보안공학연구논문지 *Journal of Security Engineering*
Vol.13, No.1 (2016), pp.27-40
http://dx.doi.org/10.14257/jse.2016.02.06

하이브리드 인증을 위한 키관리서버 모델

이병천1)

Model of Key Management Server for Hybrid Certification

Byoungcheon Lee[1)]

요 약

한 사용자가 복수의 컴퓨팅기기들을 사용하게 되는 유비쿼터스 환경에서는 각 기기마다 사용자의 인증키를 안전하게 배포하고 이용할 수 있어야 상호인증 환경에서 통신할 수 있다. 그런데 복수기기 환경에서는 이러한 키관리가 매우 복잡하고 불편하게 되는데, 이 문제를 해결하기 위하여 사용자 스스로 하나의 인증서에 기반하여 자체확장인증을 통해 기기별로 서로 다른 인증키를 배포하고 활용할 수 있도록 하는 하이브리드 키관리[1][2] 방식이 제안되었다. 이러한 접근방법에서 자체확장인증키를 배포하기 위해서는 사용자의 개인키를 장착한 키관리서버를 운영할 필요가 있는데 비전문가인 개인이 자신만이 이용하기 위한 키관리서버를 직접 운영하는 것은 매우 불편하고 부담스러운 일이 아닐 수 없다.

이 논문에서는 실세계에서 안전하고 편리하게 사용 가능한 하이브리드 키관리의 시나리오를 검토하고, 이를 기반으로 사용자의 개인키를 가지고 있을 필요가 없도록 운영되는 하나의 키관리서버를 다수의 사용자가 함께 사용할 수 있도록 키관리 프로토콜을 설계하였다. 아울러 웹 환경에서 동작하는 키관리서버를 구현하고 이것의 활용성을 분석하였다.

핵심어 : 하이브리드 인증, 하이브리드 키관리, 자체확장인증, 유비쿼터스 컴퓨팅, 키관리서버

Abstract

In ubiquitous computing environment where users own and use multiple computing devices each device has to be equipped with certified key of the user to be able to communicate with other users/devices in authentic manner. But key management in multiple devices becomes very complex and inconvenient. To solve this complexity hybrid key management scheme [1][2] was proposed in which user issues self-extended certificate (SEC) by himself to each device based on user's single certificate issued by certification authority. In this approach a key management server which is equipped with user's private key was used to generate and distribute SEC to each device, but it is very inconvenient and insecure for a non-professional to run a key management server of this kind.

In this paper we consider more realistic scenario of hybrid key management and design a new hybrid key management scheme such that a single server which does not need to have user's private key can provide hybrid key management service for multiple users. We also implement a web-based hybrid key management server and analyze its utilizations.

Keywords : hybrid certification, hybrid key management, self-extended certificate, ubiquitous computing, key management server

Received(January 16, 2016), Review request(January 17, 2016), Review Result(1st: February 03, 2016)

Accepted(February 11, 2016), Published(February 29, 2016)

[1]Dept. of Information Security, Joongbu Univ., 305 Dongheon-ro, Deogyang-gu, Goyang-si, Gyeonggi-do, 10279, Korea.
email: sultan@joongbu.ac.kr

* 이 논문은 교육과학기술부의 재원으로 지원을 받아 수행된 중부대학교 산학협력선도대학(LINC) 육성사업의 연구결과로 수행되었음.

ISSN: 1738-7531 JSE

1. 서론

1.1 유비쿼터스 환경에서의 하이브리드 키관리

공개된 인터넷 환경에서 사용자간, 기기간 상호 분명한 신분인증을 제공하고 신뢰성 있는 통신을 할 수 있도록 하기 위해서는 전자서명 기반의 인증이 필요하며 이를 위해서는 기기마다 인증키를 설치하여 사용하도록 해야 한다. 사용자의 신분인증을 제공하기 위한 방법으로는 인증기관으로부터 X.509 기반의 인증서[3]를 발급받아 사용하는 것이 가장 일반적인 방법이다.

발급받은 키를 컴퓨팅기기 내에 안전하게 관리하는 것이 더욱 중요해지고 있는데 최근 안전한 키관리 도구로서 하드웨어보안모듈을 활용할 수 있는 환경이 확대되고 있다. 하드웨어보안모듈이란 기존의 스마트카드 기술을 기반으로 여러 가지 편리한 인터페이스를 가진 방식으로 확장된 것으로 USIM, USB보안토큰, TPM[4-6], NFC[7][8] 등의 다양한 인터페이스 형식의 제품으로 발전하고 있다. 이러한 하드웨어 보안모듈들은 암호키의 안전한 저장소로서의 역할뿐만 아니라 RSA기반의 키생성, 암호화, 전자서명, 해쉬, 난수생성 등의 기본 보안기능들을 가지고 있어서 비밀키의 외부 누출이 없이 암호화, 전자서명 등의 보안기능을 하드웨어보안모듈 내부에서 수행할 수 있도록 하는 안전한 사용환경을 제공한다. 개인이 소유하는 하드웨어보안모듈의 내부에서 생성되고 관리되는 암호키는 밖으로 꺼낼 수 없는 안전한 암호키이며 이것을 사용자의 인증키로 활용할 수 있도록 할 수 있다면 매우 안전하면서도 편리하게 사용할 수 있을 것이다.

