Journal of the Korea Academia-Industrial cooperation Society
Vol. 15, No. 8 pp. 5294-5302, 2014

http://dx.doi.org/10.5762/KAIS.2014.15.8.5294
ISSN 1975-4701 / eISSN 2288-4688

OpenSSL을 이용한 키쌍(공개키·개인키) 충돌율 분석

이광형[1*], 박정효[2], 전문석[2]
[1]서일대학교 인터넷정보과, [2]숭실대학교 컴퓨터공학과

Key-pair(Public key, Private key) conflict analysis using OpenSSL

Kwang-Hyoung Lee[1*], Jeong-Hyo Park[2], Moon-Seog Jun[2]
[1]**Department of Internet Information, Seoil University**
[2]**Department of Computer Science, SoongSil University**

요 약 공개키 기반 기술의 발전은 전자정부, 전자금융, 전자결제 등 다양한 서비스를 가능하게 하였으며, 완벽한 안전성을 가지고 있는 것으로 평가된다. 하지만, 최근 허트블리드 버그 등 공개키 기반 이용 기술에 대한 취약점이 지속적으로 발견되고 있다. 본 논문에서는 공개키 기반구조의 안전성 및 신뢰성을 검증하기 위해, OpenSSL을 이용하여 키쌍의 충돌율을 분석하였다. 실험은 OpenSSL을 이용하여 5개의 사설인증기관을 생성하고, 각 사설인증기관에서 200만개의 인증서를 생성해 총 1,000만개의 인증서를 생성하여 키쌍 충돌 여부를 분석하였다. 실험은 다음과 같은 과정으로 수행되었다. Openssl을 이용하여 5개의 사설인증기관 생성, 각 사설인증기관에서 200만개의 인증서를 생성, 총 1,000만개의 인증서를 생성하여 키쌍 충돌 여부를 분석하였다. 실험 결과 1,000만건 중 35,000건, 즉 0.35%의 확률로 공개키·개인키가 충돌을 발생하였다. 이는 전자상거래, 보안서버 등 다양한 분야에서 충분한 위협이 될 수 있는 요소이다. 향후에는 공개키 기반기술의 위협요소를 제거하기 위해 난수생성기, 큰 소수 선택 문제 등 깊이 있는 연구를 진행할 것이다.

Abstract The development of public-key-based technique that enables a variety of services(E-government, e-banking, e-payment, etc.) evaluated as having complete safety. On the other hand, vulnerabilities(e.g, heartbleed bug, etc.) are constantly being discovered. In this paper, a public key infrastructure to verify the safety and reliability, the collision rate using OpenSSL key pair was analyzed. the experiment was performed using the following procedure. Openssl was used to create five private certification agencies, and each of the private certificate authority certificates to create 2 million, generating a total of 10 million by the certificate of the key pair conflicts analysis. The results revealed 35,000 in 1 million, 0.35% chance of a public key, a private key conflict occurred. This is sufficient in various fields(E-payment, Security Server, etc.). A future public-key-based technique to remove the threat of a random number generator, large minority issues, in-depth study of selection will be needed.

Key Words : The public key based technique, RSA Cryptosystem, The key pair conflict analysis. The collision rate.

1. 서론

공개키 기반 기술의 발전은 인증서를 통한 다양한 서비스를 가능하게 하였으며 완벽한 안전성을 가지고 있다고 평가되고 있다. 하지만, 최근 공개키 기반 이용기술에 대한 취약점이 발견되고 있다[1,2].

2014년 3월 공인인증서 탈취 및 비밀번호 해킹 여부를 실험한 결과, 40여 초 만에 실제 비밀번호가 생성된 것으로 보도되었다. 이는 개인정보 유출로 보안에 대한 중요성이 강조되고 있는 가운데 온라인에서 자신을 증명하는 사실상 유일한 도구인 공인인증서가 보안에 취약한 것을 의미한다.

본 논문은 2014년도 서일대학교 학술연구비에 의해 연구되었음
*Corresponding Author : Kwang-Hyoung Lee(Seoil Univ.)
Tel: +82-2-490-7226 email: dreamace@seoil.ac.kr
Received July 22, 2014 Revised August 6, 2014 Accepted August 7, 2014

또한, 2014년 4월 하트블리드 버그에 보안인증서가 해킹되는 사고가 발생하였다. 이는 웹사이트 진위여부를 가리는 보안인증서가 해킹을 당하면 가짜 웹사이트를 구분할 수 없어 사용자들이 위조 웹사이트에 무방비로 노출되게 된다. 이러한 사고의 여파로서 개인정보 및 금융정보가 유출되어 2차로 금전적 피해가 발생할 수 있다.

