전체 인증경로를 알 수 있는 인증서 일련번호 부과 방법

박 재 관, 김 광 조
국제정보보호기술연구소
한국정보통신대학원대학교

A New Scheme of Assigning Serial Number for Full Path Validation

Jaegwan Park, Kwangjo Kim
International Research center for Information Security(IRIS)
Information and Communications Univ.(ICU), Korea
{jgpark, kkj}@icu.ac.kr

요 약

인증서는 공개키(public key)와 그의 소유주를 대응(mapping) 시켜주는 문서이다. 그리고, 그 대응의 적법성을 검증하기 위하여 검증자(verifier)는 소유자(owner)의 인증서와 그 인증서를 인증한 공인인증기관(Certificate Authority: CA)의 인증서들을 검증하여야만 한다. 그러나, 현재의 인증서의 일련 번호로는 그 전체 인증서 검증 경로를 미리 한번에 파악할 수 없다. 따라서, 인증서를 하나씩 검증을 하여야만 한다. 본 논문에서는 그 전체 인증경로를 미리 한번에 파악할 수 있도록 인증서 일련번호를 부과하는 방법을 제안한다. 이를 이용하면, 인증서를 병렬적으로 검증하여 그 검증 시간을 단축시키는 장점과 함께 공인인증기관의 식별자를 정책에 따라 다르게 지정한다면 정책의 대응에도 효과적으로 이용할 수 있다.

I. 서 론

인터넷이 보편화되고 활성화되면서, 인터넷을 이용한 전자상거래의 규모도 커지고 점점 복잡해지고 있다. 인터넷을 활용한 전자상거래의 증가는 전자거래 문서의 인증성, 무결성,

부인방지성 등의 기능 제공이 용이하게 하는 공개키 기반구조에서의 인증서의 사용이 그 활력소가 되고있다.

공개키 기반구조에서 사용되는 현재 인증서(certificate)는 X.509v3이다. 이 인증서는 공개키(public key)와 그 소유자(owner)를 대응(mapping) 시켜주는 문서이며, 또한 공인된 공인인증기관(CA)이 서명을 함으로써 진위성이 확인된다.

그러나, 인증서는 전자적으로 생성되고 보관되는 것이므로 이를 신뢰하는 것, 즉 검증자(verifier)가 소유자의 실체를 신뢰하는 것이 무엇보다도 중요하다. 이를 위하여, 검증자는 소유자의 인증서로부터 자신이 신뢰할 수 있는 공인인증기관까지의 경로를 확인하고, 그 경로에 있는 모든 인증서들을 검증함으로써 그 신뢰관계를 새롭게 생성한다.

현재의 X.509v3 인증서에는 일련번호(serial number), 소유자 이름(subject name), 인증기관 이름(issuer name) 등이 포함되어 있다. 검증자는 소유자 이름이 실제로 인증서에 있는 공개키에 대응하는 비밀키를 소유하고 있는 지를 확인하고, 이 인증서가 법적인 효력(정당성)을 가지고 있는 지를 검증하기 위하여, 공인된 인증기관이 인증서를 인증했는지를 검증하기 위하여서는 공인인증기관의 인증서를 다시 받아야만 한다. 또한, 이 공인인증기관을 신뢰할 수 없을 경우, 위에서처럼 다시 이 공인 인증기관의 인증서를 인증한 상위 공인 인증기관에 대한 인증서를 받아서 검증할 필요가 있다.

즉, 인증서 검증 경로는 단계별 인증서의 검증을 통하여 드러나게 되어 있으며, 검증자는 순차적으로 인증서 검증 경로에 있는 인증서를 검증하고, 자신이 신뢰할 수 있는 인증기관이 나올 때까지 반복 검증하여야 한다. 여기서는, 인증서 검증 경로를 한번에 파악할 수 없기 때문에 인증서의 검증을 병렬적으로 한꺼번에 처리하는 것은 불가능하다.

