공개키 기반 구조의 암호 통신을 위한 키 복구 프로토콜

○
전은아, 유형준, 이강수
한남대학교 컴퓨터공학과

A Key Recovery Protocol for Cryptographic Communication on Public Key Infrastructure

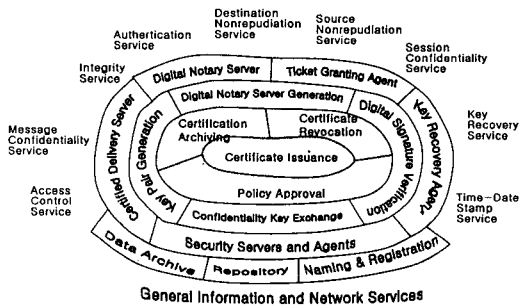
Eun-A Jun, hyung-Joon You, Gang-Soo Lee
Department of Computer Engineering, Hannam University

요약

인터넷의 사용 증가와 더불어 보안의 중요성이 증가되면서, 인터넷을 통한 전자상거래에서의 안전성 및 신뢰성 확보를 위한 기술의 필요성이 커지게 되었다. 두 사용자간의 안전한 문서의 전송과 상호 인증을 보장은 국가 기관이나 산업계, 또는 개인이 인터넷을 이용한 안전한 통신을 가능하게 한다. 정보보호 기술의 중요 요소로 부각되고 있는 공개키 기반 구조 (PKI : Public Key Infrastructure)는 인증(authentication), 기밀성(confidentiality), 무결성(integrity), 부인봉쇄(non-repudiation) 등의 보안의 기본 요소를 제공하고 있다. 공개키 기반 구조에서의 공개키/개인키쌍은 상대방에 대한 신뢰와 자신의 정보 보호를 위해 사용되는 도구이며, 이 키를 잃어 버릴 경우 암호 및 인증 서비스를 제공받을 수 없게 된다. 이를 위한 대비책으로 키 복구 기술(key recovery agent)이 필요하게 되었다. 우리는 본 연구를 통해서 키 복구 기술을 적용한 키 복구 시스템을 개발하였으며, 이를 통해서 키 복구 시스템의 프로토콜을 제안하고 검증하였다.

1. 서론

인터넷의 확산과 정보 기술의 발달로 인한 거대한 정보흐름은 전통적인 유통 체계에서 전자상거래라는 가상공간에서의 거대 시장을 형성 시켰다. 전자상거래에서의 주도권 확보를 위한 노력으로, 세계 각국은 정보서비스 산업 경쟁력을 강화시키고, 전자상거래 지원을 위한 각종 법과 정·책을 추진하고 있다[1]. 선진국에서는 전자상거래의 핵심 기술인 공개키 기반구조(PKI:Public Key Infrastructure) 구축을 추진하고 있는데, 미국의 FPKI(Federal Public Key Infrastructure), 유럽의 ICE-TEL (Internet -working Public Key Certification Infrastructure), 캐나다의 GOC PKI (Government of Canada PKI), 호주의 PKAF(Public Key Authentication framework)등이다[1]. 우리나라에서도 한국정보보호센터를 중심으로 활발히 진행되고 있다.



[그림 1] PKI 중심의 인증서 기반 보안 서비스

PKI에서의 보안 서비스로 접근 제어 서비스(access control service), 기밀성 서비스(message confidentiality service), 메시지 무결성 서비스(integrity service), 인증 서비스(authentiction sevice), 수신자 부인봉쇄 서비스(destination nonrepudiation service), 송신자 부인봉쇄 서비스(source nonrepudiation service), 세션 무결성 서비스(session confidentiality service), 키 복구 서비스(key recovery service), 타임스탬프 서비스(time-date stamp service)를 제공하고 있다.

본 논문에서는 2장에서 공개키 기반 구조에서의 키 복구 서비스를 소개하고 3장에서 신뢰성 있는 안전한 통신을 위해서 제공되는 키 복구 프로토콜를 제시하고 검증하였으며, 마지막으로 4장에서 결론을 내렸다.