인증서는 RSA 암호화 알고리즘을 이용하여 공개키 및 개인키를 생성한다. 그러나 공개키 기반 기술을 이용한 오픈소스 소프트웨어는 키쌍을 생성하는 난수 생성기의 안전성 문제와 큰 소수의 선택에 따른 문제를 내재하고 있다. 소수의 범위는 아주 크지만 일반적으로 인증서를 생성하기 위해 오픈소스 소프트웨어에서 사용되는 소수의 선택 범위는 제한적이다[17].

이와 더불어, 일반적으로 기관 및 기업에서 사용하는 오픈소스 소프트웨어는 공개키의 중복검사를 올바르게 수행하지 않고 있다.

위와 같은 이유로 인하여, 공개키 및 개인키의 충돌이 발생하고 있고, 이는 하나의 인증서로 다목적서비스에 사용하는 등 심각한 문제를 발생할 소지가 있다.

공개키 기반 이용기술이 갈수록 다양화 및 고도화됨에 따라 인증서를 이용한 서비스가 증가하고 있는 추세이다. 하지만 공개키 기반 이용기술에 대한 안전성 및 신뢰성 검증이 올바르게 이루어지고 있지 않아 향후 문제가 될 수 있다.

본 논문은 공개키 기반 구조에서 공개키와 개인키가 합치하는 특성을 이용하여, 키쌍의 충돌율을 분석하였다.

본 논문은 총 5개장으로 구성되어 있으며, 각 장의 내용은 다음과 같다. 제2장은 RSA 암호화 알고리즘 동작원리 등 관련 연구를 기술하였고, 제3장에서는 OpenSSL을 이용하여 인증서 생성시 키쌍 충돌 여부를 조사하는 테스트 환경을 설명하였다. 제4장에서는 OpenSSL 소프트웨어를 이용하여 공개키의 충돌율을 분석하였고,. 마지막으로 5장에서는 결론으로 향후의 연구를 기술하였다.

2. 관련연구

2.1 RSA 암호화

2.1.1 RSA 개요

가장 많이 사용되는 공개키 알고리즘으로 Rivest, Shamir, Adleman의 이름을 사용하여 만든 RSA 암호 시스템은 두 개의 지수 e와 d를 Key로 사용한다. 여기서 Key란, 메시지를 열고 잠그는 상수(Constant)를 의미한다. 일반적으로 많은 공개키 알고리즘의 공개키(Public Key)는 모두에게 알려져 있으며, 메시지를 암호화(Encryption) 하는데 쓰이며, 암호화된 메시지는 개인키(Private Key)를 가진 자만이 복호화(Decryption)하여 열어볼 수 있다. 그러나 RSA 알고리즘은 이러한 제약조건이 없으며 개인키로 암호화하여 공개키로 복호화 할 수 있다. 공개키 알고리즘은 누구나 어떤 메시지를 암호화화 하는 것이 가능하다, 그러나 복호화하여 확인할 수 있는 사람은 개인키를 소유하고 있는 사람만이 존재한다는 점에서 대칭키 알고리즘과 차이를 가진다. RSA 암호 시스템은 소인수분해의 난해함에 기반을 두며, 공개키만을 이용하여 개인키를 쉽게 분석할 수 없도록 디자인 되어 있다[6,7,16].

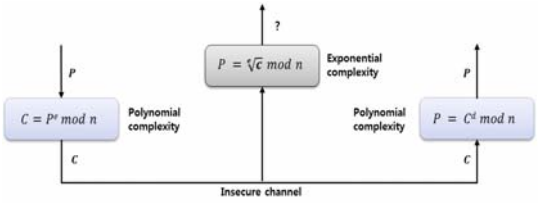
암호화와 복호화에서는 모듈로 지수계산을 사용한다. 모듈로 지수연산은 고속 지수연산 알고리즘을 이용할 경우 다항식 정도의 복잡도로 계산이 가능하다. 하지만 모듈로 로그는 모듈로 값을 소인수분해하는 것만큼 어렵다. 왜냐하면 아직까지 이를 수행하는데 다항식 정도의 복잡도를 갖는 알고리즘이 개발되지 않았기 때문이다. 다시 말해 다항식 수준의 복잡도를 갖는 계산으로 암호화를 및 복호화를 할 수 있지만 공격자는 모듈로 계산을 하여 암호문에 대하여 공개키 값을 얻기 힘들다[5,7,9].

2.1.2 RSA 암호화 기법의 원리

RSA 암호화 기법은 구현이 간단하며, 안전성이 높기 때문에 많이 사용된다. 블록단위로 암호화를 하며, 각 블록은 n(키값의 곱)보다 작은 바이너리 값으로 이루어져 있다. 블록사이즈는 $\log_2(n)$ 보다 작거나 같다. 암호화 방법은 $C = M^e \bmod n$ 방법으로 되며, 복호화는 $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$ 의 형태로 된다. 송신자와 수신자는 서로 n 값을 알고 있으며, 공개키는 $\{e,n\}$ 이며, 비밀 키는 $\{d,n\}$ 이 된다. 암호화 방법에 있어서 $M^e \bmod n$ 의 연산은 매우 쉽지만, 반대로 역 연산은 매우 어렵다. $\bmod$ 연산을 통해 수 없이 많은 값들이 M 값이 될 수 있기 때문이다[12].

