동적 암호키 생성 및 복구 방법

신영선° 오송석, 김황래*, 박진섭
대전대학교 컴퓨터공학과, 공주대학교 컴퓨터공학부*
ysshin@zeus.dju.ac.kr°, songseuko@hanmail.net, hrkim@kongju.ac.kr*, jspark@dju.ac.kr

Dynamic Encryption Key generation and recovery Method

Youngsun Shin° SongSeuk Oh, Hwang Rae Kim*, JinSub Park
Dept. of Computer Engineering, Daejeon Univ., Gongju Univ.*

요 약

유비쿼터스 환경이 도래하면서 언제, 어느 곳에서든 네트웍을 사용하여 정당한 사용자임을 확인하고자 하는 요구가 증대하고 있다. 기존의 사용자 인증 방식은 인증기관으로부터 받은 인증서와 비밀키를 하드웨어 장치나 스마트카드 등의 보조 기억장치에 저장하여 휴대하고 다녀야 하는 번거로움이 있다. 또한 사용자의 비밀키를 키위탁 센터에 위탁하여 필요할 때 사용할 수 있도록 하고 있으나 여러 가지 위협으로부터 안전할 수 없는 상황이다.

본 논문에서는 이러한 휴대의 불편함과 여러 가지 위협으로부터 보호하고, 사용자가 언제 어디서든 인증을 제공받을 수 있는 동적 키생성 및 키복구 모듈을 제안한다.

1. 서 론

정보통신망을 통한 서비스 제공이 확대되고, 그 이용이 증대됨에 따라 언제, 어디서든 네트워크에 대한 접근 용이성이 높아지는 유비쿼터스 환경이 도래하고 있다.

이러한 환경에 맞추어 인터넷과 같이 공개된 네트워크에서의 안전한 통신을 위하여 공개키기반(PKI) 시스템을 이용한 사용자 인증 및 전자서명을 제공하고 있다.

이러한 방법은 공개키를 이용하여 데이터를 암복호 하고, 디지털 인증서를 통한 사용자 인증을 수행하는 시스템을 이용하는 것이다.

기존의 사용자 인증 방법은 공인인증기관으로부터 인증서를 발급받고 사용자의 개인키를 사용하여 인증을 수행하였다. 이 방법은 개방된 네트워크 환경에서 개인 인증 정보의 통신상의 노출 위험과, 사용자가 인증서와 개인키를 자신의 하드웨어에 보관하거나, 휴대용 장치(스마트카드, USB)에 휴대하고 다녀야 하는 불편함이 있었다.

사용자의 이동이 많아지고, 하드웨어 플랫폼에 상관없이 전자상거래를 하고자 하는 요구가 증대됨에 따라, 인증서와 개인키를 휴대해야 하는 불편함은 유/무선 환경을 지원하는 환경에서의 큰 제약이라고 할 수 있다.

차선책으로 휴대에 따른 불편함을 줄이기 위해 키위탁 및 키관리 시스템이 개발되어 사용되고 있다. 키위탁 방식은 복구될 사용자의 비밀키에 관련된 정보를 하나 이상의 신뢰되는 위탁기관에 위탁하고, 정당한 키 복구 요청에 대해 위탁기관이 보관하고 있는 키 정보들로 부터 키 또는 암호문의 평문을 얻어내는 방식이다. 하지만 기본적으로 사용자의 키를 안전한 곳에 보관해야만 하는데 이러한 세션키를 모두 저장하기 위해서는 막대한 저장소가 필요하며, 키를 위탁하는 과정에서 상당한 오버헤드가 발생하는 등 여러 가지 문제들이 발생하게 된다. 또한 키위탁 및 키관리 시스템에 대한 위협 발생 시 사용자 정보를 누출의 위험이 발생 할 수 있는 취약점이 있다.

본 논문에서는 이러한 점들을 고려하여 휴대의 불편함과 키위탁의 안전성에 관한 문제를 해결하고, 다양한 환경에서 유연하면서도 효율적인 키생성 및 키복구 모듈을 제안한다.

본 논문은 다음과 같이 구성된다. 우선 2장에서는 기존의 사용자 인증을 위한 키생성 및 복구 방법에 대해 알아보고, 3장에서는 다양한 환경에서의 키생성 및 키복구 방법을 제안한다. 마지막으로 4장에서는 결론 및 향후 연구를 제시한다.

