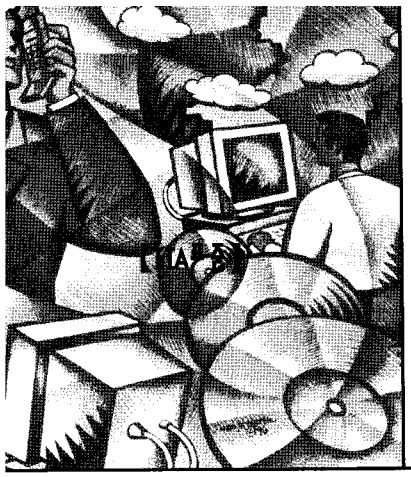


부가형 전자서명 알고리즘 표준 소개 (TTA.KO-12.0001)

김병천, 이홍섭
한국정보보호센터 기술개발부

1. 전자서명의 개요

정보사회에서 대부분의 정보는 통신망을 통하여 교환된다. 이에 따라 메시지의 수신자는 받은 메시지가 전송 도중 내용이 바뀌지 않았는지, 또는 제 3자가 송신자를 가장하여 메시지를 보내지 않았는지 등을 확인할 수 있기를 바란다. 이런 목적으로 사용되는 가장 효율적인 도구가 전자서명이다.

전자서명은 지금까지 널리 일반화되어 종이 문서에 표기되던 수기 서명이나 인장의 효과를 전자적 매체 내에 저장 또는 전송되는 전자 문서에 효과적으로 서명 효과를 부여하는 전자적 서명 방식을 말한다. 특히 전자 문서는 사본과 원본을 구별하는 것이 불가능하므로 전자문서상의 서명은 정보사회에서의 정보통신망을 이용한 전자 문서 교환에 필수적인 요소로 대두되고 있다.

종이문서에 사용되는 수기 서명이나 인장을 이용한 서명은 디지탈 신호로 변환하였을 때 위조가 간단하여 서명으로써의 효과를 기대할 수 없다. 암호를 이용하면 전자서명은 간단하게 실현할 수 있다. 암호의 고유 기능인 정보보호 기능과 인증 기능 중에 인증 기능을 활용하면 서명에 참여한 사용자 인증과 서명 대상인 전자 문서의 인증을 동시에 실행함으로써 전자서명 효과를 얻을 수 있다. 메시지에 대한 전자서명은 누구나 그 메시지의 출처를 확인할 수 있게 해 주는 인증서비스, 획득한 메시지가 도중에 변경되었는지의 여부를 확인할 수 있게 해 주는 무결성 서비스 및 메시지에 서명을 했다는 사실을 부인할 수 없게 하는 부인방지 서비스 등 정보사회에서 필요한 다양한 보안 서비스를 제공하는데 필수적인 도구이다.

선진 각국에서는 정보화를 통한 국가경쟁력 제고의 방법으로 인터넷을 통한 전자상거래 활성화를 위하여 노력을 경주하고 있는데 전자상거래의 안전하고 신뢰성 있는 서비스를 위한 핵심기술이 전자서명기술이다. 따라서 전자서명의 법/제도적 환경인 전자서명법을 제정하고, 전자서명기술의 글로벌 사용환경 제공을 위한 PKI(Public Key Infrastructure) 구축에 힘쓰고 있다.

우리 나라에서도 전자서명기술을 지식정보사회의 중요 수단으로 인식하여 전자서명법을 입법추진하고 있으며, 본고에서 소개 할 전자서명 알고리즘을 정보통신단체표준으로 제정한 바 있다.

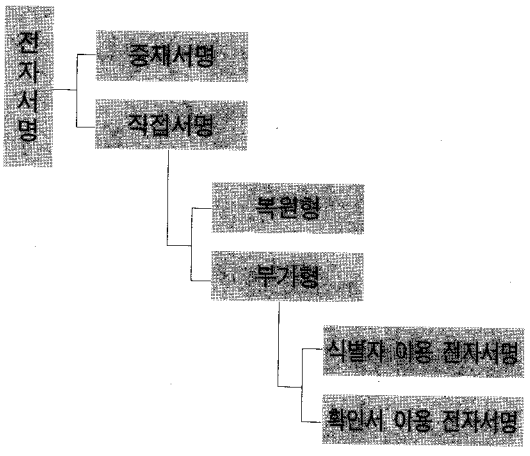
본고에서는 국내·외 전자서명기술관련 표

준화 현황을 살펴보고, '98년 10월에 정보통신단체표준으로 제정된 부가형 전자서명 알고리즘 표준을 소개한다.

