소 특 집

공개키 기반구조와 전자서명

이 향 진, 이 홍 섭
한국정보보호진흥원

현대사회는 신뢰를 기반으로 하는 고도의 지식 정보화 사회로 급변하고 있으며, 이에 따라 사이버 공간에서 유통되는 정보의 가치 또한 커지고 있다. 특히, 사이버 공간상에서 이루어지는 인터넷 뱅킹이나 증권거래 등과 같은 전자 거래는 온라인으로 수행된다는 특성으로 인하여 거래하는 상대방을 확인할 수 없고, 전송되는 메시지의 위·변조 위협이 항상 존재하게 된다. 이는 신뢰를 기반으로 하는 전자상거래의 발전을 크게 저해하는 요소로 작용되고 있으며, 이를 해결하기 위한 방법으로 공개키 암호 방식의 사용이 널리 확대되고 있다. 공개키 암호 방식에서는 각 사용자의 공개키를 안전하게 관리하고 전달하며, 공개키에 대한 정당성을 검증할 수 있는 메커니즘이 필요한데, 이에 대한 해결방안으로 공개키 기반구조(PKI : Public Key Infrastructure)가 대두되었다.

PKI는 인터넷과 같은 개방된 분산 네트워크 환경에서 전자상거래 시스템과 같은 정보 시스템에 안전성 및 신뢰성을 높이기 위한 기반구조로, 여러 보안 알고리즘을 이용하여 전송메시지의 기밀성(confidentiality) 및 무결성(integrity), 네트워크 상에 연결된 각 사용자에 대한 인증(authentication)과 거래 수행에 대한 부인방지(non-repudiation) 등의 정보보호 서비스를 제공한다.

본 고에서는 PKI의 개념과 구성 및 이를 구성하는 보안 알고리즘으로 암호 알고리즘과 해쉬함수에 대해 소개하고, 특히, PKI를 구성하는 핵심 기술인 전자서명에 대해 그 개념과 중요 알고리즘들을 소개한다.

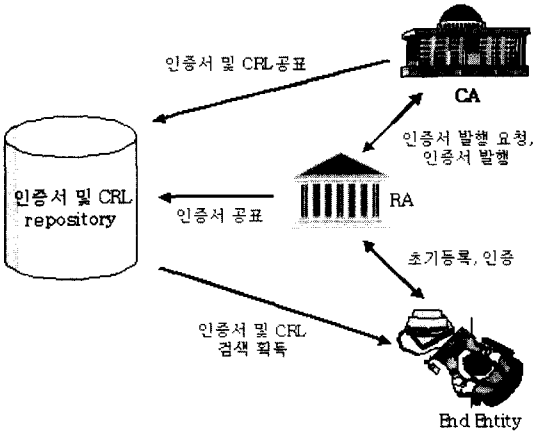
Ⅰ. 공개키 기반구조

현대 사회가 점차 고도의 정보화 사회로 발전해 가면서 정보를 보호하는 문제가 크게 대두되었으며, 이를 해결하기 위한 일반적인 방법으로 암호화 알고리즘이 사용되었다.

초기에 암호 방식은 송신자와 수신자가 암·복호화 과정에서 동일한 비밀키(secret key)를 사용하는 방식으로 비밀키 암호 방식이라 불렸다. 이 방식은 송·수신자간에 사전에 안전한 채널을 이용한 비밀키의 교환이 있어야 한다는 문제점과 통신하는 상대에 따라 각각 다른 비밀키를 관리해야 하므로 네트워크에 가입자 수가 증가함에 따라 관리해야 하는 키의 수가 급증한다는 문제점으로 인해 인터넷과 같은 환경에서의 전자상거래에 이용되기 어렵다는 단점을 가진다.

1976년, Diffie-Hellman[1]이 제안한 공개키 암호 방식은 송신자의 암호화키와 수신자의 복호화키가 서로 다른 암호 시스템을 이용함으로써 이러한 비밀키 암호 방식의 문제점들을 해결하고 있다.

