공개키 암호 시스템에서의 Key Escrow

정 경임°, 이 필 중
포항공과대학교 전자전기공학과

Key Escrow on Public Key System

Kyung Im Jung, Pil Joong Lee
Dept. of Eletronics and Electrical Engineering, POSTECH

요 약

개인의 사생활을 보호하는 것과 법집행기관의 합법적인 도청/복호화 요구 사이에서 균형을 맞출 수 있는 Key Escrow를 스마트 카드를 사용하는 공개키 암호 시스템에서 적용한다. 사용자의 비공개키는 권위있는기관/키관리센타와의 상호 작용또는 사용자가 만든 비공개키를 escrow하는 신뢰받는기관과의 상호작용에 의해 만들어지며 time warrant를 가져서 사용자가 세션키를 만드는 프로토콜을 따른다면 합법적인 도청/복호화의 경우 사용자의 비공개키는 공개되지 않는다.

1. 서 론

컴퓨터 네트웍을 통한 메세지의 교환이 많아지고 컴퓨터 속에 보관되어 있는 개인 자료가 많아지면서 이들에 대한 보호의 관심이 높아지고 있다. 그리고 암호학의 발달에 따라 이런 자료들을 암호화해 줄 수 있는 많은 암호학적 도구들이 나타나게 되었다. 그런데 이렇게 개인의 자료를 암호화 해 두는 경우에 어떤 필요한 경우에 법집행기관의 합법적인 도청이 매우 힘들게 되는 일이 생기게 되었다. 그래서 개인의 사생활 보호를 하면서도 테러나 국가 안보 혹은 어떤 위급한 경우에 합법적인 복호화와 도청을 할 수 있는 방법을 간구하게 되었고 그 결과 Key Escrow의 개념이 나타나게 되었다.

Key Escrow란 개인의 비밀키 혹은 비공개키를 신뢰받는기관에게 맡겨 두고 위급한 경우나 어떤 필요한 경우 법집행기관이 법원의영장을 받아서 비밀키나 비공개키를 복구해서 암호문을 복호화해서 평문을 얻어낼 수 있게 하는 것을 말한다. 미국에서는 이에 대한 표준 *Escrowd Encryption Standard(EES)* [3]가 94년에 제정되었다. 그러나 공개되지 않은 암호화 알고리듬을 사용하고 사용자의 비밀키를 두 기관에 나누어 escrow하고 사용자의 비밀키가 복구되면 영구적으로 사용자의 사생활이 노출되는 것에 대한 많은 반대([1]을 참조)가 있었으며 이에 따라 여러가지 다양한 연구[4, 5, 7, 8, 9]가 있어 왔다.

Key Escrow에 대한 이들 연구들은 비밀 알고리듬대신에 공개된 알고리듬을 사용하고 사용자의 비밀키 혹은 비공개키를 두 기관에 나누는 것이 아니라 여러 기관에 나누어 맡기는 방법을 제안했다. 그리고 어떤 필요에 의해 합법적인 도청이나 복호화를 하는 경우에도 사용자의 비밀키나 비공개키가 공개되어 사생활이 영구히 노출되는 것이 아니라 법원의영장이 허용하는 기간만큼만 도청이나 복호화가 가능하고 사용자의 비공개키나 비밀키가 공개되지 않게 하는 방법이 제안되었다. 본 논문에서는 이런 연구들을 이용하여 사용자의 비공개키를 여러 신뢰받는기관에 맡기고 사용자의 비공개키 형성에 키관리센타/신뢰받는기관이 참여하게 하고 비공개키 자체가 공개되는 것을 막는 방법을 공개키 암호 시스템에 적용하려고 한다.

