클라이언트와 서버의 연산시간을 줄여주는 실시간 인증서 상태 검증에 관한 연구

A Real-time Certificate Status Validation Protocol for Reducing the Computational Time in Client and Server - RCSVP

이영교, 이영숙, 조석향, 원동호
성균관대학교 컴퓨터공학부

Young-Gyo Lee, Young-Sook Lee, Seok-Hyang Cho, Dong-Ho Won
{yglee, yslee, shcho, dhwon}@dosan.skku.ac.kr

요약

OCSP는 서버가 클라이언트에게 request와 response를 이용하여 실시간 인증서 상태 검증을 제공해주는 온라인 프로토콜이다. 서버와 클라이언트 사이에 전송되어지는 request와 response는 사용자 인증과 데이터의 무결성을 위하여 보내는 측에서 전자서명이 되고 수신측에서 서명검증이 된다. E-commerce 상에서 서버는 초당 수백만 개의 클라이언트들의 인증서 상태검증 요구를 처리해야 하는데 OCSP 서버는 전자서명과 서명검증을 위한 연산 시간으로 이 조건을 만족할 수 없다. 따라서 본 논문에서는 클라이언트-서버 환경에서 서버의 연산 시간을 줄여 수백만 개의 인증서 상태 검증 요청을 처리할 수 있는 방법을 제안하고자 한다. 이 방법은 또한 클라이언트에서의 연산 시간도 줄여준다.

■ 중심어 : | CRL | OCSP | 인증서 상태 검증 |

Abstract

As a research on PKI is being very popular, the study relating to certificate status validation is being grown with aim to reduce an overhead of the protocol and to provide an efficient operation. The OCSP of the standard protocol related to the study enables applications to determine the revocation state of an identified certificate. However, the OCSP server can not service millions of certificate status validation requests from clients in a second on E-commerce because of the computational time for signature and verification. So, we propose the Real-time Certificate Status Validation Protocol(RCSVP) that has smaller computational time than OCSP. RCSVP server reduce the computational time of certificate status validation using hash function and common secret value. Also RCSVP client does not need the computational time of certificate verification to acquire the public key from an identified certificate. Therefore, the proposed protocol enables server to response millions of certificate status validation requests from clients in a second on E-commerce.

■ Keyword : | CRL | OCSP | Certificate Status Validation |

접수번호 : #041110-001
접수일자 : 2004년 11월 10일
심사완료일 : 2004년 12월 15일
교신저자 : 이영교, e-mail : yglee@dosan.skku.ac.kr

I. Introduction

As a research on PKI(Public Key Infrastructure) is being very popular, the study relating to certificate status validation is being grown with aim to reduce an overhead of the protocol and to provide an efficient operation. OCSP(Online Certificate Status Protocol) enables applications to determine the revocation state of an identified certificate. OCSP may be used to satisfy some of the operational requirements of providing more timely revocation information than is possible with CRLs(Certificate Revocation Lists) and may also be used to obtain additional status information. The client must confirm the availability of the current status of certificate before verifying it on E-commerce. The client digitally signs a request message of certificate status validation using its private key and sends the request to OCSP server. The OCSP server verifies the request message using the client's public key. Then the server signs a response message of certificate status validation using its private key and sends the response to client. The client verifies a response message using the server's public key. The server must response millions of certificate status validation requests from clients in a second on E-commerce. However, OCSP server does not satisfy this condition because of computational time for signature and verification. In this paper, we propose a real-time certificate status validation protocol that has smaller computational time than OCSP. The proposed protocol decreases computational time of server using common secret value, hash function, and timestamp. Also, it does not need computational time of certificate verification to acquire public key from identified certificate in client. Therefore it decreases not only computational time of server but also computational time of client. The proposed protocol can response millions of requests from clients in a second. This paper is organized as follows. In section 2, we explain the related works and analyze the problem. In section 3, we explain a structure of server's database needed for our new protocol and operation of the proposed model. In section 4, we analyze the performance of the proposed protocol. Finally, in section 5, we bring to a conclusion of this paper[1,4-5,7-9].