본 논문에서는 이 모든 인증경로를 한번에 파악할 수 있는 인증서 일련번호를 부과하는 방법을 새롭게 제시하고자 한다. 이렇게 모든 인증경로를 한번에 파악할 수 있기 때문에 인증서의 검증을 병렬적으로 한꺼번에 처리할 수 있다.

본문의 구성은 다음과 같다

제 2장에서는 일련번호 및 인증서 검증 경로에 대한 정의 및 기존의 연구에 대하여 기술하고, 제 3장에서는 새로운 일련번호를 부과하는 방법 및 제안되는 일련번호를 사용하여 인증서 검증 경로를 어떻게 파악하는 지를 설명한다. 제 4장에서는 결론 및 향후 과제를 기술한다.

II. 관련연구

1. 일련번호

일반적으로 문서에 기록되는 일련번호는 문서를 구분하기 위한 번호로써, 그 유일성이

보장되어야 한다. 인증서에서도 동일하다. 인증서에 기록되는 일련번호는 그 유일성을 지니고 있다. 그러나, 이 유일성이라는 것이 전 세계의 모든 인증서와 비교하여 그 유일성을 보장하는 것이 아니고, 그 인증서를 인증한 공인인증기관 안에서 그 유일성을 보장한다.

즉, 같은 공인인증기관이 발급한 두 개의 인증서가 서로 같은 일련번호를 가지고 있다면, 이는 그 두 개의 인증서는 같은 인증서이며, 따라서, 공인인증기관의 이름과 인증서의 일련번호를 연결시킨 것은 유일하다.[1]

[1]에서는 일련번호를 다음과 같이 정의하고 있다.

"The serial number is an integer assigned by the CA to each certificate. It MUST be unique for each certificate issued by a given CA (i.e., the issuer name and serial number identify a unique certificate)."

또한 [3]에서는 다음과 같이 정의하고 있다.

"Serial Number : Unique number of the certificate, assigned by the issuer."

[3]에서는 일련번호의 길이를 12 바이트로 제한하고 있다.

2. 인증서 검증 경로

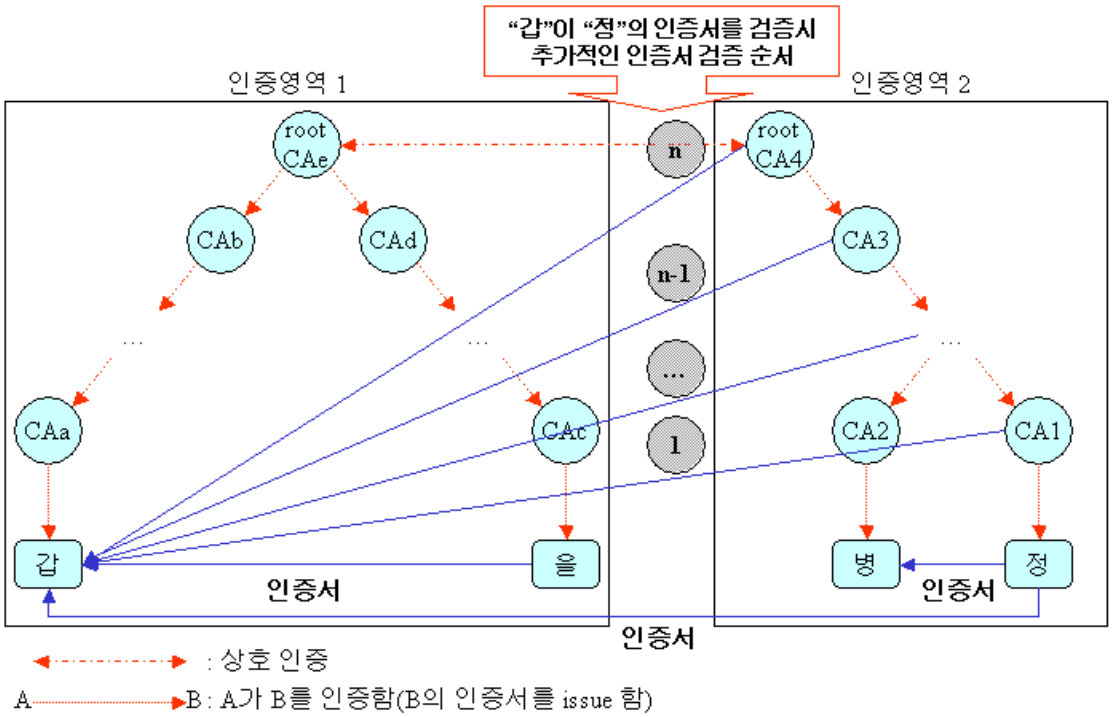
[1]에서는 인증경로 검증 알고리즘이 기술되어 있는데 최상위 공인인증기관으로부터 종단 공인인증기관까지의 인증관계를 복잡한 알고리즘을 통하여 사용자 시스템이 순차적으로 모두 검증하도록 되어 있다. 인증경로 검증의 목적은 인증서에 기술되어 있는 사용자 정보(Subject Distinguished Name)와 사용자 공개키와의 연결관계를 모두가 신뢰하는 최상위 공인인증기관의 공개키에 기반하여 검증하고자 하는 것이다.