2. 관련연구

본 장에서는 사용자의 인증을 위한 키생성 및 키복구 방식에 대해 분석한다. 공개키기반 구조에서의 키생성 및 키복구 방식은 암호화 알고리즘이나 일방향 함수를 이용하여 키를 생성하고 비밀키를 휴대하거나 위탁하는 형태였다. 최근에는 키위탁 센터의 위험성으로부터 좀더 안전하게 키를 생성하고 복구하고자 여러개의 분산 서버에 키를 저장하여 복구할 수 있는 시스템이 연구되어지고 있다.

본 절에서는 키생성 및 키복구 방법의 하나로 분산서버에 키를 나누어 저장하고 복구할 수 있는 시스템에 대해 설명하고 문제점을 분석한다.

2.1 기존의 키생성 및 키복구 방식

사용자 인증을 위한 키생성 기술은 사용자의 패스워드를 이용하여 원격지에 사전에 저장된 비밀키를 다운로드 하여 사용하는 방법이 제안되었다. 하지만 이러한 방법은 기밀정보를 저장하는 중요한 수단으로 사용되지만 실제적인 환경에 적용하기 어렵고 사용자 패스워드는 상대적으로 작은 공간에서 선택되고 쉽게 기억될 수 있다는 문제점으로 인한 공격이 발생할 수 있다.

또한 종래 기술은 n개의 서버를 이용한 로밍 프로토콜로서 사용자는 n개의 서버로부터 비밀값을 정상적으로 얻어야만 사용자가 원하는 비밀키를 도출하였다. 이때 단 하나의 서버라도 공격자로부터 손상된 경우 정상적인

비밀키 도출이 어렵다는 단점이 있다.

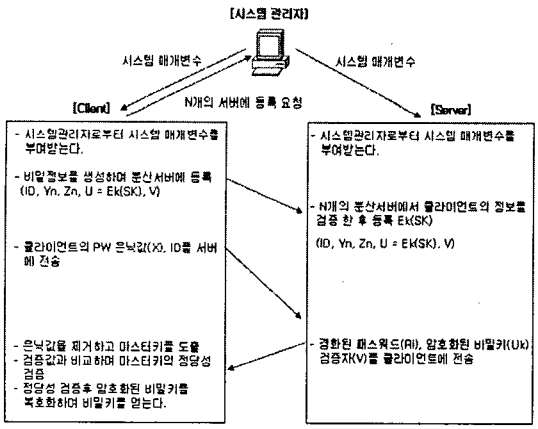
이러한 문제점을 보완하여 (그림1)과 같이 키생성 및 키관리 기술의 구조로 다수개(n개)의 인증서버를 두어 사용자의 비밀정보를 분산하여 저장한 후 정상적인 서버(K개)들만을 이용하여 비밀키를 복원할 수 있는 기술이 연구되었다.

그림에서 보는바와 같이 시스템관리자는 시스템 매개변수를 생성하여 클라이언트와 서버에게 동일하게 전달해야 한다.

클라이언트는 등록을 위해 ID, PW, 비밀정보를 생성하여 분산서버에 키생성 시스템에 의해 생성된 비밀키 SK를 등록한다. 이때의 클라이언트의 비밀정보는 ID, PW, K(사용자 키), V(검증자)를 말한다.

분산서버는 시스템관리자로부터 받은 매개변수를 사용하여 클라이언트 정보를 검증하여 정당한 사용자인지를 판단하고 분산서버에 키를 등록 하게 된다.

비밀키 복구시 클라이언트로부터 은닉된 패스워드와 사용자 아이디를 n개의 서버보다 적은 개수의 k개의 서버에 송신한다. 클라이언트는 서버에 등록되어 있는 암호화된 비밀키와 검증자(V), 패스워드를 서버로부터 수신하고 검증자를 비교하여 일치하는 경우 암호화된 비밀키를 복호화 하여 비밀키를 얻는다.



(그림 1) 비밀키 관리기능 및 비밀키 단말장치기술

이러한 새로운 개념으로 키를 생성함으로써 서버 손상시에도 비밀키를 복원할 수 있으나 저장을 해야 한다는 측면에서 볼 때 여러개의 서버가 위협에 노출되어 있을 경우 저장된 키에 대한 정보가 유출될 위험이 있다. 따라서 좀 더 안전한 환경에서의 키보관 방법이 연구되어져야 한다.

3. 키생성 및 키복구 시스템의 제안

본 장에서는 기존의 사용자 인증 방식의 문제점을 보완하고 사용자의 이동성을 고려하여 자신의 비밀키를 휴대하거나 위탁 저장하지 않고, 언제 어디서나 실시간으로 키를 생성하고 복구할 수 있는 기술을 설계한다.