II. Related Works

In this section, we examine the problems of OCSP, SCVP(Simplified Certificate Validation Protocol), and possible methods for certificate status validation.

1. OCSP

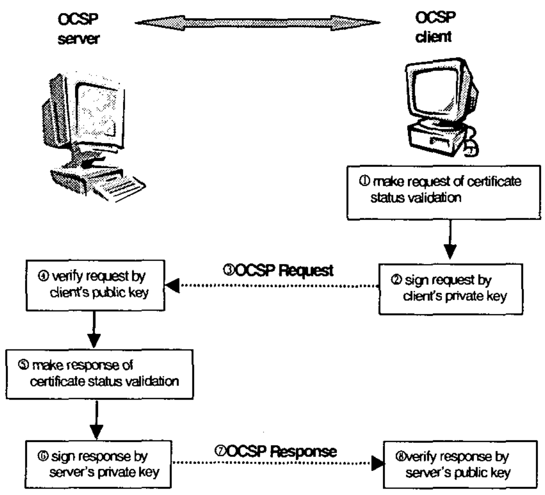
OCSP server
OCSP client
① make request of certificate status validation
② sign request by client's private key
③OCSP Request
④ verify request by client's public key
⑤ make response of certificate status validation
⑥ sign response by server's private key
⑦OCSP Response
⑧verify response by server's public key

Fig.1 Procedure of certificate status validation using OCSP

[Fig.1] shows the procedure of certificate status validation using OCSP. OCSP client digitally signs a request message of certificate status validation using its private key and sends it to OCSP server. OCSP server verifies a request message using client's public key. Then OCSP server signs a response message of certificate status validation with its private key and sends it to client. OCSP client verifies a response message with server's public key. Therefore OCSP client generates and verifies 1 signature for 1 certificate status validation. Also OCSP server generates and verifies 1 signature for 1 certificate status validation.

Cryptography	Algorithm	Computational time (msec/operation)
Asymmetric	RSA 1024 signature	4.647
	RSA 1024 verification	0.186
	RSA 2048 signature	29.174
	RSA 2048 verification	0.445
Symmetric	DES (1024 bit)	0.005
	DES (2048 bit)	0.01
Hash function	SHA-1 (160 bit)	0.0002
	SHA-256 (256 bit)	0.0006
	SHA-512 (512 bit)	0.005

Fig.2 Computational time comparison of the crypto algorithms

[Fig.2] shows computational time comparison of cryptographic algorithms for digital signature, verification, and hash function and MAC(Message Authentication Code). OCSP server and client have computational time of 4.833 msec for 1 certificate status validation in case using RSA 1024 signature/verification. Server that offers certificate status validation generally must response millions of requests from clients in a second on E-commerce. However, OCSP server cannot satisfy this condition because of computational time of signature and verification. SCVP allows a client to offload certificate handling to a server. SCVP server can provide the client with a variety of valuable information about the certificate, such as whether the certificate is valid, a certification path to a trust anchor, and revocation status.

Because SCVP has the option to sign at response message, it can decreases a computational time in server. However because unreliable session can not be accepted in transaction, the option is difficult to implement in real environment[2,7,12].

2. The method using a symmetric key

The server and client distribute a symmetric key, session-key, at first. The client encrypts a request message using session-key and sends it to server. To verify received request message, the server decrypts a request message using session-key. Also the server encrypts a response message using session-key and sends it to client. To verify a response message, the client decrypts a response message using session-key. The symmetric key algorithm has smaller computational time than public key algorithm in general. The server and client have computational time of 1 encryption and 1 encryption in case of using symmetric key. They have computational time of 0.01 msec for 1 certificate status validation in case of DES 1024 encryption/decryption. Therefore the method is also difficult to satisfy the condition for E-commerce[12].