인증서 검증 경로가 아래의 조건을 만족하면서 n개의 인증서가 있다고 가정하자.

- {1,(n-1)}에 속해 있는 모든 x에 대하여, 인증서 x의 소유자는 인증서 x+1을 인증한다.
- x=1인 인증서는 자기 스스로가 인증한다.
- x=n인 인증서는 최종의 마지막 인증서이다.

이 경우 검증자는 인증서 n을 검증하기 위해서는, 인증서 n을 인증한 CA의 인증서 n-1이 필요하다. 마찬가지로, 인증서 i를 검증하기 위해서는, 인증서 i를 인증한 CA의 인증서 i-1가 필요하다. 즉, n개의 인증서를 검증하기 위하여 검증자는 n개의 인증서를 받아, n-1쌍의 인증 관계를 검증하여야 하는 것이다. 이 인증 절차는 순차적으로 이루어지며, (i, i-1)의 관계를 검증하기 위하여 i-1의 인증서를 받기 전에는 i-2가 어떤 것인지 알 수가 없다.

3. 일반적인 인증서 검증 방법



< 그림 1 > 일반적인 인증서 검증 경로

일반적으로 다음의 두 가지 경우가 있다.

① 같은 인증 영역에서 검증을 할 때

② 다른 인증 영역에서 검증을 할 때

<그림 1>에서 "갑", "을", "병", 그리고 "정"은 일반 사용자이며, CA#는 공인인증기관이다. 또한, rootCA#는 각 인증 영역에서 최상위 공인인증기관이다. "갑"과 "을"은 최상위 공인인증기관 만을 공유하는 같은 인증영역에 포함되어 있으며, "병"과 "정"은 공통의 공인인증기관이 존재하는 인증영역에 포함된다. 기본적으로 일반 사용자(갑, 을, 병, 정)는 자신의 인증서를 발행한 공인인증기관과 그 공인인증기관의 직속(하부의 공인인증기관의 인증서를 발행한) 상위 공인인증기관 만을 신뢰한다. "갑"이 "을"의 인증서를, "병"이 "정"의 인증서를 검증하는 것은 같은 인증 영역에서 검증을 할 경우(①)이며, "갑"이 "정"의 인증서를 검증하는 것은 다른 인증 영역에서 검증을 할 경우(②)이다.

가. "병"이 "정"의 인증서를 검증할 때

1) 먼저 병은 정(owner)의 인증서를 받는다.

2) 인증서에 있는 서명자(issuer)의 이름을 확인하여

3) 인증서를 발급한 공인인증기관의 인증서를 받아

4) 공인인증기관이 소유자의 인증서를 인증했는지(서명을 검사) 검사한다.
5) 병은 공인인증기관을 신뢰할 수 있는지를 판단한다.
6) 병이 신뢰할 수 있는 공인인증기관의 이름이 나올 때까지 2)~5)를 반복 수행한다.

