通信情報保護學會誌
第 10 券 第 1 號, 2000. 3

# 키 복구 시스템 및 안전성에 관한 고찰

유 준 석*, 원 동 호*, 이 인 수**, 김 병 천**, 박 성 준**

요 약

최근들어 암호의 사용이 민간 부문으로 급속히 확산되고 있으며, 암호 사용으로 인한 부작용이 큰 문제로 대두되고 있다. 이 문제를 해결하기 위한 대안으로 키 복구에 대한 연구가 활발히 진행되고 있으며, 지금까지 많은 키 복구 기술들이 제시되었다. 그러나, 현재의 기술로는 법 집행 기관의 암호문에 대한 접근권 보장과 사용자들의 프라이버시 보호라는 키 복구의 두 가지 목적을 충족시키는데 어려움이 있다. 본 고에서는 일반적인 키 복구 시스템의 모델과 지금까지 제시된 대표적인 키 복구 시스템들을 분류하여 간단히 살펴보고, 각 시스템에 대한 공격 및 취약점들을 살펴본다.

## Ⅰ. 서 론

군사적 또는 국가적 용도로 주로 사용되던 암호는 20세기 후반에 접어들면서부터 정보통신 분야의 급속한 발전과 함께 그 활용 범위가 민간 부문으로 급속히 확산되었고, 이에 따라 은행 업무나 상거래와 같은 실생활의 일들이 네트워크를 통해 이루어지는 등 많은 편리함을 제공하게 되었다. 그러나, 암호의 보급에 따라 그러한 암호의 긍정적인 면 뿐 아니라 암호의 역기능도 부각되었다. 다시 말해서 범죄자들에 의한 암호의 악용과 키의 분실, 손상 등에 따른 암호문의 복호 불가 등이 그것이다. 이에 대한 대책으로 현재 키 복구에 대한 연구가 세계적으로 활발히 진행 중이며, 관심이 집중되고 있다.

우선 키 복구란 개인이 소지한 비밀키가 있어야만 평문으로 복호할 수 있는 암호화 데이터를 제한적인 경우에 한해서 개인이나 기관 등이 합법적인 수단을 통해 복호할 수 있는 기술 및 체계라고 정의할 수 있으며, 키 복구 기술은 사용자의 프라이버시 보호와 정부의 암호문에 대한 접근권 보장이라는 상반되는 두 가지 목적을 만족시켜야 한다. 이러한 상반되는 두 요구를 모두 충족시키는 것은 현실적으로 매우 어려우며, 그 요구들의 균형점을 찾는 것이 키 복구 시스템의 설계 목적이라고 할 수 있다.

이러한 키 복구 시스템의 설계 목적과 관련하여 키 복구 기술은 개인의 사생활을 침해할 수 있고, 사용자들에 의해 키 복구 기능이 우회될 가능성이 존재하는 등 많은 논쟁의 여지가 남아있다. 그러나, 급속도로 발전하는 정보화 사회에서 키 복구의 필요성은 의심할 여지가 없으며, 현재 세계 주요 국가의 암호 정책들도 기본적으로 키 복구에 대한 내용들을 포함하고 있다.

위에서 언급한 키 복구 시스템의 설계 목적에 부합하는 키 복구 시스템을 만들기 위해서는 지금까지 제시된 여러 가지 키 복구 시스템과 각 시스템에 대한 취약점 및 공격들을 살펴볼 필요가 있으며, 이는 계속되는 절에서 다루도록 하겠다.

본 고는 다음과 같은 구성을 가진다. 2장에서는 키 복구 기술에 상관없이 모든 키 복구 시스템에 적용 가능한 키 복구 시스템의 일반적인 모델과 그 구성요소들에 대해 살펴본 후, 현재 사용되고 있는 키 복구 기술을 분류한다. 또한 3장에서는 지금까지 제시되었던 키 복구 시스템들 중에서 대표적인 것들을 살펴보며, 각각에 대한 취약점과 가능한 공격 방법들을 고찰해 보고 마지막 4장에서 결론을 맺는다.

* 성균관대학교 전기전자 및 컴퓨터 공학부
** 한국정보보호센터
※ 이 원고는 1999년 한국정보보호센터의 과제(98- 260-07)에 의하여 연구되었음

## II. 키 복구 시스템 모델 및 분류

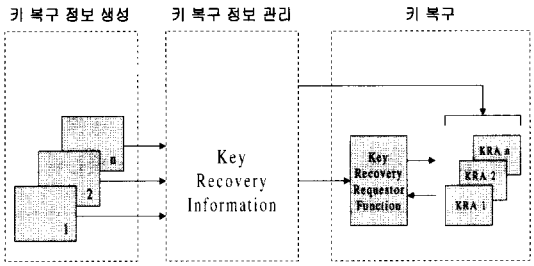
이번 장에서는 먼저 특정 키 복구 기술에 관계없이 모든 키 복구 시스템에 적용 가능한 키 복구 시스템의 일반 모델 및 구성에 대해 살펴보고, 다음으로 키 복구 시스템을 키 위탁 방식, 캡슐화 방식 및 TTP 방식의 세 가지로 분류하여 그 개념 및 특징을 알아본다.

### 1. 키 복구 시스템의 일반 모델[1]

일반적으로 키 복구 시스템(key recovery system : KRS)은 복호화 키를 사용할 수 없는 경우에 정당한 권한이 있는 사람 또는 기관으로 하여금 암호화 데이터에서 평문을 복호해 낼 수 있도록 하는 시스템으로써 키 복구라는 용어는 다양한 키 복구 기술(예 : 키 위탁 방식, 캡슐화 방식 등)에 광범위하게 적용될 수 있다. 각 키 복구 기술은 목적키(target key)라고 불리는 키의 복구를 목적으로 하고 있으며, 목적키는 다음과 같은 것이 될 수 있다.

- 데이터를 복호하는데 사용되는 데이터 암호화 키 (data encryption key : DEK)
- 암호화된 DEK를 복호하는데 사용되는 키

목적키를 복구하기 위해서 필요한 정보는 각 키 복구 기술마다 다르며, 그 정보들을 키 복구 정보(key recovery information : KRI)라고 하는데 키 복구 정보는 여러 가지 방법으로 관리될 수 있다. 즉, 키 복구 정보는 암호문이 전송되는 짧은 기간만 존재할 수도 있고, 저장되어 비교적 긴 기간 동안 존재할 수도 있으며, 여러 장소에 분산되어 존재할 수도 있다.




(단, 화살표는 KRI의 흐름을 나타낸다.)

그림 1. 키 복구 시스템의 일반 모델

그림 1은 크게 키 복구 정보 생성, 키 복구 정보 관리, 키 복구의 세 부분으로 이루어지는 키 복구 시스템의 일반 모델을 나타내고 있으며, 이 모델은 키 복구 시스템의 기능적인 면에 초점을 맞추고있다.

키 복구 시스템의 일반 모델은 크게 목적키의 복구에 사용되는 키 복구 정보를 생성하는 기능(KRI generation function)과 키 복구 정보를 관리하는 기능(KRI management function), 그리고 키 복구 정보들로부터 목적키를 복구해 내는 기능(key recovery function)으로 이루어진다. 이 중에서 키 복구 정보 관리 기능은 키 복구 정보 전달 기능(KRI delivery function)과 키 복구 정보 검증 기능(KRI validation function)으로 세분될 수 있으며, 키 복구 기능은 키 복구 요청 기능(key recovery requestor function)과 키 복구 대행 기능(key recovery agent function)으로 세분화된다. 각 기능들에 대한 내용은 다음과 같다.

#### 1.1 키 복구 정보 생성 기능

키 복구 정보 생성 기능은 키 복구 정보 제공자(KRI provider)라고 불리는 하나 이상의 키 복구 정보 생성 개체들로 구성된다. 키 복구 정보 생성 기능은 목적키 복구를 위해 필요한 키 복구 정보를 생성, 조립, 형성하고, 형성된 키 복구 정보를 키 복구 전달 기능에 제공하게 되는데 키 복구 정보 생성 기능은 다수의 시스템이나 장소에 분산되어 존재할 수 있다. 키 복구 정보 제공자로는 송신자나 수신자, 키 분배 센터, CA(certification authority) 등이 될 수 있으며, 키 복구 정보는 키 복구 대행 시스템의 식별자, 키의 식별자, 날자와 시간, 인증 정보, 알고리즘 식별자, 암호화된 키 등을 포함한다.

