한 것이 암호화(cryptography)이다[1]. 암호화는 인가된 사람만이 허가된 정보를 이해할 수 있도록 정보를 변형하는 제반 기법으로 암호화 기법은 오래전 부터 연구되어 왔으며, 최근 이에 대응하여 암호화를 파괴하는 암호 분석학(crypto analysis) 또한 컴퓨터 성능의 발달로 활발히 연구되고 있다. 따라서 암호화 기법과 암호 분석학은 상호 보완적인 입장에서 더욱더 발전할 것이다.

최근에 통신망을 통하여 제공되는 고도 통신망 서비스가 활발히 연구, 개발되고 있다. 특히, 메세

* 정회원, 숭실대학교 전산과
** 정회원, 서울산업대학교 전산과
*** 종신회원, 숭실대학교 전산과
**** 종신회원, 숭실대학교 전산과

지 처리 시스템(MHS : Message Handling System) 서비스, 전자 교환(EDI : Electronic Data Interchange) 서비스 등은 앞으로 기업체는 물론 일반 사용자에게까지 널리 이용될 것으로 생각된다. 이러한 서비스의 특징은 통신망 상에서 메세지를 이용하여 다양한 형태의 서비스를 제공하기 때문에 메세지 자체에 대한 인증이나 송신자, 수신자 상호 쌍방간의 인증에 관한 문제가 중요하게 대두되고 있다. 인증(authentication)이란 전송되는 메세지와 송·수신자의 정당성을 입증하는 것이다[14]. 인증의 방법에는 여러가지가 있으나, 그 중에서 디지탈 서명은 컴퓨터로 생성한 메세지를 인증하는 가장 이상적인 메카니즘으로 알려져 있다.

디지탈 서명이란 송신자가 메세지의 송신을 부정할 수 없고 정당한 수신자 조차도 메세지를 위조할 수 없도록 하는 것으로 이러한 디지탈 서명을 이용하여 구축한 인증 시스템은 매우 안전한 시스템이 될 것이다[14]. 디지탈 서명의 방법에는 두가지 형태가 있다. 첫째는 직접서명 방법으로 공개키 암호화 방식을 이용한 것이다. 이것은 공개키와 비밀키만을 이용하여 디지탈 서명을 하는 방식이다. 둘째는 간접서명 방법으로 관용키 암호화 방식을 이용한 방법이다. 이것은 디지탈 서명을 하기 위하여 신뢰할 수 있는 제3의 중재자를 필요로 한다. 간접 서명방식은 중재자를 필요로 하기 때문에 서명절차가 복잡하고 많은 정보량을 보관해야 하는 단점이 있다. 이에 반하여 직접서명 방식은 서명 절차가 매우 간단하다. 직접 서명 방식에는 RSA(Rivest, Shamir, Adleman) 서명방식과 knapsack 문제를 이용한 서명방식 등이 있으나 최근에 knapsack 문제를 이용한 방법에 대하여 안전성 여부에 몇가지 문제점이 제기되었다. 지금까지 발표된 디지탈 서명방식 중에서 가장 신뢰할 수 있는 것은 RSA 서명방식이라고 할 수 있다. 그러나 RSA 서명방식은 서명을 생성하고 검증을 하는데 많은 계산량이 요구되기 때문에 서명을 생성 및 검증하는 처리 속도가 매우 늦다는 문제점을 가지고 있다. 최근 이러한 문제점을 해결하기 위한 연구가 활발히 진행되고 있으며, 특히 Ong, Schnorr, Shamir등은 modulo n 상에서 등식 $x^2 + ky^2 \equiv m \pmod{n}$으로부터 x,y를 구하는 문제의 어려움에 기초한 디지탈 서명방식[8]을 제안 하였으나, J.M. Pollard에 의하여 n을 인수분해하지 않고 등식 $x^2 + ky^2 \equiv m \pmod{n}$의 해를 구하는 다항식 시간 알고리즘이 개발됨으로써 판독되었다.

본 논문은 RSA 서명방식의 처리 속도에 대한 문제점을 해결하기 위하여 다음과 같은 특성을 지닌 새로운 고속 디지탈 서명방식을 제안하였다.

- 비도(security level)는 큰 수에 대한 인수분해의 어려움에 둔다.
- 저차 합동 다항식에 기초한다.
- 서명을 생성하는데 난수를 이용한다.
- 서명에 대한 검증은 부등식을 이용한다.
- 처리 속도를 빠르게 한다.

위의 특성을 지닌 새로운 고속 디지탈 서명방식은 궁극적으로 디지탈 서명의 처리 속도 및 안전성을 향상시키는데 있다. 즉, 저차 합동 다항식을 사용함으로써 처리 속도의 개선을 가져올 수 있으며, 서명을 생성할때 난수를 이용함으로써 비도(security level)를 높일 수 있다. 또한 검증을 위해 부등식을 사용함으로써 서명의 안전성 및 처리 속도를 향상 시킬 수 있다.

본 논문의 구성은 제2장에서 디지탈 서명의 개념과 원리에 관하여 상세히 언급하였으며, 제3장에서는 디지탈 서명에 대한 기존의 접근 방법들 중에서 대표적인 방식을 소개하였다. 제4장에서는 새로운 고속 디지탈 서명 방식과 처리 속도 향상을 위한 방법을 제시하고, 이를 검증하였다. 제5장에서는 RSA 서명방식과 새로운 고속 디지탈 서명방식의 처리 속도를 비교, 분석, 평가하였으며, 제6장에서는 결론을 내렸다.

제 2 장 디지탈 서명의 개요

디지탈 서명은 컴퓨터로 생성한 메세지들을 인증하는데 사용되는 방법 중에서 가장 이상적인 메카니즘이다. 이러한 디지탈 서명의 목적은 수신자가 받은 메세지를 위조할 수 없도록 하고 송신자가 자신이 전송한 메세지의 내용을 부인할 수 없도록 하는 것이다.

디지탈 서명을 위한 요구사항은 다음과 같다[9]. ((그림 2-1) 참조)

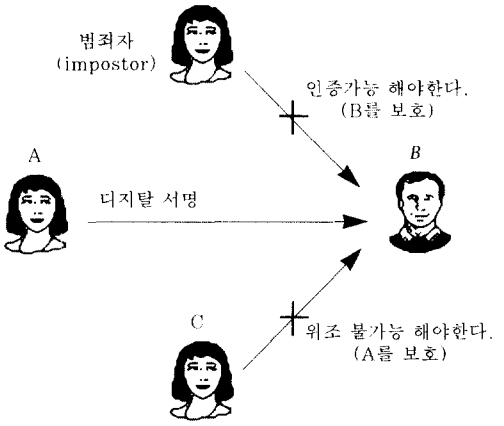


그림 2-1 디지탈 서명의 요구사항

① 위조 불가능해야 한다.(unforgeable) : *A*가 생성한 디지탈 서명을 임의의 *C*가 생성할 수 없어야 한다.

② 인증 가능해야 한다.(authentic) : *B*가 정당한 송신자 *A*로부터 디지탈 서명을 수신했다면, *B*는 디지탈 서명이 정말 *A*로부터 송신된 것인지 확인할 수 있어야 하며, 오직 *A*만이 이 디지탈 서명을 생성할 수 있어야 한다.

③ 변경 불가능해야 한다.(not alterable) : 메세지 *M*이 전송된 후에 *A*나, *B*나 또는 다른 사용자에 의하여 메세지의 내용이 변경되어서는 안된다.