이것은 을의 인증서를 갑이 검증할 때도 동일하다. 단지, 을의 인증서를 갑이 검증하는 경우 <그림 1>에서처럼 최상위 공인인증기관까지 2)~5)를 반복 수행하여야 한다는 것이 다르다.

나. "갑"이 "정"의 인증서를 검증할 때
갑은 정이 다른 계층적 구조에 있는 지 알 수가 없다. 따라서 가.의 경우에서처럼 1)~6)의 단계를 거치게 된다. 하지만, 마지막 경우, 인증 영역 2에 있는 Root CA4는 자신이 스스로 서명한 인증서를 가지고 있기 때문에 그 상위 공인인증기관이 없음을 확인할 수가 있고, 따라서, 갑은 자신이 신뢰하는 Root CAe와 신뢰하여야 하는 Root CA4가 상호 인증한 인증서를 필요로 하게 된다.

III. 제안방법

1. 가정

일반적으로 본 연구를 적용하기 위한 상황으로 다음을 가정한다.

- 공인인증기관은 초당 하나의 인증서를 발급할 수 있다.

이 가정은 인증서 일련번호를 생성할 때 초 단위까지만 생성하기에 초당 하나의 인증서를 발급한다고 가정한다. 만약, 초당 2개 이상의 인증서를 발급할 수 있다면 일련번호를 생성 시 그 부분을 고려한다.

- 한 공인인증기관은 255개 이하의 하부 공인인증기관을 둘 수 있다.

이 것은 공인인증기관의 식별자를 생성할 때에 8 비트를 할당함으로써 그 길이를 제한하였다.

- 최상위 공인인증기관(Root CA)은 각 나라에 유일하게 하나씩 존재한다.

이는 각 나라를 대표하는 공인인증기관으로써, 이 공인인증기관과 국가는 1:1 대응이 된다. 만약, 한 나라에 최상위 공인인증기관이 여러 개가 있을 경우, 이 최상위 공인인증기관을 인증하는 인증기관이 필요하며, 이는 그 나라를 대표한다.

- 각 공인인증기관은 계층적 구조로 이루어져 있으며 그 깊이는 최대 6이다.

이 가정은 전체 공인인증기관의 식별자 구조의 크기를 48 비트(8 * 6)로 두어 그 길이를 제한하기 때문에 공인인증기관의 계층적 구조의 깊이를 6으로 제한한다.

- 최소한 각 나라의 최상위 공인인증기관은 신뢰할 수 있다.

다른 나라에서 발급한 인증서를 인증하기 위하여 가장 선행되는 것이 가장 최상위 공인인증기관을 신뢰하는 것이다. 최상위 공인인증기관을 신뢰할 수 있어야만 하부 공인인증기관 및 사용자에 대한 검증이 시작될 수 있다.

2. 일련번호의 구조

년(Year)부터 초(Second)까지가 인증서의 실 소유자에 대한 부분이며, CAid부터 RootCA까지는 순차적으로 그 인증서를 인증한 공인인증기관의 식별자이다.

Flag	**Year**	**Month**	**Date**	**Hour**	**Minute**	**Second**	**CAid**	...	**RootCA**

- **Flag** : 4 비트, 인증서의 소유자가 공인인증기관인지 일반 사용자인지를 확인(1 비트) 0000인 경우 공인인증기관, 1000인 경우는 일반 사용자이다.
- **Year ~ Second** : 인증서가 발급된 시간을 의미한다. **Year**는 1000년을 기준으로 하여 총 10 비트가 할당되며, **Month**는 4 비트, **Date**와 **Hour**는 각각 5 비트, **Minute**와 **Second**는 각각 6 비트가 할당되어 전체적으로 36 비트가 할당된다.
- **CAid** : 각 공인인증기관들의 식별자이다.
- **RootCA** : 오직 하나의 나라에 하나씩 주어지는 이름으로, 그 나라를 대표한다.

인증서 일련번호의 전체 크기는 4+10+4+5+5+6+6+8*6 = 40+48 = 88 비트 즉, 11 바이트이다.

