情報保護學會論文誌
第11券 第3號, 2001. 6

키 복구 기능을 가지는 키 공유 프로토콜의 안전성에 관한 연구

김 대 호*, 박 상 우*, 이 동 훈**

On the Security of Key Recovery enhanced Key Exchange Protocol

Dae-Ho Kim*, Sangwoo park*, Dong-hoon Lee**

요 약

본 논문에서는 키 복구 기능을 가지는 키 공유 프로토콜의 안전성을 검토한다. 1999년 ICISC'99에서 P. Paillier와 M. Yung이 제안한 자기 위탁 공개키 시스템 [2] 중에서 Diffie-Hellman형 자기 위탁 키 공유 시스템에 대한 메시지 은닉 공격을 제안하고, 그 대책을 제시한다. 또한, 2001년 PKC 2001에서, 사용자와 서비스 제공자간의 모의에 의한 공격 가능성을 배제하고 제안한 키 복구 기능을 가지는 키 공유 프로토콜 [3]에 대해서, 사용자 모의에 의한 공격 가능성을 배제하지 않을 경우에는 서비스 제공자의 키 복구 기관을 무력화하는 공격 방법이 존재함을 보이고, 사용자 모의에 의한 공격에 대해서도 안전한 키 복구 방식 설계의 필요성을 지적한다.

ABSTRACT

In this paper, we study the security of key recovery enhanced key exchange protocol. We present a subliminal channel of self-escrowed Diffie-Hellman key exchange protocol proposed by P. Paillier and M. Yung in ICISC'99, and also we present a method to prevent such a subliminal channel. In addition, we review and analyze the weakness of the modified key recovery enhanced key exchange protocol proposed by C. Kim and P. Lee in PKC 2001.

keyword : *키 복구(Key Recovery), 키 공유 방식(Key Exchange Algorithm)*

Ⅰ. 서 론

암호 기술의 사용은 중요 정보 누출 방지 등의 많은 장점을 가지는 반면, 잘못 사용될 경우에는 큰 피해를 야기할 수 있다. 우선, 적법한 사용자가 암호문의 복호에 필요한 키를 분실할 경우에는 암호문을 복호화할 수 없게 되어 중요 정보가 손실될 수 있다. 다음으로, 암호가 오용됨으로써 발생할 수 있는 잠재적 위협이 있는데, 악의의 사용자가 중요 정보를 암호화하고 키를 담보로 금품을 요구할 수도 있으며, 키의 도난이나 손상 등의 위협이 항상 존재한다.

키 복구 시스템이란 이러한 암호 기술을 사용할 때의 문제점을 해결하기 위한 것으로 암호문의 소유자가 아닐지라도 사전에 약속된 어떤 특정한 조건하에서 허가된 사람에게 복호가 가능한 능력을 제공하는 암호 시스템으로 정의된다.

키 복구 개념은 1993년 4월 16일에 클린턴 행정

* 국가보안기술연구소({dhokim,psw}@etri.re.kr
** 고려대학교 정보보호대학원 부교수

부가 민간의 암호 제품 이용을 허용하는 동시에 국가 안보와 치안에서 발생할 수 있는 상충 관계를 해결하고자 하는 목적으로 합법적인 도청을 수행하는 법률 집행 기관의 업무를 지원하면서도 개인의 사생활 보호를 위해 키 위탁 방식을 채택하는 것을 주요 내용으로 하는 Clipper 정책 발표에서 시작되었다. Clipper 키 위탁 방식[1]의 제안 이후 하드웨어 및 소프트웨어 환경 등의 다양한 환경을 대상으로 하는 많은 키 복구 방식이 제안되었으며, 그 안전성이 분석되어 왔다. 본 논문에서는 제안된 많은 키 복구 방식들 중 키 복구 기능을 가지는 키 공유 프로토콜의 안전성을 검토하기로 한다. 검토 대상 키 복구 방식은 1999년 P. Paillier와 M. Yung이 제안한 자기 위탁 공개키 시스템[2]과 2001년에 C. Kim과 P. Lee가 제안한 키 복구 기능을 가지는 키 공유 프로토콜[3]이다. 자기 위탁 공개키 시스템에 대해서는 Diffie-Hellman형 자기 위탁 키 공유 시스템에 대하여 메시지 은닉 공격을 제안하고, 그 대책을 제시한다. 제시하는 대책은 P. Paillier와 M. Yung이 제안한 자기 위탁 공개키 시스템의 기본 개념인 공개키 기반 구조를 부가 정보의 사용 없이 키 복구에 활용한다는 것을 만족한다. C. Kim과 P. Lee가 제안한 키 복구 기능을 가지는 키 공유 프로토콜은 사용자와 서비스 제공자간의 모의에 의한 공격 가능성을 배제하고 제안된 것으로, 이러한 가정을 할 수 없을 경우에는 서비스 제공자의 키 복구 기관을 무력화하는 공격 방법이 존재함을 기술하고, 개선이 필요함을 지적한다.

II. 자기 위탁 공개키 시스템 안전성 분석

1998년 A. Young 과 M. Yung[4]에 의해 제안된 자기 위탁 공개키 시스템은 소프트웨어 기반의 키 위탁 시스템이며, 사용자는 정상적인 암호 통신을 하고, 암호 통신 과정에 키 복구를 위한 특별한 부가 정보 없이도 키 위탁 기능을 제공하는 시스템이다. 자기 위탁 공개키 시스템에서는 시스템 관리자(또는 키 복구 기관)만이 위탁된 키를 복구할 수 있으며, 추가적인 정보 없이 일반 사용자가 원래의 키를 복원하는 어려움은 두 수를 인수 분해하는 어려움과 동일하다. 또한, A. Young과 M. Yung은 제안 시스템을 확장하여 초기화가 빠르고, 위탁 구조를 갖는 시스템을 발표하였다[5].