공개키 암호 방식은 수학적으로 연관된 서로 다른 공개키·개인키(public key·private key) 쌍을 생성하여 공개키를 공개함으로써 송·수신자간에 비밀키의 교환 없이도 공개된 키를 이용한 암호화가 가능하다. 특히, 공개키 암호 방식은 비밀키 암호 방식이 제공하는 기밀성 및 무결성 뿐만 아니라 전자서명을 이용한 인증 및 부인 방지의 기능도 제공한다. 그러나 공개키 암호에서 공개키는 공개된 정보로 위·변조가 가능해지므



〈그림 1〉 공개키 기반 구조의 구성 요소

로 공개키의 무결성에 대한 문제가 발생할 수 있다. 이러한 공개키의 무결성에 대한 문제를 해결하기 위해 나온 것이 공개키 기반구조이다.

공개키 기반구조는 공개키 암호 방식의 한 응용으로 신뢰할 수 있는 제3자가 사용자의 정보와 그의 공개키를 자신의 서명키로 서명한 뒤 이를 배포함으로써 공개키의 위·변조 문제를 해결한다. 이 때 신뢰할 수 있는 제3자를 인증기관(CA : Certificate Authority)이라 하며, 사용자의 정보와 공개키에 인증기관이 서명한 데이터를 인증서(certificate)라고 한다. 즉, 공개키 기반구조에서 사용자는 인증서를 통해 제공되는 무결성이 보장된 공개키를 이용하여 서로 비대면인 사용자들과도 안전한 전자 상거래를 수행할 수 있다.

공개키 기반 구조는 〈그림 1〉과 같이 공개키 인증서, 인증기관, 등록기관, 인증서 보관소, 사용자 등의 요소를 가지며 각각의 기능은 다음과 같다.[2]

- 인증서(certificate) : 객체의 공개키와 사용자를 연결하기 위하여 사용자의 유일한 이름과 사용자의 공개키 및 인증 정책 등의 정보를 인증기관의 서명용 비밀키로 서명한 문서로 공개키의 정당성 검증을 위해 사용된다.
- 인증기관(CA : Certification Authority): 인증 정책에 따라 개인, 조직, 또는 다른 개체와 다른 인증기관들에게 인증서를 발행하거나 취소할 수 있는 신뢰성 있는 제3의 기관으로 모든 인증기관들은 자신의 키 쌍을 생성하고 선택적으로 사용자의 키를 생성할 수 있다.
- 등록기관(RA : Registration Authority): 인증서를 발행하는 인증기관과 발급받는 객체 사이의 중간 매개체 기능을 수행하는 기관으로 인증기관은 등록기관을 신뢰하고 인증서 발행 요청을 받아들인다.
- 저장소(repository) : 인증서 및 인증서 폐지 목록을 저장하고 공개하는 저장소로 사용자들로 하여금 그 정보에 접근할 수 있는 서비스를 제공한다.
- 사용자(End Entity) : 인증 경로의 유효성과 서명의 유효성을 검증하는 객체로 공개키 기반구조내의 사용자는 사람뿐만 아니라 사람이 이용하는 시스템 모두를 의미한다.

공개키 기반구조에서 인증기관은 각 사용자의 공개키의 정당성을 보장하기 위해 자신의 비밀키로 서명한 공개키 인증서를 발급하고 사용자들은 인증기관의 서명을 검증함으로써 인증서의 정당성을 확인할 수 있게 된다.

II. 암호 알고리즘

암호 알고리즘은 공개키 기반구조에서 전송되는 메시지의 기밀성을 제공하기 위한 정보보호기술로, 인가된 사람에 한해서만 원하는 정보를 얻을 수 있도록 하며, 인가되지 않은 자에 대하여 아무런 정보도 노출시키지 않는 기술이다.

암호화 기술을 구현하기 위한 방법으로서 키의 형태에 따라 암·복호화 키가 같은 비밀키 암호화 알고리즘과 암·복호화 키가 다른 공개키 암호화 알고리즘으로 크게 구분할 수 있다.

