通信情報保護學會論文誌
第 11 券 第 1 號, 2001. 2

다수의 위탁 기관 참여가 가능한 SE-PKI 키 복구 시스템

유 희 종*, 최 희 봉**, 오 수 현***, 원 동 호***

SE-PKI Key Recovery system with multiple escrow agents

Hui-jong Yu*, Hee-bong Choi**, Soo-hyun Oh***, Dong-ho Won***

요 약

1998년 A. Young 등은 공개키 기반구조(PKI)를 이용한 키 복구 시스템인 ARC를 제안하였다. 또한 1999년 P. Paillier 등은 ARC를 개선하여 필요한 저장 공간을 제거한 SE-PKI 키 복구 시스템을 제안하였다. 그러나 SE-PKI 키 복구 시스템은 저장 공간이 줄어든 반면, 다수의 위탁 기관을 참여시키지 않고 하나의 위탁 기관만을 사용하고 있다. SE-PKI는 이에 대해서 임의의 비밀분산 방식을 사용할 수 있다고만 언급하고 있다. 따라서 본 논문에서는 SE-PKI와 마찬가지로 부가적인 저장 공간이 요구되지 않으면서 다수의 위탁 기관이 참여 가능한 키 복구 시스템을 제안한다. 또한 제안 시스템은 적법한 사용자나 법 집행 기관만이 사용자의 비밀 메시지를 복구할 수 있으며 키 복구가 이루어진다고 하더라도 위탁 기관은 사용자의 비밀 메시지를 알 수 없다는 장점이 있다.

ABSTRACT

In 1998, A. Young and M. Yung introduced the concept of ARC that conjugates functionalities of a typical PKI with the ability to escrow private keys of the system users. Also in 1999, P. Paillier and M. Yung proposed a new notion - called SE-PKI - which presents other additional advantages beyond ARC. But SE-PKI system uses only one escrow agent. The storage of user's secret information at a single agent can make it significant point of attack and arouse controversy about invasion of privacy. This paper presents SE-PKI key recovery system that multiple escrow agents can participate in it. Also, in our system, escrow agents can't recover user's ciphertext.

keyword : *key recovery, ARC, SE-PKI, multiple escrow agency*

Ⅰ. 서 론

현대 사회가 점차 고도의 정보화 사회로 발전해 가면서 다양한 정보의 개방과 공유, 네트워크를 통한 업무 처리의 일반화는 무한한 가능성과 편리함을 주었지만 정보의 침해라는 문제를 발생시켰다. 이로 인하여 정보 보호의 문제가 부각되었으며, 이전에 군사상의 목적 등 국가적 차원에서 주로 이용되었던 암호의 사용이 민간 부문으로 확대되었다.

암호의 사용은 정보의 누출 및 오용을 방지하고 상대방의 신원을 확인할 수 있게 함으로써 온라인 상에서 전자상거래나 전자 계약을 가능하게 하는 등 많은 장점을 가지고 있다. 그러나 이렇게 편리한 면도 가지고 있지만 잘못 사용하면 다음과 같은 역기

* 한국전자통신연구원 정보보호기술연구본부 차세대보안응용연구부(anny5@etri.re.kr)
** 국가보안기술 연구소(wongchoi@hananet.net)
*** 성균관대학교 전기전자및컴퓨터공학과 정보통신보호연구실(shoh@dosan.skku.ac.kr, dhwon@simsan.skku.ac.kr)

능도 발생하게 된다.

첫째, 국가가 범죄 수사 등의 합법적인 이유로 키에 접근해야 할 필요성이 있을 경우 발생되는 문제점이다. 암호는 키를 아는 사람만이 암호문을 복호할 수 있는 기밀성 기능을 포함하고 있기 때문에, 범죄자들은 암호를 사용함으로써 합법적인 수사를 방해할 수 있다.

둘째, 키의 분실이나 손상으로 인하여 사용자가 자신의 정보에 접근할 수 없는 경우이다. 이 경우 적법한 키의 소유자라고 할지라도 자신의 정보에 대한 접근을 할 수 없으므로 많은 손실을 가져올 수 있다.

셋째, 암호가 오용됨으로써 발생할 수 있는 잠재적 위협이 존재하는 경우로, 사업장에서 피고용인이 중요한 정보를 암호화하고 키를 담보로 금품을 요구할 수도 있으며, 키의 도난이나 손상 등의 위협이 항상 존재한다.

전자상거래 등의 민간 부문으로 암호 사용이 점차 확산·보급되고 있는 상황에서 앞에서 언급한 암호 사용의 부작용을 해결하는 것이 급선무이다. 이러한 맥락에서 현재 세계 각 국에서는 암호키 관리에 대한 연구가 활발히 진행 중에 있으며, 그 가운데 키 복구가 중심 연구과제로 부상되고 있다.