본 논문에서는 키생성 모듈과 키복구 모듈로 나누어 설명한다.

이 시스템은 랜덤하게 생성된 질문에 대한 사용자의 답변을 SHA(secure hash algorithm)를 이용하여 해쉬값을 생성한다. 생성된 해쉬값들을 JOIN연산을 수행하는데 이때 생성되는 값들의 크기가 크기 때문에 일반적인 프로그램에서의 데이터형으로 정의할 수 없다. 따라서 큰 수 계산을 위해 오픈된 GNU mp 라이브러리를 사용하여 큰 수 연산을 수행하여 Key를 생성하고 복구한다. 키복구를 위해 난수와 소수를 생성하고 다항식을 이용하여 S값을 생성한다. 이와같이 생성된 키에 대한 정보는 저장되지 않아 기존의 키위탁 시스템의 취약점을 보안하였다.

3.1 사용된 기호 정의

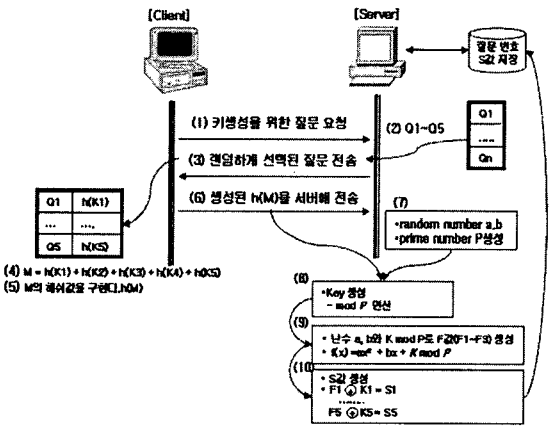
사용되는 용어와 다음과 같이 정의된다.

- Q1 ~ Q5 : 서버에서 랜덤하게 선택된 질문
- K1 ~ K5 : 질문에 대한 클라이언트의 답변
- h(K1) ~ h(K5) : 질문에 대한 해쉬 값
- M : 해쉬값을 JOIN연산한 결과
- h(M) : M의 해쉬값
- K : h(M)을 큰 수 연산을 수행한 후의 생성된 키
- a, b : 랜덤하게 생성한 값
- P : prime number
- S1 ~S5 : 다항식과 K1~K5값으로 생성된 S값

3.2 키 생성 모듈

사용자 인증을 위한 키생성 모듈은 서버가 랜덤하게 질문을 생성하고 클라이언트가 질문에 대한 답변을 함으로써 키를 생성하게 된다.

다음은 키생성 모듈에 대한 구조이다.



(그림 2) 키생성 모듈

(1) client는 server에게 키생성을 위한 질문을 요청한다.
(2) server는 client의 질문 요청을 수락하고 서버에 저장된 질문 중 랜덤하게 5개의 질문(K1, K2, K3, K4, K5)을 선택한다.(이때, 질문번호를 저장한다.)
(3) 선택된 5개의 질문을 client에게 전송한다.
(4) server로부터 받은 질문에 대해 각각 SHA를 사용하여 해쉬값을 구하여 JOIN연산을 수행한다.

h(K1), h(K2), h(K3), h(K4), h(K5)

M=h(K1)+h(K2)+h(K3)+h(K4)+h(K5)

(5) JOIN된 M에 대해 또 한번의 해쉬값을 구하고 큰수 연산을 하여 Key를 생성한다.

(6) 생성된 Key를 서버에 전송한다.

(7) 서버는 다항식에 적용할 prime number P, random number a,b를 구한다.

(8) 생성한 K값과 P에 대해 $K \bmod P$ 연산을 수행한다.

(9) 다항식에서 난수 a, b와, $K \bmod P$ 값을 사용하여, F값(F1 ~ F5)을 구한다.

$$f(x) = ax^2 + bx + K \bmod P$$

(10) F값과 각 답변에 대한 해쉬값(K1~K5)을 XOR연산을 하여 키복구시 사용될 S값을 구한다.

$$f(0x11) \oplus K1 = S1$$

.....

$$f(0x99) \oplus K5 = S5$$

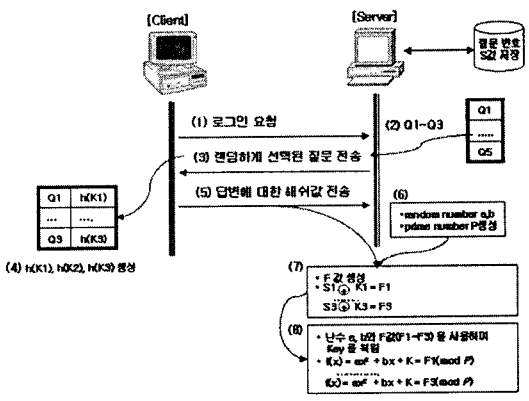
S값을 생성한 후 각각의 답변에 대한 해쉬값과 random number a,b를 삭제하고, S값을 데이터베이스에 저장한다.