역 연산이 어려운 이유도 e 와 d가 $\bmod \varphi(n)$ 에서 많은 inverse가 존재하기 때문이다. 이러한 수학적인 연산의 어려움(Discrete Logarithm Problem) 때문에 키 없이 암호문을 해독하기가 매우 어렵다. 다음 Fig. 1은 RSA

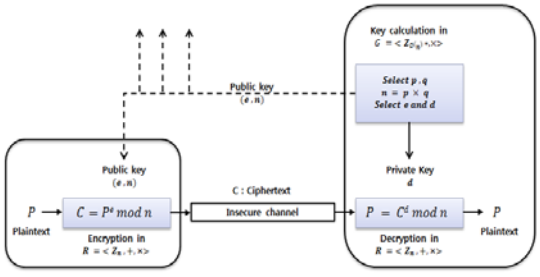
암호화 연산의 복잡도를 타나낸다[5,6].



[Fig. 1] RSA computation complexity

2.1.3 RSA 암·복호화 및 키생성

A와 B가 보안이 보장되어 있지 않은 환경에서 비밀 메시지를 주고받고 싶다고 가정하였을 때, B가 A에게 메시지를 전달하기 위해서는 A의 공개키가 필요하다. A는 Fig. 2와 같은 방법을 통해 공개키와 개인키를 생성한다.



[Fig. 2] RSA encryption, decryption and key generation

• RSA 암호화

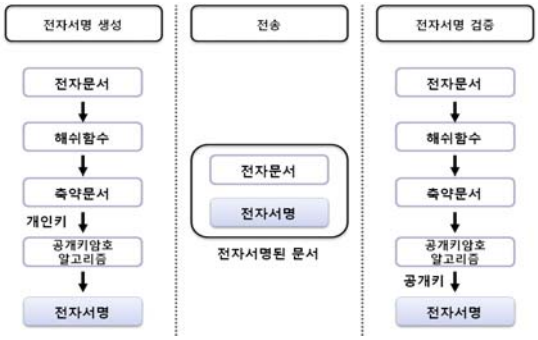
- A가 B에게 메시지를 보내고 싶을 때 A는 평문 메시지 P을 N보다 작은 숫자로 변환한다. A는 B의 공개키 $\{e,n\}$ 을 획득하고, 공개키를 이용하여 암호문 $C= P^e \bmod n$ 을 획득하여 B에게 전송한다.

• RSA 복호화

- B는 암호화된 메시지 C 를 A로부터 받았고 n과 개인키 d를 알고 있을 때 $P= C^d \bmod n$ 를 이용하여 암호문 C 를 복호화한다.

2.1.4 RSA 암호화 전자서명

공개키 암호화는 전자서명에 사용된다. 전자서명은 우리가 일상생활에서 신원을 확인하고 전자거래를 이용할 할 때 주민등록증, 인감 날인, 서명 등이 필요하듯이, 전자적으로 신원을 확실히 보장해주는 수단이 바로 전자서명(Digital Signature)이다. 즉, 전자서명은 인증서 형태로 발급되는 개인의 전자적 인감이며 서명이다. 또한 전자서명을 이용하여 어떤 사람이 문서를 작성했다는 증명할 수 있으며, 자신의 개인키를 이용하여 작성한 문서를 암호화하여 첨부하는 것이다. 암호화된 문서는 공개키에 의해서 복호화 되어 원문과 비교될 수 있다. 그러므로 전자서명을 사용하면 어떠한 사람이 서명하였다는 것을 증명할 수 있다[7,18-21]. Fig. 3은 공개키 암호화와 전자서명에 대한 그림이다.