그런데 한 사용자가 여러대의 기기들을 사용하게 되는 복수기기 환경에서는 각 기기마다 사용자의 인증키를 배포하는 것이 쉽지 않은 문제이다. 첫 번째 방법으로, 하나의 인증키를 발급받은 후 기기마다 똑같은 인증키(개인키)를 복사하여 사용하는 것은 보안적인 측면에서 매우 위험한 방법이다. 공격자는 인증키(개인키)를 복사해내는 기능을 집중 공격하게 될 것이다. 또한 IC카드에 기반한 하드웨어보안모듈에서 키쌍을 생성하고 인증서를 발급받는 방식으로 이용하면 개인키를 하드웨어보안모듈의 밖으로 복사할 수 없으므로 이런 방법론을 사용할 수 없다. 두 번째 방법으로, 기기마다 각각 독립적인 인증서를 발급받아 사용하는 것은 사용자가 각각의 기기를 이용하여 인증서 발급과정을 여러 번 수행해야 하는 불편함이 있고 기기마다 서로 다른 인증서를 발급받았으므로 이들을 모두 안전하게 관리해야 하는데 기기의 수가 증가할수록 이것은 매우 어려운 문제가 된다. 기기가 파손되거나 분실하게 되는 경우 인증서를 폐지해야 하는데 이것을 관리하는 것이 인증기관의 입장에서는 매우 큰 부담이 될 것이다. 복수기기를 사용하게 되는 유비쿼터스 환경에서는 이 두 가지 모두 바람직하지 않은 방법이다[1][2].

이러한 인증키 관리의 복잡성을 해결하기 위하여 사용자 스스로 하나의 인증서에 기반하여 자체확장인증을 통해 기기별로 서로 다른 인증키를 만들어 활용할 수 있도록 하는 하이브리드 키관리 방식이 제안되었다[2]. 이 방법은 개인이 자신의 영역내에서 마치 인증기관처럼 자신의 소유기

기에 대하여 자체확장인증서를 발행하여 사용하는 방식으로서, 개인이 몇 개의 기기를 사용하든지 상관없이 필요할 때마다 스스로 기기를 등록하고 인증키를 발행하여 사용할 수 있다는 장점이 있다. 소유기기들은 모두 하나의 인증서에 논리적으로 연결되어 있어서 타인에게 기기소유자의 신분을 확인받을 수 있다. 이것은 개인이 사용하는 컴퓨팅 기기의 수가 크게 증가하고 있는 유비쿼터스 환경에서 개인 사용자가 소유기기들의 멤버십을 일관된 형식으로 관리하는데 있어서 큰 편의성을 제공할 수 있는 방식이다.

1.2 하이브리드 인증을 위한 키관리서버

하이브리드 인증을 활용하기 위해서는 사용자가 인증서를 이용하여 자체확장인증서를 생성하는 과정이 필요하며 이를 위한 키관리서버를 운영할 필요가 있다. [2]에서는 사용자의 개인키를 장착하고 있어야 하는 키관리서버를 개인이 직접 운영하는 것을 고려하고 있는데 키관리서버의 자세한 동작 프로토콜을 제시하지는 않고 있다. 그런데 비전문가인 개인이 자신만이 이용하기 위한 키관리서버를 직접 운영하는 것은 매우 불편하고 부담스러운 일이 아닐 수 없다. 또한 사용자의 개인키를 장착하고 있는 서버를 개인이 운영한다면 공격자들의 집중 공격을 받게 될 것이다. 모든 개인 사용자가 키관리서버를 운영하는 것은 안전하게 운영하기도 어렵고 경제적으로도 부담이 클 것이다.

이 논문에서는 실세계에서 안전하고 편리하게 사용 가능한 하이브리드 키관리의 시나리오를 검토하고, 이를 기반으로 사용자의 개인키를 보유할 필요가 없도록 운영되는 하나의 키관리서버를 다수의 사용자가 함께 사용하는 방식으로 키관리 프로토콜을 설계하였다. 하나의 키관리서버를 다수의 사용자가 공유할 수 있다면 전문가가 운영하면서 안전한 서비스를 제공할 수 있을 것이다. 이러한 설계를 기반으로 웹 환경에서 동작하는 키관리서버를 구현하였고 이것의 활용성을 분석하였다.

1.3 논문의 구성

2장에서는 [2]에서 제시한 하이브리드 키관리 방식에 대해 간단히 소개한다. 3장에서는 하나의 키관리서버를 다수의 사용자가 함께 사용할 수 있도록 하는 방식으로 설계한 개선된 하이브리드 키관리 프로토콜을 제시한다. 4장에서는 설계된 내용을 바탕으로 웹 환경으로 구현한 키관리서버의 사례를 제시하고 이것의 활용성을 분석한다. 5장에서는 결론 및 향후 연구과제를 제시한다.

