2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

일방향 키 분배가 가능한 키 복구 시스템

유준석*, 최승복*, 손기욱*, 원동호*
*성균관대학교 전기 전자 및 컴퓨터 공학부
e-mail : jsyu@dosan.skku.ac.kr

A Key Recovery System with One-way Key Distribution Property

Joon-Suk Yu*, Seung-Bok Choi*, Ki-Wook Sohn*, Dong-Ho Won*
*School of Electrical & Computer Eng., Sungkyunkwan University

요약

최근들어 암호기술은 급속도로 확산되고 있으며, 암호 사용으로 인한 부작용을 방지하기 위한 대안으로 키 복구에 대한 연구가 활발히 진행되고 있다. 그러나 지금까지 제시된 기존의 키 복구 기술들은 그 대부분이 정부의 법 집행권 보장에만 설계 초점을 맞추고 있기 때문에 다양한 사용자들의 요구를 충족시키기 어렵다. 본 논문에서는 키 복구 시스템에 대한 사용주체들의 입장을 고려하여 다양한 환경에 적용할 수 있는 키 복구 시스템을 제안한다. 제안하는 방식은 암호통신 과정 중에 키가 분배되는 일방향 키 분배가 가능하고 기존 키 복구 시스템 만큼 효율적이면서도 충분한 유연성을 제공한다.

1. 서론

암호의 사용은 정보의 기밀성 및 무결성을 보장해 주며, 인증 기능 등을 제공해 줌으로써 상거래나 결제 등과 같은 현실세계의 일들이 전자적으로 실현될 수 있게하여 일반 사용자들에게 많은 편리함과 이점을 제공한다. 그러나 암호기술이 범죄자에 의해 악용될 경우에는 사회의 안전을 위협할 수 있으며, 암호키가 손상되거나 분실되었을 경우에는 정당한 사용자라도 암호문을 복호할 수 없는 등의 문제점이 존재한다.

키 복구(Key Recovery)는 이와 같은 암호 사용에 따른 부작용을 해결하기 위한 효과적인 대안으로써 주목받고 있는 기술이다. 키 복구는 그 방식에 따라서 크게 하나 이상의 신뢰되는 기관에 위탁된 키 복구 정보로부터 키를 복구해내는 키 위탁(Key Escrow) 방식과 신뢰되는 기관만이 복호할 수 있는 암호화 영역에 포함된 키 복구 정보로부터 키를 복구해내는 키 캡슐화(Key Encapsulation) 방식으로 나누며[1], 경우에 따라 TTP(Trusted Third Party) 방식을 추가하여 분류하기도 한다.

이러한 여러 가지 키 복구 방식 중 법 집행권 보장을 목적으로 하는 키 복구 시스템들은 유사시에 신뢰기관에 보관된 정보들로부터 확실한 키 복구가 가능한 키 위탁 방식이나 TTP 방식을 주로 이용하며[2, 3, 4], 기업이나 일반 사용자 입장에서의 키 복구를 목적으로 하는 시스템들의 경우에는 저장 데이터의 복구가 용이하고 개인의 프라이버시 보호에 있어서 우수한 특징을 가지는 키 캡슐화 방식이 주류를 이루고 있다[5, 6]. 그러나 이러한 시스템들은 여러 사용주체들의 키 복구에 대한 다양한 요구사항을 수용할 만큼 유연하지 않으며, 모든 요구사항을 완벽히 만족하는 키 복구 시스템을 설계하는 것은 기술적·정책적 이유 때문에 어려운 것이 사실이다.

본 논문에서는 이러한 점들을 고려하여 다양한 환경에 사용될 수 있는 유연하면서 효율적인 키 복구 시스템을 제안한다.

2. 키 복구 시스템 요구사항

본 장에서는 키 복구 시스템을 이용하는 사용 주체들의 상이한 목적이나 대상을 고려하여 키 복구 시스템이 만족해야 할 사항들에 대해서 포괄적으로 설명한다.