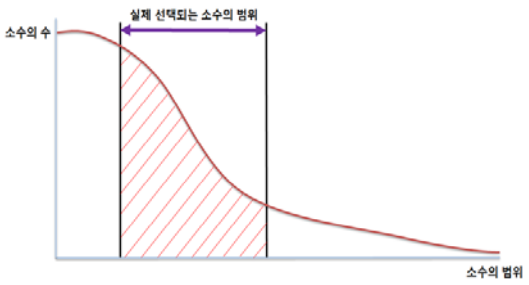


[Fig. 3] Electronic signature process

2.2 RSA 암호화의 문제점

2.2.1 소수의 제한성

공인인증서를 생성하기 위해 개인키와 공개키를 생성하는 과정에서 소수를 사용한다. 하지만 소수의 특성상 수가 커짐에 따라서 소수의 개수는 감소된다. RSA 암호화를 이용하여 개인키 및 공개키 생성 시 매우 작거나 큰 소수는 RSA에 키 생성 시 사용이 불가하다. 매우 작은 소수를 사용하면 소인수 분해가 가능하게 되어 개인키를 쉽게 추출할 수 있으며 너무 큰 소수를 사용하면 수학적 연산의 기반을 둔 RSA 암호 시스템을 계산하는데 많은 시간이 사용된다. 다음 Fig. 4는 소수의 범위가 커짐에 따라 소수의 수의 감소를 보여주며 인증서를 생성하는 소프트웨어에서 키 쌍을 만들기 위해 실제 사용되는 소수의 범위는 한정적이라는 것을 알 수 있다. 소수의 범위가 매우 크지만 제한 범위 안에서 소수가 선택되기 때문에, 많은 수의 인증서에서 공개키가 충돌이 나는 원인이 된다[1,17].



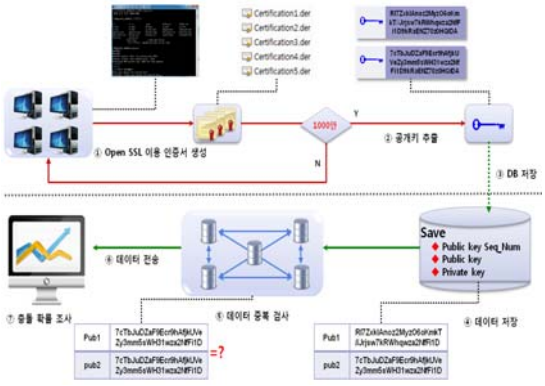
[Fig. 4] The limitations of prime

3. OpenSSL을 이용한 공개키 충돌 분석

본 논문에서는 오픈소스 소프트웨어인 OpenSSL을 이용하여 인증서 생성시 키쌍 충돌 여부를 조사하는 테스트 환경을 다음과 같이 구축하였다.

3.1 테스트 환경

공개키 충돌 여부를 분석하기 위한 테스트 환경의 구성은 다음과 같다.



[Fig. 5] The configuration of the test environment

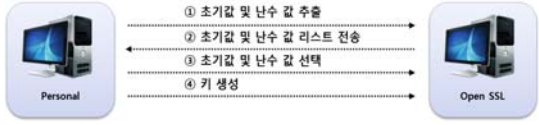
테스트를 위한 구축한 환경에서는 자체적으로 인증서를 생성하고 RSA 암호화를 적용할 수 있다. 또한 실제 인증서를 생성하는 인증서 트러스트 체인과 같아 다수의 컴퓨터를 이용할 수 있다.

오픈소스 소프트웨어인 OpenSSL을 이용하여 5개의 사설인증기관을 생성하고, 각 사설인증기관에서 200만개의 인증서를 생성해 총 1000만개의 인증서를 가지고 키쌍 충돌 여부를 분석하였다.

OpenSSL에서 인증서를 생성할 때 헤더에 붙는 의미없는 값들은 삭제하고, 데이터베이스에 순수 공개키 정보만을 수집하여, 충돌율을 조사하였다.

3.2 IV 및 난수값 수집

OpenSSL에서 제공하는 라이브러리를 이용하여 공개키 및 개인키를 추출하는 과정에서 순수한 공개키만을 얻기 위하여 키 값을 만드는 SEED 초기 값 및 난수를 고정시켜야 한다. 절차는 다음과 같다.



[Fig. 6] The Fixed procedure of IV and random number

OpenSSL을 이용하여 RSA 암호 알고리즘을 사용할 때 다음과 같은 함수가 사용된다.

[Table 1] the RSA key generation functions

Function	Description
① RSA_generate_key()	generate a key
② RAND_screen()	generate a random seed value
③ RSA_new()	create object for making key
④ RSA_check_key()	verify key validation

① RSA_generate_key 함수는 공개키 및 개인키를 생성한다.

• RSA_generate_key(int num, ussigned long e, void (*callback)(int, int, void *), void)

매개변수 num은 키의 길이를 말하며, e는 public exponent로 3, 17, 65537의 수가 사용되며 대개는 3을 사용하게 된다. callback 함수는 키 생성과정에 대한 feedback을 제공하는데 사용되는 함수이다.

② RAND_screen(void) 함수는 PRNG 값에 seed를 공급해준다.

• void RAND_seed(const void *buf, int num)

매개변수 buf는 PRNG(pseudo random number generator : 의사 난수 숫자 생성기)와 MIX되어질 값이고, num은 buf의 크기를 의미한다. 키를 만들기 위해서는 초기값이 필요하며, 이 초기값을 만들기 위해서 PRNG가 시드값을 기반으로 한 무작위 숫자 스트림을 만들어 낸다. 또한, PRNG는 이전 및 미래의 출력에 추측 저항성을 갖게 된다.

