Personal PKI에서 효율적인 서명 및 인증서 상태 검증 기법

An Efficient Signing and Certificate Status Management Scheme in Personal PKI

서 철, 신 원*, 이경현
부경대학교, 동명정보대학교*

Sur Chul, Shin Weon*, Lee Kyung-Hyune
Pukyong National University,
TongMyong University*

요약

최근 Personal Area Network (PAN)에서 신뢰성 있고 인증된 서비스를 제공하기 위하여 Personal Public Key Infrastructure (PKI)가 소개되었다. 그러나, PAN을 구성하는 대다수의 모바일 디바이스들은 일반적으로 제한된 컴퓨팅 능력을 가지므로, 기존 PKI에서 사용되는 공개키 서명 기법과 CRL등의 인증서 상태 검증 기법 같이 많은 오버헤드를 요구하는 기법들은 적합하지 않다. 본 논문에서는 PAN을 구성하는 모바일 디바이스들을 위한 효율적인 서명 생성과 인증서 상태 검증을 지원하는 새로운 기법을 제안한다. 제안 기법은 해쉬체인 기법을 사용함으로써 전자서명 생성시 모바일 디바이스의 계산상 오버헤드를 감소시킬 수 있으며 또한, 인증서 상태 검증을 위한 통신상의 오버헤드를 감소시킬 수 있다.

Abstract

Recetly, the term Personal Public Key Infrastructure (PKI) was introduced to supprot reliable and authenticated service in a Personal Area Network (PAN). However, traditional public key signature schemes and certificate status management are not suitable for a PAN environment since mobile devices that constitute the PAN have limited computing capability. In this paper, we propose a new scheme that efficiently provides signature generation and certificate status management for mobile devices. Based on hash chain technique, we intend to reduce computational overhead on signature generation, and further, to minimize communication overhead for managing certificate status.

I. 서 론

최근 PDA와 랩탑과 같이 네트워킹 능력을 가진 이동 디바이스들은 이미 널리 사용되어지고 있으며, 이동 디바이스들의 기능과 성능은 나날이 발전하고 있다. 이러한 기술들의 급속한 발전으로 인하여 미래의 모바일 통신에서 네트워크에 대한 접근방법과 네트워크 접근에 사용되는 터미널 유형은 현재 사용되고 있는 방식과 다를 것으로 예상되어지며 물리적으로 사용자에 근접한 범위 내에서 분산 다기능 모바일 터미널은 로컬 무선 통신을 통해 연결되는 몇몇의 컴퍼넌트들을 구성할 것으로 예상되어진다.

특히, 개인적인 범위 내에서 구성되어지는 고정된 수의 이동 컴퍼넌트들간의 상호연결된 무선 네트워크를 Personal Area Network (PAN)라 부르며, 전형적으로 Bluetooth 등을 이용하여 10미터이내의 범위에서 구성된다. PAN내에서의 통신은 기밀정보와 개인 데이터가 주를 이루므로 신뢰성 있고 인증된 통신 서비스가 필수적이며 이러한 보안 서비스를 위한

기반구조로써 Personal PKI(Public Key Infrastructure)가 제안되었다[2].

그러나, PAN을 구성하는 대다수의 모바일 디바이스들은 일반적으로 제한된 컴퓨팅 능력을 가지므로, 기존 PKI에서 사용되는 공개키 서명 기법과 CRL등의 인증서 상태검증 기법 같이 많은 오버헤드를 요구하는 기법들은 적합하지 않다. 따라서, 본 논문에서는 PAN을 구성하는 모바일 디바이스들을 위한 효율적인 서명과 인증서 상태 검증을 지원하는 새로운 기법을 제안한다. 제안 기법은 해쉬체인 기법을 사용함으로써 컴퓨팅 파워가 약한 모바일 디바이스가 효율적으로 전자서명을 수행할 수 있으며 통신 개체의 인증서 상태를 검증할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 Personal PKI에 대하여 살펴본 후 제안 기법의 설계를 위한 기반기술에 대하여 설명한다. Personal PKI에서 효율적인 서명 생성 및 인증서 상태 검증 기법을 4장에서 제안한 후 5장에서 결론을 맺는다.