#### 1.2 키 복구 정보 전달 기능

키 복구 정보 전달 기능은 키 복구 정보 관리 기능의 일부분으로써 키 복구 정보가 키 복구 정보 검증 기능과 키 복구 기능에서 사용될 수 있도록 한다. 여기서 사용될 수 있도록 한다는 것은 키 복구 정보를 다른 키 복구 기능으로 직접 전송하거나 다른 키 복구 기능들이 접근할 수 있는 일정한 곳에 위치시키는 것을 의미한다. 이 기능 또한 키 복구 정보 생성 기능처럼 다수의 시스템이나 장소에 분산되어 존재할 수 있다.

### 1.3 키 복구 정보 검증 기능

키 복구 정보 검증 기능은 키 복구 정보 관리 기능의 일부분으로써 키 복구 기술에 따라 선택적으로 사용된다. 이 기능은 키 복구 요청자에게 키 복구 정보를 통하여 올바른 목적키를 복구해 낼 수 있다는 확신을 제공한다.

### 1.4 키 복구 요청 기능

키 복구 요청 기능은 키 복구 요청자와 키 복구 요청 시스템으로 이루어진다. 키 복구 요청자는 암호화된 데이터의 복호를 가능하게 해 주는 복구 정보를 얻으려는 개체이며, 키 복구 요청자에 의한 키 복구 요청은 합법적인 것이어야 한다. 키 복구 요청자는 키 복구 정보를 키 복구 요청 시스템에 제공하는데 키 복구 요청 시스템은 목적키가 복구될 수 있도록 하는 목적키 정보(TKI)를 얻기 위해 하나 이상의 키 복구 대행 기능과 상호작용을 하게 된다. 이렇게 얻어진 목적키는 데이터를 복구하기 위해 직접 또는 간접적으로 사용될 수 있다. 키 복구 정보는 일반적으로 하나의 키 복구 대행 시스템이 목적키를 복구하는데 필요한 모든 정보를 제공하지 못하도록 설계되어 있으므로 키 복구 요청 시스템은 각 키 복구 대행 기능들이 제공하는 키의 조각들로부터 목적키를 복구해 낸다.

### 1.5 키 복구 대행 기능

키 복구 대행 기능은 키 복구 대행 시스템에 의해 수행되는 기능으로 키 복구 대행 시스템은 키 복구 요청 시스템에 의한 정당한 키 복구 요청에 대하여 키 복구 서비스를 수행하는 신뢰되는 개체이다. 키 복구 대행 기능에 의해 수행되는 키 복구는 키 복구 요청 기능이 키 복구 대행 기능에게 제공하는 키 복구 정보의 부분이나 전부를 처리하여, 그 결과를 키 복구 요청 기능으로 돌려줌으로써 행해진다. 여기서 키 복구 요청 기능으로 전해지는 결과값은 목적키, 여러 개의 키 조각, 또는 목적키를 복구할 수 있도록 하는 키 관련 정보가 될 수 있다.

## 2. 키 복구 시스템의 구성요소

D. E. Denning은 키 복구 시스템을 구성하는 요소를 다음의 세 가지 논리적 구성요소로 나누고 있으며,[2] 이 구성요소들은 키 복구 시스템에 일반적으로 적용이 가능하다.

- 사용자 보안 구성요소(user security component) : 데이터의 암호화 및 복호화를 수행하는 하드웨어나 소프트웨어이며, 데이터 복구 필드(data recovery field : DRF)를 암호문에 덧붙이는 역할을 한다.
- 복구 기관 구성요소(recovery agent component) : 키 복구 기관에 의해 운영되며 데이터 복구 키들의 보관, 사용 등을 관리한다.
- 데이터 복구 구성요소(data recovery component) : 이 요소는 데이터 복구 필드와 복구 기관 구성요소에서 얻어지는 정보로부터 평문을 얻어내는데 필요한 알고리즘, 프로토콜, 또는 장치들로 구성되며, 허가된 데이터 복구 시에만 작동한다.

앞 절에서 살펴본 키 복구 시스템의 기능들은 본 절에서 살펴본 키 복구 시스템의 어느 특정한 구성요소에 존재하는 것이 아니며, 사용되는 키 복구 기술에 따라 구성요소가 수행하는 키 복구 시스템의 기능은 달라질 수 있다. 그러나 일반적으로 키 복구 대행 기능은 복구 기관 구성요소에서, 키 복구 요청 기능은 데이터 복구 구성요소에서 실행된다.

## 3. 키 복구 방식의 분류

지금까지 제안된 키 복구 시스템들은 키 복구 시스템의 구성요소나 키 복구 정보 관리 기능의 특징에 따라 크게 위탁 방식, 캡슐화 방식, TTP 방식으로 나누어 볼 수 있으며, 각각의 개념 및 특징은 다음과 같다.

### 3.1 키 위탁(Key Escrow) 방식

이 방식은 암호문 복호를 위한 키, 또는 키의 조각들을 신뢰되는 기관에 위탁하고, 필요시에 위탁 기관으로부터 그 정보들을 얻어내 키를 복구해 내는 방식이다. 이 방식에서 위탁되는 키나 키 조각들은 사용자들의 비밀키와 관련된 것들이므로 사용자의 프라이버시 보호를 위해서는 위탁 기관의 신뢰성이 절대적으로 보장되어야한다는 문제가 있으며, 이를 위하여 비밀 분산 방식(secret sharing scheme)이 주로 사용되고 있다. 또한 사용자의 키가 복구되었을 경우 키의 사용 기간을 제한하는 것과 위탁되는 정보가 유효한 것인가를 확인하는 것도 해결되어야 할 문제이다. 반면에 키 위탁 방식은 유사시에

확실한 키 복구가 가능하다는 장점이 있다.

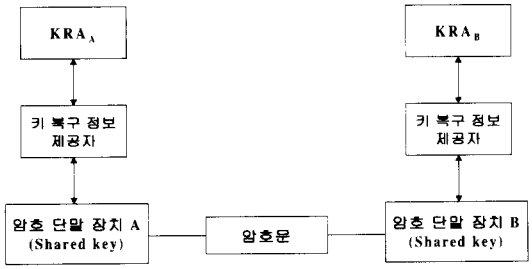



그림 2. 키 위탁 방식

그림 2는 키 복구를 위해 두 개의 서로 다른 KRA를 사용하여 암호 통신을 하는 두 사용자 단말 장치 사이의 상호작용을 도식화한 것으로 키 위탁 방식은 복구될 키나 키의 조각, 또는 키 관련 정보를 키 복구 대행 시스템, 즉 위탁 기관에 보관하게 된다. 이 방식에서 제 3의 기관이나 사용자 단말 시스템은 키 복구 정보 제공자로써 키 복구 정보 및 키를 생성하여 키 복구 대행 시스템들로 전달하고, 키 복구 대행 시스템은 키 복구가 필요한 경우에 키 복구 요청 시스템으로부터의 복구 요청을 받아 목적 키를 복구할 수 있는 정보를 제공한다.

### 3.2 캡슐화(Encapsulation) 방식

이 방식은 생성되는 각각의 암호문에 대해 키 복구 정보를 생성하여 암호문과 함께 전송 또는 저장하는 방식으로 필요시에 암호문에 부가되어있는 키 복구 정보로부터 키를 복구해낼 수 있는 방식으로 복구되는 키는 키 위탁 방식과는 달리 세션키이다. 이 방식에서는 복구되는 키가 세션키이므로 감청 기관의 복구 능력을 제한할 수 있어 키 위탁 방식보다는 사용자의 프라이버시 보호에 유리하며, 기존 프로토콜의 확장 필드를 이용하여 간단하게 사용할 수 있다는 이점이 있다. 그러나 키 복구에 필요한 정보를 사용자들이 생성하므로 조작이나 변조를 통해 키 복구 기능을 우회할 수 있는 문제점이 있다.

그림 3은 키 복구를 위해 캡슐화 방식을 사용하여 암호 통신을 하는 두 사용자 단말 장치 사이의 상호작용을 도식화한 것이다. 이 방식에서는 먼저 목적키를 복구 가능하게 하기 위해 사용자 단말 장치 내의 키 복구 정보 생성 기능은 목적키에 대응하는 키 복구 정보를 생성하여 캡슐화 한 후, 상대편 사용자 단말 장치로 암호문과 함께 전송을 한다.

### 3.3 TTP(Trusted Third Party) 방식

TTP 방식은 신뢰할 수 있는 제 3자인 TTP가 복구될 사용자의 비밀키를 생성하고 사용자에게 분배하는 방식으로 실제적인 키 위탁은 일어나지 않으나 사용자의 long-term 키를 TTP가 직접 보관하고 있으므로 위탁된다고 할 수도 있다. 이 방식에서는 TTP가 사용자의 비밀키를 생성·분배하므로 각 TTP가 신뢰성 보장이 절대적으로 중요하다. 이 방식에서는 TTP가 사용자들의 비밀키를 모두 가지고 있으므로 필요시에 TTP에 의한 키 복구가 확실히 보장되며, TTP 사이의 키 생성 방식이 통일된다면 국가간 호환이 용이하다는 장점이 있다. 반면에 많은 TTP가 필요하며 TTP와 사용자, TTP와 TTP 사이의 병목현상이 심하다는 것이 단점으로 지적되고 있다.

