

공개키 기반 구조에서의 효율적인 인증서 상태 검증 방법에 관한 모델링 및 시뮬레이션*

최지혜**, 조대호*

Modeling and Simulation of the Efficient Certificate Status Validation System on Public Key Infrastructure

Ji-Hye Choi, Tae-Ho Cho

Abstract

공개키 기반 구조(PKI; Public Key Infrastructure)에 필수적인 요소인 인증서의 상태 검증에 있어서 인증서 상태 검증 서버인 OCSP (Online Certificate Status Protocol) 서버는 실시간 상태 검증을 제공한다. 그러나, 서버와 클라이언트의 메시지 인증을 위해 전자 서명을 수행해야 하며, 이 때 사용되는 공개 암호 연산 과정의 복잡성은 동시에 많은 클라이언트의 요청이 발생할 경우에 응답 시간을 크게 지연시킨다는 단점을 가지고 있다. 본 논문에서는 이러한 문제를 해결하기 위한 인증서 상태 검증 서버의 시뮬레이션 모델을 DEVS (Discrete Event system Specification) 방법론을 이용하여 설계하였다. 이 모델은 인증서의 상태 검증을 요청하는 영역에 위치하여 해쉬 함수를 적용한 인증을 수행하도록 구성되었으며, 시뮬레이션 결과는 제시한 방법이 인증서 상태 검증 속도를 증대시켜 결과적으로 사용자의 응답 시간이 감소되는 것을 보여준다.

Key Words: PKI, Certificate Status Validation, DEVS formalism, Simulation

* 본 연구는 정보통신연구진흥원(Institute of Information Technology Assessment)에 의해 지원되었음

** 성균관대학교 컴퓨터공학과

1. 서론

공개키 암호 시스템(Public Key Cryptography)은 전자 상거래 등의 정보 보호를 위한 분야에서 대표적으로 사용되는 보안 메커니즘이며, 이를 효율적으로 적용 및 활용하기 위해 등장한 개념이 공개키 기반 구조(PKI; Public Key Infrastructure)이다[1]. PKI는 기업 보안 구조에 있어서 중요한 부분으로서 보안 관리 측면에 있어서 핵심적인 기능을 제공하며, 안전한 전자 상거래 환경을 위해 인증서를 발급하여 이를 통해 사용자와 메시지의 인증을 수행한다[2].

PKI에서의 신뢰 당사자는 인증서를 받은 후, 이 인증서가 유효한지를 검사하는 인증서의 상태 검증 과정을 수행해야 하며[3], 인증서 상태를 확인하기 위하여 인증서 폐지 목록(CRL; Certificate Revocation List)[4], OCSP(Online Certificate Status Protocol)[5] 등을 사용한다.

현재 대부분의 PKI 제품에서는 OCSP의 기능을 제공하여 실시간 인증서 상태 검증을 수행하고 있다[5-6]. 그러나, OCSP 서버와 클라이언트는 상태 검증 요청 메시지와 응답 메시지 생성시 전자 서명을 수행해야 하며, 이 과정에서 공개키 암호 연산을 사용하기 때문에 사용자의 응답 시간이 지연된다는 단점이 있다[5].

본 논문에서는 이 문제점을 해결하기 위한 방안으로 인증서 상태 검증 서버와 클라이언트간의 메시지 인증시 해쉬 함수를 사용하는 시스템을 제안하고, 기존의 시스템에 비해 응답 시간이 감소되는 결과를 시물레이션을 통해 확인한다.

2. 관련 연구

2.1 인증서 상태 검증 방법

인증서와 관련한 국제 표준인 X.509에서는 인증서 폐지 방법으로 CRL이라는 폐지된 인증서를 식별하는 목록을 정의하고 있으며, 이를 사용하여 인증서의 유효성을 검증한다.

PKI에서 인증서가 발급되고 나면 그 인증서는 정해진 유효기간 동안 사용될 것으로 기대

된다. 그러나, 소유자의 개인키가 노출되었거나 소유자와 인증기관의 관계가 변화하는 등의 이유로 유효기간이 만료되기 전에 무효화될 수도 있다. 이러한 상황은 해당 개인키의 손상 또는 손상 가능성을 포함하기 때문에 인증기관은 인증서를 폐지할 필요가 있다[4].