2. 하이브리드 키관리

여기에서는 [2]에서 제안된 하이브리드 키관리 시스템에 대해 간단히 소개하고자 한다. 하이브리

드 키관리 방식에서는 각 컴퓨팅기기들에 인증키를 안전하게 저장하고 이용할 수 있는 IC카드 기반의 하드웨어보안모듈이 장착되어 있는 환경을 가정한다. 하드웨어보안모듈 내부에서 키쌍을 생성하고 저장하며 개인키를 이용하는 연산을 하드웨어보안모듈 내부에서 수행하는 방식으로 보안서비스를 설계함으로써 키관리의 안전성을 근본적으로 보장할 수 있다. 외부의 신뢰기관으로 개인의 신분을 확인하고 인증서를 발급해주는 인증기관이 존재하며 개인은 하나의 키관리서버와 복수의 일반 컴퓨팅기기를 소유하고 있다고 가정한다. 상세 키관리 프로토콜은 다음과 같다.

2.1 인증서 발급 (인증기관→키관리서버)

개인은 키관리서버를 이용하여 인증기관에 접속하여 인증서 발급을 요청하고 인증서를 발급받는다. 키쌍은 키관리서버에 장착된 하드웨어보안모듈에서 생성하며 인증서의 개인키는 하드웨어보안모듈 내부에 저장된다.

2.2 자체확장인증서명 발행 (키관리서버→소유기기)

개인은 하드웨어보안모듈이 장착된 소유기기를 동작하여 새로운 키쌍을 생성하고 여기에서 생성된 공개키를 키관리서버로 전송한다. 키관리서버는 전송된 공개키를 포함하여 자체확장인증서명을 발행하고 이것을 소유기기에 전달, 저장하게 한다. 자체확장인증서명이 발행되면 생성된 키는 자체확장인증키가 된다.

2.3 자체확장인증키의 활용

사용자는 자체확장인증키를 장착한 소유기기를 이용하여 타인과 암호화, 전자서명, 전자봉투, 인증 등의 보안통신을 수행할 수 있다. 예를 들어 송신자가 자체확장인증키를 사용하여 서명된 메시지를 전송한 경우 수신자는 이것을 검증하기 위해서는 다음의 세가지를 함께 확인해야 한다.

1) 인증서 검증을 통해 송신자의 신분을 확인한다.
2) 자체확장인증서명을 검증하여 사용된 자체확장인증키가 사용자의 인증을 받은 것임을 확인한다.
3) 자체확장인증키로 생성된 서명문의 유효성을 확인한다.

이 과정에서 수신자는 상대방 송신자의 신분을 인증서를 통해 확인할 수 있고 실제 통신에 사용된 자체확장인증키가 사용자의 승인을 받은 유효한 것인지 확인할 수 있다.

3. 개선된 하이브리드 키관리 프로토콜

3.1 키관리 서비스 시나리오

위에서 제시된 것과 같은 키관리 프로토콜에서는 인증개인키를 장착하고 있으며 서버로 상시

동작하고 있는 형태의 키관리서버를 사용자가 직접 운영해야 하는데 이것은 비전문가인 사용자가 안전하게 운영하기가 쉽지 않은 단점이 있다. 사용자는 상시 동작하는 서버를 운영하기보다는 사용자가 필요시에만 사용할 수 있도록 클라이언트 입장에서 하이브리드 키관리를 수행하기를 바랄 것이다. 그러므로 키관리서버를 사용자 영역의 밖에서 운영하는 모델을 고려할 필요가 있다. 또한 하나의 키관리서버를 많은 사용자가 함께 사용한다면 경제적인 측면의 이득도 얻을 수 있다. 그런데 자체확장인증서를 발행하기 위해서는 사용자가 소유한 인증개인키를 사용하여 서명해야 하는데 이것을 키관리서버가 가지고 있을 필요가 없도록 수정되어야 한다.

이런 문제를 해결하기 위해 이 논문에서는 사용자의 인증개인키를 가질 필요가 없고 다수의 사용자가 함께 사용할 수 있는 형태로 사용자 영역의 밖에서 운영되는 하나의 키관리서버의 존재를 고려하고, 사용자는 인증키를 장착한 하나의 마스터기기와 자체확장인증키를 장착하게 되는 복수개의 추가기기를 사용하는 경우를 고려하여 하이브리드 키관리 프로토콜을 설계하였다. 키관리서버는 마스터기기와 추가기기 사이의 자체확장인증서 발행 프로세스를 중개하는 역할만 하며, 사용자의 인증개인키는 마스터기기 내에 안전하게 저장되고 자체확장인증서 발행도 마스터기기 내에서 수행된다.

이러한 키관리 모델에서 프로토콜의 참여자와 그 역할은 다음과 같다.