Ⅱ. 관련연구

1. Personal PKI

PAN에서 신뢰성있고 인증된 서비스를 제공하기 위한 기반구조로써 Personal PKI가 제안되었다[2]. Personal PKI에서는 적어도 하나의 디바이스가 인증기관(Certificate Authority, CA)의 역할을 수행하며 PAN내의 모든 디바이스에 대하여 공개키 인증서를 발행하고 공개키 인증서의 상태에 대한 정보를 제공한다. 이러한 디바이스를 Personal CA라하며 모바일 디바이스들과 Personal CA간의 보안결합(Security Association, SA)를 형성하기 위하여 아래와 같은 사항이 요구된다.

- Personal CA는 사용자가 명령어를 입력할 수 있어야 하고 사용자에게 결과를 디스플레이할 수 있어야 한다. 예를 들면, 모바일 폰, PDA 그리고 PC등이 Personal CA로써 가능한 디바이스들이다.
- 모바일 디바이스는 사용자가 일련의 숫자를 입력할 수 있어야 하며 Y/N로써 성공과 실패를 지시할 수 있어야 한다.

Personal CA의 주요 기능은 초기화 부분과 인증서 상태 관리 부분으로 나누어진다. 초기화 부분에서는 모든 모바일 디바이스가 자신의 공개키 인증서를 가지고 PAN에 장비될 수 있게 확장된 imprinting 기법[3]을 사용하고 있으며, 인증서 관리 부분에서는 전통적인 온라인 인증서 상태 프로토콜(Online Certificate Status Protocol, OCSP)과 인증서 폐지 리스트(Certificate Revocation List, CRL)의 사용을 가정하고 있다.

2. 제안 기법 설계를 위한 기반기술

해쉬체인 기법은 패스워드 도청에 대한 보호기법으로 Lamport에 의해 처음 소개되었다[4]. 해쉬체인 기법은 해쉬함수의 일방향성에 기반하며 아래의 식 (1)과 같이 입력 스트링 x 에 대하여 반복적으로 해쉬함수 $h()$ 를 적용함으로써 구현되어질 수 있다.

$$h^n(x) = h(h(h(\cdots h(x)\cdots))) \quad (1)$$

서버지원 서명 기법(Server Supported Signature)은 해쉬체인 기법과 전통적인 서명 기법에 기반한 부인방지(Non-Repudiation) 기법으로 Asokan 등에 의해 제안하였다[1]. 서버지원 서명 기법에서는 사용자의 전자서명 생성을 서명서버가 대신 수행함으로써 공개키 연산의 수행에 따른 사용자의 계산상의 오버헤드를 감소시킬 수 있다.

Zhou 등은 해쉬체인 기법에 기반하여 인증서의 최대 생명주기를 짧은 기간들로 나누고 인증서가 인증서 소유자(혹은 조직적인 환경에서 관리자)의 통제아래 어떤 기간의 끝 지점에서 만기될 수 있는 새로운 공개키 프레임워크를 제안하였다[6]. 그러나, 이는 비밀키 훼손 등으로 인해 발생하는 인증서 소유자에 의

한 인증서 폐지의 필요성에 관해서만 고려하고 있으므로 실제 PKI 환경에는 심각한 문제점을 발생시킨다. 이에 대한 해결방안으로 Yang등은 컨트롤 윈도우 메카니즘에 기반한 개선된 공개키 프레임워크를 제안하였다[5]. 컨트롤 윈도우 메카니즘에서 CA가 사용자에 대한 인증서 발급시 사용자의 컨트롤 윈도우를 설정하며 사용자는 컨트롤 윈도우 기간 동안 해쉬체인에 의존하여 인증서 상태를 검증할 수 있다. 따라서, 사용자의 공개키 인증서 검증을 위한 통신상, 계산상의 오버헤드를 신뢰성있게 감소시킬 수 있다.