④ 재사용이 불가능해야 한다.(not reusable) : 수신자 *B*가 전송된 디지탈 서명 및 메세지 *M*을 다시 사용할 수 없어야 한다.

디지탈 서명의 가장 큰 장점은 메세지의 송신자와 수신자간의 분쟁을 해결할 수 있다는 것이다. 디지탈 서명의 방법에는 디지탈 서명을 하기 위하여 제3의 중재자가 요구되는지에 따라 직접서명과 간접서명으로 나눌 수 있다. 즉, 직접서명은 디지탈 서명을 위하여 중재자가 필요없는 방식이고, 간접서명은 중재자가 필요한 방식이다.

2.1 직접서명

직접서명 방식은 비대칭적 암호화(asymmetric cipher)구조를 가지는 시스템에서 쉽게 구현할 수 있는 방법으로 디지탈 서명을 위하여 중재자가 필요없이 공개키와 비밀키만을 이용하여 실현할 수 있으며, 그 대표적인 예는 공개키 암호화 방식을 이용한 RSA 서명방식이다. (그림 2-2)는 직접서명방식의 개념[2]을 나타냈다. *F*, *G* 함수는 난수 *ks*로 부터 공개키 *ke*와 비밀키 *kd*를 생성한다. *D*는 복호화 함수, *E*는 암호화 함수로 정의한다. *D*와 *E*는 서로 동일한 함수이지만 다른 키를 사용하기 때문에 결과는 서로 다르다.

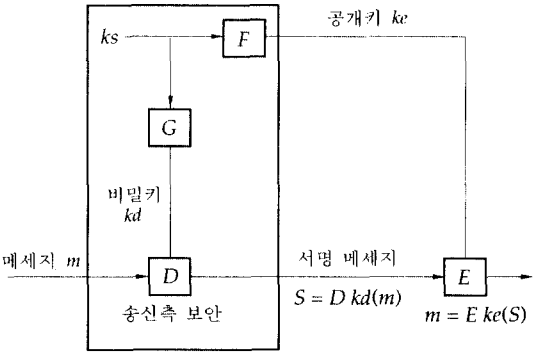


그림 2-2 직접서명 방식의 개념

공개키 암호화 시스템에서는 공개키와 비밀키가 역의 관계를 가지기 때문에 암호화와 복호화의 순서에 관계없이 결과는 동일하다. 즉,

$$D\ kd(E\ \mathrm{mod}(m)) = E\ ke(D\ kd(m))$$

이 성립한다.

송신측은 먼저 디지탈 서명을 하기 위하여 메세지 m을 자기자신이 보유하고 있는 비밀키 kd를 가지고 서명 메세지 $S = D\ kd(m)$을 생성한다. 이 서명 메세지는 다시 공개키 ke를 가지고 $m = E\ kc(S)$인 평문을 생성할 수 있다. 따라서 공개키 암호화 시스템을 이용한 디지탈 서명은 비밀키 kd에 의하여 메세지 m 및 송신자에 대한 보안(secrecy)과 인증이 유지된다. 즉, 서명 S는 송신자외에는 어느 누구도 생성할 수 없게 된다. 또한 공개키에 의하여 평문이 생성되기 때문에 수신자에 대한 보안(secrecy)과 인증이 보장된다. 이러한 방법은 송신자와 수신자간의 분쟁이 발생해도 명확하게 해결할 수 있으며 서명 절차가 매우 간단하다.

2.2 간접서명

간접서명은 대칭적 암호화(symmetric cipher) 구조를 가지는 시스템에서 사용되는 방법으로 암호화 키와 복호화 키가 같은 관용키 암호화 시스템에서 사용할 수 있다. 암호화 시스템에서는 송신자나 수신자가 그들의 고유키인 비밀키를 가지고 있기 때문에 키에 대한 안전도는 자명하다. 따라서 메세지와 송신자 그리고 수신자의 인증은 가능하다. 그러나 디지탈 서명의 조건인 "위조가 불가능해야 한다."를 만족하지 못한다. 즉, 송·수신측에서 같은 키를 공유하기 때문에 서로 다른 서명을 만든다해도 어떤것이 옳은 서명인지 판별하기가 어렵다. 이러한 문제점을 해결하기 위하여 어떤 신뢰할 수 있는 제3의 중재자를 둠으로써 송·수신자간에 서명에 대한 분쟁이 발생할 경우 명확한 판결을 할 수 있다.

(그림 2-3)은 간접서명 방식의 개념을 그림으로 나타냈다. 간접서명의 프로토콜을 설명하면 다음과 같다[9].

- 가정

i) S : 송신측, R : 수신측, A : 중재자, m : 메세지

ii) S와 A는 비밀키 K_S를 갖는다.

iii) R과 A는 비밀키 K_R을 갖는다.

iv) S가 디지탈 서명을 한 메세지를 R에게 전송하려고 한다.

- 절차

① S는 A에게 메세지 m을 비밀키 K_S로 복호화하여 전송한다.

$$m_1 = E(m,K_S)$$

② A는 송신된 메세지 m_1이 A로부터 전송되었는지 확인한다(비밀키 K_S로 복호화).

$$m = D(m_1,K_S) = D(E(m,K_S),K_S)$$

③ 메세지 m_1의 정당성이 입증되면 A는 R에게 메세지 m과 S가 송신한것임을 증명하는 S를 비밀키 K_S로 암호화하여 전송한다.

$$m_2 = E((m,S,E(m,K_S)),K_R)$$

④ R은 A가 S로부터 메세지를 수신했다고 입증한 A의 메세지 m과 S 그리고 R이 복호화할 수 없는 $E(m,K_S)$를 수신한다. R은 메세지 m과 $E(m,K_S)$ 내용을 한 부 복사하여 보관한다.

$$m = D(m_2,K_R) = D(E((m,S,E(m,K_S)),K_R),K_R)$$

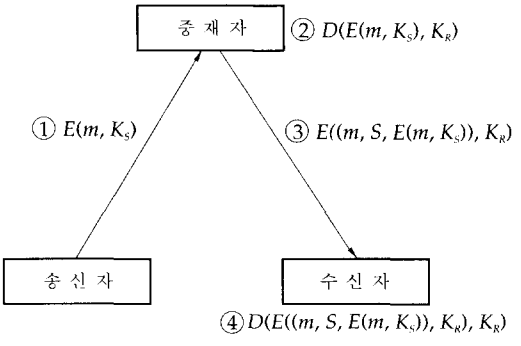


그림 2-3 간접서명 방식의 개념

* 메세지 m과 $E(m, ks)$보관

3. 디지탈 서명에 관한 기존의 접근방법

본 장에서는 지금까지 연구되어 온 디지탈 서명

방식들 중에서 대표적이라고 할 수 있는 몇가지 방식들에 관하여 언급하였다.

3.1 Diffie-Hellman(DH) 방식

1976년 Diffie와 Hellman은 기존의 관용키 암호화 방식의 문제점 즉, 키 관리의 어려움을 해결하기 위한 새로운 개념의 공개키 암호화 방식을 발표하였다[3]. 이 논문에서는 단방향 함수(one way function)에 기초한 디지탈 서명의 개념에 관하여 언급하였으며, (그림 3-1)에 나타난 것처럼 공개키 암호화 시스템을 이용한 디지탈 서명방식을 제안하였다.