1) 인증기관: 사용자의 신분을 확인하고 사용자의 마스터기기에 인증서를 발급하는 역할을 한다.
2) 키관리서버: 사용자의 추가기기와 마스터기기 사이의 자체확장인증서 발행 프로세스를 중개하는 역할을 한다. 사용자의 계정에 하나의 인증서와 이를 기반으로 발행된 다수의 자체확장인증서 정보를 저장하고 관리하는 역할을 한다.
3) 사용자의 마스터기기: 사용자는 하드웨어보안모듈이 장착된 하나의 마스터기기를 보유하고 있으며 이것을 이용하여 인증기관으로부터 인증서를 발급받는다. 인증서와 인증개인키는 마스터기기 내부의 하드웨어보안모듈에 저장된다. 사용자는 마스터기기의 인증개인키를 이용하여 추가기기에 자체확장인증서를 발행한다.
4) 사용자의 추가기기: 사용자는 하드웨어보안모듈을 장착한 복수의 추가기기를 보유하고 있으며 이것은 자체적으로 새로운 키쌍을 생성하고 이것에 대하여 마스터기기로부터 자체확장인증서를 발급받는다. 자체확장인증서를 발급받으면 생성된 키는 자체확장인증키가 된다. 사용자는 타인과의 보안통신에 추가기기를 이용한다.

[2]에서는 자체확장인증을 위해 기존의 인증서 포맷을 따르지 않고 간소화된 별도의 자체확장인증서명 양식을 사용하는 것을 제안하였다. X.509 표준을 따르는 인증서는 발급한 인증기관이 취소할 수 있고 취소여부를 확인하기 위해 인증서취소목록을 확인하는 등 많은 계산량과 통신량이 요구된다. 자체확장인증은 발급자와 사용자가 동일인이므로 이러한 취소 메커니즘을 이용할 필요가 없어서 인증서를 사용하지 않고 별도의 간소화된 서명을 활용하도록 제안된 것이다. 그러나 X.509의 인증서 양식이 표준화되어 널리 사용되고 있고 소프트웨어로 잘 구현되어 있어서 우리는 X.509

인증서 양식을 그대로 사용하여 자체확장인증서를 생성하도록 하였으며, 다만 자체확장인증서에서 인증서 취소 기능은 사용하지 않는 것으로 하였다.

3.2 하이브리드 키관리 프로토콜

하이브리드 키관리 프로토콜은 인증서 발급, 사용자 등록, 자체확장인증서 발행의 3가지 세부 프로토콜로 이루어져 있다.

1) 인증서 발급: 사용자는 마스터기기를 이용하여 인증기관에 접속하고 인증서를 발급받는다. 인증서 발급에 필요한 키쌍은 마스터기기에 내장된 하드웨어보안모듈에서 생성하며 인증서와 인증개인키도 하드웨어보안모듈 내에 저장된다.
2) 사용자등록: 사용자는 마스터기기를 이용하여 하이브리드 키관리서버에 접속하여 인증서를 제출하고 계정을 등록한다. 하이브리드 키관리서버는 인증서의 유효성을 검증하여야 하며 사용자 계정정보에 인증서를 저장한다.
3) 자체확장인증서 발행: 사용자는 자신이 소유한 마스터기기와 추가기기를 다음과 같이 순차적으로 동작시켜 키관리서버와의 통신을 통하여 자체확장인증서를 발행하고 추가기기에 설치한다. 프로토콜은 다음과 같은 세부 절차로 이루어져 있다.
    가. 기기등록 신청: 사용자는 추가기기를 이용하여 키관리서버에 접속하고 기기등록 신청을 한다. 이 과정에서 사용자 추가기기는 새로운 키쌍을 생성하고 <사용자명, 기기명, 공개키> 정보를 제출한다. 제출하는 공개키 정보의 유효성을 증명하기 위하여 제출하는 정보를 생성한 개인키로 서명하여 전송한다.
    나. 기기등록 허가: 사용자는 마스터기기를 이용하여 키관리서버에 접속하고 기기등록 신청 내역을 검색하고 정보를 받아온다. 사용자는 기기등록 신청한 내용을 확인하고 자신이 추가기기를 이용하여 신청했던 내용과 동일한 경우 인증개인키를 이용하여 자체확장인증서를 발행하고 기기등록을 허가한다. 발행한 자체확장인증서를 키관리서버에 전송하고 키관리서버는 이것을 저장한다.
    다. 기기등록 완료: 사용자는 기기등록을 신청했던 추가기기를 이용하여 키관리서버에 접속하고 기기등록 허가 내역을 검색한다. 자체확장인증서가 발행된 경우 이것을 다운로드하여 저장한다.

