# 무선 환경에 적합한 타원곡선 암호 알고리즘의 검증도구

서 창 호[†] · 홍 도 원[††] · 윤 보 현[†††] · 김 석 우[††††] · 이 옥 연[†††††] · 정 교 일[††††††]


## 요 약

기존의 공개키 암호알고리즘의 표준적합성에 대한 검증도구는 유선환경에 치중되어 개발되어 왔으나, 무선환경에서 사용되는 타원곡선 알고리즘의 표준적합성 검증도구에 대한 개발은 미약한 실정이다. 유선 환경보다 무선 환경에서 정보보호 시스템 간의 상호연동성 확보 및 사용자 편익을 증대시키기 위한 검증도구의 개발이 더욱 중요하다. 따라서 본 논문에서는 X9.62 기술표준을 정확하게 준용하여 구현되었는지를 테스트할 수 있는 타원 곡선 암호 알고리즘의 검증도구를 설계 및 구현하였다. 구현된 검증도구는 ECDSA, ECKCDSA, ECDH 등을 이용한 모든 정보보호 제품에 적용할 수 있다. 아울러 충분한 테스트 항목을 통해 검증의 정확성을 높였으며, 검증도구와 검증 대상이 온라인상에서 검증될 수 있도록 하였다.


# Validation Tool of Elliptic Curves Cryptography Algorithm for the Mobile Internet

Changho Seo[†] · Dowon Hong[††] · Bohyun Yun[†††]
Seokwoo Kim[††††] · Okyeon Lee[†††††] · Kyoil Chung[††††††]


## ABSTRACT

Conventional researches of standard tool validating public key cryptographic algorithm have been studied for the internet environment, not for the mobile internet. It is important to develop the validation tool for establishment of interoperability and convenience of users in mobile internet. Therefore, this paper presents the validation tool of Elliptic Curve Cryptography algorithm that can test if following X9.62 technology standard specification. The validation tool can be applied all information securities using ECDSA, ECKCDSA, ECDH, etc. Moreover, we can enhace the precision of validation through several experiments and perform the validation tool in the online environment.


**키워드 : CMVP(Cryptographic Module Validation Program), ECC 표준 적합성 검정 Tool(ECC Standards Validation Tool)**

## 1. 서 론

무선통신기술의 발달은 인터넷상에서 제공되는 전자상거래 서비스들이 무선상에서 가능하도록 무선 인터넷이라는 새로운 패러다임을 창출하였다. 무선 인터넷상에서의 정보보안 문제를 해결하기 위해서는 기존의 공개키 암호에 비하여 단위 비트 당 안전도가 높고 키 사이즈가 작으며 구현했을 때 수행속도가 빠른 무선 공개키 암호가 필요하다. 이러한 요구는 기존의 RSA 암호시스템[1]으로 해결하기 어렵다는 것이 일반적인 견해이며, 현재까지 보고 된 바에 따르면 타원곡선 암호시스템은 RSA시스템과 비교해서 10~20배 정도 빠르게 작동하는 것으로 알려져 있다.

무선 PKI 기술 기준에 정의된 타원 곡선 암호 알고리즘(ECC)[2]은 RSA와 DSA에 비교하였을 때 여러 가지 장점을 지니고 있어 주목을 받아왔다. 타원곡선 암호(ECC)는 특히 에너지 소모가 적고, 키 사이즈 작으며 서명의 길이가 짧다는 점 때문에 IC카드나 무선 단말기 등에 적용이 가능하다. 특히 무선 PKI에서 전자 서명 알고리즘으로 사용되는 ECDSA는 타원곡선(Elliptic Curve)상에서 군(Group)을 정의하고 이에 대한 이산대수 문제의 어려움에 근거를 두고 있다. 타원 곡선상에서의 이산대수 문제는 일반적인 군에서 정의되는 이산대수 문제보다 훨씬 어려우며, 이에 따라 작은 키로도 RSA 보다 높은 비도를 유지할 수 있다는 장점 때문에 많이 사용되고 있다.

ECDSA는 ANSI X9.62[5]와 IEEE P1363[6] 표준 위원회에서 표준으로 채택되어지고 있다. 무선 환경에서 WPKI를 지원하기 위한 알고리즘의 커브 파라미터 등이 각각 WTLS[4], ANSI X9.62[5], FIPS 186-2[3]에서 권고하고 있다.

※ 본 논문은 2004년도한국과학재단 NO.R01-2003-000-10236-0 연구비에 의해 연구되었음.
† 정 회 원 : 공주대학교 응용수학과 교수
†† 준 회 원 : 한국전자통신연구원 정보보호기반연구팀 팀장
††† 정 회 원 : 목원대학교 컴퓨터교육과 교수
†††† 종신회원 : 한세대학교 IT학부 교수
††††† 정 회 원 : 국민대학교 수학과 교수
†††††† 종신회원 : 한국전자통신연구원 정보보호기반연구 그룹장
논문접수 : 2004년 3월 22일, 심사완료 : 2004년 7월 12일

기존 유선 PKI를 구축시 공인 인증기관이 전자 서명 알고리즘(RSA, DSA, KCDSA), 해쉬 함수(HAS160, SHA-1) 및 키 생성 관련 등에 관하여 실질심사 수감시 효과적으로 사용하기 위한 알고리즘 검증도구를 개발하였듯이, 무선 PKI 체계에서 사용되는 타원곡선 기반의 공인 시스템에 대한 적절한 평가기준의 마련이 시급하다. 특히, 이러한 공인 시스템은 공인 인증기관 지정 기준에 부합하는 안전성, 신뢰성 등을 갖추어야 하며, 관련 표준문서를 준수하여 개발하여야 할 필요성이 있다. 평가의 정확성과 신뢰성을 높이기 위해서는 평가시 신청자의 타원곡선 기반의 공인 시스템이 표준에 따라 적합하게 사용하였는지 검증하는 절차가 필요하다. 이 때 검증의 원활한 수행을 위해서는 검증을 전문적으로 수행할 수 있는 검증도구의 개발이 요구된다.