2.1.1 CRL 기반 인증서 상태 검증 방법

인증서 상태 검증을 원하는 신뢰 당사자가 인증기관으로부터 CRL을 주기적으로 발급받는 방식으로, 개념이 간단하다는 장점이 있으나 주기적인 다운로드에 따라 해당 주기 사이에 폐지되는 인증서의 상태는 검증할 수 없다는 단점이 있다[7]. 대표적으로 Delta CRL, Indirect CRL, Freshest CRL 등의 방법이 있다[4].

2.1.2 온라인 인증서 상태 검증 방법

CRL 기반 인증서 상태 검증 방법론이 가지고 있는 시간 격차(time gap) 문제를 해결하기 위해 제안된 방법으로, OCSP가 대표적이다. OCSP는 온라인상에서 OCSP 서버와 인증서 상태 검증을 요구하는 신뢰 당사자인 OCSP 클라이언트 간에 수행되는 프로토콜로서, OCSP 서버는 신뢰 당사자의 해당 인증서의 상태를 묻는 질의에 대해 인증서 폐지 여부를 실시간으로 확인시켜준다[5-6].

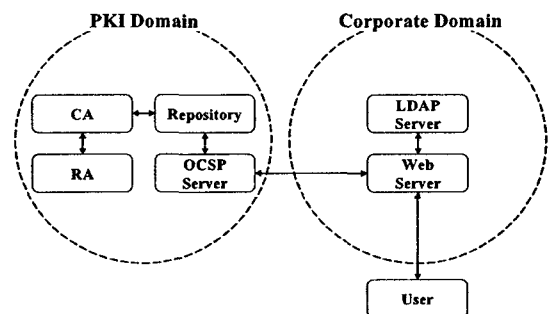


그림 1. OCSP의 인증서 상태 검증 구조

그림 1은 기업 영역 내의 웹 서버가 서비스를 요구하는 사용자의 인증서를 검증하기 위해 OCSP 서버에게 질의하는 과정을 나타낸 것이다.

대부분의 PKI 기반 환경에서 사용되는 방법이지만, OCSP 서버와 클라이언트간 전송되는

메시지를 서명하고 검증하는 과정에 공개키 암호 연산을 사용하기 때문에 사용자의 응답 시간이 지연된다는 단점이 있다[5].

2.2 DEVS 형식론

DEVS 형식론은 집합 이론에 근거하여, 연속적인 시간상에서 발생하는 이산 사건 시스템을 시뮬레이션하기 위해 정립된 모델링 방법론이다. 이는 모델의 구조와 행동을 시뮬레이션 수행으로부터 추상화시키기 위해 모델을 집합 이론적 방법으로 이용한 것으로, 시스템을 계층적(hierarchical)이고 모듈화(modular)된 형식으로 기술한다[8].

DEVS에서는 기본 모델(basic model)과 결합 모델(coupled model)을 정의한다. 기본 모델은 시간 명세 상태 천이 레벨(timed state transition level)에서 시스템의 동작을 표현하는 모델로서 다음의 항으로 명세할 수 있다.

- $M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$
- X : 입력 사건 집합
 - S : 상태들의 집합
 - Y : 출력 사건 집합
 - $\delta_{int} : S \rightarrow S$ 내부 상태 전이 함수
 - $\delta_{ext} : Q \times X \rightarrow S$ 외부 상태 전이 함수
 - $\lambda : S \rightarrow Y$ 출력함수
 - $ta : S \rightarrow \mathbb{R}^+$ 시간 진행 함수

결합 모델은 시스템 구성 요소간의 상호 작용을 표현하기 위한 모델로서 기본 모델이나 결합 모델로 기술된 구성 요소 모델들을 연결하여 만든 모델이다. 이 모델은 다음과 같은 항으로 명세가 가능하다.