2. 키 복구 시스템

키 복구 기술이란 암호 시스템에서 키를 가지고 암호문을 복호하는 정상적인 절차 외에 다른 방법을 통해서 유사시에 암호문을 안전하게 복호할 수 있는 방법이다. 키 복구 방식은 크게 키 위탁 방식(key escrow)과 키 캡슐화(key encapsulation)방식이 있다.[4]

[표 1] 키 복구 방식

복구 방식		설 명
Key Escrow	Escrow	사용자의 비밀키의 전부 또는 일부를 하나 또는 그 이상의 신뢰기관에 위탁하는 방식
	TTP(Trusted-Third party)	신뢰할 수 있는 기관을 키 분배 센터로 지정하여 이 기관에 암호학적 세션키를 생성, 분배하는 방식
	상업적 키 백업	내부적인 신뢰기관을 이용하여 사용자의 비밀키의 사본을 저장
Key Encapsulation		키 복구 필드를 생성해서 데이터에 부가하는 방식, 비밀키의 정보는 포함하지 않으며, DRF(Data Recovery Field)가 부가된 해당 데이터만을 복구

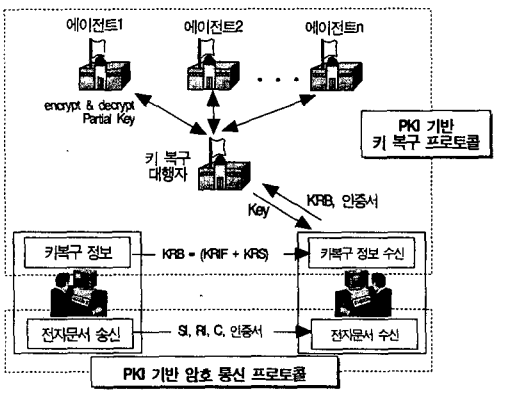
키 위탁방식은 비밀키의 전부 또는 일부를 신뢰받는 제 3자(trusted third party)에게 위탁하는 방식으로 유사시에 키를 확실하게 얻을수 있는 반면에, 신뢰받는 제 3자의 신뢰도에 많은 영향을 받는다. 또한 위탁한 키의 안전한 보관과 관리 문제, 키가 법률에 의해 합법적인 도청 목적으로 공개되었을 경우에 키의 사용기간의 제한 등이 문제가 된다.

키 캡슐화 방식은 각각의 메시지 전송 또는 파일 저장시 마다 키 복구 필드를 생성해서 해당 암호문에 키 복구를 할 수 있는 정보를 부가하는 방식으로 유사시에 키의 복구가 필요한 경우에 복구 기관이 가지고 있는 복구키를 이용해서 키 복구 필드를 복호화 후 해당키를 얻을수 있다. 이 방식의 장점은 복구할 수 있는 정보가 사용자의 비밀키가 아니라 한시적으로 사용되는 세션키이기 때문에 합법적인 도청시에 도청 기간의 제한이 불가능하다는 것과 모든 복구 정보를 특정 기관에 위탁하지 않고 사용자의 데이터와 함께 존재하기 때문에 사용자의 입장에서는 보다 안전하다는 확신을 가질수 있다는 것이다. 또한 키 복구 필드에 들어가 있는 정보를 일정 수준으로 제한함으로써 복구 기관의 능력을 조절할 수 있는 장점을 가진다.[4]

본 논문에서는 사용자에게 더 많은 복구 권한을 부여한 키 캡슐화 방식의 키 복구 프로토콜을 제시하였으며, PKI의 암호/인증 서비스 제공을 위해 RSA, DES, SHA-1 알고리즘과 한국형 전자 서명 알고리즘인 KCDSA[8]을 사용하였다.

3. PKI 기반 키 복구 프로토콜

본 연구에서 제시한 키 복구 시스템은 상호 암호 통신을 수행하는 두 사용자 시스템(User Agent; KRR)과 키 복구 대행자 시스템(Key Recovery Center System : KRC System), 그리고 키 복구 에이전트 시스템(Key Recovery Agent : KRA System)으로 [그림 2]와 같이 구성된다. 암호 통신을 수행하는 도중에 발생할 수 있는 키의 손망실시 사용자는 KRC에게 키 복구를 요청함으로써 메시지를 안전하게 획득할 수 있다. KRA는 KRC의 견제자 역할을 수행하며, 하나 이상 존재할 수 있다. KRC는 KRA를 통해서만 세션키 복구가 가능하다.



