韓國 컴퓨터情報學會 論文誌
第13卷 第1號, 2008. 1.

2008-13-1-3-4

해쉬체인을 이용한 인증서 상태 검증 방법들의 문제점과 해결 방법

강 현 중*, 안 정 희**

Problems of certificate status validation methods using hash chain and their countermeasure

Kang Hyun Joong*, Ahn Jeong Hee**

요 약

해쉬체인을 이용한 사용자의 인증과 무결성 제공방법이 활성화되면서 이를 이용한 인증서 상태검증 방법들이 다양하게 제안되었다. 제안된 방법들중에 NOVOMODO에서는 CA가 해쉬값을 생성하여 각 사용자에게 배포하고 Jianying Zhou의 프레임워크와 양종필의 개선된 프레임워크에서는 사용자가 해쉬값을 생성하여 검증자에게 전달한다. 그러므로 해쉬값의 생성 및 배포를 위한 작업량이 각 사용자에게 분산되어진다. 그러나 이들 프레임워크는 CA가 발행한 인증서에 오류가 없고 CA의 개인키가 노출되지 않는다는 가정을 전제로 하고 있다. CA도 공격자에 의해 언제든지 개인키가 노출될 수 있으므로 이들 프레임워크는 실제 PKI 환경에 적합하지 않다. 따라서 본 논문에서는 사용자의 인증서에 CA가 제어할 수 있는 해쉬값을 추가하여 이를 해결하고자 한다. 추가되는 해쉬값은 CA내의 사용자들에게 동일하며 CA에서 해쉬값을 생성, 저장 및 배포하기 위한 비용이 적다. 그리고 우리는 또한 양종필의 프레임 워크에서 서명 및 검증 절차의 문제점을 지적하고 개선 방법을 제안한다. 우리의 제안은 다른 프레임워크들보다 실제 PKI 환경에 적합하다.

Abstract

As the authentication and the integrity methods based on the hash chain are popular, several certificate status validation methods based on the same function are proposed at the moment. In NOVOMODO, a CA generates and releases the hash value to each user. In Jianying Zhou's framework and Jong-Phil Yang's framework, a user generates and releases the hash value to verifier. Therefore, the CA loads are distributed to each user. However, these frameworks are based on the assumption that the CA's secret key is not lost or compromised and the certificates issued by the CA are error-free. Therefore, these frameworks are not suitable in real PKI environments. In this paper, as one hash value generated by CA is included in the user's certificate in addition, the certificate revocation published by CA using that value can be managed. The hash value included in user's certificate is the same for all users. The computation costs, the storage amounts and the release costs are small in the CA. And we modify the procedure for the signature and its validation in Jong-Phil Yang's framework. Our solution is more suitable than those frameworks in real PKI environments.

▸ Keyword : PKI, 해쉬체인(Hash Chain), 인증서 상태 검증(Certificate Status Validation)

• 제1저자 : 강현중
• 접수일 : 2007. 12. 11, 심사일 : 2007. 12. 15, 심사완료일 : 2008. 1. 25.
* 서일대학 인터넷정보과 교수 **두원공과대학 컴퓨터정보과 교수
※ 본 논문은 2007년 서일대학 학술 연구비에 의해 연구되었음

Ⅰ. 서 론

PKI(Public Key Infrastructure)는 공개키 기술을 이용하여 정보보호 서비스(무결성, 인증, 부인 방지)를 제공해주는 강력하고도 광범위한 기술이다. PKI의 기본 개념은 엔티티(사용자 혹은 기관)의 개인 정보와 공개키를 묶어 CA(Certificate Authority)의 개인키로 서명한 엔티티의 인증서를 사용한다는 것이다. 만약 엔티티의 개인키가 노출되거나 개인정보가 변경되면 엔티티는 CA에게 자신의 인증서 폐지를 요청하게 된다. 한 인증서가 폐지되었는지의 여부를 나타내는 정보를 CSI(Certificate Status Information)라고 하고 CRL(Certificate Revocation Lists)은 잘 알려진 CSI 방법 중에 하나이다 [2,9,10].