본 장에서는 먼저 Paillier[6]가 제안한 공개키 암호 시스템을 소개하고, Paillier 공개키 암호 시스템을 기반으로 P. Paillier와 M. Yung[2]이 제안한 자기 위탁 공개키 시스템인 Diffie-Hellman형 자기 위탁 키 공유 시스템과 ElGamal형 자기 위탁 공개키 암호 시스템을 소개한다. 다음으로 Diffie-Hellman형 키 공유 시스템의 안전성을 검토하고, 대책을 제시한다.

2.1 자기 위탁 시스템

자기 위탁 시스템(self-escrowed system)은 다음으로 정의된다.

정의 1. 키 복구 기관의 암호화 시스템 $\Sigma=\langle G, E, D\rangle$가 존재하고, 사용자 암호화 시스템 $S=\langle G, E, D\rangle$가 존재하여 키 복구 기관의 비밀키, 공개키 쌍 (X, Y)와 사용자의 비밀키, 공개키 쌍 (x, y)에 대하여

$$y=E_Y(x)$$

인 관계식이 성립할 때 이 암호 시스템을 자기 위탁 시스템(self-escrowed system)이라 한다. 한편 Σ는 master encryption system이라 한다.

2.2 Paillier의 공개키 암호 시스템

1999년 ICISC에서 제안된 P. Paillier와 M. Yung의 자기 위탁 공개키 시스템을 소개하기에 앞서 이 시스템의 이론적 배경이 되는 Paillier 공개키 암호 시스템[6]을 소개한다. Paillier 공개키 암호 시스템은 집합

$$U_n=\{u<n^2 | u=1 \mod n\}$$

에 대하여 다음의 함수를 필요로 한다.

$$L(u)=\frac{u-1}{n}$$

이 함수를 사용한 공개키 시스템의 구성은 다음과 같다.

1. 키 생성 과정

- 공개키 : $n=p\cdot q$ (p,q:소수), $g\in Z^*_{n^2}$
- 비밀키 : $\lambda = \lambda(n) = lcm(p-1,q-1)$

2. 암호화 과정: 임의의 $r \in [0,2^l]$을 선택하여 암호문 $c=g^{m+n\cdot r} \bmod n^2$를 생성한다. 이 때 l은 n의 비트 길이이다.

3. 복호화 과정

$$m=\frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)} \bmod n$$

Paillier 공개키 암호 시스템의 복호화 과정은 함수 $L(\cdot)$의 성질에 의해 성립하며, 또한 함수 $L(\cdot)$은 본 고에서 다루게 될 키 위탁 시스템의 중요한 도구로 사용된다.
위의 암호 시스템의 어려움은 밑수를 g로 갖는 부분 이산 대수 문제와 동치이며, m이 큰 수일 경우에는 실제적인 공격이 불가능하다. 또한 원래의 시스템은 난수 r에 대한 확률론적 암호 시스템이지만[2]에서는 난수 r을 제거하고 결정론적 시스템으로 바꾸었다. 이로 인해 참고문헌[6]에서 다룬 암호 시스템이 동일한 평문에 대해서도 다른 암호문이 생성되는데 반해, 참고문헌[2]에서 다룬 암호 시스템은 동일한 평문에 대한 암호문이 동일하게 결정이 된다는 차이점을 발생시키나 암호 시스템에 대한 공격의 본질적인 어려움은 동일하다.

2.3 Diffie-Hellman형 자기 위탁 키 공유 시스템

Diffie-Hellman 키 공유 시스템[7]은 이산 대수 문제의 어려움에 기반한 키 공유 시스템으로 현재 키 공유 시스템의 대부분이 이 방식을 사용할 정도로 일반화되어 있는 시스템이다. Paillier 공개키 암호 시스템을 사용하여 교환되는 키를 키 위탁 기관이 복구해 내는 Diffie-Hellman형 자기 위탁 키 공유 시스템의 동작 과정은 다음과 같다.

1. 키 복구 기관 구성: 키 복구 기관은 $n=pq$를 계산한 후, n을 공개한다.

2. 키 교환 프로토콜
- A는 난수 $a<n$을 선택하여 $g^a \bmod n^2$을 B에게 보낸다.
- B는 난수 $b<n$을 선택하여 $g^b \bmod n^2$을 A에게 보낸다.
- 각 사용자는 공유키 $K=g^{ab} \bmod n^2$을 공유한다.

3. 키 복구 과정
- 키 복구 기관은 $\lambda(n)$을 계산하여

$$a=\frac{L(g^{a\lambda})}{L(g^{\lambda})} \bmod n$$

를 구하고, 공유키 $K=(g^b)^a \bmod n^2$을 복구한다.

위의 시스템에서도 공격의 어려움은 $Z^*_{n^2}$상에서의 이산 대수 문제와 동일하다. 이 때 n은 앞의 시스템과 마찬가지로 충분히 큰 소수 p와 q의 곱이며, 키 복구 기관의 비밀키는 $\lambda=lcm(p-1,q-1)$이므로 키 복구기관에 대한 공격의 어려움은 큰 수 n에 대한 인수 분해와 같으며 n을 충분히 크게 잡으면 현실적인 공격은 불가능하다.

