비계층적 PKI에서의 인증 경로 처리 기법에 대한 새로운 방안

최연희° 박미옥 추연수 전문석
숭실대학교 컴퓨터 학과
lovejung22°@naver.com
mjun@computing.ssu.ac.kr

A New Approach of Certificate Path Validation Scheme in Non-Hierarchical PKI

Yeonhee Choi° Miog Park Yeoun Soo Choo Moonseog Jun
Dept. of Computing, Soongsil University

요 약

사용자 어플리케이션에서 인증 경로를 검증하는 일은 매우 복잡하고 많은 시간이 소요됨으로서 사용자 측에 상당한 부담을 줄 수 있다. 특히, 비계층적 PKI에서의 검증 작업은 다수의 후보 경로가 존재함에 따라서 그 부담이 더욱 커진다. 따라서 본 논문에서는 비계층적 PKI의 인증 기관(Certificate Authority : CA)을 서명 검증 작업에 참여시킴으로서 사용자 측 검증 작업에 대한 부담을 감소시킬 수 있는 새로운 형태의 인증 경로 처리 방안을 제안하였다. 제안한 방안은 서명 검증에 수반되는 사용자 측 암호화 작업을 크게 줄임으로서 검증에 대한 부담을 감소시킬 뿐만 아니라 CA들의 검증 작업의 참여로 인해 정적인 CA들의 활용도를 높일 수 있다.

1. 서 론

고도의 정보화 사회가 도래하면서 인터넷을 비롯한 정보 통신 기술이 급속하게 발전되었다. 이로 인해 인터넷을 이용한 전자 상거래와 같은 상업적 서비스가 널리 사용되면서 메시지의 무결성과 네트워크를 통하여 상대방의 신분 확인을 수행할 수 있는 인증 기술과 같은 정보 보호의 중요성이 점차 증대되고 있다. 현재 전자상거래 등 보안을 요구하는 정보 보호 시스템은 대부분 공개키 기반 구조 (Public Key Infrastructure : PKI)를 사용하고 있다. PKI구축을 위한 가장 중요한 기술 중의 하나는 사용자 편의성과 시스템의 효용성을 동시에 고려하는 인증서 검증 기술이다. 인증서의 검증 기술은 거래하고자 하는 상대방의 인증서 및 공개키가 올바른 것인지를 확인하기 위한 실제 전자 거래의 유효성에 관한 것이므로 신중하게 처리되어야 한다.[1].

인증서를 검증을 위해서는 특정 인증서에 대한 인증 경로를 구성해야 하고, 특정 인증서의 취소 상태를 인증서 취소 목록(Certificate Revocation List : CRL)[2] 또는 Online Certificate Status Protocol (OCSP)[3] 서버와 같은 제 3의 신뢰 기관을 통하여 확인해야 하며, 인증 경로상의 인증서들에 대한 유효성을 검증하기 위한 다양한 검사를 수행해야 한다. 특히 인증서의 서명의 유효성을 검증하기 위해서는 경로상의 모든 인증서들에 대해 공개키 서명 알고리즘을 통한 암호학적 연산을 행해야 하기 때문에 시간적으로 매우 비효율적인 작업이다. 이러한 모든 검증 작업을 사용자 어플리케이션에서 수행하게 되면 어플리케이션의 기능이 무겁게 되어 궁극적으로 PKI 및 인증서의 이용 확산을 저해하는 가장 큰 원인이 된다.

이러한 문제를 해결하기 위한 온라인 검증 서버의 도입을 위한 OCSPv2[4], Simple Certificate Validation Protocol(SCVP)[5]와 같은 다양한 온라인 프로토콜들이 제안되어 왔다. 온라인 검증 서버의 채택은 사용자 측의 인증서 검증 모듈을 단순화하고 검증으로 인한 연산적인 부담을 크게 축소시키는 장점을 제공하는 반면, 검증 서버를 위한 부수적인 시스템 도입의 필요성, 온라인 상태 유지, 서버와 사용자 사이의 키 분배 및 상호 인증서 검증을 위한 메커니즘의 필요성 등의 까다로운 문제가 수반될 수 있다.