2. 전자서명 방식의 분류 및 특성

2.1 전자서명 방식의 분류

전자서명은 서명방법에 따라 대칭키 암호알고리즘과 중재자를 사용하는 중재서명과 공개키 암호알고리즘을 이용하는 직접서명으로 구분할 수 있고, 직접서명은 서명의 형태에 따라 서명 확인과정에서 서명으로 부터 메시지가 복구되는 형태로 복원형(Digital Signature giving Message Recovery)과 서명 확인과정에서 메시지와 메시지의 서명이 입력의 일부분이 되는 방식인 부가형(Digital Signature with Appendix)으로 구분할 수 있으며, 부가형 서명방식은 다시 공개키(서명 검증키)를 인증하는 방식에 따라 식별자 이용 전자서명 방식(Identity-based Digital Signature)과 확인서이용 전자서명방식(Certificate-based Digital Signature)으로 구분할 수 있다.



[그림 1] 전자서명방식의 분류

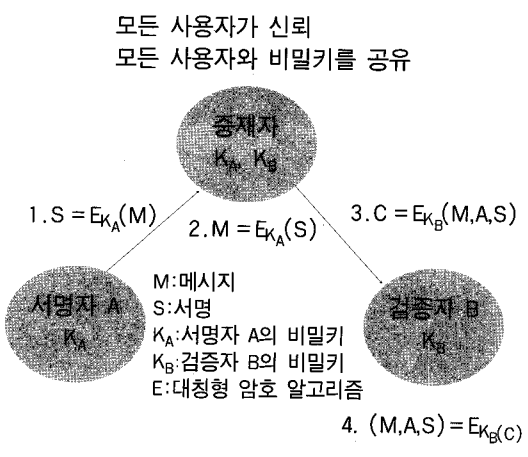
가. 중재 서명

대칭키 암호 방식을 이용하여 전자서명을 구성할 때 검증자의 서명 진위 확인과 서명자의 서명 부인을 막기 위해서는 대칭키 암호 특성상 중재자를 필요로 한다. 이러한 서명 방식을 중재 서명 방식이라 하며 중재자가 전자서명 과정에 개입하여 서명자의 서명을 검증자에게 확인시켜주며, 서명자와 검증자의 부정 행위를 방지한다.

경우에 따라서는 서명자가 비밀 서명 정보의 노출로 자신이 작성한 서명이 아니라고 부인하는 경우의 분쟁은 서명의 정당성을 확인하기가 곤란하다. 이러한 경우 중재자가 사전에 서명절차에 관여한 경우 부인을 봉쇄할 수 있다. 물론 중재 서명 방식에서의 중재자는 믿을 만한 제삼자인 것을 전제로 한다.

나. 복원형 전자서명 VS 부가형 전자서명

직접서명은 복원형 전자서명과 부가형 전자서명으로 구분할 수 있다.