2. 용어 정리

Key Escrow 시스템에서 일반적으로 쓰이는 용어들은 [2]을 참조 하고 여기서는 Key Escrow 시스템을 구성하기 위해 쓰인 수학적인 구조에 대해서 설명한다.

p와 q는 큰 소수이며 $q \mid p-1$이다. 그리고 g는 $\mathbb{Z}_p$의 원소로 위수가 q로 $\mathbb{G}_q$의 생성자이다. $hash(\ )$는 일방향 해쉬 함수로서 충돌 회피성을 갖는다. $E_S(\ )$는 블럭 암호화 함수로 암호화에 쓰이는 키는 S이다. 사용자 A의 비공개키는 $X(A)$이며 공개키는 $Y_A = g^{X(A)} \ mod\ p$ 이다. T는 신뢰받는기관을 나타내고 $x_i \in \mathbb{Z}_q^*, (x_i \neq x_j \text{ for } i \neq j)$ 는 T_i에 대한 공개 정보이다. n은 신뢰받는기관의 전체 수를 나타내며 $n < \mid \mathbb{Z}_q = \mathbb{F} \mid$ 이다. k는 escrow되었던 사용자의 비공개키를 복구할 수 있는 최소한의 신뢰받는기관의 수를 나타낸다. 그리고 $d \in \mathbb{Z}_p$로 날짜 혹은 특정한 기간을 나타낸다.

3. 기존의 Key Escrow 방법

여러가지 Key Escrow 방법들([2, 6]를 참조)중에서 이 논문에서 이용되는 방법을 간략하게 설명하고자 한다.

3.1 Failsafe Key Escrow

Failsafe Key Escrow(FKE)[7]는 shadow public key system을 피하기 위한 것이다. 공개키 암호 시스템에서 사용자들은 비공개키와 공개키를 자신이 정한다. 이 상태에서 비공개키를 escrow하면 신뢰받는기관에 맡겨놓은 비공개키대신에 다른 쌍을 이용하여 암호문을 주고 받을 수 있다. 즉 맡겨 둔 비공개키와 공개키의 쌍 대신에 계산하기 쉽고 잘 알려진 어떤 함수의 입력으로 공개키를 사용하고 그 결과를 법집행기관이 복호화할 수 없는 메세지의 전달에 이용할 수 있다. 이러한 시스템을 shadow public key 시스템이라 하며 계산하기 쉬운 함수와 신뢰받는기관에 맡길 비공개키와 이에 해당하는 공개키, 그리고 이 공개키에서 계산된 또 하나의 공개키와 이에 해당하는 비공개키(shadow secret key)를 만들어내는 효율적인 방법을 찾아야 한다.

FKE는 이러한 shadow public key 시스템을 피하기 위하여 권위있는기관/키관리센타가 사용자와 상호작용을 통해서 사용자의 비공개키를 형성하게 한다. 권위있는기관/키관리센타는 사용자 A에게 $X(MA)$를 commit하고 사용자는 $X(UA)$를 만들어 신뢰받는기관에게 escrow한다. 이 escrow가 끝나면 사용자에게 $X(MA)$를 가르쳐 주고 사용자의 비공개키는 $X(A) = X(MA) + X(UA)$가 된다. 이로써 사용자가 shadow secret key를 만들수 없게 한다. 그리고 사용자가 비공개키를 만들때 좋은 난수 발생기가 없어도 권위있는

기관/키관리센타가 비공개키를 안전하게 선택하게 할 수 있으며 사용자가 권위있는기관/키관리센타를 믿지 못하더라도 자신이 스스로 비공개키에 대한 조절을 할 수 있다.

3.2 Key Escrow with time warrent

사용자의 비공개키가 escrow되고 사용자가 비공개키로 암호문을 만들었고 이것을 합법적인 도청을 해야 한다고 하자. 그러면 사용자의 비공개키를 복구해야 하는데 이렇게 되면 사용자의 사생활은 비공개키를 바꾸기 전까지 노출이 된다. 물론 합법적인 도청 기간이 끝나면 복구된 비공개키는 폐기가 되어야 하지만 이것을 보장할 수는 없다. 그래서 Lenstra등[8]은 암호문의 세션키에 날짜에 대한 정보 즉 time warrent를 줄 수 있는 방법을 제안했다.

