情 報 保 護 學 會 誌
第 12 券 第 4 號, 2002. 8

암호라이브러리 및 암호API 개발현황

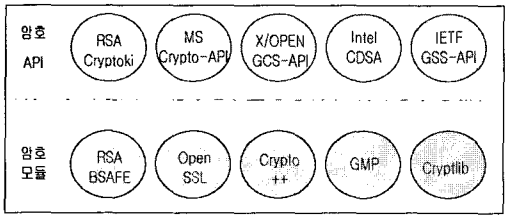
주 학 수*, 이 언 경*, 김 승 주*

요 약

본 고에서는 암호라이브러리 제품들에 기술을 제공하는 암호API의 국외 표준화 현황을 알아보고 이들 국외 API들을 비교·분석한다. 그리고 국내·외 암호라이브러리제품들을 지원 알고리즘, OS, API, 속도 등으로 비교·조사한 뒤, 안전성 면에서 많은 지적을 받고 있는 난수생성기를 기준으로 암호라이브러리 제품들을 분석한다. 또한 효율성 측면에서는 공개키 암호의 속도를 위주로 암호라이브러리 제품들의 성능을 비교·정리한다.

Ⅰ. 서 론

최근에 전자상거래, e-business, 무선 인터넷 결재 등 전자적 업무처리가 활성화가 됨에 따라 이에 따른 보안 위협요소 역시 증가하였다. 이러한 보안 위협요소를 방어하는데 요구되는 보안 시스템을 설계하고 구현하기는 어려우며 실제로 보안 시스템을 사용하기 위해서는 기본적인 원리에 대한 많은 양의 이해를 요구하는 경향이 있다.



※ 암호모듈 : 암호알고리즘 및 암호프로토콜을 구현한 모듈로 암호기술의 기본적인 기능만을 제공하는 모듈
※ 암호API : 암호에 대한 지식이 없는 소프트웨어/하드웨어 엔지니어가 암호모듈을 이용하여 보안제품을 개발하기 위해 사용하는 인터페이스.

〔그림 1〕 암호라이브러리 제품들의 전체적인 구조

이러한 보안에 대한 많은 지식 없이 보안에 대한 위협요소를 방어해주는 암호기능을 이용할 수 있게 해주는 것이 암호라이브러리 제품들이다. 즉, 암호라이브러리란 비전문 프로그래머가 보안 기능이 없는 소프트웨어에 암호화와 인증 등의 암호 서비스를 쉽게 추가할 수 있도록 해주는 강력한 보안 툴킷이다. 암호라이브러리 제품들의 전체적인 구성을 정리하면 그림 1과 같다.

암호API는 암호에 대한 비전문가에게 하위레벨에 대한 지식 없이 짧은 시간으로 어플리케이션 프로그래머들에게 국제적으로 인증된 암호 알고리즘과 인증 보안 서비스를 설치할 수 있게 한다. 이런 인터페이스의 활용은 새로운 어플리케이션 혹은 이미 있는 어플리케이션에 안전성을 추가하는데 드는 비용을 줄이는 데 큰 특징이 있다고 할 수 있다.

〔표 2〕 암호라이브러리 제품들의 특징

항 목	설 명
프로그램 인터페이스(API) 지원	표준화된 API를 지원함으로써 임의의 암호모듈과 알고리즘을 암호에 대한 비전문 프로그래머들에게 손쉽게 이용할 수 있게 함.
사용자 인터페이스 지원	인증서 뷰어, 키 생성 및 관리 장치 등과 같은 사용자 인터페이스를 제공해 줌으로써 복잡한 키 쌍 생성 및 인증서 요청과 같은 처리 절차를 자동으로 수행해 줌.
표준 호환성	FIPS , ISO, IETF, ANSI 등 국제 표준화 기구 및 산업표준화 기구에서 표준화되거나 국제적으로 공인된 암호 알고리즘들을 지원하고 있음
다양한 범위의 암호 응용 및 서비스 지원	인증서 관리 및 저장, S/MIME, 암호 디바이스(하드웨어 가속기, PKCS #11 디바이스, 스마트 카드 등)와 같은 다양한 응용에 대한 지원을 해줌.

* 한국정보보호진흥원(KISA) ({hsju, eonkyung, skim}@kisa.or.kr)

암호라이브러리 제품들은 사용자가 선택한 다양한 암호알고리즘, 키 길이선택과 같은 매개변수를 지원하는 등 다양한 보안 서비스를 제공한다. 또한, 암호라이브러리 제품들은 다양한 범위의 상위레벨의 응용 X.509인증서, OCSP, SMIME, PKCS#15 등을 지원해주고 있다. 이러한 암호라이브러리 제품들의 특징을 정리하면 표 2와 같다.

