PKI에서 안전한 비밀키 보관 시스템 설계

최희봉*, 유희종**,이훈재***,오수현****,원동호****

*국가보안기술연구소,**전자통신연구원,***경운대,****성균관대

Design of the Secure Deposit System of the Secret Key

Heebong Choi*, Huijong Yu**, Hyunjae Lee***, Suhyun Oh****, Dongho Won*

*NSRI, **ETRI, ***Sung Kyun Kwan Univ. ****Kyungwoon Univ..

요 약

본 논문은 공개키 암호 시스템에서 사용자의 비밀키를 안전하게 보관할 수 있는 시스템을 제안한다. 키 보관 시스템은 사용자가 자신의 비밀키를 저장할 환경을 갖추지 못하였거나 분실하였을 경우 온라인 통신에 의해 안전하게 비밀키를 얻을 수 있는 시스템이다. 키 보관 시스템은 공개키 기반구조(PKI)와 연동하여 운용되는 암호 시스템이다. 이 논문에서 비밀키 보관 시스템을 설계하는 방법으로서 비밀키를 메모리에 저장하여 인증기관을 통하여 인증된 사용자에게 발송하는 방법과 마스터 키 개념을 이용한 공개키 연동 키 복구 시스템을 변경하여 사용하는 방법을 제안하고 이들 방법을 안전성과 효율적인 측면에서 비교 분석한다.

1. 서 론

현대 사회가 점차 고도의 정보화 사회로 발전해 가면서 다양한 정보의 개방과 공유, 네트워크를 통한 업무처리의 일반화는 무한한 가능성과 편리함을 주었지만 정보의 침해라는 문제를 발생시켰다. 이로 인하여 정보 보호 문제가 부각되었다.

정보 보호 시스템에서 매우 중요한 사용자의 비밀키를 관리할 때 발생되는 문제점은 다음과 같다. 첫째 사용자 암호 시스템의 안정성이 불안하여 비밀키 분실이나 손상이 발생할 수 있다. 둘째 템퍼 보호 기능을 갖추지 못한 사용자의 암호 시스템은 비밀키 안전성이 취약하므로 비밀키 저장이 어려울 수 있다. 셋째 대량의 문서를 장기적으로 보관하는 경우 비밀키를 변경하면 불편함이 발생할 수 있다.

안전한 비밀키 보관 시스템은 위에서 살펴본 비밀키 관리를 해결할 수 있는 방법이다. 그러나 비밀키 보관 시스템에서 사용자 비밀키 복구에 필요한 절차가 오프 라인으로 수행될 경우 걸리는 시간과 절차의 복잡성 등으로 매우 불편하다. 따라서 온라인으로 사용자 비밀카 복구가 수행될 수 있도록 비밀키 보관 시스템을 구성하는 것이 필요하다.

본 논문에서 비밀키 보관 시스템을 온라인으로 수행하기 위하여 PKI와 연동시켰다. 즉 공개키 기반 구조를 연동시킴으로써 비밀키 복구를 요청하는 사용자를 인증하기 위해 인증기관을 이용할 수 있는 것이다.

비밀키 보관 시스템으로서 가장 기본적인 것은 비밀키 보관 기관이 자신의 메모리에 각 사용자의 비밀키를 보관하는 것이다. 이것은 사용자의 프라이버시 침해 및 메모리 관리 등 문제점이 있다. 둘째 키 보관 기관의 마스터 키 개념을 도입하여 비밀키 보관 시스템을 설계할 수 있다. 1999년 P. Paillier와 M. Yung은 키 복구 시스템 SE-PKI(Self-Escrowed Public Key Infrastructure)를 제안하였다[2]. 이 키 복구 시스템을 변경하면 마스터 키 개념을 갖는 비밀키 보관 시스템이 될 수 있다. 그러나 위의 비밀키 보관 시스템은 사용자의 프라이버시 문제를 해결 할 수 없다. 따라서 사용자의 프라이버시 문제를 해결한 비밀키 보관 시스템은 2001년 유희종, 최희봉, 오수현, 원동호가 제안한 공개키 기반 구조와 연동 가능한 키 복구 시스템을 변경하여 설계될 수 있다[8]. 이것은 사용자의 비밀키를 사용자만이 신뢰기관의 도움을 받아 키 보관 기관으로부터 온 라인 상에서 안전하게 복구할 수 있는 비밀키 보관 시스템이다. 비밀키 보관 기관을 다수로 참여시킨 시스템에 암호 프로토콜을 설계함으로써 모든 키 보관 기관 중 한 개 이상이 공모하지 않는다는 조건에서 사용자는 자신의 비밀키가 사용자 외에 다른 기관에 누출됨이 없이 사용자의 키를 복구할 수 있다.