③ RSA_new() 함수는 RSA 자료구조를 메모리상에 확보하고, RSA_free() 함수는 확보된 자료구조를 메모리에서 제거한다.

• RSA_new(void)
• void RSA_free(RSA *rsa)

④ RSA_check_key() 함수는 해당키가 정당한 키를 가지고 있는지 검사한다.

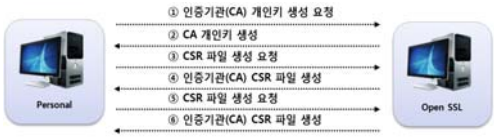
• RSA_check_key()
• int RSA_check_key(RSA *rsa)

해당하는 키(rsa)가 정당한 키를 가지고 있는지 검사를 하는 함수이다. return값이 1이면 validate 키를 가지고 있다는 것을 의미한다.

난수를 생성하는 과정의 안전성에 결함이 있다면 이는 암호 알고리즘 자체의 안전성에도 영향을 미치게 된다. RSA 암호 알고리즘에서 키를 생성하기 위한 소수 p와 q는 이 난수를 기반으로 생성하게 된다. 하지만 온전히 난수를 생성하는 것에 대해 사용하는 컴퓨터에 안전성을 맡기는 것은 안전하지 않다. 따라서 정확한 난수의 의존도와 순수한 공개키만을 추출하기 위해서는 키를 생성하기 위한 난수를 고정시켜서 난수의 의존도 및 순수 공개키의 충돌을 분석한다.

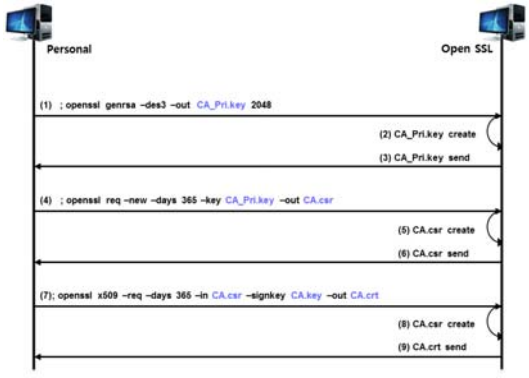
3.3 사설 인증기관 생성

실제적으로 인증서가 이용되는 환경과 동일한 환경을 구축하기 위해, 5개의 사설 인증기관을 생성한다. 사설 인증기관을 생성하는 절차는 다음과 같다.



[Fig. 7] The create procedures of the private certification authority(CA)

테스트 환경에서 OpenSSL을 이용하여 해당 사설인증기관의 공개키 및 개인키, CSR 정보를 생성한다. 최상위 인증기관이 생성한 정보에 서명하여, 사설인증기관의 인증서가 생성된다. 따라서 사설인증기관의 인증서는 최상위인증기관에서 전자서명된 신뢰할 수 있는 인증서로 실제 인증서 트러스트 체인을 따른다. 다음은 사설인증기관을 생성하는 세부 과정이다.



[Fig. 8] The create protocol of the private certification authority(CA)

(1) openssl genrsa -des -out CA_Pri.key 2048
: openssl genrsa는 rsa용 키생성과 관련된 커맨드이며, -des3는 생성되는 키값을 triple des 암호 알고리즘을 이용하여 암호화 한다. 생략하게 되면 key 파일에서 암호가 제거된다. -out은 키 이름을 지정하고, 2048은 생성되는 키의 사이즈를 의미한다.

(4) openssl req -new -days 365 -key CA_Pri.key -out CA.csr
: openssl req는 인증서 생성을 위한 인증서 생성 요청 파일(csr)을 만들며 -new는 신규 인증서 요청 파일을 생성한다. -key는 인증서 요청 파일에 들어갈 키 값을 지정해준다. -out는 생성될 인증서 요청 파일명이며,

CA_Pri.key 정보를 이용해 CA.csr 파일을 생성한다.

(7) openssl x509 -req -days 365 -in CA.csr -signkey CA.key -out CA.crt

: 최상위인증기관으로부터 사설인증기관에 인증서를 생성해주는 과정으로 x.509 형식으로 인증서를 생성하고 유효기간을 365일로 제한하며, 사설인증기관의 csr 정보를 이용하여 -sign.key 셀프 서명을 하여 자기 자신의 개인키로 자기 자신의 공개키가 포함된 인증서를 생성한다.

3.4 인증서 생성 및 공개키 추출

사용자의 인증서를 생성하기 위하여 사용자의 개인키와 공개키를 생성하게 되고 OpenSSL 소프트웨어로 공개키와 CSR 정보를 전송하게 된다. CSR은 인증서에 들어가는 식별정보라 할 수 있다. 다음은 CSR 정보에 대한 내용이다.