따라서, 본 논문에서는 검증 서버를 따로 구축하지 않고서도 사용자 측의 검증 작업의 수행으로 인한 부담을 감소시킬 수 있는 새로운 인증 경로 처리 방안을 제안하였다. 제안한 방안은 서명 검증 작업을 비 계층적 PKI 영역내의 다른 CA들에게 위임함으로서 사용자 측 부담을 줄이고 상호 인증서나 하위 사용자들의 인증서 발행 및 관리 등의 작업 외에는 정적인 상태로 있는 CA들을 동적으로 활용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 rfc2459에서 제공하는 인증서 검증 과정을 간단히 소개한다. 3장에서는 제안한 방안을 자세히 기술하고, 4장에서는 제안한 기법을 rfc2459의 인증서 검증 방식[2]과 비교, 분석한다. 마지막 5장에서는 결론을 내린다.

2. Rfc2459 인증서 검증 과정

타겟의 공개키 인증서 및 공개키가 올바른지를 검증하기 위해서는 검증자 자신의 신뢰 CA와 타겟 사이의 인증 경로가 존재하고 이 인증 경로가 올바른지를 확인하기 위한 경로 설정 및 검증 작업을 수행해야 한다. rfc 2459 알고리즘은 검증을 위한 기본 알고리즘으로 주로 사용된다.

Rfc 2459의 인증서 검증 과정은 크게 경로 설정, 기본 검증, 경로 검증 등의 3가지 과정으로 수행된다[6]. 그림 1은 이들의 과정을 간단히 도식화한 것이다.

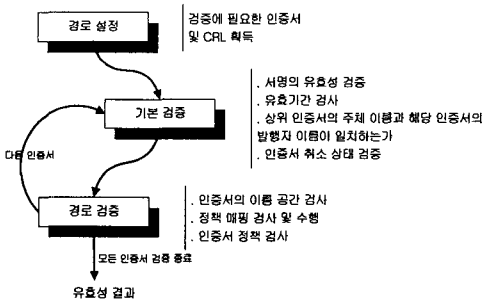

그림 1. 인증서 검증 과정

그림 1에서 보이는 것과 같이, 검증자는 경로설정에 필요한 모든 인증서 및 CRL들을 수집하여 적합한 인증 경로를 설정하고 설

정된 경로에 대한 기본 검증 및 경로 검증을 수행한다. 이 때 검증 작업은 경로상의 모든 인증서들에 대해 수행되며, 각 인증서들은 검증을 위한 조건들 중의 하나라도 만족하지 못하면 설정된 인증 경로는 유효하지 않은 것으로 판단되어 다른 후보 경로를 새롭게 설정하고 이를 입력으로 받아들여 검증을 다시 시작한다. 여기서, 각 인증서와 CRL들을 획득하기 위한 다운로드시간을 무시할 때, 검증 시간은 크게 CRL을 검색해서 이의 서명을 검증하는 시간과 그 외의 나머지 검증 과정을 수행하는 시간으로 구성된다. 이들을 각각 t_{sign} 과 t_{verify} 라고 가정 했을 때, 이 알고리즘에 따른 경로 n의 검증 시간은 $t_{sign} + t_{verify} = O(2nt)$가 될 것이다. 즉, 하나의 인증서를 검증하는데 걸리는 2t의 시간이 n번 반복되는 것이다. 즉, 이것은 하나의 경로를 검증하는데 걸리는 시간이고, 하나 이상의 후보 경로가 존재하는 비계층적 PKI에서는 유효한 경로가 발견될 때 까지 반복해서 수행되기 때문에 훨씬 많은 시간이 요구될 것이다.