3. 제안하는 일련번호 부과 예

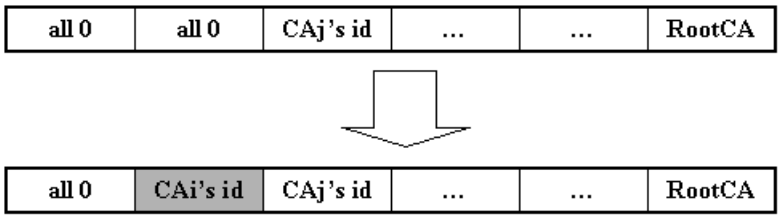
가. 인증서의 소유주가 공인인증기관인 경우

공인인증기관 i의 인증서를 공인인증기관 j가 인증하는 경우, j는 i의 인증서의 일련번호를 다음과 같이 부과한다.

1) Flag : 0000

2) Year ~ Second : 인증서에 서명하는(인증하는) 시각

3) CAid ~ RootCA : CAi의 id를 지정한 후, 자신의 인증서의 일련번호 중 CAid를 표현하는 부분에서 자신 다음에 삽입하여 생성. (<그림 2> 참조)



< 그림 2 > CAid의 부과

※ 최상위 공인인증기관인 경우, 최상위 공인인증기관은 각 나라를 대표함으로 미리 지정하여야 한다. 그리고, 인증서의 서명은 자신 스스로가 한다.

나. 인증서의 소유주가 일반 사용자일 경우

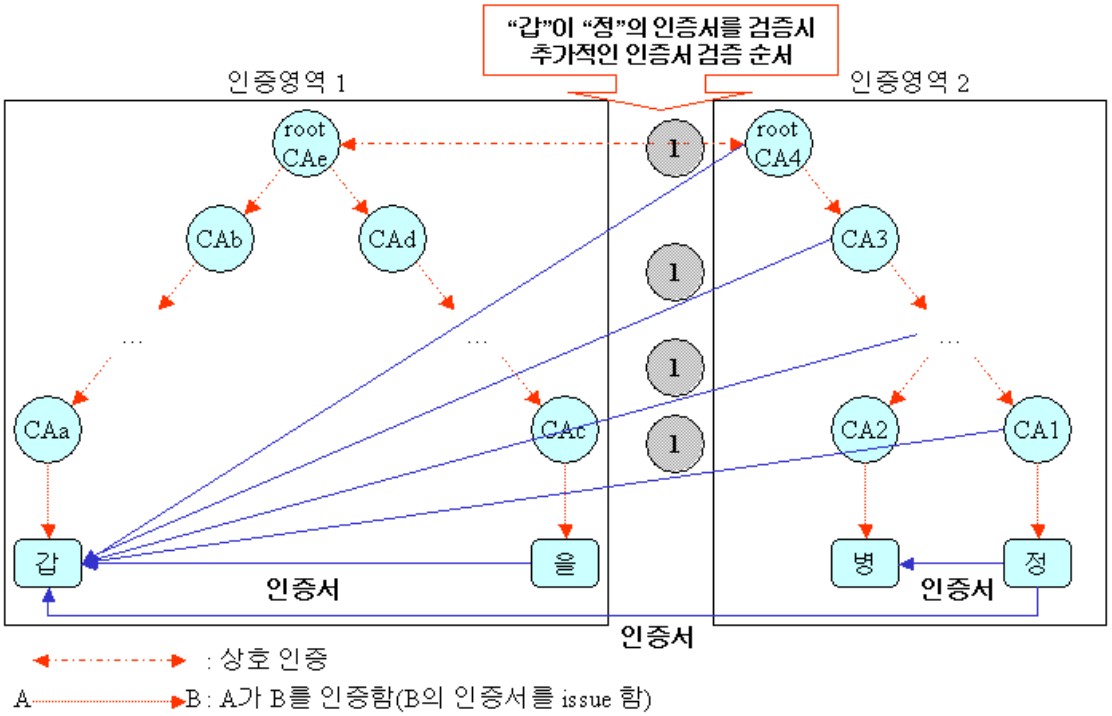
사용자의 인증서를 공인인증기관 i가 인증하는 경우, 공인인증기관 i는 사용자의 인증서의 일련번호를 다음과 같이 부과한다.