Ⅲ. 제안 시스템

본 장에서는 해쉬체인에 기반하여 Personal PKI에서 컴퓨팅 파워가 약한 모바일 디바이스가 효율적으로 전자서명을 수행할 수 있으며 통신 개체의 인증서 상태를 검증할 수 있는 기법을 제안한다.

1. 시스템 모델

제안 시스템은 다음과 같은 환경을 가정한다.

- PAN은 n 개의 모바일 디바이스들로 구성되며 각각의 디바이스들은 무선 인터페이스를 통하여 통신한다.
- PAN 형성시 적어도 한 개의 디바이스가 Personal CA의 역할을 수행하며, 다른 모든 모바일 디바이스들은 자신의 공개키 인증서를 가지고 PAN에 장비된다.
- 자신의 전자서명 생성을 위임하기를 원하는 모바일 디바이스는 더욱 강력한 컴퓨팅 능력을 가진 디바이스를 찾을 수 있거나 이미 알고 있어야 한다. 예를 들면, 모바일 디바이스의 부트스트랩 단계에서 강력한 컴퓨팅 능력을 가진 디바이스들의 리스트가 제공되어질 수 있다.

제안 시스템은 Personal CA, SSs(Signature Servers), 모바일 디바이스들로 구성되어지며 각각의 구성요소는 다음과 같은 역할을 수행한다.

- Personal CA : Personal CA는 PAN내의 모든 디바이스에 대한 공개키 인증서를 발급하며 인증서 상태 정보를 제공한다. 또한, 사용자가 명령어를 입력할 수 있고 사용자에게 결과를 디스플레이할 수 있는 기능을 가지고 있다.
- SSs(Signature Servers) : SSs는 SS(Signature Server)로 불리는 특별한 k 디바이스 $(1 \le k < n)$ 의 집합으로 구성된다. 각각의 SS는 컴퓨팅 파워가 약한 모바일 디바이스를 대신하여 전자서명을 수행한다.
- 모바일 디바이스 : PAN에 장비되는 컴퍼넌트들로써 일반적으로 제한된 컴퓨팅 파워를 가진다.

2. 용어정리

본 논문에서는 아래와 같은 용어를 사용한다.

- PCA : Personal CA의 식별자
- SS_h : h 번째 Signature Server의 식별자, $1 \le h \le k$
- S, R : 메시지 송신자와 수신자의 식별자
- PK_X, PS_X : 모바일 디바이스 X 의 공개키/비밀키 쌍
- $h_X()$: 모바일 디바이스 X의 일방향 해쉬함수. 모바일 디바이스는 자신의 식별자를 사용하여 해쉬함수를 개인화한다. 예, $h(X,M)$에서 X는 식별자, M은 메시지
- SK_X^i : 모바일 디바이스 X의 $(n-i)$ 번째 서명키. 랜덤하게 선택되어진 SK_X를 기반으로 하여 모바일 디바이스 X는 아래와 같은 해쉬체인을 생성

  $SK_X^0 = SK_X, SK_X^i = h_X^i(SK_X) = h_X(SK_X^{i-1})$

  SK_X^n을 모바일 디바이스 X의 루트 서명키라 하며, 현재 i값을 서명 카운터, SK_X^i를 X의 현재 서명키라고 한다.
- RK_X^i : 모바일 디바이스 X의 $(m-i)$ 번째 인

증서 상태 관리키. 랜덤하게 선택되어진 RK_X를 기반으로 하여 모바일 디바이스 X는 아래와 같은 해쉬 체인을 생성 $RK_X^0 = RK_X, RK_X^i = h_X^i(RK_X) = h_X(RK_X^{i-1})$ RK_X^m을 모바일 디바이스 X의 루트 인증서 관리키라 하며, 현재 j값을 인증서 관리 카운터, RK_X^j를 X의 현재 인증서 관리키라고 한다.