사용자들이 암호화된 통신을 원할때는 세션키를 날짜 혹은 기간과 비공개키와 공개키를 해쉬한 것으로 암호화해서 서로 교환한다. 세션키를 암호화 한 것은 통신의 당사자들이 복호화 할 수 없으며 복호화 할 필요도 없다. 세션키를 복구할 때는 법집행기관이 신뢰받는기관에게서 날짜 혹은 기간에 대한 정보가 있는 비공개키의 부분을 받게 해서 사용자의 비공개키가 노출되는 것을 막는다.

4. Integrated Security System(ISS)

공개키 암호 시스템과 스마트 카드를 사용하여 인증, 디지탈 서명, 비밀키 분배, 데이터 암호화의 서비스를 제공하는 시스템을 Integrated Security System(ISS)라고 하자. 이 시스템의 서비스 중에서 본 논문이 관심을 두는 것은 비밀키 분배 서비스이다. 비밀키를 분배한다는 것은 비밀 정보를 교환하고자 하는 두 사용자가 만나지 않고 비밀키를 서로 공유할 수 있게 하는 것을 말한다. 비밀키는 문서 암호화를 위한 대칭키 알고리듬에 이용되며 한 세션만 이용되는 세션키를 의미한다. ISS에서의 비밀키 분배 서비스는 사용자 쌍방 인증과 키 인증기능을 가지며 키가 상대방에게 전달되었음을 확신할 수 있다. 그리고 쌍방이 키 조정능력을 가진다.

- **비밀키 분배 프로토콜**

1. 사용자 A는 스마트 카드를 Card Reader에 꽂고 password를 입력한다.
2. 사용자 A는 난수를 발생한 후 A의 첫번째 Key Token KT_{A1}을 생성한다.

$$K_A \in_r \mathbb{Z}_q,\ T_A = g^{K_A} \bmod p,\ KT_{A1} = T_A \parallel Time_A \parallel A \ldots$$

3. 사용자 A는 Key Token KT_{A1}을 사용자 B에게 전송한다.
4. 사용자 B는 난수를 발생한 후 signed Key Token KT_{B1}를 계산한다.

$$K_B \in_r \mathbb{Z}_q,\ T_B = g^{K_B} \bmod p,\ M_B = T_A \parallel T_B \parallel A \parallel Time_B \ldots$$

$$S_B = X_B^{-1}(K_B - hash(M_B) \cdot T_B) \bmod q,\ KT_{B1} = T_B \parallel S_B \parallel Time_B \ldots$$

5. 사용자 B는 KT_{B1}를 A에게 전송한다.
6. 사용자 A는 B의 공개키를 키관리센타에서 가져와 확인을 한다.

7. M_B를 만들어 B의 공개키로 B의 Key Token을 검증한다.

$$M_B = T_A \parallel T_B \parallel A \parallel Time_B \ldots, Y_B^{S_B} \cdot g^{hash(M_B)\cdot T_B} \bmod p \stackrel{?}{=} T_B$$

8. 사용자 A는 검증된 T_B와 자신의 T_A로 메세지 M_A를 만들고 signed Key Token KT_{A2}를 계산한 후 세션키 K_{AB}를 계산한다.

$$M_A = T_A \parallel T_B \parallel A \parallel Time_{A2} \ldots, S_A = X_A^{-1}(K_A - hash(M_A) \cdot T_A) \bmod q,$$

$$KT_{A2} = S_A \parallel Time_{A2} \parallel \ldots, \; K_{AB} = hash(T_B^{K_A} \bmod p)$$

9. 사용자 A는 KT_{A2}를 B에게 전송한다.

10. 사용자 B는 A의 공개키를 키관리센타에서 가져와 확인을 한다.

11. M_A를 만들어 A의 공개키로 A의 Key Token KT_{A2}를 검증한다.

$$M_A = T_A \parallel T_B \parallel B \parallel Time_{A2} \ldots, \; Y_A^{S_A} \cdot g^{hash(M_A)\cdot T_A} \bmod P \stackrel{?}{=} T_A$$

12. 비밀 공유키 K_{AB}를 계산한다.

$$K_{AB} = hash(T_A^{K_B} \bmod p)$$

5. ISS에서의 Key Escrow

ISS에 Key Escrow를 적용할 때 FKE[7]와 Lenstra등의 방법[8]를 이용한다. 즉 사용자의 비공개키는 사용자가 만든 부분과 권위있는기관/키관리센타가 준 값이나 신뢰받는기관이 준 값에서 사용자가 선택한 부분으로 이루어진다. time warrant는 세션키의 형성에 쓰이는 난수를 기간에 대한 정보를 갖고 있는 것으로 암호화하거나 기간에 대한 정보를 가지고 난수를 만드는 방법으로 하여 합법적인 도청을 하는 경우에도 사용자의 비공개키가 공개되는 일이 없도록 한다.

