特輯

디지틀 서명에 관한 고찰

宋周錫, 李娥蘭
延 世 大 學 校

Ⅰ. 디지틀 서명의 개요

현재의 컴퓨터 네트워크는 수천개의 터미날을 통해 액세스가 가능한 전세계적인 규모를 갖는 것으로 발전되어 왔다. 이에 따라 보안에 대한 요구는 점차로 중요한 문제가 되고 있다. 따로 보안이 보장되지 않는 링크를 따라 송신이 이루어지는 현재의 네트워크 내에서 중요한 정보를 교환하기 위해서는 내용의 확인(validation)과 인증(authentication)이 보장되는 안전한 교환 방법이 요구된다.

내용의 확인이란 송신자가 보낸 정보의 내용이 통신도중에 제3자에 의해 변경됨이 없이 본래의 정보 그대로임을 확인할 수 있는 방법을 말하며, 인증이란 그 정보를 보낸 송신자가 A라는 것을 주장했을 때, 그것을 증명할 수 있는 방법을 말한다. 이 두가지 기능을 동시에 달성할 수 있도록 해주는 것이 디지틀 서명(digital signature)으로, 이는 송신되는 메세지에 덧붙여지거나 또는 메세지의 일부분으로 포함되어 메세지와 함께 수신자로 보내어진다.

디지틀 서명은 손으로 쓰여진 서명과 비슷한 특성을 갖는다. 그러나 디지틀 서명은 '0'과 '1'로 이루어진 비트열이므로 손으로 쓰여지는 서명과 다른 특성도 또한 가져야 한다.[1] 즉, 손으로 쓰여지는 서명은 그 사용자에 따라 항상 일정하나 디지틀 서명은 각 메세지에 따라 동일한 송신자라 할지라도 생성되는 결과가 달라야 한다. 이러한 메세지 의존적인 특성에 의하여 디지틀 서명이 변하지 않는한 메세지의 내용중의 한 비트도 변화시킬 수 없으므로, 메세지가 전달되는 과정에서 내용이 변경되지 않았다는 것을 수신자는 디지틀 서명을 통해 알 수 있게 된다. 디지틀 서명의 방법은 디지틀 서명에 사용되어지는 암호 시스템에 의해 비밀키 암호 시스템에 기반을 둔 디지틀 서명 방법과 공개키 암호 시스템에 기반을 둔 디지틀 서명 방법II)를 기반으로 한 디지틀 서명 방법으로 나누어진다. 각각의 디지틀 서명 방법에 대해 살펴 보기 전에, 다음 절에서는 이에 사용되어지는 암호 시스템을 간략하게 알아보고 넘어가도록 한다.

Ⅱ. 암호 시스템

현재 컴퓨터의 보안을 해결하기 위해 사용되는 암호화 시스템은 크게 비밀키 암호 시스템과 공개키 암호 시스템으로 나눌 수 있다.[2] 공개키 암호 시스템은 비밀키 암호 시스템에 비해 비교적 최근에 발표된 것이다. 비밀키 암호 시스템은 원문을 암호화(encryption), 복호화(decryption)하는데 있어서 동일한 키(key)를 사용하며, 이 키는 암호문을 주고 받는 당사자 둘만이 공유하는 비밀키이다.

이에 반해 공개키 암호 시스템 은 암호화 하는 절차(키)와 복호화하는 절차(키)가 서로 틀리고 암호화하는 절차(키)는 일반에게 공개되어지지만 복호화하는 절차(키)는 암호문을 받아 암호를 푸는 측밖에 알지 못한다.

비밀키 암호 시스템은 이를 사용하여 통신하는 사용자 둘사이에 쌍방향 채널을 제공할 수 있다는 장점이 있다. 즉, A와 B가 비밀키 암호 시스템을 사용하여 통신을 할 경우 하나의 키를 사용하여 A가 암호화한 메세지를 B가 받아 복호화할 수도 있고, B가 암호화하여 보낸 암호문을 A가 받아 복호화할 수도 있

다. 또, 키가 공개되지 않고 비밀로 남아 있는 한은 암호화를 통해 자동적으로 인증 메카니즘이 제공되어 진다.

즉, 비밀키 시스템에서 키를 공유하고 있는 것은 당사자인 A,B이외에는 없으므로 암호문을 보낸 것은 자연히 암호화를 할 수 있는 키의 소유자라는 것이 증명된다. 이런 장점이 있는 반면, 다음과 같은 단점 또한 갖는다. 첫번째로 통신하고자하는 당사자들사이의 비밀키의 할당 문제가 있으며, 두번째 문제는 키의 수가 통신망의 사용자의 수의 제곱에 비례하여 증가한다는 것이다.[7]

이러한 문제를 고려하여, 1976년 Diffie와 Hellman이 제안한 공개키 암호 시스템은 그 뒤 Rivest, Shamir, Adleman이 제안한 RSA를 통해 발전하였다.[8] 공개키 암호 시스템은 앞에서 든 비밀키 암호 시스템의 단점을 극복하였다. 여기서, 공개된 암호 절차 E와 비밀키의 사용자만이 아는 복호화 절차 D는 다음과 같은 특성을 가져야 한다.