- SI (Signature Information) : 메시지 송신자의 인증 정보
- RI (Recipient Information) : 메시지 수신자의 암호 정보
- C (Cipher Message) : 암호화된 메시지
- KRB (Key Recovery Block) : 키 복구를 위한 부가 정보
- Key : 암호문을 복구할 수 있는 키
- Partial Key : KRA에 의해 복구가능한 부분 키

[그림 2] 키 복구 시스템 구성

위 [그림2]의 키 복구 시스템은 상세한 절차는 3.1 절의 PKI 기반 암호 통신 프로토콜, 3.2 절의 PKI 기반 키 복구 프로토콜로써 설명하고자 한다. 프로토콜의 설명과 검증을 위해서 다음의 표기 규칙을 적용하기로 한다.

알고리즘과 데이터와의 관계는 다음과 같이 표기한다.
생성물 = 알고리즘 { 알고리즘을 위한 객체, ... }

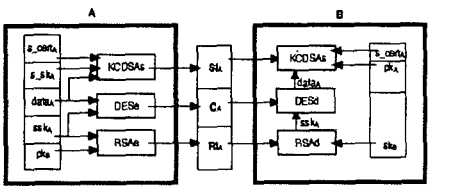
프로토콜의 검증을 위해서 스택을 사용하였다. 한 객체에 대해서 Push/Pop쌍이 하나씩 존재하며, 같은 객체가 반복 사용될 경우에는 *이탤체*를 사용하여 표기하였다. empty stack 가 나오면 검증이 올바르게 되었음을 나타낸다.
시스템.stack of Push = Push(Pop(객체, ...)) ⇒ 생성물

3.1 PKI 기반 암호 통신 프로토콜

PKI 기반 암호 통신 프로토콜은 키 복구 시스템에서의 상호 인증과 암호 통신을 위한 프로토콜이다. 메세지(메일, 복구 요청서등) 전달시, 또는 인증서에 대한 신뢰된 기관의 서명 검증시에 사용되며, 무결성, 기밀성, 인증 서비스를 제공해 준다.

PKI 기반 암호 통신 프로토콜은 다음과 같은 절차에 의해 이루어 진다.
[메시지 송신자]
① SI = KCDSAs{ s_cert_A, s_sk_A, $data_A$ }
② C = DESe{ $data_A$, ssk_A }
③ RI = RSAe{ ssk_A, pk_B }
[메시지 수신자]
④ 세션키 복구 = RSAd{ RI, sk_B }
⑤ 메시지 복구 = DESd{ C, ssk_A }
⑥ 서명 검증 = KCDSAv { SI, pk_A, s_cert_A, $data_A$ }

PKI 암호 통신 프로토콜은 다음과 같이 정의 할 수 있다.
KCDSAv { SI = **KCDSAs**{ s_cert_A, s_sk_A, $data_A$ }, pk_A, s_cert_A, $data_A$ = **DESd**{ C = **DESe**{ $data_A$, ssk_A }, ssk_A = **RSAd**{ RI = **RSAe**{ ssk_A, pk_B }, sk_B }}}

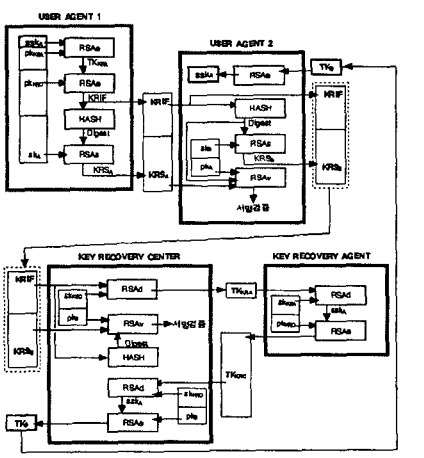


- s_cert* (Certificate for Signature using KCDSA Algorithm of *) : KCDSA 알고리즘을 사용하는 사용자의 공개키 인증서
- s_sk* : KCDSA 알고리즘을 사용한 서명을 위한 *의 개인키
- data* : 송신자 *의 메시지
- KCDSAs/v : KCDSA 알고리즘을 사용한 서명/서명검증
- DESe/d : DES 알고리즘을 사용한 암호/복호