- $Sig_X()$: 모바일 디바이스 X의 비밀 키를 통한 전자서명
- $Cert_X$: 모바일 디바이스 X 의 공개키 인증서
- $X \rightarrow Y$: msg : X 로부터 Y 로 메시지 msg 을 전송

3. 초기화 프로토콜

제안 시스템에서 모든 모바일 디바이스들은 초기화 프로토콜을 통하여 PAN에 장비된다. 프로토콜의 자세한 설명은 아래와 같다.

[단계 1] Personal CA는 자신의 식별자와 공개키를 모바일 디바이스 M_i 에게 전송한다.

$$PCA \rightarrow M_i : PCA, PK_{PCA}$$

[단계 2] 모바일 디바이스 M_i 는 자신의 공개키/비밀키 쌍을 생성한 후, 해쉬체인 기법을 이용하여 루트 서명키 $SK_{M_i}^n$ 과 루트 인증서 관리키 $RK_{M_i}^m$ 를 생성하고 자신의 식별자와 공개키와 함께 Personal CA에게 전송한다.

$$M_i \rightarrow PCA : M_i, PK_{M_i}, n, SK_{M_i}^n, m, RK_{M_i}^m$$

[단계 3] 단계 2까지 성공적으로 수행한 이후, Personal CA와 모바일 디바이스 M_i 는 다음과 같은 매뉴얼 인증(Manual Authentication)을 수행한다: Personal CA는 랜덤키 k 를 생성한 후, $PCA, PK_{PCA}, M_i, PK_{M_i}, SK_{M_i}^n, RK_{M_i}^m$ 에 대한 MAC값을 계산한다. 이후, MAC값과 랜덤키 k 값은 Personal CA에 디스플레이되고 사용자는 MAC값과 랜덤키 k 를 모바일 디바이스에 입력하여 위의 파라메터에 대한 MAC값을 재계산한다. 만약, 두 개의 MAC값이 일치하면, 모바일 디바이스는 성공신호를 표시하며 그렇지 않으면, 실패신호를 표시한다.

[단계 4] 만약 단계 3에서 모바일 디바이스가 성공신호를 표시하면, 사용자는 Personal CA에게 모바일 디바이스에 대한 공개키 인증서를 생성할 것을 지시한다. 인증서 생성시 효율적으로 인증서 상태 검증을 수행하기 위하여 Personal CA는 시스템 보안 정책에 기반하여 컨트롤 윈도우를 설정하며 아래와 같은 인증서를 모바일 디바이스에게 전송한다.

$$PCA \rightarrow M_i :$$
$$Cert_{M_i} = Sig_{PCA}(Ser\#, M_i, PK_{M_i}, n, SK_{M_i}^n, m, RK_{M_i}^m, CW)$$

[단계 5] 인증서를 전송받은 모바일 디바이스는 다음과 같이 두가지 검증을 수행한다. 1) 모바일 디바이스는 Personal CA의 공개키를 사용하여 인증서상의 전자서명을 검증한다. 2) 모바일 디바이스는 인증서 내의 모든 데이터 필드들이 올바른 값인지를 검증한다. 만약 모든 검증이 성공하면, 모바일 디바이스에 대한 초기화 프로토콜을 종료한다.

4. 효율적인 서명 생성 및 인증서 상태 검증 프로토콜

[단계 1] 메시지 msg 에 대하여 서명하기를 원하는 모바일 디바이스 S 는 아래와 같은 서명 요청 메시지를 특정 Signature Server SS_h 에게 전송한다.

$$S \rightarrow SS_h : Cert_S, H(msg), i, SK_S^i, j, RK_S^j$$

[단계 2] 서명 요청 메시지를 전송받은 SS_h 는

S 의 인증서와 현재 서명키에 대한 검증을 다음과 같이 수행한다. 1) PCA 에게 S 의 인증서가 유효한지 OCSP를 통하여 질의한 후, 만약 유효하다면 인증서내의 루트 인증서 관리키를 기반하여 현재 인증서 관리키의 유효성을 검증한다. 즉, $h_S^{m-i}(RK_S^i) = RK_S^m$ 를 만족하는지를 검증하고 컨트롤 윈도우에 기반하여 인증서 유효성 시작 지점과 끝 지점을 세팅한다. 따라서, 인증서 유효성 끝 지점까지 S 의 인증서 상태 검증은 OCSP 질의없이 인증서 관리키를 기반으로 수행되어질 수 있다. 2) S 의 루트 서명키에 기반하여 현재 서명키를 검증한다. 즉, $h_S^{n-i}(SK_S^i) = SK_S^n$ 을 검증한다.

