논문 14-39B-04-07
The Journal of Korea Information and Communications Society '14-04 Vol.39B No.04
http://dx.doi.org/10.7840/kics.2014.39B.4.242

암호 알고리즘 구현 적합성 평가 시스템 설계

하 경 주◆, 서 창 호°, 김 대 엽*

Design of Validation System for a Crypto-Algorithm Implementation

Kyeoung-Ju Ha◆, Chang-Ho Seo°, Dae-Youb Kim*

요 약

정보보호 시스템의 수준 향상과 안전성 및 신뢰성 확보를 위해서는 암호 알고리즘 자체에 대한 검증뿐만 아니라 암호 알고리즘을 구현한 구현물에 대한 검증이 필요하다. 특히, 암호 알고리즘에 대해서 국내외적으로 폭 넓은 표준화가 진행되고 있으며, 이들 암호 알고리즘에 대한 기술 표준을 정확하게 구현하는 것은 정보보호 시스템의 안전성, 신뢰성 향상 및 정보보호 시스템 간의 상호 연동성 확보면에서 매우 중요하다.

따라서 본 논문에서는 X9.62 기술표준을 정확하게 준용하여 구현되었는지를 테스트할 수 있는 암호 알고리즘의 검증도구를 설계 및 구현하였다. 구현된 검증도구는 DES, SEED, AES, SHA-1/256/384/512, RSA-OAEP V2.0, V2.1, 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.

Key Words : Cryptographic Module Validation Program, Validation of Cryptographic Standards Algorithms

ABSTRACT

Conventional researches of standard tool validating cryptographic algorithm have been studied for the internet environment, for the mobile internet. It is important to develop the validation tool for establishment of interoperability and convenience of users in the information systems.

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 DES, SEED, AES, SHA-1/256/384/512, RSA-OAEP V2.0, V2.1, ECDSA, ECKCDSA, ECDH, etc. Moreover, we can enhance the precision of validation through several experiments and perform the validation tool in the online environment.

Ⅰ. 서 론

네트워크 보안 시스템 평가는 국내의 경우 침입차단시스템과 침입탐지시스템에 대한 보안성 정도에 따른 평가가 이루어지고 있으며, CC(Common Criteria)를 도입하고 있다. 이러한 평가체계는 평가 시스템의 보안성에 대한 기능과 보증에 대한 요구사항의 검증을 실시하고, 암호 모듈에 대한 평가는 제외하고 있다.

CC의 경우 암호 모듈 평가는 미국과 캐나다에서 공동으로 개발한 CMVP(Cryptographic Module Validation

◆ First Author : Daeguhaany University, kjha@dhu.ac.kr, 정회원
° Corresponding Author : Kongju National University, chseo@kongju.ac.kr, 정회원
* 수원대학교 정보보호학과, daeyoub69@gmail.com
논문번호 : KICS2014-04-120, Received April 7, 2014; Revised April 17, 2014; Accepted April 17, 2014

Program)에서 암호 모듈과 암호 알고리즘 구현 적합성에 대한 검증을 실시하고 있다. 국내의 경우 아직까지 암호 모듈 평가를 위해 선정된 암호 알고리즘은 없으나 일부 표준화된 암호 알고리즘은 있으며, 세계적으로 사실상 표준으로 선정된 암호 알고리즘도 다수 있다. 따라서 향후 국내에서 개발 될 암호 모듈의 호환성 및 국제적인 수준의 제품 개발을 위해서는 국내 표준 및 사실상의 세계적인 표준으로 자리 잡은 암호 알고리즘을 대상으로 구현 적합성 평가 시스템 구축이 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 사이즈가 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 곡선 상에서의 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[12]와 IEEE P1363[13] 표준 위원회에서 표준으로 채택되어지고 있다. 무선 환경에서 WPKI를 지원하기 위한 알고리즘의 커브 파라미터 등이 각각 ANSI X9.62[12], FIPS 186-2[4]에서 권고하고 있다.

이에 본 논문에서는 대칭키, 공개키, 서명 알고리즘 등이 기술표준을 정확하게 준용하여 구현되었는지 여부를 테스트하는 검증도구를 설계 및 구현하였다. 본 논문의 2장에서는 타원곡선 암호 시스템에 관하여 기술한다. 그리고 3장과 4장에서는 DES, SEED, AES 및 전자서명 알고리즘인 DSA, KCDSA, ECDSA, EC-KCDSA와 RSA-OAEP V2.0, V2.1, 해쉬 알고리즘 SHA-1/256/384/512, HAS-160, MD5, RIPEMD 128/160 등의 구현물에 대한 검증을 수행할 수 있는 검증도구의 설계 및 구현 내용에 대해서 살펴본다. 마지막으로, 5장에서 결론을 맺는다.