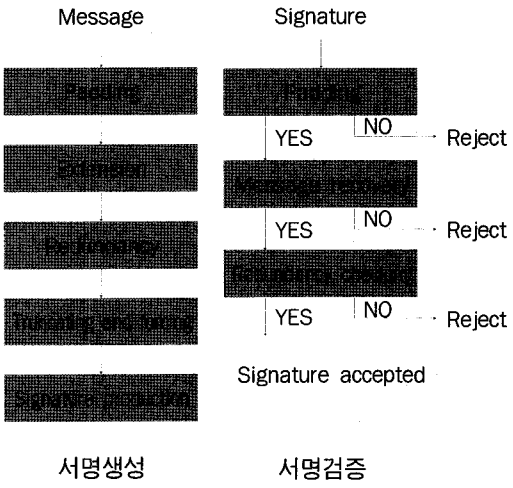
[그림 2] 중재 서명

1) 복원형 전자서명

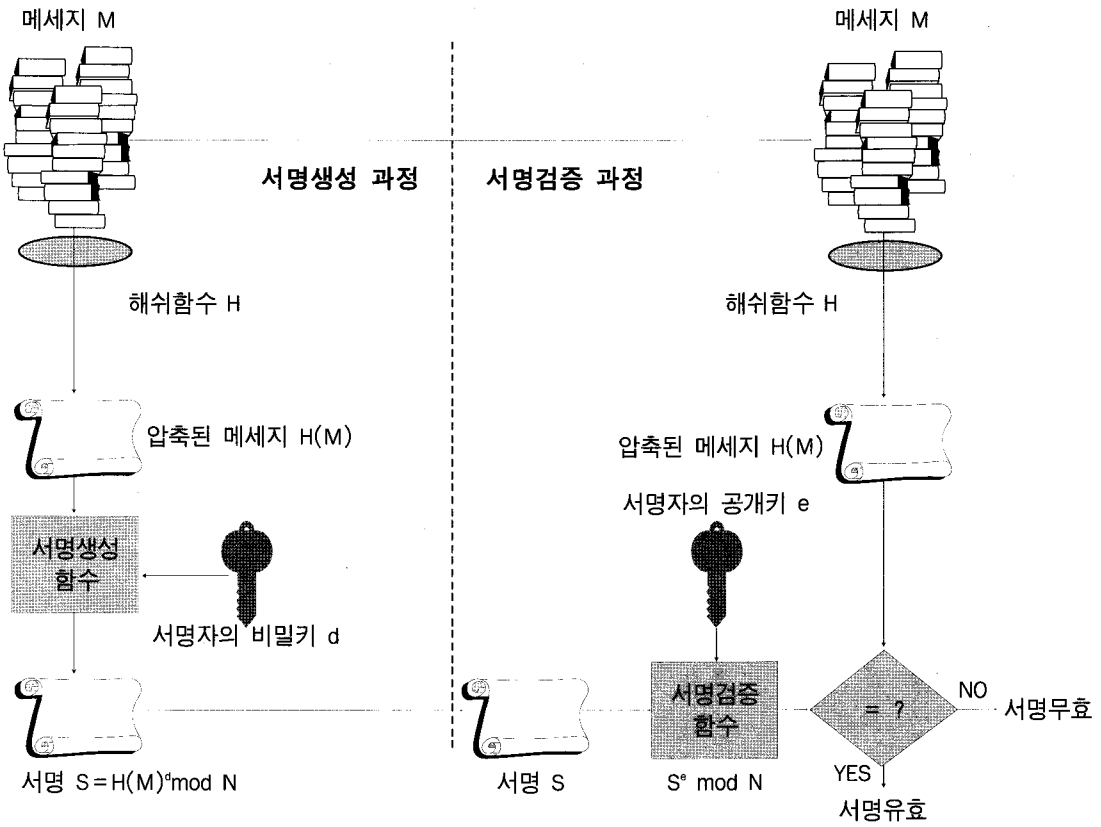
서명의 검증과정에서 원래의 메시지가 복원되는 형태의 서명을 말한다. 용장 생성함수(redundancy generating function)를 이용하여 서명할 제한된 길이의 메시지를 일정한 규칙에 따라 변환시켜 서명을 생성하고, 복원된 메시지가 용장 생성함수의 규칙에 따라 생성된 것인지를 확인함으로써 서명을 검증한다.

2) 부가형 전자서명

임의의 길이의 메시지에 서명부분을 부가하는 방식으로서 서명생성 및 검증 과정에서 메시지가 입력의 일부분이 되는 것을 부가형 전자서명이라 하며, 생성 및 검증 과정에서 동일한 해쉬함수를 사용한다.



[그림 3] 복원형 전자서명 생성 및 검증 과정



[그림 4] 부가형 전자서명 생성 및 검증 과정

다. 식별자 이용 전자서명 VS 확인서 이용 전자서명

부가형 전자서명은 공개 검증키의 관리방법에 따라 식별자를 이용한 서명과 공개키 확인서를 이용한 서명으로 나눌 수 있다.

서명 검증 과정에서 서명자의 올바른 공개 검증키를 얻어야 하는데, 일반적으로 CA (Certification Authority)라 부르는 모든 사람이 인정하는 제 3자가 각 개인의 ID와 그의 공개키 등을 결합한 공개 검증키 정보를 CA의 비밀키로 서명한 확인서(인증서)를 배포함으로써 공개 검증키의 소유자를 보증한다. 이와 같이 CA가 발급하는 공개 검증키에 대한 확인서를 통해 각 사용자와 이 사용자의 공개 검증키를 결합시킴으로써, 사용자가 해당 비공개 서명키를 이용해 생성한 전자서명을 그 사용자의 공개 검증키를 이용하여 전자서명을 검증하는 방식이 확인서 이용 전자서명 방식이다.

이에 반해 식별자 이용 전자서명은 개인식별 정보 ID가 공개 검증키이거나 이로부터 공개 검증키를 유추해낼 수 있는 전자서명 방식으로 공개 검증키를 보증해 주는 신뢰기관은 필요치 않으나 개인식별 정보 ID와 쌍을 이루는 서명 생성키(비밀키)를 발급해주는 고신뢰 센터가 필요하다.

식별자 이용 방식의 대표적인 알고리즘은 F-S(Fiat-Shamir)방식과 G-Q방식이 있고, 확인서 이용 전자서명 알고리즘은 RSA, DSS, Shnorr, ESIGN 등과 본고에서 소개할 KCDSA가 있다.

2.2 전자서명의 특성

가장 널리 사용되는 전자서명방식은 공개키 암호시스템을 사용하는 직접서명으로 특히 부가형 전자서명방식이 주로 이용된다. 전자서명의 안전성을 확보하기 위하여 검증에 사용되는 공개키로부터 서명생성에 사용되는 비공개키가 계산되는 것이 실행 불가능해야 하며, 서명은 메시지 내용, 서명자 비공개키와 사용자 정보에 의존되어야 한다. 즉 전자서명이 유효하기 위해서는 다음의 조건을 만족해야 한다.