본 고의 구성은 다음과 같다. 2절에서는 암호 API에 대한 국내 외 개발현황을 살펴보고 3절에서는 암호라이브러리의 제품별 개발현황을 살펴본다. 4절에서는 암호라이브러리 제품들을 난수생성기와 공개키 암호알고리즘의 성능을 위주로 비교정리하기로 한다. 마지막으로 5절에서 결론을 내린다.

Ⅱ. 암호 API

본 절에서는 암호라이브러리에서 기반이 되는 암호 API의 현황을 알아본다.

1. 암호API 현황

최근까지 널리 알려진 암호 API 관련 기술로는 IETF의 GSS-API, X/OPEN의 GCS-API, MicroSoft (MS)사의 CryptoAPI, RSA사의 Crytpoki, 인텔에서 제안한 CDSA 등이 있다. 1990년대 초반까지 괄목할 만한 정보보호 API가 없던 때에 1992년 Linn이 IETF에 범용 암호 API로 GSS-API를 제안하였으며 이 API는 시스템 내부의 보안서비스에 접근하는 인터페이스를 이용하기 위해 시도되는 다수의 개발활동으로 주목을 받고 있었다. GSS-API는 2000년 1월에 인터넷 표준 RFC2743으로 작성되었으며 Cybersafe, IBM, Bull, Bell 등 회사들의 제품에 적용되고 있다.

GCS API의 개발자 X/OPEN[1]은 NIST, NSA, RSA, Fisher international, Trusted Information Systems, Hewlett Packerd 등으로 구성된 그룹으로 안전하고 신뢰성 있는 정보기술 하부구조를 개발하고 구현하는데 있어서 발생하는 여러 가지 문제점들에 대한 해답을 얻고자 설립되었다. 하지만 1997년 이후 지속적인 연구진행 상황이 발표되고 있지 않으며 관련된 설계 기본개념이 인텔의 CDSA로 발전된 것으로 보인다.

1) X/OPEN은 1996년에 The Open Group(TOG)로 합병됨

RSA의 Cryptoki는 RSA사의 공개키 암호표준인 PKCS#11로서 어플리케이션이 암호 토큰과 함께 동작하는 요구를 지원해 주기 위한 인터페이스로 개발되었다. 현재 CDSA와 GSS-API 등의 하부구조로 운용되는 방안이 연구되고 있으며 CDSA와 인터페이스 함수, 서비스 구조 등을 비교연구하고 있다.

Microsoft(MS)사의 CryptoAPI는 MS에서 지원하는 윈도우 운영체계의 일부로 제공되는 어플리케이션 프로그램 인터페이스이다. 윈도우 환경에서 작동하는 응용 프로그램에 인증서, 인코딩, 그리고 암호 기능을 추가하여 개발이 가능토록 하고 있으며 기초적인 내용을 모르더라도 API함수를 이용하여 암호프로그램을 사용할 수 있도록 되어 있다. 또한 안전성을 획득하기 위하여 NIST의 FIPS 140-1의 검증을 통과하였다.

〔표 3〕 국외 암호 API 표준화 현황〔17〕

암호 API	개발자	표준	비고
GSS-API	IETF	· RFC 2743 : Generic Security Service Application Program Interface Version 2, Update 1, 2000.1 · RFC 2744 :Generic Security Service API Version 2 : C-bindings · RFC 2853 : Generic Security Service API Version 2 : Java Bindings	-
GCS-API	The Open Group (미국)	· ISBN 1-85912-195-0, Open Group Document P442, 1996. 6	CDSA로 발전
Crypto API	MS (미국)	-	FIPS 140-1 승인
Crypoki (PKCS #11)	RSA (미국)	· PKCS#11 : Cryptographic Token Interface Standard (Technical Standard), 1995. 4	-
CDSA CSSM API	Intel (미국)	· C914(Technical Standard) : Common Security - CDSA and CSSM, Version 2 (C914), 2000.5	-

※ TOG(The Open Group)은 HP(Hewlett Packard), IBM(International Business Machines Corporation), NIST, NSA, RSA, OpenVision, Trusted Information Systems(Inc), ICL(International computers Limited) 등으로 구성됨

마지막으로 인텔의 CDSA는 1999년에 "CDSA Version 2 Technical Standard (C902)" 를 발표하였다. 그 후 2000년 5월에 애플과 인텔의 상세한 구현작업으로 교정된 내용이 포함된 "Common Security : CDSA and CSSM, Version 2" (C914)를 발표하였다. CDSA는 IBM, Apple, Compaq, Hewlett-Packard, AT&T, Motorola 등 시장을 선점하고 있는 보안 업체들에 의해 연구되어 자기 업체의 주요제품들에 적용되고 있다. 이런 이유로 CDSA가 향후 암호 라이브러리 및 암호 응용 및 서비스 사업에 광범위하게 사용될 것으로 판단된다.