이에 본 논문에서는 ECC기반 알고리즘(ECDSA, ECKCDSA) 등이 기술표준을 정확하게 준용하여 구현되었는지 여부를 테스트하는 검증도구를 설계 및 구현하였다. 본 논문의 2장에서는 타원곡선 암호 시스템에 관하여 기술한다. 그리고 3장과 4장에서는 ECDSA, ECKCDSA 등의 구현물에 대한 검증을 수행할 수 있는 검증도구의 설계 및 구현 내용에 대해서 살펴본다. 마지막으로, 5장에서 결론을 맺는다.

## 2. 타원곡선 암호 시스템

일반적으로, 공개키 암호시스템으로는 소인수 분해의 어려움에 근거한 시스템(RSA)[1]과 이산대수 문제의 어려움에 근거한 시스템(DSA), 그리고 타원곡선상의 이산대수 문제에 근거한 시스템(ECC)이 주로 사용된다. 이중 타원곡선 암호 시스템 비트당 안전도가 가장 높은 암호 시스템으로 작은 사이즈의 키 값만으로도 높은 안전성을 보장한다. <표 1>은 타원곡선 암호시스템(ECC), RSA, DSA의 비트당 안전성을 비교한 표이다[2].

〈표 1〉 RSA, DSA, ECC 안전성 비교

| Time to break in MIPS years | RSA/DSA key size | ECC key size | RSA/ECC key size ratio |
|---|---|---|---|
| $10^4$ | 512 | 106 | 5 : 1 |
| $10^8$ | 768 | 132 | 6 : 1 |
| $10^{11}$ | 1,024 | 160 | 7 : 1 |
| $10^{20}$ | 2,048 | 210 | 10 : 1 |
| $10^{78}$ | 21,000 | 600 | 35 : 1 |

모듈러 지수승 연산이 RSA 암호시스템의 성능을 좌우하듯이 타원곡선 암호시스템의 성능은 스칼라 곱셈 연산에 의하여 좌우된다. 스칼라 곱셈 연산은 임의의 랜덤수 $k$와 타원 곡선 위의 한점 P의 곱셈 연산으로 정의되며, 타원곡선 위의 점 $P$의 $k$번 덧셈연산으로 계산된다. 이때 타원곡선 의 덧셈연산은 결과값이 다시 타원곡선 위의 점이 되도록 (알고리즘 1)과 같이 정의되어야 한다(단, Polynomial 기반 유한체를 위한 타원곡선 식은 $y_2+xy=x_3+ax_2+b$과 같이 주어지며, $P_1$과 $P_2$이 타원곡선 상의 존재한다.).

Input : $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$.
Output : $P_3 = P_1 + P_2 = (x_3, y_3)$.
1. If $P_1 = P_2$ (doubling)
   $x_3 = \lambda^2 + \lambda + a$,
   $y_3 = x_1^2 + \lambda(\lambda + 1)\, x_3$
   where ( $\lambda = x_1 + y_1 / x_1$)
2. Else if $P_1 \neq P_2$ (point addition)
   $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$,
   $y_3 = \lambda(x_1 + x_3) + x_3 + y_1$
   where ( $\lambda = (y_2 + y_1)/(x_2 + x_1)$)
3. Return ($x_3$, $y_3$)

(알고리즘 1) Point Addition Equation

## 3. 검증도구 설계

본 장에서는 본 논문에서 설계 및 구현한 검증도구를 소개한다. 검증도구는 크게 전자서명 알고리즘에 대해서 테스트하는 부분으로 구분할 수 있다. 검증 대상이 되는 전자서명 알고리즘은 ECDSA와 ECKCDSA이며, 각각의 알고리즘 검증은 여러 개의 세부항목으로 구성된다. 검증 도구는 ANSI X9.62[5]와 Certicom사에서 나온 SEC 2[9], ECKCDSA 표준안에 나오는 테스트 벡터들에 대해 검증을 하였고, 모든 경우에 대해 오류 없이 잘 프로그램이 작동하는지를 일일이 검사하였다. 검증은 검증을 수행하는 검증도구와 검증을 받는 검증 대상으로 이루어진다. 검증도구에서 제공하는 정보를 이용해 검증 대상은 각각의 항목에 해당하는 정보를 생성하여 검증도구에 제출해야 한다. 또한 검증 도구와 검증 대상이 원격으로 떨어져있는 경우에도 검증이 수행 가능하도록 하였다.

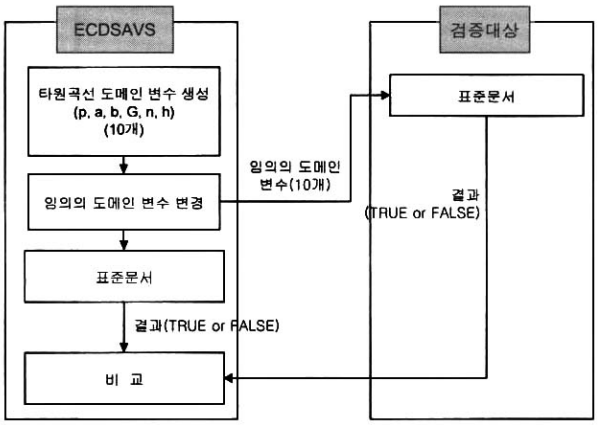
### 3.1 ECDSA 검증

ECDSA는 타원곡선 위에서 정의된 전자 서명 알고리즘으로, 전자 서명 표준 알고리즘인 DSA에 대한 타원곡선 버전이다. ECDSA(Elliptic Curve DSA)는 DSA를 타원곡선 알고리즘으로 옮긴 것으로 ANSI X9.62로 표준화되었다. 따라서 본질적인 알고리즘은 유한체 위에서의 DSA와 동일하다. ECDSA는 ANSI X 9.62에서 기술한 알고리즘을 적합하게 구현하였는가에 대해 검증한다. ECDSAVS에 의한 ECDSA테스트는 다음과 같은 5가지 항목으로 구성된다. 이 때, 5가지의 테스트를 모두 통과해야만 적합하게 구현한 것으로 간주한다.