서명 검증시에 특히 암호화 계산 시간은 해쉬 계산 시간보다 알고리즘에 따라 대략 100-10000배 정도 더 오래 걸림으로서 부담의 주된 원인이 된다[6].

3. 제안한 기법

3.1 서명의 위임

제안한 기법은 비 계층적 PKI 에서의 수행을 기본으로 하고, 각 CA는 자신을 신뢰 CA로 하는 하위 사용자들과 상호 CA들의 공개키 및 비밀 키를 생성하여 분배하는 것을 기본으로 하며, 각 CA는 같은 키를 사용하여 인증서와 CRL에 서명한다고 가정한다.

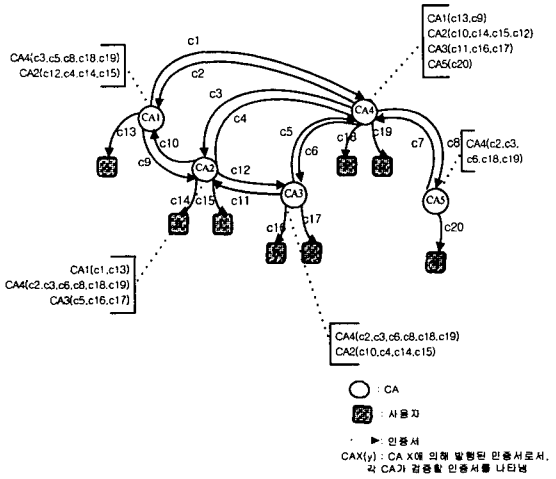

그림 2. 각 CA의 서명 검증할 인증서의 위치

따라서 비 계층적 PKI 영역내의 모든 CA들은 자신이 발행한 인증서 주체인 사용자나 상호 CA들의 공개키를 알기 때문에 이들이 발행한 인증서들의 서명을 검증할 수 있다. 그림 2는 하나의 비 계층적 PKI 영역에서 각 CA들이 서명 검증할 인증서들의 위치를 나타낸다.

그림 2에서, 각 CA는 자신이 인증서를 발행한 상호 CA와 하위 사용자들의 공개키를 알기 때문에 이들이 발행한 인증서들의 서명 검증을 수행한 후 이 작업의 수행 결과인 서명 검증 정보인 DSVD를 발행하여 공개해야 한다. 예로서, CA1은 상호 CA인 CA4와 CA2의 키를 알기 때문에, CA4가 발행한 c3,c5,c8,c18,c19와 CA2가 발행한 c4,c12,c14,c15를 검증하여 이들에 대한 DSVD를 발행할 수 있다.

제안한 기법에서는, 검증자가 자신의 신뢰 CA의 공개키를 알기 때문에 신뢰 CA가 발행한 인증서들의 서명은 검증자가 직접 검증하고, 신뢰 CA 의 자체 서명된 인증서에 대한 서명 검증은 수행하지 않는 것을 기본으로 한다.

3.2 DSVD

자신의 상호 CA들이 발행한 인증서들에 대한 서명을 검증한 각 CA들은 검증한 결과로서 DSVD를 생성하여 이를 공개한다. DSVD는 각 CA가 인증서의 서명 검증한 결과를 자신의 비밀 키로 서명한 정보로서 그림 3의 구조를 가진다.