2.3 ElGamal형 자기 위탁 공개키 암호 시스템

위 소절에서 다룬 것과 유사한 방법으로 ElGamal형의 자기 위탁 공개키 암호 시스템을 구성할 수 있다. 기본적인 동작 과정은 Diffie-Hellman형 자기 위탁 키 공유 시스템과 유사하다. ElGamal형 자기 위탁 공개키 암호 시스템의 상세 동작 과정은 다음과 같다.

1. 키 복구 기관 공개키: $n,g,l=2|n|$

2. 키 복구 기관 비밀키: $\lambda=lcm(p-1,q-1)$

3. 사용자 공개키: $y=g^x \bmod n^2$, $x<n$인 난수

4. 사용자 비밀키: $x(<n)$

5. 암호화 과정: 평문 $m<n^2$에 대하여

$$c=(my^k, g^k)$$

를 계산한다. 여기서 k는 2^l보다 작은 임의의 정수이다.

6. 복호화 과정

$$m=\frac{my^k}{(g^k)^x} \bmod n^2$$

7. 키 복구 과정: 다음의 식을 이용하여 키 복구 기관은 사용자 비밀키 x를 복구할 수 있다.

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

이 시스템은 앞에서 언급한 바와 같이 Pailliar 공개키 암호 시스템과는 달리 사용자에 대해서 결정론적 암호 시스템이다. 이는 동일한 사용자 공개키에 대해서 동일한 사용자 비밀키를 가지고 있음을 뜻한다. 이는 Pailliar 공개키 암호 시스템과 같이 확률론적 시스템으로 바꿀 경우에 키 복구 기관이 메시지 또는 사용자 비밀키를 정확하게 복호화 할 수 없기 때문에 결정론적 시스템으로 바꾼 것이며, 시스템의 공격 관점에서 바라보았을 때 Pailliar 공개키 암호 시스템에 비하여 안전성이 취약해지는 부분은 없다.

2.4 Diffie-Hellman형 자기 위탁 키 공유 시스템의 안전성

키 복구 기능을 가지는 키 공유 시스템에서 키 복구 기관은 키 공유 과정에 사용자간에 소통되는 정보를 이용하여 키 복구를 할 수 있다. 그러나, 이러한 키 복구 기능을 가지는 키 복구 시스템의 키 공유 과정에 사용자 A가 사용자 B에게 메시지 m을 숨겨서 보낼 수 있고, 키 복구 기관이 이를 감지할 수 없다면, 메시지 은닉 공격이 성공하는 것으로 볼 수 있다.

즉, 두 사용자 A와 B는 키 복구 기능을 가지는 키 공유 시스템을 통해 키 복구 기관으로부터 보호하고자 하는 메시지 m을 서로 주고받을 수 있고, 키 공유 과정 중에 필요한 메시지를 서로 교환한 후, 공유된 키를 이용하여 키 공유 과정 후에 메시지 m'에 대한 형식적인 암호 통신을 수행할 수 있으면 메시지 은닉 공격이 성공하는 것이다.

Diffie-Hellman형 자기 위탁 키 공유 시스템에서 사용자 A가 사용자 B에게 키 복구 시스템을 무력화하면서 메시지를 전달하는 방법은 다음과 같다.

1. 두 사용자 A와 B는 Diffie-Hellman형 자기 위탁 키 공유 시스템을 통해 정상적으로 키 K_1을 공유한다. 이는 subliminal channel을 구성하려는 준비 단계에 해당한다.

2. 키 K_1을 공유한 사용자 A와 B는 다음 세션(또는 그 이후의 임의의 약속된 세션)에서 subliminal channel 구성을 시도한다.

3. 키 복구 기능을 무력화하면서 메시지를 전달하려는 사용자 A는 난수 $a<n$을 선택하여 $g^a \bmod n^2$를 사용자 B에게 보내는 대신에 키 복구 기관에 대하여 안전하게 전달하려는, 즉, 키 복구 기능에 의해 복호화 되기를 원하지 않는 메시지 m을 함수 $\alpha(m,K_1)$으로 처리하여 사용자 B에게 보낸다. 함수 $\alpha(m,K_1)$의 예로는 $\alpha(m,K_1)=m\oplus K_1$이 있다. 또한, $\alpha(m,K_1)$은 임의의 y에 대하여 g^y이 될 것이나, $\alpha(m,K_1)=g^y$가 되는 y를 찾는 것은 이산 대수 문제를 해결하는 것과 동치인 것으로 매우 어렵다.

4. 사용자 B는 사용자 A로부터 받은 $\alpha(m,K_1)=m\oplus K_1$을 이용하여, 이전에 공유한 공유키 K_1을 사용하여 메시지 m을 추출한다.

5. 사용자 B는 사용자 A에게 난수 $b'<n$을 선택하여 $g^{b'} \bmod n^2$을 보낸다. 이 과정은 키 복구 기관에 두 사용자 A와 B가 자기 위탁 키 공유 시스템을 통하여 정상적인 과정을 수행하는 것으로 보이기 위함이다.

6. 이제 사용자 A와 사용자 B는 메시지 m을 비밀리에 전송하는데 성공하였다. 이 과정에서 사용자 B는 키 $K_2=\alpha(m,K_1)^{b'}=(g^y)^{b'} \bmod n^2$를 계산할 수 있으며, 이를 이용하여 암호 통신을 수행한다. 키 복구 기관이 복구하는 키 역시 $K_2=(g^y)^{b'} \bmod n^2$이 되는데, K_2에 의해 암호화된 메시지는 키 복구 기관으로부터 보호하고자 하는 메시지가 아닌 의미 없는 메시지임에 주목한다.