1. 비밀키 암호 알고리즘

비밀키 암호 알고리즘은 대칭키 암호(symmetric cryptography)라고도 불리는데, 송신자의 암호화키와 수신자의 복호화키가 같은 암호 시스템으로 암·복호화 과정에 동일한 비밀키(secret key)를 사용하여 입력된 평문 데이터로부터 랜덤한 암호문 데이터를 생성한다.

비밀키 암호 알고리즘은 컴퓨터 구현시 속도가 매우 빠르다는 장점을 가지고 있으나, 인터넷 환경에서 비대면인 사용자와의 거래를 위해서는 사전에 안전한 채널을 통한 키 교환이 이루어져야 한다는 문제점과 키 관리의 문제점을 가진다.

대표적인 비밀키 암호화 알고리즘은 다음과 같다.

1) DES

DES는 평문 64비트를 암호문 64비트로 변환시키는 암호 방식으로 64비트의 키를 사용한다. 이 키는 8비트마다 패리티(parity) 비트 하나씩을 포함하고 있어 DES의 암호화 과정에는 56비트만이 적용된다.

DES 암호 알고리듬은 Feistel 구조로 기본 동작은 전치, 환자와 mod 2 연산으로 구성되어 있다. 다시 전치는 평형 전치, 확대 전치 그리고 축약 전치 등의 세 종류가 있으며 환자는 S box라는 환자 장치에서 이루어진다.[3]

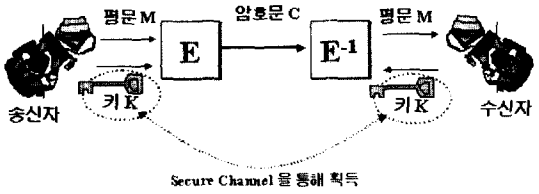
2) 3DES

3DES(triple DES)는 DES의 보안적 효과를 증가시키기 위해서 사용되며, 각 64비트의 키를 사용하여 암호화-복호화-암호화 과정을 수행한다. 따라서 키의 크기는 192비트가 되며, 평문과 암호문의 크기는 64비트이다.

3) SEED

SEED는 정보통신 및 정보 보호의 비밀성 서비스를 제공하기 위하여 한국정보보호진흥원이 중심이 되어 1998년 개발한 128비트 한국표준 암호 알고리즘으로, 1999년 국내 단체표준화(TTA.KO-12.0004, '99. 9)가 완료되었다.

SEED는 평문 128비트를 암호문 128비트로 변환시키는 암호방식으로 128비트의 키를 사용하며, DES와 유사한 Feistel 구조를 가진다.[4]



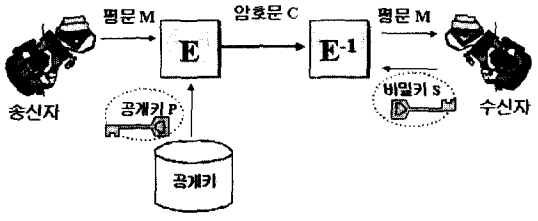
〈그림 2〉 비밀키 암호 알고리즘

2. 공개키 암호 알고리즘

Diffie와 Hellman은 1976년 발표한 논문 "New directions in Cryptography"[1]에서 송신자의 암호화키와 수신자의 복호화키가 서로 다른 암호 시스템인 공개키 암호 알고리즘을 제안했다.

공개키 암호 방식은 어떤 정보를 암호화 할 경우, 공개키를 이용하여 해당 정보를 암호화하고, 암호문을 복호화할 경우, 해당 공개키에 대응하는 개인키로 복호화 함으로써 원하는 평문을 얻을 수 있게 된다.

