인증서 검증을 위한 프로토콜에 관한 연구

이옥경*,이용준, 정제동, 오해석
숭실대학교 컴퓨터학과
e-mail: oklee@lycos.co.kr

A Study on Protocol for Certificate Verify

Ok-Kyoung Lee*, Young-Jun Lee,
Jae-Dong Jung, Hae-Seock Oh
*Dept of Computer Science, Soongsil University

요약
최근 인증서에 대한 중요성이 높아지고 있으며, 이에 따른 많은 인증서에 관한 시스템이 개발되고 있다. 실시간 처리를 위해 OCSP(Online Certificate Status Protocol)가 제안되었으나, 네트워크의 제한으로 인하여 이용하는데 어려움이 있다. 본 논문은 이에 따라 부하를 줄이고, 좀더 효율적인 인증서 검증을 위해 방안을 제시하고, 이 시스템의 서버와 클라이언트 사이의 인증서 검증을 위해 필요한 request와 response에 대한 프로토콜을 제안한다.

1. 서론
최근 컴퓨터와 네트워크의 급속한 보급과 발전으로 인해 전자상거래의 인터넷 비즈니스가 보편화되고 있다. 이에 따라 지식 정보화 사회에서 각 분야의 비즈니스 환경에도 변화가 가져오고 있다. 예전의 오프라인으로 처리된 많은 일들이 지금은 온라인으로 신속하게 처리되고 있다. 하지만, 중요한 전자 문서가 네트워크 상에서 공개되어지기 때문에, 이 문서에 대한 위조와 변조 및 신분위장 등 각종 역기능에 의한 위협이 증가되고 있다. 이로 인하여 정보 보호 문제가 부각되었으며, 군사적 용도나 국가적 차원에서 주로 이용되었던 전자서명과 암호 기술이 전자상거래로 확대로 확장되었다. 정보의 안전성과 신뢰성을 위해 각종 분야에 공개키 암호기술을 적용하여 기밀성(confidentiality), 인증(authentication), 무결성(integrity), 부인방지(nonrepudiation)를 제공하는 PKI(공개키 기반구조: Public Key Infrastructure)를 구축하고 있다. 인증서를 사용함에 있어, 소유자 확인과정은 PKI기반에서 인증서 상태 검증을 통해서 이루어지게 되는데 인증서에는 개인키에 해당하는 공개키를 가지고 있기 때문에 소유자 확인이 가능하다. 그러나 사용자의 실수로 개인키(비밀키)가 노출되었거나 자격의 범람, 등의 이유로 인증서를 폐지해야 할 경우가 발생한다. 이런 경우 사용자는 수신한 인증서가 유효한 것인지지를 확인하는 절차를 받아야 하는데 이는 인증기관이 폐지된 인증서에 대한 정보를 공개하거나 사용자가 원하는 인증서의 상태를 인증기관에 의뢰할 기회 없으므로 알 수 있다. 이 인증서 폐지 여부를 검증하는 기존의 방법은 CRL(Certification Revocation List)이 일반적이다. 또한 실시간 인증 방법으로는 OCSP(Online Certification Status Protocol)가 제안되고 있다. 그러나 OCSP는 요청되는 인증서 검증에 대하여 인증하여 보내는 절차를 가지게 되므로 네트워크의 부하를 피할 수 없다. 그 외에 본 연구는 OCSP의 이러한 부하를 줄이는 방안으로 실시간 인증서 검증하는 방안을 제안하고 있는 시스템에 알맞은 서버와 클라이언트 사이의 통신 프로토콜을 제안한다.
2. 관련 연구

실시간 인증서 검증 점을 위한 방안으로 가장 많이 알려진 것은 OCSP(Online Certificate Status Protocol)이다. OCSP는 사용자가 서버에게 온라인으로 인증서 상태 검증을 요구하는 것으로서, 서버는 요구받은 인증서의 상태 상태를 하여 Response를 사용자에게 준다. 또한 사용자가 인증서를 폐지하게 되면 즉시 반영하여 온라인으로 검증 서비스하는데 차질이 없도록 한다. 하지만 네트워크 부하가 심하다는 단점을 가지고 있어 개선점이 필요하다. 이에 이 논문에서는 이러한 네트워크 부하를 줄일 수 있는 방안으로, 데이터베이스를 이용한 서버와 클라이언트를 개념을 도입한 보다 효율적이면서 부하를 줄이는 방안을 제안하고, 이를 적용하기 위한 프로토콜에 대해 연구한다. [1].