폐지된 인증서 목록을 담고 있는 CRL은 간편하다는 장점을 갖고 있지만 CRL을 저장하고 있는 CA의 저장소와 사용자간의 통신비용이 높다는 단점이 있다. 인증서 폐지 목록을 위한 통신비용, 계산 비용 그리고 저장공간을 줄이기 위해서 다양한 방법들이 제안되고 있다. 이중에는 Delta-CRL, CRL DP(Distributed Point), Over-issued CRL, Indirect CRL, Dynamic CRL DP, Freshest CRL, CRT(Certificate Revocation Tree), Authenticated Directory, OCSP, SCVP 등이 있다 [1,2,5,6,7,8,9,15].

일방향 해쉬함수를 이용한 메시지 인증과 무결성 제공방법이 활성화되면서 이를 이용한 인증서 상태 검증방법들도 제안되었다. 2002년 Micali에 의해 제안된 NOVOMODO에서는 CA가 각 사용자들을 위한 해쉬체인을 생성하고 각 주기마다 해당하는 해쉬값을 사용자에게 배포하였다 [11]. 2003년 Jianying Zhou 등에 의해 제안된 "A Efficient Public-key Framework"에서는 CA 대신 각 사용자가 해쉬체인을 생성하여 서명과 함께 검증자에게 전달하도록 하였다. 검증자는 CA로부터 인증서 폐지 정보를 획득하지 않고서도 해쉬연산만으로 서명자의 인증서 상태를 파악할 수 있다 [3]. 2003년 양종필 등은 Jianying Zhou의 프레임워크를 실제 PKI 환경에 적용하기 쉽도록 개선시켰다. 양종필의 개선된 프레임워크에서는 인증서 내에 추가되는 파라미터가 재정의되었으며 사용자의 서명과 검증자의 검증방법도 개선되었다 [4]. 또한 이영교 등은 해쉬체인을 다양한 방법으로 이용하였다 [12-14].

그러나 해쉬체인을 이용한 인증서상태 검증방법들은 몇몇 문제점들을 가지고 있다. 특히 Jianying Zhou의 프레임워크와 양종필의 개선된 프레임워크는 동일한, 근본적인 문제점을 가지고 있다. 본 논문에서 우리는 이 방법들을 분석하고 문제점을 보인다. 논문의 나머지 부분은 다음과 같이 구성되어진다. 2장에서 해쉬체인을 이용한 인증서상태 검증방법들을 소개하고 3장에서는 이들을 비교, 분석하여 문제점을 제시한다. 4장에서는 문제점들을 해결하기 위한 개선 방법을 제안하고 5장에서는 제안한 방법의 특징을 설명한다. 그리고 마지막으로 6장에서 결론을 맺는다.

Ⅱ. 해쉬체인을 이용한 인증서 상태 검증 방법들

이 장에서는 Micali의 NOVOMODO, Jianying Zhou의 프레임워크 그리고 양종필의 개선된 프레임워크를 각각 소개한다.

2.1 Micali의 NOVOMODO

NOVOMODO에서 사용자의 인증서에는 2개의 20 바이트 해쉬값이 추가되는데 하나(X365)는 유효표시(validity target)로 사용되고 다른 하나(Y1)는 폐지표시(revocation target)로 사용된다. 다음의 식 (1)은 NOVOMODO에서 사용되는 인증서 형식을 나타낸 것이다.

$$Cert_{user} = Sig_{SK_{CA}}(PK_{user}, SN, I, S, V, \cdots, X_{365}, Y_1) \cdots (1)$$

