제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

최적화된 인증 경로와 안전한 공개키 인증서 구조

송성근*, 윤희용*
*성균관대학교 정보통신 공학부
e-mail : *songsk@mail.skku.ac.kr, *youn@ece.skku.ac.kr

Optimal Certification Path and Secure Public Key Certificate Architecture

Sung Keun Song*, Hee Yong Youn*
*School of Information and Communication Engineering, Sungkyunkwan University

요 약

오늘날 대부분의 인증 시스템들은 PKI 환경으로 변화하는 추세이며, 인증서의 역할은 날로 중요해지고 있다. 만일 이런 인증서가 위조 된다면 심각한 정보 사고가 발생할 것이다. 따라서 인증서는 위조되면 안될 것이다. 그러나 인증서 위조 가능성은 존재한다. 왜냐하면 디지털 서명 방식을 사용하고 있기 때문이다. 인증서 위조 방법은 두 가지가 있다. 첫 번째가 인증기관의 비밀키를 알아내는 방법이고, 두 번째는 디지털 서명에 사용되는 해쉬 알고리즘의 충돌(Collision) 문제를 이용하여 위조하는 방법이다. 어느 것으로든 인증서가 위조되면 어느 누구도 기술적으로 위조라는 사실을 증명할 수 없다. 위조 인증서는 디지털 검증 방식에 의해 모두 유효하게 판정되기 때문이다. 첫 번째 방법은 디지털 서명에 있어서 원천적인 문제이다. 따라서 본 논문은 두 번째 방법인 해쉬 알고리즘의 충돌 문제를 이용한 위조를 해결하는 방법에 대해 연구한다. 또한 인증 경로를 최적화하는 방법에 대해서도 연구한다.

1. 서론

오늘날 통신 서비스나 인터넷 서비스에서 사용자 인증은 중요한 문제가 되었다. 그리고 인증 시스템의 현 추세는 PKI(Public Key Infrastructure) 환경에서 인증이 이루어지는 방향으로 나아가고 있다. PKI 환경의 인증 시스템에서 핵심적인 역할을 하는 것은 사용자의 공개키 인증서이다. 만일 이러한 인증서가 위조된다면, 인가된 사용자가 아닌 악의적인 제 3자가 위조된 인증서를 이용하여 정보 시스템에 접속하여 정보의 절취 및 정보의 변경 등 많은 정보 사고가 발생 할 것이다. 인증서의 위조가 가능한 이유는 인증서가 디지털 서명 방식으로 이루어져 있기 때문이다. 디지털 서명을 위조하는 방법에는 두 가지가 있는데, 첫 번째가 디지털 서명 방식의 원리가 되는 공개키 암호의 비밀키를 알아내는 방법이다. 두 번째는 디지털 서명 방식에 사용되는 해쉬 알고리즘의 충돌 문제를 이용하여 위조하는 방법이다. 첫 번째 방법은 필연적이기 때문에 피할 수 없는 문제이다. 따라서 본 논문에서는 위조는 어렵지만 가능성이 존재하는 두 번째 문제에 초점을 맞추어 인증서 위조 문제를 해결하는 방법에 대해 연구하고 새로운 인증서 구조를 제안한다.

인증 과정에서 인증 경로(Certification Path)는 중요하다. 인증 경로는 PKI 구조에 영향을 받는데, 경로가 길어질수록 인증에 걸리는 시간이 길어진다. 본 논문에서는 이러한 인증 경로를 최적화하는 방식에 대해서 연구한다.

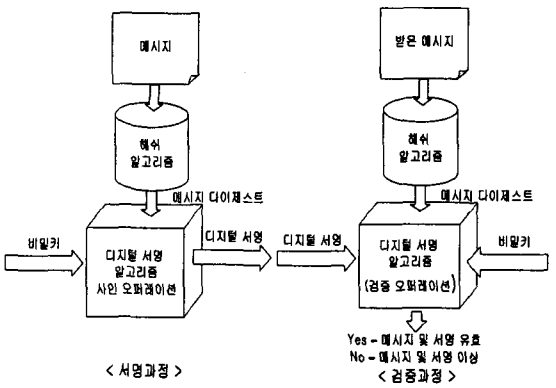
본 논문의 구성은 다음과 같다. 2장에서는 디지털 서명 방식에 대해서 알아보고, 인증서 위조가 가능한 이유에 대해서 알아본다. 3장에서는 PKI 구조와 이에 따른 인증 경로에 대해서 알아본다. 4장에서는 안전한 인증서 구조 및 인증 경로를 최적화하는 방식을 제안한다. 5장에서는 안전성에 대해서 논의하고, 마지막으로 6장에서 결론을 맺는다.