(a) 메세지 M을 암호화한 뒤 복호화하면 다시 원래의 M을 얻을 수 있어야 한다. 즉, D(E(M)) = M.

(b) 절차 E와 D는 쉽게 계산할 수 있어야한다.

(c) 절차 E를 공개함으로써 D를 쉽게 계산하는 방법을 드러내지 않도록 고안되어져야한다.

(d) 메세지 M을 먼저 복호화한 뒤 암호화한 결과는 M이어야한다. 즉, E(D(M)) = M.

(a)-(d)의 특성을 만족하는 E를 trap-door 일방향 치환(permutation)이라고 한다. 함수가 아니라 치환이라고 하는 이유는 이 특성을 만족함으로 하여 모든 메세지는 어떤 다른 메세지의 치환이라고 볼 수 있기 때문이다. 이 (d)의 특성을 이용함으로써 디지틀 서명에 공개키 방식을 사용할 수 있게 된다.

비밀키 암호 시스템과 공개키 암호 시스템은 둘다 키를 사전에 교환해야 한다는 번거로움이 있다. 이를 수행하기 위해 별도로 공개키의 디렉토리를 유지하거나, 제 3 자를 통해 비밀키의 분배를 받아야 하기도 한다. ID를 기반으로한 암호 시스템에서는 이와 같은 절차들을 없애기 위하여 ID를 공개키로 사용한다. ID에 대응하는 비밀키를 생성하기위한 신뢰할 수 있는 키 생성 센터(trusted key generation center)가 존재하여 센터만이 가지고 있는 정보를 이용하여 각 사용자의 ID에 대응되는 비밀키를 생성한다.이러한 공개키 암호 시스템, 비밀키 암호 시스템, ID를 기반으로한 암호 시스템 사이의 차이를 비교해보면 그림 1과 같다.

모든 시스템에서 메세지 m은 키 ke를 사용하여 암호화되어, 암호문의 상태로 일반 사용자에게 노출된 채널을 통해 전달되어져 송신측에서 키 kd를 사용하여 복호화된다. 이 키들의 선택은 랜덤 seed k에 의해 선택된다.

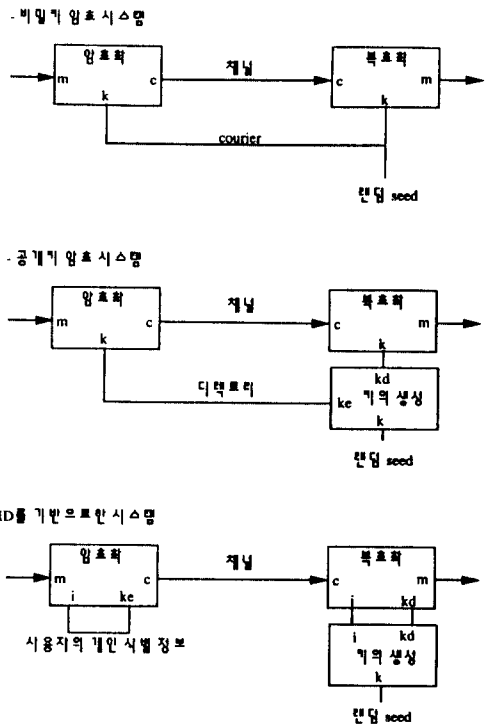

그림 1. 세 암호 시스템의 비교

비밀키 암호 시스템에서는 ke = kd = k가 되며 분리된 키 채널(예 : courier) 은 비밀(secrecy)과 확실성(authenticity)을 보존할 수 있어야한다. 공개키 암호 시스템에서는 암호화키와 복호화키는 두개의 다른 함수 fe와 fd를 통해 k로 부터 ke = fe(k), kd = fd(k) 로 유도된다. 분리된 키 채널(예 : 디렉토리)은 키의 확실성만 보존할 수 있으면 된다. ID 를 기반으로 하는 암호 시스템에서는 암호화 키는 ke = i로 사용자의 개인 식별 정보이고 복호화 키는 i와 k로 부터 kd = f(i, k) 와 같이 유도된다.

또한 사용자들간의 분리된 키 채널은 완전히 제거될 수 있다. 그대신 사용자는 네트워크 가입시에 키 생성 센터와 한번의 교류가 필요하다.

Ⅲ. 암호 시스템에 의한 디지틀 서명의 분류

1. 비밀키 암호 시스템을 기반으로한 디지틀 서명

1) Diffie-Lamport 서명 기법

이 기법에서는 n비트의 메세지에 대해 어떠한 압축 방법도 적용하지 않고 디지틀 서명을 생성한다.[6] 송신자가 n비트의 메세지를 보내고자 할 때, 우선, 다음과 같이 n개의 키의 쌍을 선택한다.

$$(K_{10},K_{11}),(K_{20},K_{21}),\cdots(K_{n0},K_{n1}) \tag{1}$$

이들은 송신자만이 알고 있는 키이다. 그리고나서 송신자는 S와 R이라고 하는 다음과 같은 열을 생성한다.