3. The method using hash function and common secret value

The server and client distribute a common secret value at first. The client calculates the hash function over the concatenation of secret value and the Request: $H(S_{CS} \| Request)$. Client then sends $"Request, H(S_{CS} \| Request)"$ to server. Because server possesses S_{cs}, it can recalculate $H(S_{CS} \| Request)$ and verify $H(S_{CS} \| Request)$. Also server sends $"Response, H(S_{CS} \| Request)"$ to client. Client can recompute $Response, H(S_{CS} \| Response)$ and verifies $Response, H(S_{CS} \| Response)$. Because the secret value itself is not sent, it is not possible for an attacker to modify an intercepted message. As long as the secret value remains secret, it is also not possible for an attacker to generate a false message. The method only needs computational time of hash function. Server and client have computational time of 2 hash function. Therefore server and client have computational time of 0.0004 msec for 1 certificate status validation in case of SHA-1(160 bit). The method was used in ECSPVS that was proposed by Youngchul Choi et al.

ECSPVS needs VA(Validation Authority) exception server and client in addition. VA is authority that client receives VC(Validation Certificate) of certificate status and path validation for self's certificate before client sends request message to server. VA is equivalent authority with authority that manages server. Server have only computational time of 1 hash function in case of ECSPVS. Therefore ECSPVS server have computational time of 0.0002 msec for 1 certificate status validation and can service 5000000 (5×10^6) requests in a second. However ECSPVS has a little complex procedure[1],[12].

III. A Real-time Certificate Status Validation Protocol for reducing the computational time in client and server-RCSVP

In this section, we propose a protocol of certificate status validation for reducing the computational time in client and server.

1. Motivation

The ECSPVS server was reduced the computational time than OCSP server. However ECSPVS client was needed the calculation time of certificate verification in order to get the public key from an identified certificate like OCSP client. When ECSVPS client verifies the identified certificate that was signed by CA(Certification Authority), the computational time of 1 hash function and 1 signature verification is needed. The client needs the computational time of 0.1862 msec/operation in case of RSA 1024 bit and SHA-1(160 bit). So, in this paper, we propose the protocol to reduce the computational time of client as well as server[1,7-9].

2. The structure of database in RCSVP server

The RCSVP server needs a database for the proposed protocol. [Fig.3] shows the structure of database in RCSVP server. RCSVP database stores the certificate's identification number, certificate itself and validity and reason of revocation. Certificate's identification number is the serial number that is allocated at each certificate. Certificate's identification number in DB on RCSVP server is used as searching key. Certificate is that CA issues to each user and is received from CA's DIR(Directory). Validity shows whether each certificate is valid. Reason of revocation is the reason that the invalid certificate is revoked.

Certificate's identification number	Certificate	Validity	Reason of revocation
000001	A's Certificate	Y	-
000002	B's Certificate	Y	-
000003	C's Certificate	Y	-
000004	D's Certificate	N	Loss of private key
000005	E's Certificate	N	Expiration of the available period
000006	F's Certificate	Y	-
⋮	⋮	⋮	⋮

Fig.3 Structure of database in RCSVP server

3. Terminology Definition

S_{CS} : common secret value that server and client shares

$Cert_{AC}$: certificate of user A that client has. Client acquires it from user A

$Cert_{AS}$: certificate of user A that server has. Server acquires it from DIR of CA

$H_C()$: hash value calculated in client

$H_S()$: hash value calculated in server

TS_1 : $TimeStamp1$

TS_2 : $TimeStamp2$

4. The procedure of proposed RCSVP (Real-time Certificate Status Validation Protocol)

[Fig.4] shows a procedure of proposed RCSVP. RCSVP assumes that server and client share a common secret value S_{CS} at first. The following is detailed procedure of RCSVP.