[그림 3] PKI 기반 암호 통신 프로토콜

위에서 정의된 키 복구 시스템의 암호 통신 프로토콜을 스택을 통해 검증해 보면 다음과 같다.
step 1. A.KCDSAs : Push(Push(Push(Pop(Pop(Pop(s_cert$_A$, s_sk$_A$, data$_A$)))))) ⇒ SI
step 2. A.DESe : Push(Push(Pop(*Pop*(*data*$_A$, ssk$_A$)))) ⇒ C
step 3. A.RSAe : Push(Push(Pop(*Pop*(*ssk*$_A$, pk$_B$)))) ⇒ RI
step 4. B.RSAe : Push(Push(Pop(Pop(RI, sk$_B$)))) ⇒ ssk$_A$
step 5. B.DESe : Push(Push(Pop(Pop(C, ssk$_A$)))) ⇒ data$_A$
step 6. B.KCDSAv : Push(Push(Push(Push(Pop(Pop(Pop(Pop(SI, pk$_A$, s_cert$_A$, data$_A$)))))))) ⇒ 서명 검증 ⇒ empty stack

3.2 PKI 기반 키 복구 프로토콜



- ssk$_A$ (session key of A) : 복구될 세션키
- pk*/sk* (public/private key of *) : *의 공개키/개인키
- TK* (target key of *) : *를 통해서만 복구되는 키
- KRIF (Key Recovery Information Field) : 키 복구를 위한 정보 필드
- KRS* (Key Recovery information Signature of *) : KRIF의 무결성 정보
- Digest : 해쉬 함수에 의해 생성된 다이제스트
- RSAe/d/s/v : RSA 알고리즘을 사용한 암호/복호/서명생성/서명검증
- HASH : 해쉬 알고리즘을 사용한 다이제스트 생성

[그림 4] PKI 기반 키 복구 프로토콜

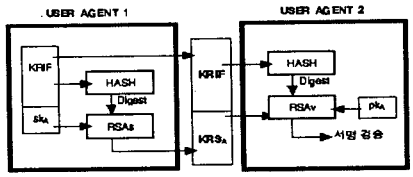
송신자는 키 복구를 위한 정보인 KRB(KRIF+KRS)를 생성하여 SI, RI,

C, 인증서(certificate)와 함께 송신한다. 수신자 개인키의 손망실시 경우 수신자는 수신된 KRB를 KRC에 송신하여 키 복구를 요청할 수 있다. KRIF는 세션키를 KRA의 수 만큼 분리하여 각 KRA의 공개키로 암호화하고 이를 다시 KRC의 공개키로 암호화하여 생성되며, KRS는 KRIF와 기타 정보를 포함한 데이터에 대한 무결성 값이다. KRS는 KRIF에 대한 임의적인 변경을 방지하는 목적으로 사용된다. KRC는 수신된 KRB로부터 KRIF를 추출하여 자신의 개인키로 복호하고, 이것을 각 KRA에 송신한다. KRC로부터 부분 세션키를 수신한 KRA는 자신의 개인키로 부분 세션키를 복호하여 KRC에 송신하면, KRC를 각 KRA로부터 수신된 부분 세션키들을 조합하여 키 복구를 요청한 수신자에게 전달하게 된다.

키 복구를 위해서는 KRB의 무결성 검증이 필요하며, 검증이 된 KRB로부터 추출된 복구 정보(KRIF)로부터 키 복구가 가능하다

3.2.1 KRB의 무결성 검증 프로토콜

KRB는 키 복구 정보인 KRIF와 OID, 날짜·시간등의 기타정보 그리고 무결성 정보를 포함한다. KRB의 무결성 검증은 KRIF에 대한 임의적인 변경을 방지한다.



[그림 5] KRB 무결성 검증 프로토콜

KRB의 무결성 검증 프로토콜은 다음과 같은 절차에 의해 이루어진다.

① Digest = **HASH** { KRIF }
② KRS_A = **RSAs** { Digest, sk_A }
③ 무결성 검증 = **RSAv** { Digest, KRS_A, pk_A }

키 복구 시스템의 KRIF 무결성 검증은 다음과 같이 정의 할 수 있다.