- 충분한 강도 : 키 복구 기능의 추가로 인해 기존 암호 시스템의 안전성이 저하되어서는 안된다.
- 확실한 키 복구 : 정당한 키 복구 요구에 대해 키 복구가 가능해야 한다.
- 집행권 제한 : 국가의 법 집행권 보장을 목적으로 하는 키 복구 시스템은 사용자의 프라이버시 침해를 막을 수 있는 기능을 제공해야 한다.
- 어려운 우회 : 키 복구 시스템을 사용하는 사용자

들이 부당하게 키 복구 기능을 우회하면서 암호 통신을 할 수 없어야 한다.

- 키 복구 대행기관 선택의 융통성 : 키 복구 시스템은 사용자가 선택한 다수의 키 복구 대행기관을 통하여 키 복구를 수행할 수 있어야 한다.
- 시스템 세부사항 공개 : 시스템에 사용되는 알고리즘 및 시스템의 구체적인 동작과정 등이 공개되어 검증될 수 있어야 한다.
- 비용 및 성능 : 키 복구 기능이 암호 시스템에 추가될 때에는 기존 암호 시스템의 효율 저하가 최소가 되도록 하여야 하며, 추가되는 비용이 암호화된 정보의 가치를 넘지 않아야 한다.

3. 제안하는 키 복구 시스템

제안하는 시스템의 구성 요소 및 파라미터는 다음과 같고 모든 연산은 모듈러 p상에서 이루어진다. 또한 본 논문에서는 편의상 각 사용자가 두 개의 KRA를 사용한다고 가정한다.

[구성 요소 및 용어]

- 암호 단말 시스템(CES) : 데이터의 암호화 및 복호화를 수행하는 하드웨어 또는 소프트웨어로써 송신측에서는 데이터 복구 필드(Data Recovery Field : DRF)를 생성된 암호문에 덧붙이는 역할을 수행하며, 수신측에서는 필요에 따라 DRF의 유효성 검사한다.
- 키 복구 대행기관(KRA) : 사용자의 암호화된 데이터나 키를 복구하는데에 필요한 정보를 안전하게 보관하고 있으며, 키 복구 요청자의 정당한 키 복구 요청에 의해 키 복구를 수행한다.
- 키 복구 요청자(KRR) : 키 복구 요청자는 법 집행 또는 사용자의 키 복구 필요에 의해 암호화된 데이터의 복구를 키 복구 대행기관에 요청할 수 있는 권한을 가진 허가된 개체이다.

[기호 및 파라미터]

- p : 2q+1 형태의 큰 소수(q는 충분히 큰 소수)
- g : Z_p^*상의 생성자
- x_A : 사용자 A의 비밀키
- y_A : 사용자 A의 공개키
- KT_{A_i} : 각 KRA의 비밀값($1 \le i \le$ A의 KRA 수)
- h_1, h_2 : 일방향 해쉬함수

이 때 해쉬함수는 비밀키 k를 입력으로 가지며, 다음과 같은 특성을 지닌다.

- 계산 불가성(Computation-resistance) : 하나 이상의 입력과 해쉬값의 쌍 (m_i, H_i)가 주어졌을 때, 비밀키 k를 모르고 어떤 입력 m ($\neq m_i$)에 대한 해쉬값 H를 구하는 것과 $h(k, m_i)=h(k, m)$인 m ($\neq m_i$)을 찾는 것이 계산상 불가능하다[7].

3.1 사용자 등록 단계

사용자와 KRA 사이에 공유되는 비밀정보를 설정하는 단계로 다음과 같이 수행된다.

① 사용자 A는 다음과 같은 비밀키, 공개키 쌍 ($x_A \in_R Z_p^*$, $y_A = g^{x_A}$)을 생성하며, 공개키 y_A를 자신이 선택한 KRA들에게 전송한다.