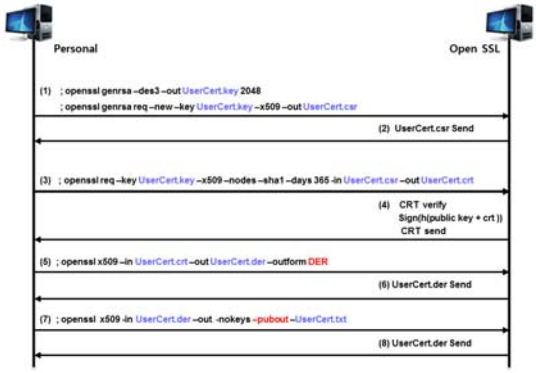
[Table 2] The CSR contains information about Signature

the main components	description
Country Name	Country
State or Province Name	City / State / Province
Locality Name	City / Town
Organization Name	Company Name
Organizational Unit Name	Server domain
Common Name	Name
Email Address	Email Address

개인인증서를 생성하고 공개키를 추출하는 절차는 다음과 같다.

[Fig. 9] The certification generation and the extraction procedure of public key

개인인증서를 생성하는 과정은 사설인증기관을 생성하는 과정과 동일하다. 개인키 및 공개키를 생성하고 사용자의 정보가 들어있는 CSR 파일을 생성하여 사설인증기관에 인증을 요청한다. CRT 파일 형식의 인증서는 OpenSSL 소프트웨어에서 발급시 옵션을 주지 않으면 기본파일 형태인 base64 encoding 값으로 기본 설정되어 있기 때문에 공개키 데이터를 추출하기 위하여 바이너리 포맷인 der 파일 형식으로 변경을 요청하게 된다. 공개키 데이터를 얻기 위해서 인증서에 포함된 인증서의 공개키 데이터를 추출하여 데이터베이스에 저장하기 위한 텍스트 데이터를 생성한다. 다음은 개인키를 생성하고 공개키를 추출하는 세부 과정이다.



[Fig. 10] The protocol of the certification generation and extraction public key

(1) openssl genrsa -des3 -out -UserCert.key 2048
openssl genrsa req -new -key UserCert.key -x509 -out UserCert.csr

: rsa용 키 생성 관련 커맨드를 입력하고 des3 알고리즘을 이용하여 암호화한다. UserCert.key를 생성하여 -x509 형식의 csr 정보를 요청한다.

(3) openssl req -key UserCert.key -x509 -nodes -sha1 -days365 -in UserCert.csr -out UserCert.crt

: UserCert.key 키파일을 이용하여 -x509 형식의 인증서를 생성하게 된다. 사용기간을 365일로 지정하게 되고 해쉬 알고리즘은 sha1을 사용하게 된다.

(5) openssl x509 -in UserCert.crt -out UserCert.der -outform DER

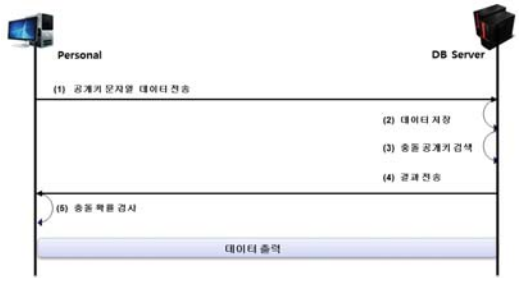
: UserCert.crt 인증서를 -out 옵션을 이용하여 UserCert.der 파일 형태로 변환하게 된다. -outform 은 변경할 파일 형태를 지정하게 된다.

(7) openssl x509 -in UserCert.der -out -nokeys -pubout -UserCert.txt

: 생성한 UserCert.der 인증서에서 공개키를 추출하여 UserCert.txt 파일을 생성한다. txt 파일 안에는 인증서에 해당하는 공개키의 바이너리 값이 저장되어 있다.

3.5 공개키 충돌 검사

다수의 컴퓨터에서 추출한 순수 공개키 데이터를 데이터베이스 서버로 모두 전송하고, 1000만개의 공개키 데이터를 생성한 후, 데이터베이스 서버를 이용하여 충돌이 발생되는 데이터를 검출하였다.



[Fig. 11] The collision detection protocol of the public key

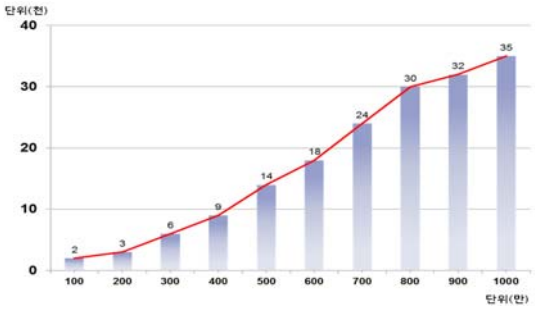
4. 실험 결과

OpenSSL 소프트웨어를 이용하여 공개키의 충돌율을 분석하였다. 실험에서는 초기 씨드값과 난수값을 고정시킨 후 순수한 공개키만을 추출하여 공개키의 충돌을 분석하였다. 다음은 인증서 1000만개를 생성하여 추출한 공개키 중에서 실제로 충돌이 발생한 공개키의 개수를 나타낸 것이다.