5.1 사용자의 비공개키를 한 기관과 만드는 경우

사용자가 비공개키를 만들때 권위있는기관/키관리센타 한 곳과 상호작용을 한다. 이 경우 사용자의 비공개키는 $X(A) = X(MA) + X(UA)$이다. 여기서 $X(MA)$는 권위있는기관/키관리센타가 만들어서 갖고 있는 값이며 $X(UA)$는 사용자가 만들어서 신뢰받는기관에 escrow한 값이다. Lenstra등의 방법[8]은 세션키를 암호화해서 보내는 것인데 ISS에서는 서로 세션키에 대한 조절이 가능하므로 이 방법을 그대로 쓸 수는 없다. 세션키를 복구하기 위해서는 난수를 알아내면 되는데 이렇게 하기 위해서 난수를 암호화시켜 보내거나 난수를 만드는 방법을 바꾼다.

5.1.1 Secret Sharing

사용자가 만든 $X(UA)$를 신뢰받는 기관에 escrow할 때 Pedersen의 Verfiable Secret Sharing(VSS)[10]을 이용한다. 이 방법은 n개의 신뢰받는기관중에서 어떤 k개의 신뢰받는기관이 협력하면 사용자의 비공개키

를 복구할 수 있지만 $k-1$개 이하의 신뢰받는기관은 사용자의 비공개키를 복구할 수 없는 방법이다. 그리고 신뢰받는 기관은 자신이 받은 비공개키의 부분이 실제로 사용자의 비공개키의 부분인지 확인할 수 있다.

• **비공개키 분배 방법**

1. $\mathbb{Z}_q$에서 $f(0) = X(UA)$를 만족하는 $f = X(UA) + f_1 x + \ldots + f_{k-1}x^{k-1}$을 선택해서 신뢰받는기관에게 나눠줄 비밀인 $X(UA_i)$ 를 계산한다.

$$X(UA_i) = f(x_i) \text{ for } i = 1, \ldots, n \tag{1}$$

2. $X(UA_i)$를 비밀 채널을 통하여 각각의 T_i에게 보내고 모든 T_i에게 $(g^{X(UA)}, g^{f_1}, \ldots, g^{f_{k-1}})$를 보낸다.

신뢰받는기관에게 비공개키를 escrow하는 사용자는 G_q의 원소 k개를 broadcast해야 하고 $\mathbb{Z}_q$의 원소 n개를 비밀 채널을 통해서 보내야 한다. 비공개키의 분배가 끝나면 신뢰받는기관 T_i들은 자신이 받은 비공개키의 부분들을 다음과 같이 확인한다.

• **신뢰받는기관 T_i에서의 비공개키의 부분에 대한 검증 방법**

1. 모든 $l = 1, \ldots, n$에 대하여 $h_l = \sum_{j=0}^{k-1} (g^{f_j})^{x_l^j}$를 계산한다.
2. $h_i = g^{X(UA_i)}$인지 확인한다.
3. 확인이 되지 않으면 $X(UA_i)$를 공개하고 프로토콜을 그만 두고 확인이 되면 비공개키의 부분 $X(UA_i)$를 받아 들인다.

k개의 $f(x_i) = X(UA_i)$를 만족하는 답아야 $k-1$차인 다항식은 오직 1개밖에 없으므로 위의 VSS에서 k개의 신뢰받는기관 $(T_1, \ldots, T_k)$는 f를 다음의 식으로부터 구할 수 있다.