② 사용자 A의 공개키를 전송받은 각 KRA들은 KT_{A_i} 값을 랜덤하게 선택하고 다음의 값들을 계산하여 $cert_{A_i}$, r_{A_i}, g^{a_i}를 사용자 A에게 전송한다.

$$a_i = h_1(KT_{A_i}, y_A), \quad \alpha_{A_i} = y_A^{a_i}, \quad r_{A_i} = g^{\alpha_{A_i}}$$

$$cert_{A_i} = Sig(y_A, r_{A_i})$$

③ 사용자 A는 각 KRA들이 전송한 정보의 유효성을 확인하기 위해 다음을 계산한다.

$$\alpha_{A_i} = (g^{a_i})^{x_A}, \quad r_{A_i} = g^{\alpha_{A_i}}$$

④ 사용자 A는 단계 ③에서 계산된 r_{A_i}와 $cert_{A_i}$에 포함된 r_{A_i}의 일치 여부를 검사함으로써 KRA로부터 전송된 정보의 유효성을 확인하며, 그 결과에 따라 각 KRA에게 Accept 또는 Reject 신호를 전송한다.

⑤ 각 KRA들은 사용자 A로부터 Accept를 수신하면 $cert_{A_i}$를 공개하고, Reject를 수신하면 프로토콜을 종료한다.

3.2 암호통신 단계

두 사용자 A와 B가 사용할 세션키 KS는 사전에 분배되어 있다고 가정하며, 구체적인 통신과정은 다음과 같다.

① 사용자 A가 사용자 B와 처음 통신을 하는 경우에 사용자 A는 공개된 사용자 B의 r_{B_1}, r_{B_2} 값과 자신의 α_{A_1}, α_{A_2} 값을 이용하여 다음을 계산한다.

$$\omega_1 = r_{B_1}^{\alpha_{A_1}}, \quad \omega_2 = r_{B_2}^{\alpha_{A_2}}$$

② 사용자 A는 다음과 같이 KEK(Key Encryption Key)를 생성한다. (단, SID(Session ID)는 임의의 세션 식별자로써 암호화하려는 파일 또는 메시지 마다 유일한 값이다.)

$$KEK_1 = h_2(\omega_1, SID), \quad KEK_2 = h_2(\omega_2, SID)$$

$$KEK = KEK_1 \oplus KEK_2$$

③ 사용자 A는 이미 설정된 세션키 KS를 KEK로 암호화하여 ESK(Encrypted Session Key)를 생

성한다.

$$ESK = E_{KEK}(KS)$$

④ 사용자 A는 파일 또는 메시지 M을 세션키 KS로 암호화하여 암호문을 생성한다.

$$C = E_{KS}(M)$$

⑤ 사용자 A는 다음과 같은 형태의 DRF를 암호문 C에 덧붙여서 사용자 B에게 전송한다.

$$DRF = ESK \| SID \| cert_{A_1} \| cert_{A_2} \| cert_{B_1} \| cert_{B_2}$$

이상은 사용자 A의 암호 단말 시스템에 의해 수행되며, 암호문을 수신한 사용자 B는 세션키가 KRA에 의해 복구될 수 있음을 확인하기 위해 DRF의 유효성을 다음과 같이 검사한다.

⑥ 사용자 B가 사용자 A와 처음 통신을 하는 경우에는 다음과 같이 ω_1과 ω_2를 계산한다.

$$\omega_1 = r_{A_1}^{\alpha_{B_1}}, \quad \omega_2 = r_{A_2}^{\alpha_{B_2}}$$

⑦ 사용자 B는 계산된 ω_1, ω_2로부터 다음과 같이 KEK를 계산한다.

$$KEK_1 = h_2(\omega_1, SID), \quad KEK_2 = h_2(\omega_2, SID)$$

$$KEK = KEK_1 \oplus KEK_2$$

⑧ 사용자 B는 다음을 검사함으로써 사용자 A로부터 전송된 정보의 유효성을 확인한다.

$$ESK \stackrel{?}{=} E_{KS}(KEK)$$

⑨ DRF 유효성 검사를 통과한 경우에만 수신자 B는 사전에 분배된 세션키 KS를 사용하여 다음과 같이 암호문 C를 복호한다.

$$M = D_{KS}(C)$$

3.3 키 복구 단계

정당한 키 복구 요청자는 암호문을 취득하여 다음과 같이 키 복구를 수행한다.