$$S=\left[(S_{10},S_{11}),(S_{20},S_{21}),\cdots(S_{n0},S_{n1})\right]$$
$$R=\left[(R_{10},R_{11}),(R_{20},R_{21}),\cdots,(R_{n0},R_{n1})\right] \tag{2}$$

이들은 후에 디지틀 서명을 인증하는 과정에서 사용된다. 송신자가 이들을 생성하는 방법은 우선 S의 원소들을 임의로 선택하고, 선택된 S로부터 다음과 같은 암호화를 거쳐 R을 생성한다.

$$R_{ij}=E_{Kij}(S_{ij}) \quad i=1,\cdots n, j=0,1 \tag{3}$$

암호시스템의 구조에 의해 S와 R의 길이는 정해진다. 이 S와 R은 사전에 수신자에 알려져야 한다. 그러므로 S와 R은 공공 레지스터(public register)에 저장되고 송신자와 수신자 이외의 제3자는 이 데이타를 읽을 수는 있으나 내용을 변경할 수는 없다. n비트의 메세지 M이 M = (m1,...,mn), mi = 0,1 (1 〈= i 〈= n)이라고 할 때 M의 디지틀 서명은 다음과 같은 비밀키의 열로 이루어진다.

$$SG(M)=(K_{1i_1},K_{2i_2},\cdots K_{ni_n}) \tag{4}$$

이때 mj = 0 이면 ij = 0 이고 mj = 1 이면 ij = 1(1 〈= j 〈= n)이다. 예를 들어, 6비트의 메세지 M = (1,0,1,1,0,0)이 있다고 할때 디지틀 서명은 다음과 같이 된다.

$$SG(M)=(K_{11},K_{20},K_{31},K_{41},K_{50},K_{60})$$

수신자는 SG(M)을 받아 해당하는 S와 R의 쌍이 그 키에 의해 암호화하면 서로 일치하는가를 보아 인증을 할 수 있다. 앞의 예의 경우에는 송신자로부터 알려진 키 (K11, K20, K31, K41, K50, K60)를 가지고, 수신자는 공공 레지스터로부터 이 키들의 인덱스에 해당하는 S의 원소를 읽어와 다음과 같이 암호화한다.

$$E_{11}(S_{11}),\ E_{20}(S_{20}),E_{31}(S_{31})$$
$$E_{41}(S_{41}),\ E_{50}(S_{50}),E_{60}(S_{60})$$

이 값의 결과가 (R11, R20, R31, R41, R50, R60)과 같다면 수신자는 디지틀 서명 SG(M)을 진짜로 받아 들인다.이 방법의 가장 큰 단점은 서명의 길이가 길다는 것이다. 이런 단점을 극복하기 위해서는 Diffie-Lamport 기법에 압축을 적용하여 n비트의 메세지를 r비트의 요약으로 줄인 후 디지틀 서명을 생성할 수 있다.[9],[그림 2]

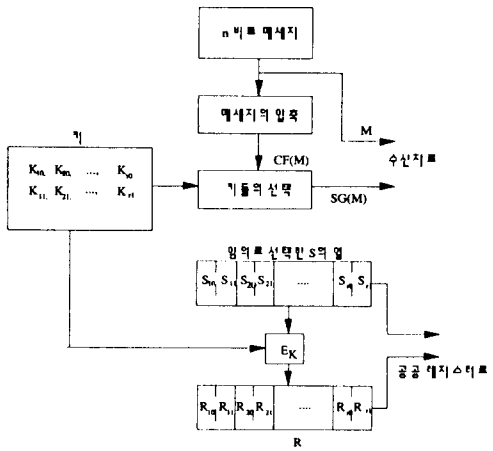

그림 2. Diffie-Lamport 서명 (송신자측)

이 경우 수신자의 인증 과정도 몇 단계를 더 거쳐서 이루어져야 한다. 첫째로, 수신자는 메세지의 요약을 생성해야 한다 - 즉, 압축 기법은 공개되어 있어야 할 것이다. 그 다음 단계로 수신자는 공공 레지스터에서 필요한 r개의 S의 원소를 읽어와 SG(M)에 포함된 키로 암호화하여 같이 읽어온 r개의 R의 원소와 비교한다.[그림 3] 비교한 결과가 같으면 인증은 성공하여 그 서명은 받아들여진다.

이 기법에서는 전송되는 SG(M)을 통해 r개의 키가 노출된다. 반면, 그것들과 쌍을 이루는 또다른 r개의 키는 아직은 비밀이 보장된 상태이다. 그러나,

이를 반복하여 사용할 경우, 궁극적으로는 모든 키가 노출이 될 수도 있다. 그런 위험을 막기 위해서 이 기법을 사용할 경우에는 생성된 키는 단 한번만 사용하도록 한다.