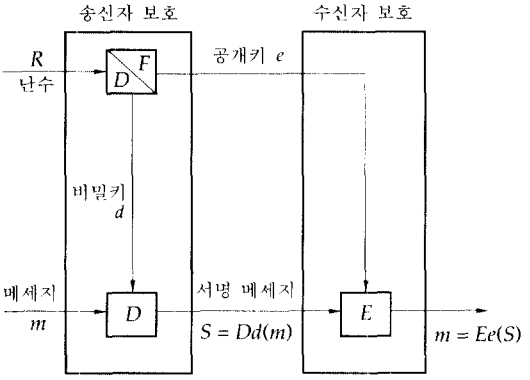


그림 3-1 DH의 디지탈 서명방식

메세지의 송신자는 시스템내에서 두개의 암호화 키 즉, 공개키 e와 비밀키 d를 생성한다. 이때 암호화 키를 생성하기 위하여 임의의 난수 R을 발생시킨다. 함수 F와 G는 이 난수 R을 입력으로 하여 e와 d를 생성한다. 메세지에 대한 서명을 하기 위하여 송신자만이 보유하고 있는 비밀키 d와 메세지 m을 입력으로 하여 복호화 함수 D를 거치면 서명 메세지를 생성할 수 있다. 이 서명된 메세지 $S = Dd(m)$는 수신자에게 전송된다. 수신자는 S와 공개키 e를 입력하여 암호화 함수 E를 수행하면 원래의 메세지 m을 복원할 수 있다. 이 논문에서는 암호화 시스템의 개념을 언급 하면서 이를 이용한 디지탈 서명방식의 개념을 제안하였으나, 실제로 응용할 수 있는 구체적인 디지탈 서명 알고리즘에 관해서는 기술하지 않았다.

3.2 Rivest, Shamir, Adleman(RSA) 방식

RSA 서명방식은 R.L. Rivest와 A. Shamir 그리고 L. Adleman에 의해 제안된 방식[10]으로 큰 수의 인수분해 어려움을 비도로 하고 있다. 비록 인수분해 문제가 NP-complete 문제로 증명되지는 않았으나, 다항식 시간 알고리즘이 알려져 있지 않은 NP문제이다.

사용자 i가 사용자 j에게 디지탈 서명을 전송할 경우 서명절차를 살펴보면 다음과 같다.

키의 생성

각 사용자는 자신의 공개키 (e,n)과 비밀키 (d, n)을 생성한다.

① 2개의 큰 소수 p와 q를 생성하여 $n = pq$를 계산한다.

② Euler의 ϕ함수 $\phi(n) = (p-1)(q-1)$과 서로 소가 되는 e를 계산한다.

$$\gcd(e, \phi(n)) = 1$$

③ (n)과 공개키 e로부터 유클리드 알고리즘에 의하여 비밀키 d를 계산한다.

$$ed = 1 \pmod{\phi(n)}$$

메세지 m에 대한 서명

사용자 i는 서명을 하고자 하는 메세지 m을 자신만이 보유하고 있는 비밀키 $d(i)$로 복호화한다.

$$S = D(m) = m^{d(i)} \pmod{n}$$

서명 메세지 전송

사용자 i는 사용자 j의 공개키인 $e(j)$를 암호화하여 전송한다.

$$C = E(S) = m^{d(i)e(j)} \pmod{n}$$

서명에 대한 인증

사용자 j는 자신이 보유하고 있는 비밀키 $d(j)$를 복호화한다.

$$S = D(c) = m^{d(i)e(j)d(j)} \pmod{n} = m^{d(i)} \pmod{n}$$

다시 사용자 i의 공개키 $e(i)$를 암호화하면 검증이 된다.

$$m = E(S) = m^{d(i)e(i)} \pmod{n}$$

위 방식은 비도 측면에서 매우 안정되어 있다고 평가되고 있지만 몇가지 단점을 지니고 있다. 첫째는 암호화하는데 많은 시간이 소요된다는 것이다. 즉 키값의 지수승을 계산하기 때문에 처리 속도가 늦다. 둘째는 메세지를 reblocking 해야 하는 문제가 발생한다. 송신측 i의 ni가 수신측 j의 nj 보다 클 경우 암호화 메세지 C는 $(0 < C < nj)$의 범위를 갖기 때문에 암호화 메세지를 나누어 (reblocking 송신하고 수신측에서는 이를 다시 하나로 묶어야 한다. 이러한 reblocking의 문제는 임계값을 적절히 설정하면 쉽게 해결할 수 있다.

3.3 Merkle-Hellman(MH) 방식

R. Merkle와 M. Hellman은 1978년 NP-complete 문제로 잘 알려져 있는 knapsack 문제를 이용한 공개키 암호화 시스템을 발표하였다[5]. knapsack 문제란 벡터 $A = (a_1, a_2, \cdots, a_n)$($a_i$는 양의정수)와 양의 정수 C가 등식

$$C = \sum_{i=1}^{n} a_i x_i$$

를 만족할때 이 등식의 해 $X = (x_1, x_2, \cdots, x_n)$을 구하는 문제이다. 이러한 knapsack 문제를 이용하여 다음과 같이 암호화 시스템을 구성할 수 있다.

전송하려는 데이타를 $X = (x_1, x_2, \cdots, x_n)$이라 할때 송신자는 수신측의 공개키 $A = (a_1, a_2, \cdots, a_n)$($a_i$는 양의정수)를 찾아 평문 X를

$$C = AX = \sum_{i=1}^{n} a_i x_i$$

와 같이 암호화할 수 있다. 그러나 암호문 C에서 평문 X를 유도하는 과정은 knapsack 문제의 해를 구하는 과정이므로 정당한 수신자 조차도 해를 구할 수 없게되어 이런 방법으로는 암호화 시스템을 구성할 수 없다. Merkle-Hellman에 의하여 제안된 암호화 방식은 모든 knapsack 문제가 해를 구하기 위한 시간이 지수시간을 요하지 않는 사실에 근거한 것이다. 즉, trapdoor 단방향 함수를 구성함으로써 암호화 시스템을 구성하였으며, 이러한 trapdoor knapsack 암호화 시스템을 이용한 디지탈 서명방식은 다음과 같다.

전송하려는 메세지를 m이라할 때 송신자는 수신측의 $A = (a_1, a_2, \cdots, a_n)$을 찾아 $A(i) * X = m$이 되는 X를 전송한다. 수신측에서 X를 수신하면 $A = (a_1, a_2, \cdots, a_n)$와 X를 이용하여 메세지 m을 복원하면 된다. 이 논문에서는 위에서 설명한 바와 같이 trapdoor knapsack 단방향 함수를 이용한 디지탈 서명에 관하여 설명을 하였으나, 실제 구현에 관한 사항은 언급하지 않았다. 이 암호화 시스템은 1984년 A. Shamir에 의해서 다항식 시간 알고리즘[13]으로 해결할 수 있다는 것이 증명되어 판독되었다. 현재까지 발표된 knapsack 문제를 이용한 공개키 암호화 시스템은 대부분 안전하지 못함이 입증되고 있다.

3.4 Shamir 방식

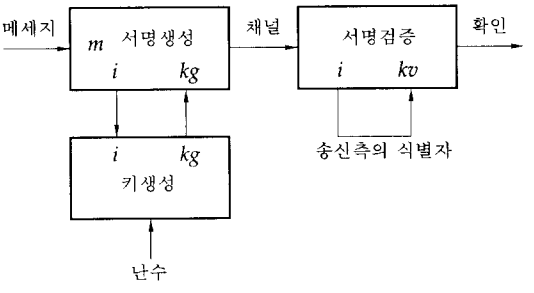


그림 3-2 Shamir 방식

1984년 A. Shamir는 사용자를 유일하게 구분할 수 있는 키로써 이름이나 망주소 또는 개인 식

별번호, 전화번호 등을 이용한 디지탈 서명방식[12]을 발표하였다. 이 방식은 비밀키나 공개키의 교환이 필요없으며, 공개키의 관리를 위한 디렉토리 혹은 공개 화일 목록없이 또한 신뢰할 수 있는 제3의 중재자를 이용하지 않고 디지탈 서명이 가능한 방식이다. (그림 3-2)는 Shamir 방식의 개념을 나타내었다.