이 키 보관 시스템은 분실한 사용자의 비밀키를 복구할

뿐만 아니라 템퍼 보호가 불가능한 시스템에서나 비밀키 생성기능은 가지고 있으나 저장 기능이 없는 시스템에서 편리하게 사용할 수 있다. 또한 사용자가 비밀키를 자주 변경하더라도 비밀키 관리가 용이하다. 예를 들면 장기로 보관하는 대량의 문서를 암호화한 경우 비밀키를 변경할 때 암호화된 문서를 다시 복호화하여 새로운 공개키로 암호화할 필요 없이 처음에 암호화한 공개키를 함께 보관하면 되기 때문에 키 관리가 편리하다.

본 논문의 구성은 다음과 같다. 2장에서는 모든 사용자의 비밀키를 저장할 수 있는 메모리가 필요한 비밀키 보관 시스템을 설명하고 3장에서 마스터 키 개념을 갖는 비밀키 보관 시스템을 설명하며, 4장에서는 비밀키 보관 시스템에 사용자 프라이버시 보호 가능한 비밀키 보관 시스템을 설명한다. 5장 비밀키 보관 시스템들을 비교하고 6장은 결론으로 구성되어 있다.

2. 비밀키를 저장하는 키 보관 시스템

모든 사용자의 비밀키를 저장하고 있는 비밀키 보관 시스템을 설명한다. 여기서 키 보관 시스템은 PKI와 연동하도록 설계되어 있으며 사용자의 비밀키 등록과정이 있다.

1. 키 보관 기관은 공개키/비밀키 쌍 (Y, X)을 생성하여 공개키 Y를 공개한다.
2. 사용자는 자신의 공개키/비밀키 (y, x) 쌍을 생성하여 공개키 인증서를 인증기관에 요청한다.
3. 인증기관은 사용자의 공개키 y를 서명하여 인증서를 발행하고 이것을 공개키 데이터 베이스에 보낸다.
4. 사용자는 생성한 공개키/비밀키 쌍에서 자신의 비밀키 x를 키 보관 기관의 공개키 Y로 암호화한 $E_Y(x)$을 키 보관 기관에 보낸다.
5. 키 보관 기관은 사용자의 비밀키가 암호화된 $E_Y(x)$를 안전한 메모리에 저장한다.

사용자가 자신의 비밀키 x를 필요할 때 수행되는 비밀키 복구 과정은 다음과 같다.

1. 사용자는 키 복구용 공개키/비밀키 쌍 (y_{rec}, x_{rec})을 생성하여 공개키 인증서를 인증기관에 요청한다.
2. 인증기관은 사용자 키 복구용 공개키 y_{rec}에 서명하고 키 복구용 공개키 인증서를 사용자에게 보낸다.
3. 사용자는 키 복구 요청서와 키 복구용 공개키 인증서를 키 보관 기관에게 보낸다.
4. 키 보관 기관은 사용자의 키 복구용 공개키 y_{rec}를 검증하고 메모리에 저장되어 있는 해당 사용자의 비밀키 x를 키 복구용 공개키 y_{rec}로 암호화한 $E_{yrec}(x)$를 사용자에게 보낸다. 이때 키 보관 기관은 메모리에 저장되어 있는 사용자의 비밀키 x는 키 보관 기관의 공개키 Y로 암호화되어 있기 때문에 키 보관 기관의 비밀키 X로 복호화하면 사용자의 비밀키 x를 얻을 수 있다.
5. 사용자는 키 복구용 비밀키 x_{rec}로 키 보관 기관으로부터 얻은 데이터를 복호화한 $D_{xrec}(E_{yrec}(x))=x$를 얻을 수 있다.