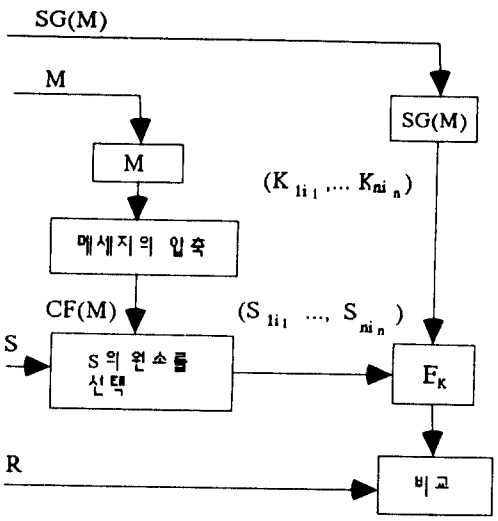

그림 3. Diffie-Lamport 서명의 인증

2) Rabin 서명 기법

1978년 Rabin이 제안한 이 기법은 송신자가 2r개의 임의의 키를 생성함으로써 시작된다. 패러미터 r은 요구되는 보안의 수준에 의해 결정되는 수치이다. 이들 키들이 K1, K2, K3, . . . , K2r 이라고 하자. 이들 키들은 송신자만이 알고 있는 비밀값들이다. 다음으로 송신자는 수신자가 검증할 때에 사용할 다음과 같은 두 수열을 생성한다.

$$S=(S_1,S_1,\cdots S_{2r}),\ R=(R_1,R_2,\cdots R_{2r})$$

수열 R은 다음의 함수를 통해 수열 S 로부터 생성된다.

$$R_i=E_{Ki}(S_i)\ \ i=1,\cdots,2r$$

이렇게 생성된 수열 S와 R은 읽을 수만 있는 공공 레지스터에 저장된다. 서명은 다음과 같은 절차로 이루어진다. [그림 4] 메세지의 요약의 CF(M)은 앞에서 정해진 2r 개의 임의의 키에 의해 암호화된다. 다음과 같은 그 결과가 메세지 M의 디지틀 서명으로써 M과 함께 전송되어진다.

$$SG(M)=(E_{K1}[CF(M)],\cdots,E_{K2r}[CF(M)]) \qquad (6)$$

이 디지틀 서명을 검증하기 위해서 수신자는 r개의 1과 r개의 0으로 이루어진 2r비트의 임의의 수열을 선택한다.

이 수열의 복사가 송신자로 보내지면, 송신자는 이 2r 비트의 수열을 사용하여 r개의 원소로 이루어지는 키의 부분집합을 생성한다. 즉 2r비트 수열의 i번째 원소(i = 1, ... ,2r) 가 1이면 Ki 는 이 집합에 포함된다. 이 집합은 다시 수신자로 보내어지고, 수신자는 이 키의 집합을 인증하기 위해서 공공 레지스터의 S를 이 키들에 의해 암호화하여 각각에 해당되는 R과 비교를 한다. [그림 5]

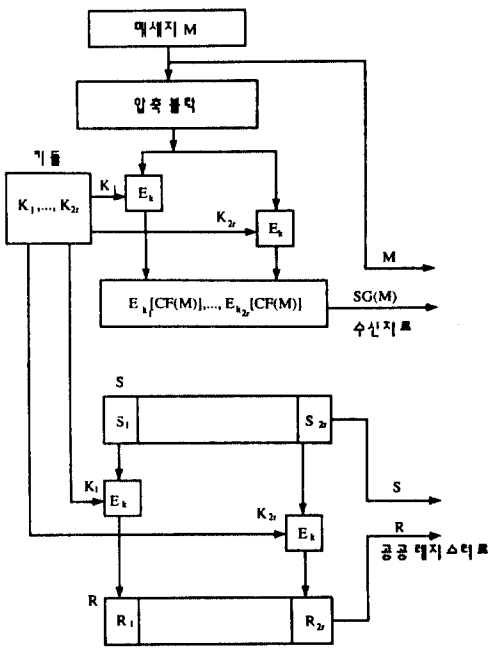

그림 4. Rabin 서명 기법의 서명

키가 인증이 되면 수신자는 메세지 M의 요약 CF(M)을 생성하여 다음과 같이 암호화한다.

$$E_{K_{i1}}[CF(M)],\cdots,E_{K_{ir}}[CF(m)] \qquad (7)$$

이때 {Ki1, ... Kir}은 수신자에게 알려진 키

의 부분집합이다. 식(7)의 각 원소는 식(6)의 적절한 부분집합과 비교된다. 만약 두집합이 동일하다면 수신자는 이 디지틀 서명의 인증에 성공한 것이다. Rabin 의 서명 기법은 이 기법의 특성상 매번 새로운 비밀키를 생성해야 한다.