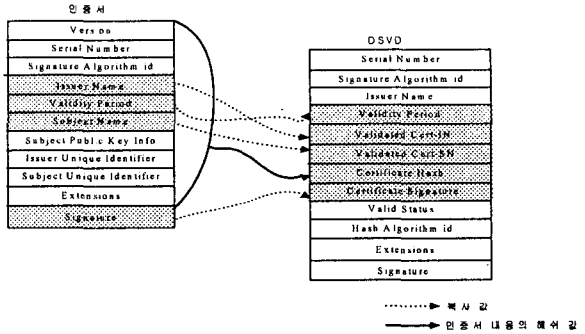

그림 3. DSVD 구조

◎ Serial Number : DSVD를 식별하기 위한 발행자에게 유일한 일련번호.
◎ Signature Algorithm id : DSVD를 서명하는데 사용한 알고리즘 식별자.
◎ Issuer Name : DSVD 발행자의 X.500 식별 이름.
◎ Validity Period : DSVD의 유효기간으로서 서명 검증한 주체 인증서의 유효기간과 같게 설정된다.
◎ Validated Cert-IN: 검증한 인증서의 Issuer Name.
◎ Validated Cert-SN: 검증한 인증서의 Subject Name.
◎ Certificate Hash : 검증한 주체 인증서의 내용에 해쉬를 계산한 값으로서, 주체 인증서의 무결성을 검증하기 위해 사용한다.
◎ Certificate Signature : 검증한 주체 인증서의 서명 값으로서 주체 인증서의 합법성을 검증하기 위해 사용한다.
◎ Valid Status : 서명 검증 상태를 나타내고, 이는 "valid" "invalid" 의 2가지 형태로 표시된다.
◎ Hash Algorithm id : Certificate Hash를 생성하는데 사용한 해쉬 알고리즘 식별자.
◎ Extensions : 부가적인 정보를 입력하기 위한 확장자
◎ Signature : DSVD 내용에 대한 발행자의 서명 값.

여기서, Validated Cert-IN과 Validated Cert-SN 항목은 검증자로 하여금 경로상의 인증서에 해당하는 올바른 DSVD를 검색하도록 하고, DSVP 수행시에도 CA들로 하여금 검증자가 요구하는 올바른 DSVD를 첨부하도록 하는데 이용된다.

각 CA는 하위 CA들의 행위를 감시하여 이들의 새로운 인증서의 발행, 취소, 갱신 등에 따라 새로운 DSVD의 생성 및 기존 DSVD의 내용을 갱신해야 한다. 이 때, 제안한 기법은 CA의 DSVD와 관련된 모든 작업에 대해 신속함을 요구하지는 않는다. 이러한 작업은 CA의 idle time에 수행하도록 함으로서 CA들로 하여금 연산적인 부담이 최소한으로 감소될 수 있도록 하였다.

검증시에 검증자는 필요한 DSVD들을 수집하여, 기존의 암호화 계산을 통한 서명 검증 작업 대신 그림 4의 DSVD를 이용한 검증 작업을 통해 서명을 검증함으로서 서명 검증한 인증서 내용의 무결성과 서명의 합법성을 검사하여 DSVD의 내용이 원래의 인증서에 대한 서명 결과인지를 확인하게 된다.

```
Published_ Hash □ H[C(cont)]
Calculated_ Hash □ H[C(cont)]
Published_ Signature □C(Sig)
Original_ Signature □ C(Sig)
IF Published_Hash = Calculated_Hash AND
   Published_ Signature = Original_ Signature THEN
      C becomes verified
ELSE
      C has not been verified
```

그림 4. DSVD를 통한 서명 검증 과정

비록 그림 4의 검증을 통해 서명 검증한 주체 인증서에 대한 무결성과 합법성을 확인한다 하더라도 검증자는 DSVD의 검증 결과나 DSVD를 발행한 CA들이 신뢰할 만한 것인지는 확신할 수 없다. 왜냐하면 위의 검증만으로는 DSVD가 정당한 CA로부터 발행되었는지, 또는 DSVD의 내용이 제 3자에 의해 변조되었는지를 확인할 수 없기 때문이다. 따라서 이를 확인하기 위한 부가적인 검증 작업이 수행되어야 할 것이다. 이 검증 작업은 DSVP를 통해 이루어지고 이 DSVP를 통한 검증 과정이 성공적으로 완료되기 전까지는 사용자는 인증서의 유효성과 관련한 어떠한 결정도 내릴 수 없다.