2. 국외 암호API 평가 기준

암호API를 분석하는 평가 기준은 7가지로 나누어 볼 수 있으며 이 기준은 암호API들의 특징을 잘 나타내고 있다. NSA[13]의 암호API 평가기준을 설명하면 다음과 같다.

- 알고리즘 독립성 : 암호서비스를 제공하기 위해 특정한 알고리즘을 사용하도록 명시하지 않는다는 것을 의미하며 이런 성질은 암호API들의 대표적인 특징이다. 따라서 암호API는 현재와 향후 제시될 암호 알고리즘에 대한 광범위한 선택을 수용하여야 한다.
- 어플리케이션의 독립성 : 어플리케이션 독립성이란 임의의 어플리케이션을 설계할 때 특정 암호 API가 똑같은 역할을 할 수 있는 특성으로 정의될 수 있다. 즉 암호API는 현재 작성되고 있는 다양한 어플리케이션 뿐 아니라 향후 제시될 어플리케이션에도 암호 서비스를 제공할 수 있어야 한다.
- 암호모듈의 독립성[2] : 암호 서비스를 제공하는데 있어서 다른 부품의 기능 및 작업방식에 상관없이 암호 모듈을 사용할 수 있다는 것을 의미한다. 어플리케이션은 기반이 되는 암호 구현에 대해서 상세하게 알 필요가 없다. 예를 들면, 어플리케이션은 암호가 하드웨어로 제공되든지, 소프트웨어로 제공되는지에 대해 알 필요가 없다. 암호모듈의 독립성은 여러 개의 암호모듈을 사용하기 위한 기초적인 특성이다.
- 암호학적 지식의 정도 : 암호학적 지식이란 어플리케이션 개발자의 암호에 대한 지식의 양을 의미한다. 대부분 어플리케이션의 목표는 개발자로부터 최소한의 암호학적인 지식을 요구한다. 암호 API는 암호학에 대한 지식을 거의 요구하지 않는 것에서부터 상당한 수준의 숙련도를 요구하는 것에 이르기까지 다양하다. 그래서 암호API 들은 반드시 암호학적인 지식이 있는 어플리케이션과 암호학적인 지식이 없는 어플리케이션 모두를 지원할 수 있어야 한다.
- 모듈설계와 보조 서비스 : 모듈설계란 각기 특정한 목적이 있는 보안 서비스들을 제공할 수 있도록 함께 사용되는 암호API를 단위별로 그룹화하여 분리하는 것을 말한다. 보조 서비스는 암호 서비스의 목적에 따라 사용되는 지원 서비스이다. 이것은 키의 생명주기 관리, 암호 모듈 검증, 사용자 인증, 인증서 관리, 쿼리 능력 그리고 세션의 설정/해제 능력 등을 포함하고있다.
- 안전한 프로그래밍 : 안전한 프로그래밍은 프로그래머에 의해 초래되는 보안 사고를 방지하기 위해 취해지는 단계를 말한다. 안전한 프로그래밍의 개념은 일관성 있는 이름 부여 습관(consistent naming conventions), 정보 감춤(information hiding) 그리고 쉬운 사용(ease of use)이 있다. 일관성 있는 이름 부여 습관은 프로그래머로 하여금 각 절차의 목적과 사용에 대한 잘못된 이해를 줄일 수 있다. 또한 부주의로 인한 정보의 유출을 방지하기 위해 정보 감춤기능이 제공된다. 쉬운 사용은 암호API의 암호연산을 매개변수 형태로 만들거나 일관성 있게 순서를 주는 등 많은 일을 해주며 프로그래머가 실수할 확률을 줄여준다. 이 기능은 암호 비전문가에게 프로그램을 관리할 수 있게 하는데 있어 매우 중요하다.
- 보안 경계 : 보안에 관계된 정보가 신뢰할 수 있는 기반에서 신뢰할 수 없는 어플리케이션에 누출되는 것을 방지하는 경계를 의미한다. 암호API들이 보안 범주를 위반하는 창으로 사용되어서는 안 된다. 기밀성이 보존 되야 하는

2) 모듈의 독립성이란 하나의 제품을 구성하는 각각의 모듈이 다른 부품의 기능 및 작업 방식에 상관없이 주어진 업무를 처리할 수 있는 정도를 말함. 즉, 다른 모듈과의 연관성이 없을수록 모듈의 독립성은 높아짐.

암호 데이터에 대한 접근을 제한하여야 하며, 암호API를 벗어나는 정보가 전달되어서는 안 된다.

표 4는 NSA의 평가기준에 따라 앞에서 제시된 암호API를 평가한 비교자료이다.