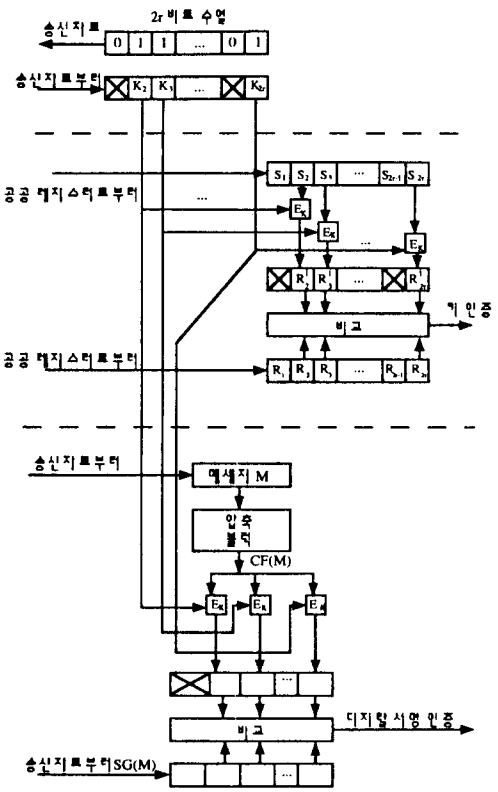

그림 5. Rabin 서명 기법의 인증

2. 공개키 암호 시스템을 기반으로한 디지틀 서명

1) RSA 서명 기법

RSA 암호 시스템을 사용하는 사용자는 우선 두개의 큰 소수 p와 q를 임의로 선택하여, p와 q의 곱 m을 계산한다. m은 공개키의 일부분으로 공개되지만 p와 q는 비밀키의 소유자만이 알고 있는 값이다. 여기서 m은 충분히 큰 값으로 암호를 깨고자 하는 제3자가 m으로부터 p와 q를 유도해내기가 거의 불가능할 정도여야한다. 이 요구를 만족시키기 위해서는 m이 이진수로 표현될 때 500자리수 이상이어야한다. 암호화될 메세지 x는 0과 m-1 사이의 수여야하고(이 범위를 넘어가는 메세지는 적절히 나누어 이 범위안에 들도록 한다.) x의 암호문 y는 $y = x^e \pmod m$으로 얻어지고, y로부터 x로 복호화하는 함수는 $x = y^d \pmod m$이 된다. 여기서 e와 d는 p-1과 q-1에 대해 서로 소이며, 또 $ed = 1 \pmod n$을 만족해야 한다. 단 n = (p-1)(q-1)이다. 이를 만족하는 e와 d에 대해서 앞의 두 함수 사이에는 서로 역함수의 관계가 성립된다. 이렇게 정해진 e는 m과 함께 공개키로서 공개되어지고 d는 비밀키로 사용된다. RSA 암호 시스템에서 A의 공개키로 M을 암호화하는 것을 $E_{KA}(M)$으로, 복호화하는 것을 $D_{KA}(M)$로 나타내기로 할때, 다음과 같은 방법으로 디지틀 서명의 기능과 메세지의 보안 기능을 제공할 수 있다.

송신자 A는 메세지 M을 자신의 비밀키로 암호화한다. 그런 후 수신자 B의 공개키로 암호화하여 이 결과인 $E_{KB}(D_{KA}(M))$을 송신한다. 수신자는 $E_{KB}(D_{KA}(M))$을 받아 자신의 비밀키로 복호화하고, 그 결과를 송신자 A의 공개키로 암호화하여 보내어진 메세지 M과 계산의 결과가 같은 경우 서명은

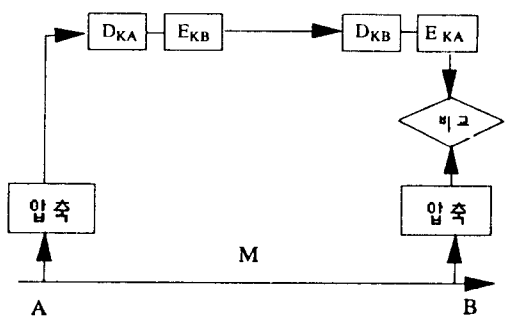

그림 6. RSA 서명 기법

정당하게 받아 들여진다. 이와 같은 디지틀 서명 방법에서는 한가지 문제가 발생한다. 즉 A와 B사이에서 사용되는 mA와 mB는 서로 값이 다르므로 복호화 과정 D_{KA}, 즉 디지틀 서명의 결과가 암호화 E_{KB}의 입력의 범위를 벗어날 수도 있다. 이런 블럭킹(blocking) 문제를 해결하기 위해 Kohnfelder는 다음과 같은 해결방안을 제시하였다.[5] 즉 modulus 값이 작은 것을 먼저 계산하고 큰 값을 나중에 계산하도록 두 값 mA와 mB의 대소를 비교하여 암호화와 디지틀 서명의 순서를 조정해 준다. modulus값 mA와 mB은 일반적으로 공개된 데이타이므로 이 해결방법은 이론적으로는 수행이 가능하다. 그러나, 실제 구현에서는 이런 식으로 modulus값의 대소에 따라

디지틀 서명과 암호화의 문제를 바꾸어서 문제를 해결할 수는 없다. 암호화는 메세지가 전달되는 도중의 보안을 위한 것이므로 전달이 끝난 후에는 즉시 복호화가 되고 그에 대해서는 더이상 고려를 하지 않으나, 디지틀 서명의 경우는 틀리다. 디지틀 서명은 보통 서류에 손으로 기입된 서명과 같이 통신이 끝난 후에도 서명된 상태로 메세지와 함께 남아서 후에 그 메세지를 사용하고자 하는 사용자가 필요한 때에 인증을 할 수 있어야 한다. 그러므로 실제로는 디지틀 서명이 암호화보다 항상 먼저 행해져야 한다. 그래서 제시된 두번째 방법에서는 메세지 M을 압축하여 요약 CF(M)을 생성한 뒤 이를 비밀키로 암호화함으로써 디지틀 서명 $D_{KA}(CF(M))$를 생성하여 메세지 M과 함께 수신자로 보낸다. 수신자는 이를 받아 M은 알려진 압축 기법에 의해 압축을 하고 디지틀 서명 $D_{KA}(CF(M))$는 송신자의 공개키로 암호를 푼다. 압축과 암호를 푼 두 결과가 같으면 디지틀 서명은 인증이 된 것이다.