- 타원곡선의 종류 및 타원곡선에서 사용되는 파라미터 적합성 테스트
- 키 쌍 생성 테스트
- 전자서명생성테스트
- 전자서명 검증 테스트
- 의사난수 생성 테스트

3.1.1 타원곡선의 종류 및 타원곡선에서 사용되는 파라미터 적합성 테스트

실질적으로 대부분 타원곡선 서명 알고리즘을 사용하는 곳에서는 표준에서 정하고 있는 표준 곡선에 대해서만 구현하고 있기 때문에 도메인 파라미터 생성 및 검증에 대해 테스트 하지 않고, 구현물에서 사용하고 있는 표준 곡선에 대한 값의 정확성 검증을 원칙으로 설계하였다. 본 논문에서는 ANSI X9.62, SEC2 등 표준문서에서 사용하는 유한체 GF(p)[10]와 GF(2m)[10]와 곡선(Curve)에 따라 가장 작은 112비트부터 571비트의 다양하게 구현하여 검증하였다. 검증대상이 표준문서에서 승인된 방법을 통해 타원곡선 도메인 변수인 커브 번호(Curve ID) 기저 필드(Based field), 소수(p), a, b, 기저점(G), 기저점의 위수(n), 여인자(h)가 선택되었는지 검증한다. 이에 대한 테스트 절차는 (그림 1)과 같다.

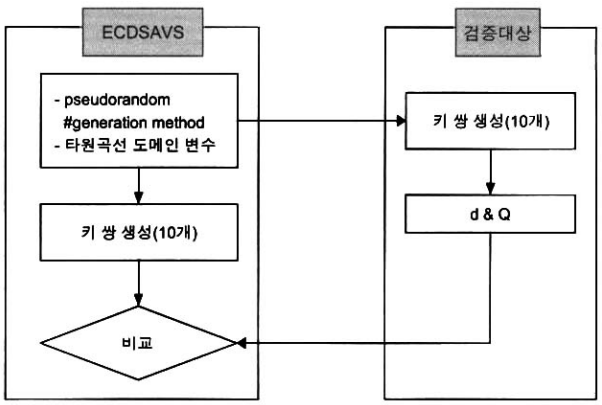



(그림 1) 타원곡선 도메인 변수 테스트

① 검증도구(ECDSAVS)는 임의의 도메인 변수(p, a, b, G, n, h) 10개를 생성한다.
② ECDSAVS는 생성한 도메인 변수(p, a, b, G, n, h) 일부를 변경한 후 테스트 데이터를 검증대상에게 전달한다.
③ ECDSAVS는 테스트 데이터에 대하여 표준문서의 승인 여부에 따른 결과 파일을 저장한다.
④ 검증대상은 테스트 데이터에 대한 결과를 ECDSAVS에게 제출한다.
⑤ 결과를 비교한다.

3.1.2 키쌍 생성(Key Generation for Private and Public Key Pairs)

타원곡선 도메인 변수를 기반으로 올바른 전자서명키를 생성할 수 있는 능력을 검증한다. 본 검증 도구에서는 통계적으로 유일하고 예측이 불가능한 정수 d를 [1, n-1]에서 선택한다. 난수 생성기가 사용된 경우, ANSI X9.62의 Annex A.4에 나와있는 방법으로 구현하였다. 이에 대한 테스트 절차는 (그림 2)와 같다.




(그림 2) 키 쌍 생성

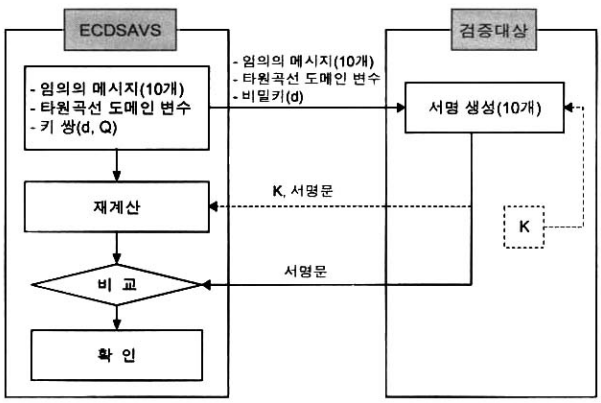
① 검증도구(ECDSAVS)는 키 쌍을 생성하는데 필요한 정보인 난수 생성 방법, 타원곡선 도메인 변수를 검증 대상에 전달하고. 이 정보를 이용하여 키 쌍을 생성한다.
- 난수 발생기 함수 Rand( ) 함수를 수행할 때, seed 값은 xkey의 값으로 사용하고, 업데이트한다.

② 검증대상은 ECDSAVS에게 받은 정보를 이용하여 키 쌍을 생성하여 ECDSAVS에게 전달한다.
③ ECDSAVS는 자신이 생성한 키쌍과 검증 대상으로부터 받은 키쌍을 비교하며, 모두 동일하면 테스트를 통과한 것으로 한다.

3.1.3 전자서명 생성(Signature Generation)

이 테스트는 검증 대상이 비밀키를 이용하여 정확한 전자서명을 생성할 수 있는 능력 검증한다.

전자서명 생성 테스트는 검증대상자가 정당한 개인키 d를 생성해서 메시지에 대한 올바른 서명 값(r, s)를 생성할 수 있는지를 검증한다. 여기서는 단순히 비밀키를 이용한 전자서명만을 테스트하며, 해쉬함수를 이용하여 메시지 다이제스트를 생성하는 것에 대한 테스트는 제외한다. 이에 대한 테스트 절차는 (그림 3)과 같다.