식 (1)에서 PKuser는 사용자의 공개키이고 SN은 인증서의 순서번호, I와 S는 각각 인증서의 발행자와 소유자이고 V는 유효기간이다. 그리고 X365와 Y1은 추가된 해쉬값이다. CA는 일방향 해쉬함수 h를 이용하여 해쉬값을 생성한다. 사용자의 인증서 유효기간이 1년이고 CA가 인증서의 상태를 1일 주기로 확인해준다고 가정했을 때 CA는 서로 다른 입력값 X0으로부터 365번 해쉬연산으로 유효표시 X365 ($X_0 \xrightarrow{h} X_1 \xrightarrow{h} X_2 \cdots X_i \cdots \xrightarrow{h} X_{365}$)를 얻으며 서로 다른 입력값 Y0로부터 한번 해쉬연산으로 폐지표시 Y1($Y_0 \xrightarrow{h} Y_1$)을 얻는다. CA는 초기값인 X0, Y0 그리고 모든 중간값인 Xi를 안전하게 보관한다. 그런 다음 각 주기의 시작 시점에 CA는 대응하는 중간 해쉬값을 사용자에게 인증서에 대한 유효 혹은 폐지를 확인해주는 유효증명(validity proof) 혹은 폐지증명(revocation proof)으로 배포한다. PKI를 기반으로 하는 E-business 혹은 E-commerce에서, 사용자로부터 인증서와 해당하는 해쉬값을 수신한 클라이언트는 인증서에 포함된 해쉬값과 함께 수신된 해쉬값을 해쉬연산을 통하여 비교한다.

비교 결과가 동일하면, 클라이언트는 해쉬값에 대한 문서 인증과 무결성을 확인할 수 있고 또한 해쉬값에서 사용자의 인증서 상태를 확인할 수 있다. 이 해쉬값은 작지만 역연산을 할 수 없으므로 누구(공격자)도 흉내낼 수 없다.

2.2 Zhou의 공개키 프레임워크

Jianying Zhou의 공개키 프레임워크에서 사용자의 인증서 CERTU에는 한 개의 해쉬값(Hj(r))과 3개의 파라미터(D, j, L)가 다음의 식 (2)와 같이 추가된다.

$$CERT_U = SIGN_{CA}(U, PK_U, D, H^j(r), j, L) \quad \cdots\cdots (2)$$

식 (2)에서 SIGNCA는 CA의 개인키에 의한 서명이고 U는 사용자의 정보이고 PKU는 사용자의 공개키, D는 인증서의 유효 시작일, Hj(r)는 해쉬체인의 마지막 값, r은 사용자만이 알고 있는 랜덤 수, j는 리프레싱 주기 수 그리고 L은 인증서의 유효성을 리프레싱하기 위한 시간 주기이다.

CERTU의 최대 유효기간은 인증서의 유효성을 리프레싱하기 위한 시간주기로 나누어진다. 리프레싱 시점들은 D1=D+L, D2=D+2*L, … , Dj=D+j*L로 정의된다. 사용자는 일방향 해쉬체인을 생성하는데 Hi(r)=H(Hi-1(r))(i=1,2, … ,j) 여기서 H0(r)=r이고 r은 사용자에게만 알려진 랜덤 수이다. 사용자가 인증서 발행을 원할 때에 (PKU,D,Hj(r),j,L)을 CA에게 보낸다. 그리고 CA는 CERTU를 사용자에게 발행한다.

사용자는 각 리프레싱 시점마다 대응하는 해쉬값을 배포한다. 달리 말하면 사용자가 검증자와 상호과정 중에 서명을 생성하게 되면 서명에다가 CERTU의 유효를 증명하는 (Hi(r),i)를 추가하여 보낸다. 그리고 검증자는 먼저 서명자의 서명을 확인하기 위하여 Hj-i(Hi(r))=Hj(r) (여기서 0≤i<j)을 확인한다. 이것이 성립하면 검증자는 CERTU가 De=D+(j-i)*L (여기서 De는 다음의 리프레싱 시점)때까지 유효함을 확인하게 된다.