Ⅱ. 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 암호시스템 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].

모듈러 지수승 연산이 RSA 암호시스템의 성능을 좌우하듯이 타원곡선 암호시스템의 성능은 스칼라 곱셈 연산에 의하여 좌우된다. 스칼라 곱셈 연산은 임의의 랜덤수 k와 타원 곡선 위의 한점 P의 곱셈 연산으로 정의되며, 타원곡선 위의 점 P의 k번 덧셈연산으로 계산된다. 이때 타원곡선 의 덧셈연산은 결과값이 다시 타원곡선 위의 점이 되도록 [Algorithm 1]과 같이 정의 되어야 한다. (단, Polynomial 기반 유한체를 위한 타원곡선 식은 $y_2 + xy = x_3 + ax_2 + b$과 같이 주어지며, P_1과 P_2이 타원곡선 상의 존재한다.)

표 1. RSA, DSA, ECC의 안전성 비교
Table 1. Stability Comparison of RSA, DSA, ECC

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

[Algorithm 1] Point Addition Equation

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)

Ⅲ. 검증 도구 설계

본 장에서는 검증 도구에 대한 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[12]와 Certicom사에서 나온 SEC 2[14], ECKCDSA 표준안에 나오는 테스트 벡터들에 대해 검증을 하였고, 모든 경우에 대해 오류 없이 프로그램이 잘 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 112 비트부터 571 비트의 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 1과 같다.

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.

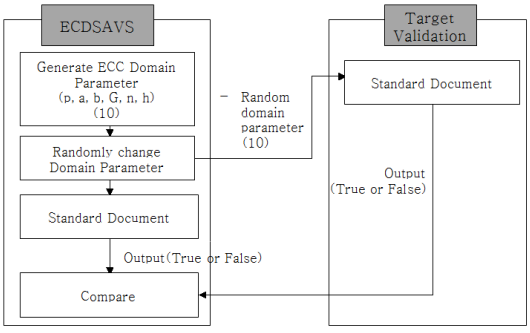

그림 1. 타원곡선에서 사용되는 파라미터 적합성 테스트
Fig. 1. Test of Elliptic curve domain parameters

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

타원곡선 도메인 변수를 기반으로 올바른 전자서명 키를 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 2와 같다.

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.

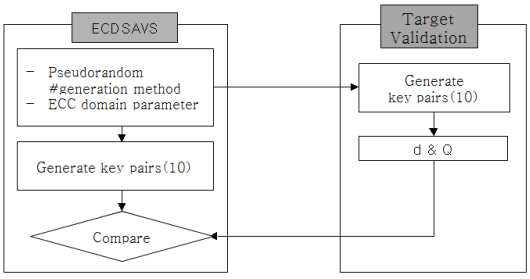

그림 2. 비밀키와 공개키 쌍 생성
Fig. 2. Key generation for Private and Public Key Pairs

② 검증대상은 ECDSAVS에게 받은 정보를 이용하여 키쌍을 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 능력을 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 3과 같다.

① 검증도구(ECDSAVS)는 검증된 타원곡선 도메인 변수를 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. 이 때, 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 있는지를 테스트하며 이에 대한 테스트 절차는 그림 4와 같다.

① ECDSAVS는 키 쌍과 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, s 을 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 1/2 는 정당한 전자 서명으로 1/2는 정당하지 않는 전자서명으로 위조한것이다.

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.

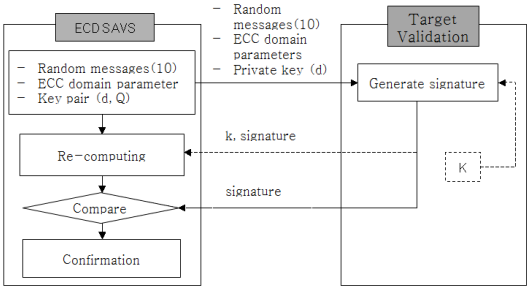

그림 3. 전자서명 생성
Fig. 3. Digital signature generation

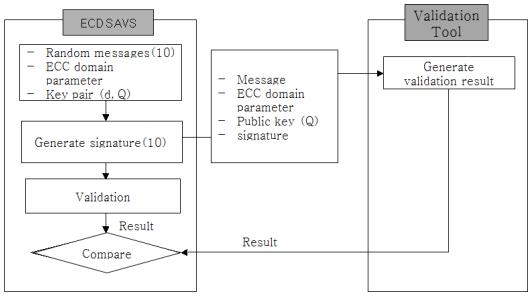

그림 4. 전자서명 확인
Fig. 4. Digital signature verification

3.1.5 의사난수 생성 테스트

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

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.