일반적으로 키 복구란 암호문의 소유자(일반적으로 암호 시스템에서 키를 소유한 사람)만이 평문으로 복호할 수 있는 암호화된 데이터에 대해 특정한 조건(암호가 나쁜 목적으로 사용되었을 경우 법 집행권을 확보하기 위한 합법적 허가, 데이터 암호용 키를 분실·손상하였을 경우)이 만족될 경우에 한해서 허가된 사람 또는 기관에게 복호가 가능한 능력을 제공하는 기술 및 체계를 말한다.

키 복구는 각 특성에 따라 키 위탁 방식, 캡슐화 방식, TTP 기반의 방식으로 나눌 수 있으며 일반적으로 사용자의 비밀키를 신뢰기관에 위탁함으로써 법 집행 기관의 데이터 접근권을 보장하는 키 위탁 방식이 많이 사용된다. 그러나 개인의 사생활 보호와 정부의 법 집행 능력 보장이라는 두 가지 상반된 목적에 대하여 끊임없는 논쟁이 진행되고 있다. 실제 키 복구 실증 프로젝트를 수행한 미국과 같은 경우 다수의 신뢰 위탁 기관을 사용하여 비밀 정보를 분산 위탁함으로써 사생활 침해라는 문제점을 완화하고 있다. 그러므로 키 복구 방식을 실제 수행할 경우 다수의 키 위탁 기관의 참여는 필수적이라 하겠다.

위에서 살펴본 바와 같이 키 복구는 암호의 대중화와 함께 발생한 여러 역기능을 해결할 수 있는 방법이다. 그러나 키 복구 방식을 현재의 암호 사용자 환경에 무조건 도입하는 것은 시간과 비용, 방법적인 측면에서 무리가 있다.

현재 각 나라에서는 전자 상거래와 같은 암호 응용 분야에서 유용하게 사용될 수 있는 공개키 기반 구조 구축이 진행되고 있으며 여러 선진국에서는 이미 구축이 되어 서비스가 진행중이다. 따라서 이러한 공개키 기반구조에 키 복구 방식을 대입하는 것은 효율적인 방법이라 할 수 있으며 또한 현재까지 이러한 연구가 상당히 이루어진 상태이다.

1998년 A. Young과 M. Yung은 공개키 기반 구조를 이용하여 다수의 위탁 기관이 참가 가능한 키 복구 시스템(ARC, Auto-Recoverable Auto-Certifiable Cryptosystem)을 제안하였다.[1] 사용자는 위탁 기관의 공개키를 이용하여 자신의 비밀키를 생성하고 이를 인증 기관에 증명함으로써 공개키에 대한 인증서를 발급 받는다. 그러나 이 방식은 사용자의 비밀키가 위탁 기관의 공개키를 사용하여 생성되었음을 증명하는 증명서를 공개키 인증서와 함께 저장해야 하는 저장 공간이 요구된다. 이는 사용자의 비밀키를 복구할 때에 이 증명서가 필요하기 때문이며 따라서 위탁 기관과 인증 기관 사이의 통신이 필요하다는 단점도 있다.

따라서 1999년 P. Paillier와 M. Yung은 ARC를 개선시킨 키 복구 시스템 SE-PKI(Self-Escrowed Public Key Infrastructure)를 제안하였다.[2] SE-PKI는 ARC에서 요구되었던 저장 공간을 없애고, 이에 따라 위탁 기관과 인증 기관의 통신도 필요없게 되었다는 장점을 가지고 있으나, 이 방식에 따르면 사용자의 비밀키를 복구할 수 있는 위탁 기관의 비밀 정보가 한 위탁 기관에만 집중되어 있기 때문에 사용자의 프라이버시 침해라는 키 복구

정책 본래의 논쟁점을 가져오게 되었다. SE-PKI 시스템에서는 임의의 비밀 분산 방식을 사용함으로써 이를 해결할 수 있다고 언급되어 있다. 따라서 본 논문에서는 SE-PKI의 장점에 다수의 위탁 기관을 참가시켰으며 사용자의 암호문을 사용자의 비밀키가 위탁되어 있는 위탁 기관이 복구할 수 없다는 장점이 있다. 제안하는 키 복구 시스템에서의 사용자는 SE-PKI에서의 사용자와 동일하게 행동하며 부가적인 계산이나 저장 공간은 요구되지 않는다. 요구되는 것은 키 복구 시스템 설정 과정에서 위탁 기관들이 비밀 정보를 나누어 갖기 위한 연산 뿐이며, 여기서 제 삼자의 도움이 요구된다. 그러나 제 삼자와의 모

든 통신이 랜덤한 값을 포함하고 있기 때문에 이 시스템에 참가하는 제 삼자가 얻을 수 있는 정보는 결과 값인 공개 정보뿐이며, 또한 위탁 기관이 비밀 정보를 저장하고 있는 어느 정도의 신뢰성을 가진다고 가정한다면 제안하는 시스템은 안전하다.