4.1 When RCSVP client requests the certificate status validation of user A's certificate to RCSVP server, RCSVP client sends the certificate of user A, hash function over the concatenation of the secret value S_{CS} and $Cert_{AC}$, and TS_1.

$$Cert_{AC},\ H_C(S_{CS} \parallel Cert_{AC}),\ TS_1$$

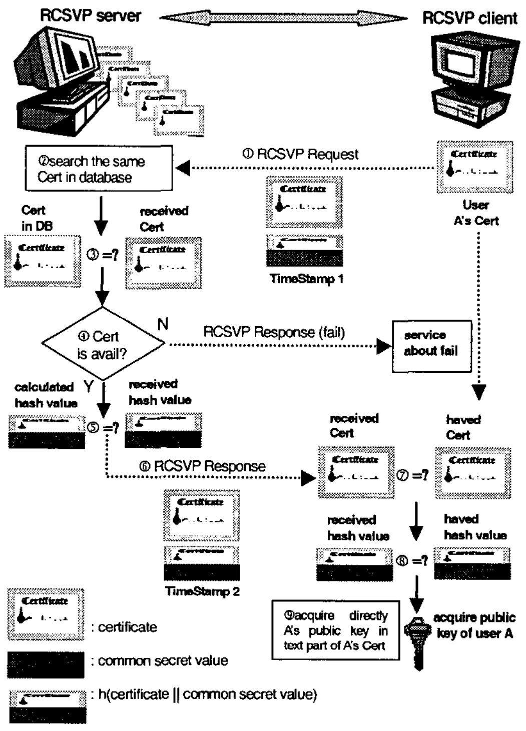

Fig.4 Procedure of proposed RCSVP

4.2 RCSVP server receives request message and searches the database with certificate identification number of A's certificate. If the same certificate is found in database, then RCSVP server confirms the integrity of the received certificate.

$$Cert_{AC} \overset{?}{=} Cert_{AS}$$

4.3 Then RCSVP server checks the validity of A's certificate in database. If validity of certificate is "Y", A's certificate is valid. Because RCSVP server possesses secret value S_{CS} and $Cert_{AS}$, it can calculate $H_S(S_{CS} \parallel Cert_{AS})$ and verify the received $H_C(S_{CS} \| Cert_{AC})$. If calculated hash value and received hash value are same, RCSVP server can have the entity authentication of RCSVP client.

$$H_C(S_{CS} \| Cert_{AC}) \overset{?}{=} H_S(S_{CS} \| Cert_{AS})$$

4.4 If validity of an identified certificate is "Y", RCSVP server responses the message that involves the certificate of A, calculated hash value, and TS_2 to RCSVP client.

$$Cert_{AS},\ H_S(S_{CS} \| Cert_{AS}),\ TS_2$$

4.5 RCSVP client receives the response message and compares the received certificate and the sent certificate. If two certificates are same, client can confirm the integrity of the received certificate. Also RCSVP client can confirm the validity of A's certificate.

$$Cert_{AS} \overset{?}{=} Cert_{AC}$$

4.6 RCSVP client compares the received hash value and holding the hash value of step 1. If two hash values are same, RCSVP client can also confirm the entity authentication of server. Then RCSVP client obtains directly A's public key from text part of A's certificate without the additional computational time for signature verification.

$$H_S(S_{CS} \| Cert_{AS}) \overset{?}{=} H_C(S_{CS} \| Cert_{AC})$$

TS_1 and TS_2 are used to prevent replay attack by attacker. In general, for the data integrity and entity authentication that is used in digital signature, OCSP and SCVP and etc use the method that compares the original part and the compared part included in one message. The compared part is made by the hash value of the original message signed using self's private key. Sender sends the message including the original part and the compared part. Then receiver receives the message and compares the original part and the compared part. Receiver calculates the hash value of the original part at first. And receiver acquires the hash value by the signature verification of the compared part using sender's public key. If two hash value are same, receiver could conform the data integrity and entity authentication. If the method be used in client-sever environment, client and server must have computational time of 1 signature and 1 signature verification.

Therefore these methods can not overcome the overhead of computational time at peak time on the large-scale client-server environment. However, the proposed protocol locates the compared part at server and original part at client. And the protocol offers data integrity and entity authentication using hash computational time and common secret value without signature and signature verification. Also the client obtains directly A's public key from text part of A's certificate without computational time of signature for verification. Therefore the proposed protocol reduces not only computational time of server but also computational time of client.