여기서 메세지 m은 신뢰할 수 있는 키 생성 센타로부터 생성된 키 kg로 서명된다. 이 서명 메세지와 송신측의 식별자 i를 전송하면, 수신측에서는 이 정보를 가지고 검증을 위한 키 kv가 되는지를 확인한다. 서명절차를 자세히 살펴보면 다음과 같다.

키 생성

신뢰할 수 있는 키 생성 센타로 부터 다음 값을 결정한다.

① $n = pq$ 를 생성한다.(p,q : 큰자리 소수)

② i : 사용자 식별자

③ e : $\phi(n)$과 서로소인 큰자리 소수

$$\gcd(e, \phi(n)) = 1$$

④ $g^e \equiv i \pmod n$

모든 사용자는 같은 n과 e를 가지며, i는 각자 다르다. 비밀키는 g이다.

메세지 m에 대한 서명

① 난수 r을 선택한다.

② $t \equiv re \pmod n$인 t를 계산한다.

③ $S \equiv gr^{f(t,m)} \pmod n$인 S를 계산한다.(f : 단방향 함수)

서명 메세지의 전송

송신자는 수신자에게 다음과 같은 정보를 전송한다.

$$(i, m, s, t)$$

서명에 대한 인증

전송된 정보(i, m, s, t)와 n, e를 이용하여 다음 식으로 검증을 한다.

$$S^e \equiv it^{f(t,m)} \pmod n$$

3.5 Ong, Schnorr, Shamir(OSS) 방식

1984년 H. Ong과 C. Schnorr 그리고 A. Shamir는 modulo n 상에서 등식 $x^2 + ky^2 \equiv m \pmod n$으로부터 x,y를 구하는 문제의 어려움을 이용한 디지탈 서명방식을 제안하였다[8]. 이는 RSA 서명방식의 단점인 서명 생성과 검증시 처리속도의 문제를 해결하기 위한 방법으로 제안되었으나, J.M. Pollard에 의하여 n을 인수분해하지 않고 등식 $x^2 + ky^2 \equiv m \pmod n$의 해를 구하는 다항식 시간 알고리즘이 개발되어 판독되었다. 서명방식은 다음과 같다.

서명 구성

① p,q 는 큰 자리 소수, $n = pq$

② $k \equiv -u^{-2} \pmod n$인 k와 u계산

③ 공개키 : k,n , 비밀키 : u

메세지 m에 대한 서명

① $x_1x_2 \equiv m \pmod n$이 되는 임의의 x_1, x_2를 생성한다.

② $x_1 \equiv S_1 + u^{-1}S_2 \pmod n$과 $x_2 \equiv S_1 + u^{-1}S_2 \pmod n$을 만족하는 S_1과 S_2를 계산한다.

$$x_1x_2 \equiv S_1^2 - u^{-2}S_2^2 = S_1^2 + kS_2^2 \equiv m \pmod n$$

서명에 대한 검증

공개키 k, n 을 이용하여 m, S_1, S_2가 $S_1^2 + kS_2^2 \equiv m \pmod n$이 되는지 확인한다.

4. 새로운 고속 디지탈 서명방식의 제안

본 장에서는 먼저 새로 제안한 고속 디지탈 서명방식 중에서 기본적인 서명방식의 절차를 기술하고, 서명방식에 대한 관계식을 유도하였다. 다음에 서명을 하는데 소요되는 계산량을 줄이기 위하여 처리속도 향상을 위한 방법을 제시하였으며, 끝으로 새로 제안한 서명방식이 디지탈 서명으로써 타당한지 검증하였다.

4.1 기본적인 서명방식

새로 제안한 고속 디지탈 서명방식은 신뢰할 수 있는 통신망 상에서의 통신에 기초를 하고 있으며, 공개키 암호화 시스템을 이용하였다. 서명방식의 비도는 큰 수의 인수분해 어려움에 두고 있으며, 서명의 처리 속도를 높이기 위하여 저차 합동 다항식을 이용하였고, 안전성을 높이기 위하여 서명의 구성시 난수를 이용하였다. 또한 서명의 검증에 있어서는 부등식을 이용하였는데 이는 등식을 사용 할 경우 결정적인 정보(deterministic information)에 의하여 키 값이 정해지기 때문에 역으로 암호 분석가들이 이를 이용하면 쉽게 해독이 가능한 것을 방지하기 위함이다.

(1 단계) : 송신자의 암호화 키 생성

송신자는 다음과 같은 암호화를 위한 비밀키와 공개키를 생성한다. 여기서 p, q는 비밀키이고 f, k, w, q는 공개키이다.

① 충분히 큰 소수 p와 q를 선택한다. ($p > q$)

② 다음 조건을 만족하는 정수 k와 w를 결정한다.

$$k = p^2q \quad (1)$$

$$w = k^{2/3} \quad (2)$$

여기서 $[i]$은 실수 i와 같거나 보다 큰 최소의 정수이다.

③ 정수 f와 g를 결정한다.

$$f = k - w - 1 \quad (3)$$

$$g = [w/p] \quad (4)$$

위에서 생성된 비밀키와 공개키는 공개키 암호화 시스템에 등록된 정보가 변경되는 경우를 제외하고는 공개키 암호화 시스템의 구축시 처음 한 번만 수행한다.

(2 단계) : 메세지 M에 대한 디지탈 서명

메세지 M은 이진수로 표현된 정수이며 크기는 다음과 같다.

$$g \le M \le k - g$$

만약 $M < g$ 이고 $M > k - g$ 이면, 메세지 M에 추가 정보를 덧붙이거나, 혹은 M을 분할해야 한다.

메세지 M에 대한 서명 S는 다음 절차에 따라 구성된다.

① 아래의 조건을 만족하는 정수인 난수 x_1을 결정한다.

$$1 \le x_1 \le pq - 1 \quad (5)$$

$$\gcd(x, k) = 1 \quad (6)$$

gcd(greatest commom divisor)는 x_1과 k의 최대공약수를 나타낸다.

② 아래의 조건을 만족하는 정수 x_2를 결정한다.

$$0 \le h \le k - 1 \quad (7)$$

$$h \equiv x_1^2 + M^2 \pmod{k} \quad (8)$$

$$x_2 = [(f - h)/pq] \quad (9)$$

③ 아래의 조건을 만족하는 정수 x_3를 결정한다.

$$0 \le x_3 \le k - 1 \quad (10)$$

$$2x_1x_3 \equiv x_2 \pmod{k} \quad (11)$$

④ 아래의 조건을 만족하는 정수 S를 결정한다.

$$1 \le S \le k - 1 \quad (12)$$

$$S \equiv x_1 + x_3pq \pmod{k} \quad (13)$$

(3 단계) : 서명 메세지의 전송

송신자는 수신자에게 메세지 M과 서명 정보 S를 전송한다.

$$(M, S)$$

(4 단계) : 디지탈 서명에 대한 인증

수신자는 공개키 (f, k, w, g)와 아래의 조건식을 기초하여, 수신된 메세지의 송신자가 공개키 암호화 시스템에 등록된 사람인지 여부를 판단한다.

$$f \le M^2 + S^2 \le f + w \pmod{k} \quad (14)$$

$$g \le S \le k - 1 \quad (15)$$

$$g \le M \le k - g \quad (16)$$

위 조건식을 모두 만족하면 송신자와 메세지는 인증이 된 것이다. 위에서 제시한 기본적인 서명방식을 간단한 예를 들어 설명하면 다음과 같다.