비밀키를 저장하는 키 보관 시스템에서는 모든 사용자의 비밀키 x 크기만큼 메모리 자원이 필요하다. 그리고 키 보관 기관은 모든 사용자의 비밀키를 알고 있으므로 이 시스템은 사용자 프라이버시에 대한 문제를 갖고 있다. 사용자 프라이버시 문제를 해결하기 위하여 다수의 키 보관 기관을 두고 사용자의 비밀키를 분산하여 저장할 수 있으나 이러한 시스템을 구성하는데 너무 복잡하고 비효율적이다. 그러나 사용자와 키 보관 기관이 같은 암호 방식을 사용할 수 있다.

3. 마스터 키를 갖는 키 보관 시스템

우리는 2장에서 키 보관 시스템이 사용자의 비밀키를 저장해야 하는 메모리가 필요함을 알 수 있었다. 이러한 불편을 없애기 위하여 마스터 키를 갖는 키 보관 시스템을 도입한다. 사용자가 ElGamal 암호 방식을 사용하고 키 보관 기관이 Paillier 암호 방식과 ElGamal 암호 방식을 사용하는 시스템이다. 이들 키 보관 시스템은 PKI와 연동하여 운용된다. 키 보관 기관은 마스터 공개키/비밀키 쌍을 갖고 있으며 사용자의 공개키/비밀키 쌍을 마스터 공개키를 사용하여 생성한다. 사용자의 비밀키는 사용자의 공개키로부터 마스터 공개키에 의해 복구될 수 있다. 1999년에 제안된 P. Paillier의 SE-PKI(Self-Escrowed Public Key Infrastructure)와 같은 PKI 연동 키 복구 시스템에서는 사용자의 공개키로부터 키 복구 가능한가를 인증기관이 ZKIP로 증명하였지만 키 보관 시스템에서는 이런 절차가 필요하지 않다. 그 이유는 비밀키 복구는 사용자의 필요에 의해서 수행되기 때문이다. 그러나 추가로 비밀키 복구 과정에서 인증기관을 이용하여 사용자를 인증해야 하는 절차가 필요하다.

1999년 P. Paillier가 n이 RSA modulus $n=pq$인 $Z^*_{n^2}$ 상에서 연산되는 공개키 확률 암호 방식을 제안하였다[2]. 이 암호 방식을 결정적 암호 방식으로 변환하여 다음과 같이 설계할 수 있다. 메시지 $m<n$의 암호화는 단순한 연산을 적용하여 $c=g^m \bmod n^2$으로 수행되고, 복호화는 $m=\frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$ 에 의하여 수행된다.

마스터 공개키

$n, g, l = 2|n|$

마스터 비밀키

$\lambda = lcm(p-1, q-1)$

사용자 공개키

$y = g^x \bmod n^2$ where $x <_R n$

사용자 비밀키

$x < n$

암호화

plaintext $m < n^2$

ciphertext $c = (my^k, g^k)$ where $k <_R 2^l$

복호화

ciphertext $c = (a, b)$

plaintext $m = a/b^x \bmod n^2$

키 복구 과정

$$\frac{L(y^\lambda) \bmod n^2}{L(g^\lambda)} \bmod n$$

$= x$

그림 1: Paillier 암호 방식을 이용한 키 복구 과정

Paillier 암호 방식을 이용하여 키 복구 과정을 나타낸 것은 그림 1에서 설명한다.

키 보관 기관은 마스터 공개/비밀키 쌍을 생성하고 있고 키 보관 기관의 공개 파라메터와 인증기관의 파라메터는 배포되어 있다고 가정한다. 사용자의 공개키/비밀키 쌍 (y,x)를 생성하고 공개하는 과정은 일반 PKI와 같다.