3.3 자체확장인증키의 활용

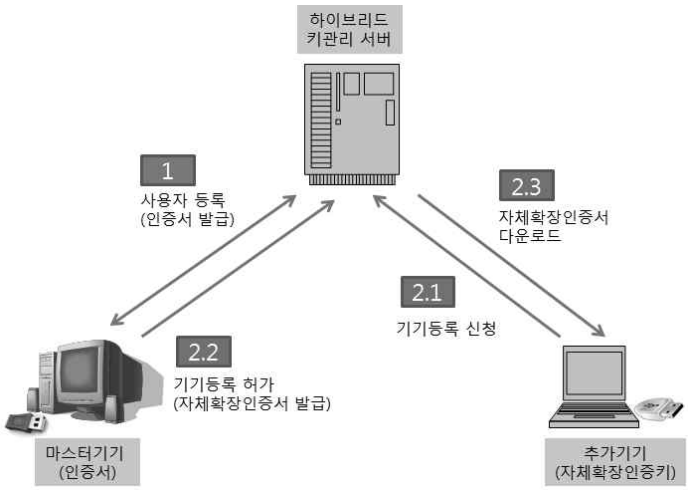
사용자의 추가기기가 하이브리드 키관리 서비스를 통해 사용자의 마스터기기로부터 자체확장인증서를 발급받으면 자체확장인증키를 장착하게 된다. 사용자는 자체확장인증키를 이용하여 타인의 기기와 상호인증 환경에서 보안통신을 이용할 수 있게 된다. 예를 들면 서버에 로그인하는 경우

자체확장인증키를 이용하여 전자서명 로그인할 수 있으며, 서버는 자체확장인증키에 연결된 인증서를 통해 사용자의 신분을 확인하고 자체확장인증서를 검증하여 기기의 유효성을 확인하고 로그인을 허용할 수 있다.

4. 하이브리드 키관리서버 구현

4.1 설계

위에서 제시된 하이브리드 키관리 서비스의 실세계 활용성을 분석해보기 위하여 웹서버 형식으로 키관리서버를 구현하였다. 구현된 하이브리드 키관리 서비스의 전체적인 동작은 다음의 [그림 1]과 같다. 여기에서는 키관리서버가 인증기관의 역할을 겸하도록 하여 사용자 등록시 인증서를 발급하고 서버에도 저장하도록 하였다. 사용자가 별도의 인증기관에서 발급받은 인증서를 등록하여 사용하도록 구현할 수도 있을 것이다.



[그림 1] 하이브리드 키관리 서비스의 동작 구성도

[Fig. 1] Protocol flow of the hybrid key management service

4.2 구현 환경

키관리서버는 리눅스 운영체제의 서버에 node.js[9] 기반의 웹서버 형식으로 구현하였으며 서버와 사용자기기 사이의 자체확장인증서 발급절차를 구현하기 위하여 웹소켓[10] 기술을 이용하였다. 사용자의 마스터기기, 추가기기는 PC, 스마트폰, 태블릿 등이 될 수 있는데 다양한 사용자 기기에서 똑같은 환경으로 사용할 수 있도록 하기 위하여 웹페이지는 jQuery-mobile[11]을 이용하여 모바일 페이지 형식으로 작성하였다. 자바스크립트 환경에서 키생성, 전자서명, 인증서발급 등의 암호

기능을 구현하기 위하여 forge[12] 패키지를 이용하였다.

사용자 기기에서의 안전한 키관리를 위해서는 하드웨어보안모듈을 활용할 수 있도록 해야 하지만 이것을 이용할 수 있는 환경이 아직 미비하여 우선은 개인키와 인증서를 하드디스크 상에 파일로 저장하도록 구현하였다. 향후 하드웨어보안모듈을 활용할 수 있는 환경이 준비되면 사용자 인터페이스가 좀 더 편리하고 안전한 키관리가 가능한 시스템으로 구현할 수 있을 것이다.

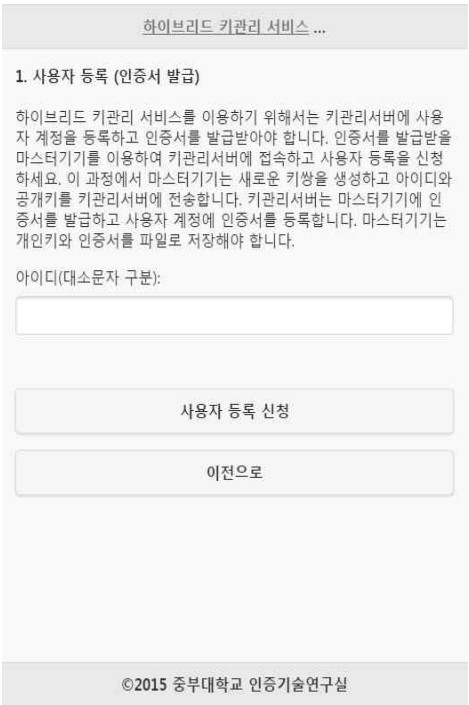
4.3 구현된 하이브리드 키관리 서비스 소개

4.3.1 서비스 초기 화면

본 서비스의 전체 메뉴는 다음의 [그림 2]와 같다. 서비스에 대한 안내를 위해 하이브리드 키관리 서비스에 대해 소개, 서비스 이용 방법에 대해 소개하는 페이지가 준비되어 있다.