2. 인증서 위조 방법

2.1 디지털 서명 방식

디지털 서명 알고리즘은 디지털 서명을 생성하고 검증할 수 있는 능력을 제공한다. 디지털 서명은 서명자의 비밀키를 이용하여 생성하고, 검증은 비밀키와 부합되는 서명자의 공개키를 이용하여 실행한다. 각 사용자는 공개키와 비밀키의 한 쌍을 가지고 있으며 비밀키는 공개되지 않고 사용자만이 가지고 있으며, 반면 공개키는 PKI 환경에서 인증기관에 의해 인증되어 일반에 공개된다. 이로 인해 사용자는 비밀키가 노출되기까지 자신만의 서명을 만들 수 있으며, 그 서명을 받은 사람들 누구나 검증을 할 수 있다. (그림 1)은 디지털 서명 생성 과정과 검증 과정을 나타내고 있다.[1]

해쉬 알고리즘은 디지털 서명 생성 과정과 검증 과정에서 메시지 다이제스트(Message Digest)라 불리는 압축된 데이터를 얻기 위해 사용된다. 해쉬 알고리즘은 Secure Hash Standard, FIPS 180-1에 명시되어 있다.[2-3] (그림 1)의 서명 과정과 같이 메시지 다이제스트는 서명을 생성하기 위해 디지털 서명 알고리즘에 입력된다. 그리고 비밀키에 의해 암호화되어 서명으로 만들어진 후 메시지와 함께 검증자에게 보내진다. 검증자는 (그림 1)의 검증 과정과 같이 서명자의 공개키를 이용하여 메시지의 무결성과 디지털 서명의 유효성을 검증한다.



(그림 1) 디지털 서명 방식.

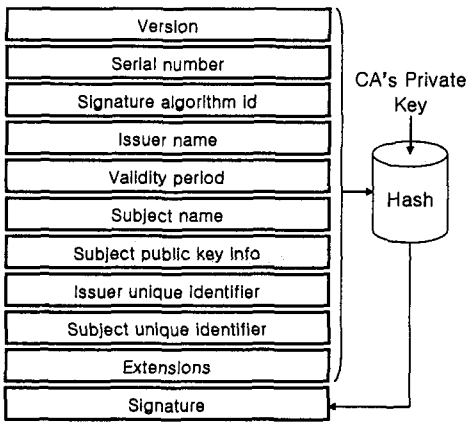
2.2 인증서 위조 방법

위에서 설명한 것처럼 디지털 서명을 위조하는 방법에는 두 가지가 있다. 첫 번째가 디지털 서명 방식의 원리가 되는 공개키 암호의 비밀키를 알아내는 방법이다. 두 번째는 디지털 서명 방식에서 사용되는 해쉬 알고리즘의 충돌 문제를 이용하여 위조하는 방법이다. 인증서도 디지털 서명을 사용하기 때문에 마찬가지이다. (그림 2)는 X.509 v3 인증서[4]를 나타내고 있다. X.509 인증서는 버전 3까지 있으며, 발행인, 발행 대상자, 대상자의 공개키, 발행인의 서명 등으로 구성된다.

해쉬 알고리즘의 충돌 문제란 서로 다른 메시지가 같은 해쉬값을 갖는 것을 말한다.

$$h(M)=h(M')$$

이 문제는 해쉬 알고리즘에 있어서 필연적 문제인데, 더욱더 중요한 문제는 같은 해쉬값을 갖는 메시지들이 한두 개가 아니란 것이다. 대표적인 해쉬 알고리



(그림 2) X.509 v3 인증서.