그림 4는 키 복구를 위해 TTP 방식을 사용하여 암호 통신을 하는 두 사용자 단말 장치 사이의 상호작용을 도식화한 것이다. 이 방식에서 두 사용자 단말 장치가 통신을 하기 위해서는 각 TTP들 사이에 공

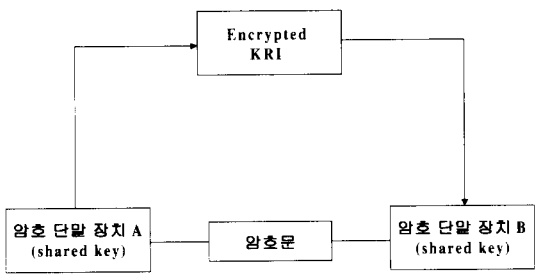



그림 3. 캡슐화 방식

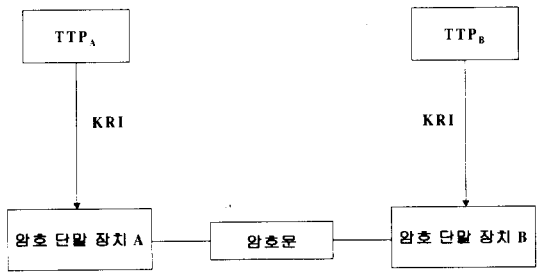



그림 4. TTP 방식

유된 키가 필요하며, 각 TTP는 암호 단말 장치가 생성하는 암호문의 복호를 위한 키 복구 정보를 생성, 제공한다. 즉, 이 방식에서 TTP는 KRA 역할뿐 아니라 KRI 제공자의 역할도 수행한다.

## Ⅲ. 키 복구 기술 및 안전성

이번 장에서는 지금까지 제시되었던 키 복구 기술 중에서 몇 가지 대표적인 기술에 대해서 살펴보고, 그에 대한 취약점 및 공격들을 살펴본다. 여기서 유의할 점은 키 복구에서 말하는 공격은 암호학에서 의미하는 것과는 다른 의미로 사용되고 있다는 것이다. 즉, 암호학에서 말하는 일반적인 의미의 공격은 송신자가 의도한 수신자 외의 권한이 없는 제 3자가 암호화 데이터에서 키 또는 평문을 얻어내는 것을 말하는 반면에 키 복구에서의 공격이라 함은 송신자가 의도한 수신자 외의 제 3자가 암호화 데이터에서 키 또는 평문을 얻어 낼 수 없도록 함을 의미한다.

### 1. EES(Escrowed Encryption Stan- dard)

키 복구 제도는 1993년 4월에 미국 행정부가 정부 및 민간 부문의 정보 보호를 위한 새로운 대칭키 암호 시스템 개발을 명시하는 클리퍼(clipper) 정책을 발표하면서 실제적인 추진이 이루어졌으며, 이 정책은 1994년에 EES라는 표준으로 승인되었다.[3]

EES 시스템은 크게 tamper-resistant 특성을 지니는 클리퍼 또는 캡스톤(capstone) 칩이 내장된 암호 단말 장치, 암호문을 감청하여 키 복구의 요청 및 암호문의 복호를 수행하는 법 집행 기관의

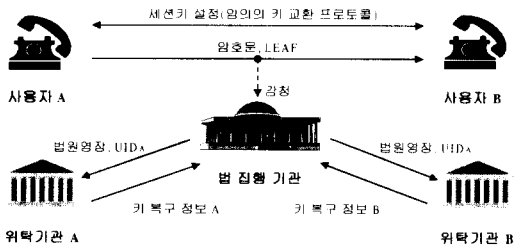



그림 5. EES의 개념도

LED(law enforcement decryptor), 그리고 LED의 요청을 받아 키를 복구할 수 있는 정보를 제공해 주는 위탁 기관의 세 부분으로 구성된다. 그림 5는 EES의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

#### 1.1 동작 과정

두 사용자 A, B 사이의 암호 통신 과정과 법 집행 기관에 의한 암호문의 복호 과정은 다음과 같다.

〔암호 통신 과정〕

사용자 A는 사용자 B와 SKIPJACK이라는 비공개 대칭 암호 알고리즘을 사용하여 다음과 같이 암호 통신을 수행한다.

① 사용자 A와 사용자 B는 임의의 키 교환 프로토콜을 사용하여 암호 통신에 사용할 세션키 KS를 설정한다.

② 사용자 A의 클리퍼 칩은 세션키 KS와 칩이 생성한 IV(initial vector), 그리고 그 외의 파라미터를 통해 검사합을 구해내고 다음과 같은 형태의 LEAF(law enforcement access field)를 생성한다.

$$LEAF = E_{KF}(UID_A \parallel E_{KU_A}(KS) \parallel Chksum)$$

③ 사용자 A는 세션키 KS를 사용하여 암호문을 생성하고, ②에서 생성한 LEAF와 IV를 암호문과 함께 사용자 B에게 전송한다.

④ 사용자 B의 클리퍼 칩은 수신한 LEAF내의 검사합을 통하여 LEAF의 무결성을 검사한다.

⑤ ④의 검사가 통과되면 ①에서 설정된 세션키 KS로 암호문을 복호한다.

〔키 복구 과정〕

위의 암호문을 감청한 법 집행 기관은 다음과 같이 사용자 A의 세션키를 복구한다.

① 법 집행 기관의 LED는 감청한 LEAF로부터 $UID_A$와 $E_{KU_A}(KS)$를 구한 후, 법원 영장과 $UID_A$를 각 키 위탁 기관에게 전송한다.

② 각 위탁 기관은 $UID_A$에 해당하는 사용자 A의 키 복구 정보를 법 집행 기관으로 전송한다.

③ 법 집행 기관은 각 위탁 기관으로부터 수신한

사용자 A의 키 복구 정보로부터 구해진 $KU_A$를 사용하여 세션키 KS를 구해낸다.

### 1.2 안전성

EES는 그 계획이 발표되면서부터 사용자들의 프라이버시 침해가 될 수 있다는 우려와 함께 다음과 같이 많은 문제점들이 지적되었다.

· 비공개 암호 알고리즘의 사용

비공개 암호 알고리즘인 SKIPJACK의 사용은 사용자들로부터 알고리즘 내에 trapdoor가 존재할 가능성이 있다는 의심을 사고있으며, DES 등에 비해 그 안전성이 경험적으로 증명되지 않았다.

· 사용자 암호문에 대한 불법적 복구 가능

일단 한번 사용자의 키를 복구한 법 집행 기관은 법원의 허가 없이도 그 사용자의 암호문을 복호할 수 있다.

· 비싼 비용

EES는 tamper-resistant 하드웨어의 사용을 필수로 하기 때문에 시스템을 구현하는데 많은 비용이 소모된다.

· 키 복구 기능의 우회 가능

법 집행 기관이 키를 복구할 수 없도록 키 복구 기능을 우회할 수 있는 다양한 공격들이 존재한다.

EES는 위에서 언급한 것처럼 다음과 같은 다양한 공격이 가능하다.[4][5]

#### 1.2.1 LEAF Obscuring 공격

이 공격은 법 집행 기관이 LEAF나 암호문을 복구해 낼 수 없도록 하는 공격으로 다음과 같이 두 가지 방법으로 나누어 볼 수 있다.

· Double Encryption

송신자는 키 복구 시스템의 암호 알고리즘이 아닌 상대방과 미리 동의된 임의의 암호 알고리즘을 사용하여 LEAF와 암호문 또는 LEAF만을 한 번 더 암호화시키며, 이것을 수신한 수신자 또한 그 알고리즘을 이용하여 복호를 수행한다. 이 공격은 송수신자 사이에 합의된 암호 알고리즘의 적용 순서에 따라 사전 암호(pre-encryption) 방식과 사후 암호(post-encryption) 방식으로 나눌 수 있다. 이 공격이 더욱 정화된 형태의 예는 다음과 같다.

Diffie-Hellan의 키 교환 프로토콜을 사용하는 두 사용자는 512 비트의 키를 공유할 수 있게되는데 각 사용자는 공유된 512 비트의 키 중 80비트만 SKIPJACK을 위한 세션키로 사용하며, 나머지 128비트는 vernam 암호 시스템으로 LEAF를 암호화하기 위한 키로 사용한다. 이 방법은 EES 뿐 아니라 임의의 다른 키 복구 시스템에도 적용이 가능한 공격이다.