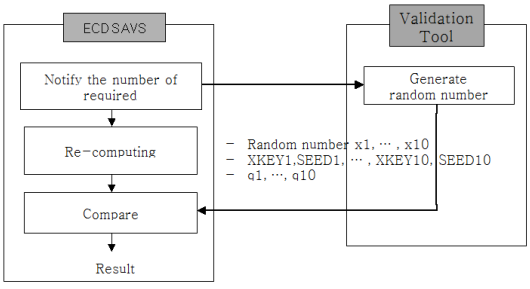

그림 5. 의사난수 생성 테스트
Fig. 5. Pseudo-random number generation test

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 6과 같다.

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.

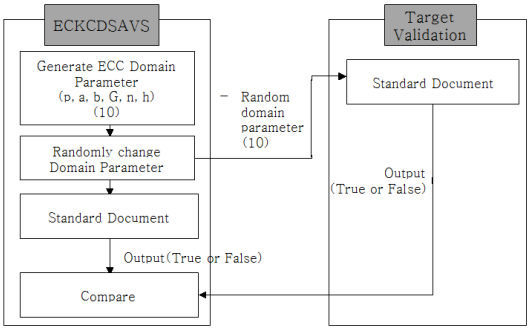

그림 6. 타원곡선 파라미터 테스트
Fig. 6. Test of Elliptic curve domain parameters

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 7과 같다.

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 키 쌍과 검증 대상으로부터 받은 키 쌍을 비교하며, 모두 동일하면 테스트를 통과한 것으로 한다.

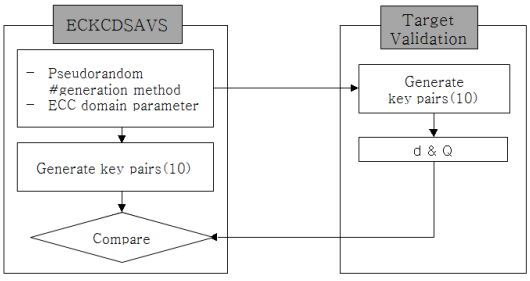

그림 7. 비밀키, 공개키 쌍 생성
Fig. 7. Key generation for Private and Public Key Pairs

3.2.3 전자서명 생성(Signature Generation)

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

① 검증도구(ECKCDSAVS)는 검증된 타원곡선 도메인 변수를 생성하고 이를 통해 하나의 공개키 쌍을 생성하고, 10개의 임의의 메시지를 생성하여 검증대상에게 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.

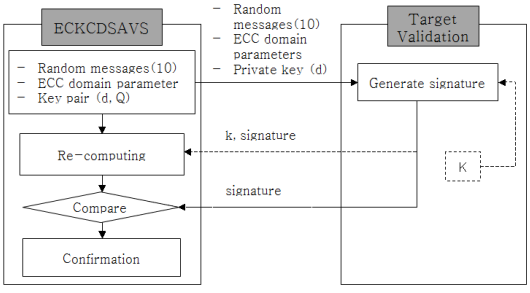

그림 8. 전자 서명 생성
Fig. 8. Digital signature generation

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 9와 같다.

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, s 을 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 1/2 는 정당한 전자 서명으로 1/2는 정당하지 않는 전자서명으로 위조한것이다.

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.

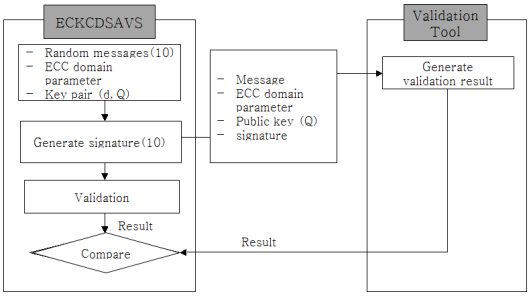

그림 9. 전자서명 확인
Fig. 9. Digital signature verification

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

Ⅳ. 검증도구 구현

본 논문에서 구현한 암호기술 구현물 검증도구는 DES, SEED, AES 및 전자서명 알고리즘인 DSA, KCDSA, ECDSA, EC-KCDSA와 RSA-OAEP V2.0, V2.1, 해쉬 알고리즘 SHA-1/256/384/512, HAS-160, MD5, RIPEMD 128/160 의 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 검증대상으로부터 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 2와 같다.