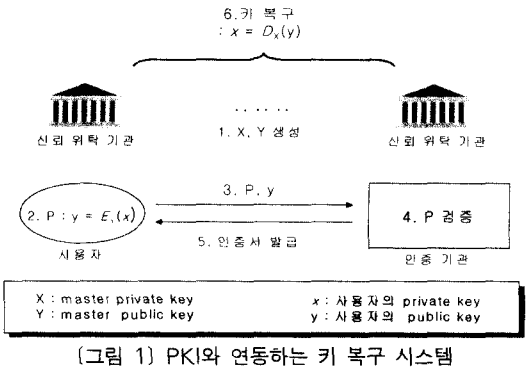
본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 배경이 되는 PKI와 연동하는 키 복구 시스템에 대하여 알아보고 3장에서는 제안하는 시스템을 설명한다. 4장에서는 제안 시스템의 안전성과 효율성에 대하여 논의하며 5장은 결론으로 구성되어 있다.

II. 배 경

PKI와 연동하는 키 복구 시스템의 기본적인 형태는 [그림 1]과 같다. 먼저 신뢰 위탁 기관들이 마스터 비밀/공개키를 생성하여 마스터 공개키를 공개하면 사용자는 이것을 이용하여 자신의 비밀/공개키 쌍을 생성한다. 그리고 사용자는 자신의 키 쌍이 위탁 기관의 마스터 비밀키를 사용하여 복구 가능하다는 것을 증명하는 증명서 P와 자신의 공개키를 인증기관에 전달한다. 인증기관은 전송 받은 P를 검증하고 이 검증에 성공하면 사용자의 공개키에 대한 인증서를 발급한다. 차후에 법 집행 기관이나 사용자의 키 복구는 위탁 기관들의 마스터 비밀키를 사용함으로써 이루어진다. 시스템의 세부 과정은 2.1절에 나타내었고 SE-PKI 시스템은 2.2절에 설명하였다.

2.1 PKI와 연동 가능한 키 복구 시스템

$S=\langle G, E, D\rangle$는 사용자의 키 생성 알고리즘, 암/복호 알고리즘이며 $\Sigma=\langle G, E, D\rangle$는 위탁 기관의 마스터 키 생성 알고리즘, 암/복호 알고리즘이다.



[그림 1] PKI와 연동하는 키 복구 시스템

(1) 시스템 설정

위탁 기관들은 G를 사용하여 마스터 비밀/공개 키 (X, Y)를 생성한다.

(2) 키 생성과 공개 위탁 검증

각 사용자는 $G(Y, 1^k)$ 알고리즘을 이용하여 공개/비밀키 쌍 (x, y)를 생성하고 $y=E_Y(x)$를 만족하는 증명서 P를 공개키 y와 함께 인증기관에 전송한다.

인증 기관은 P를 검증하여 검증에 성공하면 사용자의 공개키 y에 대한 인증서를 발급한다.

(3) 암/복호화

사용자는 인증 기관으로부터 수신자의 공개키와 인증서를 얻은 후 이를 검증하고 검증이 성공하면 전송하고자 하는 메시지를 수신자의 공개키로 암호화하여 전송한다.

수신자는 비밀키를 이용하여 암호문을 복호한다.

(4) 키 복구

어떤 사용자에 대하여 키 복구가 인증되면 위탁 기관은 자신의 비밀정보를 이용하여 $x=D_X(y)$를 계산하여 사용자의 비밀키를 복구해낸다.

2.2 SE-PKI(Self-Escrowed Public Key Infrastructure)

SE-PKI에서는 두 개의 시스템을 제안하고 있다. 첫 번째는 Diffie-Hellman의 키 구축 방식을 이용한 시스템이고 두번째는 Paillier의 암호 방식과 ElGamal의 암호 방식을 사용한 시스템이다.

본 논문에서는 두 번째 방식을 설명하고 이를 개선한 키 복구 시스템을 제안한다. 2.2.1절에서는 두 번째 방식의 시스템 설정, 암/복호 프로토콜을 설명하고 2.2.2절에서는 두 번째 방식의 복구 가능한지를 증명하는 증명서를 생성하고 이를 검증하는 프로토콜을 설명한다.