위의 모든 검증이 성공하면 SS_h 는 candidate NRT(Non-Repudiation Token)로써 $Sig_{SS_h}(S, SS_h, H(msg), i, SK_S^i)$ 를 생성한 후 자신의 인증서, 현재 인증서 관리키와 함께 S 에게 전송한다.

$$SS_h \rightarrow S : Cert_{SS_h}, j, RK^j{}_{SS_h},$$
$$Sig_{SS_h}(S, SS_h, H(msg), i, SK_S^i)$$

[단계 3] SS_h 로부터 candidate NRT를 수신한 S 는 단계 2.와 같이 SS_h 에 대한 인증서 상태 검증을 수행한 후, SS_h 의 인증서 상태 검증을 OCSP 질의 없이 인증서 관리키를 기반으로 수행하기 위하여 컨트롤 윈도우에 기반하여 인증서 유효성 시작 지점과 끝 지점을 세팅한다. 그리고, candiate NRT상의 전자서명을 검증한다. 모든 검증이 올바르다면, S 는 다음 서명키 SK_S^{i-1} 을 계산한 후 아래와 같은 유효한 NRT를 생성하고 현재 서명 카운터를 $i-1$ 로 대처한다.

$$NRT = Sig_{SS_h}(S, SS_h, H(msg), i, SK_S^i), SK_S^{i-1}$$

마지막으로, S 는 유효한 NRT와 함께 SS_h 의 인증서 상태 정보 등을 메시지를 수신할 모바일 디바이스 R 에게 전송한다.

$$S \rightarrow R : Cert_{SS_h}, j, RK^j{}_{SS_h}, msg, NRT$$

[단계 4] 모바일 디바이스 R 은 SS_h 의 인증서 유효성과 NRT상의 서명의 유효성을 검증한 후, NRT내의 SK_S^{i-1} 이 candidate NRT상의 SK_S^i 의 pre-image인가를 검증한다. 만약 모든 검증이 올바르다면, 유효한 메시지와 부인방지 토큰(NRT)를 획득한다.

Ⅳ. 결 론

본 논문에서는 PAN에서 제한된 컴퓨팅 파워를 가진 모바일 디바이스가 효율적으로 전자서명을 위임할 수 있고 상대방의 인증서 상태를 검증할 수 있는 기법을 제안하였다. 제안 기법은 안전한 전자서명 위임과 효율적인 인증서 상태 검증을 위하여 해쉬체인 기법을 사용함으로써 컴퓨팅 파워가 약한 모바일 디바이스의 계산상의 오버헤드를 감소시켰으며 컨트롤 윈도우 메카니즘에 기반하여 인증서 상태 질의 횟수를 줄임으로써 통신상의 오버헤드를 감소시켰다.

▌참고문헌▐

[1] N. Asokan, G. Tsudic and M. Waidner, "Server-Supported Signatures", European Symposium on Research in Computer Security, September 1996.

[2] C. Gehrmann, K. Nyberg and C. Mitchell, "The personal CA - PKI for a Personal Area Network", Proceedings - IST Mobile & Wireless ommunications Summit 2002, June 2002.

[3] F. Stajano and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless

networks", Proceedings of the 7th International Workshop on Security Protocols, LNCS 1796, 1999.

[4] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, 24(11), 1981.

[5] J-P. Yang, C. Sur, H.S. Jang and K.H. Rhee, "Practical Modification of An Efficient Public-Key Framework", 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, March 2004.

[6] J. Zhou, F. Fao and R. Deng, "An Efficient Public-Key Framework", 5th International Conference on Information and Communications Security, LNCS 2836, October 2003.