- ㅇ위조 불가(unforgeable) 조건으로 합법적인 서명자만이 전자서명을 생성할 수 있어야 한다.
- ㅇ서명자 인증(user authentication) 조건으로 전자서명의 서명자를 누구든지 검증할 수 있어야 한다.
- ㅇ부인 불가(nonrepudiation) 조건으로 서명자는 후에 서명한 사실을 부인할 수 없어야 한다.
- ㅇ변경 불가(unalterable) 조건으로 서명한 문서의 내용을 변경할 수 없어야 한다.
- ㅇ재사용 불가(not reusable) 조건으로 문서의 서명을 다른 문서의 서명으로 사용할 수 없어야 한다.

또한 전자서명방식을 실제 응용에 적용하기 위해서는 전자서명에 사용되는 비밀키와 공개키의 생성이 간단해야 하며, 서명과정과 검증과정에 필요한 계산이 효율적이어야 한다.

이러한 조건을 만족하는 전자서명 방식에서 사용하는 일방향 함수는 수학적으로 어려운 문제로 알려진 다음의 3가지 문제를 가장 많이 사용한다.

- ㅇ소인수 분해 문제(Factorization Problem) : 합성수 N 의 소인수를 찾는 문제로 N

의 자릿수가 매우 큰 경우(10^{150} 이상)에는 N의 소인수를 효율적으로 찾는 알고리즘이 존재하지 않는다.

- 제곱근 문제(Square Root Problem) : 이차식 $X^2 \equiv a \bmod N$가 해를 가질 때 제곱근 X를 구하는 문제로 합성수 N의 소인수 분해 문제와 동치이다.
- 이산대수 문제(Discrete Logarithm Problem) : 소수 P가 주어지고, $Y \equiv g^x \bmod P$인 경우, 역으로 $x \equiv \log_g^Y \bmod P$인 x를 계산하는 문제. 여기서 x를 mod P상의 Y의 이산대수라 한다. 소수 P가 매우 크고(2^{512} 이상), g의 위수가 2^{140} 이상인 경우 x를 계산하는 효율적인 알고리즘이 존재하지 않는다.

3. 전자서명 관련 표준화 동향

전자서명 관련 국제 표준 활동은 ISO/IEC JTC1 WG2에서 메시지 복원형 전자서명(ISO/IEC 9796)과 부가형 전자서명(ISO/IEC 14888), 그리고 타원곡선을 이용한 전자서명(ISO/IEC 15946-2)에 대한 표준화 작업이 있다. 세부 표준화 활동은 표 1과 같다.

ISO/IEC #	상태	제목
9796-1	FCD	Digital Signature Scheme giving message recovery -Part 1 : Mechanisms using redundancy
9796-2	IS	Digital Signature Scheme giving message recovery -Part 2 : Mechanisms using a hash-function
9796-3	CD	Digital Signature Scheme giving message recovery -Part 3 : Mechanisms using a check-function
9796-4	FCD	Digital Signature Scheme giving message recovery -Part 4 : Discrete log based mechanisms
14888-1	FDIS	Digital Signatures with appendix -Part 1 : General
14888-2	FDIS	Digital Signatures with appendix -Part 2 : Identity-based mechanisms
14888-3	FDIS	Digital Signatures with appendix -Part 3 : Certificate-based mechanisms
15946-2	WD	Cryptographic techniques based on elliptic curves -Part 2 :Digital Signatures

[표 1] ISO/IEC 전자서명 관련 표준

ISO/IEC 9796은 1991년에 제정하여 1996년에 검토하였고, 2000년에 재검토 할 계획이었으나 ISO/IEC 9796에 부 파트가 추가되면서 수정작업을 진행중 이다. 특히 ISO/IEC CD 9796-3은 1997년 CRYPTO '97에서 "A Multiplicative Attack Using LLL Algorithm on RSA Signatures with Redundancy" 논문에서 취약점이 발견되어 삭제하기로 결정하였

고, ISO/IEC 9796-4가 ISO/IEC 9796-3을 대체할 것으로 예측된다.

ISO/IEC WD 15946-2에서는 타원곡선 암호 시스템을 이용한 전자서명의 표준화 작업을 추진중인데 기존 전자서명방식을 타원곡선 기반으로 변환한 전자서명방식을 예시로 소개하고 있다. ElGamal, DSA을 변환한 EC-ElGamal, EC-DSA과 국내에서 표준으로 채택된 KCDSA를 변환한 EC-KCDSA를 포함하고 있다.

특히 EC-KCDSA에 관한 채택은 '98년 10월 브라질에 개최된 제17차 ISO/IEC JTC1 SC27 회의에서 한국 대표단 활동에 의해 반영되었다.