2.2.1 시스템 설정과 암/복호 프로토콜

본 절에서는 SE-PKI에서 제안된 두 방식 중에서 위탁 기관의 마스터 암호 방식으로는 1999년 제안된 P. Paillier의 확률론적 암호 방식을 결정적 암호방식으로 수정하여 사용하고,[5] 사용자의 암호 방식으로는 ElGamal의 방식을 이용한 시스템을 소개한다. 기본적인 암/복호 과정은 다음과 같으며 전체 과정을 [표 1]에 나타내었다.

〔표 1〕 SE-PKI의 암/복호 과정

마스터 공개키	n, g, $l = 2|n|$
마스터 비밀키	lcm(p-1, q-1)
사용자의 공개키	$y = g^x \bmod n^2$ where $x <_R n$
사용자의 비밀키	$x < n$ $m < n^2$
암호화	plaintext ciphertext $c = (my^k, g^k)$ where $k <_R 2^l$
복호화	ciphertext $c = (a, b)$ plaintext $m = a/b^x \bmod n^2$

① 위탁 기관은 마스터 비밀/공개키를 생성한다. P. Paillier의 프로토콜에 따라 마스터 공개키는 $n = pq$, $g \in Z^*_{n^2}$($\gcd(n, \alpha) = 1$인 어떤 α에 대한 $n\alpha$를 위수로 갖음), $al = 2|n|$이며 마스터 비밀키는 lcm(p-1, q-1)이다. 위탁 기관은 마스터 공개키를 공개한다.

② 사용자는 공개된 마스터 공개키 n을 이용하여 자신의 비밀/공개키 쌍을 생성한다. 사용자의 비밀키는 $x < n$, 공개키는 $y = g^x \bmod n^2$이며 2.1.2절에서 설명되는 과정을 거쳐 인증기관에게 공개키에 대한 인증서를 발급받으면 암호 통신을 할 수 있다.

③ 사용자들이 암호 통신을 할 때에 이용되는 암호 방식은 ElGamal 방식이며 평문이 $m < n^2$이면 암호문은 $c = (my^k, g^k)$, $k <_R 2^{|l|}$이며 이를 전송받은 측에서는 암호문을 $c = (a, b)$로 설정하고 $m = a/b^x \bmod n^2$을 계산하여 복호한다.

2.2.2 복구 가능의 증명서 생성과 검증

이 과정은 사용자가 자신의 비밀키가 위탁 기관의 공개키를 사용하여 생성되었기 때문에 위탁 기관의 비밀키를 사용하여 복구 가능하다는 증명서 P를 생성하여 인증 기관에 전송하고 이를 인증 기관이 검증하는 과정으로 이루어져 있다. 검증이 성공하면 인증 기관은 사용자에게 공개키 인증서를 발급하게 된다. 이 과정은 〔표 2〕에 나타나 있으며 사용자는 증명자, 인증 기관은 검증자의 역할을 수행한다.

2.2.3 키 복구

복구 과정은 다음과 같다. 이는 P. Paillier의 복호화 방식이다.

$$\frac{L(y^{\lambda(p-1,q-1)})}{L(g^{\lambda(p-1,q-1)})} = x$$

〔표 2〕 복구 가능의 증명서 생성과 검증

파라미터	$n, l = 2|n|, g \in Z^*_n$ (최대 위수를 가짐) p_0 : n이 $p_0 - 1$로 나누어 떨어지는 소수 $h_1, h_2 : Z^*_{p_0}$에서 위수 n을 가지는 수
증명자 - P 생성	
입력	$x < n$ and $y = g^x \bmod n^2$ 1. pick $z <_R n$ and compute $\bar{y} = h_1^z h_2^x \bmod p_0$ 2. pick $r_0 <_R 2^l$ and $r_1, r_2 <_R n$ 3. compute $e = H(g^{r_0} \bmod n^2, h_1^{r_1} h_2^{r_2} \bmod p_0)$ (H : 해쉬함수) 4. compute $s_0 = r_0 + ez$ $s_1 = r_1 + ez \bmod n$ $s_2 = r_2 + ez \bmod n$
출력	$P = (\bar{y}, e, s_0, s_1, s_2)$
검증자 - 검증	
입력	$y < n^2$ and $P = (\bar{y}, e, s_0, s_1, s_2)$ $e = H(g^{s_0}/y^e \bmod n^2, h_1^{s_1} h_2^{s_2}/\bar{y}^e \bmod p_0)$인지 검증
출력	검증 결과에 따른 인증서 발급/발급 거부

여기서 L은 다음과 같은 함수이다.

$$L(u) = \frac{u-1}{n} \text{ where } U_n = \{u < n^2 \mid u = 1 \bmod n\}$$

Ⅲ. 제안하는 키 복구 시스템

위탁 기관들의 키 생성 시 마스터 비밀키의 분산을 위해서 1997년 D. Boneh등이 제안한 방법을 사용한다.[3] 이 방법은 RSA 키 생성시에 비밀키의 분산을 가능하게 하는 방법으로 이를 SE-PKI에도 적용할 수 있다.