즘으로 MD5, SHA 등이 있는데, MD5는 512비트 단위 메시지를 128비트 해쉬값으로 출력하고, SHA는 512비트 단위 메시지를 160비트 해쉬값으로 출력한다. MD5는 동일한 해쉬값 갖는 메시지의 수는

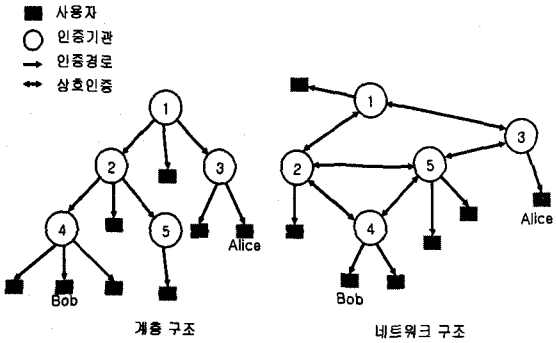
$$\frac{2^{516}}{2^{128}} = 2^{388}$$

즉, 해쉬값당 2^{388} 개의 메시지들이 같은 값을 갖는다. SHA의 경우는 해쉬값당 2^{356} 개의 메시지들이 같은 값을 갖는다. 이 문제를 이용하여 디지털 서명을 위조하는 방법은 유효한 메시지의 해쉬값과 위조 메시지의 해쉬값이 같도록 하여 유효한 메시지의 디지털 서명을 위조 메시지에 첨부하는 방법이다. 여기서 디지털 서명은 (그림 1)의 검증과정에서 위조된 메시지와 해쉬값이 같게 나오기 때문에 유효하다. 따라서 위조된 메시지도 유효한 것으로 인식될 것이다.

해쉬 알고리즘의 충돌 문제를 이용하는 위조 방법을 3가지로 분류할 수 있다. 그 첫 번째가 공격자가 같은 해쉬값을 얻기 위해 해쉬 알고리즘의 설계 구조의 문제점을 찾는 방법이다. 두 번째는 공격자가 공격 대상자의 유효한 디지털 서명을 공개키 인증서의 유효기간 동안 데이터 베이스화하여 메시지를 위조할 때 데이터 베이스에서 위조된 메시지와 같은 해쉬값을 갖는 서명을 찾아 위조 메시지에 첨부하므로서 위조하는 방법이다. 세 번째는 공격자가 공격 대상 메시지의 해쉬값과 같을 때까지 위조 문서를 수정하는 것을 반복하는 방법이다. 인증서를 위조하는 방법은 3가지 모두 가능한데, 위조의 가능성이 가장 높은 방법은 두 번째 방식이 되겠다. 그 이유는 유무선 PKI의 인증기관들은 비밀키의 긴 유효기간 동안 수많은 사용자들에게 인증서를 발행하기 때문이다.

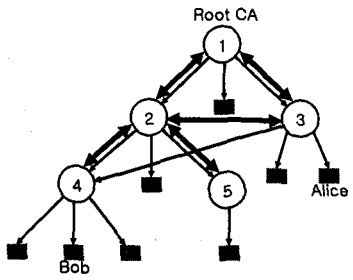
3. PKI 구조

PKI 구조는 크게 계층 구조와 네트워크 구조, 계층 구조와 네트워크 구조를 결합한 혼합형(Hybrid) 구조가 있다. (그림 3)은 계층 구조와 네트워크 구조를 나타내고 있다. (그림 4)는 혼합형 구조를 나타내고 있다.



(그림 3) 계층 구조와 네트워크 구조.

인증서는 인증 경로를 형성하기 위해 가상적인 체인으로 연결되어 있다. PKI의 계층 구조에서 루트 CA의 공개키는 모든 사용자에게 알려져 있다. 이 구조에서 어떤 사용자의 인증서라도 루트 CA의 인증 경로까지 검증하면 인증서가 검증된다. Alice가 CA 4가 발행한 Bob의 인증서를 검증하기 원한다면, CA 2가 발행한 CA 4의 인증서, CA 1이 발행한 CA 2의 인증서를 검증하므로서 검증한다. 이 구조는 깊이가 길어질수록 인증경로가 길어지는 단점을 가지고 있다.[5-6]



(그림 4) 혼합형 구조.

네트워크 구조에서 두 CA는 서로 인증서를 발행하면서 상호 인증을 할 수 있다. 사용자들은 자신의 인증서를 발행한 CA의 공개키만을 알고 있다. 인증서 검증은 상호 인증으로 형성된 CA들의 신뢰관계를 통해 검증된다. 이 구조에서 Alice는 CA 5가 발행한 CA 4의 인증서, CA 3이 발행한 CA 5의 인증서를 검증하므로서 Bob의 인증서를 검증한다. Alice는 CA 3를 신뢰하고 있으며, CA 3의 공개키를 알고 있기 때문이다. 이 구조도 사용자간의 거리가 떨어져 있을 경우 인증 경로가 길어지는 단점을 가지고 있다.[5-6]

혼합형 구조는 계층 구조의 장점과 네트워크 구조의 장점이 결합된 구조이다. 이 구조에서 Alice가 Bob의 인증서를 검증하기를 원한다면, Alice는 자신의 부모 CA인 CA 3에 의존되는 인증 경로를 찾거나, 루트까지의 Bob의 인증 경로를 찾으면 된다.[5] 이 구조는 계층 구조와 네트워크 구조의 단점을 많이 보완 했지만 여전히 인증 경로가 일정하지 않다는 단점을 가지고 있다.