사용자 A는 $\alpha(m,K_1)=g^y$이 되는 y를 알 수 없으므로, 사용자 B와 키 $K_2=(g^{b'})^y \bmod n^2$를 공유할 수 없다. 그러나, 사용자 A와 사용자 B는 이미

비밀리에 메시지 m을 키 공유 시스템을 통해 은닉하였으며, 계속되는 암호 통신은 의미 없는 메시지에 대한 것이므로 사용자 A가 복호화를 수행하지 못하여도 문제가 되지 않는다.

주의할 점은 제안한 공격에서 $m \oplus K_1 \in Z_n^*$이 되어야 하므로, 한번의 공격에서 은닉할 수 있는 메시지 m의 크기는 n^2 비트가 된다.

2.5 공격 대응법

2.4절에서 소개한 메시지 은닉 공격은 두 사용자가 키 공유 시스템을 통해 공유한 키를 이용하여, 이후 세션의 키 공유 과정 중에 메시지를 은닉하여 키 복구 기관으로부터 보호하는 것을 특징으로 한다.

그런데, P. Paillier와 M. Yung이 제안한 자기 위탁 공개키 시스템은 이미 구축되어 있는 공개키 기반 구조에서 키 복구를 위한 특별한 부가 정보 통신 없이 키 복구 기관이 키 복구를 할 수 있게 하는 것을 특징으로 하므로 2.4절에서 기술한 것 이외에 키 변환 공격(key conversion attack)[8]등에 대하여 취약한 문제점을 가진다.

또한, 키 복구 기능을 가지는 키 공유 시스템의 안전성을 분석할 때 송신자와 수신자인 두 사용자가 서로 공모를 하는 것을 가정하는 것은 매우 강력한 조건으로 고려하기도 한다.

그러나, 두 사용자가 공모를 하여도 단순한 대응 방법에 의해 2.4절에서 제안한 공격 방법을 탐지할 수 있으며, 다음에서 제안하는 대응 방법은 기존의 공개키 기반 구조에 특별한 부가 정보 통신 없이 키 복구 기능을 부여하려는 P. Paillier와 M. Yung의 자기 위탁 공개키 시스템의 당초 목적과도 부합한다.

대응 방법으로는 키 공유 시 사용하는 난수에 대하여 특정한 조건을 주는 방법이 있다. 키 공유 시 사용하는 난수에 대하여 특정한 조건을 준다면 2.4절의 공격에서 사용자 A는 $\alpha(m, K_1) = g^y$의 y가 특정한 조건을 만족하게 $\alpha(m, K_1)$를 구성하여야 한다. 그러나, 이것은 $\alpha(m, K_1)$의 g에 대한 이산 대수 문제를 해결하여야 하는 것으로 매우 어려운 일이다. 따라서, 이산 대수 문제를 풀 수 있는 능력을 가진 키 복구 기관은 불법적인 방법을 통해 키를 공유하려 하고, 이와 함께 공격 발생 여부를 탐지할 수 있다.

대응 방법으로 사용할 수 있는 특정한 조건으로는 다음이 있다.

- 난수의 LSB 32비트를 모두 1로 구성한다.
- 난수의 MSB 32비트를 모두 1로 구성한다.
- 난수의 LSB 32비트를 사용자의 ID로 구성한다.
- 난수의 MSB 32비트를 사용자의 ID로 구성한다.

위에서 제시한 대책들 중에서 난수의 MSB 32비트를 사용자의 ID로 구성하는 방안의 안전성을 검토한다. 공격자가 대책을 무력화하면서, 메시지 은닉 공격을 성공하기 위해서는 공격 단계 3에서 $m \oplus K_1 = g^y$이 되는 어떤 y의 상위 32 비트를 사용자의 ID로 구성할 수 있어야 한다. 그러나, g와 $m \oplus K_1$이 주어지고, y를 구하는 것은 이산 대수 문제를 해결하여야 하는 것으로 이는 매우 어려운 문제이다. 따라서, 공격자가 대책을 무력화하면서, 메시지 은닉 공격을 수행하는 어려움은 이산 대수 문제의 어려움을 기반으로 하므로, 난수의 상위 32 비트를 사용자의 ID로 구성하는 대책은 2.4절에서 제안한 메시지 은닉 공격의 효과적인 대책이 된다. 난수의 상위 32 비트를 사용자의 ID로 구성하는 대책 이외의 다른 3가지 방법들도 동일한 이유로 2.4절의 공격 방법에 대한 효과적인 대책이 된다.

제시한 대책들에 대하여 공격자가 랜덤한 K_1에 대해 메시지 은닉 공격에 성공할 확률은 2^{-32}이다. 따라서, 약 2^{32}번 정도의 온라인 상의 메시지 은닉 공격을 시도하였을 경우에 한 번 정도 공격이 성공하게 된다는 것을 의미한다.

또한, 이들 대책은 P. Paillier와 M. Yung이 제안한 자기 위탁 공개키 시스템의 기본 개념인 공개키 기반 구조를 부가 정보의 사용 없이 키 복구에 활용한다는 것을 만족하는 특징을 가진다.

Ⅲ. 키 복구 기능을 가지는 WAKE 프로토콜의 안전성

이동 통신 분야는 정보 기술 분야에서도 가장 빠르게 발전하는 분야이며, 특히, UMTS(Universal Mobile Telecommunications System)[9]와 IMT-2000 (International Mobile Telecommunications System)[10]은 양질의 멀티미디어 서비스와 인터넷 서비스를 제공할 수 있을 것으로 기대된다. 이동 통신 서비스에서의 정보 보호 요소로는 기밀성, 사용자 익명성, 사용자의 서비스 제공자에 대한 인증 등이 있는데, 네트워크 관리자와 서비스 제공자의 수가 증가할수록 사용자의 서비스 제공자에 대

한 인증은 중요한 문제가 된다. 더불어, 사용자와 서비스 제공자간의 암호 통신 환경도 필요하게 될 것이다. WAKE 프로토콜은 UMTS 시스템에 사용되는 정보보호 기술의 연구, 개발을 수행하는 ASPeCT 프로젝트에서 제안한 키 공유 및 상호 인증 프로토콜로서, 이동 통신에서 사용자와 네트워크 관리자 또는 서비스 제공자간에 인증과 키 교환을 제공하는 프로토콜이다[11~12].