[Table 3] A public key collision probability of the certification generated

Creating a certificate number	Collision rate
1,000,000	2,000
2,000,000	3,000
3,000,000	6,000
4,000,000	9,000
5,000,000	14,000
6,000,000	18,000
7,000,000	24,000
8,000,000	30,000
9,000,000	32,000
10,000,000	35,000

다음은 위의 테스트 결과를 보기 쉽게 그래프로 표현한 것이다.



[Fig. 12] A public key collision probability of the certification generated

전체 공개키 충돌율은 인증서 1000만건 중에 35000건으로 0.35% 정도를 차지하였다. 이는 100만단위로 충돌율을 조사해본 결과 500만개 이상이 되었을 때 처음 충돌수치의 약 0.5배가 증가하는 것을 알 수 있다.

5. 결론

본 논문에서는 오픈소스 소프트웨어(OpenSSL)를 이용하여 공개키 기반 이용기술에 대한 안전성 및 신뢰성을 분석하였다. 실제 인증서를 생성하는 과정과 동일한 절차로 인증서를 생성하여 공개키를 추출하고 추출된 1,000만개 공개키의 충돌율을 분석하였다. 1,000만개 인증서의 공개키 중에서 동일한 공개키가 나오는 확률은 약0.35%로 35,000개의 공개키가 중복된 수치이다. 이는 국내 공인인증 환경에서 서로 다른 사용자의 인증서 상 공개키와 개인키가 일치할 수 있는 확률을 의미한다. 그리고 전자상거래, 보안서버 등 다양한 분야에서 보안적인 위협이 될 수 있다.

인증서를 생성하는 과정에서 난수를 사용할 경우, 암호화키를 만드는데 중요한 난수를 온전히 컴퓨터에게만 의존한다는 것은 인증서 자체의 안전성에 영향을 미치게 되고, 제한적인 소수의 사용 및 유사 소수의 사용 역시 인증서를 생성할 시 문제점이 될 수 있다.

본 논문에서는 단순히 오픈소스 소프트웨어를 이용하여 공개키·개인키의 충돌율을 분석하였지만, 이를 실험하는 과정에서 난수 생성, 초기화 벡터 생성, 헤더값 생성

등 원시 단에서의 충돌이 인증서 생성 후 키쌍에 영향을 미치고 있는 것이 확인되었다.

추후 공개키의 충돌을 최소화하고 유일한 키를 생성해 낼 수 있는 문제를 해결하기 위하여 난수를 생성하게 되는 난수생성기에 대한 연구가 필요하고, 큰 소수를 사용하더라도 키를 생성하는데 있어 속도에 영향을 미치지 않게 할 수 있는 방법이 요구된다.

References

[1] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. de Weger, "Short chosen-prex collisions for MD5 and the creation of a rogue CA certicate", In S. Halevi, editor, Crypto 2009, volume 5677 of Lecture Notes in Computer Science, pages 55-69. Springer, Heidelberg, 2009.
DOI: http://dx.doi.org/10.1007/978-3-642-03356-8_4

[2] D. Loebenberger and M. NÜusken, "Analyzing standards for RSA integers", In A. Nitaj and D. Pointcheval,editors, Africacrypt '11, volume 6737 of Lecture Notes in Computer Science, pp.260-277, Springer, 2011.

[3] In Bum Kim, "A Study on Enforce the Policy of User Certification in Public Certificate System", Journal of Korea Information Assurance Society 10(4), PP.69-76, 2010.

[4] Yeon-ho Jung, "Domestic PKI Construction and technology", Journal of Korea Information Assurance Society 17(6), pp.122-131, December, 2007.

[5] Seon-keun Lee, "A Study on the Modulus Multiplier Speed-up Throughput in the RSA Cryptosystem." THE JOURNAL OF KOREA INFORMATION AND COMMUNICATIONS SOCIETY 4(3), pp.217-233, September, 2009.

[6] Kwang-Eun Gil, Yi-Roo Baek, Whan-koo Kim, Jea-cheol Ha, "Fault Analysis Attacks on Control Statement of RSA Exponentiation Algorithm", Journal of The Korea Institute of Information Security and Cryptology 19(6), pp.63-70, December, 2009.

[7] Behrouz A. Forouzan, "Cryptography and Network Security", McGrawHillKorea, 2008.