1. 각 사용자는 위탁기관의 공개키를 사용하여 공개/비밀키 쌍 (y,x)를 생성한다. 이 때 키 생성은 아래 식을 만족해야 한다.
   $y = g^x \bmod n^2$ where $x <_R n$, $x < n$
   ID와 함께 공개키 y를 서명 받기 위해 인증기관에게 보낸다.
2. 인증기관은 사용자의 공개키 y를 서명하여 공개키 데이터베이스에 인증서를 보낸다.

사용자의 비밀키 x를 복구하는 과정은 다음과 같다.

1. 특정 사용자가 자신의 비밀키 x를 복구하고자 할 때 키 복구용 공개키/비밀키 쌍 (y_{rec},x_{rec})를 생성한다. 키 복구용 공개키 y_{rec}를 서명 받기 위하여 자신의 ID와 공개키를 인증기관에 보낸다.
2. 인증기관은 공개키 y_{rec}를 서명한 인증서를 사용자에게 보낸다.
3. 사용자는 비밀키 복구 요청서와 키 복구용 공개키 인증서를 키 보관 기관에게 보낸다.
4. 키 보관 기관은 사용자를 검증하고 공개키 데이터 베이스에서 키 복구를 요청한 사용자의 공개키 y를 얻어 마스터 비밀키로 아래 식과 같이 복호하여 사용자의 비밀키를 얻는다.
   $$\frac{L(y^\lambda) \bmod n^2}{L(g^\lambda)} \bmod n = x$$
   키 보관 기관은 사용자의 비밀키를 키 복구용 공개키로 암호화한 $E_{yrec}(x)$를 사용자에게 보낸다.
5. 사용자는 $D_{xrec}(E_{yrec}(x))$키 복구용 비밀키 y_{rec}로 복호하여 자신의 비밀키 x를 얻는다.

마스터 키를 갖는 키 보관 시스템은 PKI와 연동하여 사용하기에 편리하지만 모든 사용자의 비밀키는 키 보관 기관에 노출되어 있다. 즉 사용자의 비밀키 복구 수행자는 키 보관 기관이다.

4. 다수 키 보관 기관을 갖는 키 보관 시스템

이 장에서는 사용자의 프라이버시를 보호할 수 있는 키 보관 기관을 다수로 하는 키 보관 시스템을 설명한다. 2001년 유희종, 최희봉, 오수현, 원동호는 1997년 D. Boneh 등이 제안한 방법[3]을 사용하여 키 위탁 기관들의 키 생성 시 마스터 비밀키의 분산을 SE-PKI에 적용하였다. 여기서 Paillier 암호 방식의 특성을 이용하면 안전한 키 보관 시스템을 설계할 수 있다. 사용자만을 위한 비밀키 보관 시스템이기 때문에 유희종 등의 논문에서 키 복구 가능함을 증명하는 영지식 증명 프로토콜을 사용할 필요는 없고 인증기관의 도움을 얻어 비밀키를 복구하고자 하는 사용자를 인증하는 것이 중요하다.

키 보관 기관의 마스터 공개키는 n(=pq), g, 마스터 비밀키는 $\varphi(n)$이다. 키 보관 기관들이 나누어 갖게 되는 비밀키는 $\varphi(n)$ 값이다. 일반적으로 임의의 참여기관 수만큼 비밀 분산이 성립하기 때문에 i 개 참여한 키 보관 기관에 대해서도 성립한다[3].

$$n = pq = (p_1 + p_2 + \ldots + p_i)(q_1 + q_2 + \ldots + q_i)$$

$$\varphi(n) = (n - p_1 - q_1) - (p_2 + q_2) - \ldots - (p_i + q_i)$$

$$\varphi(n) = \varphi(1) + \varphi(2) + \ldots + \varphi(i)$$

이러한 i 개 키 보관 기관은 각 각 자신의 비밀키 $\varphi(1)$, ..., $\varphi(i)$를 분산하여 갖게 된다. 안전하게 분산하여 갖는 방법은 D. Boneh 등이 제안한 논문에서 설명하고 있다.

이렇게 키 보관 기관들이 비밀키를 분산하여 갖고 있고 공개 파라메터와 인증기관의 파라메터가 배포되어 있다고

가정하자. 사용자의 공개키/비밀키 쌍 (y,x)을 생성하고 공개하는 과정은 3장과 똑 같다.