IV. Performance Analysis

[Fig.5] shows comparison of proposed RCSVP and others in case of SHA-1(160 bit), MD5-MAC 128 bit, and RSA 1024 signature/verification. RCSVP server needs a computational time of 1 hash function and 1 time stamping for a request of certificate status validation in [Fig.5]. RCSVP server needs the computational time of 0.00028(2.8×10^{-4})

msec/operation.

Therefore, RCSVP server can service certificate status validation of 3571428.571 ($\cong 3.6 \times 10^6$) operation/sec. And, ECSPVS server can service certificate status validations of 5000000($\cong 5 \times 10^6$), and OCSP server can service that of 206.894($\cong 2.1 \times 10^2$) operation/sec. RCSVP client needs a computational time of 1 time stamping, 1 time verification, and 1 hash function for a request of certificate status validation in [Fig.5]. RCSVP client needs a computational time of 0.00036(3.6×10^{-4}) msec/operation in case of SHA-1(160 bit) and MD5-MAC 128 bit. Therefore, RCSVP client can service certificate status validation and signature verification of 2777777.778 ($\cong 2.8 \times 10^6$) operation/sec. ECSPVS client can service that of 5370.569($\cong 5.4 \times 10^3$) operation/ sec, and OCSP client can service that of 199.219 ($\cong 2.0 \times 10^2$) operation/sec.

RCSVP is more simple than ECSPVS but is more complex than OCSP in procedure. The following is detailed performance comparison of the each protocols[1],[2],[7].

Method / Item	RCSVP (Proposed)	ECSPVS	OCSP
computational quantity 1	hash computation: 1 time stamping computation:1	hash computation:1	hash computation: 2 signature verification : 1 signature : 1
computational time (1)	0.00028 msec/operation 3571428.571 operation/sec	0.0002 msec/operation 5000000 operation/sec	4.8334 msec/operation 206.894 operation/sec
computational quantity 2	hash computation: 1 time stamping computation: 1 time verification computation:1	x	hash computation: 2 signature : 1 signature verification :1
computational quantity 3	x	hash computation: 1 signature verification :1	hash computation: 1 signature verification :1
computational time(2+3)	0.00036 msec/operation 2777777.778 operation/sec	0.1862 msec/operation 5370.569 operation/sec	5.0196 msec/operation 199.219 operation/sec
Add of authority	x	필요(VA)	x
common secret value	o	o	x
initial procedure	o (1 pass)	o (4 pass)	x
validation procedure	2 pass	5 pass	2 pass
needed database	certificate, validity, reason of revocation	SCSDH, SVADH, CSVADH, RN, DNCC	x
use of timestamp	o	o	x

*based on SHA-1(160 bit), MD5-MAC 128 bit, and RSA 1024signature / verification

Fig.5 Performance comparison of the proposed RCSVP and others

1. The computational time of RCSVP server

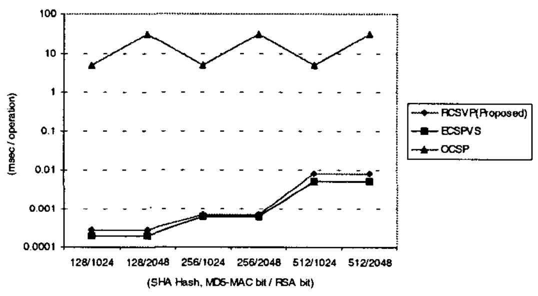

Fig.6 Performance comparison of the computational time by crypto algorithms on server

[Fig.6] shows a detailed performance comparison of the computational time of each protocols by crypto algorithms. It is compared between all possible compounding about SHA Hash and MD5-MAC of 128, 256, and 512 bit and RSA of 1024 and 2048 bit in comparison. Performance of RCSVP and ECSPVS shows better than OCSP on server in case of all. Although having little performance difference, RCSVP is similar to ECSPVS[1],[12].