본 논문에서는 두 개의 위탁 기관을 가정하고 시스템을 구성하였다. 3개 이상의 위탁 기관의 참여도 가능하며 이 역시도 D. Boneh등의 방법을 사용하여 구성 가능하다.

3.1 시스템 설정

위탁기관의 마스터 공개키는 $n(=pq)$, g, 마스터 비밀키는 $\varphi(n)$이다. 이 키들을 두 위탁 기관 A, B가 나누어 갖기 위해 제 삼자 H의 도움으로 다음을 수행한다.

3.1.1 소수 생성

A와 B는 각각 다음과 같은 p_a와 q_a, p_b와 q_b를

생성한다. 위탁 기관이 세 개일 경우에는 p_c, q_c까지 생성될 것이다. 이러한 방법으로 n개의 위탁 기관까지 생성할 수 있다.

$$p_a = q_a = 3 \bmod 4$$
$$p_b = q_b = 0 \bmod 4$$

다음과 같은 공개키 n을 구성하기 위해 어떤 수가 소수인지를 검사하는 Trial Division 방법을 사용한다.

$$n = (p_a + p_b)(q_a + q_b)$$

즉, $q_a + q_b$가 소수인지를 알기 위해 다음과 같이 검사하며 $p_a + p_b$도 같은 방법으로 수행한다.

① A, B는 $p_1, p_2, \cdots, p_j$를 작은 소수들의 집합으로 구성한다.
② A는 임의의 $c_i \in Z_{p_i}$, $d_i \in Z_{p_i}^*$를 선택하여 다음을 계산한다.
$u_i = c_i + d_i q_a \bmod p_i$ for all i, $1 \le i \le j$
A는 B에게 $c_1, d_1, \cdots, c_j, d_j$를, H에게 $u_1, \cdots, u_j$를 전송한다.
③ B는 다음을 계산하고
$v_i = c_i - d_i q_b \bmod p_i$ for all i, $1 \le i \le j$
H에게 $v_1, \cdots, v_j$를 전송한다.
④ H는 $u_i \neq v_i$ for all i, $1 \le i \le j$이면 '성공'을 아니면 '실패'를 A와 B에게 전송한다.

3.1.2 공개키 n의 계산

A와 B는 각각 위탁 기관들이며 H는 제 삼자이다. 공개키는 다음과 같이 계산한다.

① A는 n보다 큰 임의의 소수 P와 $c_a, d_a \in Z_P^*$를 선택하고 $x_a = 1, x_b = 2, x_h = 3$를 설정하여 for i = a, b, h에 대해 $p_{a,i} = c_a x_i + p_a$과 $q_{a,i} = d_a x_i + q_a$를 계산한다. A는 $p_{b,a}, q_{b,a}$와 $r(0) = 0, r_i = r(x_i)$인 임의의 이차 다항식 r(x)를 선택하여 $N_a = (p_{a,a} + p_{b,a})(q_{a,a} + q_{b,a}) + r_a$를 계산한다. A는 $p_{a,b}, q_{a,b}, p_{b,a}, q_{b,a}, r_b$를 B에게 $p_{a,h}, q_{a,h}, r_h, N_a$를 H에게 전송한다.
② B는 $c_b = (p_{b,a} - p_b)/x_a$, $d_b = (q_{b,a} - q_b)/x_b$를 계산하고 i = b, h에 대해 $p_{b,i} = c_b x_i + p_b$, $q_{b,i} = d_b x_i + q_b$를 계산한다. 그리고 이 값들을 통하여 $N_b = (p_{a,b} + p_{b,b})(q_{a,b} + q_{b,b}) + r_b$를 계산해 $p_{b,h}$, $q_{b,h}$, N_b를 H에게 전송한다.
③ H는 $N_h = (p_{a,h} + p_{b,h})(q_{a,h} + q_{b,h}) + r_h$를 계산하고 $(x_a, N_a), (x_b, N_b), (x_h, N_h)$를 지나는 이차 다항식 $\alpha(x)$를 계산한다. i = a, b, h에 대해

$$\alpha(x) = ((c_a x + p_a) + c_b x + p_b))((d_a x + q_a) + (d_b x + q_a)) + r(x)$$

이므로 $\alpha(0) = n$이다. H는 계산된 n 값을 공개한다.
이 과정을 [표 3]에 정리하였다.