· 별도의 채널을 통한 LEAF와 키의 사전 설정

LEAF를 암호문이 전송되는 채널 외의 별도의 채널을 통해 나누어 가지거나 사전에 나누어 가지는 방법으로 실제 응용에서는 그다지 효과적이지 못 하다.

#### 1.2.2 LEAF Feedback 공격

이 방법은 송신자가 LEAF와 IV를 보내지 않고 단지 암호문만을 송신함으로써 법 집행 기관이 암호문을 복호할 수 없도록 하는 방법으로, 수신자는 암호문을 복호하기 위하여 송신자의 실제 LEAF와 IV 대신에 자신이 임의로 생성한 LEAF'와 IV'를 사용한다. 이 공격에서는 수신자가 LEAF는 다른 것을 사용하더라도 IV만은 같아야만 암호문을 복호해 낼 수 있게 되는데 이와 같은 IV 동기화 문제는 사용되는 네 가지 암호화 모드(ECB, CBC, CFB, OFB)에 따라서 송신자가 적당한 크기의 dummy 블록을 메시지에 추가하는 방법 등을 이용하여 해결하고 있다.

#### 1.2.3 Brute-Force LEAF Search 공격

키 복구 기능을 우회하려는 부당한 사용자와 그렇지 않은 정당한 사용자 사이의 암호 통신을 가능하게 하는 공격으로 그 내용은 다음과 같다.

수신측에서 이루어지는 LEAF 검증 과정은 LEAF내에 포함되어 있는 16 비트의 검사합 필드에 전적으로 의존하게 되고 송신측에서 임의로 생성한 128비트의 LEAF' 1/216의 확률로 LEAF 검증 과정을 통과할 수 있다. 즉, 송신자는 216 정도의 시도를 통하여 수신측의 검증 과정은 통과하지만 정당한 KS가 아닌 임의의 KS' 포함하는 LEAF'를

생성할 수 있는 것이다.

### 1.2.4 Squeezing 공격

LEAF Feedback 공격에서는 LEAF를 전송하지 않음으로써 복구 기관이 키 복구를 수행하지 않고도 불법 통신 사실을 알 수 있었다. Squeezing 공격은 복구 기관이 키 복구를 수행하지 않고는 통신자의 불법 사실을 알아차릴 수 없는 공격으로, 키 복구 기능을 우회하려는 공격자는 자신이 생성한 LEAF 대신에 합법적인 사용자의 LEAF'를 이용하여 암호 통신을 수행하게된다. 이 공격을 이용하면 복구 기관이 불법 통신 사실을 알아차리고, 그 사용자를 추적할 경우에 불법 사용자가 아닌 정당한 사용자가 추적된다. 그러나, 이 공격에서는 불법적 사용자가 정당한 사용자의 세션키와 LEAF를 알고 그 세션키를 자신의 세션키로 성립시켰을 경우를 가정하고 있다.

### 1.2.5 Self-Squeezing 공격

Self squeezing 공격은 LEAF feed- back 공격을 발전시킨 것으로써 복구 기관은 키 복구 없이 사용자의 불법 통신 사실을 알 수 없으며, squeezing 공격과 달리 제 3의 합법적인 사용자의 LEAF나 세션키를 필요를 필요로 하지도 않는다. 이 공격의 내용은 다음과 같다.

먼저, 부정한 두 사용자는 세션키 KS와 그로부터 파생된 KS'를 다음과 같이 생성한다.

$$KS' = f(KS)$$

(단, f()는 두 사용자 사이에 합의된 일방향 함수)

송신자는 KS와 KS'를 사용하여 LEAF와 LEAF'를 생성하며, EKS(m)||LEAF 대신 EKS(m)||LEAF'를 수신자에게 전송한다. 수신자는 수신한 EKS(m)||LEAF'에서 LEAF'를 LEAF로 교체한 후, EKS(m)을 복호한다. 나중에 복구 기관이 키를 복구할 경우, 불법 통신 사실이 적발되지만 사용자들은 기기 테스트 중이었다고 주장할 수 있다.

## 2. TIS-Software Key Escrow

1994년에 미국의 TIS(trusted infor- mation systems)사는 정부의 암호문에 대한 접근 능력에 있어서 EES와 동일한 능력을 가지면서도 하드웨어적으로 구현된 EES에 비해 몇 가지 장점을 지닌 소프트웨어적으로 구현된 키 위탁 시스템을 제안하였다.[6] 이 시스템은 EES와 비교하여 다음과 같은 차이를 가지고 있다.

- 공개 암호 알고리즘의 사용
- 공개 파라미터들을 이용한 LEAF 검증
- 소프트웨어적 구현에 따른 저렴한 비용

그림 6은 TIS-software key escrow의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

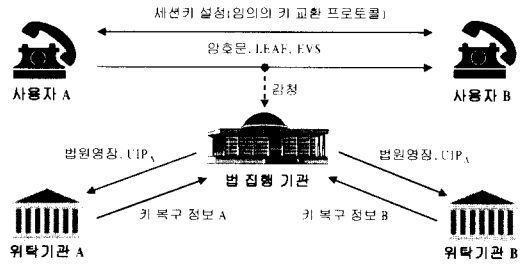



그림 6. TIS-software key escrow의 개념도

### 2.1 동작 과정

두 사용자 A, B 사이의 암호 통신 과정과 법 집행 기관에 의한 암호문의 복호 과정은 다음과 같다.

[암호 통신 과정]

사용자 A는 사용자 B와 다음과 같이 암호 통신을 수행한다.

① 사용자 A의 프로그램은 사용자 B와 미리 설정한 세션키 KS를 이용하여 암호문을 생성하고 다음과 같이 LEAF와 EVS (escrow verification string)을 생성하여 사용자 B에게 암호문과 함께 전송한다.

$$LEAF = E_{KFpub}(E_{KUpub_A}(KS) \,||\, UIP_A)$$

$$EVS = E_{KS}(\, UIP_A,\ KUpub_A,\ E_{KEPFpriv}(UIP_A,\ KUpub_A))$$

② 사용자 B의 프로그램은 EVS를 통해 유효성이 확인된 정보인 $UIP_A$와 $KUpub_A$를 사용

하여 다음과 같이 LEAF의 정당성을 검사한다.

$$LEAF \stackrel{?}{=} E_{KFpub}(E_{KUpub_A}(KS) \parallel UIP_A)$$

③ 사용자 B의 프로그램은 ②에서 LEAF가 정당하다고 판정되었을 경우에만 세션키 KS를 이용하여 암호문을 복호한다.

〔키 복구 과정〕

위의 암호문을 감청한 법 집행 기관은 다음과 같이 사용자 A의 세션키를 복구한다.

① 법 집행 기관의 LED는 감청한 LEAF로부터 구해진 $UIP_A$를 법원 영장과 함께 각 위탁기관으로 전송한다.
② 각 위탁 기관은 $UIP_A$에 해당하는 사용자 A의 키 복구 정보를 법 집행 기관으로 전송한다.
③ 법 집행 기관은 각 위탁 기관으로부터 수신한 사용자 A의 키 복구 정보로부터 구해진 $KUpriv_A$를 사용하여 세션키 KS를 구해낸다.

### 2.2 안전성

TIS-software key escrow는 하드웨어적 키 복구 시스템 구현에 따른 고비용을 절감하면서 소프트웨어적으로 키 복구 시스템을 구현하겠다는 의도로 개발되었으나 소프트웨어로 구현된 키 복구 시스템은 일반적으로 다음과 같은 문제점을 지니고 있다.

- 비공개 암호 알고리즘은 사용할 수 없다.
- 키 복구 소프트웨어가 사용자에 의해 변경되지 않고 올바르게 동작한다는 것을 확신하기 어렵다.
- 소프트웨어에 내장된 비밀키는 노출되기 쉽다.

이러한 소프트웨어 키 복구 시스템의 문제를 해결하기 위해 TIS-software key escrow에서는 공개 암호 알고리즘의 사용과 공개키 암호 시스템의 사용을 채택하였지만 여전히 사용자에 의한 프로그램 조작 문제는 남아있다.

## 3. Commercial Key Escrow

EES를 소프트웨어적으로 구현한 TIS- software key escrow 방식은 수신측에서 LEAF의 정당성을 확인하기 위해 공개 정보들로부터 LEAF를 재생성함으로써 M. Blaze에 의한 공격을 부분적으로 해결하고, 암호 알고리즘의 사용에 큰 제약이 없는 등 EES의 문제를 다소 해결하였지만 사용자의 입장에서의 키 복구보다는 여전히 정부의 입장이 강조되고 있다. 이에 따라서 1994년 8월에 TIS는 정부의 암호문에 대한 접근권을 보장하면서 사용자 입장의 키 복구, 즉 사용자의 키가 분실, 파괴, 또는 손상되었을 경우에 키 복구를 통해 암호문 복호가 가능한 commercial key escrow 시스템을 제안하였다.[7] 이 시스템은 또한 이전의 EES나 TIS-software key escrow 시스템이 전화 시스템을 이용한 전송 데이터에만 그 범위가 한정된 것에 비해 저장된 데이터에까지 그 범위를 확장하고 있다.