비대칭키 암호(asymmetric cryptography)라고도 불리는 공개키 암호 방식에서는 비밀키를 사전에 교환해야 하는 기존의 비밀키 암호 방식과 달리 수학적으로 연관된 서로 다른 공개키·개인키 쌍을 생성하여 공개키는 공개하고, 비밀키는 소유자만이 비밀리에 간직한다. 이 방식은 기존의 비밀키 암호 방식이 가지고 있던 키 교환 및 키 관리의 문제점을 해결하고 있으나, 멱승 계산을 수행하는 암·복호화 과정으로 인해 비밀키 암호 시스템에 비해 처리 속도가 현저히 느리다는 문제점을 가지고 있다. 그러므로 공개키 기반 구조에서 공개키 암호 알고리즘은 일반적으로 메



〈그림 3〉 공개키 암호 알고리즘

시지의 암·복화에 사용되지 않고 키 교환을 위해 사용된다.

공개키 암호 알고리즘의 안전성은 합성수의 인수분해 문제나 유한체의 이산대수 문제, 제곱근 문제 등 수학적으로 어려운 문제들을 바탕으로 설계되었다.

대표적인 공개키 암호 알고리즘은 다음과 같다.

1) RSA

RSA는 1978년 MIT의 Rivest, Shamir와 Adleman이 발표한 논문인 "A Method for Obtaining Digital Signatures and Public Key Cryptosystems"[5]에 제안된 암호 시스템으로 매우 큰 정수 $n=p\cdot q$의 소인수 p와 q를 찾는 것이 어렵다는 소인수 분해 문제를 기반으로 기반으로 설계된 암호 방식이다.

2) Diffie-Hellman

Diffie-Hellman은 1976년에 개발된 최초의 공개키 암호 알고리즘이다. 이 방식은 제한된 영역에서 멱의 계산에 비하여 이산대수의 계산이 어렵다는 것을 기반으로 설계되었다. 그러나 Diffie-Hellman이 1976년에 제안한 방식은 메시지의 암/복호화에 사용되지 못하고 키 분배에만 사용되었다.[1]

3) ElGamal

ElGamal 알고리즘은 큰 소수 p로 만들어진 집합 Z_p상에서의 원시 원소를 g라 할 때, $g^x\equiv y \bmod p$의 g와 y값을 알고 있어도 $\log_g y\equiv x$를 구하는 것이 어렵다는 이산대수 문제를 기반으로 설계된 암호 방식이다.[6]

Ⅲ. 해쉬 알고리즘

해쉬 알고리즘은 임의의 입력 비트열에 대하여 고정된 짧은 길이의 출력 비트열을 내는 것으로, 정보보호의 여러 메커니즘에서 활발히 이용되는

〈그림 4〉 해쉬함수

요소 기술이다.

일반적으로 해쉬함수의 조건은 다음과 같다.

- 약한 일방향성(one-wayness, weak) : 주어진 출력에 대하여 입력값을 구하는 것이 계산상 불가능하다.
- 강한 일방향성(one-wayness, strong) : 주어진 입력에 대하여 같은 출력을 내는 또다른 입력을 찾아내는 것이 계산상 불가능하다.
- 충돌 회피성 (collision freeness) : 같은 출력을 내는 임의의 서로 다른 두 입력 메시지를 찾는 것이 계산상 불가능하다.

위와 같은 조건을 만족하는 해쉬함수는 데이터의 무결성, 인증, 부인 방지 등에서 응용되는 중요한 함수 중의 하나로 전자서명에 많이 이용되고 있다. 입력 M에 해쉬함수를 취한 결과인 해쉬코드 h(M)에 송신자는 비밀키로 서명을 하고, 수신자는 이를 공개키로 확인한 후 그 결과 h(M)을 수신된 M에 해쉬함수를 취한 결과의 값과 비교하여 서명의 진위 여부를 밝힌다. 또한, 해쉬함수는 정보의 무결성 검증에도 활용된다. 송신자가 메시지와 함께 그 메시지의 해쉬코드를 전송하면, 수신자는 메시지를 동일한 해쉬함수로 압축한 후 송신자로부터 받은 해쉬값과 비교함으로써 메시지의 무결성을 검증할 수 있다.

대표적인 해쉬함수는 다음과 같다.