[그림 2] 서비스 초기화면의 전체 서비스 메뉴

[Fig. 2] Service menu of the main page



[그림 3] 사용자 등록 및 인증서 발급

[Fig. 3] User registration and issuing certificate

1) 사용자 등록 페이지에서는 아이디를 선택하여 등록하게 되는데 이때 키관리서버로부터 인증서를 발급받게 된다.

2) 사용자 기기등록 페이지에서는 사용자 마스터기기가 사용자 추가기기에 자체확장인증서를 발급하는 기능을 제공한다.
3) 인증서/자체확장인증서로 로그인 페이지는 자체확장인증키의 활용 사례를 제시하기 위한 것으로 인증서 또는 자체확장인증서로 키관리서버에 전자서명 로그인을 할 수 있다.
4) 사용자 기기등록 현황 검색 페이지에서는 사용자가 자체확장인증서를 발급받은 내역을 검색할 수 있다.
5) 전체 사용자 현황 검색 페이지에서는 현재 하이브리드 키관리 서비스를 이용하고 있는 등록된 사용자의 목록을 검색할 수 있다.

4.3.2 사용자 등록 및 인증서 발급 (마스터기기)

사용자는 마스터기기로 사용할 기기에서 사용자 등록 페이지를 접속하여 아이디를 선택하고 사용자 등록을 신청하게 된다. 이때 마스터기기에서는 새로운 키쌍을 생성하고 <아이디, 공개키> 정보를 키관리서버로 전송하고 그 공개키에 대하여 키관리서버로부터 인증서를 발급받게 된다. 사용자는 개인키와 인증서를 다운로드하여 파일로 저장하여야 한다. 키관리서버는 사용자의 계정정보에 생성된 인증서를 저장한다. 이때 개인키는 키관리서버로 전송되지 않는다.

4.3.3 사용자 기기 등록 및 자체확장인증서 발급

사용자는 마스터기기 이외에 다수의 추가기기를 등록하여, 즉 자체확장인증서를 발급하여, 사용하려고 한다. 사용자의 추가기기 등록 과정은 [그림 1] 및 [그림 4]에 나타낸 바와 같이 마스터기기, 추가기기, 키관리서버 사이의 다음과 같은 3단계에 걸친 프로토콜로 수행된다.

1) 사용자 기기등록 신청: 사용자는 추가기기를 이용하여 사용자 기기등록 신청을 하게 되는데 이때 사용자의 아이디와 기기명을 입력하고 기기등록 신청을 한다. 기기등록을 신청하면 추가기기에서는 새로운 키쌍을 생성하고 <아이디, 기기명, 공개키> 정보를 키관리서버에 전달한다. 사용자 추가기기의 화면에는 등록신청 해쉬값을 SHA-256 알고리즘으로 계산하여 표시하는데 이것은 사용자가 다음단계에서 기기등록 허가시 키관리서버로부터의 전송값이 변조되지 않았음을 확인하기 위한 것이다. 사용자는 생성된 개인키를 다운로드하여 파일로 저장해야 한다. 키관리서버는 기기등록 신청 내역을 사용자 계정에 저장한다.
2) 사용자 기기등록 허가: 사용자는 마스터기기를 이용하여 이 페이지에 접속하여 사용자명과 기기명을 입력하고 기기등록 신청 내역을 검색한다. 검색된 신청내역의 해쉬값이 위에서 자신이 신청시 생성된 값과 같으면 변조되지 않은 것으로 신뢰할 수 있다. 이후 인증개인키를 선택하여 읽어오고 등록하기 버튼을 누르면 마스터기기는 자체확장인증서를 생성하게 되고 이것을 키관리서버에 전송하게 된다. 키관리서버는 발급된 자체확장인증서를 사용자 계정에 저장한다.
3) 사용자 기기등록 완료: 사용자는 기기등록 신청을 했던 추가기기를 이용하여 위 서비스에 접

속하고 아이디와 기기명을 입력하고 기기등록 허가 내역을 검색하며 허가된 내용이 있으면 자체확장인증서를 다운로드하여 파일로 저장한다.

[그림 4] 사용자 기기 등록 - 자체확장인증서 발행

[Fig. 4] Registration of user device – issuing self-extended certificate

4.3.4. 인증키/자체확장인증키로 로그인

이제 사용자의 추가기기는 자체확장인증서를 발급받았으므로 자체확장인증키를 이용하여 여러 가지 보안통신에 이용할 수 있다. 가장 대표적인 사례로서 자체확장인증키로 전자서명 로그인을 수행하는 사례를 제시하였다. 사용자는 추가기기를 이용하여 서버의 로그인 페이지에 접속하고 로그인에 사용할 자체확장인증서와 자체확장인증개인키를 선택한 후 로그인 버튼을 누르면 사용자기기는 로그인 요청 메시지를 자체확장인증개인키로 전자서명하고 <인증서, 자체확장인증서, 전자서명>을 서버로 전송하여 로그인 요청을 한다. 서버는 다음의 3가지 검증과정을 확인한 후에 로그인을 허용한다.