[8] Woo Hyun Ahn, Hyungsu Kim, "Attacking OpenSSL Shared Library Using Code Injection", Journal of KISS : Computer Systems and Theory, pp.226-238, August, 2010.

[9] Jong-Hoon Park, Chul-won Kim, "Design and Implementation of Web Service System for secure Message Transmission in Electronic Commerce", THE JOURNAL OF KOREA INFORMATION AND COMMUNICATIONS SOCIETY 14(8), August, 2010.
DOI: http://dx.doi.org/10.6109/jkiice.2010.14.8.1855

[10] Yunyoung Lee, Soonhaeng Hur, Sangjoo Park, Donghwi Shin, Dongho Won, Seungjoo Kim, "CipherSuite Setting Problem of SSL Protocol and It's Solutions", Korea Information Processing Society Review, pp.359-366, October, 2008.

[11] Soo-jong Mo, Won-hi Cho, Sun-young Yu, Jae-hong Yim, "Design and Implementation of PKI based Cryptography Communication Component", Journal of the Korea Institute of Information and Communication Engineering, pp.1316-1322, 2005.

[12] R. Holz, L. Braun, N. Kammenhuber, and G. Carle. The SSL landscape: a thorough analysis of the x.509 PKI using active and passive measurements. In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, IMC '11, pages 427-444. ACM, 2011.
DOI: http://dx.doi.org/10.1145/2068816.2068856

[13] S. Cavallar, Zimmermann, "Factorization of a 512-bit RSA modulus", In B. Preneel, editor, Eurocrypt 2000, volume 1807 of Lecture Notes in Computer Science, pages 1-18, Springer, Heidelberg, 2000.

[14] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage, "When private keys are public: results from the 2008 debian OpenSSL vulnerability", In A. Feldmann and L. Mathy, editors, Internet Measurement Conference, pp.15-27, ACM, 2009.
DOI: http://dx.doi.org/10.1145/1644893.1644896

[15] Kyoung-Soon Hong, "Accessibility Evaluation of Accredited Certificate Subscriber Software", Journal of the Korea Contents Association, pp.40-53, February, 2011.
DOI: http://dx.doi.org/10.5392/JKCA.2011.11.2.040

[16] P. Q. Nguyen and I. Shparlinski, "The insecurity of the digital signature algorithm with partially known nonces", Journal of Cryptology 15(3), pp.151-176, 2002.
DOI: http://dx.doi.org/10.1007/s00145-002-0021-3

[17] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. "Internet X.509 Public Key Infrastructure Certicate and Certicate Revocation List (CRL) Prole", RFC 5280, 2008.

[18] Pil-Yong Kang, "Certificate usage and policy direction of the mobile revolution era", KIISC, Review 21(1), pp.51-56, Fedbruary, 2011.

[19] W.-J. Kang, "An Efficient Privacy Preserving Method based on Semantic Security Policy Enforcement", The

Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 6, pp. 173-186, Dec. 2013.
[20] J.-M. Kang, Y.-J. Song, "A Study on Structural Holes of Privacy Protection for Life Logging Service as analyzing/processing of Big-Data", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 1, pp. 189-193, Feb. 2014.
[21] J.-H. Jun, M.-J. Kim, J.-H. Cho, C.-W. Ahn, S.-H. Kim, "Detection Method of Distributed Denial-of-Service Flooding Attacks Using Analysis of Flow Information", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 1, pp. 203-209, Feb. 2014.

이 광 형(Kwang-Hyoung Lee) **[종신회원]**

- 1998년 2월 : 광주대학교 컴퓨터공학과 졸업 (공학사)
- 2002년 2월 : 숭실대학교 컴퓨터공학과 (공학석사)
- 2005년 2월 : 숭실대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 ~ 현재 : 서일대학 인터넷정보과 부교수

<관심분야>
멀티미디어 데이터 검색, 영상처리, 멀티미디어 보안, DRM, USN, 학습콘텐츠

박 정 효(Jeong-Hyo Park) **[정회원]**

- 2009년 2월 : 숭실대학교 컴퓨터학과 졸업 (공학사)
- 2011년 2월 : 숭실대학교 일반대학원 정보보안 (정보보안석사)
- 2011년 3월 ~ 현재 : 숭실대학교 일반대학원 컴퓨터공학과 (박사과정)

<관심분야>
정보통신, 통신보안, 암호이론

전 문 석(Moon-Seog Jun) **[정회원]**

- 1989년 2월 : University of Maryland Computer Science 박사
- 1989년 9월 ~ 1991년 2월 : New Mexico State University Physical Science Lab. 책임연구원
- 1991년 3월 ~ 현재 : 숭실대학교 정교수

<관심분야>
정보보호, 네트워크 보안, 인증 시스템, 암호학