(그림 3) 전자서명 생성

① 검증도구(ECDSAVS)는 검증된 타원곡선 도메인 변수

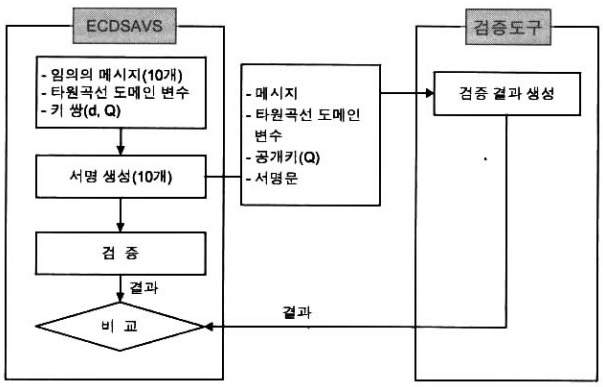
를 생성하고 이를 통해 하나의 공개키 쌍을 생성하고, 10개의 임의의 메시지를 생성하여 검증 대상에게 전달한다.

② 검증대상은 ECDSAVS에게 받은 정보를 이용하여 메시지에 대해 전자서명을 수행한 결과를 ECDSAVS에게 전달한다. 이때, 전자 서명시 사용된 k의 입력은 optional user input 이 있는 경우에는 이를 xseed 값으로 받아들여서 전자 서명을 생성하거나 난수 발생기로부터 고정된 k를 발생하여 전자 서명을 생성한다.

③ k와 서명된 메시지를 전달받아 ECDSAVS는 k를 이용하여 재계산하고 이를 비교한다.

3.1.4 전자서명 확인(Signature Verification)

테스트 대상이 전자서명의 유효성 확인을 올바르게 수행하는지 테스트한다. 즉, 전자서명 메시지가 변조된 경우, 이를 확인할 수 있는지 테스트한다. 이에 대한 테스트 절차는 (그림 4)와 같다.




(그림 4) 전자서명 확인

① ECDSAVS는 키쌍과 임의의 메시지를 생성한 뒤, 이를 이용해 메시지에 전자서명 한다.

- 전자서명값에 대한 1/2 정도를 임으로 선택하여 위조한다. 본 논문에서는 서명 위조하는 방법은 서명값 중 각각 r, s을 해쉬값으로 대치 한다. 또는 r 값과 s 값을 '0' 스트링으로 대치하는 방법을 사용한다.

② 타원곡선 도메인 변수, 공개키, 메시지, 전자서명된 메시지를 검증 대상에게 전달한다.

- 검증 대상에게 보낸 메시지 중에서 1/2는 정당한 전자 서명으로 1/2는 정당하지 않는 전자서명으로 위조한 것이다.

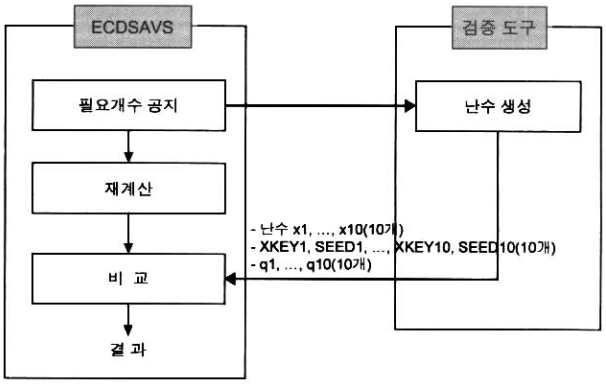
③ 검증 대상은 전달받은 정보를 통해 전자서명의 유효성 여부를 확인하여 그 결과를 ECDSAVS에게 전달한다.

- 검증한 결과가 정당하면 '1', 정당하지 않을 경우, '0' 또는 '2'를 보낸다. 여기서 '0'은 전자 서명이 올바르지 않을 경우, '2'는 전자 서명값 r, 혹은 s가 올바르지 않을 경우이다.

④ ECDSAVS는 검증 대상으로부터 전달 받은 결과를 자신의 확인 결과와 비교한다.

3.1.5 의사난수 생성 테스트

검증대상이 표준에서 제시하고 있는 의사난수 알고리즘을 올바르게 준용하고 있는지의 여부 확인한다. 이에 대한 테스트 절차는 (그림 5)와 같다.




(그림 5) 의사난수 생성 테스트

① 검증도구는 필요한 난수 및 관련 데이터의 수를 공지(10개)한다.

② 검증대상은 10개의 난수와 각 난수를 생성하는데 사용된 관련 데이터(XKEY, XSEED, Q(mod 연산에 사용))를 함께 검증도구에게 전달한다.

③ 검증도구는 제출된 값들을 계산후 비교한다.

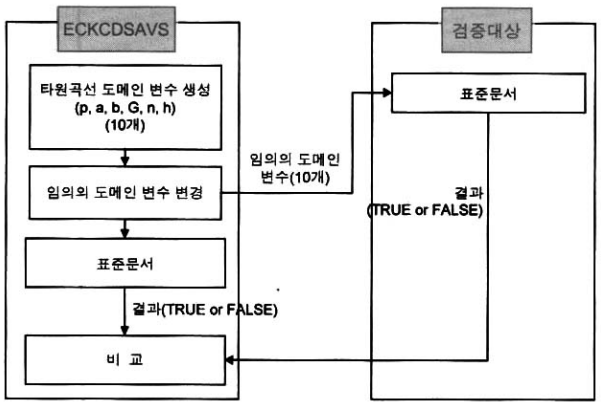
3.2 ECKCDSA

ECKCDSA는 KCDSA를 타원곡선 위의 알고리즘으로 옮겨 놓은 것으로 현재 표준화 작업이 진행 중에 있다. 본질적인 알고리즘은 유한체 위에서의 KCDSA와 동일하다. ECKCDSA 검증은 다음과 같은 4가지 항목으로 구성된다. 이 때, 4가지의 테스트를 모두 통과해야만 적합하게 구현한 것으로 간주한다.