2) Elgamal의 이산대수 문제를 기반으로한 서명 기법

이 기법은 Diffie-Hellman의 키 분배 방식을 기반으로 한다. 그러므로 우선 Diffie-Hellman 키 분배 방식을 먼저 알아보기로 한다.

A와 B사이에 비밀키 K_{AB}를 공유하고자 할 때 A와 B는 각각 임의의 수 xA와 xB를 선택한다. xA와 xB는 각각 자신만이 아는 비밀 값이다. p는 매우 큰 소수(prime number)이고 는 GF(p)의 원시근이며 이 두 값은 공개되어 있어 공공 레지스터에 저장되어 있다. 그 다음, A는 $yA = a^{xA} \bmod p$를 계산하여 B로 보내고 B는 $yB = a^{xB} \bmod p$를 계산하여 A로 보낸다. 그뒤 둘 사이의 비밀키 K_{AB}는 다음과 같이 계산된다.

$$\begin{aligned} K_{AB} &= a^{xAxB} \bmod p \\ &= yA^{xB} \bmod p \\ &= yA^{xA} \bmod p \end{aligned} \tag{8}$$

A와 B는 위의 계산을 쉽게 할 수 있으나 다른 xA 또는 xB값을 모르는 한 제3자가 계산하기는 거의 불가능하다. 단 이 p를 선택할 때, p-1이 적어도 한 개 이상의 인수를 갖도록 정해야 한다. 만약 이 조건이 만족되지 않는 p를 택했을 때, 위의 이산대수의 문제는 비교적 쉽게 풀려 암호가 깨질 위험성이 있다.

A가 B로 메세지 m(0 〈= m 〈= p-1)을 보내고자 할 때 위의 키분배 방식을 기반으로한 서명 기법은 다음과 같이 이루어진다.[3] 이때 A의 공개키는 a, p와 함께 yA = axA mod p가 되고 메세지 m의 디지틀 서명에 사용되는 비밀키는 xA가 되며 xA를 모르는 한 아무도 서명을 위조할 수는 없다. 메세지 m에 대한 서명은 (r,s)(0 〈= r,s 〈= p-1)의 쌍으로 이루어지며 이때 r과 s는 다음 방정식을 만족시키는 값이다.

$$a^m = yA^r r^s \bmod p \tag{9}$$

자세한 디지틀 서명의 절차는 다음과 같다. 우선 gcd(k, p-1) = 1을 만족하는 0과 p-1 사이의 임의의 수 k를 하나 선택한다. 이 k 값을 가지고 다음과 같이 r을 계산한다.

$$r = a^k \bmod p \tag{10}$$

이때 (9)의 식은 다음과 같이 쓸 수 있다.

$$a^m = a^{xr} a^{ks} \bmod p \tag{11}$$

그러므로 s는 다음 식을 이용해 풀 수 있다.

$$m = xr + ks \bmod (p-1) \tag{12}$$

만약 k가 gcd(k, p-1) = 1의 조건을 만족하도록 선택되어졌다면 (12)의 식은 근을 갖는다. 따라서 이렇게 구해진 r과 s를 디지틀 서명으로 하여 메세지 m과 함께 수신자 B에게 보내어진다. B는 m, (r,s)를 받아서 서명의 인증을 하게 되는데 식(9)을 이용하면 이는 쉽게 수행할 수 있다. 즉 a와 p는 공공 레지스터 내지는 공개된 화일등에 각 사용자마다 저장되어 있는 값이므로 식(9)의 양변을 이 값들을 이용해 각각 계산하여 얻어진 결과가 비교하면 된다. 이렇게 해서 얻어진 디지틀 서명은 m의 두배의 크기가 된다.

3. ID를 기반으로한 디지틀 서명

1) Shamir 서명 기법

이 디지틀 서명은 다음과 같은 검증(verification) 조건을 기반으로 수행된다.[10]

$$S^e = it^{f(t,m)} (\bmod n) \tag{13}$$

- m : 메세지
- s, t : 디지틀 서명
- i : 사용자의 개인 식별 정보
- n : 두 큰 소수 p, q의 곱
- e : (p - 1)(q - 1)과 서로 소인 큰 소수
- f : 일방향 함수

파라메터 n, e와 함수 f는 키 생성 센터에 의해 선택되어진다. 그리고, 모든 사용자들은 공통된 n, e의 값과 공통된 를 사용하여 서명을 하게 된다. 이러한 값들은 공개되어질 수 있는 것들이지만, n의 인수 분해(즉, p와 q의 곱)는 키 생성 센터만이 아는 비밀 정보이다.