1) Flag : 1000

2) Year ~ Second : 인증서에 서명하는(인증하는) 시각

3) CAid ~ RootCA : 자신의 인증서의 일련번호 중 CAid를 표현하는 부분과 동일

4. 제안하는 일련번호를 부과하여 인증서를 검증하는 방법



< 그림 3 > 제안하는 인증서 일련번호를 부과한 인증서의 검증 경로

제안하는 인증서 일련번호를 부과하였을 경우, 인증서 검증 경로는 <그림 3>과 같이 수

행된다. "갑", "을", "병", 그리고 "정"은 제 2장의 3. 일반적인 인증서 검증 방법에서와 동일한 조건이다.

가. "병"이 "정"의 인증서를 받았을 경우

정의 인증서의 일련번호가 Year~SecondCA1,,,CAi,,,RootCA4의 형태로 되어 있다고 하자. 그리고 병은 중간에 CAi에서 그 분기를 다르게 한다. 즉 병의 인증서의 일련번호는 다음처럼 가정할 수 있다. Year~SecondCA2,,,CAi,,,RootCA4

1) 병은 자신의 인증서의 일련번호와 정의 인증서의 일련번호를 비교하여 마지막으로 서로 같은 공인인증기관을 찾는다.

compare(정의 인증서, 병의 인증서) = CAi

2) 병은 공인인증기관 i를 신뢰함으로 그 하부의 공인인증기관들의 인증서를 요청하여 받는다.

3) 병은 그 인증서들을 검증한다.

나. "갑"이 "을"의 인증서를 받았을 경우

을의 인증서의 일련번호가 Year~SecondCAc,,,RootCAe 형태로 되어 있다고 하자. 그리고 갑의 인증서의 일련번호는 다음처럼 가정할 수 있다. Year~SecondCAa,,,RootCAe.

1) 갑은 자신의 인증서의 일련번호와 을의 인증서의 일련번호를 비교하여 마지막으로 서로 같은 공인인증기관을 찾는다.

compare(갑의 인증서, 을의 인증서) = RootCAa

2) 갑은 RootCAa를 신뢰함으로 그 하부의 공인인증기관들의 인증서를 요청하여 받는다.

3) 갑은 그 인증서들을 검증한다.

다. "갑"이 "정"의 인증서를 받았을 경우

정의 인증서의 일련번호가 Year~SecondCA1,,,RootCA4 형태로 되어 있다고 하자. 그리고 갑의 인증서의 일련번호는 다음처럼 가정할 수 있다. Year~SecondCAa,,,RootCAe.

1) 갑은 자신의 인증서의 일련번호와 을의 인증서의 일련번호를 비교하여 마지막으로 서로 같은 공인인증기관을 찾는다.

compare(갑의 인증서, 을의 인증서) = None

2) 갑은 RootCA4를 신뢰함으로(III.1 가정) 그 하부의 공인인증기관들의 인증서를 요청하여 받는다.

3) 갑은 그 인증서들을 검증한다.

※ *compare*(인증서, 인증서)는 입력으로 두 개의 인증서를 받아, 두 인증서의 일련번호를 비교하여(일련번호의 우측부터 시작), 서로 다르지 않은 마지막 공인인증기관의 식별자를 출력한다.

5. 비교분석

검증자는 인증서를 검증하기 전에 자신의 인증서와 비교를 함으로써 자신이 검증하여야 하는 공인인증기관들을 파악할 수 있다. 따라서, 그는 동시에 인증서를 받아서 그 인증서를 검증할 수 있다. 즉, 인증서 검증 경로가 병렬적으로 이루어질 수 있다. 이는 기존의 방법에서는 할 수 없다.