- $N = \langle X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC, Select \rangle$
- X : 입력 사건 집합
 - Y : 출력 사건 집합
 - D : 구성 요소 이름의 집합
 - M_d : DEVS 모델의 구성 요소
 - EIC : 외부 입력 관계
 - EOC : 외부 출력관계
 - IC : 내부 입출력 관계
 - $Select : 2^D - \{ \} \rightarrow D$

3. 제안하는 인증서 상태 검증 시스템

3.1 제안 시스템의 개념

본 논문에서는 사용자의 응답 시간을 감소시키는 방법으로 메시지 인증 방식에서 해쉬 함수를 적용할 것을 제안한다. 해쉬 연산을 이용한 인증 방식에서는 인증을 원하는 사용자 간에 미리 비밀 정보를 나눠가지고 있어야 하므로, 네트워크의 크기가 커질수록 비밀 정보의 수 또한 급격하게 증가하게 되는 단점을 가지고 있으나, 연산의 속도는 공개 암호 방식에 비해 최소 10,000배 이상 빠르다[9].

따라서 본 논문에서는 인증서 상태 검증의 속도를 줄이기 위해 해쉬 함수를 적용하였으며, 비밀 정보 수의 증가 문제는 상태 검증 서버를 상태 검증 요청 서버와 같은 영역에 설치하는 것으로 해결하였다. 내부 네트워크 내에서 해쉬 함수를 사용하는 인증 방식을 사용할 경우에는 당사자간의 비밀 정보만 보관하면 되므로, 다수의 비밀 정보 보관에 대한 번거로움을 없앨 수 있다.

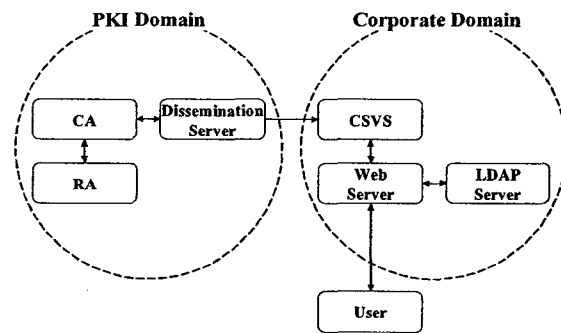


그림 2. 제안 시스템의 인증서 상태 검증 구조

본 논문에서 설계한 인증서 상태 검증 구조는 그림 2와 같다. 제안하는 인증서 상태 검증 서버(CSVS; Certificate Status Validation Server)는 기업 영역 내에 위치하여 웹 서버와 인증 관련 정보를 주고받으며, 이때 해쉬 함수가 적용된다. PKI 영역에서의 인증 관련 정보를 저장하고 있는 서버(Dissemination Server)는 CRL이 새로 발행될 때마다 CSVS에게 배포하는 방법으로 CRL 기반 방법이 가지고 있는 시간 격차 문제를 해결한다.

3.2 제안 시스템의 모델 구성

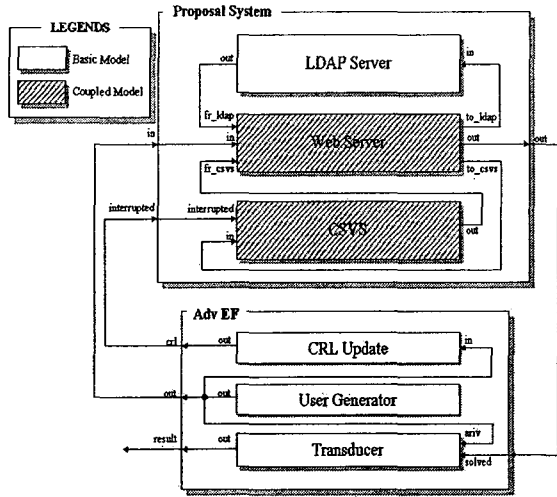


그림 3. DEVS 시뮬레이션 모델의 구조

제안 시스템을 구성하는 모델은 DEVS 형식론에 근거하여 모델링 하였으며 그림 3과 같다. Adv EF 모델은 사용자의 서비스 요청과 CRL의 발급을 지수 분포에 따라 발생시키며, 시뮬레이션을 통해 얻은 결과를 분석한다. Proposal System 모델은 사용자의 서비스 요청에 대한 처리를 담당하는 부분으로 Web Server 모델, LDAP Server 모델, CSVs 모델로 구성된다.