(1 단계) 송신자의 암호화 키 생성

i) $p = 7$, $q = 5$ ($p > q$)를 선택

ii) $k = p^2q = 245$

iii) $w = [k^{2/3}] = 40$

iv) $f = k - w - 1 = 201$

v) $g = [w/p] = 6$

(2 단계) 메세지 M에 대한 디지탈 서명

$-g \le M \le k-g$ 를 만족하는 메세지 $m = 10$을 선택하면,

① $1 \le x_1 \le pq-1$

$\gcd(x_1, k) = 1$ 를 만족하는 $x_1 = 3$을 선택

② $0 \le h \le k-1$

$h \equiv x_1^2 + M^2 \pmod k = 109$

$x_2 = [(f-h)/pq] = 3$

③ $0 \le x_3 \le k-1$

$2x_1x_3 \equiv x_2 \pmod k$, $x_3 = 123$

④ $1 \le S \le k-1$

$S \equiv x_1 + x_3pq \pmod k = 143$

(3 단계) 서명 메세지의 전송

- $(M, S) = (10, 143)$

(4 단계) 디지탈 서명에 대한 인증

공개키 (f, g, w, k)와 전송된 서명 메세지 (M, S)를 이용하여 다음 조건식이 성립하는지를 확인하여 검증 한다.

$f \le M + S \le f + w \pmod k$, $204 \le 214 \le 244 \pmod{245}$

$g \le S \le k-1$, $6 \le 143 \le 244$

$g \le M \le k-g$, $6 \le 10 \le 240$

전송된 메세지 (10, 143)은 위의 조건식을 모두 만족함으로 정당한 메세지이다.

4.2 관계식의 유도

식 (1) 즉, $k = p^2q$의 조건하에서, 다음의 관계를 만족하는 임의의 정수 S

$$1 \le S \le k-1$$
$$(S, k) = 1$$

를 다음과 같이 표현할때

$$S \equiv x_1 + x_3pq \pmod k \quad (16)$$

여기서,

$$1 \le x_1 \le pq-1 \quad (17)$$
$$(x_1, k) = 1 \quad (18)$$
$$0 \le x_3 \le k-1 \quad (19)$$

이다. x_1과 x_3는 S로부터 유일하게 결정된다.

x_1이 주어지면 식 (14)가 적용되기 위한 조건 즉, x_3에 대한 조건은 다음과 같이 유도된다.

$x_2 \equiv 2x_1x_3 \pmod k$로 놓으면 (20)

$M^2 + S^2 \equiv M^2 + x_1^2 + 2x_1x_3pq + (x_3pq)^2 \pmod k$

$\equiv M^2 + x_1^2 + 2x_1x_3pq \pmod k$ (21)

$\equiv h + x_2pq \pmod k$ (22)

이다. 이것은 $p^2q \equiv 0 \pmod k$과 식 (8)로부터 유추할 수 있다.

식 (22)가 식 (14)를 만족시키기 위해 x_2는 다음 조건을 만족해야 한다.

$$f \le h + x_2pq \le f + w \quad (23)$$

p, q가 큰 소수이고, $p > q$ 이므로,

$$pq > w \quad (24)$$

따라서, 다음과 같이 나타낼 수 있다.

$$f \le h + x_2pq \le f + pq \quad (25)$$
$$(x_2 - 1)pq < f - h \quad x_2pq \quad (26)$$

식 (26)으로부터 x_2의 값이 정해진다.

$$x_2 = [(f-h)/pq] \quad (27)$$

식 (20)과 식 (27)로부터, x_2는 다음의 관계를 만족한다.

$$2x_1x_3 \equiv x_2 \pmod k$$
$$\equiv [(f-h)/pq]$$

x_1과 k는 서로소이므로, 식 (28)을 만족하는 x_2에 대해 x_3가 항상 존재한다. 따라서 식 (14)를 만족하는 S는 k와 서로소인 임의의 $x_1(1 \le x_1 \le pq-1)$으로부터, 그리고 식 (28)을 만족하는 x_2로부터 x_3를 결정함으로써 정해진다.

4.3 처리 속도 향상을 위한 방법

서명의 생성(2 단계)에 대한 처리 속도를 높이기 위해서는 식 (11)의 계산을 빠르게 수행해야 한다. 식 (11)에서와 같이 곱셈과 나눗셈이 다중 정밀도 연산(multiple precision operation)으로 이루어 진다면 계산의 복잡도(computational complexity)는 RSA 서명방식의 복잡도와 거의

같을 것이다.

식 (11)의 처리 속도를 개선하기 위하여 다음과 같은 선행 처리(preprocessing)을 이용하였다.

i) (2 단계)를 수행하기 전에 식 (5)와 식 (6)의 조건을 만족하는 난수 x_1를 결정한 후에 아래 조건을 만족하는 정수 $\theta(1 \leq \theta \leq k-1)$를 유클리드 제법(Euclid's division)으로 결정한다.

$$2x_1\theta \equiv 1 \pmod{k} \qquad (29)$$

식 (6) 즉, $\gcd(x_1, k) = 1$이 만족되지 않을 확률은 매우 작기 때문에, 만약 x_1이 이 조건을 무시한 채 결정되더라도 실제로는 문제가 발생하지 않는다.

ii) 위에서 계산된 θ를 이용하여 x_3를 다음과 같이 결정할 수 있다. 식 (28)에서 $2x_1x_3 \equiv x_2 \pmod{k}$ 이므로 식 (29)의 θ을 대입하면,

$$x_3 \equiv \theta x_2 \pmod{k} \qquad (30)$$

로 x_3를 계산할 수 있다.

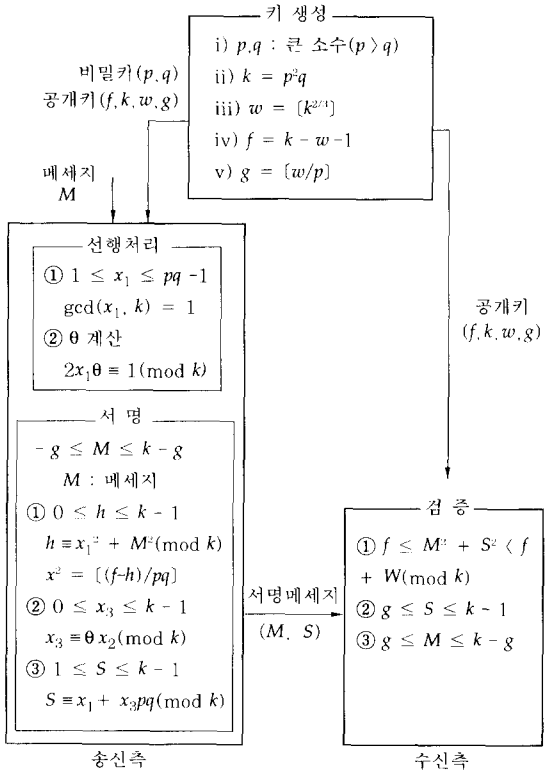


그림 4-1 고속 디지탈 서명방식의 흐름

이러한 선행처리는 서명의 생성(2 단계)과는 별도로 수행할 수 있으며, x_3의 계산량을 감소시킬 수 있기 때문에 결과적으로 처리 속도를 4.1절에서 제시한 기본적인 서명방식의 처리 속도보다 개선할 수 있다. (그림 4-1)에 기본적인 서명방식에 선행처리 방법을 추가한 고속 디지탈 서명방식의 전체적인 흐름을 나타냈다.