사용자의 비밀키 x를 복구하는 과정은 다음과 같다.

1. 특정 사용자가 자신의 비밀키를 복구하고자 할 때 키 복구용 공개키/비밀키 쌍 (y_{rec}, x_{rec})을 생성한다. 키 복구용 공개키 y_{rec}를 서명받기 위하여 자신의 ID와 공개키 y_{rec}를 인증기관에 보낸다.
2. 인증기관은 키 복구용 공개키 y_{rec}를 서명한 인증서를 사용자에게 보낸다.
3. 사용자는 비밀키 복구 요청서와 키 복구용 공개키 인증서를 키 보관 기관들 모두에게 보낸다.
4. 각 키 보관 기관들은 사용자를 검증하고 자신 비밀정보 $\varphi(i)$를 이용하여 $(g^{\varphi(i)}, y^{\varphi(i)})$를 계산하여 사용자의 키 복구용 공개키 y_{rec}로 암호화한 $E_{yrec}(g^{\varphi(i)}, y^{\varphi(i)})$을 사용자에게 보낸다.
5. 사용자는 모든 키 보관 기관들로부터 얻은 $E_{yrec}(g^{\varphi(i)}, y^{\varphi(i)})$을 키 복구용 비밀키 x_{rec}로 복호화 하여 $D_{xrec}(E_{yrec}(g^{\varphi(i)}, y^{\varphi(i)}))=(g^{\varphi(i)}, y^{\varphi(i)})$를 얻는다. 그리고 $(g^{\varphi(1)}, y^{\varphi(i)})$, ..., $(g^{\varphi(i)}, y^{\varphi(i)})$로부터 아래와 같이 비밀키 x를 계산한다.

$$\frac{L(y^{\phi(1)} \cdot y^{\phi(2)} \cdot \cdot y^{\phi(i)}) \bmod n^2}{L(g^{\phi(1)} \cdot g^{\phi(2)} \cdot \cdot g^{\phi(i)}) \bmod n^2} \bmod n$$

$$= \frac{L(y^{\phi(n)} \bmod n^2)}{L(g^{\phi(n)} \bmod n^2)} \bmod n = x$$

위에서 살펴본 바와 같이 사용자의 비밀키를 복구하는데 사용자 외에 어떤 기관도 비밀키를 알지 못함을 알 수 있다.

물론 참여하고 있는 키 보관 기관이 공모하지 않는 조건을 만족해야 한다. 여기서 사용자의 비밀키 복구 수행은 사용자임을 알 수 있다.

5. 키 보관 시스템의 비교

비밀키를 저장하는 키 보관 시스템은 모든 사용자의 비밀키 저장할 수 있는 메모리가 필요하다. 그리고 사용자의 비밀키 복구 수행자는 키 보관 기관이다. 그러므로 키 보관 기관은 모든 사용자의 비밀키를 알 수 있다. 그러나 사용자와 키 보관 기관의 암호 방식은 동일한 것으로 설계할 수 있다. 키 보관 시스템의 안전성은 사용된 암호 방식과 PKI에 의존하며 키 보관 기관의 메모리에 대한 템퍼 보호도 중요하다.

마스터 키를 갖는 키 보관 시스템은 마스터의 비밀키만 저장하면 된다. 따라서 많은 메모리는 필요하지 않다. 그러나 키 보관 기관은 모든 사용자의 비밀키를 알 수 있다. 즉 사용자의 비밀키 복구 수행자는 키 보관 기관이다. 3장에서 사용자는 ElGamal 암호 방식으로 설계되어야 하고 키 보관 기관은 Paillier 암호 방식과 ElGamal 암호 방식 모두 구현되어 있어야 함을 알 수 있었다. 키 보관 시스템의 안전성은 사용된 ElGamal 암호 방식과 Paillier 암호 방식 및 PKI에 의존한다.