〔표 4〕 암호API의 비교분석[13][17]

비교기준		GSS API	GCS API	Crypto API	Cryptoki (PKCS#11)	CDSA CSSM-API
알고리즘 독립성		Y	Y	Y	Y	Y
어플리케이션의 독립성		Y	Y	Y	Y	Y
암호모듈의 독립성		Y	Y	Y	Y	Y
암호학적 지식의 정도		N	Y	Y	Y	Y
설계와 보조 서비스	키 생명주기 관리	N	Y	N	N	Y
	암호모듈 검증	N	Y	Y	N	Y
	사용자 인증	Y	Y	Y	Y	Y
	인증서 관리	Some	Y	N	N	Y
	쿼리 능력	N	Y	Y	Y	Y
	설정/해제 능력	Y	Y	Y	Y	Y
안전한 프로그래밍(※) 〔가중치 : 1(불안전)~5(안전)〕		5	2	2	2	2
보안 경계		Y	Y	Y	Y	Y

※ 암호API들의 안전한 프로그래밍의 가중치는 일관성 있는 이름 부여 습관, 정보감춤기능, 쉬운 사용이라는 기준에 따라 NSA[13]가 측정한 결과임.

Ⅲ. 암호라이브러리

본 절에서는 국외의 암호라이브러리 제품들 중 RSA의 BSAFE와 공개된 암호라이브러리들의 제품 현황 및 지원 암호알고리즘들을 위주로 조사 비교한다.

1. RSA사의 암호라이브러리

〔표 5〕 RSA사의 BSAFE 암호라이브러리(1)

회사 (나라)	암호 라이브러리		지원알고리즘				
			난수 생성기	대칭키	해쉬함수	공개키	CSS
RSA (미국)	BSAFE	Crypto-C	Intel RNG	DES 3DES RC2 RC4 RC5 RC6 AES	MD2 MD5 SHA1	RSA DSA DH ECC	PKCS #1 PKCS #5 PKCS #8 PKCS #11
		Crypto-J			MD2 MD5 HAMC SHA1,2 RIPE MD160	RSA DSA DH	PKCS #1 PKCS #5 PKCS #12 PKCS #8 PKCS #11
		Crypto-C Micro Edition			MD2 MD5 SHA1,2	RSA DSA DH	PKCS #1

〔표 6〕 RSA사의 BSAFE 암호라이브러리(2)

회사 (나라)	암호 라이브러리		성능	지원OS	API	비고
RSA (미국)	BSAFE	Crypto-C	RSA Multi prime 기술	Windows 95,98,NT,2000,XP Sun/Solaris HP-UX Linux	PKCS#11 BHAP1	FIPS 140-1 level1 검증
		Crypto-J				
		Crypto-C Micro Edition		JDK1.1.8, 1.2,1.3,1.4	PKCS#11 Java Security API	
				Window CE,Palm, Linux		

※ CSS(Cryptographic Syntax Services) : PKCS #1 RSA formatting, PKCS#1 OAEP, PKCS #5, #12 PBE, PKCS #8 private key formatting, PKCS#11 hardware 지원을 나타냄.

2. 공개된 암호라이브러리

국외의 공개된 암호라이브러리는 특허로 등록된 몇 개의 암호알고리즘에 대한 소유권만을 제외하곤 비 상업용 및 상업용으로 무료 사용이 가능하였다.

하지만 일부 몇 개는 수정 및 배포를 할 때는 저작권자의 허가를 받아야 하는 것으로 조사되었다.

〔표 7〕 국외의 공개된 암호라이브러리 현황(1)

공개된 암호 라이브러리	소유권자	지원알고리즘	언어	지원 OS	비고
Crypto ++	Wei Dai (미국)	부록 표16	Borland C++ MS VC (4.0/4.1/5.0) gcc 2.7.2	UNIX WIN32 MSDOS MsOS BeOS	사용과 수정,배포에 대한 요금을 지불하지 않아도 됨. 하지만 미국으로부터 수출될 때 미국정부의 licence가 요구될 수 있음. 소프트웨어 중 어떤 부분은 특허에 의해 보호될 수 있음.
RSAeuro	J.S.A. kapp Reaper Technologies (영국)	부록 표16	ANSI C gcc	Unix	비상업용을 전제로 무료 사용가능. 상업용에 대한 licence는 J.S.A.Kapp에 연락해야 함.
SSLeay	Eric Young (미국)	부록 표16	C gcc	Solaris 2 SunOS LINUX Win32 DGUX OSF1 alpha	사용과 수정 분배가 무료, ANS. 1지원
Cryptix	David Hopwood 와 Raif Naffah (미국)	부록 표16	Java Perl	Win95 NT Solaris Linux Irix	무료사용가능 소스코드 배포는 저작권 지침을 포함해야 함