검증을 위해서는 그림 10과 같이 CMVS 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 11와 그림 12는 Request 파일 생성 기능과 알고리즘과 테스트 타입 선택의 내용이다. 이 때 초기정보 파일은 해쉬 알고리즘에 대한 파일과 전자서명 알고리즘에 대한 파일이 각각 별개로 유지되며, 프로파일 정보 파일은 검증 대상에 대해 하나가 존재한다.

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. 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 해쉬 값을 계산하여 그

표 2. 검증 중 생성되는 파일
Table 2. Generated files during the validation

Class (File Name)	Content
Profile Informatiom (.pfd)	• Generation Information before CMVS
Initial Informatiom (.gep)	• Initial Information before CMVS
Test Informatiom (.rep)	• Test Information before CMVS
Result Informatiom (.res)	• Generation Information after CMVS
log Informatiom (.log)	• Log Information after CMVS

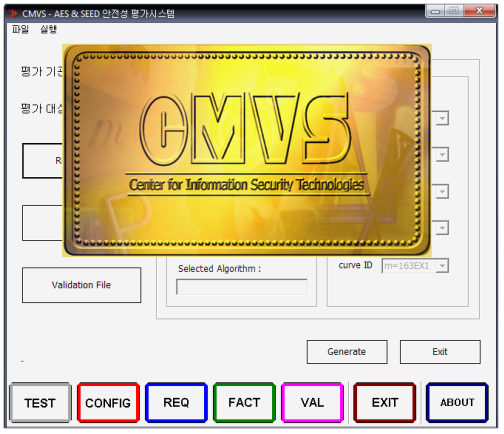

그림 10. CMVS 초기 실행
Fig. 10. CMVS initial execution

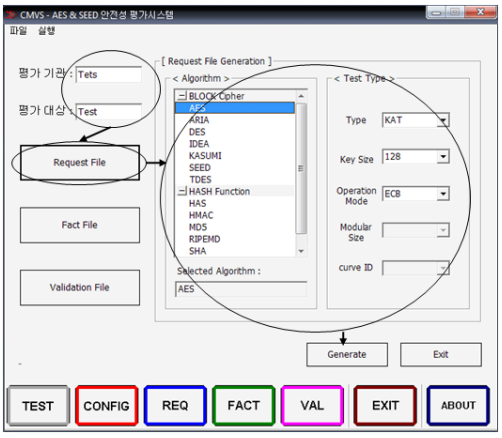

그림 11. 파일 생성 요구사항
Fig. 11. Request file generation

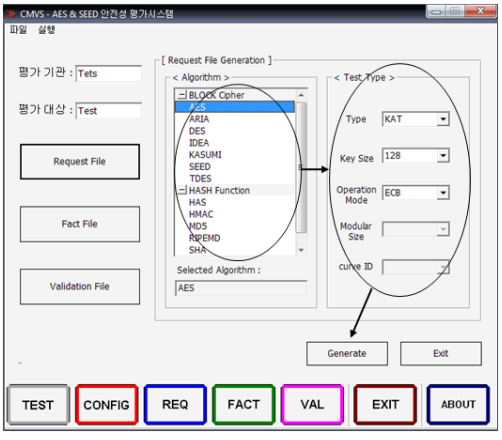

그림 12. 알고리즘과 테스트 타입 선택
Fig. 12. Selection of algorithm and test type

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 13은 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 14은 검증도구를 통해서 볼 수 있는 검증 결과이다.

이와 같은 검증도구와 구현 환경은 표 3과 같다.

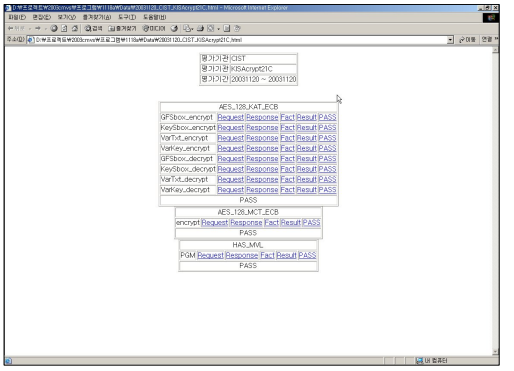

그림 13. 검증도구 결과 파일의 내용
Fig. 13. Contents of validation result file

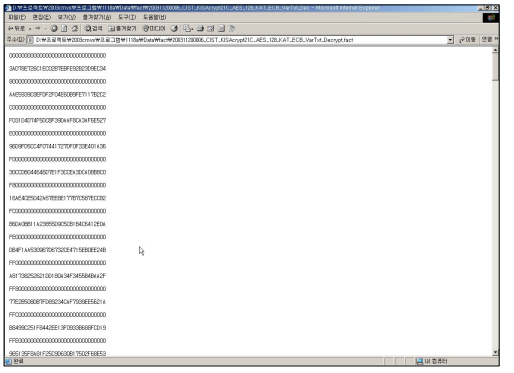

그림 14. 검증 결과
Fig. 14. Validation result

표 3. 개발 및 운용 환경
Table 3. Environment of development and operation

Development environment				using environment
Type	OS	Language	Lib	OS
Pentium V	Windows XP Professional	Visual C/C++	- Self Development	Windows XP Professional