1. SHA-1

SHA-1는 미 연방정부의 디지탈 서명 표준인 DSA를 위해 개발된 해쉬함수로 MD5와 유사한 구조로 설계되어 안전성이나 특성이 비슷하다. 160비트 길이의 출력을 내는 SHA-1은 1995년 4월에 정식 표준으로 승인(FIPS PUB 180-1)되었으며 현재, 대부분의 인터넷 응용이나 국제/

업계 표준들에서 디폴트 해쉬함수로 사용되고 있습니다.[7]

2. HAS160

HAS160은 한국형 디지탈 서명 표준인 KCDSA와 함께 사용할 목적으로 개발되었으며, 1998년 10월의 국내 단체표준화 (TTAS.KO-12.0011)를 거쳐 2000년 12월에 개정되었다. HAS160은 메시지를 512비트 블럭단위로 처리하여 160비트의 해쉬코드를 출력하는 Little endian 구조의 32비트 마이크로프로세서를 기본으로 설계된 충돌저항성의 해쉬함수이다.[8]

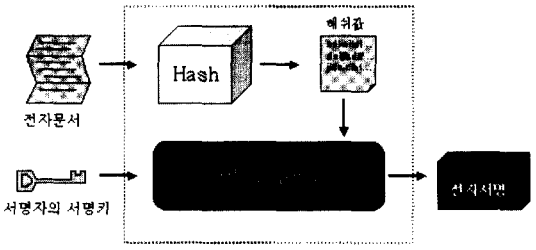
Ⅳ. 전자서명

전자서명은 공개키 암호 방식의 한 응용으로 전자문서에 수기 서명과 같은 서명 효과를 부여하는 전자적 서명 방식이다. 전자문서는 그 특성상 사본과 원본을 구별하는 것이 불가능하고, 위조되기 쉬우므로, 전자서명을 통해 이를 방지할 수 있다.

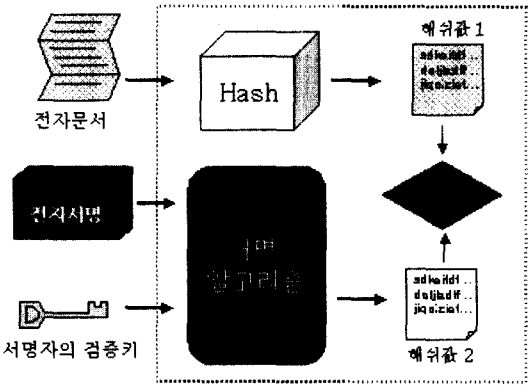
전자서명은 암호의 고유 기능인 정보 보호 기능과 인증 기능 중에 인증 기능을 활용하여 서명에 참여한 사용자 인증과 서명 대상인 전자문서에 대한 인증을 수행하며, 서명의 검증자 또는 제삼자에 의한 서명의 위·변조 및 서명자에 의한 서명문 전송 행위 부인과 같은 형태의 부정 행위를 방지할 수 있다.

전자서명이 위와 같은 기능을 제공하기 위해서는 다음과 같은 조건을 만족시켜야 한다.

- 위조 불가(unforgeable) : 정당한 서명자만이 전자서명을 생성할 수 있다.
- 서명자 인증(user authentication) : 전자서명의 서명자를 누구든지 검증할 수 있다.
- 부인 불가(non-repudiation) : 서명자는 서명한 사실을 부인할 수 없다.
- 변경 불가(unalterable) : 서명한 문서의 내용을 변경할 수 없다.
- 재사용 불가(not reusable) : 해당 문서의 서명을 다른 문서의 서명으로 사용할 수 없다.



〈그림 5〉 서명 생성



〈그림 6〉 서명 검증

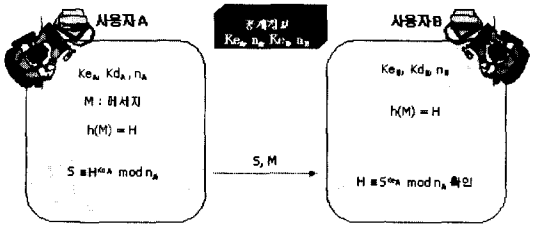
서명의 생성은 〈그림 5〉와 같이 서명자의 개인키(서명키)를 사용하여 서명을 생성한다. 이때 공개키 암호 방식이 가지는 계산상의 비효율성으로 인해 서명되는 데이터는 평문 전체가 아니라 평문에 대한 해쉬값이 된다. .