〔표 8〕 국외의 공개된 암호라이브러리 현황(2)

Cryptlib	Peter Gut mann (뉴질랜드)	부록 표16	C	Unix,Dos, Window	상업용 비상업용 상관없이 무료사용가능,배포 허가 안됨, 소스변경시 저자에게 연락
SSLave	Phaos Technology copora tion (미국)	부록 표17	Java	Windows 95/98/NT Solaris HP/UX AIX OS/390 Linux Licensing	비상업용을 전제로 사용허가됨, 사용에도 특허권에 대한 제약이 있음.
LibTom Crypt	Tom st Dennis (독일)	부록 표17	C gcc	-	상업용 비상업용 상관없이 분배, 수정 사용이 가능 Serpent/AES는 Braian Gladman에 의해 저작권이 있음.

Ⅳ. 암호라이브러리 제품 비교

〔14〕〔15〕〔16〕의 문서를 분석한 결과 암호라이브러리를 이용하고 있는 암호시스템에서의 공격가능성이 있는 부분은 크게 난수 생성기, 짧은 키의 사용, 구현상의 오류, 프로토콜 상의 오류로 구분할 수 있었다. 이에 따라 본 고에서는 앞 절에서 정리된 암호라이브러리 제품들을 난수생성기와 성능으로 비교 정리하고자 한다.

1. 난수생성기 비교

난수는 대칭키 암호의 세션키 생성, 공개키 암호 혹은 전자서명의 시스템 매개변수의 생성, SSL과 같은 보안 통신 프로토콜에서 사용되는 난수 등 많은 암호 시스템 전반에 걸쳐 사용되고 있다. 이와 같이 난수 생성기가 많은 응용 환경에 필요한 요소임에도 불구하고, 실제로는 안전하지 못한 난수 생성기들이 사용되어 오곤 했다. 예를 들어 취약한 난수 생성기의 취약성으로 인해 Netscape 브라우저 내의 암호화 알고리즘, Kerberos VS, MIT-MAGIC-COOKIE 등이 공격당하였다. 여기서는 앞에서 조사된 암호라이브러리의 의사난수 생성기를 세부적으로 조사·비교하고자 한다. 암호라이브러리 제품들에서 사용되고 있는 의사난수생성기들의 현황을 정리하면 표 9와 같다.

〔표 9〕 의사난수 생성기 현황

(의사)난수 생성기	특성	비고
Applied Cryptography 생성기〔22〕	MD5해쉬함수를 이용하며 카운터 값과 결합해 16 바이트열을 출력함. 안전성은 해쉬함수에 의존함.	연관키 공격(related attack)에 영향을 받음
ANSI X9.17 생성기	3DES를 이용하며 비도는 3DES에 사용된 킷값을 보호하는 사용자의 능력에 의존함.	-
PGP2.x 생성기	순수난수(True Random)를 생성하기 위해 사용자로부터 입력받은 키보드 정보와 각종 잡음정보(시스템 시간, Performance Counter등)를 이용, 이 값을 Seed로 입력받아 ANSI X9.17의사난수 생성알고리즘을 이용함.	-
Intel 생성기	인텔 펜티엄 Ⅲ CPU에 내장된 칩셋을 이용, 레지스터에 있는 열소음에서 표본을 취한 값을 seed로 사용함.	-

(의사)난수 생성기	특성	비고
Cryptlib 생성기	MD5 혹은 SHA1에 기반. 메모리 블록을 순환 버퍼로 간주하고 그 버퍼에 있는 데이터를 해쉬함수로 처리함.	-
SSLeay/ OpenSSL 생성기	전역변수 128비트 길이의 MD버퍼, 1024바이트의 State변수를 유지, 전역변수를 초기화·갱신하는 RAND_Seed모듈, 의사난수를 추출하는 RAND_GetBytes모듈로 구성됨.[13]	-
RSA -Euro	비교적 엔트로피가 낮은 시스템 시간과 시스템 클럭을 사용하지만 많은 양의 데이터를 이용하여 status를 생성함.[13]	시스템 시간과 클럭만을 사용하는 것이 단점[13]
Capston /Forezza 생성기	다양한 소스와 메커니즘을 사용해 한 메커니즘이 실패해도 나머지 메커니즘들이 적당한 여분의 안전성을 제공하도록 설계됨. ANSIX9.17과 같은 생성기를 포함하는데 3DES 대신 Skipjack을 사용하고 입력은 X9.17의 시간값 대신 물리적인 잡음원을 취함.	-

표 10에서는 표 9에서 조사된 의사난수 생성기들의 안전성을 비교해본다. 성능은 3072비트 길이의 의사난수를 출력하는데 필요한 MD5 해쉬 연산수를 나타낸 것이며, α는 Intel RNG로부터 seed를 얻는데 소요되는 시간으로 조사되었다.