- 타원곡선의 종류 및 타원곡선에서 사용되는 파라미터 적합성 테스트
- 키 쌍 생성 테스트
- 전자서명 생성 테스트
- 전자서명 검증 테스트

3.2.1 타원곡선의 종류 및 타원곡선에서 사용되는 파라미터 적합성 테스트

실질적으로 대부분 타원곡선 서명 알고리즘을 사용하는 곳에서는 표준에서 정하고 있는 표준 곡선에 대해서만 구현하고 있기 때문에 검증대상이 표준문서에서 승인된 방법을 통해 타원곡선 도메인 변수(p, a, b, G, n, h, seed)가 선택되었는지 검증한다. 이에 대한 테스트 절차는 (그림 6)과 같다.

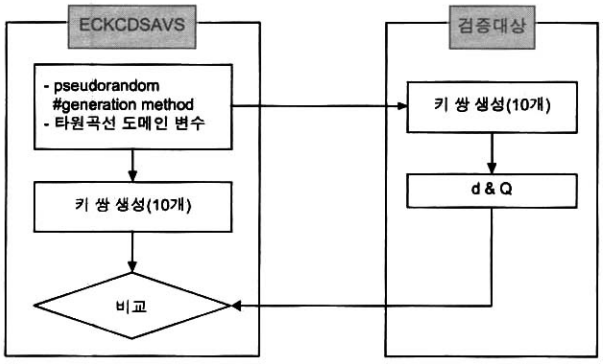



(그림 6) 타원곡선 도메인 변수 테스트

① 검증도구(ECKCDSAVS)는 임의의 도메인 변수(p, a, b, G, n, h) 10개를 생성한다.
② ECKCDSAVS는 생성한 도메인 변수(p, a, b, G, n, h) 일부를 변경한 후 테스트 데이터를 검증대상에게 전달한다.
③ ECKCDSAVS는 테스트 데이터에 대하여 표준문서의 승인 여부에 따른 결과 파일을 저장한다.
④ 검증대상은 테스트 데이터에 대한 결과를 ECKCDSAVS에게 제출한다.
⑤ 결과를 비교한다.

3.2.2 키쌍 생성(Key Generation for Private and Public Key Pairs)

E, G, n을 기반으로 올바른 전자서명키를 생성할 수 있는 능력을 검증한다. 이에 대한 테스트 절차는 (그림 7)과 같다.

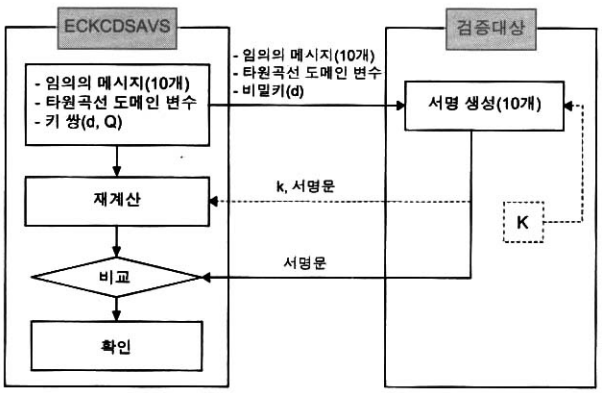



(그림 7) 키 쌍 생성

① 검증도구(ECKCDSAVS)는 키 쌍을 생성하는데 필요한 정보인 난수 생성 방법, 타원곡선 도메인 변수를 검증 대상에 전달하고. 이 정보를 이용하여 키 쌍을 생성한다.
② 검증대상은 ECKCDSAVS에게 받은 정보를 이용하여 키쌍을 생성하여 ECKCDSAVS에게 전달한다.
③ ECKCDSAVS는 자신이 생성한 키쌍과 검증 대상으로부터 받은 키쌍을 비교하며, 모두 동일하면 테스트를 통과한 것으로 한다.

3.2.3 전자서명 생성(Signature Generation)

검증대상의 정확한 전자서명을 생성할 수 있는 능력을 검증한다. 이에 대한 테스트 절차는 (그림 8)과 같다.

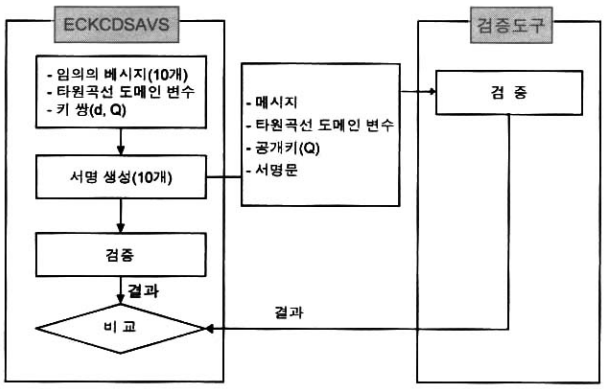



(그림 8) 전자서명 생성

① 검증도구(ECKCDSAVS)는 검증된 타원곡선 도메인 변수를 생성하고 이를 통해 하나의 공개키쌍을 생성하고, 10개의 임의의 메시지를 생성하여 검증 대상에게 전달한다.
② 검증대상은 ECKCDSAVS에게 받은 정보를 이용하여 메시지에 대해 전자서명을 수행한 결과를 ECKCDSAVS에게 전달한다. 이 때 테스트 대상에서 전자서명을 생성하는데 내부 파라미터 k가 사용된다. 따라서 k와 서명된 메시지를 ECKCDSAVS에 전달하여야 한다.
③ k와 서명된 메시지를 전달받아 ECKCDSAVS는 k를 이용하여 재계산하고 이를 비교한다.

3.2.4 전자서명 확인(Signature Verification)

테스트 대상이 전자서명의 유효성 확인을 올바르게 수행하는지 테스트한다. 즉, 전자서명 메시지가 변조된 경우, 이를 확인할 수 있는지 테스트한다. 이에 대한 테스트 절차는 (그림 9)와 같다.




(그림 9) 전자서명 검증

① ECKCDSAVS는 키쌍과 임의의 메시지를 생성하여 뒤, 이를 이용해 메시지에 전자서명 한다.
- 전자서명값에 대한 50% 정도를 임으로 선택하여 위조한다. 서명 위조하는 방법은 서명값 중 각각 r, s을 해쉬값으로 대치 한다. 또는 r 값과 s 값을 '0' 스트링으로 대치하는 방법을 사용한다.