3.1.3 공개키 n의 검증

본 절에서는 n이 두 소수의 곱인지를 검사한다. A와 B가 다음과 같은 수를 선택하였기 때문에 n이 두 소수의 곱이라면 n은 블럼 정수가 된다.

$$p_a = q_a = 3 \bmod 4$$
$$p_b = q_b = 0 \bmod 4$$
$$n = (p_a + p_b)(q_a + q_b)$$

검사 과정은 다음과 같다.

① A와 B는 $g \in_R Z_n^*$ 값을 동의한다.
② $\left(\frac{g}{n}\right) \neq 1$이면 ①로 돌아간다.
③ A는 $v_a = g^{(n - p_a - q_a + 1)/4} \bmod n$을,
B는 $v_b = g^{(p_b + q_b)/4} \bmod n$을 계산하고 이 값들을 교환한 후 다음을 검증한다.

$$v_a = \pm v_b \pmod n$$

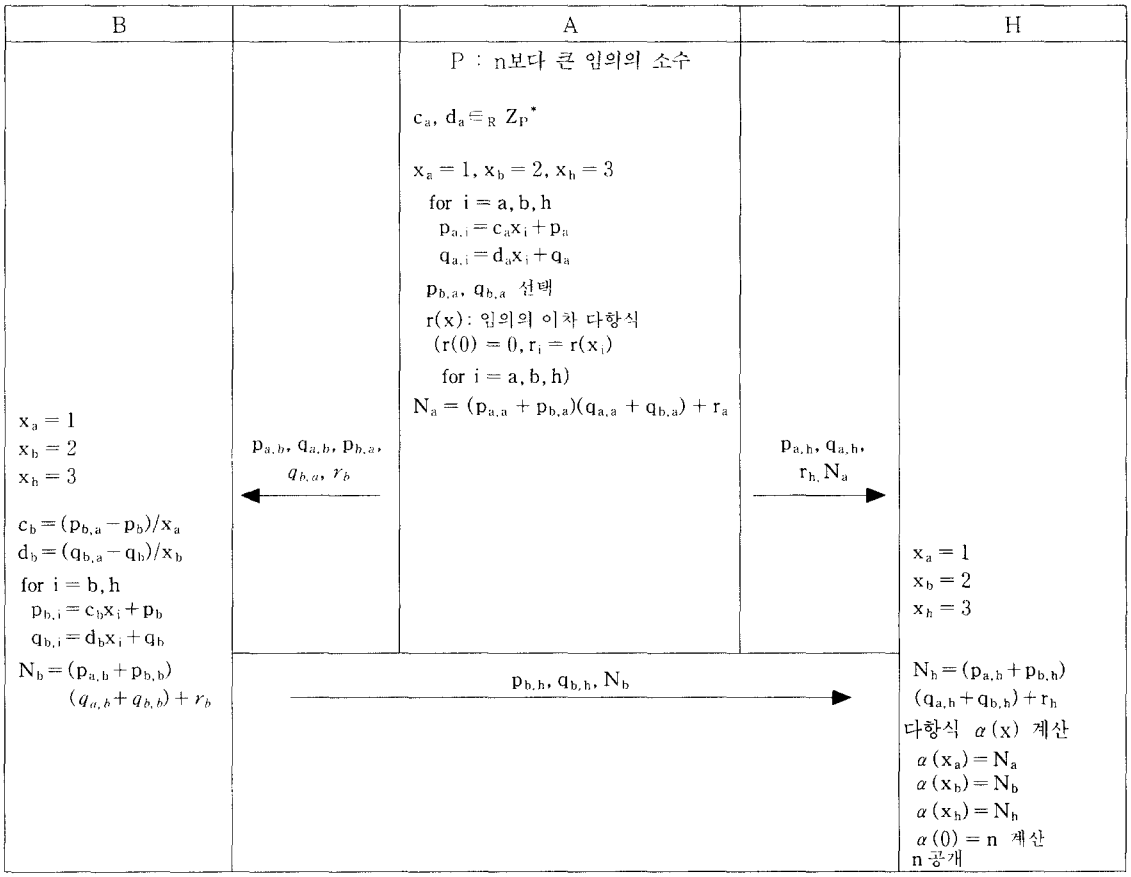
검증이 실패하면 n은 두 소수의 곱이 아니며 처음부터 다시 시작한다.

3.1.4 마스터 비밀키의 분산

위탁 기관들이 나누어 갖게 되는 비밀키는 $\varphi(n)$ 값이다. 이는 다음과 같다.

$$n = pq = (p_a + p_b)(q_a + q_b)$$
$$\varphi(n) = (n - p_a - q_a) - (p_b + q_b)$$

〔표 3〕 공개키 n의 계산



즉, A는 φ_a를 가지고, B는 φ_b를 가지게 되는 다음의 형태로 비밀키의 분산이 가능하게 된다.

$$\varphi_a = (n - p_a - q_a)$$
$$\varphi_b = -(p_b + q_b)$$
$$\varphi(n) = (n - p_a - q_a) - (p_b + q_b) = \varphi_a + \varphi_b$$

3.2 키 생성과 공개 위탁 검증

사용자의 비밀키는 SE-PKI의 과정과 동일한 $x < n$, 공개키는 $y = g^x \bmod n^2$ 이며 이에 따라 공개 위탁 검증 과정도 동일하게 된다.