미국의 경우 1994년에 연방표준으로 DSS (Digital Signature Standard)와 이와 함께 사용할 해쉬함수 SHS(Secure Hash Standard)를 제정하였고, ANSI 은행 정보보호 표준에 RSA, DSA, EC-DSA를 표준전자서명방식으로 규정하고 있다. 또한 FPKI(Federal Public Key Infrastructure)에서도 표준전자서명알고리즘으로 DSA와 EC-DSA를 채택하고 있다. 전자서명관련 미국 표준은 표 2와 같다.

표준 번호	제목
FIPS 180-1	Secure hash standard(SHA-1)
FIPS 186	Digital signature standard(DSA)
ANSI X9.30-1	DSA
ANSI X9.30-2	SHA
ANSI X9.31-1	RSA
ANSI X9.31-2	Hash algorithms for RSA
ANSI X9.62	EC-DSA

[표 2] 미국의 전자서명관련 표준

국내의 전자서명 관련 표준은 ISO/IEC 9796(1991)을 수용한 KS X 1207과 전자서명, 해쉬함수의 기본구조를 각각 규정하는 KICS.KO-090003, KICS.KO-10.0076이 있고, 세부내용은 표 3과 같다.

표준 번호	표준명	비고
KS X 1207	메시지 복원형 전자서명 방식	ISO/IEC 9796(1991)
KICS.KO-09.0003	부가형 전자서명방식 표준/제1부 기본구조 및 모델	1996(구 KCS 221)
TTA.KO-12.0001	부가형 전자서명방식 표준/제2부 확인서 이용 전자서명 알고리즘	1998. 10.
KICS.KO-10.0076	해쉬함수 표준/기본구조	1996
TTA.IS-10118	해쉬알고리즘 표준	1998. 10.

[표 3] 국내 전자서명 관련 표준

본고에서 소개할 부가형 전자서명 방식 표준/제2부 확인서 이용 전자서명 알고리즘(KCDSA:Korea Certificate-based Digital Signature Algorithm)과 더불어 KCDSA와 함께 사용할 수 있는 160비트 출력값을 갖는 해쉬알고리즘(HAS-160 : Hash Algorithm Standard-160)와 함께 '98년 10월 TTA 정보통신단체표준으로 제정되었다.

4. 부가형 전자서명방식 표준/제2부 확인서 이용 전자서명 알고리즘

이 표준은 확인서를 이용한 부가형 전자서명의 생성 및 검증 알고리즘뿐만 아니라 제안되는 알고리즘에 대한 수학적 검증, 안전도 분석 결과를 포함하고 있으며, 제안 알고리즘을 실제 적용하고자 할 때 요구되는 소수 및 난수 생성과 같은 세부 사항들을 이 표준의 부기로 제공하고 있다. 이 밖에 제안되는 알고리즘을 시범 구현한 결과 값을 명시함으로써 향후 표준에 따른 구현제품의 적합한 구현여부를 확인할 수 있도록 기준을 제시하고 있다.

4.1 기호와 표기

영대문자로 표시된 기호는 음이 아닌 정수값, 혹은 그에 상응하는 비트 열이다.

가. 시스템 변수 및 함수

- P : $2^{|P|-1} < P < 2^{|P|}$, $|P| = 512+256i(0 \le i \le 6)$의 크기를 가지며, (P-1)/2Q 역시 소수이거나 최소한 Q보다 큰 소수들의 곱으로 구성되는 정수.
- Q : P - 1을 나누는 소수로 $2^{|Q|-1} < Q < 2^{|Q|}$, $|Q| = 128+32j(0 \le j \le 4)$의 크기를 가짐.
- G : $a(P-1)/Q \bmod P$, $1 < a < P - 1$이고, $a(P-1)/Q \bmod P > 1$을 만족함.
- h(): |Q| 비트 길이의 출력값을 갖는 충돌저항성의 해쉬함수.

소수 P, Q 그리고 기본원소 G는 적용 대상 영역 안의 모든 사용자들이 공용으로 사용할 수도 있는 시스템 변수이고, 그 크기는 적용 목적에 따라 다르게 정할 수 있다.

충돌저항성 해쉬함수인 h()는 국내표준 해쉬알고리즘표준(HAS-160)이 존재하므로 이를 사용할 것을 권고한다.

나. 사용자 변수

X : $0 < X < Q$인 비공개 서명키.

Y : $G^{X^{-1}} \bmod P$로 계산되는 공개 검증키 (X^{-1}는 mod Q상에서 X의 곱셈 역원).

Cert_Data : 서명자의 식별번호 또는 신원정보, 시스템 변수 P, Q, G와 공개 검증키 Y 등을 포함하는 공개키 확인서의 생성에 이용되는 사용자의 인증 데이터.