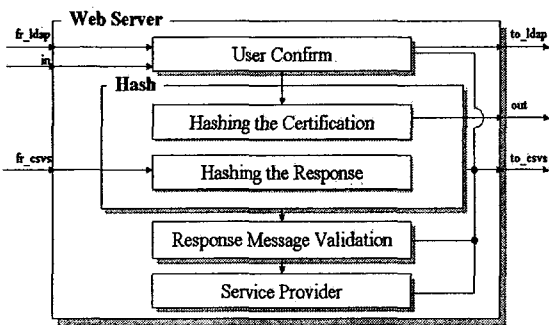


그림 4. Web Server 모델의 구성

Web Server 모델은 그림 4와 같이 구성되며, LDAP Server 모델과 연동하여 서비스를 요청하는 사용자가 사전에 등록된 사용자인지 확인하고, CSVs 모델에게 인증서 상태를 검증을 요청한다. CSVs 모델로부터 해당 인증서에 대한 응답 메시지를 받아 유효한 인증서임을

이 확인되면 사용자에게 서비스를 제공한다.

그림 5는 Web Server 모델의 인증서 검증 요청에 대한 처리를 하는 CSVs 모델로서 상태 검증 과정 중 CRL이 발급된다면 CRL 갱신 과정을 먼저 수행한다.

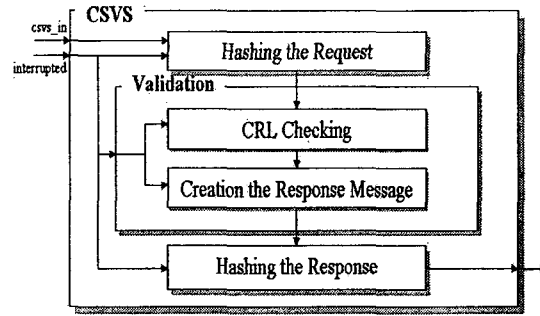


그림 5. CSVs 모델의 구성

3.3 제안 시스템의 동작 과정

3.3.1. 초기 등록 과정

- 사용자와 CSVs는 공인 인증 기관(CA)에 등록하여, 인증서를 발급 받는다.
- Web Server와 CSVs는 해쉬 연산 수행에 필요한 비밀 키(WSVs)를 생성하여 나눠 가진다.

3.3.2 인증서 상태 검증 동작 과정

그림 3의 Proposal System 모델의 동작 과정으로서 사용자로부터 요청을 입력받아 서비스를 제공하기까지의 처리과정을 기술한다.

[Step 1]

- 사용자가 Web Server에게 서비스를 요청한다.
- Web Server는 LDAP 서버를 참조하여 서비스를 요청하는 사용자가 사전에 등록되어 있는지 확인한다.

[Step 2]

- 등록된 사용자일 경우 Web Server는 사용자에게 인증 정보를 요청하여 전송받는다.
- Web Server는 사용자의 인증 정보와 WSVs에 해쉬 함수를 적용하여, CSVs에게 전송한다.

[Step 3]

- CSVs는 Web Server로부터 받은 메시지를

검증하고, 사용자의 인증 정보를 이용해 해당 인증서의 폐지 여부를 확인한다.

- 상태 확인 결과를 응답 메시지로 작성하고, 메시지와 WSVS에 해쉬 연산을 하여 Web Server에게 전송한다.

[Step 4]

- Web Server는 CSVS로부터 받은 응답 메시지를 검증한다.
- 응답 메시지의 내용을 확인하여 사용자에게 서비스를 제공한다.

3.3.3 인증서 폐지 목록 갱신시 동작 과정

인증서 폐지 목록이 새로 발행될 때마다 PKI 영역의 CRL 배포 서버에서 기업 영역 내의 CSVS에게 Delta CRL을 발급해준다. CSVS는 배포 서버로부터 받은 정보를 자신의 개인키로 복호화하여 서명을 검증하며, 자신이 가지고 있는 CRL을 갱신시킨다.