사용자들 사이에서 오직 다른 점은 i의 값과 그에 대응되는 비밀 키인 g이다. g는 다음의 식을 만족하는 유일한 수로 구해진다.

$$g^e = i(\mathrm{mod}\ n)$$

g의 값은 비밀 정보를 가지고 있는 키 생성 센터에서는 비교적 쉽게 구할 수 있다. 그러나 n의 인수분해를 모르는 이상 g의 e제곱근을 구하기란 불가능하다. 비밀 키 g를 아는 경우는 n의 인수분해를 모르더라도 어떤 메세지의 디지틀 서명은 다음과 같은 간단한 방법을 통해 구해진다. 메세지 m의 서명을 하기 위해서는 사용자는 임의의 수 r을 선택하여 다음 식을 계산한다.

$$t = r^e(\mathrm{mod} n)$$

그러면 검증 조건의 식(13)은 다음과 같이 다시 쓸 수 있다.

$$S^e = g^e r^{ef(t,m)}(\mathrm{mod}\ n)$$

여기서 e는 (p-1)(q-1)과 서로 소이므로, 위의 식에서 지수의 공통 인수인 e는 제거할 수 있다.

$$s = gr^{f(t,m)}(\mathrm{mod}\ n)$$

따라서 g를 아는 경우에는 위의 식으로부터 디지틀 서명 s는 곱셈만으로 쉽게 구해진다.

3. Fiat-Shamir 서명 기법

이 기법은 영지식 대화형 증명과 Shamir의 ID를 기반으로한 기법을 결합하여 제안된 방식이다.[4] Shamir의 서명 기법과 마찬가지로 n의 인수분해가 알려지지 않았을 때에는 어떤 수의 모듈라 제곱근을 구하기가 어렵다는 것에 기반을 두고 있다. 또한 키 생성 센터가 존재하여 각 사용자마다 사용자의 비밀 키, 개인 식별 정보등을 저장한 스마트 카드를 발급한다. 키 생성 센터는 우선 공개되는 modulus n과 임의의 스트링들을 [0,n)사이의 값으로 대치시켜 주는 의사 랜덤 함수 f를 선택한다. n의 인수분해 pq는 센터만이 알고 있는 비밀 정보이다. 함수 f는 진짜 랜덤 함수와 다항식에 의해 제한된 계산을 통해 구분할 수 없는 함수이어야 한다. 또한, 키 생성 센터계산한다.는 각 사용자의 비밀키를 다음과 같은 순서로 생성한다. 여기서 I는 각 사용자의 개인 식별 정보이다.

1. 작은 수인 j에 대한 $v_j = f(I, j)$를 계산한다.
2. v_j가 모듈라 n에서의 평방 잉여가 되는 서로 다른 k개의 j를 뽑아 각 v_j의 가장 작은 제곱근 s_j를 구한다.
3. I, k개의 s_j와 각각의 인덱스를 포함한 스마트 카드를 발급한다.

디지틀 서명의 생성과 검증은 다음과 같은 절차를 통해 이루어진다.

- 디지틀 서명의 생성 절차

1. A는 범위 [0,n)의 임의의 수 $r_1, \ldots, r_t$를 뽑아, $X_i = r_i 2 \ (\mathrm{mod}\ n)$를 계산한다.
2. A는 $f(m, X_1, \ldots, X_t)$를 계산하여 결과의 처음 kt비트를 eij(1<=i<=t, 1<=j<=k)의 값으로 사용한다.
3. A는 $yi = r_i \prod s_j^{e_{ij}} (\mathrm{mod}\ n)$를 계산하여 I, m, eij행렬과 모든 yi값들을 B로 보낸다.

- 디지틀 서명의 검증 절차

1. B는 $vj = f(I, j)$ $(j = 1, \ldots, k)$를 계산한다.
2. B는 $z_i = y_i^2 \prod V_j^{e_{ij}} (\mathrm{mod}\ n)$ 를 계산한다.
3. B는 f(m, z1,...,zt)의 결과와 eij가 같은지를 검사한다.

Ⅳ. 디지틀 서명의 응용 분야

디지틀 서명의 가장 기본적인 기능은 네트워크를 통해 교환되는 메세지의 인증을 제공해 주는 것이다.

이 기능의 전형적인 응용분야로는 은행과 고객간의 사무처리, 사령관으로부터 그 부대로의 군사 명령, 그리고 개인과 단체사이의 계약등의 내용을 포함하는 다양한 문서의 교환등을 들 수 있다. 그러한 경우의 대표적 예로 전자식 자금 이동(EFT : Electronic Funds Transfer) 시스템을 들 수 있다. 그 외에 디지틀 서명이 이용되는 분야에 소프트웨어의 배분이 있다. 네트워크상의 모든 노드들에게 개선된 소프트웨어를 배분하는 기능을 수행하는 중앙 소프트웨어 지급 센터가 있다고 하자. 개선된 소프트웨어를 각 노드에 배분할 때에는 항상 중앙 지급 센터가 디지틀 서명을 한뒤 배분하고, 소프트웨어를 수신한 노드는 반드시 인증을 한 뒤 사용을 한다. 이렇게 함으로써 고의로 잘못된 소프트웨어를 네트워크내에 퍼뜨리려는 침입자로부터 각 노드를 보호할 수 있고, 또 실수로 다른 소프트웨어를 실행시키는 것을 막을 수도 있다. 디지틀 서명은 운영체제의 보안에도 응용될 수 있다. 즉, 정당한 권한이 부여되지 않은 프로그램이 수행되는 것을 막기 위하여 각 프로그램은 그 프로그램의 작성자와 authority에 의한 서명을 포함하게 한다. 하드웨어가 정당한 서명이 없는 프로그램의 수행은 하지 못하게 함으로써 이 서명은 소프트웨어 바이러스를 감지하는 한 수단이 되기도 한다.