$$\begin{aligned} f(x) &= \sum_{i=1}^{k} \left(\prod_{h \neq i} \frac{x - x_h}{x_i - x_h}\right) f(x_i) \\ &= \sum_{i=1}^{k} \left(\prod_{h \neq i} \frac{x - x_h}{x_i - x_h}\right) X(UA_i) \end{aligned} \tag{2}$$

따라서 $X(UA)$는 다음의 식에서 구할 수 있다.

$$X(UA) = \sum_{i=1}^{k} a_i X(UA_i), \tag{3}$$

$$a_i = \prod_{h \neq i} \frac{x_h}{x_h - x_i} \tag{4}$$

5.1.2 난수를 암호화시켜 보내는 방법

세션키는 난수 K_A나 K_B를 알면 얻어낼 수 있으므로 난수 K_A나 K_B를 암호화해서 보내고 암호화할때의 비밀키에 time warrant를 준다. 즉 다음의 식에서처럼 A는 $c(A,d)$를 B에게 보내면 된다. 물론 상대방 B는 $c(A,d)$에서 난수를 복구할 수 없다.

$$S(A,d) = hash(d^{X(A)}) \tag{5}$$

$$c(A,d) = E_{S(A,d)}(K_A) \tag{6}$$

필요에 의해 세션키를 복구할 때는 다음의 프로토콜을 통해서 세션키를 복구한다.

• **세션키 복구 방법**

1. 법집행기관은 법원에서 사용자 A에 대한 합법적인 도청을 위한 법원의영장을 받아서 신뢰받는기관과 키관리센타에게 보낸다.
2. 법원의영장을 받은 n개의 신뢰받는기관은 법집행기관에게 $d^{X(UA_i)}$를 보낸다.
3. 키관리센타는 법원의영장을 받은 뒤 법집행기관에게 $d^{X(MA)}$를 보낸다.
4. 법집행기관은 $d^{X(UA)}$를 계산하고 $d^{X(MA)}$를 곱해서 $d^{X(A)}$를 얻는다.

$$X(UA) = \sum_{i=1}^{k} a_i X(UA_i) \tag{7}$$

$$d^{X(UA)} = d^{\sum_{i=1}^{k} a_i X(UA_i)} = \prod_{i=1}^{k} (d^{X(UA_i)})^{a_i} \tag{8}$$

$$d^{X(A)} = d^{X(MA)+X(UA)} = d^{X(MA)} d^{X(UA)} \tag{9}$$

그리고 $d^{X(A)}$를 해쉬해서 얻은 $S(A,d)$로 $c(A,d)$를 복호화해서 K_A를 얻는다.

5. 법집행기관은 T_B와 K_A로 세션키를 복구한다.

$$K_{AB} = hash(T_B^{K_A}) = hash(g^{K_B K_A}) \tag{10}$$

5.1.3 난수를 구조적으로 만드는 방법

난수 K_A를 암호화시켜 보내는 것이 아니라 난수 K_A가 time warrant를 가질 수 있게 한다. 즉 다음과 같이 난수 K_A를 만든다.

$$K_A = hash(d^{X(A)} \| Time_A) \tag{11}$$

$Time_A$에 의해 난수 K_A는 매번 다른 값이 되게 된다. 그리고 앞에서 세션키를 복구하기 위해서 보냈던 암호문 c를 만들어 보낼 필요가 없다. 세션키를 복구하는 방법은 앞의 방법과 거의 똑같다. 앞의 세션키 복구 방법의 단계 4에서 $d^{X(A)}$ 를 얻은 후에 $Time_A$를 붙여서 해쉬를 하면 난수 K_A를 얻을 수 있다.

5.1.4 일방향 통신에 사용하기 위한 방법

사용자 A가 사용자 B에게 이메일이나 팩스를 보낸다고 하는 경우 일반적으로 세션키에 대한 조절을 할 수 없다. 그래서 ISS의 세션키를 수정해서 $hash(Y_B^{K_A})$라고 한다.