이 과정은 CSVS에서 인증서 상태 검증 작업을 수행하는 도중에 CRL을 발급 받는다면 상태 검증 작업보다 우선하여 처리되도록 설계되어 상태 확인의 실시간 처리를 보장한다. 이 과정에서의 시스템 처리 시간은 서명의 검증 시간만큼 길어지게 되나, 만일 사용자가 서비스를 요구하지 않을 때 CRL이 갱신된다면, 이 때의 처리 속도는 사용자의 응답 시간에 영향을 미치지 않는다.

4. 실험 및 결과 분석

본 논문에서는 기존 시스템과 제안된 시스템의 성능 측정을 위한 시뮬레이션을 수행하였다. 시뮬레이션 환경은 본 연구진이 개발한 DEVS-ObjC를 사용하였으며, 시뮬레이션을 위한 성능지표로 각 시스템에서의 응답 시간을 측정하였다.

시스템에 접속하는 사용자의 수는 하루에 10,000명으로 가정하고 실험하였으며, 동시에 접속하는 사용자의 수가 많아짐에 따른 응답 시간을 측정하기 위해 사용자의 접속 간격을 변화시켜 입력 값으로 사용하였다. 각 시스템의 속도는 전자 서명과 해쉬 함수에 대한 처리 시간만 고려하였으며[9-10], 제안된 시스템에서의 CRL 발급 수는 인증서 발급 수의 10%로 가정하였다[11].

그림 6의 시뮬레이션 결과는 각각 기존 시스템(위)과 제안된 시스템(아래)에서의 응답 시간을 측정한 것으로 단시간에 접속하는 사용자의 수가 많아질수록 응답 시간이 급격하게 증가하는 것을 보여준다. 그러나 제안된 인증서 상태 검증 시스템에서의 응답 시간은 동시 접속자 수가 많아지더라도 그 값이 매우 작기 때문에, 기존 시스템에 비해 매우 효율적이라는 사실을 알 수 있다.

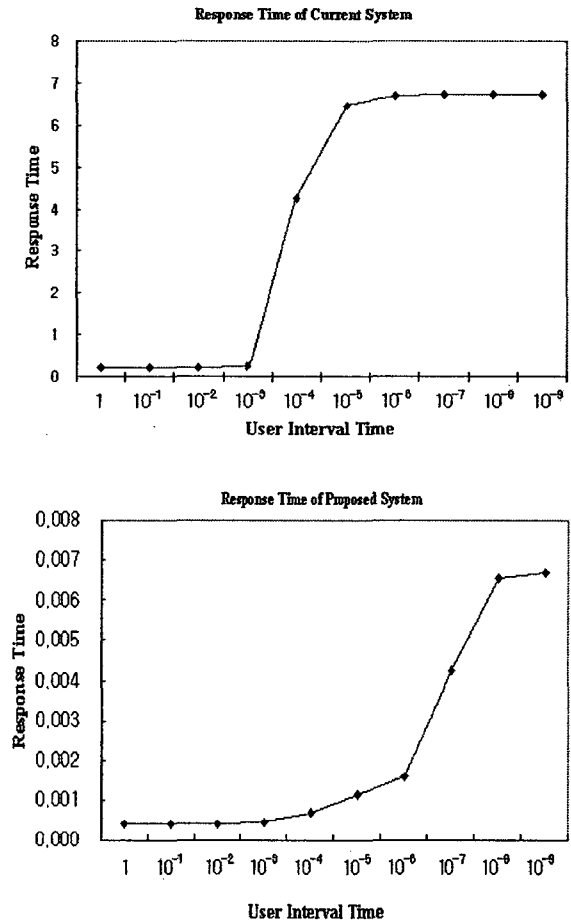


그림 6. 기존 시스템과 제안 시스템의 응답시간

제안된 시스템에서의 처리 속도는 CRL의 갱신에 영향을 받으므로, 이에 따른 사용자의 응답 시간이 달라질 수 있다. 때문에 CRL의 발급 수를 변화시켜 실험하였고, 결과는 그림 7과 같다. 이 결과는 CRL이 빈번하게 발급되더라도 기존의 시스템보다 처리 속도가 빠르다는 것을 보여준다.