[표 10] 의사난수 생성방법들에 대한 암호라이브러리 비교분석[13]

	BSAFE	RSA Euro	Intel RNG를 이용한 방법	SSLeay	PGP
Seed값에 대한 entropy	높음	보통	높음	보통	비교적 높음
성능	48+α	86	96+96α	101	384

표에서 알수 있듯이 하드웨어 기반의 Intel RNG를 이용한 방법이 안전성 면에서 가장 우수하다는 것을 알수 있으며, 하드웨어의 지원을 받지 못할 경우에는 PGP에서 사용하고 있는 의사난수 생성 방법이 우수함을 알 수 있다.[13] 성능 면에서는 초기에 Intel RNG로부터 seed를 한번 얻고 해쉬연산을 48회 수행하는 BSAFE가 가장 우수하며, MD5transform 연산을 384회 수행하는 PGP가 가장 떨어짐을 알 수 있다.[13]

2. 성능 비교

RSA 시큐리티사의 멀티프라임 고속화 기술[3]은 기존의 RSA 공개키 암호알고리즘 보다 복호화 속도가 약 2배 빠른 기법으로 무선 통신 환경에서 고속으로 사용할 수 있도록 하는 기술이다. 다음 표들은 공개된 암호라이브러리 중 Crypto++과 Libtomb Crypt의 벤치마킹 자료와 RSA사의 멀티프라임 기법에 대한 조사자료[19]로 비교·조사한 자료이다.

[표 11] BSAFE : 공개키 암호연산에 대한 속도
(단위 : msec)

분류		BSAFE	구현환경
RSA Encrypt (1024bit)	n=pq	0.6 [21] 미만	• 200MHz Pentium Pro
	n=pqr	0.6 미만(추정)	• GMP bignum 라이브러리를 사용 • e=65537로 취함
	$n=p^2q$	0.6 미만(추정)	• GMP bignum 라이브러리를 사용 • e=65537로 취함
RSA Decrypt (1024bit)	n=pq	43 [21] 미만	• 200MHz Pentium Pro
	n=pqr	24 [19] 미만(추정)	• GMP bignum 라이브러리를 사용
	$n=p^2q$	18 [19] 미만(추정)	• GMP bignum 라이브러리를 사용

[표 12] Crypto++ : 공개키 암호연산에 대한 속도
(단위 : msec)

분류		Crypto++ 4.0	구현환경
RSA Encrypt (1024bit)	n=pq	0.32	• C++ 구현 • Microsoft Visual C++6.0 SP4 사용
RSA Decrypt (1024bit)	n=pq	10.23	• Celeron 850MHz processor 사용 • 다정도 연산을 위해 어셈블리 루틴 사용

3) ※ NTT의 $N=p^2q$ 형태의 멀티프라임 고속화 기술은 기존의 RSA보다 2.3배 빠름[19]. KISA는 이 기술을 일반화에서 최적의 효율성을 줄 수 있는 ($n=p^rq^s$ 형태)경우를 제시함.[23]

〔표 13〕 LibTomCrypt : 공개키 암호연산에 대한 속도
(단위 : msec)

분류		Libtomb Crypt	구현환경
RSA Encrypt (1024bit)	n=pq	10 미만	• MPI bignum code사용 • e=65537로 취함 • 1.2Ghz(thunderbird) processor 사용
RSA Decrypt (1024bit) (CRT사용)	n=pq	120 미만 (70 미만)	

※ MPI bignum code는 공개되서 자유롭게 이용할 수 있는 다정도 연산 라이브러리로 Michael Fromberger 이 저작권을 갖고 있음.

V. 국내 현황

국내의 경우, 암호API에 대한 표준화 연구가 수행된 적이 없으며, 몇몇 기관들이 자체적으로 조사 분석하여 일부 제품으로 개발하여 판매하고 있다. 또한, 암호라이브러리의 개발은 대부분 국외의 인터넷 상의 오픈 소스코드를 기반으로 이루어져 왔으므로, 자체 기술력이 극히 미약하다. 국내는 국외의 공개 암호라이브러리처럼 다양한 암호 프리미티브에 대한 구현소스, 암호모듈을 쉽게 이용할 수 있는 API 등 전반적인 암호라이브러리 소스코드를 공개하고 있는 곳은 없다. 표 11은 국내 암호라이브러리의 현황을 홈페이지에 제시된 자료를 토대로 작성한 것이다.