Z : Cert_Data의 해쉬코드. 즉, Z=h(Cert_Data).

C : 신뢰할 수 있는 제3자(즉, CA)가 서명한 서명자의 공개키 확인서.

사용자의 비공개 서명키 X는 그 사용자만이 비밀리에 간직해야 하며, 공개 검증키 Y는 가능한 모든 다른 사용자들에게 알려지도록 해야 한다.

다. 메시지 변수

M : 서명 될 메시지.

H : 메시지의 해쉬코드. 즉, H = h(Z||M).

K : $0 < K < Q$인 일회용 난수값.

W : $W = G^K \bmod P$로 계산되는 K에 대한 증거값.

R : 서명의 첫 부분으로 R = h(W).

E : $E = R \oplus H \bmod Q$로 계산되는 서명과정 중의 중간값.

S : 서명의 두 번째 부분으로, $S = X(K - E) \bmod Q$.

Σ : 서명값. Σ = {R||S}.

난수값 K는 각 메시지에 대한 서명 생성 시마다 새롭게 선택되며, 비공개 서명키 X와 마찬가지로 서명자 외의 제 3자에게 알려지지 않아야 한다.

라. 기타 표기

A′ : 수신되었거나 다시 계산된 A의 값
|A| : 임의의 값 A의 비트 길이.
|| : 연접 (concatenation).
a mod P : a를 P로 나눈 나머지

이 표준에 규정되어 있는 확인서 이용 전자서명 알고리즘을 실제 적용하고자 할 때 이 표준의 부기에 참조사항으로써 권고되어 있는 방법을 사용할 수도 있다. 특히, |P|와 |Q|의 선택은 서명 알고리즘의 안전도 및 성능과 밀접한 관련이 있으므로 주의 깊게 선택하여야 한다. P, Q, X, K, G를 만드는 방법은 부기에 권고되어 있는 방법을 사용할 수도 있다.

4.2 서명 과정

서명 과정에서는 메시지 M을 받아 다음 단계를 거쳐 서명된 메시지 {M||Σ}을 출력하며, 서명Σ는 {R||S}이다.

단계 1. 난수값 K를 {1〈K〈Q - 1}에서 랜덤하게 선택한다.
단계 2. 증거값 W = G^K mod P를 계산한다.
단계 3. 서명의 첫 부분 R = h(W)를 계산한다.
단계 4. 메시지의 해쉬코드 H = h(Z||M)을 계산한다.
단계 5. 중간값 E = R⊕HmodQ를 계산한다.
단계 6. 서명의 두 번째 부분 S = X(K - E)modQ를 계산한다.
단계 7. 서명 Σ={R||S}를 만들어 서명된 메시지 {M||Σ}를 출력한다.

단계 1부터 단계 3까지의 과정은 서명할 메시지와 관계가 없으므로 사전에 계산해 둘 수도 있다. 즉 K와 R을 미리 계산해서 보관하고 있다가 서명할 메시지가 들어오면 단계 4부터 실시간 계산을 할 수 있다. 이런 사전 계산 방식은 다수의 서명을 실시간으로 계산할 필요가 있을 때 유용하게 사용될 수 있으며, 이를 위해서는 사전 계산된 {K, R}을 저장할 여분의 메모리를 갖추어야 한다. 이 표준의 부기에서 사전 계산을 이용하여 서명하는 알고리즘의 예가 기술되어 있다.

단계 3과 6에서 R또는 S가 0인지를 검사할 수도 있다. 이 경우 만일 R = 0이거나 S = 0이면 단계 1로 되돌아가야 한다. 그러나 R이나 S가 0이 될 확률은 R이나 S가 다른 특정 값을 가질 확률과 동일하며, 또한 R이나 S가 0이 된다고 하더라도 서명을 위조하거나 비공개 서명키 X를 얻는데 도움이 되지는 않는다.

4.3 검증 과정

먼저 서명을 검증하기 전에 검증자는 서명자의 공개 검증키 Y와 시스템 변수 P, Q, G 등에 대한 확인된 값들을 얻어야 한다. 이들은 통상 공개키 확인서 C로 제공되며 공개키 확인서를 얻으면 검증자는 우선 Z = h(Cert_Data)를 계산하여 CA의 서명을 확인

한 다음, Cert_Data로 부터 서명 검증 과정에서 사용할 시스템 변수 P, Q, G와 공개 검증키 Y를 추출한다.

단계 1. 서명된 메시지 $\{M' \| \Sigma'\}$로 부터 검증할 메시지 M′서명의 첫 부분 R′, 서명의 두 번째 부분 S′를 추출한다. 이 때 $0<R'<2|Q|$이고 $0<S'<Q$임을 확인한다.