본 장에서는 WAKE 프로토콜과 키 복구 기능을 추가한 WAKE 프로토콜의 변형들을 먼저 알아보고, 키 복구 기능을 가지는 WAKE 프로토콜의 안전성을 검토한다.

3.1 WAKE 프로토콜

UMTS에서 인증과 키 공유 기능을 제공하는 WAKE 프로토콜을 설명한다[11~12]. 먼저, 사용된 기호를 정의한다.

- A: 사용자의 identity
- B: 서비스 제공자의 identity
- TTP_A: 사용자 A가 신뢰하는 신뢰기관의 identity
- g: 유한군의 생성자
- r_A: 사용자 A가 선택하는 랜덤수
- r_B: 사용자 B가 선택하는 랜덤수
- K_{AB}: 사용자 A와 사용자 B가 공유하는 세션키
- K_A^{-1}: 사용자 A의 서명용 개인키
- A_{Cert}: TTPA가 서명한 사용자 A의 공개키 인증서
- B_{Cert}: TTPA가 서명한 사용자 B의 공개키 인증서
- b: 사용자 B의 키 교환용 개인키(private key)
- g^b: 사용자 B의 키 교환용 공개키
- $\{m\}_{K_A^{-1}}$: 사용자 A의 서명용 개인키 K_A^{-1}로 서명한 메시지 m의 서명문
- $\{m\}_{K_{AB}}$: 대칭키 암호를 이용하여 키 K_{AB}로 암호화한 메시지 m의 암호문
- h: 해쉬 함수

WAKE 프로토콜은 다음과 같다.

1. A → B: g^{r_A}, TTP_A
2. A ← B: $r_B, h(K_{AB}, r_B, B), B_{Cert}$
3. A → B: $\{\{h(g^{r_A}, g^b, r_B, B)\}_{K_A^{-1}}, A_{Cert}\}_{K_{AB}}$

A는 랜덤수 r_A를 이용하여 g^{r_A}를 생성하고, 이를 TTP_A와 함께 B에게 전송한다. B는 랜덤수 r_B를 생성하고, 공유키 $K_{AB}=h(r_B,(g^{r_A})^b)$을 계산한다. 다음으로, B는 A에게 $r_B, h(K_{AB}, r_B, B), B_{Cert}$를 전송한다. A는 B로부터 받은 정보로부터 공유키 $K_{AB}=h(r_B,(g^b)^{r_A})$을 계산한 후, $h(K_{AB}, r_B, B)$를 계산하여 수신한 정보와 비교한다. A는 서명을 생성하여 $\{\{h(g^{r_A}, g^b, r_B, B)\}_{K_A^{-1}}, A_{Cert}\}_{K_{AB}}$를 B에게 전송한다. 서비스 제공자 B가 사용자 A를 인증하는 과정도 유사하게 수행된다.

3.2 RM-WAKE 프로토콜

위에서 설명한 WAKE 프로토콜에 키 복구 기능을 추가한 것으로는 K. Rantos와 C. Mitchell이 제안한 키 복구 기능을 가진 WAKE 프로토콜이 있다(이하 RM-WAKE 프로토콜)[13]. RM-WAKE 프로토콜의 내용은 다음과 같다.

1. A→B: $g^{r_A}, s_A, \{A\}_L, TTP_A$
2. A←B: $r_B, h(K_{AB}, r_B, B), B_{Cert}$
3. A→B: $\{\{h(g^{r_A}, g^b, r_B, B)\}_{K_A^{-1}}, A_{Cert}\}_{K_{AB}}$

여기서, r_A는 WAKE 프로토콜에서와 같이 사용자 A가 생성하는 단순한 난수가 아니라, 일방향 함수 f에 의해

$$r_A = f(w_A, s_A)$$

로 계산되는 정보이다. 여기서, s_A는 난수이며, w_A는 사용자 A와 사용자 A의 키 복구 기관인 KRA_A가 공유하는 비밀 정보이다. 또한, $L=(g^{x_A})^{r_A}$인데, g^{x_A}는 KRA_A의 공개키이다. 그리고, 사용자 A와 서비스 제공자 B가 공유하는 키는

$$K_{AB} = h(r_B, g^{br_A})$$

이 된다.

키 복구를 위해서 사용자 A의 키 복구 기관 KRA_A

는 키 공유 과정에서 $s_A, \{A\}_L, r_B, g^b$를 얻는다. 다음으로, $(g^{r_A})^{x_A}$를 이용하여, $\{A\}_L$을 복호화하여, 사용자 A의 신원을 확인한다. 다음으로, 사용자 A의 신원으로부터 w_A를 얻고, $r_A = f(w_A, s_A)$를 계산한다. r_A를 계산한 KRA_A는 $K_{AB} = h(r_B, g^{br_A})$를 구할 수 있다.

서비스 제공자 B의 키 복구 기관 KRA_B의 키 복구 과정은 사용자 A의 키 복구 기관 KRA_A의 키 복구 과정과 다르다. 서비스 제공자 B는 KRA_B에게 자신의 비밀키 b를 사전에 위탁한다. 그러면, KRA_B는 $K_{AB} = h(r_B, g^{br_A})$를 계산할 수 있다.