〔표 14〕 국내 암호라이브러리 현황(1)

회사 (나라)	암호 라이브러리	성능	지원OS	API	비고
퓨처	Cryp Tool (C)	-	-	RSA사의 BSAFE와 PKCS#11 CryptoKi를 참고로 하여 설계	-
이니텍	INI Crypto (C)	-	-	-	-
STI	J/LOCK (Java)	-	-	JDK1.2	Sun Microsystems의 인증획득
장미디어	CEAL98 (Java)	-	-	-	-
트러스컴	trusCrypt (C)	-	-	-	-

〔표 15〕 국내 암호라이브러리 현황(1)

회사 (나라)	암호 라이브러리	지원알고리즘			
		난수 생성기	대칭키	해쉬 함수	공개키
퓨처	Cryp Tool (C)	Spring (자체 개발)	DES DES3 DESX BLOWFISH CAST128 IDEA SAFER RC2,5,6 SEED CRYPTON CAST256 TWOFISH RIJNDAEL MARS SERPENT	MD5 SHA1 SHA256 SHA384 SHA512 RMD128 RMD160 RMD256 RMD320 HAS160 TIGER	RSA DSA ECDSA KCDSA EC-KCDSA
이니텍	INI Crypto (C)	-	SEED DES 3DES IDEA RC2,4,5 BlowFish AES	MD5 SHA1 HAS160	RSA DSA ECDSA KCDSA ECKCDSA DH ECDH
STI	J/LOCK (Java)	-	SEED DES 3DES Blowfish CAST RC2,4,5 SAFER	MD계열 SHA1 RIPEMD 128, 160 HAS160 HAVAL	RSA RSA OAEP DSA ECDSA KCDSA DH
장미디어	CEAL98 (Java)	LFSR 이용한 RNG	DES DESX RC2,4,5 IDEA Blowfish SEED SAFER	MD계열 SHA1 RIPEMD 128,160	RSA ElGamal KCDSA DSA DH
트러스컴	trus Crypt (C)	-	DES 3DES DESX IDEA RC2,4,5 SEED AES	MD2 MD4 MD5 SHA-1 HAS160	RSA, DSA, KCDSA ECDSA DH ECDH

〔표 16〕 공개된 암호라이브러리의 지원알고리즘(1)

암호 라이브러리	의사난수 생성기	압축	대칭키	해쉬 함수	공개키		
					암/복호	전자서명	키합의
Croto++	ANSI X9.17, PGP's RandPool	gzip (RFC 1952) zlib (RFC 1950)	Blowfish, CAST128 DES DESX 3DES Diamond2 GOST IDEA Lubyrackoff RC2,5 SAFER MARS RIJNDAEL SHARK SQUARE SERPENT TEA BBS(BlumBlumShub) WAKE	CRC MD계열 SHA HAVAL HMAC RIPEMD 160 TIGER	RSA EGamal LUC Rabin BlumGoldwassre (512,1024,2048 지원) ECIES	RSA RABIN LUC NR DSA RW (512,1024, 2048 지원) ECNR	XTR-DH DH MQV ECDHC ECMQVC
RSA EURO	자체개발	×	DES DESX 3DES	MD계열 SHA	RSA	RSA	DH
SSLeay	자체개발	-	DES RC2,4,5 Blowfish IDEA CAST	MD계열 SHA SHA-1 RIPEMD MDC2	RSA,	RSA DSA	DH
Cryptlib	자체개발	×	DES 3DES Blowfish IDEA SAFER RC2,4,5 CAST128	MD계열 SHA MDC2 RIPEMD160	RSA ElGamal	DSA	DH
Cryptix	-	×	Blowfish CAST5 LOKI91 RIJNDAEL DES 3DES IDEA LOKI97 RC2,4,5 SAFER SPEED SQUARE	HAVAL MD계열 RIPEMD 128,160 SHA1	RSA ElGamal	RSA ElGamal	×

〔표 17〕 공개된 암호라이브러리의 지원알고리즘(2)

암호 라이브러리	의사난수 생성기	압축	대칭키	해쉬 함수	공개키		
					암/복호	전자서명	키합의
SSLave	자체개발	×	DES 3DES RC4 ARC4	MD5 SHA	RSA	×	DH
LibTom Crypt	자체개발	Zlib gzip	Blowfish X-Tea RC5 RC6 SAFER SERPENT RIJNDAEL TWOFISH	SHA (256,384,512) TIGER192 SHA1 MD5	RSA ECC	×	DH

※ • GOST(28147-89) : 러시아 표준 블록암호 알고리즘(256비트 키)
• SAFER: 블록암호 IDEA의 개발자중의 한 명인 J. L. Massey가 개발한 128비트 블록암호알고리즘
• LOKI91: 1990년에 L. Brown, J. Pieprzyk 와 J. Seberry 가 개발한 64비트 블록암호알고리즘
• SQUARE : 1997년 Joan Daemen 과 Vincent Rijmen 에 의해 개발된 128비트 블록암호알고리즘
• SPEED : 1997년에 Y.Zheng에 의해 개발된 RC5 방식의 블록암호 알고리즘(다양한 키길이 지원)
• TEA(Tiny Encryption Algorithm) : David Wheeler 와 Roger Needham가 개발한 128비트 블럭암호알고리즘
• Lubyrackoff : 해쉬함수 기반의 블록암호 알고리즘
• 스트림 암호 : RC4, ARC4, WAKE, BBS(BlumBlumShub)
• NR : Nyberg Ruppel, RW : Rabin-Willams