예를 들면, <그림 1>에서 "갑"이 "정"의 인증서를 검증할 때, "갑"은 "정"의 인증서를 인증한 CA1의 인증서를 받고(①), 계속적으로 CA1의 인증서를 발행한 공인인증기관의 인증서를 받게 된다. 마지막으로 CA3의 인증서를 인증한 RootCA4의 인증서를 받게 된다(⑯). 여기서 인증서 검증 경로의 길이를 n, 하나의 인증서와 그 인증서를 인증한 CA의 서명을 검증하는 데 걸리는 시간을 t 라고 보았을 때, "갑"은 "정"의 인증서를 검증하는데 총 (n-1) x (t) 시간(O(n))이 걸린다. 그러나, <그림 3>에서처럼, "갑"이 "정"의 인증서를 검증할 때, "갑"은 "정"의 인증서의 일련번호를 통하여 한번에 인증서 검증 경로의 전체를 파악할 수 있으므로, CA1 ..., CA3, RootCA4 의 인증서를 동시에 받을 수 있다. 따라서, "갑"이 "정"의 인증서를 검증할 때에는 (1) x (t) 시간(O(1))이 필요하다.

즉, 인증경로의 길이가 n이라고 하였을 때,

일반적인 인증서 검증 경로 알고리즘을 다음과 같다.

```
인증서1 := 일반 사용자(1)의 인증서

for(i=1 ; i<n ; i++){
        인증서2 := 인증서1의 서명자(i+1)의 인증서를 다운받음
        만약 verify(인증서1, 인증서2)의 결과가 참이라면 계속 수행
        그렇지 않다면, 실패.

        만약 i를 신뢰할 수 있다면 성공
        그렇지 않다면, 계속 수행.
        인증서1 := 인증서2
}
실패.
```

따라서, 전체 시간은 O(n)이 된다.

그러나, 제안하는 일련번호를 사용하여 인증서 검증 경로 알고리즘을 구현을 하였을 경우는 다음과 같다.

인증서를 n+1의 배열로 지정
인증서1 := 일반 사용자(1)의 인증서
인증서0 := 자신의 인증서
compare(인증서0, 인증서1)
인증서1의 CAid 구조에서 신뢰할 수 없는 모든 공인인증기관의 인증서를 다운받음
(인증서[2]~인증서[n])

병렬적으로 *verify*(인증서i, 인증서i+1)를 수행하여
결과가 참이라면 성공
그렇지 않다면 실패

따라서, *verify*( )가 병렬적으로 이루어지므로, 그 시간은 O(1)이다.

또한, 부과적으로 공인인증기관의 식별자를 지정할 때 정책과 대응해서 지정한다면, 인증서 검증을 통하여서가 아니라, 인증서의 일련번호를 비교함으로써, 검증자가 검증할 수 있는 정책를 지원하는 인증서인지도 확인할 수 있다.

※ *verify*(인증서1, 인증서2)는 입력으로 두 개의 인증서를 받아, 인증서2에 기록되어 있는 공개키를 가지고 인증서1에 있는 서명부분을 검증하여, 인증서2의 소유자가 인증서1을 서명했는지를 검증하는 함수이다.

IV. 결론 및 향후 과제

본 논문에서는 인증서 검증 경로를 한번에 파악할 수 있는 일련번호를 부과하는 새로운 방법을 제시하였다. 제안되는 방법을 이용하여 일련번호를 부과하였을 경우, 전체 인증서 검증 경로를 파악할 수 있기 때문에 인증서의 검증 과정을 병렬적으로 수행할 수 있으며, 이에 따라 인증서 검증 시간을 효율적으로 줄였다. 또한, 공인인증기관의 식별자를 정책에 따라 다르게 지정한다면, 인증서를 검증할 필요없이 인증서의 일련번호를 확인한 후 정책을 확인할 수 있다.

향후에 이 일련번호를 이용하여 좀더 효율적으로 인증서의 적합성을 검증하는 방법에 대한 연구가 필요하다.

참고문헌

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

[2] M. St Johns, "The PKIX UserGroupName GeneralName Type", July 2001, IETF draft-ietf-pkix-usergroup-00

[3] ASPeCT, "Report on final trial and demonstration", AC095/PFN/W12/DS/P/19/1, April 1998