K_A를 얻어내기 위해서 앞의 두 가지 방법을 사용할 때 어느 방법이든 사용자 A에 대한 법원의영장을 받아서 합법적인 도청을 하는 경우는 time warrant가 지켜진다. 그러나 사용자 B에 대한 합법적인 도청을 해야 하는 경우 사용자 B의 비공개키 $X(B)$를 알아야 하고 이 경우 time warrant가 깨어진다. 그래서 난수 K_A를 구조적으로 만들고 일방향통신을 시작하는 사람에 대해서 세션키를 복구할 때는 앞의 방

법으로 복구하고 사용자 B처럼 암호화된 문서를 받는 사람에 대해서는 다음의 방법으로 세션키를 복구한다.

• 일방향통신에서 세션키 복구 방법

1. 법집행기관은 법원에서 일방향통신의 문서를 받는 사람인 B에 대한 법원의영장을 받는다.
2. 법집행기관은 $T_A = g^{K_A}$를 신뢰받는기관과 키관리센타에게 보낸다.
3. 법원의 영장과 T_A를 받은 n개의 신뢰받는기관은 $T_A^{X(UB_i)}$를 법집행기관에게 준다.
4. 키관리센타는 법원의 영장과 T_A를 받은 후 $T_A^{X(MB)}$를 법집행기관에게 준다.
5. 법집행기관은 아래의 계산을 통해서 세션키를 복구한다.

$$X(UB) = \sum_{i=1}^{k} a_i X(UB_i) \tag{12}$$

$$T_A^{X(UB)} = T_A^{\sum_{i=1}^{k} a_i X(UB_i)} = \prod_{i=1}^{k} (T_A^{X(UB_i)})^{a_i} \tag{13}$$

$$T_A^{X(B)} = T_A^{X(MB)+X(UB)} = g^{K_A X(B)} \tag{14}$$

$$K_{AB} = hash(T_A^{X(B)}) = hash(Y_B^{K_A}) \tag{15}$$

5.2 사용자의 비공개키를 여러 기관과 만드는 경우

FKE[7]에서는 사용자의 비공개키를 만들때 권위있는기관/키관리센타 한 곳과 같이 만들었다. 그리고 사용자가 만든 비공개키의 부분은 신뢰받는기관과 VSS를 통하여 escrow를 하는데 여기서는 그 방법을 변형하기로 한다. 즉 사용자가 비공개키를 만들때 권위있는기관/키관리센타 한 곳과 하는 것이 아니라 신뢰받는기관이 준 값들 중에서 정해진 수만큼 사용자가 선택하여 비공개키의 일부분으로 하는 것이다. 그렇게 해서 사용자의 비공개키의 부분을 안전하게 보관해야 할 곳은 신뢰받는기관만으로만 하며 사용자의 비공개키를 복구하는것도 신뢰받는기관만이 참여하게 한다.

5.2.1 사용자의 비공개키 형성 방법

비공개키를 만들때 신뢰받는기관에서 받은 부분을 이용하고 사용자의 세션키를 복구해야 할 때는 이들 사용자가 선택한 모든 신뢰받는기관이 협력을 한다. 그런데 모두 협력할 수 없는 상황이나 이 중의 어떤 신뢰받는기관이 세션키 복구에 대한 거부를 방지하기 위하여 신뢰받는기관이 준 부분을 사용자가 신뢰받는기관에 escrow한 키($X(UA)$)로 암호화시켜서 키관리센타가 보관하게 한다.

• 사용자의 비공개키 형성 방법

1. 사용자는 Pedersen의 VSS[10]과 같이 f를 정하고 $s = X(UA)$라 한다. 그리고 신뢰받는기관에게 다음을 broadcast한다.

$$(g^{X(UA)}, g^{f_1}, \ldots, g^{f_{k-1}}) \tag{16}$$

2. 신뢰받는기관은 $X(TA_i) \in \mathbb{Z}_q$를 선택해서 암호화하고 서명한 $DS(X(TA_i))$를 사용자에게 준다.

$$S(T_i, A) = hash(g^{X(UA)X(T_i)}) \quad (17)$$

$$DS(X(TA_i)) = Sign(E_{S(T_i,A)}(X(TA_i), g^{X(TA_i)}) \quad (18)$$

3. 사용자는 서명을 검증한 뒤 신뢰받는기관에게 다음을 비밀 채널을 통하여 보낸다.

$$X(UA_i) = f(x_i) \text{ for } i = 1, \ldots, n \quad (19)$$

4. 신뢰받는기관은 받은 비공개키의 부분에 대한 검증을 한 뒤 검증이 되었음을 키관리센타나 사용자에게 알려 준다.