2. The computational time of RCSVP client

[Fig.7] shows detailed performance comparison of the computational time of each protocols by crypto algorithms. It is compared between all possible compounding about SHA Hash and MD5-MAC of 128, 256, and 512 bit and RSA of 1024 and 2048 bit in comparison. [Fig. 7] shows that Performance of RCSVP is better than ECSPVS and OCSP on client in case of all. Because client also must service a lot of user, computational time of client is very important to have an effect on processing time of total service in E-commerce[12].

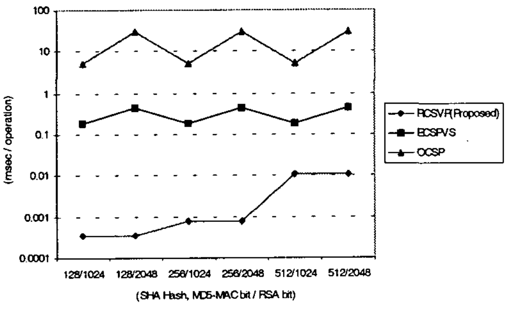

Fig.7 Performance comparison of the computational time by crypto algorithms on client

3. Role of common secret value

Common secret value, S_{CS}, offers entity authentication in proposed protocol. Because common secret value is different from private key or session key, it is not critical in case of loss. Also because common secret value itself is not sent and hash function is one way, an attacker can not easily discover the common secret value. Therefore as long as the common secret value remains secret, it is also not possible for an attacker to generate a false message. Common secret value must be shared between client and server before main procedure on PKI. After secret value is made in server and is signed with client's public key, it can be transferred from server to client. Then client acquires secret value with signature verification. Or inverse procedure is possible between client and server. Change of common secret value periodically can increase the security of proposed protocol.

4. The computational time for time stamping and time verification

TS_1 and TS_2 will prevent the replay attack by attacker. TSA(Time Stamping Authority) services the time stamping and the time verification. Client and server must send hash value for time stamping and time verification to TSA. TSA can use digital signature or MAC for time stamping and time verification. To save computational time, client and server can select a method using MAC. Computational time of MD5-MAC is 0.00008 ($\cong 8.0 \times 10^{-5}$) msec/operation for 128 bit, 0.00016 ($\cong 1.6 \times 10^{-4}$) msec/operation for 256 bit, and 0.00032 (3.2×10^{-4}) msec/operation for 512 bit. Because client and server did hash calculation already in the proposed RCSVP, the addition of hash computational time for time service is not required[3],[12].

5. Saving the computational time for signature verification

If the certificate received from user A and the certificate returned from RCSVP server are same, RCSVP client confirms the integrity of certificate not only between server and client but also between user and client. Therefore RCSVP client directly can acquire A's public key from text part of user A's certificate without more computation. This can save computational time of 1 hash function and 1 signature verification in RCSVP client. The proposed protocol has an advantage to save the computational time of not only certificate status validation but also certificate verification simultaneously.

V. Conclusion

The server that offers certificate status validation can have the critical processing time problem because of millions of certificate status verifications if millions of users simultaneously connect and send the message of log-on or digital signature on the large-scale client-server environment based on the real-time like Internet banking or Internet stock trading at peak time. The standardized OCSP in recent also has the same problem continuously. This paper proposed the certificate status verification protocol capable to reduce the computational time of client/server on the large-scale client-server environment based on the real-time. The proposed RCSVP reduces the computational time of server for certificate status verification by sending/receiving certificate between client and server using hashing-function and common secret. Also, the RCSVP reduce the computational time of client because of deleting the computational time of certificate verification needed to acquire the public key of user. Therefore it decreases not only computational time of server but also computational time of client. The computational time of RCSVP client is reduced than OCSP and ECSPVS. Also, the computational time of RCSVP server is reduced than OCSP and is similar to ECSPVS. The proposed RCSVP is better than OCSP and ECSVPS in aspects of decreasing the computational time of server and client. Therefore, the proposed RCSVP is typically the protocol applicable to the large-scale client-server based on the real-time.

References

[1] Y.C. Choi, S.J. Park, and D.H. Won, "An Efficient Certificate Status and Path Validation System for Client-Server Environment," Journal of Korea Institute of Information Security and Cryptology, Vol.13, No.1, 2003.