다수 키 보관 기관을 갖는 키 보관 시스템은 많은 메모리는 필요하지 않다. 사용자의 비밀키 복구 수행자는 사용자이기 때문에 사용자의 비밀키를 사용자 외 어떤 기관도 알 수 없다. 그러나 4장에서 본 바와 같이 사용자는 ElGamal 암호 방식과 Paillier 암호 방식 모두 구현되어 있어야 하며 키 보관 기관도 ElGamal 암호 방식과 Paillier 암호 방식 모두 구현되어 있어야 함을 알 수 있었다. 이 키 보관 시스템의 안전성은 ElGamal 암호 방식과 Paillier 암호 방식 및 PKI에 의존한다.

6. 결 론

본 논문에서는 공개키 기반 구조와 연동할 수 있는 비밀키 보관 시스템 3개를 제안하였다. PKI와 연동할 수 있기 때문에 온 라인으로 비밀키 복구가 가능하다. 3개의 비밀키 보관 시스템 중 가장 안전한 것은 다수 키 보관 기관을 갖는 키 보관 시스템이다. 이 시스템은 사용자와 키 보관 기관이 두 개의 암호 방식(ElGamal 암호 방식과 Pallier 암호 방식)으로 구현되어야 하며 사용자의 비밀키를 사용자 외에 어떤 기관도 알 수 없다는 장점을 가지고 있다. 이것은 키 보관 기관이 공모하지 않는 조건을 만족해야 한다.

이들 키 보관 시스템은 분실한 사용자의 비밀키를 복구할 뿐만 아니라 템퍼 보호가 불가능한 시스템에서나 비밀키 생성기능은 가지고 있으나 저장 기능이 없는 시스템에서 편리하게 사용할 수 있다. 또한 사용자가 비밀키를 자주 변경하더라도 비밀키 관리가 용이하다. 예를 들면 장기로 보관하는 대량의 문서를 암호화한 경우 비밀키를 변경할 때 암호화된 문서를 다시 복호화하여 새로운 공개키로 암호화할 필요 없이 처음에 암호화한 공개키를 함께 보관하면 되기 때문에 키 관리가 편리하다.

앞으로 연구과제로서 사용자 비밀키 보관을 안전하게 보호하기 위한 시스템을 설계하려면 참여하고 있는 키 보관 기관 중 적어도 하나라도 공모하지 못하도록 하는 대책이 필요하다.

참 고 문 헌

[1] A. Young, M. Yung, "Auto-Recoverab le and Auto-Certifiable Cryptosystems", A dvanced in Cryptology-Eurocrypt'98, Spri nger-Velag, Lecture Notes in Computer Sci ence, Springer-Verlag, pp.17-31,1998

[2] P. Paillier, Moti Yung, "Self-Escrowed Public Key Infrastructures", Proceedings of ICISC'99, The 2nd International Conference on Information Security and

Cryptology. Springer-Verlag, Lecture Notes in Computer Science, LNCS, Dec. 9 10, 1999,

[3] D. Boneh, M. Franklin, "Efficient generation of shared RSA keys", In Proceedings Crypto'97, Lecture Notes on Computer Science, Vol. 1223, Springer-Verlag. pp. 425-439, 1997

[4] G. Poupard, J. Stern, "Generation of Shared RSA Keys by Two Parties", Advanced in Cryptology-Asiacrypt'98, Springer-Velag, Lecture Notes in Computer Science, Springer-Verlag. LNCS 1514, pp.11-24,1998

[5] P. Paillier, "Public-Key Cryptosystem Based on Composite Degree Residuosity Classes", Advanced in Cryptology-Eurocrypt'99, Springer-Velag, Lect ure Notes in Computer Science, Springer-Verlag, p p. 223-238,1999

[6] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarith ms", In Crypto'84, pp. 10-18, 1984

[7] A. Camenisch, Y. Frankel and Y. Tsiounis. Easy Come-Easy Go Divisible Cash. In Advances in Cryptology, Eurocrypt'98, LNCS 1403, pp. 561- 575, Springer Verlah, 1998.

[8] 유희중, 최희봉, 오수현, 원동호, "다수의 위탁기관 참여 가능한 SE-PKI 키 복구 시스템" 한국정보보호학회 논문지, 2001. 2.