5. 사용자는 신뢰받는기관이 보내준 $X(TA_i)$중에서 t개를 선택해서 비공개키의 일부분으로 한다.

$$X(TA) = \sum_{i=1}^{t} X(TA_i) \quad (20)$$

$$X(A) = X(UA) + X(TA) \quad (21)$$

6. 사용자는 선택한 신뢰받는기관이 보내준 서명과 공개키와 $g^{X(UA)}$를 키관리센타에 보낸다.

7. 키관리센타는 사용자에게서 받은 공개키를 검증하고 보관한다.

$$Y_A \stackrel{?}{=} g^{X(UA)} \prod_{i=1}^{t} g^{X(TA_i)} \quad (22)$$

5.2.2 세션키 복구 방법

난수 K_A를 얻기 위한 방법은 앞의 두 가지 방법을 모두 사용해도 되며 복구할 때 키관리센타에서 $d^{X(MA)}$를 받는 대신에 사용자가 선택한 신뢰받는기관에게서 $d^{X(TA_i)}$를 받으면 된다.

6. 결론

이 논문에서는 스마트 카드를 사용하는 공개키 암호 시스템에서 Key Escrow를 적용시켰다. 사용자가 비공개키를 결정할 때 권위있는기관/키관리센타와 상호작용을 하거나 사용자의 비공개키를 escrow하는 신뢰받는기관과 상호작용을 한다. 그리고 time warrant를 가져서 사용자가 세션키를 만드는 프로토콜을 따른다면 법집행기관의 합법적인 도청/복호화가 필요한 경우에도 사용자의 비공개키의 노출은 없다.

감사의 글

본 논문에 도움을 준 이 은정 연구원에게 감사를 표합니다.

참 고 문 헌

[1] 한 상근, 이 영, "미국의 암호정책에 대한 연구-클리퍼 칩을 중심으로," *통신정보보호학회지*, vol. 4, no. 4, pp. 35-61, 1995.

[2] 정 경임, 이 필중, "Public Key Escrow Systems의 소개와 분석," *통신정보보호학회지*, vol. 6, no. 2, pp. 35-52, 1996.

[3] National Institute of Standard and Technology, "Escrowed Encryption Standard," *Federal Information Processing Standard Publication 185*, February, 1994.

[4] , S.Micali, "Fair Cryptosystems," *MIT/LCS/TR-579.b*, Novermber 1993.

[5] , S.Micali, "Guaranteed Partial Key Escrow," *MIT/LCS/TM-537*, August 1995.

[6] D.Denning, "A Taxonomy for Key Escrow Systems," *Communications of the ACM*, vol. 39, no. 3, pp. 34-30, March 1996.

[7] J.Kilian, T.Leighton, "Fair Cryptosystem revisited," *Advances in Cryptology-Crypto'95*, Lecture Notes in Computer Science(LNCS) vol. 963, pp. 208-221, D. Coppersmith ed., Springer-Verlag, 1995.

[8] A.Lenstra, P.Winkler, Y.Yacobi, "A Key Escrow System with Warrant Bounds," *Advances in Cryptology-Crypto'95*, Lecture Notes in Computer Science(LNCS) vol. 963, pp. 197-207, D. Coppersmith ed., Springer-Verlag, 1995.

[9] M.Bellare, S.Goldwasser, "Verifiable Partial Key Escrow," *TR-CS95-447*, Department of Computer Science and Engineering, University of California at San Diego, October 1995.

[10] T. P. Pedersen, "Distributed provers with application to undeniable signatures," *Advances in Cryptology-Eurocrypt'91*, Lecture Notes in Computer Science(LNCS) vol. 963, pp. 222-242, D. Coppersmith ed., Springer-Verlag, 1991.