3.3 암/복호화

사용자들의 암/복호 알고리즘은 ElGamal 암호 알고리즘이다.[6]
암/복호화 과정을 〔표 4〕에 정리하였다.

〔표 4〕 제안하는 키 복구 시스템의 암/복호화 과정

마스터 공개키	n, g, $l = 2\|n\|$
마스터 비밀키	$\varphi(n)$
사용자의 공개키	$y = g^x \bmod n^2$ where $x <_R n$
사용자의 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 $m = a/b^x \bmod n^2$

3.4 키 복구

위탁 기관들의 암/복호 알고리즘은 1999년 P. Paillier가 제시한 알고리즘이며 비밀키만 lcm(p-1, q-1)에서 $\varphi(n)$로 바뀌었기 때문에 약간의 계산량이 늘어날수도 있다. 원래 암호문을 복호할 때의 계산은 암호문의 lcm(p-1, q-1) 지수승을 하면 복호

가 가능했었다. 비밀키가 $\varphi(n)$이 된다면 $\varphi(n)$이 lcm(p-1, q-1)을 포함하고 있기 때문에 계산량만 약간 늘어날 수 있고 복호가 가능해진다.

키 복구를 수행하는 주체는 보통 적합한 권한을 가진 법 집행 기관이나 개인 사용자가 될 수 있다. ARC나 SE-PKI 키 복구 시스템에서는 키 복구 수행자가 키 복구를 수행하면 위탁 기관의 비밀키가 노출되어 다른 사용자들의 비밀키가 모두 노출되므로 사전에 사용자마다 다른 마스터 비밀키를 유지하거나 키 복구 후에 위탁 기관이 사용자들의 암호문을 복호할 수 있다는 단점이 있다. 그러나 제안하는 시스템에서는 키 복구가 한번 이루어진 후에도 위탁 기관들에 저장된 비밀 정보가 드러남 없이 안전하게 유지될 수 있으며 사용자나 법 집행 기관만이 키 복구를 할 수 있고 위탁 기관들은 사용자들의 암호문을 복호할 수 없다는 장점이 있다. 키 복구 과정은 다음과 같다.

① 위탁 기관 A는 g^{φ_a}, y^{φ_a}를, B는 g^{φ_b}, y^{φ_b}를 계산하여 키 복구 수행자에게 전송한다. 이 정보들로부터 키 복구 수행자가 φ_a, φ_b를 계산하는 것은 이산대수 문제이다.
② 키 복구 수행자는 다음을 계산하여 사용자의 비밀키 x를 복구한다.

$$\frac{L(y^{\varphi_a}\, y^{\varphi_b})}{L(g^{\varphi_a}\, g^{\varphi_b})} \bmod n = \frac{L(y^{\varphi_a+\varphi_b})}{L(g^{\varphi_a+\varphi_b})} \bmod n = \frac{L(y^{\varphi_n})}{L(g^{\varphi_n})} \bmod n = x$$

3개 이상의 위탁 기관 참가시에도 위의 과정에 다른 위탁 기관과 마찬가지로 동작하여 키 복구를 수행한다.

Ⅳ. 안전성과 효율성

제안하는 키 복구 시스템은 크게 두 부분으로 나눌 수 있다. 하나는 시스템 설정 과정이며 또 하나는 사용자들이 암호 통신후 키 복구를 하는 과정이다. 안전성과 효율성 모두 이 과정으로 나누어 살펴보겠다.

4.1 안전성

시스템 설정 과정에서는 참가자들이 모두 위탁 기관이라는 신뢰 기관이다. 제 삼자 H의 도움이 필요하기는 하지만 주고받는 값들이 모두 랜덤한 값을 포함하고 있기 때문에 H에게는 비밀정보에 대한 어떤 정보도 노출되지 않는다. 즉, 제 삼자는 이 프로토콜을 벗어남으로써 어떤 이득도 얻을 수 없으며 실제 프로토콜 수행시 H를 다른 위탁 기관으로 설정한다면 모두 신뢰기관으로 이루어진 프로토콜이기 때문에 안전성을 보장받을 수 있다.

암호 통신과 키 복구 과정은 모두 SE-PKI의 안전성을 따른다.