2.1 OCSP

OCSP는 인증서 상태에 대하여 실시간의 정보를 제공해 주는 대표적인 것이다. 사용자가 이용한 전자서명을 검증하기 위해 거래한 은행이나, 전자상거래 상점에서는 OCSP 서버에게 해당 인증서의 상태 정보를 요청한다. 요청을 받은 OCSP 서버는 CA(Certificate Authority)와의 요청을 통해 실시간의 인증서 상태 정보를 전달해 준다. 인증서에 대한 검증을 요청한 은행이나 상점은 OCSP 서버로부터 응답이 오지 않아 요청한 인증서 상태 확인을 대기한다. [그림 1]은 OCSP의 구성도를 나타낸다. [그림 1]에서 검증자는 인증서 상태를 검증해야 하는 상점이나 은행 등을 의미한다.

[그림 1] OCSP 구성도

OCSP의 제안은 실시간 인증서 상태 정보를 제공함으로써 기존의 CRL, Delta-CRL, Freshest CRL의 현제성 문제점을 해결하는데 주목적이 있다. 그러나 OCSP는 모든 전자서명 검증 시점에서 CA에게 인증서 상태를 조회하기 때문에 여러 가지 문제가 발생한다. 즉, 현제성은 보장이 되지만 검증 속도의 성능에 저하가 가져오기 때문에 많은 검증 속도가 요구되는 금융거래에서는 부담이 되고 있다.

3. 제안하는 인증서 검증 프로토콜

3.1 실시간 인증서 검증 방안

실시간 검증 방안의 가장 대표적인 것이라고 할 수 있는 OCSP는 검증 할 때마다 인증서에 대한 CA 확인 작업과 실시간에 따른 부하를 인하여 네트워크의 부하를 초래한다. 이에 본 논문은 논문[1]에서 제안한 실시간 인증서 검증 방안인 서버와 클라이언트 개념을 이용한 방안을 이용한다.

[그림 2] 실시간 인증서 서비스 구성도

[그림 2]는 실시간 인증서 검증을 위해 이용하는 시스템의 구성도이다. [그림 2]와 같이 인증서 발급 은 CA에서 이루어지고 이 인증서를 이용하여 거래를 하게 되는데 [그림 2]에서 보이는 검증자가 상점이나, 거래처를 의미한다. 이하에 검증자는 사용자에게 인증서를 받게 되며 이하에 받은 인증서에 대한 검증을 통하여 거래를 성립시킨다. 이러한 검증 작업은 앞에서 설명한 것처럼 다양한 방법을 이용할 수 있으나 본 논문에서는 실시간 검증방법으로 클라이언트와 서버에 데이터베이스 개념을 도입한 방법을 이용한다. 즉, 가려져 있는 검증자는 자체에 가지고 있는 클라이언트의 데이터베이스를 이용하여 검증을
하게 된다. 이때 검증을 실패할 경우, 즉, 사용자가 클라이언트에 있는 사용자가 아닌 경우 인증서를 받 은 CA의 서버로 검증을 요청하게 된다.

[그림 3] Server와 Client

이렇게 요청된 검증 작업은 서버의 데이터베이스에 존재하는지와 존재하면 그 상태에 대한 것을 검증하 여 클라이언트에 전송한다. 또한, 사용자의 요청으로 CA에 의해 폐지되었거나, 등록된 인증서는 서버를 통해 그 인증서가 이용되는 도메인내의 클라이언트 에 push 된다. 그림에 따라서 서버에 각 클라이언트 에 대한 정보를 가지고 있어야 하며, 인증서와 연결 할 수 있는 형태로 데이터베이스가 구성되어있어. [그림 2]에서 보는 것은 실시간 인증 서비스의 간단 한 구성도를 의미한다. 설계적으로 작용할 때는 검 증자와 즉, 금융권이나, 거래처와 같은 곳에서 이용될 때는 검증자가 결과 그림처럼 하나로 이루어진 것이 아니라 [그림 3]과 같이 다수가 된다.

3.2 제안하는 프로토콜

이 장에서는 이러한 인증서 검증 방안의 서버와 클라이언트 사이에 통신을 위한 프로토콜을 제안한 다. [그림 4]와 같이 서버와 클라이언트 사이에 발생 하는 처리는 클라이언트에 요청에 의한 질의응답과 인증서 변경에 따른 정보를 제공하는 작업으로 나 눌 수 있다.