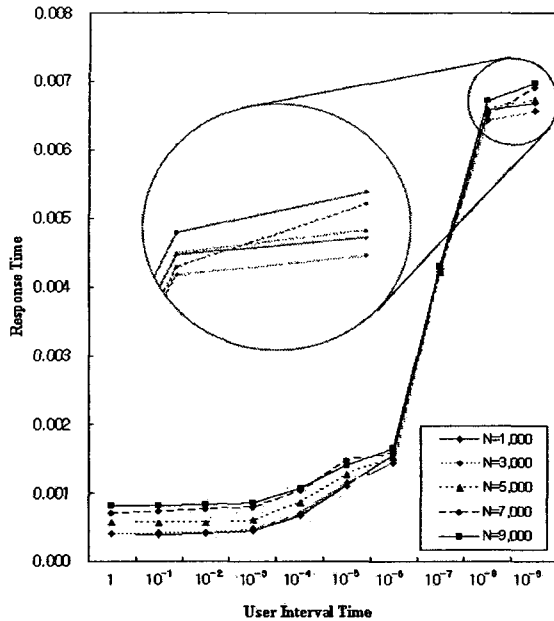


그림 7. CRL 발급 수의 변화에 따른 응답시간

5. 결론

본 논문에서는 PKI에서 반드시 수행되어야 하는 인증서 상태 검증 과정에서의 처리 속도를 줄이기 위한 시스템을 제안하였다. 기존의 시스템에서는 사용자의 서비스 요청이 단시간에 많아질 경우 응답 지연율이 커진다는 문제점이 있었다. 이는 인증서 검증 과정 중 서명문의 생성과 검증에 필요한 시간이 길기 때문이며, 지연 시간이 길어진다면 인증서 상태 검증 또한 실시간으로 제공될 수 없을 것이다.

본 논문에서는 인증서 상태 검증 과정을 실시간으로 제공하고 사용자의 응답 지연율을 낮추기 위한 방안으로 기업 영역 내에 검증 서버를 설치하여 해쉬 함수를 적용한 인증 과정을 수행하도록 하였다. 제안된 시스템의 효율성은 시뮬레이션을 통한 실험으로 예상할 수 있다.

제안한 시스템은 Delta CRL을 발급받아 파일 형식의 데이터로 보관하며, 인증서 상태 확인 과정에서 다수의 CRL 파일에 접근해야 하므로 시스템의 과부하 현상이 발생할 수 있다는 문제점을 지니고 있다. 향후에는 시스템의 처리속도 뿐만 아니라, CRL 파일에 접근하는 횟수를 줄여 전체적인 시스템의 성능을

개선시킬 수 있는 연구를 진행할 것이다.

참고문헌

- [1] R. Mahadevan, "Analytical Modeling of Electrostatic Structures," *Proc. of IEEE Workshop on Micro Electro Mechanical Systems*, pp. 120-127, Feb. 1990.
- [2] C. Adams and S. Lloyd, *Understanding PKI*, Addison Wesley, 2002.
- [3] R. Housley and T. Polk, *Planning for PKI*, Wiley, 2001.
- [4] R. Housley, W. Ford, W. Polk, and D. Solo, *Internet X.509 Public Key Infrastructure: Certificate and CRL Profile*, IETF RFC3280, 2002.
- [5] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, *X.509 Public Key Infrastructure Online Certificate Status Protocol - OCSP*, IETF RFC 2560, 1999.
- [6] M. Myers, R. Ankney, and C. Adams, *Online Certificate Status Protocol, version 2*, IETF Draft, 2001.
- [7] M. Naor and K. Nissim, "Certificate Revocation and Certificate Update," *IEEE Journal on Selected Areas in Communications*, vol. 18, issue 4, pp. 561-570, Apr. 2000.
- [8] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*, 2nd Ed., Academic Press, 2000.
- [9] F. F. Elwailly and Z. Ramzan, *Quasi-Modo: More Efficient Hash Tree-Based Certificate Revocation*, Available at <http://www.docomolabs-usa.com/researchers/ZRamzan/quasi-modo-v1.pdf>, Sep. 2003.
- [10] *Hardware for Security*, Available at <http://www.jaik.tu-graz.ac.at/>.
- [11] *Deploying PKI Inside Microsoft*, Available at <http://www.microsoft.com/technet/itsolutions/msit/security/deppkiin.mspix>, 2003.