4.4 디지탈 서명에 대한 검증

새로 제안한 고속 디지탈 서명방식의 안전성, 즉 비도는 $k = p^2q$를 인수분해하여 소수 p, q를 구하는 것이 어렵다는 사실에 근거하고 있다. 이것은 다시 말해서 공개키(f, k, w, g)와 전송정보(M, S)로 부터 p, q를 찾아내기가 거의 불가능함을 나타낸다. 디지탈 서명방식에 대한 안전성을 검증하기 위해서는 다음 조건을 만족하면 충분하다.

i) 공개키 및 전송정보로 부터 비밀키를 얻을 수 없어야 한다.

ii) 공개키 및 전송정보로 부터 검증식 (14)와 (15)를 만족하는 위조된 전송정보 (M', S')를 얻을 수 없어야 한다.

다음의 (1)과 (2)는 위의 서명조건 *i)*를 만족시키고 (3)은 서명조건 *ii)*를 만족시킨다.

(1) 공개키 (f, k, w, g)나 전송정보 (M, S)로 부터 비밀키 (p, q)를 얻을 수 없다[(조건 *i*)]. 이것은 다음 관계로 증명된다.

- 만약 p, q가 충분히 큰 소수로 결정되면 k를 인수분해하여 p, q를 찾아내기가 어렵다. 따라서 f, g로 부터 p, q를 유추할 수 없다.
- w로 부터 p, q를 결정하는 것은 식 (2)에 보인 것처럼 w가 k로 부터 p, q를 결정하는 것을 변형시킨 것에 불과하다. 따라서 큰 수에 대한 인수분해의 어려움과 같다.
- 식 (13)에서 보인 것처럼, S의 보안이 임의의 난수 x_1에 의존하기 때문에 S로 부터 p, q를 결정할 수 없다.

위에서 언급한 것처럼 공개키나 전송정보로 부터 비밀키를 얻는 유일한 방법은 k를 인수분해하여 큰 소수 p, q를 찾아내는 것이다.

(2) p, q는 다음과 같은 연립 방정식으로도 결정할 수 없다.

$$k \equiv p^e q$$
$$S_1 \equiv x_1 + z_1 pq \pmod{k}$$
$$\vdots$$
$$S_j \equiv x_j + z_j pq \pmod{k}$$
$$b_1 \equiv M_1^2 + x_1^2 \pmod{k}$$
$$\vdots$$
$$b_j \equiv M_j^2 + x_j^2 \pmod{k}$$
$$2x_1 z_1 \equiv (a - b_1)/pq \pmod{k}$$
$$\vdots$$
$$2x_j z_j \equiv (a - b_j)/pq \pmod{k}$$

(3) 공개키나 전송정보로 부터 (M, S)를 위조한 (M', S')를 서명 검증식에 만족하도록 구성할 수 없다[〈조건 ii〉]. S를 위조한 S'이 되기 위해서는 난수 x_1과 소수 p, q를 알아야만 한다. 따라서 S을 생성하기가 거의 불가능하다.

5. RSA 서명방식과 새로운 고속 디지탈 서명방식의 비교

본 장에서는 새로 제안한 고속 디지탈 서명방식의 성능을 평가하기 위하여 기존의 RSA 서명방식과 비교, 분석을 하였다. 먼저 이를 위한 실험 환경을 설명하였고, 처리 속도를 비교하기 위하여 디지탈 서명의 생성과 검증을 처리하는데 소요되는 시간을 각각 측정 하였다. 또한 이 결과를 토대로 고속 디지탈 서명방식의 성능을 분석, 평하였다.

5.1 실험 환경

RSA 서명방식과 새로 제안한 고속 디지탈 서명방식의 처리 속도를 비교하기 위한 실험 환경은 다음과 같다.

- 사용기종 : SUN 4/330
  - processor : SPARC processor(32bits)
  - clock speed : 25 MHz
  - OS : UNIX(Sun OS 4.0.3)
  - 데이타 수형의 범위 :
    integer : 2 bytes
    long integer : 4 bytes
    float : 8 bytes
    double : 8 bytes
- 사용언어 : C

처리 시간을 측정하기 위하여 UNIX에서 제공하는 시스템 호출(system call) 중에서 "times"을 이용하였다. 이 "times" 는 SUN 4/330 시스템의 운영체제인 Sun OS 4.0.3에서 기본적으로 10^{-1}초까지 사용자에게 제공하기 때문에 그 이하의 시간을 측정하기가 불가능하다. 따라서 그 이하의 시간을 측정하기 위하여 프로그래밍 상에서 10^{-6}초까지 측정할 수 있도록 작성하였으며, 또한 상세한 시간을 측정하기 위하여 각각의 프로그램을 i번 반복하는데 소요되는 시간을 측정하였다.

5.2 처리 속도 비교

기존의 RSA 서명방식과 새로 제안한 고속 디지탈 서명방식의 처리 속도를 비교하기 위하여 〈표 5-1〉에 나타낸 것처럼 서명 절차를 3단계 즉, 서명키 생성 단계, 서명 생성 단계, 서명 검증 단계로 분리하여 각각의 단계를 실제 구현하였다. 이 중에서 서명키를 생성하는 단계는 서명 시스템을 구축할때 한번만 수행하면 되기 때문에 비교 대상에서 제외하였으며, 서명 생성 단계와 서명 검증 단계의 처리 시간만을 측정하여 비교하였다. 처리 속도의 비교는 p, q의 크기에 따라 서명의 단계별로 소요시간을 측정하였으며, p, q의 범위는 사용기종(SUN 4/330)에서 표현 가능한 데이타 수형

의 범위로 한정 하였다. 즉, SUN 4/330 시스템에서 long integer를 사용할 경우 2^{31}까지 표현이 가능하므로 실험에 필요한 최대치인 고속 디지탈 서명방식의 경우 $k = p^2q$를 만족하는 k값과, RSA 서명방식의 경우 $n = pq$를 만족하는 n 값이 2^{31}을 넘지 않는 정수로 국한하여 실험을 하였다.

표 5-1 고속 디지탈 서명방식과 RSA 서명방식의 비교

방식 / 단계	고속 디지탈 서명방식	RSA 서명방식
서명키 생성	i) p, q : 큰 소수($p > q$) ii) $k = p^2q$ iii) $w = \lceil k^{2/3} \rceil$ iv) $f = k - w - 1$ v) $g = \lceil w/p \rceil$	i) $n = pg$ ii) $(\gcd(e, \phi(n) = 1)$ iii) $ed = 1 (\mathrm{mod}\ \phi(n))$
서명 생명	(전처리) ① $1 \le x_1 \le pq - 1$ $\gcd(x_1, k) = 1$ ② θ 계산 $2x_1\theta \equiv 1 (\mathrm{moc}\ k)$ (서명) $-g \le M \le k - g$, M : 메세지 ① $0 \le h \le k - 1$ $h \equiv x_1^2 + M^2 (\mathrm{mod}\ k)$ $x_2 = \lceil (f - h)/pq \rceil$ ② $0 \le x_3 \le k - 1$ $x_3 \equiv \theta x_2 (\mathrm{mod}\ k)$ ③ $1 \le S \le k - 1$ $S \equiv x_1 + x_3 pq (\mathrm{mod}\ k)$	① $S = D(m) = m^{d(j)} (\mathrm{mod}\ n)$ ② $C = E(S) = \mathrm{m}^{d(i)e(i)} (\mathrm{moc}\ n)$
서명 검증	① $f \le M^2 + S^2 < f + w (\mathrm{mod}\ k)$ ② $g \le S \le k - 1$ ③ $g \le M \le k - g$	① $S = D(c) = m^{d(i)e(i)d(j)} (\mathrm{mod}\ n)$ $= m^{d(j)} (\mathrm{mod}\ n)$ ② $m = E(S) = m^{d(i)e(i)} (\mathrm{mod}\ n)$

〈표 5-2〉에 RSA 서명방식과 새로 제안한 고속 디지탈 서명방식을 〈표 5-1〉과 같이 단계별로 나누어서 서명을 생성하는데 소요되는 시간과 서명을 검증하는데 소요되는 시간을 측정하여 그 결과를 요약하였다.