[그림 4] Server와 Client

[그림 5]는 서버와 클라이언트의 데이터베이스에 관한 설명이다. 서버의 데이터베이스는 두개의 데이터 블록으로 이루어져있으며, Domain Table에는 Client의 도메인을 가지고 있어서 분석된 인증서의 serial number에 해당하는 도메인에 정보를 push 해주는 기능을 수행할 수 있게 해두었다. 서버의 Domain Table에 Dom_name은 serial number에 해당하는 인 증서가 실제적으로 사용되는 도메인의 정보를 포함 하는 데이터베이스에 해당하는 것으로, 폐지되거나 상태를 push해 줄 때 이동되는 데이터이다. 또한, Certificate Table 같은 경우, 인증서에 대한 정보를 저장하게 된다. 인증서의 고유번호와 Time으로 정 의된 인증서의 유효기간을 나타내는 This date, Next date, 그리고 상태를 나타내는 Status 등을 하나의 데이터로 하고 있다. 또한, 클라이언트는 서버 와 별개의 데이터베이스를 가지고 있어서에서 받 은 정보를 데이터베이스에 삽입하므로 요청되는 인 증서 검증에 대한 서비스를 자체적으로 검증하게 된 다. 이렇게 하므로, 인증서 검증에 대한 작업을 분산 처리하게 되고, 한방 인증서를 지침한 것은 CA에서 인 증된 것이므로 다시 인증을 하는 번거로움을 줄일 수 있다. 그래서 부하를 줄여줄 수 있음을 실험을 통해 알 수 있다[2].

[그림 5] 클라이언트와 서버 데이터베이스 구성도

이 시스템에서 사용되는 프로토콜은 크게 두 가지 로 나눌 수 있다. 하나는 클라이언트에서 서버에 요청하는 프로토콜이다. 나머지 하나는 서버에서 클라이언트에 응답할 때 발생하는 통신 프로토콜이다. 다음 [그림 6]에는 이 프로토콜에 관한 것이 다. 이 프로토콜은 인증서와 사이즈가 작아야 하 며, 그 인증서를 대표할 수 있는 serial number와 DN (Distribute Name), 그리고 도메인을 대표할 수 있는 도메인의 유의미의 이름과 그 도메인을 증명할 수 있는 어떠한 값을 갖는 프로토콜을 이용하여 통신하게 된다.

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>DN</th>
<th>Status</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server에서 Client로</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Who</th>
<th>Serial Number</th>
<th>DN</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client에서 Server로</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

또한, 서버에서 클라이언트로 보내는 경우는 두 가지가 있는데 인증서의 상태가 변경되어 CA에 변경된 정보를 받은 경우와 클라이언트에서 요청한 것에 대한 응답이 있다. 처음과 같은 경우에는 데이터베이스의 Domain Table에서 정보를 가져와 그 인증서에 필요할 도메인에 정보를 push 해주고, 다른 경우는 그 도메인에만 응답을 보내주면 된다. 이 두가지 모두 [그림 6]에 보여주고 있는 프로토콜을 이용할 수 있다. 이것은 인증서에 대한 Serial number와 DN, 그리고 요청한 인증서에 대한 상태 정보를 클라이언트에게 전송하게 된다. 이렇게 전송된 정보는 클라이언트 데이터베이스에 삽입하여 다음에 같은 사용자가 접속을 요청하게 되면 클라이언트 내에서 처리할 수 있게 해준다.

4. 결론 및 향후 연구
인증서의 중요성이 높아가고 있는 요즘, 실시간 검증이 어울 별보도 중요하다. 그러나 실시간 처리에 여러 가지 문제가 발생할 수 있으므로, 이를 극복할 수 있는 다양한 방안이 필요하게 된다. 이를 위해 이 논문은 실시간 검증을 한 발 더 앞당길 수 있는 방안으로 분산처리로 부하를 줄이고, 중복 처리를 줄일 수 있는 방안을 제안하였다. 또한, 모든 것을 처리할 수 없을 경우에 고려하여 서버를 따로 두고, 관리할 수 있게 해준다. 하지만 이 시스템은 서버와 클라이언트의 신뢰를 어떻게 형성해야 하는지가 우선 해결되어야 할 문제이며, 통신상의 장애로 인하여 서버에서 클라이언트로 push 되지 않은 데이터를 어떻게 처리해야 할 문제에 대한 문제가 발생한다. 이러한 문제들은 서버와 클라이언트의

참고 문헌

[1] 이용준, 정제동, 오해석, “금융거래 서버스 제공자의 향상된 검증속도를 위한 인증서패키 전달 시스템”