4. 안전한 인증서 구조

4.1 새로운 디지털 서명 방식 및 인증서 구조

P_A : 알고리즘 A의 공개키; 모든 객체에게 공개됨

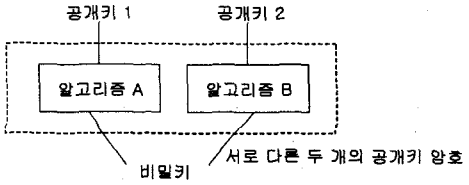
P_B : 알고리즘 B의 준공개키(Semipublic Key); 인증기관들에게만 공개됨

P_{AB}^{-1} : 두 개의 공개키 암호의 비밀키

$\{\}_A$Key : 알고리즘 A에 의한 암호화 또는 복호화

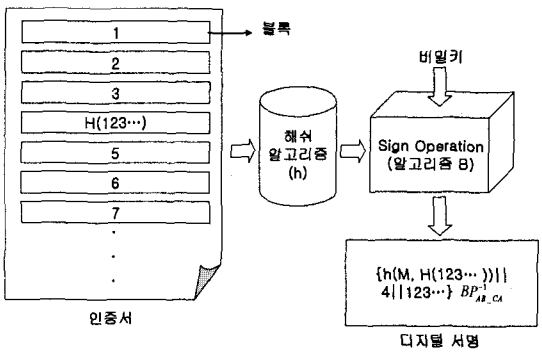
H: 임의 길이의 입력을 일정 크기의 값으로 늘리는 알고리즘

h: 임의 길이의 입력을 일정 크기의 값으로 압축시키는 알고리즘



(그림 5) 두 개의 공개키 암호.

인증서에 사용될 새로운 디지털 서명 방식은 다른 두 개의 공개키 암호가 결합된 두 개의 공개키 암호를 사용한다. 이 암호는 두 개의 공개키, 하나의 비밀키, 그리고 두 개의 알고리즘으로 구성된다. 이는 컴퓨터상에서 키들이 비트로 표현되기 때문에 가능하다. 대표적인 예로 RSA[7]와 ElGamal[8]의 결합을 들 수 있다.

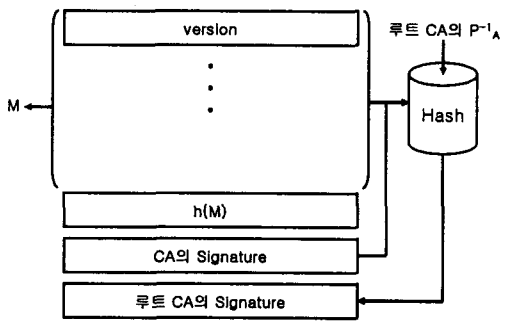


(그림 6) 인증서의 서명 과정.

인증서의 서명 과정은 다음과 같다. 먼저 랜덤넘버(RN: Random Number)의 해쉬값 H(RN)을 계산한다. H(RN)의 값의 크기는 적당한 블록 크기로 정한다.

그 다음 인증서의 메시지 부분에 H값을 추가하여 해쉬값 h(M, H(RN))을 계산한다. 여기서 h를 계산할 때, H의 값은 CA가 선택한 메시지의 특정 블록에 위치한다. 즉, 메시지를 블록 단위로 나누어 위치시킨다. CA는 h값, H값을 메시지에 위치시킨 블록 위치(BP: Block Position), RN를 비밀키를 이용하여 알고리즘 B로 암호화해서 디지털 서명을 만든다. (그림 6)는 서명 과정을 나타내고 있다.

PKI의 어느 구조든지, 계층 구조의 루트 CA처럼 중심적인 CA를 두어 하위 CA가 인증서를 발행할 때 인증서에 인증 서명을 하도록 한다. 본 논문에서는 이를 루트 CA라 정한다. 루트 CA의 인증 서명 과정은 다음과 같다. 인증서의 메시지 부분의 해쉬값 h(M)과 하위 CA의 서명을 포함한 인증서의 해쉬값 h(Certificate)을 계산한다. 그리고 h(Certificate)를 비밀키로 암호화하여 인증 서명을 만들어 h(M)과 서명을 인증서에 첨부한다. (그림 7)은 새로운 인증서 구조를 나타내고 있다.