[2] A. Malpani, R. Housley, and T. Freeman, "Simple Certificate Validation Protocol (SCVP)," IETF Internet Draft, 2002.

[3] Y.S. Lim, and K.H.. Kang, "Technical Standardization Tendency of Time Stamping Service," Review of Korea Institute of Information Security and Cryptology, Vol.11, No.6, 2001.

[4] C. Adams, P. Sylvestor, M. Zolotarev, and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols. IETF RFC 3029, 2001.

[5] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," IETF RFC 2458, 1999.

[6] R. Housley, W. Ford, W. Polk and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," IETF RFC 3280, 2002.

[7] M. Myers, R. Ankney, A. Mappani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," IETF RFC 2560, 1999.

[8] ANSI X 9.31, "1998 Digital signatures using reversible public crytography for the financial services industry (rDSA)," 1998.

[9] NIST FIPS (Federal Information Processing Standards Publication) 186-1, "Digital Signature Standard," 1998.

[10] Silvio Micali, "NOVOMODO, Scable Certificate Validation And Simplified PKI Management," 1st Annual PKI Rearch Workshop, Preproceedings, pp.9-19, 2002.

[11] Paul. Kocher, "A Quick Introduction to Certificate Revocation Tree(CRTs)," Technical Report, Valicert, 1999.

[12] http://www.eskimo.com/~weidai/ benchmarks.html.

저 자 소 개

이 영 교(Young-Gyo Lee) 정회원

• 1986년 2월 : 한양대학교 전자공학과 졸업(학사)
• 1991년 6월 : 한양대학교 전자공학과 졸업(석사)
• 2002년 3월~현재 : 성균관대학교 전기전자 및 컴퓨터공학부 박사수료
• 1993년 3월~1998년 9월 : 대우통신 종합연구소 선임연구원
• 1999년 2월~2001년 6월 : LG 전자/정보통신 중앙연구소 선임연구원
• 2002년 3월~현재 : 인하공업전문대학 정보통신과 강사
• 2004년 9월~현재 : 아주대학교 정보통신대학원 강사

<관심분야> : 암호 프로토콜, 정보통신 보안, 네트워크 이론

이 영 숙(Young-Sook Lee) 정회원

• 1987년 2월 : 성균관대학교 정보공학과 졸업(학사)
• 2002년 9월~현재 : 성균관대학교 정보통신대학원 정보보호학과 (석사)
• 2005년 3월~현재 : 성균관대학교 컴퓨터공학부 박사과정
• 2002년 3월~현재 : 두원공과대학 소프트웨어개발과 강사

<관심분야> : 정보통신 보안, 암호 알고리즘,

조 석 향 (Seok-Hyang Cho) 정회원

• 1986년 2월 : 이화여자대학교 수학과(학사)
• 2001년 3월~현재 : 성균관대학교 전기전자 및 컴퓨터공학부 박사수료
• 1986년~1998년 : (주)중앙교육진흥연구소 선임연구원
• 2001년 2월 : 서울산업대학교 산업대학원 전자계산학과 졸업(석사)
• 2004년 9월~현재 : 서울산업대학교 컴퓨터공학과 강사
• 2005년 3월~현재 : 성균관대학교 컴퓨터공학과 강사

<관심분야> : 암호 프로토콜, 암호 이론, 정보 보안

원 동 호(Dong-Ho Won) 정회원

• 1949년 9월 23일생, 성균관대학교 전자공학과(학사, 석사, 박사)
• 1978년~1980년 : 한국전자통신연구소(ETRI) 전임연구원
• 1985년~1986년 : 일본 동경공대 객원 연구원
• 1988년~1999년 : 성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학원장,정보통신기술연구소장
• 1996년~1998년 : 국무총리실 국가정보화 추진자문위원회 자문위원
• 2002년~2003년 : 한국정보보호학회장
• 현재 : 성균관대학교 정보통신공학부 교수, 정통부 지정 정보보호인증기술연구센터 센터장
<관심분야> : 암호학, 정보통신 보안, 암호알고리즘