사용자가 일정한 기간 동안 서명을 생성하지 않는다면 해쉬값 역시 배포할 필요가 없다. 그러므로 해쉬값의 배포는 전적으로 사용자에게 달려 있다(기본 프레임워크). 만약 사용자의 컴퓨터에서 r과 SKU(사용자의 개인키)가 노출된다면 공격자는 인증서의 최대 유효기간 동안 불법적으로 유효한 서명을 생성할 수 있다. 이 문제는 해쉬체인의 루트값인 r이 패스워드기반 프로토콜을 갖는 SS(Security Server)내에서 생성되어진다면 해결될 수 있다. 그리고 SS는 검증자에게 해쉬값을 배포할 수 있다(관리 제어된 프레임워크). CA대신 사용자(혹은 SS)가 해쉬값을 배포하므로 이들 방법은 CA의 부하를 줄여줄 수 있다. 물론 이들 프레임워크들이 CA(그리고 루트 CA)의 개인키가 노출되지 않았고 CA에 의해 발행된 인증서에 에러가 발생하지 않는다는 가정에 기초하고 있다 [3].

2.3 양종필의 개선된 프레임워크

양종필은 Jianying Zhou의 프레임워크를 실제적으로 적용할 수 있도록 개선하였는데 여기서 사용자의 인증서 CERTU는 또한 한 개의 해쉬값 (Hj(r))과 2개의 파라미터 (j, CW)가 다음의 식 (3)과 같이 추가된다.

$$CERT_U = SIGN_{CA}(U, PK_U, H^j(r), j, CW) \quad \cdots\cdots\cdots (3)$$

식 (3)에서 SIGNCA는 CA의 개인키에 의한 서명이고 U는 사용자의 정보이고 PKU는 사용자의 공개키, Hj(r)는 해쉬체인의 마지막 값, r은 사용자-정의 패스워드, j는 리프레싱 주기 수 그리고 CW는 CA의 보안 정책에 의해 선택된 콘트롤 윈도우이다. 사용자 U는 SKU를 암호화하며 해쉬체인을 생성하는 데에 사용될 사용자-정의 패스워드 r을 생성한다. U는 자신의 인증서 발행 요청을 위해 CA에게 (PKU,Hj(r),j)를 보낸다. 그리고 CA는 U에게 CERTU를 발행한다.

U가 자신을 검증자 V에게 인증시키고자 할 때에는 자신의 인증서 CERTU와 함께 현재 해쉬값 그리고 정보 (Hj(r), i)를 검증자에게 전달한다. 그러면 V는 Hj-i(Hi(r))=Hj(r)을 확인한다. 이것이 진실이면 V는 U의 인증서가 현재 유효함을 확인하게 된다. V는 현재 지역시각을 시작 포인트 i로 그리고 종료시점을 CW로 설정한다. 그리고 V는 U의 인증서를 종료시점까지 캐쉬하면서 신뢰하게 된다.

U가 V에게 서명을 보낼 때, V는 서명의 수신 시각이 종료시점을 넘기지 않았는지를 체크한다. 그렇다면 V는 U의 서명과 PKU가 유효하다고 판단할 수 있다. 그렇지 않다면 CA에게 인증서 폐지 정보를 요청하게 된다.

양종필 등은 Zhou의 프레임 워크에서 몇몇 보안 파라미터들을 실제 PKI 환경에서 응용하기에 보다 적합하도록 변경하였다. 그리고 사용자의 인증 및 서명 절차를 변경하였다. 사용자-정의 패스워드 r을 도입하였기 때문에 Zhou의 프레임워크에서 SS는 더 이상 필요가 없게 되었다. CA의 보안 정책에 따라 선택되어지는 CW를 도입하였으므로 이제 U의 인증서 종료시점은 CA에 의해 제어되어 진다. 높은 보안 레벨의 인증서인 경우에는 CW는 짧고 낮은 보안 레벨인 경우에는 CW는 길게 된다. V는 U의 인증서를 단 한번 캐싱함으로써 종료시점까지 U를 신뢰할 수 있게 된다. 이 방법은 U와 V의 상호 작용이 빈번한 경우에 특히 효율적이다. 결국 이들이 제

안한 방법은 Zhou의 프레임워크를 실제 PKI 환경에 적용하는 데에 효율적이다 [4].