이 방식에서는 키 복구를 위해 DRC(data recovery center)라는 신뢰 기관을 두고 있다. DRC는 하나의 회사가 독자적인 용도로 설립할 수도 있으며, 서비스 기관이 설립하여 일반인에게 키 복구 서비스를 제공할 수도 있다. 또한 DRC나 다른 어떠 장소에도 위탁되는 사용자의 키가 없으며, 복구되는 키는 사용자의 세션키가 되는 캡슐화 방식을 사용한다. 그림 7은 commercial key escrow의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

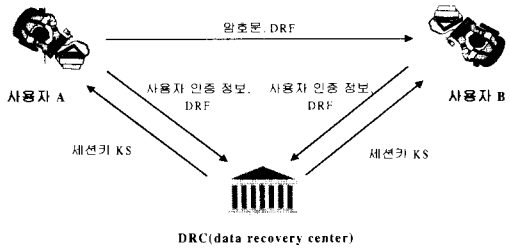



그림 7. Commercial key escrow의 개념도

### 3.1 동작 과정

키 복구를 위한 일반적인 절차는 다음과 같은 단계를 거친다.

〔프로그램 등록 과정〕

이 단계에서는 사용자가 EEP(escrow enabled

program)를 사용자 컴퓨터에 설치하고 DRC에 등록하는 절차를 포함하며, 그 과정은 다음과 같다.

① 각 사용자는 EEP를 설치할 때 각 사용자를 인증하기 위한 정보를 DRC에 제공한다.
② DRC는 각 사용자가 사용할 식별자와 DRC의 공개키, DRC의 식별자를 사용자 프로그램으로 전송한다.

〔암호 통신 과정〕

EEP는 메시지를 암호화 할 때마다 다음을 수행한다.

① EEP는 메시지를 암호화 할 때마다 다음 정보를 포함하는 데이터 복구 필드(data recovery field : DRF)를 생성한다.

$$E_{DRCpub}(ID_A)$$
$$E_{DRCpub}(KS)$$
$$ID_{DRC}$$

② ①에서 생성된 데이터 복구 필드는 해당 암호문에 부가되며, 이들은 함께 저장되거나 전송된다.

〔키 복구 과정〕

사용자의 키가 분실 또는 손상되어 암호문을 복호할 수 없을 경우에 사용자는 다음과 같은 과정을 통해 키를 복구한다.

① 암호문을 복호할 수 없는 사용자는 해당 암호문에서 얻어낸 데이터 복구 필드와 자신의 인증 정보를 DRC로 전송한다.
② DRC는 사용자로부터 전송된 인증 정보를 통해 사용자를 인증한다.
③ DRC는 사용자로부터 전송된 데이터 복구 필드를 자신의 비밀키 $DRC_{priv}$로 복호하여 사용자의 세션키 KS를 구해낸다.
④ DRC는 사용자의 세션키 KS를 사용자의 공개키로 암호화하여 사용자에게 전송한다.

만약 정부가 사용자의 키를 정당하게 복구할 필요가 있을 경우에는 법원으로부터 발급받은 영장을 사용자 회사에 제출하여 암호화 데이터를 얻을 수 있고, 그 데이터와 영장을 DRC에 제출함으로써 사용자의 키를 복구할 수 있다.

### 3.2 안전성

이 방식 또한 TIS사가 제안한 소프트웨어 방식을 채택한 키 복구 시스템으로써 TIS- software key escrow에서의 문제점인 사용자에 의한 프로그램 조작이 가능이 가능하다는 것이 문제점으로 지적되고 있다.

## 4. Active investigator를 가지는 키 위탁 시스템

1995년에 P. Horster, M. Michels, H. Petersen 등은 키가 한번 복구되고 나면 법 집행 기관이 법원의 영장 없이도 사용자의 암호문에 대한 불법적 복호가 가능하다는 EES의 문제점과 임의의 사용자가 부당하게 생성된 키 정보를 전송하여도 검출이 되지 않으며, 법 집행 기관이 키 교환 프로토콜에 참여하지 않은 경우에는 암호문 복호를 할 수 없다는 〔8〕에서의 문제점에 대한 해결책으로 active investigator를 가지는 키 위탁 시스템을 제안하였으며,[9] 키 복구 정보들은 두 사용자가 키 교환 프로토콜을 수행하는 과정 중에 다른 키 복구 기능들로 전달된다.

그들은 키 위탁 시스템에 참여하는 참여자들을 몇 가지로 분류하고 있는데 그 중 법원의 허가를 얻어 용의자의 통신을 감청하는 참여자(예 : 법 집행 기관)를 investigator로 정의하고 있으며, 감청한 암호문을 복호해 내는 주체에 따라 active investigator와 passive investigator로 나누고 있다.

이 방식의 주요 내용은 다음의 세 가지 특성으로 설명된다.

- ElGamal류의 서명 방식을 통하여 사용자가 임의로 선택한 파라미터들은 네트워크에 의해 그 유효성이 검증된다.
- 각 사용자는 매 시간 기간 $I_t$ 동안 사용할 서로 연관이 없는 비밀키와 공개키의 쌍 $(x_t, y_t)$을 소유하며, investigator는 일정 시간 기간 동안의 통신만을 복호할 수 있다.
- Investigator가 과거의 비밀키를 위탁 기관으로부터 얻는다면 네트워크가 저장한 정보들을

이용하여 과거의 암호문을 복호할 수 있다.

그림 8은 키 교환 프로토콜을 이용한 키 복구 시스템의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

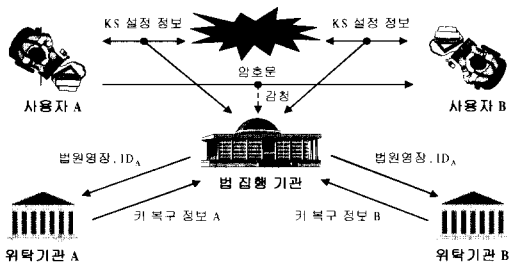



그림 8. Active investigator를 가지는 키 위탁 시스템의 개념도

## 4.1 동작 과정

〔키 교환 과정〕

시간 기간 $I_t$동안 사용자가 사용할 세션키는 다음과 같이 교환된다.

① 네트워크는 사용자 A와 B가 키를 생성하는데 사용될 정보인 $m_A$와 $m_B$를 선택하여 사용자들에게 전송한다.

② 사용자 A와 B는 각각 랜덤수 k를 선택하여 다음과 같이 계산된 r과 s를 상대 통신자에게 전송한다.

$$r_{A(B)} = \alpha^{k_{A(B)}} \pmod p$$
$$s_{A(B)} = x_{A(B),t} \cdot (r_{A(B)} \oplus m_{A(B)}) + k_{A(B)}$$

③ 네트워크는 상대 통신자에게 전송되는 r의 유효성을 다음과 같이 검사하여 정당한 경우에만 전달한다.

$$\alpha^{s_{A(B)}} \stackrel{?}{=} y_{A(B),t}^{r_{A(B)} \oplus m_{A(B)}} r_{A(B)} \pmod p$$

④ 각 사용자는 네트워크를 통해 전달된 r을 이용하여 다음과 같이 세션키를 계산한다.

$$KS_{AB} = h(r_{B(A)}^{x_{A(B),t}} \circ y_{B(A),t}^{k_{A(B)}})$$

〔키 복구 과정〕

① 법 집행 기관은 시간 기간 $I_t$에 해당하는 법원 영장을 $ID_A$와 함께 위탁기관에 제출함으로써 시간 기간 $I_t$에 해당하는 사용자 A의 비밀키 $x_{A,t}$를 얻는다.

② 법 집행 기관은 위탁 기관으로부터 얻어낸 비밀키를 이용하여 다음을 계산한다.

$$r_B^{x_{A,t}}$$
$$k_A = s_A - x_{A,t} \cdot (r_A \oplus m_A) \pmod q$$

③ ②에서 계산한 값을 이용하여 다음과 같이 세션키 $K_{A,B}$를 계산한다.

$$K_{A,B} = h(r_B^{x_{A,t}} \circ y_B^{k_A}) \pmod p$$

## 4.2 안전성

1999년에 김승주 등은 P. Horster 등이 제안한 키 위탁 시스템이 그들의 주장과는 달리 time-boundness 특성을 갖지 못함을 지적하였으며,[10] 그 내용은 다음과 같다.