3.3. DSVP

DSVP는 검증자와, 타겟의 인증서를 발행한 CA를 제외한 경로상의 모든 CA들 사이에 수행되는 프로토콜로서 검증자는 경로상의 DSVD를 발행한 주체인 CA들을 거쳐 수집된 DSVD들의 검증을 통해 DSVD의 무결성 및 서명의 합법성을 검증한다. DSVP의 수행은 다음의 3개의 과정으로 구성된다.

1st. 검증자가 신뢰 CA에게 DSVD를 요청하는 과정 : 검증자는 DSVP 경로 설정을 위해 수집된 DSVD들로부터 타겟 인증서에 대한 DSVD를 발행한 TDCA(Target-DSVD-issued CA)를 확인한 후, 신뢰 CA에게 경로상의 모든 DSVD들을 전송해달라는 요청문인 DSVDRequest를 보낸다. 이 때 DSVDRequest에는 설정된 경로로부터 확인된 DSVP 경로상의 CA들의 id, 즉 수집되어야 할 DSVD들이 존재하는 TDCA로부터 신뢰 CA까지의 일련의 CA-id들이 포함되어야 한다.

2nd. 경로상의 DSVD를 발행한 모든 CA들이 RelayMessage (RM)에 DSVD를 첨부하는 과정 : 신뢰 CA는 TDCA에게 타겟에 대한 DSVD의 첨부를 요구하는 메시지인 RM을 생성하여 전송한다. TDCA는 타겟에 대한 DSVD를 RM에 첨부하여 DSVP 경로상의 다음 CA에게 전송하고, 이 RM의 전송 및 DSVD첨부 과정은 신뢰 CA로 되돌아올 때까지 경로상의 다음 CA에게 순차적으로 전송된다. 각 CA는 RM을 통해 전송된 DSVP 경로 상 이전 CA가 발행한 DSVD를 검증하여 저장함으로서 부인 봉쇄 기능 또한 제공한다.

3rd. 신뢰 CA가 검증자에게 응답하는 과정 : 신뢰 CA는 RM을 수신한 후, RM을 통해 수집된 모든 DSVD를 포함한 DSVDResponse를 생성하여 요청한 검증자에게 돌려준다.

그림 5는 검증자 V가 타겟 T에 대한 DSVD를 요청할 때의 DSVP 수행 경로를 보인다. 그림 5에서 보이는 것처럼 T의 인증서를 발행한 CAn-1을 제외하고, CA1부터 TDCA인 CAn-2까지 모든 CA들이 DSVP의 수행에 참여하며, 신뢰 CA인 CA1으로부터 TDCA인 CAn-2를 거쳐 DSVP 경로상의 CA로 순차적으로 전송된다. 즉, DSVP의 수행 경로는 검증자 V의 요청으로부터 시작하여 CA1→CAn-2→.....→CA2→CA1이 될 것이다. 따라서 한번의 요청 시 n-2개의 RM이 전송되어 n-2개의 DSVD들이 최종 수집된다. RM의 전송을 통해 수집된 DSVD들은 DSVDResponse에 포함되어 검증자에게 전송되며, 검증자는 전송된 DSVD들과 원래의 DSVD들의 값을 비교함으로서 DSVD의 내용의 무결성과 서명의 합법성을 검증하여 CA들의 신뢰성을 확인하게 된다.