〈표 5-2〉 처리 속도의 비교

(단위 : 초)

소수의 크기 / 방식 / 단계	p = 41, q = 37		p = 131, q = 127		p = 1237, q = 1031	
	서명생성	서명생성	서명생성	서명생성	서명생성	서명생성
RSA 서명방식	2.016667	2.016667	2.110000	2.110000	2.554300	2.554300
고속 디지탈 서명방식	0.423333	0.151000	1.462130	0.144420	0.481111	0.150000

* 주) 1000회 반복 수행한 결과임

실험한 결과 RSA 서명방식은 서명 생성 기간이나 서명 검증 시간이 동일함을 알 수 있다. 이는 서명을 생성하는 절차와 검증하는 절차가 같기 때문이다. 소요 시간은 p와 q의 크기가 커짐에 따라 점차 증가함을 나타냈다. 새로 제안한 고속 디지탈 서명방식의 처리 속도는 서명을 생성하는데 있어서는 p와 q의 크기에 거의 무관함을 나타냈다.

서명을 생성하는 시간에 있어서는 새로 제안한 고속 디지탈 서명방식에 비해 평균 4.89배의 처리 속도 향상이 있었다. 이는 RSA 서명방식을 생성하는데 있어서 지수승의 합동 다항식을 이용하는 반면에 새로 제안한 고속 디지탈 서명방식은 저차 합동 다항식에 기초를 하기 때문에 계산량이 상대적으로 많이 감소 하였기 때문이며, 또한 전처리를 이용함으로써 modulo n 상에서의 곱셈에 대한 역원을 계산하는 시간을 줄일 수 있었으며, 이로 인하여 modulo 계산의 양을 대폭 간소화 시켰기 때문이다.

서명을 검증하는 시간에 있어서는 새로 제안한 고속 디지탈 서명방식이 RSA 서명방식에 비해 14.6배의 높은 처리 속도의 향상이 있었다. 이는 RSA 서명방식은 서명의 생성시와 동일하게 지수승의 합동 다항식을 이용하여 검증을 하는 반면에, 새로 제안한 고속 디지탈 서명방식은 검증의 조건으로써 간단한 3개의 부등식을 이용하여 부등식의 성립 여부로 검증을 하기 때문에 고속 디지탈 서명방식이 RSA 서명방식에 비해 상대적으로 적은 계산량을 가지고 서명의 검증을 처리할 수 있다.

따라서 새로 제안한 저차 합동 다항식을 이용한 서명의 생성과, 부등식을 이용한 서명의 검증은 그 처리 속도 면에서 매우 빠르다는 것을 실험 결과를 통하여 알 수 있었다.

(그림 5-1)은 RSA 서명방식과 새로 제안한 고속 디지탈 서명방식에 대하여 서명을 생성하는데 소요되는 시간을 p = 41, q = 37인 경우와, p = 131, q = 127인 경우, 그리고 p = 1237, q = 1031인 경우로 나누어 측정한 결과를 알기 쉽게 그래프로 나타냈다.

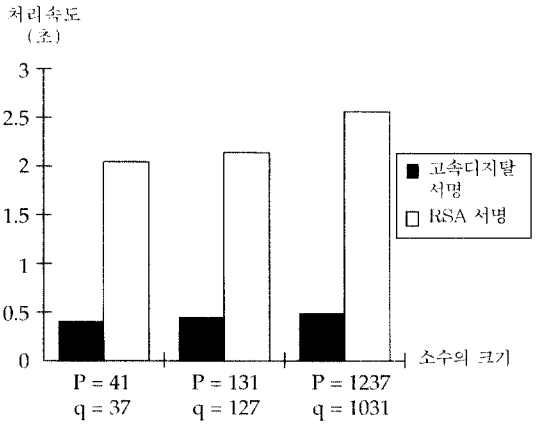


그림 5-1 서명 생성 시간의 비교

(그림 5-2)는 RSA 서명방식과 새로 제안한 고속 디지탈 서명방식에 대하여 서명을 검증하는데 소요되는 시간을 p = 41, q = 37인 경우와, p = 131, q = 127인 경우, 그리고 p = 1237, q = 1031인 경우로 나누어 측정한 결과를 알기 쉽게 그래프로 나타냈다.

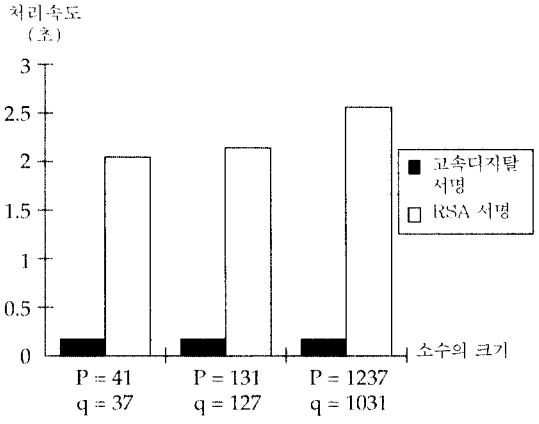


그림 5-2 서명 검증 시간의 비교

5.3 성능 분석

새로 제안한 고속 디지탈 서명방식을 서명 시스템의 안전도 즉, 비도 측면과 처리 속도 측면에서 성능에 대한 분석 및 평가를 하면 아래와 같다.

첫째, 비도 측면에서는 새로 제안한 고속 디지탈 서명방식은 기본적으로 RSA 서명방식과 같이 큰 수의 인수분해 어려움에 두고 있으며, 서명의 생성시 난수를 이용하기 때문에 RSA 서명방식보다 더욱더 안전한 서명 시스템이라고할 수 있다. 이것은 비록 큰 수를 인수분해하는 문제가 NP-complete 문제로 증명되지는 않았으나, 다항식 시간 알고리즘이 알려져 있지 않은 NP 문제이기 때문에 새로 제안한 고속 디지탈 서명방식은 계산적 실행 불가능성(computational infeasibility)을 만족한다고할 수 있다.

둘째, 처리 속도 측면에서는 앞에서 실험한 결과처럼 새로 고안한 고속 디지탈 서명방식은 RSA 서명방식에 비해 서명을 생성하는데 소요되는 시간은 평균 4.89배, 서명을 검증하는데 소요되는 시간은 평균 14.6배의 높은 처리 속도 향상이 있었다. 이것은 계산량 측면에서 새로 제안한 고속 디지탈 서명방식이 서명의 생성시에는 저차 합동 다항식의 계산을 하고, 서명의 검증시에는 간단한 부등식을 이용하여 검증을 하는 반면에, RSA 서명방식은 서명의 생성과 검증시 지수승 합동 다항식의 계산을 하기 때문에 새로 제안한 고속 디지탈 서명방식의 계산량이 현저하게 감소하였다.