② 타원곡선 도메인 변수, 공개키, 메시지, 전자서명된 메시지를 검증 대상에게 전달한다.
- 검증 대상에게 보낸 메시지 중에서 1/2는 정당한 전자 서명으로 1/2는 정당하지 않는 전자서명으로 위조한 것이다.

③ 검증 대상은 전달받은 정보를 통해 전자서명의 유효성 여부를 확인하여 그 결과를 ECKCDSAVS에게 전달한다.
- 검증한 결과가 정당하면 '1', 정당하지 않을 경우, '0' 또는 '2'를 보낸다. 여기서 '0'은 전자 서명이 올바르지 않을 경우, '2'는 전자 서명값 r, 혹은 s가 올바르지 않을 경우이다.

④ ECKCDSAVS는 검증 대상으로부터 전달 받은 결과를 자신의 확인 결과와 비교한다.

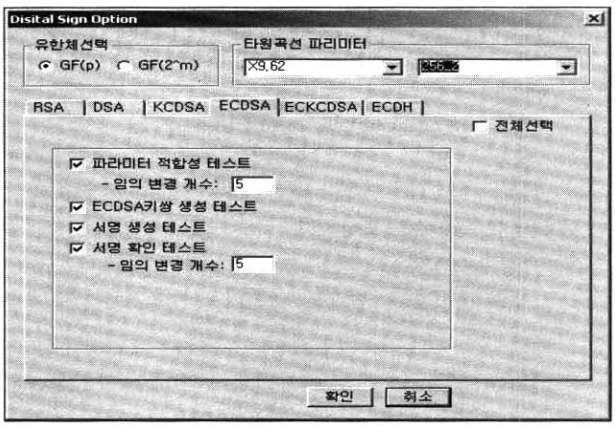
## 4. 검증도구 구현

본 논문에서 구현한 암호기술 구현물 검증도구는 전자서명 알고리즘인 KCDSA와 RSA, 해쉬 알고리즘 SHA-1과 HAS-160의 구현이 올바르게 이루어졌는지 테스트한다. 검증은 검증도구에서 생성한 검증에 필요한 정보를 검증 대상에게 전달하면, 검증 대상은 이 정보를 이용하여 전자서명 또는 해쉬를 수행한 후, 결과를 검증도구에게 전달한다. 검증도구는 검증 대상으로부터 전달받은 정보를 이용하여 검증을 수행한다. 이 때 검증에 필요한 정보는 모두 파일 단위로 전달되어지며, 이를 위해 검증도구와 검증대상은 여러 가지 파일을 생성하는데, 이를 정리하면 <표 2>와 같다.

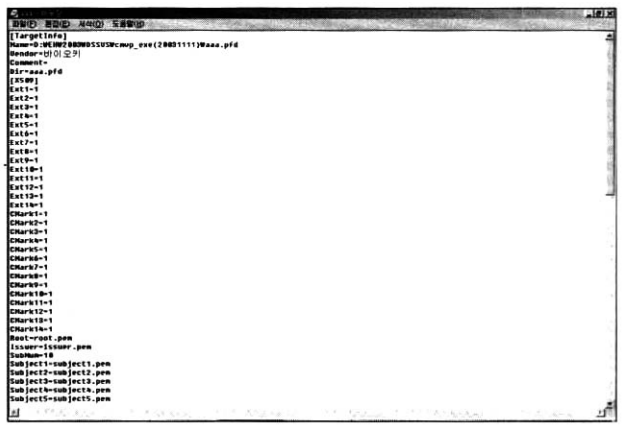
〈표 2〉 검증 수행 과정에서 생성되는 파일

| 구 분 (파일명) | 내 용 |
|---|---|
| 프로파일정보 (.pfd) | ◦ 검증에 앞서 검증도구에서 생성하는 정보<br>◦ 검증도구에서 검증을 위해 내부적으로 사용 |
| 초기정보 (.gep) | ◦ 검증에 앞서 검증도구에서 생성하는 정보<br>◦ 검증대상은 이를 이용하여 테스트 정보를 생성 |
| 테스트정보 (.rep) | ◦ 검증도구로부터 전달받은 정보를 통해 검증대상에서 생성<br>◦ 검증도구에 전달되어 검증 수행에 사용 |
| 결과정보 (.res) | ◦ 검증도구가 검증 수행후 생성하는 정보<br>◦ 검증대상의 검증 통과/실패 여부 기록 |
| 로그정보 (.log) | ◦ 검증 수행 과정에서 검증도구가 생성하는 정보<br>◦ 결과 정보보다 자세한 내용이 기록 |
| 검증도구결과 (.cot) | ◦ 검증도구에서 검증을 위해서 초기정보를 이용하여 생성하는 정보<br>◦ 테스트 정보와 검증도구 결과를 통해 검증 수행 |

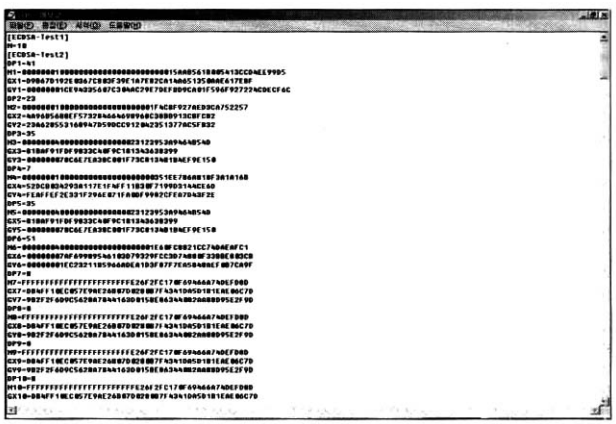
검증을 위해서는 (그림 13)과 같이 해쉬 알고리즘과 전자서명 알고리즘에 대한 검증을 위해 필요한 사항을 설정한다. 이 때, 사용자의 필요에 의해서 일부 항목에 대해서만 검증을 수행할 수 있도록 하였다. 검증도구는 이와 같은 설정 작업을 통해 프로파일 정보와 초기정보를 생성한다. (그림 14)와 (그림 15)은 프로파일 정보와 초기 정보 파일의 내용이다. 이 때 초기정보 파일은 해쉬 알고리즘에 대한 파일과 전자서명 알고리즘에 대한 파일이 각각 별개로 유지되며, 프로파일 정보 파일은 검증 대상에 대해서 하나가 존재한다.