이와 같이 본 제안 방법에서는 키생성시 필요한 어떠한 정보도 저장되지 않는 것을 볼 수 있다. 단지 다항식과 사용자의 답변 정보로 생성된 5개의 S값만을 저장하여 키복구시 5개의 S값 중 3개의 S값으로 키복구를 연산을 수행한다.

3.3 키 복구 모듈

인증된 사용자가 로그인시 키 복구 모듈은 서버가 사용자 인증시 사용했던 질문 중 3개의 질문을 랜덤하게 전송하여 클라인어트의 답변을 함으로써 키를 복구하게 된다.

다음은 키 복구 모듈에 대한 구조이다.



(그림 3) 키복구 모듈

이 키복구 과정은 키생성 과정에서의 K값, 난수 a,b, 소수 P, $K \bmod P$, F값을 구하는 과정과 동일하다.

(1) client는 server에게 로그인 요청을 수행한다.

(2) server는 5개의 질문 중 랜덤하게 3개를 선택한다.

(3) 랜덤하게 선택된 질문을 client에게 전송한다.

(4) 서버로부터 전송받은 질문에 대해 각각 해쉬값을 구하여 JOIN연산을 수행한다.

h(K1), h(K2), h(K3)

(5) 답변에 대한 해쉬값 서버로 전송

(6) server는 다항식에 적용할 prime number P, random number a,b를 구한다.

(7) 데이터베이스에 저장되어 있는 5개의 S값 중 3개를 임의로 선택한 값과, 세 개의 답변에 대한 해쉬값(K1~K3)에 대해 XOR연산을 수행하여 F값(F1~F3)을 구한다.

$$S1 \oplus K1 = F1$$
$$S2 \oplus K2 = F2$$
$$S3 \oplus K3 = F3$$

(8) 라그랑지 interpolation알고리즘 사용하여 Key값을 복원한다.

af(0x11)^2 + b(0x11) + K = F1(mod P)

.....

af(0x33)^2 + b(0x33) + K = F3(mod P)

키복구시에도 K1에서 K3까지 값과 F값, 랜덤 수 a, b의 값을 어느 곳에도 저장하지 않는 것을 볼 수 있다.

4. 결론

환경적인 특성의 변화로 기존의 키생성 및 키복구 방법은 이동성과 편리성을 강조하는 환경에서는 적용하기가 어렵다. 또한 키를 저장하여 필요할 때마다 사용한다는 것 역시 여러 가지 위협 발생으로부터 안전하지 못하다. 따라서 이동성과 편리성, 안전성을 고려한 키생성 및 키복구 기술이 필요하다. 본 논문에서 제안한 키생성 및 키복구 기술은 사용자의 휴대의 불편함과 키저장의 안전성을 고려하여 랜덤하게 생성된 N개의 질문에 대해 K개의 사용자 답변에 대해 해쉬값을 구하고 큰수 연산을 통하여 키를 생성하였다. 또한 키 복구시 어느 곳에도 저장되어 있지 않은 키정보에 대해 N개보다 적은 질문에 대해 새로이 해쉬값과 큰수 연산을 통해 키를 복구할 수 있는 방법을 제시하였다. 또한 키정보에 대해 어느 곳에도 저장하지 않고 사용자의 답변만으로 키를 생성하고 복구 할 수 있어 저장에 따른 안전성 문제에도 효율적이다.

[참고문헌]

[1] Van Dijk, Marten Erik, Secret key sharing and secret key generation, ACM, DAI-C 59/03, p. 695, Fall 1998

[2] Frye.E, Key Recovery in a Public Key Infastrcuter,, Jurimetric, Vol.33 No.3, p.485-495, 1998

[3] Cho,T, Lee.S-H, A Key Recovery Mechanicaim for Reliable Group Management, ACM, Vol.2864 No.7, p372-286, 2003

[4] 이덕규, 이임영,브로드캐스트 암호화에서의 효율적인 키생성과 갱신방법, 정보처리학회논문지, 제11-C권 제2호, 2004

[5] 유준석 외3명, 일방향 키 분배 기능을 가지는 유연한 키 복구 시스템, 정보처리학회논문지, 제8-C권 제3호, 2001.6