III. 해쉬체인을 이용한 인증서 상태 검증 방법들의 분석 및 문제점

이 장에서는 해쉬체인을 이용한 인증서상태 검증방법들을 분석하여 문제점들을 보이고자 한다.

3.1 인증서 폐지 원인의 부족

인증서는 사용자의 인적 정보와 그의 공개키를 CA의 개인키로 묶은 전자서명이다. 일반적으로 인증서는 유효기간 전에도 폐지될 수 있으며 그 이유는 다음과 같다 [5, 10].

제휴된 개인키의 손실이나 분실
사용자의 인적 정보 변경
소유자의 접근 권한 변경
발행자와의 관계 변경
암호학적 경계
CA(혹은 root CA)의 개인키 손실 및 분실

위에서 (1), (2), (3)은 사용자에서 발생하고 (4), (5), (6)은 CA에서 발생하는 원인이다. 그런데 Jianying Zhou의 프레임워크와 양종필의 개선된 프레임워크에서는 CA가 발행한 인증서가 오류가 없고 CA의 개인키가 노출되지 않는다는 가정을 전제로 하고 있어 (4), (5), (6)의 원인을 반영하지 못하고 있다. 따라서 실제 PKI 환경에 적용하는 데에 있어 완벽하지 못하다.

3.2 캐쉬로 인한 메모리 사용량 증가

양종필의 개선된 프레임워크에서는 서명자가 주기마다 자신의 인증서 유효여부를 확인해주는 해쉬값을 검증자에게 전달한다. 검증자는 이를 확인하고 유효하면 로컬 시각을 시작시간으로, CW를 최종시각으로 설정하여 해쉬값 갱신주기까지 서명자의 인증서를 캐쉬하여 신뢰하게 된다. 따라서 서명자는 해당 주기동안 별도의 해쉬값 및 인증서 배포없이 문서와 그에 대한 서명값만을 검증자에게 전달한다. 그러나 이 방법은 동일한 서명자와 검증자간에 상호작용이 빈번하게 이루어질 때에 효율적이다. 반대로 검증자가 계속 새로운 서명자의 서명을 검증하게 된다면 인증서의 캐쉬로 인한 메모리 량의 사용이 증가하며 인증서를 관리하기 위한 캐쉬 알고리즘이 필요하게 된다. 그리고 캐쉬된 인증서를 거의 사용하지 않게 되는 비효율적인 문제점이 발생한다.

3.3 해쉬값의 분실 가능성

양종필의 개선된 프레임워크에서 서명자가 주기마다 자신의 인증서 유효여부를 확인시키는 해쉬값과 인증서를 검증자에게 전달한다. 따라서 서명자는 해당 주기동안 별도의 해쉬값 배포없이 문서와 그에 대한 서명값만을 검증자에게 전달한다. 그러나 이 방법은 주기마다 검증자에게 전달되는 해쉬값이 분실되는 경우에 추후에 전달되는 문서 및 서명값을 검증자가 검증할 수 없게 될 수도 있다. NOVOMODO나 Jianying Zhou의 프레임워크에서는 서명과 함께 해쉬값, 인증서가 전달되므로 이를 분실하더라도 해당 서명만이 검증되지 못하지만 양종필의 프레임워크에서는 해당 주기의 모든 서명을 검증하지 못할 수도 있다. 따라서 해쉬값의 배포 시에 수신 여부를 확인하는 절차가 필요하게 된다.

IV. 개선 방법

본 논문에서는 양종필의 프레임워크를 실제 PKI 환경에 더욱 적합하도록 개선하고자 한다. 우리의 제안에서 사용자의 인증서는 다음의 식 (4)와 같이 2개의 해쉬값과 3개의 파라미터를 추가로 갖는다.

$$CERT_U = SIGN_{CA}(U, PK_U, D, H^j(r), Z_1, j, L) \cdots\cdots (4)$$

식 (4)에서 SIGNCA는 CA의 서명, U는 사용자의 정보, PKU는 사용자의 공개키, D는 유효 시작시각, Hj(r)은 해쉬체인의 마지막 값, r은 사용자 정의 패스워드, j는 시간주기의 수, L은 CA의 정책에 의해 선택되는 유효기간이다.