1) 사용자의 인증서를 검증하여 사용자 신분을 확인한다.
2) 자체확장인증서를 인증서의 공개키로 검증하여 서명에 사용된 자체확장인증서가 유효한지 검증한다.
3) 로그인 요청 메시지의 전자서명의 유효성을 자체확장인증서의 공개키를 이용하여 검증한다.

[그림 5] 전자서명 로그인
[Fig. 5] Login using digital signature

[그림 6] 사용자 기기등록 현황 검색
[Fig. 6] List of registered user devices

4.3.5 사용자 기기등록 현황 검색

사용자가 자체확장인증서를 발급받은 내역을 검색할 수 있다. [그림 6]에서는 아이디가 abc인 사용자가 9개의 추가기기에서 자체확장인증서를 9개 발급받았다는 것을 보여준다.

4.3.6 전체 사용자 현황 검색

하이브리드 키관리를 이용하고 있는 전체 사용자 목록을 검색할 수 있다.

5. 하이브리드 키관리 서비스의 활용성 분석

5.1 서비스의 특징

개선된 하이브리드 키관리 서비스는 다음과 같은 특징을 가지고 있다.

1) 키관리의 편의성: 사용자가 몇 개의 기기를 사용하든 상관없이 외부기관에 의존하지 않고 자신의 소유기기에 자체확장인증서를 직접 배포할 수 있어서 키관리의 편의성이 크게 향상된다.
2) 안전한 키관리: 사용자의 인증서(인증개인키) 및 자체확장인증서(자체확장인증개인키)는 하드웨어보안모듈 내부에서 생성되고 사용되며 외부로 꺼낼 수 없으므로 공격자에게 누출되지 않고 안전하게 이용할 수 있다. 즉, 하이브리드 키관리는 사용자 기기의 하드웨어보안모듈에서 생성되는 안전한 키쌍을 사용자의 인증키로 사용할 수 있게 해준다.
3) 대외적 신분확인: 자체확장인증서는 개인이 인증서를 이용하여 생성한 것이므로 검증시 인증서와 함께 사용되며 기기별로 다른 자체확장인증키를 사용하더라도 타인이 사용자의 신분을 확인할 수 있어서 부인방지 기능을 제공할 수 있다.
4) 인증서의 안전한 관리: 사용자의 인증서는 자체확장인증서 발급에만 사용하고 일상적인 통신에서는 사용할 필요가 없게 되므로 마스터기기는 꺼놓을 수 있고 인증서와 인증개인키를 더 안전하게 관리할 수 있다.
5) 키관리서버 운영의 경제성: 개선된 하이브리드 키관리 서비스에서는 하나의 키관리서버를 다수의 사용자가 함께 이용할 수 있으므로 보안 전문가가 운영하여 안전한 서비스를 제공할 수 있으며 개인이 키관리서버를 운영하는 것과 비교할 때 경제적인 이득이 크다.
6) 표준 활용: 개선된 하이브리드 키관리 서비스에서는 자체확장인증을 위하여 표준화된 인증서 양식을 사용하므로 구현 편의성이 높다. 자체확장인증서에서는 인증서의 모든 필드가 필요한 것이 아니므로 [2]에서와 같이 축약된 형식의 별도 포맷을 사용할 수도 있다.

5.2 활용성

안전하고 편리한 인증키 관리 기술은 모든 보안의 첫걸음이라고 할 수 있다. 자체확장인증서를 이용하는 하이브리드 키관리 기술은 복수의 컴퓨팅 기기를 사용하게 되는 유비쿼터스 환경에서의 인증키 관리를 편리하게 수행할 수 있는 핵심 기반기술이 된다. 이것은 유무선의 전통적인 컴퓨팅 환경 뿐만 아니라 모바일, 사물인터넷, 클라우드 등 새로운 컴퓨팅 환경에서의 키관리를 위해서도 편리하게 이용될 수 있다. 여기에서는 하이브리드 키관리 기술의 대표적인 활용방안을 제시하고자 한다.