서명 검증은 〈그림 6〉과 같이 서명자의 공개키(검증키)를 사용하여 서명을 검증한다. 이때 검증자는 전송 받은 데이터에서 서명과 메시지를 얻고, 서명자의 공개키를 이용하여 서명으로부터 해쉬값을 구하고, 메시지를 직접 해쉬한 값과 비교함으로써 메시지에 대한 서명을 검증한다.

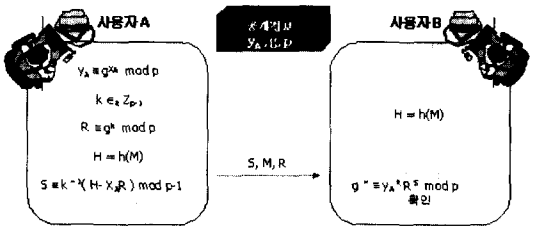
대표적인 전자서명은 다음과 같다.

1. RSA

RSA 공개키 암호 방식은 정보 보호 기능과



〈그림 7〉 RSA 서명



〈그림 8〉 ElGamal 서명

전자 서명기능을 동시에 수행할 수 있는 암호 방식으로 디지틀 서명에 널리 이용되고 있다.

RSA 암호방식을 이용한 디지틀 서명 방식은 〈그림 7〉과 같다.

서명자 A는 충분히 큰 소수 p_A, q_A를 선택하여 $n_A=p_A \cdot q_A$를 계산하고, $\phi(n_A)=(p_A-1)(q_A-1)$과 서로 소인 Ke_A를 선택하여 $Ke_A \cdot Kd_A \equiv 1 \bmod \phi(n_A)$를 만족하는 Kd_A를 구한 다음 Ke_A와 n_A를 공개하고 Kd_A는 비밀리에 보관한다. 이때 Ke_A, n_A는 서명 검증을 위한 검증키(공개키)이고, Kd_A는 서명키(비밀키)이다.

서명자 A는 공개 일방향 해쉬함수(hash function)로 서명문 M을 서명할 수 있는 크기로 압축 H=h(M)한 후, 압축된 H에 대한 서명 $S \equiv H^{Kd_A} \bmod n_A$를 계산하여 서명문 M과 함께 검증자 B에게 전송한다.

검증자 B는 서명자 A의 서명문 M과 서명 S를 수신한 다음 공개된 A의 서명 검증 정보 Ke_A, n_A로 서명을 검증한다. 수신한 서명문 M의 해쉬값 H′=h(M)을 계산하고 수신한 서명 S로부터 $H \equiv S^{Ke_A} \bmod n_A$를 계산한 다음 H와 H′를 비교하여 서명문 M과 서명 S을 검증할 수 있다[1].

RSA 서명의 안전성은 RSA 암호방식과 마찬가지로 합성수의 소인수 분해의 어려움에 기반하여 설계되었다.

2. ElGamal

ElGamal 디지틀 서명은 1985년 발표된 디지틀 서명으로 그 안전성은 이산 대수 문제를 기반으로 하고 있다.

ElGamal 디지틀 서명 방식은 〈그림 8〉과 같다.

서명자는 큰 소수 p(>512비트)를 선택하여 Z_p상에서 원시원소 g를 찾는다. 서명키 X_A를 임의로 정하고 검증키 $y_A \equiv g^{X_A} \bmod p$를 계산하여 p, g, y_A를 공개한다.

서명자 A는 $k \in_R Z_{p-1}$를 선택하여 중간값 $R \equiv g^k \bmod p$를 계산한다. 서명문 M의 서명 $S \equiv (M-X_AR)k^{-1} \bmod p-1$를 계산하여 검증자 B에게 S, M, R를 전송한다.