사용자 A와 B는 각 시간 기간 $I_1, I_2, \cdots, I_n$에 대해 세션키 $K_{(A,B),1}$, $K_{(A,B),2}$, …, $K_{(A,B),n}$을 사용하게 되고, 다음의 식을 통해 세션키를 계산한다.

$$K_{(A,B),t} = h(r_{B(A)}^{x_{A(B),t}} \circ y_{B(A)}^{k_{A(B)}}, K_{(A,B),t-1})$$
(단, t = 2, 3, …, n)

이와 같이 구해진 키를 세션키로 사용한다면 시간 기간 $I_t$에 해당하는 사용자의 비밀키 $x_{A(B),t}$를 가진 investigator라도 세션키 $K_{(A,B),t}$를 구할 수 없게된다. 물론 inves- tigator가 사용자들이 사용한 과거의 모든 세션키를 알고 있다면 세션키 $K_{(A,B),t}$를 구할 수 있지만 이것은 위에서 언급한 두 번째 특성에 위배된다.

## 5. Fair Public-Key Cryptosystem

1992년에 Silvio Micali는 민주주의 국가에서 정부의 요구와 시민의 요구, 즉 범죄 집단에 의한 암호의 악용 방지와 개인의 프라이버시에 대한 권리 유지를 위하여 기존의 암호 시스템을 공정한 공개키 암호 시스템으로 재구성할 수 있는 방법을 제안하였다.[11] FPKC는 사용자, 다수의 신뢰 위탁 기관, 그리고 인증기관으로 구성되는데 키를 다수의 신뢰 위탁 기관에 위탁하고 위탁된 키의 정당성을 인증기관이 확인하는 동시에 인증기관이나 각 위탁 기관이 사용자의 비밀정보에 대한 어떠한 부분 정보도 알아낼 수 없고 오직 N개의 위탁 기관이 모두 모여야만 비밀정보를 복구할 수 있도록 하는 비밀분산 방식을 이용한다. 그림 9는 FPKC의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

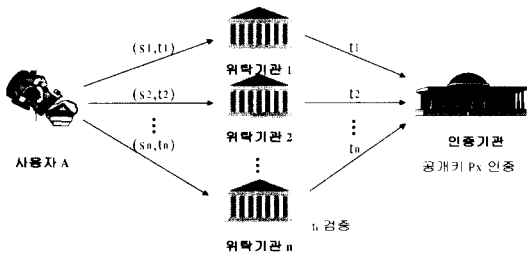



그림 9. FPKC의 개념도

### 5.1 동작 과정

〔키 위탁 과정〕

① 각 사용자는 다음의 조건을 만족하는 비밀키와 공개키 쌍을 생성한다.

$$s_1 + s_2 + \cdots + s_n = S_X$$
$$t_1 \cdot t_2 \cdot \cdots \cdot t_n = P_X$$

② 각 사용자는 위탁기관에 공개키 조각과 비밀키 조각의 쌍 $(s_i, t_i)$을 위탁한다.

③ 각 위탁 기관은 다음 식을 검사하여 공개키 조각에 대한 비밀키 조각의 정당성을 검증한 후, 인증기관에 정당한 공개키 조각 $t_i$를 전송한다.

$$t_i = g^{s_i}$$

④ 인증기관은 각 위탁기관으로부터 수신한 사용자의 공개키 조각의 정당성을 다음 식을 통하여 검증한 후, 정당하면 $P_X$를 사용자의 공개키로 인증한다.

$$t_1 \cdot t_2 \cdot \cdots \cdot t_n \stackrel{?}{=} Px$$

⑤ 사용자는 인증된 공개키를 통하여 암호 통신을 수행한다.

〔키 복구 과정〕

① 법 집행 기관은 법원의 영장을 받아 각 위탁기관에 사용자의 비밀키 조각을 요구한다.
② 각 위탁기관은 법 집행 기관의 요청에 의해 보관하고 있는 사용자의 비밀키 조각을 법 집행 기관으로 전송한다.
③ 법 집행 기관은 각 위탁기관으로부터 전송받은 비밀키 조각들로부터 다음과 같이 사용자의 비밀키를 복구해 낸다.

$$s_1 + s_2 + \cdots + s_n = S_X$$

### 5.2 안전성

#### 5.2.1 Shadow public-key[12]

FPKC의 가장 큰 약점은 사용자가 자신의 비밀키와 공개키 쌍을 스스로 생성한다는 것이다. 이것은 subliminal channel을 사용하는 공격을 가능하게 하며 그 내용은 다음과 같다. FPKC의 일반 사용자들은 자신의 공개키와 비밀키 쌍 (P, s)를 생성하여 P는 공개하고, 정부 기관이 비밀키 s를 생성할 수 있도록 s의 조각들을 위탁 기관에 위탁하게 된다. 이 때 공격자는 정당한 키 쌍 (P, s)외에 "shadow key"라고 불리우는 키 쌍 (P', s')를 다음과 같이 생성한다.

$$P' = f(P)$$

(단, f() : 계산이 쉽고 공개된 함수)

공격자는 일반 사용자와 같은 방법으로 (P, s)를 사용하지만 s'는 자신의 shadow secret key로 보관한다. 만약 누군가 공격자에게 복구 기관이 복호할 수 없도록 메시지를 보내고자 한다면, 송신자는 공격자의 P를 이용하여 P' = f(P)를 계산하고 P'를 사용하여 메시지를 암호화하여 전송한다. 공격자는 비밀로 간직하고 있던 s'를 이용하여 P'로 암호화된 메시지를 복호해 낸다. 이 공격에서 공격자는 정당한 키 쌍인 (P, s)를 생성하여 합법적인 사용자가 사용하는 것과 같은 방법으로 이용하므로 공격자에 의한 부정 행위가 발각되지 않는다.

이 공격은 사용자 스스로가 자신의 키를 생성함으로 가능한 것이므로 사용자와 신뢰 기관이 함께 키를 생성한다면 방지될 수 있다.

### 5.2.2 Lack-of-fairness 공격[5]

Silvio Micali는 FPKC가 공정하다고 주장하고 있다. 즉, 범죄자들을 감시하면서도 사용자의 프라이버시를 보호할 수 있다는 것이다. 하지만 이 공격은 FPKC가 범죄자의 감시와 사용자의 프라이버시 보호라는 두 가지 요구를 동시에 만족하지 않는다는 것을 보여준다. 예를 들어 ElGamal 암호 시스템에 기반한 FPKC의 경우 다음과 같은 공격이 가능하다.[3]

사용자 A는 다음의 식을 만족하는 공개키와 비밀키의 쌍 ( $y_A$, $x_A$)을 소유한다.

$$y_A \equiv g^{x_A} \bmod p$$

범죄자 B는 사용자 A에게 다음과 같이 이루어진 암호문 (K, C)를 전송한다.

$$K = g^R$$
$$C \equiv y_A^R \cdot M \bmod p$$

암호문을 수신한 사용자 A는 다음과 같이 메시지 M을 복호해 낸다.

$$M \equiv C \cdot (K^{x_A})^{-1} \bmod p$$

이 과정에서 법 집행 기관이 공개된 정보 K와 C로부터 메시지 M을 얻기 위해서는 사용자 A의 비밀키 $x_A$를 복구해야만 한다. 즉, FPKC에서는 메시지를 송신하는 범죄자의 키가 아니라 범죄자로부터 메시지를 수신하는 사용자의 비밀키가 노출되는 것이다.

## 6. Binding Cryptography

1997년에 E. R. Verheul 등은 부정한 사용자에 의한 메시지 조작과 정확한 키 복구 정보를 포함하지 않은 암호문을 수신자가 복호할 수 있도록 소프트웨어를 조작하는 것을 방지하고자 binding cryptography라는 것을 제안하였다.[13] 이 방법에서는 제 3자가 사용자의 비밀 정보를 모르더라도 binding data를 통해 부정 행위를 발견할 수 있다. 본 시스템은 TRP(trusted recovery party)라고 불리우는 키 복구 기관에 사용자의 키나 비밀 정보를 맡기지 않으므로 캡슐화 방식으로 분류된다. 그림 10은 binding cryptography의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

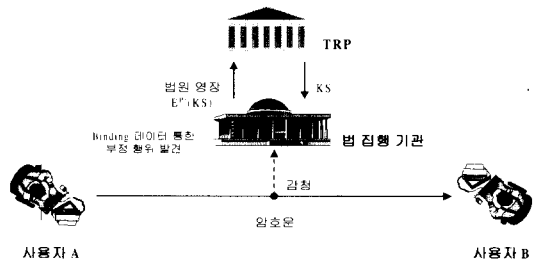



그림 10. Binding cryptography의 개념도

### 6.1 동작 과정

〔암호 통신 과정〕

다음은 사용자 A가 사용자 B에게 암호문을 전달하는 과정이다.