〈초기절차 : 인증서 발행〉

사용자는 키 쌍을 생성한다. PKU, SKU

사용자는 패스워드 r을 생성하고 이를 이용하여 SKU를 암호화한다.

사용자는 일방향 해쉬체인 Hi(r)=H(Hi-1(r))을 생성한다. 여기서 i=1,2,…, j 그리고 H0(r)=r이다.

사용자는 CA에게 (PKU,Hj(r),j)를 전송하여 인증서 발급 요청을 한다.

CA는 사용자와 사용자의 요청을 인증한다.

CA는 랜덤하게 선택한 Z0로부터 Z1=H(Z0)을 계산하고 Z0를 비밀리에 보관한다.

CA는 위와 같은 CERTU를 사용자에게 발행한다.

추후 사용자의 인증서를 폐지할 때에 사용자들에게 Z0을 배포한다.

〈서비스절차 : 서명과 서명 검증〉

사용자는 패스워드 r을 입력하여 개인키를 사용하며 대응하는 해쉬값 Hi(r)를 계산한다.

사용자는 메시지 M에 서명을 하여 M,SIGNU(M), CERTU와 대응하는 해쉬값 정보 (Hi(r), i)를 검증자에게 전송한다.

검증자는 CERTU에 대한 CA의 서명을 검증하여 U, PKU, D, Hi(r), Z1, j, L의 무결성을 확인한다.

Hj-i(Hi(r))=Hj(r), 0≤i<j임을 체크하여 참이면 수신된 Hi(r)이 유효하다고 판단한다.

검증자는 Dv≤D+(j-i)*L를 체크하여 참이면 De=D+(j-i)*L까지 CERTU는 유효하다고 판단할 수 있다. 여기서 De는 다음 유효배포주기이다.

검증자는 마지막으로 Z0가 Directory에 공개되었는지 혹은 CA로부터 수신되었는지를 확인하여 CERTU내의 Z1와 일회 해쉬연산을 통해 비교한다. 동일하면 해당 인증서는 폐지된 것이므로 처리를 중단하고 그렇지 않으면 처리를 계속한다.

V. 제안한 방법의 특징

이 장에서 우리는 제안한 방법의 특징을 자세히 분석한다.

1. 각 인증서에 포함되는 Z1은 동일하다.

인증서의 폐지 사유 중에 발행자와의 관계 변경, 암호학적 경계 그리고 CA의 개인키 손실이나 노출은 한 CA내의 모든 인증서가 폐지되는 원인이 된다. 따라서 한 CA내의 모든 사용자들의 인증서에 포함되는 Z1은 동일하다.

2. 해쉬값 Z1은 CA에 따라 다르다.

각 CA는 서로 다른 개인키를 가지며 다른 암호 알고리즘을 사용할 수 있다. 그러므로 해쉬값 Z1은 CA에 따라 다르게 된다. root CA의 개인키가 분실되거나 노출된다면 속하는 모든 CA가 각기 자신의 Z1를 사용자들에게 배포한다.

3. 한 CA에서도 Z1는 다를 수 있다.

CA는 root CA의 정책에 따라 주기적으로 자신의 개인키와 공개키를 변경할 수 있다. 또한 암호학적 경계에 따라 암호 알고리즘을 변경할 수 있다. 따라서 시각 t에 CA의 개인키가 변경되었다면 Z1도 변경되어야 할 것이다(Z1 → Z1'). 따라서 CA는 해당하는 Z1를 사용자에게 배포한다.