검증자는 S, M, R를 수신한 다음 공개 정보인 서명자의 y_A로 $y_A^R R^S \equiv g^M \bmod p$의 성립 여부를 조사함으로써 서명을 검증한다.

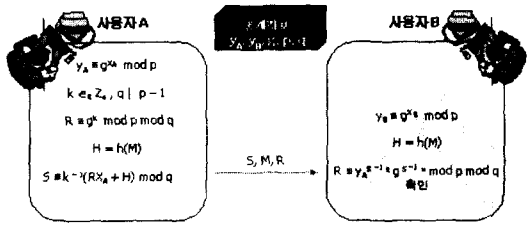
3. DSA

DSA 전자서명은 NIST(National Institution of Standard and Technology)가 DSS(Digital Signature Standard)에서 사용하기 위하여 발표한 정부용 전자서명 알고리즘으로 그 안전성은 이산대수 문제의 어려움에 기반을 두고 있다[9].

DSA는 ElGamal 서명과 유사한데, ElGamal에서 사용하고 있는 Z_p상에서의 원시 원소를 사용하는 것이 아니라 $q|p-1$인 소수 q를 위수로 갖는 원소 g를 선택하고 있다. 즉, $g^q \equiv 1 \bmod p$를 만족하는 원소 g를 선택하며 q가 160비트 크기면 충분히 안전성이 보장된다고 설명하고 있다.

DSA 서명 방식은 〈그림 9〉와 같다.

서명자 A는 서명키 X_A를 선택하고 검증키 $y_A \equiv g^{X_A} \bmod p$를 계산하여 공개키는 공개한다. 서명자는 Z_q상에서 임의의 k를 선정해서 중간값 $R \equiv g^k \bmod p \bmod q$를 계산한다. 서명문 M을 해쉬(H=h(M))한 다음, 서명키로 서명 $S \equiv k^{-1}(RX_A+H) \bmod q$를 계산하여 서명문 M, 중간



〈그림 9〉 DSA 서명

값 R과 함께 검증자 B에게 전송한다. 검증자는 S, M, R을 수신한 다음 S의 역수 S^{-1}와 검증키 y_A로 $R \equiv y_A^{S^{-1}R} g^{S^{-1}H} \bmod p \bmod q$의 성립여부를 조사함으로써 서명을 검증한다.

4. KCDSA

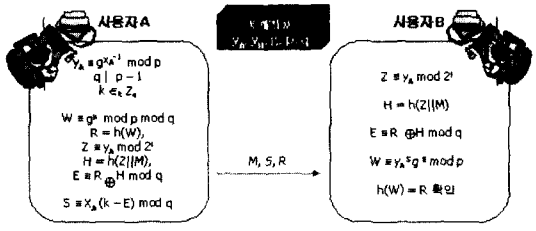
KCDSA는 이산대수 문제의 어려움에 기반을 둔 전자서명 알고리즘으로서, 한국통신정보보호진흥원의 주관으로 우리나라의 주요 암호학자들이 주축이 되어 개발된 국내표준 전자서명 알고리즘으로서 전자서명법 체계에서 사용되고 있다[10].

KCDSA 디지틀 서명 방식도 ElGamal 디지틀 서명 방식을 개선한 방식으로 이산 대수 문제를 기반한다.

KCDSA 서명 방식은 〈그림 10〉과 같다.

서명자 A는 서명키 X_A를 선택해서 검증키 $y_A \equiv g^{X_A^{-1}} \bmod p$를 계산하여 소수 p, $q|p-1$인 소수 q, Z_p상에서 위수가 q인 원소 g를 공개한다.

서명자는 Z_q상에서 임의의 원소 k를 선택하여 $W \equiv g^k \bmod p$를 계산하고, 이를 해쉬하여 서명의 첫 부분 $R=h(W)$를 생성한다. 이후, $Z=y_A \bmod 2^l$을 계산하여 이를 서명문과 함께 해쉬한 값 $H=h(Z\|M)$과 서명의 중간값 $E=(R \oplus H) \bmod q$으로 서명의 두 번째 값 $S \equiv X_A(k-E) \bmod q$를 생성하고, 이를 M, R와 함께 검증자에게 전송한다.