① 사용자 A와 사용자 B는 임의의 키 교환 프로토콜을 이용하여 세션키를 공유한다.

② 사용자 A는 다음 형태의 암호문 C를 생성하여 사용자 B에게 전송한다.

$$C = E_{KS}(M) \| E_{Kpub_B}(KS) \| E_{TRPpub}(KS) \| \text{binding data}$$

③ 사용자 B는 자신의 비밀키를 사용하여 수신한 암호문으로부터 세션키 KS를 복구한 후 암호문을 복호한다.

암호문의 정당성은 수신자와 공개된 정보들로부터 암호문 내의 binding data를 확인하여 검사할 수 있다.

〔키 복구 과정〕

법 집행 기관이 binding data로부터 사용자의 부정을 발견하게 되면 다음의 과정을 통하여 사용자의 키를 복구해 낸다.

① 법 집행 기관은 법원 영장과 암호문으로부터 얻어낸 $E_{TRPpub}(KS)$를 TRP에게 전송한다.
② TRP는 자신의 비밀키 $TRP_{priv}$를 사용하여 수신된 정보로부터 사용자의 세션키 KS를 구해낸다.
③ TRP는 복구한 세션키 KS를 법 집행 기관에게 전송하고 법 집행 기관은 그 키로부터 암호문을 복호한다.

### 6.2 안전성

B. Pfitzmann은 binding cryptogra- phy를 사용하는 사용자들이 의심을 받지 않고 키 복구 기능을 우회하여 암호 통신을 할 수 있는 방법을 제시하였다.[14] 이 공격은 LEAF obscuring 공격의 double encryption과 비슷하며 다음과 같이 수행된다.

① 사용자 A는 메시지 M에 대한 암호문 C를 다음과 같이 생성한다.

$$C = E_{KS}(M) \| E_{Kpub_B}(KS) \| E_{TRPpub}(KS) \| \text{binding data}$$

② 사용자 A는 ①에서 생성된 암호문 C에서 $E_{TRPpub}(KS) \| \text{binding data}$ 부분을 삭제하고 나머지 부분에 수신자가 메시지를 복구할 수 있도록 하는 정보를 부가한 형태의 새로운 메시지로 암호문 C′를 생성하여 사용자 B에게 전송한다.

$$D = (\text{info} \| E_{KS}(M) \| E_{Kpub_B}(KS))$$

$$C' = E_{KS'}(D) \| E_{Kpub_B}(KS') \| E_{TRPpub}(KS') \| \text{binding data}$$

③ 사용자 B는 C′로부터 D를 복호한 후 info에서 얻어지는 정보를 통하여 M을 얻어낸다.

## 7. GCHQ(Government Communica- tions Head Quarters) 프로토콜

TTP 방식에 기반한 키 복구의 대표적인 예인 GCHQ 프로토콜은 Diffie-Hellman 방식을 이용하여 정부기관 내에 사용되는 e-mail간의 키 복구와 법 집행 기능을 할 수 있도록 하는 통제된 키 액세스를 제공하려는 목적으로 영국의 RHC(royal holloway college)에서 제안되었다. 이 프로토콜은 사후에 민간 부분과의 통신에도 사용될 목적으로 설계되었으며, 모든 통신에 TTP가 관여하는 방식을 취하고 있다. 그림 11은 GCHQ의 동작 과정을 개괄적으로 보여주고 있으며, 그 내용은 아래에서 설명한다.

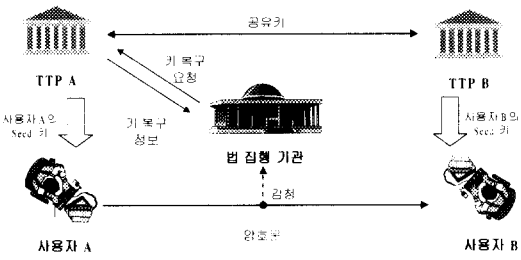



그림 11. GCHQ 개념도

### 7.1 동작 과정

〔암호 통신 과정〕

① 각 TTP는 자신의 영역에 속한 사용자들이 비밀 송/수신키를 생성할 수 있는 seed 키를 생성하여 각 사용자에게 전송한다.
② 각 TTP는 사용자들의 공개 송신키와 기타 정보를 이용하여 송신 인증서를 생성하고 공개 수신키를 이용하여 수신 인증서를 생성하며, 각각을 공개 디렉토리에 등록한다.

③ 사용자 A는 데이터 키 DK를 생성하여 데이터를 암호화하고 토큰 키 TK를 생성하여 DK를 암호화한 후 TK와 위에서 인증서를 생성하는데 사용된 파라미터들을 포함하는 인증서를 생성하여 암호문과 함께 전송한다.

④ 이를 수신한 사용자 B는 수신 인증서를 검증하고 자신이 비밀리에 보관하고 있는 정보와 송신자의 공개 정보로부터 TK를 계산해 낸 후, DK를 얻어낸다.

〔키 복구 과정〕

사용자 A의 암호문을 복구할 합법적 권한을 가진 법 집행 기관은 사용자 A의 TTP에게 그 사용자의 키 복구 정보를 요청하고, 그 요청을 받은 TTP는 해당 키 복구 정보를 전송한다. 키 복구 정보를 수신한 법 집행 기관은 키 복구 정보와 수신자의 공개 정보를 통하여 키를 복구할 수 있으므로, 법 집행 기관은 수신자를 알아야만 한다.

### 7.2 안전성

1997년에 R. Anderson은 GCHQ 프로토콜의 문제점들을 지적하였으며,[15] 그 개략적인 내용은 다음과 같다.

- 두 인증 기관 사이의 상호 작동을 위한 키가 손상된다면 두 영역의 전체 통신이 노출된다.
- 다른 방식보다 확장성은 뛰어나지만 TTP의 수가 많이 요구되며, TTP로의 병목현상이 심하다.
- 키의 관리가 한 기관에 집중되어 실세계에서의 유연성이 떨어진다.
- 공격자에 의한 정보의 추출 및 손상이 가능하다.
- X.509 인증서를 사용하는 GCHQ는 2자리로 날자를 나타내므로 2000년 문제를 일으킬 수 있다.

이 외에도 GCHQ 프로토콜에는 많은 제약 사항이 있다고 지적되고 있다.

## 8. 기타 키 복구 시스템

1995년에 A. Shamir는 사용자의 비밀키 전부를 위탁하지 않고 키의 일부분만 위탁하는 방법인 partial key escrow 방식을 제안하였다.〔16〕 이 방식에서 사용자는 자신의 비밀키의 일부만을 위탁기관에 위탁함으로써 키의 위탁되지 않은 나머지 부분은 exhaustive search를 통하여 찾도록 하는 것이다. 물론 어느 정도 이상의 계산능력을 지닌 공격자가 키의 나머지 부분을 찾는 것이 불가능한 것은 아니지만 이 방식은 대량의 키를 동시에 그리고 빠르게 복구할 수 없도록 하자는 것이 그 주요 개념이라 할 수 있다. 그러나 이 방식은 키의 위탁된 조각을 몰라도 사용자의 공개키를 통해 미리 키를 계산할 수 있다는 early recovery의 문제를 지니고 있다.[17]

또한 1999년에 채승철, 이임영은 키 복구 시스템이 가져야할 요구사항을 몇 가지로 제시하고 키 위탁 방식을 이용한 키 복구 시스템을 제안하였다.[18] 그러나, 그들이 제안한 키 위탁 시스템은 시스템의 오용이 어려워야 한다는 조건을 만족하지 못한다. 즉, 사용자는 시스템을 조작함으로써 복구 기관이 암호문을 복호할 수 없도록 하여 키 복구 기능을 우회할 수 있으며 그 내용은 다음과 같다.

먼저 이 시스템에서 암호 통신을 위해 전송되는 정보는 다음의 형태를 갖는다.

$$E_{SK}(M) \| g^k \bmod p \| SK \cdot P_R^k \bmod p$$

(단, M : 메시지, k : 랜덤수, PR : 수신자의 공개키, SK : 세션키)

복구 기관은 위의 전송 정보를 감청하여 키를 복구할 수 있는 정보인 $S'(= P_R^k)$를 구할 수 있다. 아래의 수식은 복구 기관이 S′와 감청 정보로부터 키를 복구해 내는 과정이다.

$$(SK \cdot P_R^k)/S' \equiv (SK \cdot P_R^k)/P_R^k \equiv SK \ (\bmod p)$$

그러나, 키 복구 기능을 우회할 목적을 가진 사용자가 실제 세션키 SK가 아닌 임의의 SK′를 사용하여 다음 형태의 메시지를 전송한다면, 복구 기관은 SK대신 SK′를 얻게되고 이로부터 복구된 메시지는 단지 랜덤한 정보일 뿐이다.

$$E_{SK}(M) \| g^k \bmod p \| SK' \cdot P_R^k \bmod p$$

물론 복구 기관은 감청 정보로부터 랜덤한 정보가

복구됐을 경우, 사용자를 추적할 수 있지만 사용자는 단순히 기기 검사 중이었다고 주장할 수 있다.