1) 한국의 공인인증체계는 PKI 도입의 세계적인 선구자 역할을 했지만 구현기술의 측면에서 개인키를 하드디스크의 특정 폴더에 암호화하여 저장하는 방식을 사용함으로써 개인키를 안전하게 관리할 수 없다는 점과 웹브라우저에서의 사용을 위해 특정 브라우저에 종속된 비표준 기술인 액티브엑스를 사용한다는 점 등의 비판을 받아왔다. 이제 하드웨어보안모듈을 사용하고 개인의 생체인식기술을 이용하여 패스워드 의존성을 없앨 수 있는 FIDO(Fast IDentity Online)[13] 기술을 공인인증과 결합하여 사용하는 방법이 발전하고 있는데 이것은 키관리를 안전하게 하고 사용자 편의성을 크게 높일 수 있을 것으로 기대되고 있다. 그러나 사용자가 복수기기를 사용하는 환경에서의 키관리의 복잡성에 대한 해결책은 여전히 필요한 상황이다. 이를 해결하기 위해 하이브리드 키관리 서비스를 FIDO 기술과 연동하여 활용할 수 있을 것이다.
2) 사물인터넷 환경에서는 인간의 개입이 최소화되는 환경에서 센서기기, 게이트웨이, 클라우드 서비스 플랫폼, 사용자 단말기기 등 많은 기기들 사이에 상호인증이 필요하다. 사물인터넷 환경에서 하이브리드 키관리 방식을 적용하면 사용자가 자신이 운영하는 기기들의 멤버십을 자신의 인증서에 기반하여 체계적으로 관리할 수 있게 된다. 아울러 타인의 센서기기, 단말기 등과도 인증서 기반의 명확한 접근제어, 권한제어 기능을 제공할 수 있을 것이다.
3) 현재까지 컴퓨터의 운영체제에서는 인증키 관리 기능이 매우 부족한 상황이다. 자체확장인증서를 이용하는 하이브리드 키관리 기능을 안전하게 활용하기 위해서는 컴퓨터의 운영체제에 내장하는 것이 필요하다. 우리가 사용하는 컴퓨터, 스마트폰, 웨어러블, 사물인터넷 게이트웨이 등의 다양한 운영체제에 자체확장인증서 발급 및 이용을 위한 하이브리드 키관리 기능이 내장된다면 추가 프로그램의 설치 등의 번거로움이 없이 안전하고 편리하게 인증키 관리 기능을 이용할 수 있게 될 것이다.

5. 결론 및 향후과제

이 논문에서는 [2]에서 제시된 하이브리드 키관리 방식을 개선하여 하나의 키관리서버를 다수의 사용자가 함께 사용할 수 있도록 키관리 프로토콜을 재설계 하였다. 아울러 웹서비스 형태로 키관

리서버를 구현하여 실세계에서의 활용성을 검토하였다. 이러한 방식을 이용하면 비전문가인 개인 사용자마다 키관리서버를 운영할 필요가 없이 전문가가 운영하는 하나의 키관리서버를 많은 사용자들이 함께 이용함으로써 효율적이고 안전한 서비스 운영이 가능하다.

현재까지는 하드웨어보안모듈을 활용하는 기반이 부족하여 인증서와 개인키를 파일로 하드디스크에 저장하도록 구현하여 사용자 편의성과 안전성이 부족하지만 향후 하드웨어보안모듈을 활용하는 방식으로 구현한다면 편의성과 안전성이 크게 향상될 것으로 생각된다. 아울러 사용자의 패스워드 의존성을 줄이기 위해 생체인식을 활용하는 FIDO(Fast IDentity Online)[13] 기술의 적용이 활성화되고 있는데 이 기술과의 접목 시도가 필요하다고 생각된다.

사물인터넷 시대에는 사용자가 더 많은 기기들을 소유하고 운영하게 되는데 이들에 대한 체계적인 인증키 관리가 필요하다. 사물인터넷이 개방된 환경에서 다른 사용자 및 그들의 소유기기들과 통신하며 개방형 서비스를 제공하게 되면 사용자 인증, 접근제어, 권한관리 등을 위해 전자서명 기반의 분명한 인증이 필요할 것인데 하이브리드 키관리 방식은 이것을 효율적으로 수행할 수 있는 근본 대책이 될 수 있다.

References

[1] B. Lee, Ubiquitous Key Management in Computers Equipped with Hardware Security Module, Proceedings of the 2013 Conference on Information Security and Cryptography, (2013) December 6, pp. 12-15, Seoul, Korea.

[2] B. Lee, Hybrid Key Management Using Self-Extended Certification and Hardware Security Module, Journal of Security Engineering. (2014), Vol.11, No.4, pp. 273-286.

[3] Internet X.509 Public Key Infrastructure Certificate and CRL Profile, http://www.ietf.org/rfc/rfc2459.txt, Jan. 15 (2016)

[4] http://www.trustedcomputinggroup.org/, Jan. 15 (2016)

[5] J. S. Park, T. N. Cho, J. H. Han, and S. I. Jun, Trusted Computing Technology and TCG Standard Trend, Electronics and Telecommunications Trends, (2008), Vol. 23 No. 4, pp. 48-59.

[6] Siani Pearson, et al., Editor, Trusted Computing Platforms, Prentice Hall PTR (2003)

[7] H. J. Kim, T. K. Kwon, NFC Technology Trends and Security Issues, Information and Communications Magazine (Information and Communications). (2012), Vol. 29 No. 8, pp. 57-64.

[8] http://www.nfc-forum.org/, Jan. 15 (2016)

[9] node.js https://nodejs.org/en/, Jan. 15 (2016)

[10] socket.io, http://socket.io/, Jan. 15 (2016)

[11] jQuery mobile, https://jquerymobile.com/, Jan. 15 (2016)

[12] Forge, https://github.com/digitalbazaar/forge, Jan. 15 (2016)

[13] FIDO Alliance, https://fidoalliance.org/, Jan. 15 (2016)