① 키 복구 요청자는 복호하려는 암호문에서 추출한 DRF를 해당 KRA들로 전송한다.

② 키 복구 요청을 받은 각 KRA들은 다음과 같이 ω_i 값을 계산한다.

$$a_i = h_1(KT_{A_i}, y_A), \quad \alpha_{A_i} = y_A^{a_i}, \quad \omega_i = y_{B_i}^{\alpha_i}$$

③ 각 KRA들은 계산된 ω_i 값을 통해 다음과 같이 KEK_i를 계산하여 키 복구 요청자에게 전송한다.

$$KEK_i = h_2(\omega_i, SID)$$

④ 각 KRA들로부터 KEK를 전송받은 키 복구 요청자는 다음과 같이 얻어진 KEK로부터 세션키 KS를 복구한다.

$$KEK = KEK_1 \oplus KEK_2, \quad KS = D_{KEK}(ESK)$$

4. 제안 시스템의 특징 및 안전성

4.1 제안 시스템의 특징

제안하는 시스템은 다음의 주요 특징을 가진다.

[일방향 키 분배]

일방향 키 분배는 다음과 같이 제안 시스템에서 KEK를 세션키 KS로 사용함으로써 구성할 수 있으며, 이 때 DRF의 유효성을 확인하기 위해 사용되었던 ESK는 필요하지 않다.

$$KEK_1 = h_2(\omega_1, SID), \quad KEK_2 = h_2(\omega_2, SID)$$

$$KS = KEK_1 \oplus KEK_2$$

$$DRF = SID \| cert_{A_1} \| cert_{A_2} \| cert_{B_1} \| cert_{B_2}$$

$$C = E_{KS}(M)$$

수신자는 DRF 검증 과정의 KEK를 구하는 방법으로 세션키 KS를 계산하여 암호문을 복호할 수 있다. 이와 같이 일방향 키 분배를 사용할 경우에 송신자가 KRA에 의한 키 복구를 우회할 목적으로 부당한 DRF를 암호문에 첨부한다면 수신자 또한 올바른 키를 구할 수 없게되므로 암호통신을 수행할 수 없게된다.

[저장 데이터의 복구]

사용자 A는 자신의 r_{A_i}와 α_{A_i}를 이용하여 ω_i 값을 계산하며, 제안 시스템의 KEK를 세션키 KS로 사용하여 저장 데이터에 대한 키 복구를 수행 할 수 있다. 이 경우에는 DRF의 검증 과정과 수신자가 없으므로 DRF에 ESK와 $cert_{B_i}$를 포함할 필요가 없다.

[세션키의 복구]

제안하는 시스템에서 복구되는 키는 각 세션 정보들을 통해 얻어지는 사용자의 세션키이므로 집행권의 제한이 가능하다.

[KRA 선택의 융통성]

제안하는 시스템에서 각 사용자들은 자신이 속해있는 조직의 정책에 따라 하나 이상의 KRA를 선택할 수 있으며, 통신하는 각 사용자는 서로 다른 수의 KRA를 사용할 수 있다.

[소프트웨어 구현]

제안하는 시스템은 공개키 암호를 기반으로 구성하므로 소프트웨어로 구현될 수 있고 이는 시스템의 비용이나 융통성, 조작성 등에서 하드웨어로 구현된 시스템보다 우수하다는 장점을 지닌다.

4.2 효율성

본 절에서는 각 수행 단계별로 시스템의 효율성에 관해서 살펴본다.

[사용자 등록 단계]

사용자 등록은 키 복구 시스템을 사용하기 위해 단 한 번만 수행되며, 이는 전체 키 복구 시스템에서의 오버헤드를 고려할 때 크지 않다.