실험 결과에서 RSA 서명방식은 서명의 생성과 검증에 있어서 p, q의 크기 ($2^5 \sim 2^{10}$)가 커짐에 따라 소요시간이 증가함을 나타냈다. 새로 제안한 고속 디지탈 서명방식은 서명의 생성시에 p, q의 크기 ($2^5 \sim 2^{10}$)가 증가함에 따라 소요시간이 다소 증가함을 나타내었지만, 서명의 검증시에 p, q의 크기와 거의 무관함을 나타냈다. 다만 2^{11} 이상의 큰 수는 실험에 사용한 컴퓨터 시스템(SUN 4/330)에서 long integer로 데이타를 표현할 수 있는 범위를 벗어나기 때문에 측정할 수 없었음을 밝혀둔다.

6. 결 론

지금까지 본 논문은 디지탈 서명방식 중에서 가장 안전한 시스템으로 알려져있는 RSA 서명방식의 단점인 처리 속도의 문제를 해결하기 위하여 저차 합동 다항식에 기초한 새로운 고속 디지탈 서명방식을 제안하였으며, 실제로 구현하여 RSA 서명방식과 처리 속도 측면에서 성능 평가를 하였다.

시뮬레이션을 실시한 결과 새로 제안한 고속 디지탈 서명방식은 RSA 서명방식에 비해, 서명을 생성하는데 소요되는 시간은 평균 4.89배, 서명을 검증하는데 소요되는 시간은 평균 14.6배의 높은 처리 속도 향상을 나타냈다. 이것은 서명을 생성하기 위하여 소요되는 계산량이 RSA 서명방식은 지수승 합동 다항식의 계산을 하는 반면에, 새로 제안한 고속 디지탈 서명방식은 저차 합동 다항식의 계산을 하기 때문이다. 따라서 본 논문에서 제안한 고속 디지탈 서명방식은 디지탈 서명의 처리 속도면에서 매우 빠르다는 것을 확인하였다.

앞으로의 연구 방향은 디지탈 서명에 대한 시뮬레이션을 하는데 있어서 보다 큰 수(2^{100})를 처리할 수 있는 실험 환경을 구성하는 방법과 프로그래밍을 하는 기술 등이 연구되어져야 할 것이다. 성능 향상을 위한 측면에서는 제안한 서명 알고리즘을 병렬처리 알고리즘화하여 하나의 모듈러 칩으로 설계함으로써 성능면에서 획기적인 향상을 가져올 수 있는 방안이 연구되어야 할 것이다. 또한 실제적으로 통신망 상에서 이를 이용한 보안 프로토콜의 설계 및 운용에 관한 연구가 이루어져야 할 것이다.

참 고 문 헌

[1] Davies, D.W., Applying the RSA Digital Signature to Electronic Mail, IEEE Tran. on Computer, Feb. 1983, pp. 55-62.

[2] Davies, D.W. and Price, W.L., Security for Computer Network, Joh Wiley & Sons Ltd., 1989, pp. 254-281.

[3] Diffie, W., and Hellman, M. E., New

Direction in Cryptography, IEEE Trans. on Information Theory, Vol-22, No.6, Nov. 1976, pp. 644-654.

[4] Merkle, R., and Secure Communication over Insecure Channels, Comm. ACM, Vol 21, No.4, 1978, pp. 294-299.

[5] Merkle, R., and Hellman, M., Hidling Information amd Signature in Trapdoor Knapsacks, IEEE Trans. on Information Theory, Vol-24, May.1978,pp. 525-530.

[6] Needham, R.M. and Schoeder, N.D., Using Encryption for Authentication in Large Networks for Computer, Comm. ACM Vol-2, N0.12, Dec.1978, pp. 993-999.

[7] Ong, H., Schnorr, C. and Shamir, A., An Efficient Signature Scheme Based on Quadratic Equations, Proc. 16th Annual ACM Symposium on Theory of Computing, Washington, D. C., April 1984, pp. 208-217.

[8] Ong, H., Schnorr, C. and Shamir, A., An Efficient Signature Scheme Based on Polynomial Equations, Crypto '85, 1985, pp. 37-46.

[9] Pfleeger, C.P., Security in Computing, Prentice Hall, 1989, pp. 132-136.

[10] Rivest, R., Shamir, A., and Adleman, L., A. Method for Obtaining Digital Signature and Public Key Cryptosystems, Comm. ACM, Vol-21, N0.2, Feb., 1976, pp. 120-126.

[11] Shamir, A., A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem, Proc. of the 23th IEEE Symposium on the Foundations of Computer Science (FOCS), Nov. 1982, pp. 145-152.

[12] Shamir, A., Identity-based Cryptosystems and Signature Schemes, Crypto '84, 1984, pp. 47-53.

[13] Shamir, A., A Polynomial-Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem, IEEE Trans. on Information Theory, Sep. 1984, pp. 699-704.

[14] 암호학 입문, 한국전자통신연구소, 1987. 10., pp. 115-141.

□ 著者紹介

서 장 원

1992년 2월 : 서울산업대학교 전산과
1993년 3월 ~ 현재 : 숭실대학교 대학원 전산과 석사과정
※ 관심분야 : 병렬컴퓨터 구조, 디지탈 시스템, 정보이론

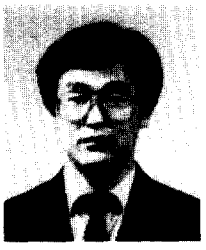
문 필 주

1988년 2월 : 숭실대학교 전산과
1992년 2월 : 숭실대학교 대학원 전산과(석사)
1993년 3월 ~ 현재 : 숭실대학교 대학원 전산과 박사과정
※ 관심분야 : 분산처리 이론, 암호학 이론, 네트워크 설계

방 혜 자

1977년 2월 : 숭실대학교 전산과
1983년 5월 : North Texas State Univ. 전산과(석사)
1993년 6월 : 숭실대학교 대학원 전산과(박사)
1977년 ~ 1980년 : 산업연구원 연구원
1984년 : 유한공전 전산과 전임강사
1985년 3월 ~ 현재 : 서울산업대학교 전산과 부교수
※ 관심분야 : 계산이론, 그래프 이론, 통신망 설계, 알고리즘

□ 著者紹介



전 문 석

1980년 2월 : 숭실대학교 전산과
1986년 12월 : Univ. of Maryland 전산과(석사)
1988년 12월 : Univ. of Maryland 전산과(박사)
1989년 8월 : Morgan State Univ. 전산수학과 조교수
1991년 2월 : New Maxico State Univ. 부설
Physical Science Lab. 책임연구원
1991년 3월 ~ 현재 : 숭실대학교 전산과 조교수
※ 관심분야 : 병렬 알고리즘, 병렬컴퓨터구조, 대규모 직접회로, 암호학 이론, 폴트톨로런스

이 철 희

1958년 6월 : 육군사관학교
1962년 8월 : Univ. of Purdue 전기공학과(석사)
1988년 2월 : 중앙대학교 대학원 전산과(박사)
1962년 9월 ~ 1973년 2월 : 육군사관학교 전자공학과 교수
1973년 3월 ~ 현재 : 숭실대학교 전산과 교수
※ 관심분야 : 네트워크 이론, 통신망 이론, 분산시스템