3.3 Nieto-WAKE 프로토콜

Nieto 등은 RM-WAKE 프로토콜이 서비스 제공자 B의 비밀키 b를 서비스 제공자 B의 키 복구 기관 KRA_B에 위탁하기 때문에, 사용자 A가 서비스 제공자 B와 통신하는지 아니면, 서비스 제공자 B의 키 복구 기관인 KRA_B와 통신하는지를 확인할 수 없다는 단점을 가짐을 지적하고, 이를 개선한 프로토콜을 제안하였다(이하 Nieto-WAKE 프로토콜)[14]. 그러나, Nieto-WAKE 프로토콜은 RM-WAKE 프로토콜의 단점을 보완한 반면, 키 복구 정보의 추가로 인하여 통신량 및 계산량이 증가한 단점을 가진다.

Nieto-WAKE 프로토콜에서 키 복구 정보 생성 과정을 설명한다. A는 $w_A (1 \le w_A \le q-1)$를 생성하고, 이를 KRA_A와 공유한다. 그리고, g^{w_A}를 공개한다. A는 랜덤수 $r_A (1 \le r_A \le q-1)$를 선택하고, g^{r_A}를 계산한다. 그리고, s_A를 다음과 같이 계산한다.

$$s_A = (w_A h(g^{r_A}) + r_A) \bmod q$$

서비스 제공자 B는 일방향 함수 f_B를 이용하여, 랜덤수 $r_B = f_B(w_B, s_B)$를 생성한다. 여기서, w_B는 서비스 제공자 B와 KRA_B가 공유한 비밀 정보이고, w_B는 랜덤수이다.

다음은 Nieto-WAKE 프로토콜의 인증 및 키 공유 과정이다.

1. A → B: g^{r_A}, TTP_A
2. A ← B: $r_B \oplus (g^{r_A})^b, h(K_{AB}, r_B, B), \{s_B\}_{K_{AB}}, B_{Cert}$
3. A → B: $\{\{h(g^{r_A}, g^b, r_B, B)\}_{K_A^{-1}}, A_{Cert}\}_{K_{AB}}, s_A, s_B$

다음으로 키 복구 과정을 설명한다. KRA_A는 다음과 같이 r_A를 계산한다.

$$r_A = s_A - w_A h(g^{r_A}) \bmod q$$

다음으로, KRA_A는 $K_{AB} = h(r_B, (g^b)^{r_A})$를 계산한다. KRA_B는 $r_B = f_B(w_B, s_B)$를 계산하고, 이를 이용하여 $(g^b)^{r_A} = (r_B \oplus (g^b)^{r_A}) \oplus r_B$를 계산한다. 다음으로 $K_{AB} = h(r_B, (g^b)^{r_A})$를 얻는다.

KRA_B는 $r_B = s_B - w_B h(g^{r_B}) \bmod q$로부터 r_B를 얻고, $g^{br_A} = (r_B \oplus g^{br_A}) \oplus r_B$를 이용하여 $K_{AB} = h(r_B, (g^b)^{r_A})$를 구한다.

사용자 A의 키 복구 정보 검증 과정은 다음과 같다. 공개 정보 g^{r_A}, s_A, A로부터, 제 3자는 다음 과정을 통해 사용자 A가 생성한 키 복구 정보의 무결성을 확인할 수 있다.

- g^{w_A}를 얻는다.
- $c' = h(g^{r_A}) \bmod q$를 계산한다.
- 제 3자는 $g^{s_A} = (g^{w_A})^{c'} \cdot g^{r_A}$임을 확인한다.

3.4 KL-WAKE 프로토콜

C. Kim과 P.Lee는 Nieto-WAKE 프로토콜에 서비스 제공자 B에 대한 키 복구 정보 검증 과정이 없으며, 서비스 제공자 B에 대한 키 복구 정보 검증을 사용자 A에 대한 키 복구 정보 검증과 유사하게 수행하더라도 r_A를 알고 있는 임의의 사용자는 서비스 제공자 B의 키 복구 정보로부터 w_B를 구할 수 있다는 단점을 지적하고, 사용자 B의 키 복구 정보 검증 과정을 추가한 프로토콜(이하 KL-WAKE 프로토콜)을 제안하였다[3]. KL-WAKE 프로토콜의 내용은 다음과 같다.

1. A→B: g^{r_A}, TTP_A
2. A←B: $g^{r_B}, w_B \oplus r_B \oplus (g^{r_A})^b$, $h(K_{AB}, g^{r_B}, B), \{s_B\}_{K_{AB}}, B_{Cert}$
3. A→B: $\{\{h(g^{r_A}, g^b, g^{r_B}, B)\}_{K_A^{-1}}, A_{Cert}\}_{K_{AB}}, s_A, s_B$

여기서, A와 B는 $K_{AB}=h(g^{br_A+r_B})$를 공유한다.

3.5 KL-WAKE 프로토콜의 안전성

KL-WAKE 프로토콜의 제안자들은 KL-WAKE 프로토콜에 대한 사용자들의 모의에 의한 공격을 배제하였다. 즉, 사용자들의 모의에 의한 키 복구 무력화 공격이 불가능한 것으로 가정하였다. 본 절에서는 사용자들의 모의에 의한 키 복구 무력화 공격을 제안하여, KL-WAKE 프로토콜이 사용자 모의에 의한 공격을 가정하지 않으면, 취약점을 가짐을 지적한다.