[암호통신 단계]

가장 빈번히 수행되는 단계로써 처음 통신을 하는 사용자들 사이에서 지수연산을 필요로 하므로 전체 키 복구 시스템의 효율성에 가장 큰 영향을 미친다. 필요한 지수 연산의 최대 횟수는 통신자가 선택한 전체 KRA의 수와 같지만 한 번 계산된 ω_i 값을 이후 동일한 사용자와의 통신을 위해 저장함으로써 지수연산에 따른 오버헤드를 줄일 수 있다.

[키 복구 단계]

정당한 키 복구 요청이 있을 경우에만 수행되는 단계로써 각 KRA들은 ω_i 값을 계산하는데 지수연산을 수행한다. 이 단계에서 필요한 지수연산의 횟수도 한 번 계산된 ω_i 값을 키 복구 요청자의 정당한 키 복구 요청기간 동안 저장함으로써 줄일 수 있다. 또한 저장 데이터의 경우에는 사용자가 ω_i 값을 저장한다면 키 복구 수행시에 사용자가 KRA와 연결되어야 하는 부담을 줄일 수 있다.

4.3 안전성

[파라미터의 선택]

시스템 파라미터 p는 전체 시스템의 안전성에 영향을 미치므로 다음과 같은 형태를 지닌 적절한 소수 p를 선택한다. 이러한 파라미터의 설정은 [8]에서 설명된 이산대수를 계산하는 공격을 어렵게 한다.

$p=2q+1$ (단, p와 q는 큰 소수)

[공격 시나리오]

<표 1> 제안 시스템에 대한 공격

주어진 정보	구하는 값	안전성 기반
r_A, g	α_A	이산대수 문제
y_A, g^a	α_A	Diffie-Hellman 문제
α_B, r_A, ω	α_A	Diffie-Hellman 문제
r_A, r_B	ω	Diffie-Hellman 문제
KEK=h(ω, SID)인 KEK와 SID	ω	해쉬함수의 계산 불가성

제안한 시스템에서 공격자는 공개된 정보들로부터 α와 ω를 구하려는 시도를 할 수 있을 것이며, 이러한 공격들은 <표 1>에서 나타난 안전성 기반이 되는 문제들이 안전하다면 성공할 수 없다.

5. 결론

본 논문에서는 사용 주체에 따른 키 복구 시스템에 대한 입장을 고려한 요구사항을 살펴보고 이러한 요구를 만족할 수 있는 새로운 키 복구 시스템을 제안하였다. 제안한 시스템은 사전 키 분배가 필요없는 시스템으로도 사용될 수 있고 사용 주체들의 다양한 응용에 적용될 수 있도록 충분한 유연성을 가지면서도 효율적이다.

참고문헌

[1] NIST, "Requirements for key recovery products", Report of the Technical Advisory Committee to Develop a Federal Informaion Processing Standard for the Federal Key Management Infrastructure, 1998.

[2] NIST, "Escrowed Encryption Standard", Federal Information Processing Standards Publication 185, 1994.

[3] David M. Balenson, Carl M. Ellison, Steven B. Lipner and Stephen T. Walker, "A New Approach to Software Key Escrow Encryption", Building in Big Brother : The Cryptographic Policy Debate, Springer-Verlag, 1995.

[4] Ross Anderson and Michael Roe, "The GCHQ Protocol and its Problems", Eurocrypt'97, 1997.

[5] Stephen T. Walker, Steven B. Lipner, Carl M. Ellison and David M. Balenson, "Commercial key recovery", Communications of the ACM, Vol. 39, No. 3, 1996.

[6] David Paul Maher, "Crypto Backup and Key Escrow", Communications of the ACM, Vol. 39, No. 3, 1996.

[7] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[8] S. C. Pohig and M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", IEEE Transaction on Information Theory, Vol. IT 24, No. 1, 1978.