단계 2. 검증할 메시지의 해쉬코드 값 $H' = h(Z\|M')$을 계산한다.

단계 3. 중간값 $E' = R' \oplus H' \bmod Q$를 계산한다.

단계 4. 서명자의 공개 검증키 Y를 이용하여 증거값 $W' = Y^{S'}G^{E'} \bmod P$를 계산한다.

단계 5. $h(W') = R'$이 성립하는지 확인한다.

단계 1부터 단계 5까지의 확인 과정이 모두 통과되면 서명은 받은 메시지 M에 대하여 공개 검증키 Y에 대응하는 비공개 서명키 X로 서명하였음이 확인된 것이다. 위 검증 단계에서 하나라도 그 검증이 실패하면 메시지 M에 불법적인 방법으로 서명이 되었거나 메시지가 변경된 것이므로 다음 단계로 넘어갈 필요 없이 M에 대한 서명이 거짓임이 밝혀진 것이다.

올바른 서명과 메시지가 수신되었을 때, 즉 $M' = M$, $R' = R$, $S' = S$일 때, $h(W') = R'$이 성립하는 것은 다음과 같이 보일 수 있다.

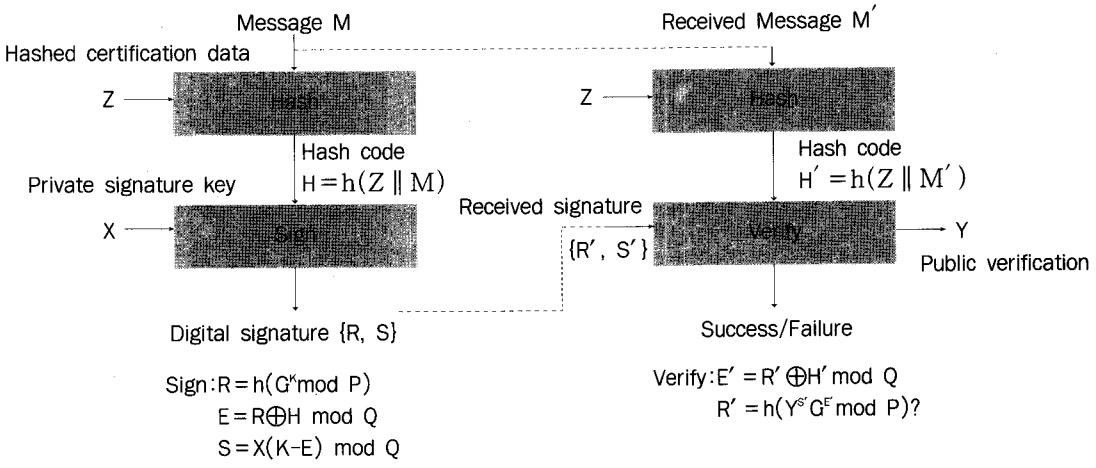
$H' = h(Z\|M') = h(Z\|M) = H$, $E' = R' \oplus H' \bmod Q = R \oplus H \bmod Q = E$ 이고,

$S' = S = X(K-E) \bmod Q$ 이므로

$$\begin{aligned} W &= Y^{S'}G^{E'} \bmod P \\ &= Y^{S}G^{E} \bmod P \\ &= Y^{X(K-E) \bmod Q}G^{E} \bmod P \\ &= G^{X^{-1}X(K-E) \bmod Q}G^{E} \bmod P \\ &= G^{(K-E) \bmod Q}G^{E} \bmod P \\ &= G^{K} \bmod P \end{aligned}$$

이다. 따라서, $h(W') = h(G^K \bmod P) = R$이고, $R' = R$이므로 $h(W') = R'$이다.

[그림 5]에 이상에서 기술된 서명의 생성 및 검증 과정을 간략히 도시하였다. 그리고 이 표준에 따라 구현된 참조구현의 각 서명 생성 및 검증 단계에서의 수치 예를 부기에 기술하고 있다.



[그림 5] KCDSA 서명 생성 및 검증 과정

4.4 안전성 분석

KCDSA는 유한체 GF(P)={0, 1, 2, ..., P-1} 위에서 정의된 알고리즘으로, KCDSA의 안전도는 유한체에서 이산대수 문제를 풀기 어렵다는 사실에 기초하고 있다. 이산대수 문제란 GF(P)의 원소 G(G^Q mod P = 1, G≠1을 만족)로 생성되는 부분 곱셈군 〈G〉의 주어진 원소 Y에 대하여 Y = G^X mod P를 만족하는 X를 찾는 문제이며, 이러한 이산대수 문제가 쉽게 풀리면 이 전자서명 알고리즘은 전혀 안전하지 않게 된다. 따라서 이산대수 문제를 푸는 것이 계산상 불가능하도록 소수 P와 Q의 크기를 선택해야 한다.