우선, 사용자 A와 서비스 제공자 B는 정상적인 KL-WAKE 프로토콜을 수행하여 인증 및 키 공유를 수행한다. 그러나, 이때 두 사용자는 공유키 이외의 공유 정보 R을 공유한다. R의 공유 방법으로는 다음이 있다.

$$R=(g^b)^{r_A}\oplus s_B$$

KL-WAKE 프로토콜 과정의 3예에서 사용자 A는 자신이 생성한 랜덤수 r_A와 서비스 제공자 B의 공개키 g^b, 그리고, KL-WAKE 프로토콜의 두 번째 과정에서 s_B를 얻을 수 있다. 또한, 서비스 제공자 B는 g^{r_A}와 s_B, 그리고, 자신의 비밀키 b를 이용하여 예에서 제시한 R을 계산할 수 있다.

사용자 A와 서비스 제공자 B는 공유 정보 R을 이용하여 약속된 키 공유 과정에서 다음을 수행한다.

1. 사용자 A는 서비스 제공자 B에게 g^{r_A}와 TTP_A를 송신한다. 이 과정은 정상적인 KL-WAKE 프로토콜과 동일하다.

2. 서비스 제공자 B는 공유키 K'_{AB}를 다음과 같이 계산한다.

$$K'_{AB}=h(g^{r_B}\cdot R)$$

다음으로, 서비스 제공자 B는 다음 정보를 사용자 A에게 전송한다.

$$g^{r_B}, w_B\oplus r_B\oplus R, h(K'_{AB}, g^{r_B}, B), \{s_B\}_{K'_{AB}}, B_{Cert}$$

3. 사용자 A는 이전에 서비스 제공자 B와 공유한 R과 서비스 제공자 B로부터 전송 받은 g^{r_B}를 이용하여 $K'_{AB}=h(g^{r_B}\cdot R)$를 계산하고, 다음을 서비스 제공자 B에게 전송한다.

$$\{\{h(g^{r_A}, g^b, g^{r_B}, B\}_{K_A^{-1}}, A_{Cert}\}_{K'_{AB}}, s_A, s_B$$

이 과정에서 사용자 A와 서비스 제공자 B는 $K_{AB}=h((g^b)^{r_A}\cdot g^{r_B})$를 공유할 수 있다. 그러나, KRA_B는 K_{AB}를 얻을 수 없다. 왜냐하면, KRA_B는 서비스 제공자 B와 공유한 비밀 정보 w_B를 가지고 있고, s_B 및 g^{r_B}를 이용하여 r_B를 얻을 수는 있으나, 사용자 A와 서비스 제공자 B의 통신 과정에서 얻을 수 있는 정보는 $w_B\oplus r_B\oplus R$ 이므로, K_{AB} 생성에 사용된 $(g^b)^{r_A}$ 대신에 R을 얻게 된다. 즉, KRA_B가 구하는 공유키 정보는 두 사용자가 공유한 $K_{AB}=h((g^b)^{r_A}\cdot g^{r_B})$가 아닌 $h(R\cdot g^{r_B})$가 된다. 따라서, 위의 공격을 통하여 서비스 제공자 B의 키 복구 기관인 KRA_B의 키 복구를 무력화할 수 있다. 그러나, 사용자 A의 키 복구 기관 KRA_A는 올바른 키 $K_{AB}=h((g^b)^{r_A}\cdot g^{r_B})$를 얻을 수 있다. 또한, KRA_A는 잘못된 키 공유를 수행함을 탐지할 수 있다.

이러한 공격이 성공하는 이유는 키 복구를 위하여 간접적으로 공개되는 g^{br_A}의 값이 사용자 A와 서비스 제공자 B의 키 공유를 위해서는 사용되지 않으며, KRA_B는 g^{br_A}를 검증할 다른 방법이 없기 때문이다.

KL-WAKE 프로토콜의 제안자들은 사용자 모의에 의한 공격 가능성을 배제하였다. 따라서, 본 절에서 제안한 공격은 KL-WAKE 프로토콜의 제안자들이 사전에 배제한 공격 방법에 해당된다. 그러나, 사용자 모의에 의한 공격을 배제하는 것은 매우 강력한 가정이며, 이러한 측면에서 KL-WAKE 프로토콜은 개선의 필요성을 가지고 있는 것으로 판단된다.

그리고, 분석 결과 KL-WAKE 프로토콜은 사용자 모의에 의한 공격을 배제할 경우에는 안전성에 문제가 없는 것으로 판단된다.

Ⅳ. 결 론

본 논문에서는 1999년 P. Paillier와 M. Yung

이 제안한 자기 위탁 공개키 시스템 중 Diffie-Hellman형 키 공유 시스템에 메시지 은닉에 의한 키 복구 무력화 공격 방법을 제안하고, 그 대책을 제시하였다. 제시한 대책은 공격자가 공격을 성공하기 위해서는 이산 대수 문제를 해결하여야 하는 어려움을 가지며, P. Paillier와 M. Yung이 제안한 자기 위탁 공개키 시스템의 기본 개념인 공개키 기반 구조를 부가 정보의 사용 없이 키 복구에 활용한다는 것을 만족하는 것을 특징으로 한다.