(그림 13) 검증도구 초기 설정 화면



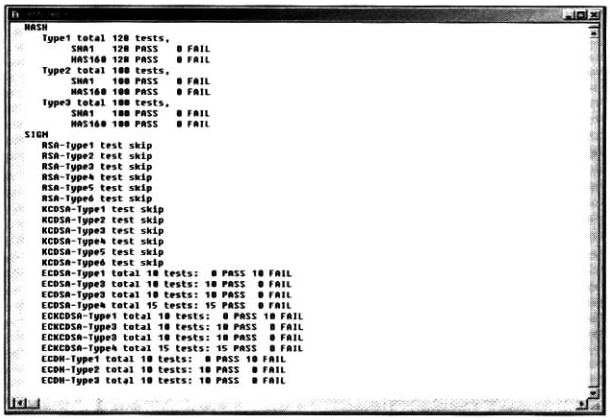
(그림 14) 프로파일 파일



(그림 15) 초기 정보 파일

검증도구는 생성된 초기정보 파일을 검증 대상에게 전달한다. 검증 대상은 초기정보 파일의 내용을 읽고, 이를 이용해서 검증 대상의 암호모듈을 이용해 초기정보 파일에 기술된 바에 따라 테스트 정보를 생성하고, 이를 테스트 정보 파일에 저장하여 검증도구에게 전달한다. 이 때, 검증 대상은 반드시 검증도구에서 요구한 형식에 따라 테스트 정보 파일을 생성하여야 한다.

(그림 16) 검증도구 결과 파일




(그림 17) 검증 결과

검증 대상으로부터 테스트 정보를 전달받은 검증 도구는 최초에 생성했던 초기정보를 이용하여 검증에 필요한 정보를 생성하고, 이를 검증도구 결과 파일에 저장한다. 예를 들어 해쉬 알고리즘에 대한 테스트의 경우, 초기정보 파일에는 메시지가 저장되어 있으며, 검증도구는 이를 통해 해쉬값을 계산하여 그 결과를 검증도구 결과 파일에 저장한다. 검증도구 결과 파일의 생성이 완료되면 검증도구는 검증도구 결과와 검증 대상이 생성한 테스트 정보를 이용해 검증을 수행한다. 검증도구는 검증도구 결과 파일 테스트 정보 파일을 이용해서 테스트를 수행하기 때문에 이 2개의 파일의 형식은 반드시 동일해야 한다. (그림 16)는 검증도구 결과 파일의 내용이다.

검증이 완료되면, 검증도구는 테스트의 성공/실패 여부를 결과 정보에 기록한다. 결과 정보 파일에는 성공/실패 여부와 함께 간략한 설명이 기록된다. 또한 검증 과정에서는 로그 정보 파일이 생성되는데, 이는 결과 정보 파일에 비해서 보다 자세한 내용이 생성된다. (그림 17)은 검증도구를 통해서 볼 수 있는 검증 결과이다.

이와 같은 검증도구와 구현 환경은 <표 3>과 같다.

〈표 3〉 개발 및 운용 환경

| 개 발 환 경 | | | | 운용환경 |
|---|---|---|---|---|
| 기 종 | 운영체제 | 언 어 | 라이브러리 | 운영체제 |
| Pentium IV | Windows 98/2000 | Visual C++ | - 자체 개발 | Windows 98/2000 |

또한 검증도구의 신뢰성 확보를 위해 관련 표준에서 제공하는 테스트 벡터와 이미 구현되어 있는 다른 암호모듈과의 연동소스 암호모듈인 OpenSSL과 Ctyptlib이다. 이와 같은 오픈소스 암호모듈은 오픈소스 정책에 따라 전 세계의 다양한 개발자들이 개발에 참여하고 있고, 메일링 리스트 등을 통한 지속적인 보완이 이루어지고 있어, 이런 면에서 안전성 및 신뢰성에 대한 상당 수준의 검증을 거쳤다고 볼 수 있다.

## 5. 결 론

인터넷을 통해 제공되는 응용 서비스가 대규모화 되고 복잡해짐에 따라 인터넷상의 정보를 보호하기 위한 보안기술 역시 빠르게 발전하고 있다. 그러나 안전하고 신뢰할 수 있는 통신환경의 구축을 위해서는 정보보호 서비스를 위해 사용되는 보안기술들의 안전성이 보장되어야 한다. 또한 사용자 편익 증진 및 서비스의 원활한 운용을 위해서는 보안기술들 간의 상호연동성이 이루어져야 한다.

보안기술에 있어서 안전성과 신뢰성 및 상호연동성의 확보를 위해서는 암호기술의 정확한 구현이 필수적이며, 이는 공개되어 있는 암호 알고리즘을 정확하게 준용하여 구현함으로써 이루어진다. 계속해서 새로운 보안시스템이 개발되고 있는 가운데, 이러한 보안시스템이 암호기술을 정확하게 구현하였는지 여부를 검증하는 작업은 매우 중요하다.