Ⅵ. 결 론

본 고에서는 암호API를 NSA의 문서〔18〕를 기반으로 국외 표준화 현황 및 평가방법을 알아보았다. 또한 암호 라이브러리의 국내외 현황을 지원알고리즘, 성능, 지원OS 등으로 조사하였고 암호라이브러리에서 공격에 가장 취약하다고 알려져 있는 난수생성기를 비교하였으며, 공개키 암호연산의 속도로 성능을 비교·정리하였다.

참고문헌

〔1〕 RSAEURO Library, available at http://www.rsaeuro.com/products/RSAEuro
〔2〕 SSLeay, "SSLeay Documentation" http://www.columbia.edu/~ariel/ssleay/
〔3〕 LibTomCryptLibrary, available at http://libtomcrypt.sunsite.dk/#performance
〔4〕 Cryptlib Library, available at http://www.cs.auckland.ac.nz/~pgut001/cryptlib/
〔5〕 Cryptix Library, available at http://www.cryptix.org/products/index.html
〔6〕 SSLave Library, available at http://www.phaos.com/e_security/prod_ssl.html
〔7〕 PGP, http://www.pgpi.org/
〔8〕 INITECH, http://www.initech.com
〔9〕 Truscom Library, available at http://www.truscom.com/products/pki_t_crypt1.html
〔10〕 퓨처, http://www.future.co.kr
〔11〕 장미디어, http://crypto.jmi.co.kr
〔12〕 STI, http://www.stitec.com/product/
〔13〕 강명희, 임채훈, "소프트웨어 방식의 의사난수 생성방법에 대한 고찰", (주)퓨쳐시스템, 암호체계센타
〔14〕 "Snake Oil Warning Signs : Encryption Software to Avoid"
〔15〕 "On cryptosystems untrustworthiness"
〔16〕 "Security in the Real World : How to Evaluate Security Technology"
〔17〕 한국정보보호센터, "보안 API표준화 연구", 2000.12
〔18〕 "Security Service API : Cryptographic

API Recommendation Updated and Abridged Edition", NSA Cross Organiztion CAPI Team, The National Security Agency, July 25, 1997.
[19] CrytoBytes, "Fast Variants of RSA", 2002
[20] 한국정보보호진흥원(KISA), "모듈러스 $N=p^rq^s$ 를 이용한 RSA 공개키 고속화 기술", http://www.kisa.or.kr
[21] CryptoBytes, 1999, "Performance Comparison of Public-Key", 1998
[22] Bruce Schneier, "Applied Crypto graphy (second edition)", John Wiley and Sons, 1996.
[23] Seongan Lim, Seungjoo kim, Ikkwon Yie and Hongsub Lee, "A Generalized Tagagi-Cryptosystem with a modulus of the form prqs", Proc of Indocrypt 2000, International Conference on Cryptology in India, Springer-Verlag, LNCS 1977, Calcutta, India, December 10-13 2000, pp.283-294.

〈著者紹介〉

주 학 수 (Hak-Soo Ju)
1997년 8월 : 고려대학교 수학과 이학사
1999년 8월 : 고려대학교 수학과 이학석사
2001년 8월 : 고려대학교 수학과 박사과정 수료
2001년 9월~현재 : 한국정보보호진흥원(KISA) 연구원

이 언 경 (Eonkyung Lee)
1992년 2월 : KAIST 수학과 이학사
1994년 2월 : KAIST 수학과 이학석사 (암호학 전공)
2001년 8월 : KAIST 수학과 이학박사 (암호학 전공)
1994년 2월~1997년 8월 : 한국전자통신연구원(ETRI) 연구원
2001년 10월~현재 : 한국정보보호진흥원(KISA) 선임연구원

김 승 주(Seung-Joo Kim) 종신회원
1994년 2월 : 성균관대학교 정보공학과 공학사
1996년 2월 : 성균관대학교 대학원 정보공학과 공학석사 (암호학 전공)
1999년 2월 : 성균관대학교 대학원 정보공학과 공학박사 (암호학 전공)
1998년 12월~현재 : 한국정보보호진흥원(KISA) 암호기술팀장
2000년 6월~현재 : 한국정보통신기술협회(TTA) 정보통신기술위원회 암호기술연구반 의장
2002년 4월~현재 : 한국정보통신기술협회 국제 표준화 전문가