V. 결 론

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. 공개기 알고리즘 RSA의 경우에는 PKCS#1 v2.1의 문서를 기준으로 키 쌍생성, 암호화, 복호화 연산에 대한 평가와 엔코딩과 디코딩, 또한 엔코딩과 디코딩을 함께 수행하는 암호화, 복호화, 서명, 서명검증에 대해 평가방법에 대해서 설명한다. 서명 알고리즘의 경우 DSA와 ECDSA는 NIST에서 제시한 DSSVS, ECDSAVS를 각각 참조하였다. 또한 KCDSA와 EC-KCDSA의 경우 각각 DSSVS와 ECDSAVS를 참조하여 평가방법을 제시하였다. 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. 암호알고리즘의 구현적합성 평가는 정보보호제품의 안정성을 결정하는 주요요인이므로 매우 중요한 부분이다. 정보보호제품의 보안 요구사항에 대한 평가는 CC를 통해 제품의 인증서 발급 등으로 이루어지고 있으나 암호알고리즘 구현 적합성 평가는 CC와 같이 평가기준에 의해 이루어지고 있지 못하고 있으며, CMVP를 통해 NIST에서 제시된 몇 개의 알고리즘 평가기준을 찾아볼 수 있을 정도 이다. 그 이유는 알고리즘의 특성에 따라 각 연산의 구현적합성을 판단할 수 있는 경우와 그렇지 않은 경우가 있기 때문이다.

References

[1] NIST, “Security requirement for cryptographic modules,” FIPS 140-1, 1994.

[2] NIST, “Security requirements for cryptographic Modules,” FIPS 140-2, 2001.

[3] NIST, “NIST SP 800-17 : MOVS,” 1998.

[4] NIST, “Derived test requirements for FIPS PUB 140-1, security requirements for cryptographic modules,” 2001.

[5] NIST, “NIST SP 800-20 : TMOVS,” 2000.

[6] NIST, “NIST SP 800-21 : AES,” 1999. 11.

[7] NIST, “Advanced encryption standard(AES),” FIPS PUB 197, 2001. 11.

[8] NIST, “Digital signature standard validation system (DSSVS),” 2001. 1.

[9] H. Raddum, "The statistical evaluation of the NESSIE submission Idea," Feb. 2002.
[10] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.
[11] Certicom research, "The Elliptic Curve Crypto-system," Certicom, Apr. 1997.
[12] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Security*, vol. 1, no. 1, pp. 36-63, Aug. 2001.
[13] *IEEE P1363a : Standard Specifications for Public-Key Cryptography: Additional Techniques Draft 9*, 2001.
[14] Certicom research, "SEC 2 : Recommended elliptic curve domain parameters," Oct. 1999.

하 경 주 (Kyeoung-Ju Ha)

1991년 2월 : 경북대학교 컴퓨터공학과 졸업
1993년 2월 : 경북대학교 컴퓨터공학과 석사
1996년 8월 : 경북대학교 컴퓨터공학과 박사
1996년 9월~1999년 8월 : 한국전자통신기술연구원 부호기술연구부 선임연구원
1999년 9월~현재 : 대구한의대학교 모바일콘텐츠학부 교수
<관심분야> 정보보호, Visual cryptography, Steganography

서 창 호 (Chang-Ho Seo)

1990년 2월 : 고려대 수학과(학사)
1992년 2월 : 고려대 수학과(석사)
1996년 8월 : 고려대 수학과(박사)
1996년 8월~2000년 : 한국전자통신연구원 선임연구원, 팀장
2000년 3월~현재 : 공주대 융합과학과(정보보호전공) 교수
<관심분야> 암호 알고리즘, PKI, 무선 인터넷 보안 등

김 대 엽 (Dae-Youb Kim)

2000년 2월 : 고려대학교 수학과 이학박사
2000년 3월 : (주)텔리맨 CAS팀 책임연구원
2002년 8월 : (주)시큐아이 정보보호연구소 PKI실 차장
2012년 2월 : 삼성전자 종합기술원 전문연구원
2012년 3월~현재 : 수원대학교 정보보호학과 조교수
<관심분야> 보안프로토콜, 콘텐츠 보안, 미래 인터넷 보안, 보안 코딩