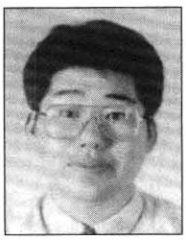
본 논문에서 설계한 검증도구는 각각의 암호기술을 여러 개의 세부항목으로 구분하고 있으며, 충분한 테스트 자료를 사용하여 검증의 정확성을 높였다. 또한 검증도구와 검증대상이 원격에 위치한 상태에서 검증을 수행할 수 있도록 하였다. 그러나 본 논문에서 설계 및 구현한 검증도구는 암호기술의 정확한 구현 여부만을 테스트할 수 있다. 즉, 암호기술을 잘못 구현하여 발생할 수 있는 보안 허점에 의한 안전성 문제 및 신뢰성에 대한 검증은 가능하지만, 암호기술 자체의 안전성 검증이나 구현물의 보안강도 평가는 수행하지 않는다.

본 논문에서 설계 및 구현한 검증도구는 ECDSA, ECKCDSA 등을 구현한 모든 보안 제품에 적용할 수 있다. 따라서 각종 암호제품의 평가 및 인증에 활용할 수 있을 것으로 기대된다.

## 참 고 문 헌

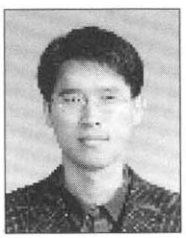
[1] R. L. Rivest, A. Shamir and L. M. Adleman, A method for ob-

taining digital ignatures and public-key cryptosystems, Communications of the ACM, Vol.21, pp.120-126, February, 1978.
[2] Certicom research, The Elliptic Curve Crypto-system, Certicom, April, 1997.
[3] Digital Dignature Standard (DSS), FIPS Publ- ication 186-2, National Institute of Standards and Technology, January, 2000.
[4] WAP WTLS Version 05-Nov-1999, Wireless Application Protocol Wireless Transport Layer Security Specification.
[5] Public Key Cryptography for Financial Services Industry : The Elliptic Curve Digital Signature Algorithm(ECDSA), ANSI X9.62-1998, January, 1999.
[6] IEEE P1363a : Standard Specifications for Public-Key Cryptography : Additional Techniques Draft 9, 2001.
[7] http://www.openssl.org.
[8] http://www.cs.auckland.ac.nz/~pgut001/dumpas.c.
[9] Certicom research, "SEC 2 : Recommended Elliptic Curve Domain Parameters," October, 1999.
[10] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography, CRC press, 1997.
[11] Richard Schroeppel, Hilarie Orman, Sean O'Malley, "Fast Key Exchange with Elliptic Curve Systems," TR-95-03 (Tucson, AZ : University of Arizona, Computer Sciences Department), 1995.
[12] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes, Software Implementation of Elliptic Curve Cryptography over Binary Fields, CHES 2000, pp.1-24. 2000.
[13] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, No.48, pp.203-209, 1987.



**서 창 호**

e-mail : chseo@kongju.ac.kr
1990년 고려대학교 수학과(학사)
1992년 고려대학교 일반대학원 수학과 (이학석사)
1996년 고려대학교 일반대학원 수학과 (이학박사)
1996년~1996년 국방과학연구소 선임연구원
1996년~2000년 한국전자통신연구원 선임연구원, 팀장
2000년~현재 공주대학교 응용수학과(정보보호전공) 부교수
관심분야 : 암호 알고리즘, PKI, 무선 인터넷 보안, 시스템 보안 등

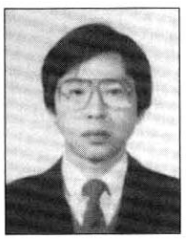


**홍 도 원**

e-mail:dwhong@etri.re.kr
1994년 고려대학교 수학과(학사)
1996년 고려대학교 일반대학원 수학과 (이학석사)
2000년 고려대학교 일반대학원 수학과 (이학박사)
2000년~현재 한국전자통신연구원 선임연구원, 팀장
관심분야 : 암호 이론, 정보보호 이론, 이동통신 정보보호 등

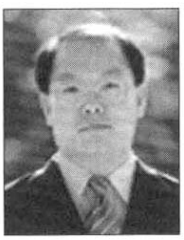
**윤 보 현**

e-mail : ybh@mokwon.ac.kr
1999년 고려대학교 컴퓨터학과(이학박사).
1999년~2003년 한국전자통신연구원(ETRI) 선임연구원 및 팀장
2001년~2003년 한국소프트웨어산업협회 언어정보산업협의회 운영위원
2002년~2002년 광인터넷 기술정책 자문위원
2003년~2004년 한국전자통신연구원(ETRI) 초빙연구원
2004년 KRnet 컨퍼런스 운영위원
2003년~현재 목원대학교 컴퓨터교육과 교수
관심분야 : 정보검색, 컨텐츠보호, 시멘틱웹, 바이오인포메틱 등



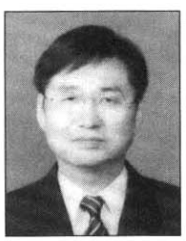
**김 석 우**

e-mail : swkim@hansei.ac.kr
1979년 한국항공대학교 통신정보공학과 (학사)
1989년 뉴저지 공과대학 전자계산학과 (공학석사)
1995년 아주대학교 컴퓨터공학과정보통신 전공(공학박사)
1980년~1997년 한국전자통신연구원 책임연구원, 실장
1997년~현재 한세대학교 IT학부 교수
관심분야 : 시스템 보안, 네트워크 보안, 시스템 평가 등



**이 옥 연**

e-mail : oyyi@kookmin.ac.kr
1988년 고려대학교 수학과 졸업
1990년 고려대학교 일반대학원 수학과 (이학석사)
1996년 University of Kentucky 수학과 (이학박사)
1999년~2001년 한국전자통신연구원 선임연구원, 팀장
2001년~현재 국민대학교 수학과 조교수
관심분야 : 정보보호, 이동통신, 암호론



**정 교 일**

e-mail : kyoil@etri.re.kr
1981년 한양대학교 전자공학과 졸업
1983년 한양대학교 전자계산학과 공학석사
1997년 한양대학교 전자공학과 공학박사
1980년~1981년 엠시스템즈 사원
1982년~현재 한국전자통신연구원 정보보호기반연구그룹장/책임연구원
관심분야 : IC 카드, Security, Biometrics, 국가기반보호, 신호처리