이 밖에도 협력하는 두 공격자가 합법적인 사용자의 메시지 헤더에 위탁되지 않은 세션키를 삽입함으로써 수행되는 spoofing 공격이나 범죄자가 정당한 사용자의 키를 사용할 경우에 정당한 사용자의 암호문도 복호될 수 있는 key cloning 공격 등 키 복구 시스템에 대한 많은 공격들이 존재한다.[5]

지금까지 살펴본 바와 같이 여러 가지 키 복구 시스템들이 제안되었지만 모두가 그 나름대로의 취약점들을 가지고 있으며 아직까지 정부의 암호문에 대한 접근권과 일반 사용자들의 프라이버시 보호라는 두 가지 요구를 충분히 만족시키는 시스템이 제안되지는 못 하였다.

## Ⅳ. 결 론

암호의 급속한 민간 부문 보급은 일상 생활에 있어서 많은 편리함을 제공하게 되었지만 범죄 집단에 의한 암호의 악용과 키의 분실 및 손상에 따른 암호문의 복호 불가와 같은 부작용 또한 크게 대두되고 있다. 이러한 암호의 부작용에 대한 여러 가지 대처 방안들 중에서 현재 세계 각 국에서는 키 복구에 대한 관심이 고조되고 있으며 연구가 활발히 진행되고 있다.

키 복구 시스템은 정부의 암호문에 대한 접근권 보장과 사용자의 프라이버시 보호라는 두 가지 요구를 동시에 만족시켜야 하지만 지금까지 제안된 시스템들은 여러 가지 취약점들이 지적되고 있으며, 이러한 문제점으로 인하여 아직까지 키 복구는 그 시행에 많은 어려움을 겪고 있다. 그러나, 현재 세계 주요 국가의 암호 정책은 세부적 내용에서는 차이를 보이고 있으나 키 복구에 대한 내용을 담고 있으며, 앞에서 언급한 설계 목적에 부합되는 키 복구 시스템 개발을 위한 연구를 활발히 진행중이다. 이러한 국제 정세로 미루어 보아 가까운 장래에는 국가와 민간이 모두 적절히 수용할 수 있는 키 복구 시스템이 개발될 것으로 기대된다.

그러나, 현재 우리 나라의 경우 키 복구에 대한 연구나 관심이 외국에 비해 미흡하다고 사료되며 암호의 사용이 점점 보편화되는 과정에서 암호의 역기능을 방지할 수 있는 대안으로 키 복구에 대한 더 많은 관심과 연구가 수행되어야 할 것이다.

본 고에서는 지금까지 제안되었던 여러 가지 키 복구 시스템 중에서 대표적인 몇 가지와 각 시스템에 대한 취약점 및 공격들을 살펴보았으며, 이는 사용자들이 보다 안심하고 사용할 수 있고 유사시에는 확실한 키 복구가 가능한 시스템을 개발하는데 있어서 필수적으로 고려되어야 할 사항이다.

## 참 고 문 헌


[1] "Requirements for key recovery products", Report of the TACD- FIPSFKMI (Technical Advisory Committee to Develop a Federal Information Processing Standard for the Federal Key Management Infrastructure), 1998.

[2] Dorothy E. Denning, "A Taxonomy for Key Escrow Encryption Systems", The Communications of the ACM, Vol.39, No.3, 1996.

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

[4] Matt Blaze, "Protocol Failure in the Escrowed Encryption Standard", The 2nd ACM Conference on Computer and Communications Security, pp. 59-67, 1994.

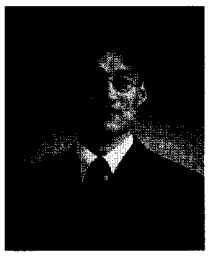
[5] Yair Frankel and Moti Yung, "Escrow Encryption Systems Visited : Attacks, Analysis and Designs", Crypto'95, Springer-Verlag, Lecture Notes in Computer Science, LNCS 963, pp.223-235, 1995.

[6] David M. Balenson, Carl M. Elli son, 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, pp. 180-207, 1995.

[7] Stephen T. Walker, Steven B. Lipner, Carl M. Ellison, Dennis K. Branstad and David M. Balenson, "Commercial Key Escrow : something for everyone now and for the future", TIS Report

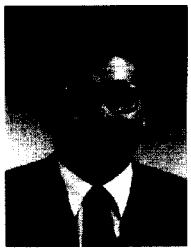
no. 541, Trusted Information Systems Inc., 1995.
[8] T. Beth, H. J. Knobloch, M. Otten, G. Simmons and P. Wichmann, "Towards Acceptable Key Escrow Systems", The 2nd ACM Conference on Computer and Communications Security, pp. 51-58, 1994.
[9] Patrick Horster, Markus Michels and Holger Petersen, "A New Key Escrow System with Active Investigator", Proc. of Securicom, Paris, La Defense, 1995.
[10] S. J. Kim, I. S. Lee, M. Mambo and S. J. Park, "On the Difficulty of Key Recovery Systems", Proc. of ISW'99, Information Security Workshop, Springer-Verlag, 1999.
[11] Sivio Micali, "Fair public-key cryptosystems", Crypto'92, Springer-Verlag, Lecture Notes in Computer Science, LNCS 740, pp. 113-138, 1992.
[12] Joseph Kilian and Tom Leighton, "Fair Cryptosystems, Revisited", Crypto'95, Lecture Noted in Comput- er Science Vol. 963, Springer-Verlag, 1995.
[13] Eric Verheul and Henk C.A. van Tilborg, "Binding ElGamal : A Fraud Detectable Alternative to Key-Escrow Proposals", Eurocrypt'97, Springer-Verlag, Lecture Notes in Computer Science, LNCS 1233, pp. 119-133, 1997.
[14] Birgit Pfitzmann and Michael Waidner, "How to Break Fraud Detectable Key Recovery", ACM Operating Systems Review 32, 1998.
[15] Ross Anderson and Michael Roe, "The GCHQ Protocol and its Problems", Eurocrypt'97, Springer-Verlag, Lecture Notes in Computer Science, LNCS 1233, pp. 134-148, 1997.
[16] Adi Shamir, "Partial key escrow : A new approach to software key escrow", Key escrow conference, 1995.
[17] Mihir Bellare and Shafi Goldwasser, "Verifiable Partial Key Escrow", The 4th ACM Conference on Computer and Communications Security, 1997.
[18] 채승철, 이임영, "안전한 키 위탁 시스템에 관한 연구", 통신정보보호학회 논문지, 제 9권 2호, pp. 83-91, 1999.

## 〈著者紹介〉



**유 준 석 (Joon-suk Yu)**
1999년 : 성균관대학교 정보공학과 졸업
1999년~현재 : 성균관대학교전기전자 및 컴퓨터 공학부 석사과정
〈관심분야〉 암호이론

**이 인 수 (In-soo Lee)**
1993년 : 연세대학교 수학과 졸업
1997년 : 연세대학교 수학과 석사
1996년~현재 : 한국정보보호 센터 연구원
〈관심분야〉 암호이론, 암호분석



**원 동 호 (Dong-Ho Won)**
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
1982년~현재 : 성균관대학교 전기전자 및 컴퓨터 공학부 교수
1999년~현재 : 성균관대학교 전기전자 및 컴퓨터 공학부장
(겸)정보통신대학원장, 한국통신정보보호학회 부회장
〈관심분야〉 암호이론, 부호이론

**김 병 천 (Byung-chun Kim)**
1993년 : 성균관대학교 정보공학과 졸업
1995년 : 성균관대학교 정보공학과 석사
1995년~1996년 : 한국전산원연구원
1996년~현재 : 한국정보보호센터 선임연구원
〈관심분야〉 암호이론, 전자서명

**박 성 준 (Sung-jun Park)**
1983년 : 한양대학교 수학과 졸업
1985년 : 한양대학교 수학과 석사
1996년 : 성균관대학교 정보공학과 박사
1985년~1994년 : 한국전자통신연구소 선임연구 원
1996년~현재 : 한국정보보호센터 기반기술 팀장
〈관심분야〉 암호이론, 계산이론, 정보이론