현재까지 알려진 이산대수를 푸는 범용 알고리즘 중 가장 빠른 알고리즘은 Generalized Number Field Sieve(GNFS)이다. 한편 Q가 작은 특수한 경우에는 Pollard's rho 알고리즘을 사용하는 것이 보다 효과적일 수 있다. GNFS는 그 계산 복잡도가 소수 P의 크기에만 의존하는 범용 알고리즘이다. Pollard's rho 법의 계산 복잡도는 주로 Q의 크기에 의존하나 소수 P의 크기도 약간의 영향을 미친다. 그러나 그 영향은 그렇게 크지 않으므로 보통 Q의 크기만으로 그 계산 복잡도를 추정한다.

1994년을 기준으로 전문가들이 예측한 GNFS 및 Pollard's rho 알고리즘의 계산 복잡도는 [표 4]와 같다. 여기서 MY는 Mips-Years의 약자이고 1 Mips-Year는 대략 1 MIPS (Million Instructions Per Second)의 프로세서가 1년간 행할 수 있는 계산량을 의미한다. 이러한 예측은 현재의 알고리즘과 기술 발달 정도를 기준으로 산정된 것이며, 향후 획기적인 알고리즘이나 하드웨어의 발전이 없는 한 어느 정도는 이산대수 문제의 난이도에 따른 매개변수의 크기 선정에 참고가 될 것이다.

|P|	512	768	1024	1280	1536	2048
MY	$3 \cdot 10^4$	$2 \cdot 10^8$	$3 \cdot 10^{11}$	$1 \cdot 10^{14}$	$3 \cdot 10^{16}$	$3 \cdot 10^{20}$

* 1994년 기준으로 2014년의 예상되는 가용 계산력 : $10^{11} \sim 10^{13}$MY
* |Q| = 160, |P| = 1024에 대한 Pollards rho법의 계산 복잡도 : $>10^{14}$MY

[표 4] |P|, |Q|의 크기에 따른 안전도

이러한 예측을 근거로 전자서명에 사용되는 소수 P, Q의 크기에 대한 권고되는 값은 대략 |P| = 1024, |Q| = 160 정도이다. 그러나 매우 중요한 응용에 사용되는 서명의 경우는 (예를 들면 최상위 인증기관의 서명키) 2048 비트의 P와 256비트 정도의 Q를 사용하도록 권고한다. 만일 전자서명을 적용하고자 하는 응용이 안전도에 민감하지 않다면 |P| = 1024, |Q| = 128 정도를 사용할 수도 있다. P와 Q의 길이는 전자서명을 적용하고자 하는 응용의 안전도 요구사항 및 사용 환경 등을 고려하여 신중히 선택되어야 한다.

|H|비트의 해쉬코드를 출력하는 해쉬함수 h()는 birthday attack을 이용하면 약 $2^{|H|/2}$ 정도의 연산으로 충돌 쌍을 찾을 수 있다. 충돌 쌍의 생성은 곧 서명의 위조를 의미하므로 이산대수 문제를 푸는 것 이외의 다른 중요한 안전 위협 요소가 된다. 따라서 해쉬함수가

birthday attack에 견디도록 출력길이 |H|가 선택되어야 한다. Birthday attack에 요구되는 연산 수 $2^{|H|/2}$는 대략 |H|비트 길이의 Q에 대한 Pollard's rho 알고리즘에 의한 이산대수 계산에 요구되는 연산 수와 비슷하고, 해쉬코드의 길이가 |Q|보다 크다고 해도 서명 생성 과정에서 mod Q 연산이 사용되므로 더 큰 해쉬코드가 더 나은 안전도를 주지는 못하므로, 보통 해쉬코드의 길이를 Q의 비트 길이와 같게 정한다.

5. 적용범위 및 기대효과

전자서명은 정보처리시스템 또는 정보통신망 환경 하에서의 정보보호 서비스 구현에 필수적인 기술로 대부분의 암호 프로토콜 설계 시 기본적인 도구가 되며, 이 표준에서 규정하는 확인서 이용 전자서명 알고리즘은 임의의 길이를 갖는 메시지에 대해 서명자가 전자서명을 생성할 때와 검증자가 서명된 메시지의 서명이 정당한 서명인지 여부를 확인하고자 할 때 사용된다. 이 표준은 일반적으로 정보처리시스템 및 정보통신망 환경에서 인증, 무결성, 부인방지 서비스를 제공하고자 하는 모든 서비스에 적용될 수 있다.

이 표준을 통해 전자상거래, 전자 금융거래, 무역업무 자동화 등 전자서명 기능을 필수적으로 요구하고 있는 각종 정보통신 응용서비스 및 시스템 개발과 활용이 활성화 될 수 있을 것으로 기대된다.