4.2 효율성

제안 시스템을 수행하는데 있어서 사용자에게 부가되는 계산량이나 저장 공간은 없다. 시스템 설정 과정에서만 신뢰기관 끼리의 연산이 요구되므로 사용자 측면에서는 효율성이 SE-PKI와 동일하며 시스템 설정 과정에서도 신뢰기관 끼리의 비밀 분배를 위한 필수적 과정만이 추가되었다.

두 위탁 기관만이 참가할 경우 RSA 키를 더 효율적으로 분산할 수 있는 방법이 1998년 G. Poupard 등에 의해 제시되었으므로 효율성을 위해서 이를 사용할 수도 있다.[4]

Ⅴ. 결 론

본 논문에서는 공개키 기반 구조와 연동 가능한 효율적인 키 복구 시스템을 제안하였다. 공개키 기반 구조와 연동 가능한 키 복구 시스템은 이미 구축된 PKI 시스템을 기반으로 하여 키 복구 방식을 구현할 수 있으므로 사용자들에게 요구되는 비용이 적기 때문에 매우 효과적인 방법이다. 따라서 이를 이용한 여러 키 복구 시스템들이 연구되었으며 실제 프로토콜들이 소개되었다.

본 논문에서 제안하고 있는 키 복구 시스템은 기존에 이미 제안된 다른 방식들과 달리 요구되는 저장 공간이 최소화되었으며 다수의 키 위탁 기관이 참가 가능하다는 두 가지 장점을 동시에 가지고 있다. 또한 사용자 측면에서도 부가되는 계산량이나 저장 공간이 매우 효율적인 시스템이다.

제안하는 키 복구 시스템이 기존의 시스템과 다른 점은 시스템 설정 과정에서 제 삼자가 참가한다는 점이다. 그러나 제 삼자에게 노출되는 정보는 랜덤한 값을 첨가한 임의의 값 혹은 공개 정보뿐이며 비밀정보에 대한 노출 위험은 없다. 때문에 제 삼자에게는 큰 신뢰성을 요구하지 않아도 좋으나 제 삼자의 역할을 다른 위탁 기관으로 수행하도록 한다면

안전성을 더욱 보장받을 수 있다.

참 고 문 헌

[1] A. Young, M. Yung, "Auto-Recoverable and Auto-Certifiable Cryptosystems", *Advanced in Cryptology-Eurocrypt'98*, Springer-Velag, Lecture Notes in Computer Science, Springer-Verlag, 1998, 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-Verlag, 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, Lecture Notes in Computer Science, Springer-Verlag, pp. 223~238, 1999.

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

〈著者紹介〉

유 희 종 (Hui-jong Yu) 정회원
1999년 2월 : 성균관대학교 정보공학과 졸업
2001년 2월 : 성균관대학교 전기전자 및 컴퓨터공학과 석사 졸업
2001년 1월~현재 : 한국전자통신연구원 연구원
〈관심분야〉 암호이론, 정보이론

최 희 봉 (Hee-bong Choi) 정회원
1984년 2월 : 부산대학교 전기공학과 졸업
1987년 2월 : 부산대학교 전기공학과 석사
1997년 3월~현재 : 성균관대학교 전기전자 및 컴퓨터공학부 박사과정
1987년 2월~2000년 1월 : 국방과학연구소 선임연구원
2000년 2월~현재 : 국가보안기술연구소 선임연구원
〈관심분야〉 암호이론, 네트워크보안, 보안시스템 설계

오 수 현 (Soo-hyun Oh) 정회원
1998년 2월 : 성균관대학교 정보공학과 졸업(공학사)
2000년 2월 : 성균관대학교 전기전자 및 컴퓨터 공학과 대학원 졸업(공학석사)
2000년 3월~현재 : 성균관대학교 전기전자 및 컴퓨터 공학과 박사 과정
〈관심분야〉 암호이론, 부호이론

원 동 호 (Dong-ho Won) 정회원
1976년 : 성균관대학교 전자공학과 졸업
1978년 : 성균관대학교 전자공학과 석사
1988년 : 성균관대학교 전자공학과 박사
1978년~1980년 : 한국전자통신연구소 전임 연구원
1985년~1986년 : 일본 동경공대 객원연구원
1992년~1994년 : 성균관대학교 전산소장
1995년~1997년 : 성균관대학교 교학처장
1996년~1998년 : 국가정보화 추진위원회 자문위원
1990년~1999년 : 한국통신정보보호학회 이사
1998년~1999년 : 성균관대학교 정보통신기술연구소장
1999년~2001년 : 성균관대학교 전기전자 및 컴퓨터 공학부장
1982년~현재 : 성균관대학교 전기전자 및 컴퓨터 공학부 교수
1999년~현재 : 정보통신대학원장
한국통신정보보호학회 부회장
〈관심분야〉 암호이론, 부호이론