다음으로, 2001년에 C. Kim과 P. Lee가 제안한 키 공유 프로토콜인 KL-WAKE 프로토콜에 대하여 사용자 모의에 의한 공격을 적용하면, 서비스 제공자의 키 복구 기관을 무력화하는 공격이 가능함을 제시하였다. KL-WAKE 프로토콜에 대한 공격이 성공하는 이유는 키 복구를 위하여 간접적으로 공개되는 정보가 사용자 A와 서비스 제공자 B의 키 공유를 위해서는 사용되지 않으며, 서비스 제공자 B의 키 복구 기관은 이 정보를 검증할 다른 방법이 없기 때문이었다. 물론, KL-WAKE 프로토콜의 제안자들은 사용자 모의에 의한 공격 가능성을 배제하고, 프로토콜을 제안하였다. 따라서, 본 논문에서 제안한 공격은 KL-WAKE 프로토콜의 제안자들이 사전에 배제한 공격 방법에 해당된다. 그러나, 사용자 모의에 의한 공격을 배제하는 것은 너무나 강력한 가정이며, 이러한 측면에서 KL-WAKE 프로토콜은 개선의 필요성을 가지고 있는 것으로 판단된다. 또한, 분석 결과 KL-WAKE 프로토콜은 사용자 모의에 의한 공격을 배제할 경우에는 안전성에 문제가 없는 것으로 판단된다.

지금까지 사용자 모의에 의한 공격을 배제한 키 복구 기능을 가지는 키 공유 프로토콜이 많이 제안되어 왔다. 그러나, 키 복구 시스템의 모든 사용자들이 안심하고 키 복구 시스템을 사용하기 위해서는 사용자 모의에 의한 공격에도 안전한 방식이 필요하며, 이에 대한 연구가 계속되어야 할 것으로 사료된다.

참 고 문 헌

[1] National Institute of Standards and Technology. FIPS PUB 185 : Escrowed Encryption Standard, 1994.

[2] P. Paillier and M. Yung. "Self-escrowed public-key Infrastructures", ICISC '99, pp. 257--268, 1999

[3] C. H. Kim and P. J. Lee, "New Key Recovery in WAKE Protocol", PKC 2001, LNCS 1403, pp. 325-338, Springer-Verlag, 2001

[4] A. Young and M. Yung. "Auto-Recoverable Auto-Certifiable Cryptosystems". In Advances in Cryptology - Eurocrypt '98, LNCS 1403, pp. 17--31, Springer-Verlag, 1998

[5] A. Young and M. Yung. "Auto-Recoverable Cryptosystems with Faster Initialization and the Escrow Hierarchy", PKC'99 LNCS 1560, pp. 306--314, Springer-Verlag, 1999

[6] P. Pailliar. "Public-Key cryptosystems based on composite degree residuosity classes". In Advances in Cryptology - Eurocrypt '99 pp. 223--238, 1999

[7] W. Diffie and M. Hellman. "New Directions in Cryptography". In IEEE transactions on information Theory, Vol. IT-22, No. 6, pp. 644--654, 1976

[8] K. Viswanathan, C. Boyd, and E. Dawson, "Strong binding for software key escrow", IWS'99, IEEE Press, 1999. available at http://sky.fit.qut. edu.au/~boyd/paper/.

[9] U. Black, "Third Generation Mobile Systems(TGMs)", Second Generation Mobile & Wireless Networks, Prentice Hall, 1999

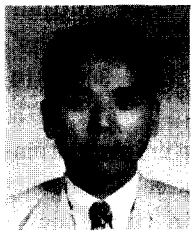
[10] T. Ojanpera and R. Prasad, "IMT-2000 Applications", Wideband CDMA for Third Generation Mobile Communication, Artech House Publishers, pp. 65-76, 1998

[11] G. Horn and B. Preneel, "Authentication and payment in future mobile systems", Computer Security - ESORICS'98, pp. 277-293, 1998

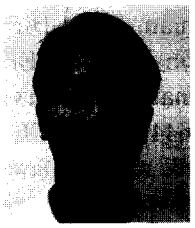
[12] ACTS AC095, ASPeCT Deliverable D02, "Initial Report on Security Requirements", AC095/ATEA/W21/DS/P/02/B, 1997, available from http://www.esat.

kuleuven.ac.be/cosic/ aspect.
[13] K. Rantos and C. Mitchell, "Key Recovery in ASPeCT Authentication and Initialization of Payment protocol", Proceedings of ACTS Mobile Summit, Sorrento, Italy, June, 1999
[14] J. Nieto, D. Park, C. Boyd, and E. Dawson, "Key Recovery in Third Generation Wireless Communications Systems", PKC 2000, pp. 223-237, 2000

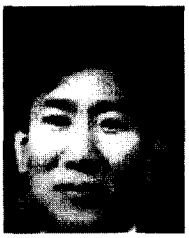
〈著者紹介〉



김 대 호(Dae-Ho Kim) 종신회원
1977년 2월 : 한양대학교 전자공학과 학사
1984년 8월 : 한양대학교 전자공학과 석사
1993년 : University of Maryland 방문 연구원
1995년 : 전기통신기술사, 정보통신기술사
1977년 3월~1999년 12월 : 한국전자통신연구원 본부장
2000년 1월~현재 : 국가보안기술연구소 소장



박 상 우(Sangwoo Park) 정회원
1989년 2월 : 고려대학교 수학교육과 졸업
1991년 8월 : 고려대학교 수학과 석사
1991년 8월~1999년 12월 : 한국전자통신연구원 선임연구원
2000년 1월~현재 : 국가보안기술연구소 선임연구원



이 동 훈(Dong Hoon Lee) 종신회원
1983년 8월 : 고려대학교 경제학사
1987년 12월 : Oklahoma University 전산학 석사
1992년 5월 : Oklahoma University 전산학 박사
1992년 8월 : 단국대학교 전자계산학과 전임강사
1993년 3월~1997년 2월 : 고려대학교 전산학과 조교수
1997년 3월~2001년 2월 : 고려대학교 전산학과 부교수
2001년 3월~현재 : 고려대학교 정보보호대학원 부교수