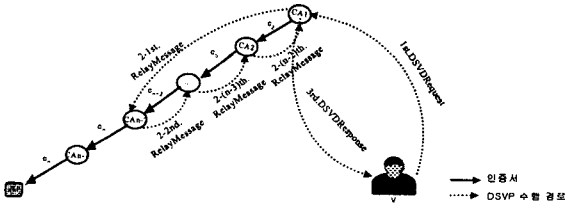

그림 5. DSVP 수행 경로

4. 비교 및 분석

제안한 방식의 분석을 위해, 검증을 위해 필요한 인증서, CRL, DSVD들을 획득하기 위한 다운로드시간을 무시하고, CRL을 검색해서 이의 서명을 검증하는 시간과 그 외의 나머지 검증 과정을 수행하는 시간을 각각 t_{sign} 과 t_{verify}라고 가정한다. 서명 검증 시에 해쉬 계산 시간은 암호화 계산 시간보다 훨씬 적게 소요되기 때문에, 해쉬 계산 시간 또한 무시한다고 했을 때, 제안한 알고리즘에 따른 경로 n의 최상의 검증 시간, 즉, 경로상의 모든 인증서에 대한 DSVD가 존재할 때의 검증시간은 $t_{sign} + t_{verify} = O(nt)$가 될 것이다. 왜냐하면, t_{verify} 의 시간이 오직 한번의 암호화 계산만으로 이루어지기 때문이다. 따라서, 제안한 방식은 시간과 연산적 부담 면에서 기존 방식보다 최대 2배 정도 더 좋은 성능을 얻을 수 있게 된다. 표 1은 제안한 검증 방식을 rfc 2459의 검증 방식과 비교한 것이다.

표 1.rfc2459를 이용한 기존방식과 제안한 방식의 비교

	rfc2459 방식	제안한 방식
수집 데이터	인증서,CRL	인증서,CRL,DSVD
서명검증을 위한 암호화 횟수	n번	1번
검증 소요 시간	*O(2nt)*	*O(nt)*
CA의 활용도	Low	High
사용자측 부담	High	Low
검증 딜레이	X	DSVDResponse수신 시간

표 1에서 보인 것처럼, 제안한 알고리즘의 가장 큰 문제는 DSVDResponse 의 응답을 기다림으로서 검증의 딜레이가 발생할 수 있다는 것이다. 그러나, 제안한 방식에서는 응답이 정해진 시간 안에 들어오지 않을 경우에는 검증자가 암호화 계산을 통해 직접 검증하도록 하기 때문에 검증 작업 자체에는 그다지 큰 영향을 끼치지는 않을 것이다.

5. 결론

본 논문에서는 비 계층적 PKI에서 인증서의 서명 검증 작업을 CA들에게 위임함으로서 사용자 측의 복잡한 인증 경로에 대한 검증 작업에 대한 부담을 줄이고 CA들의 활용도를 향상시킬 수 있는 새로운 인증 경로 처리 방안을 제안하였다. 제안한 방식은 서명 검증 작업에 정적인 상태의 CA들을 참여시킴으로서 성능 좋은 CA들을 적극적으로 활용하도록 하였으며, 이것은 사용자 측의 부담을 덜어주는 결과를 가져왔다.

그러나, DSVP의 잦은 수행으로 인한 전송 량의 증가와 DSVDResponse의 수신과 관련해 발생할 수 있는 딜레이를 해결할 수 있는 추가적인 방안은 앞으로 연구되어야 할 것이다.

4. 참고문헌

[1]심희원, "DNS를 이용한 상호 연동 및 인증서 검증 http://www.kisa.or.kr/K_trend/KisaNews/200201/focus.html

[2]R. Housley, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", IETF RFC 2459, January 1999.

[3] M. Myers, "X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol-OCSP", IETF RFC 2560, June 1999.

[4]M. Myers, A. Malpani, D.Pinkas, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol, version 2" IETF draft-ietf-pkix-ocspv2-text-01.txt, December 2002.

[5]Ambarish Malpani, Paul Hoffman, Russ Housley, and Trevor Freeman, "Simple Certificate Validation Protocol(SCVP)", IETF draft-ietf-pkix-scvp-06.txt, July 2001.

[6]Albert Levi, M.Ufuk Caglayan,"Analytical performance evaluation of nested certificates", Performance Evaluation , vols. 36-37, p213-232, August 1999.