〈그림 10〉 KCDSA 서명

M, R, S를 수신한 검증자는 $Z=y_A \bmod 2^l$를 계산하여 검증할 메시지의 M에 대한 해쉬값 $H=h(Z\|M)$을 계산하고, 이를 이용해서 중간값 $E=(R \oplus H) \bmod q$를 계산한다. 검증자는 서명자의 검증키 y_A를 이용하여 증거값 $W \equiv g^k \bmod p$를 계산하고, $h(W)=R$이 성립하는지 확인함으로써 서명을 검증한다.

V. 결 론

인터넷을 통한 전자상거래가 점차 일반화되고, 그 규모가 커짐으로 인해 안전하고 신뢰성 있는 전자상거래를 수행하기 위한 기본 인프라로 공개키 기반구조의 중요성은 점차 확대되어 가고 있다. 이미 대부분의 선진국들은 자국내의 공개키 기반구조를 구축하고 이를 통한 전자 상거래 및 중요 문서 교환이 이루어지고 있고, 국내에서도 이에 대한 연구와 구축이 활발히 진행되고 있다. 이러한 상황에서 공개키 기반구조를 구성하는 핵심 보안 기술로 전자서명에 대한 연구는 공개키 기반구조 환경 구축을 위한 기술적 토대 마련에 중요한 역할을 할 것이다.

참 고 문 헌

〔1〕 W. Diffie and M. E.Hellman, New directions in cryptography, IEEE Transaction on Information Theory IT-22 No.6, pp.644-654, 1976

〔2〕 IETF RFC2459, Internet X.509 Public Key Infrastructure Certificate and CRL Proflie, 1998

〔3〕 NIST, Data Encryption Standard, Federal Information Processing Standards Publication 46, 1977〔4〕 한국정보

보호진흥원, 128비트 블록 암호알고리즘 (SEED) 개발 및 분석 보고서, 1998
〔5〕 R. L. Rivest, A. Shamir and L. Adleman, A method of obtaining digital signature and public key cryptosystem, ACM Communication 21 No.2, pp.120-126, 1978
〔6〕 T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithm, IEEE Trans. on Information Theory IT-31, pp.469-472, 1995
〔7〕 NIST, Secure hash standard, FIPS 180-1, 1995
〔8〕 해쉬함수표준-제2부 : 해쉬함수알고리즘표준(HAS-160), TTAS.KO-12.0011/R1
〔9〕 "A Proposed Digital Signature Standard(DSS)", NIST, August 1990
〔10〕 부가형 전자서명 방식 표준-제2부 : 인증서 기반 전자서명 알고리즘, TTAS.KO-12.0001/R1

저 자 소 개

李 香 珍

2000년 성균관대학교 전기 전자 및 컴퓨터 공학부 학사, 2002년 성균관대학교 전기 전자 및 컴퓨터 공학부 석사, 현재 한국정보보호진흥원 평가인증사업단 전자서명인증관리센터 근무.

李 弘 燮

1979년 한양대학교 전자공학 공학사, 1985년 한양대학교 전자공학 공학석사, 1999년 대전대학교 컴퓨터공학 공학박사, 1980~1996년 : 한국전자통신연구원 책임연구원, 실장, 1996~현재 : 한국정보보호진흥원 연구개발부장, 기술본부장, 현 평가인증사업단 단장, 1997~2001년 : 한국정보통신기술협회 정보보호기술위원회 의장, 1999년 국가최상위인증기관 전자서명인증관리센터구축준비반장, 2000~현재 : 인터넷보안기술포럼 의장, 2001~현재 : ASIA PKI Forum Interoperability WG 공동의장, 2002~현재 : 순천향대학교 겸임교수, 2002~현재 : 한국정보보호학회 부회장.