4. Z1을 위한 계산량과 저장 공간은 적다.

NOVOMODO에서는 Z1이 사용자마다 서로 상이하였으나 제안한 방법에서는 앞에서 설명한 바와 같이 동일하다. 따라서 CA는 한번의 해쉬연산을 수행하여 2개의 해쉬값(320 bit)을 저장한다. Z1는 사용자의 인증서에 포함시키기 위하여 Z0는 사용자들에게 인증서 폐지를 알리기 위하여 사용된다.

5. Z0를 배포하기 위한 처리량은 적다.

CA는 20 byte(160 bit)의 동일한 해쉬값을 자신의 CA내에 사용자들에게 배포하여야 한다. 먼저 웹 사이트들에게 Z0를 배포하여 진행중인 상호작용을 중지시키고 다음에 사용자들에게 배포한다. 웹 사이트나 각 사용자들은 20 바이트의 비교연산만으로 인증서의 폐지여부를 확인할 수 있다. 물론 Directory에 Z0를 공개할 수 도 있다.

6. 인증서 폐지의 확인은 간단하다.

Z0의 확인은 Hi(r)의 확인보다 간단하다. V는 Hj-i(Hi(r))=Hj(r)을 확인하기 위하여 j-i 번의 해쉬연산을 수행하면 된다. j가 365인 경우에는 Hi(r)을 확인하려면 V는 평균 182.5번의 해쉬연산을 수행해야 하지만 Z0의 확인을 위해서는 단 한번의 해쉬연산이 필요하다. 그리고 20 바이트의 비교연산을 통하여 대응하는 인증서와의 상호작용을 중단할 수 있다.

7. j와 L은 CA가 결정하며 하나만 필요하다.

인증서에서 j는 시간주기의 수, L은 유효기간의 길이이다. 결국 j*L=인증서의 전체 유효기간이 된다. Jianying Zhou의 프레임워크에서는 이 파라미터들을 사용자가 자유롭게 선택할

수 있다고 하였고 양종필의 프레임워크에서는 L에 해당하는 CW를 CA의 정책에 의해 선택할 수 있다고 하였다. 그러나 현실적으로는 CA의 정책에 의해 결정되어야 한다. 즉, 인터넷 뱅킹과 같은 금융거래에 사용되는 인증서는 L이 작아야 하고 도서대출 등과 같은 분야에 사용되는 인증서는 L이 길어도 된다. 따라서 이러한 것은 CA가 결정할 사항이며 사용자는 그에 따라 해쉬체인을 생성하고 j나 L중에 한 파라미터(제한한 방법에서는 j)를 선택하여 CA에 전달한다.

8. D는 인증서 발행시각과 동일하다.

D는 인증서의 유효 시작시각으로서 Jianying Zhou의 프레임워크에서는 사용자가 CA에게 인증서 발행을 요청할 때에 D를 전달한다. 이 값은 특정 주기에 인증서가 유효한지 여부를 j와 L을 이용하여 D1=D+L, D2=D+2*L, ... ,Dj=D+j*L를 계산할 때에 기준시각으로 사용된다. 그러나 이 파라미터는 사실상 사용자가 선택하기보다는 CA가 사용자에게 인증서를 발행하는 당시의 시각으로 설정할 수 있다. 즉, 인증서는 일반적으로 발행하는 즉시에 사용되기 때문이다. 따라서 D는 필요하지 않을 수도 있다.

Ⅵ. 결 론

본 논문에서 우리는 해쉬체인을 이용한 인증서상태 검증방법들의 문제점을 제시하고 그에 대한 해결방법을 제시하였다. 해쉬체인을 CA 대신 사용자가 생성, 배포하면 CA의 연산량과 메모리 사용량이 각 사용자에게 분산되어진다. 그러나 이러한 방법은 인증서에 오류가 없고 CA의 개인키가 분실이나 손실되지 않는다는 가정을 전제로 하고 있다. 따라서 실제 PKI 환경에 적합하지 않다. 본 논문에서는 이러한 문제점을 해쉬값 Z1을 인증서에 추가하여 해결하였다. 그리고 양종필의 프레임워크에서 인증서 발행절차와 전자서명 및 검증절차도 문제점을 지적하고 이를 개선하였다.