앞의 예들로부터도 알 수 있듯이 디지틀 서명의 중요성은 나날이 증가할 것으로 보인다. 이에 따라 앞으로 사용될 디지틀 서명의 표준화 문제도 고려되어져야 할 것이다. 1991년 8월 미국에서는 NIST(Na-tional Institute of Standards and Technology)에 의해 DSA(Digital Signature Algorithm)이 기밀이 아닌 정부의 정보와 보안이 요구되는 공공의 데이타를 위한 디지틀 서명의 표준안으로 제안되었다. 그러나 다수의 암호학자들과 그 분야에 종사하는 사람들에 의해 여러 문제점이 지적되어 표준안으로 채택되기에는 어려움이 있을 것같다.[11] 무엇보다도, 현재 미국 컴퓨터 산업시장의 3분의 2이상이 RSA를 사용하고 있고, ISO, CCITT, SWIFT등의 국제 표준 조직들도 표준안으로써 RSA를 채택하고 있다는 점도, 새로운 독자적인 앨거리듬을 이용한 DSA를 표준안으로 채택하는 것의 장애가 된다. 디지틀 서명의 표준화에 대한 문제는 앞으로도 더욱 고려되어야할 문제이다.

參 考 文 獻

[1] Akl, S. "Digital Signatures: A Tutorial Survey."Computer, vol.16, no.2, Feb 1983, pp.15-26.

[2] Diffie, W., and Hellman, M. "New Direction in Cryptography." *IEEE Trans.* Info. Theory, vol.IT-22, no.6, Nov 1976, pp. 644-654.

[3] Elgamal, T. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." *IEEE Trans. Info.* Theory, vol.IT-31, no.4, July 1985, pp.469-472.

[4] Fiat, A. and Shamir, A.,"How To Prove Yourself: Practical Solutions to Identification and Signature Problems," Proceceedings of CRYPTO '86, Lecture Notes in Computer Science, no.263, Springer Verlag 1987.

[5] Kohnfelder, L."On The Signature Reblocking Problem in Public-Key Cryptosystems." Comm ACM, vol.21, no.2 Feb 1978 pp.179.

[6] Meyer, C., and Matyas, S. Cryptography: A New Dimension in Computer Data Security, Wiley 1982.

[7] Pfleeger, C. Security in Computing, Prentice-Hall 1989.

[8] Rivest, R. et al. "A Method of Obtaining Digital Signatures and Public-Key Cryptosystems." Comm ACM, vol.21, no.2 Feb 1978, pp.120-126.

[9] Seberry, J., and Pieprzyk, J. Cryptography - An Introduction to Computer Security, Prentice-Hall 1989.

[10] Shamir, A.,"Identity-based Cryptosystem and Signature Schemes,"

Proceeedings of CRYPTO '84, Lecture Notes in Computer Science, no.196, Springer Verlag 1985
[11] "Debating Encryption Standards.","The Digital Signature Standard Proposed by NIST.","Responses to NIST's Proposal." Comm ACM, vol.35 no.7 July 1992 pp.32-54

筆 者 紹 介

宋 周 錫

1953年 3月 2日生
1976年 2月 서울 대학교 전기공학과 졸업(학사)
1979年 2月 한국 과학원 전기전자과 (석사)
1988年 2月 Univ. of California at Berkeley 전산과학과 (박사)

1979年 3月 ~ 1982年 2月 한국 전자통신 연구소 전임 연구원
1982年 2月 ~ 1982年 6月 중앙 전기 주식회사 개발자문
1983年 9月 ~ 1985年 12月 Univ. of California at Berkeley Teaching Assistant
1985年 12月 ~ 1988年 8月 Electronic Research Lab Research Assistant
1988年 8月 ~ 1989年 9月 Naval Postgraduate School Assistant Professor
1989年 3月 ~ 1992年 2月 연세대학교 전산과학과 조교수
1992年 3月 ~ 현재 연세대학교 전산과학과 부교수

주관심 분야 : B-ISDN, 컴퓨터 보안, 컴퓨터 네트워크, 프로토콜 엔지니어링

李 娥 蘭

1970年 1月 4日生
1988年 3月 ~ 1992年 2月 연세대학교 전산과학과 (학사)
1992年 3月 ~ 1994年 2月 연세대학교 전산과학과 (석사)
1994年 3月 ~ 현재 연세대학교 전산과학과 (강사)

주관심 분야 : 컴퓨터 네트워크, 컴퓨터 보안, 암호학