(그림 7) 새로운 인증서 구조.

인증서 서명의 검증에서 CA들은 해당 CA의 준공개키로 인증서의 서명을 검증하고, 사용자들은 루트 CA의 공개키로 서명을 검증한다. 만일 사용자들이 인증서에 의심이 가면 자신의 CA나 다른 CA들에게 보내어 검증을 받을 수 있다.

4.2 인증 경로

새로운 인증서의 인증 경로는 어느 구조나 거리에 상관없이 동일하다. 왜냐하면 어느 CA가 발행하건 인증서에 루트 CA의 인증 서명이 들어가 있고, 루트 CA의 공개키는 PKI 구조의 모든 사용자에게 알려져 있기 때문이다. 따라서 3장에서 설명한 것처럼 여러 경로를 거치지 않고 하나의 인증서를 통해 검증이 가능하다.

5. 안전성

새로운 인증서는 2장에서 설명한 해쉬 알고리즘의 충돌 문제를 이용하여 어느 누구도 위조할 수 없다. 왜냐하면 준공개키를 가지고 있는 인증기관 외에는 해쉬값을 모르기 때문이다. 해쉬 알고리즘의 충돌 문제를 이용하여 위조할 수 있는 부분은 루트 CA의 인증 서명부분인데, 안전 장치로서 인증서에 h(M)을 첨부한다. 왜냐하면, 두 개의 해쉬값, h(M)와 h(Certificate)가 같은 위조 인증서를 만들기 어렵기 때문이다. 그러나 가능성이 존재하기 때문에 사용자들은 의심이 생기면 CA들에게 검증을 의뢰 해야 할 것이다. 인증 시스템을 사용하는 서비스 업자들은 인증기관과 준공개키를 공유하므로서 CA에게 검증의뢰 없이 하위 CA의 서명을 검증할 수 있다.

두 개의 공개키 암호의 안전성은 결합된 서로 다른 공개키 암호의 안전성에 연관된다. 서로 다른 수학적 문제들을 바탕으로 하고 있기 때문에 공개키 암호들의 안전성이 비슷하다면 두 개의 공개키 암호는 하나의 공개키 암호의 안전성과 같다. 다시 말해서 두 개의 공개키 암호를 구성하는 공개키 암호들은 안전성이 비슷해야만 한다.

6. 결론

본 논문에서 제안한 새로운 인증서 구조는 어느 누구도 위조 할 수 없다. 인증기관이나 인증 시스템을 사용하는 서비스 업자 외에는 준공개키를 알지 못하며, 따라서 해쉬값을 모르기 때문에 위조를 할 수 없다. 또한 새로운 인증서 구조는 사용자들의 거리와 상관없이 동일한 최적화된 인증 경로를 갖는다. 따라서 인증 경로에 관한 기존의 구조의 문제점들을 해결할 것으로 예상된다. 그리고 새로운 인증서 구조의 두 개의 공개키 암호는 기존 방식과 결합이 가능하다. 따라서 유용하게 변형 사용이 가능할 것으로 예상된다.

참고문헌

[1] National Institute of Standards and Technology (NIST), Digital Signature Standard, FIPS PUB 186-2, http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf

[2] National Institute of Standards and Technology (NIST), Secure Hash Standard, FIPS PUB 180-1, http://www.itl.nist.gov/fipspubs/fip180-1.htm

[3] Bruce Schneier, Applied Cryptography – Protocols, Algorithms and Source Code in C, Second Edition, John Wiley and Sons, New York, 1996.

[4] R. Housely, W. Ford, W. Polk, and D. Solo, "Internet X.509 Public Key Infrastructure," IETF RFC 2459. Jan. 1999

[5] W.T. Polk, N.E. Hastings, A. Malpani, "Public Key infrastructures that satisfy security goals", IEEE Internet Computing, July-Aug 2003

[6] National Institute of Standards and Technology (NIST), "A Proposed Federal PKI Using X.509 V3 Certificates", http://citeseer.nj.nec.com/152868.html

[7] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Communications of the ACM, 21(2):120-126, February 1978.

[8] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. Info. Theory, IT 31, 1985.