참고문헌

〔1〕 A. Malpani, R. Housley, T. Freeman, "Simple Certificate Validation Protocol (SCVP)," IETFInternetDraft,June,2002.

〔2〕 C. Adams, P. Sylvestor, M. Zolotarev and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols," IETFRFC3029,February,2001.

〔3〕 Jianying Zhou, Feng Bao and Robert Deng, "An Efficient Public-Key Framework," ICICS 2003, LNCS 2836, pp.88-99, 2003.

〔4〕 Jong-Phil Yang, Chul Sur, Hwa-Sik Jang and Kyung-Hyune Rhee, "Practial Modification of An Efficient Public-Key Framework," 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, March 2004.

〔5〕 Jose L. Munoz, Jordi Forne, Oscar Esparza, and Miguel Soriano, "A Certificate Status Checking Protocol for the Authenticated Dictionary," MMM-ACNS 2003, LNCS2776, pp.255-266, 2003.

〔6〕 M. Myers, R. Ankney, A. Mappani, S.Galperin, C.Adams,"X.509InternetPublicKeyInfrastructure OnlineCertificateStatusProtocol - OCSP,"IETFRFC2560,June,1999.

〔7〕 NIST FIPS (Federal Information Processing Standards Publication) 186-1, "Digital Signature Standard," December, 1998.

〔8〕 Paul. Kocher, "Quick Introduction to Certificate Revocation Tree (CRTs), " Technical Report, Valicert, 1999.

〔9〕 R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," IETF RFC 2458, January, 1999.

〔10〕 R. Housley, W. Ford, W. Polk and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile, " IETF RFC 3280, April, 2002.

〔11〕 Silvio Micali, "NOVOMODO : Scable Certificate Validation and Simplified PKI Management," 1st Annual PKI Research Workshop Preproceedings, pp.15-25, 2002.

〔12〕 Younggyo Lee, Injung Kim, Seungjoo Kim, and Dongho Won, "A Method for Detecting the Exposure of an OCSP Responder's Session Private Key in D-OCSP-KIS," Euro PKI 2005, LNCS 3545, pp. 215-226, 2005.

〔13〕 YounggyoLee, JeongheeAhn, SeungjooKim, and DonghoWon,"A Method for Detecting the Exposure of an OCSP Responder's Private Key using One-Time Hash Value," IJCSNS International

Journal of Computer Science and Network Security, VOL. 5 No.8, pp. 179-186, August 2005.

〔14〕 Younggyo Lee, Jeonghee Ahn, Seungjoo Kim, and Dongho Won, "A PKI System for Detecting the Exposure of a User's Secret Key," Euro PKI 2006, LNCS 4043, pp.248-250, 2006.

〔15〕 이호, 강현중, 박준홍, "D-OCSP에서의 그룹키를 이용한 CRL 배포방법에 관한 연구", 한국컴퓨터정보학회논문지 제11권 제1호, 2006. 3.

저 자 소 개

강 현 중

1980년 2월 성균관대대학교 수학교육학과 졸업

1986년 2월 연세대학교 대학원 전자계산학과 졸업(공학석사)

1996년 2월 성균관대학교 대학원 정보공학과 졸업(공학박사)

1979년 11월~1982년 2월 한국과학기술연구소(KIST)연구원

1982년 3월~1989년 2월 한화종합금융(주) 전산팀장

1989년 3월~현재 서일대학 인터넷정보과 교수

〈관심분야〉 데이타통신, 프로그래밍언어

안 정 희

1988년 2월 : 성균관대학교 정보공학과 졸업(공학사)

1993년 2월 : 성균관대학교 대학원 정보공학과 졸업(공학석사)

2000년 2월 : 성균관대학교 대학원 정보공학과 졸업(공학박사)

1996년 3월 ~ 현재 : 두원공과대학 컴퓨터정보과 부교수

〈관심분야〉 : 정보통신 보안, 전자상거래 보안, 트래픽 제어