RSAv { Digest = **HASH** { KRIF }, KRS_A= **RSAs** { Digest = **HASH** { KRIF }, sk_A }, pk_A }

위에서 정의된 KRB 무결성 검증 프로토콜을 스택을 통하여 검증하여 보면 다음과 같다.

step 1. UA1.HASH :Push(Pop(KRIF)) ⇒ Digest
step 2. UA1.RSAs : Push(Push(Pop(Pop(Digest, sk_A)))) ⇒ KRS_A
step 3. UA2.HASH : Push(Pop(KRIF)) ⇒ Digest
step 4. UA2.RSAv : Push(Push(Push(Pop(Pop(Pop (Digest, pk_A, KRS_A))))
⇒ 서명 검증 ⇒ empty stack

3.2.2 KRIF를 이용한 키 복구 프로토콜

송신자에 의해 생성된 KRIF는 수신자의 키 복구 요청시에 사용된다. KRC는 3.2의 프로토콜을 통해 무결성이 검증된 KRB로부터 KRIF를 추출하여 키 복구를 수행한다.

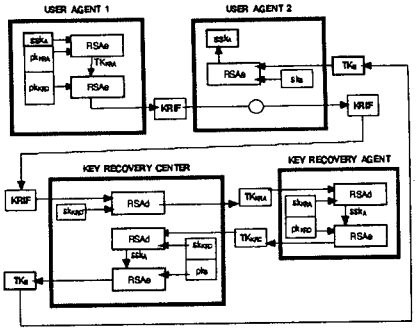
키 복구 프로토콜은 다음과 같은 절차에 의해 이루어진다.

① $TK_{\cdot}$ = **RSAe** { ssk_A, $pk_{\cdot}$ }
② KRIF = **RSAe** { TK_{KRA}, pk_{KRC} }
③ 세션키 복구 = **RSAd** { **RSAd** { KRIF, sk_{KRC} }, sk_{KRA} }

키 복구 시스템의 키 복구는 다음과 같이 정의할 수 있다.

ssk_A = **RSAd** { TK_B = **RSAe** { ssk_A = **RSAd** { TK_{KRC} = **RSAe** { ssk_A = **RSAd** { TK_{KRA} = **RSAd** { KRIF = **RSAe** { TK_{KRA} = **RSAe** { ssk_A, pk_{KRA} }, pk_{KRC} }, sk_{KRC} }, sk_{KRA} }, pk_{KRC} }, sk_{KRC} }, pk_B }, sk_B }

위에서 정의된 키 복구 시스템의 키 복구 프로토콜을 스택을 통하여 검증하여 보면 다음과 같다.



[그림 6] KRIF를 이용한 키 복구 프로토콜

step 1. UA1.RSAe : Push(Push(Pop(Pop(TK_{KRA} = Push(Push(Pop(Pop (ssk_A, pk_{KRA}))))), pk_{KRC})))) ⇒ KRIF
step 2. UA2 : no operation
step 3. KRC.RSAd : Push(Push(Pop(Pop(KRIF, sk_{KRC})))) ⇒ TK_{KRA}
step 4. KRA.RSAd : Push(Push(Pop(Pop(KRIF, sk_{KRC})))) ⇒ ssk_A
step 5. KRA.RSAe : Push(Push(Pop(Pop(ssk_A, pk_{KRC})))) ⇒ TK_{KRC}
step 6. KRC.RSAd : Push(Push(Pop(Pop(KRIF, sk_{KRC})))) ⇒ ssk_A
step 7. KRC.RSAe : Push(Push(Pop(Pop(ssk_A, pk_B)))) ⇒ TK_B
step 8. UA2.RSAd : Push(Push(Pop(Pop(TK_B, sk_B)))) ⇒ ssk_A
⇒ empty stack

4. 결론 및 향후 연구 방향

본 연구의 결과는 공개키 기반 구조 시스템에서의 전자상거래를 위한 키 복구 프로토콜을 제안하고, 검증하였다. 본 키 복구 프로토콜은 키 위탁 방식(key escrow)의 커다란 단점인 개인의 프라이버시 침해를 방지하면서도 키 복구 정보(KRB)를 둠으로써 국가의 범죄 수사시에도 접근 가능하도록 하였다. 본 연구에서 제안한 키 복구 프로토콜을 사용을 위해서는 KRA수가 보안성을 위한 변수로 작용되는데, 이는 효율성과 신뢰성 측면에서도 고려할 필요성이 있다. 또한, 키 복구 프로토콜을 실제적으로 적용하기 위해서는 네트워크 컴퓨팅 환경과 다양한 암호 환경을 고려해야 하며, 개인의 프라이버시 보장과 정당한 권한에 의한 접근 허가만을 허용을 보장해야 한다. 보다 안전하고 신뢰성있는 키 복구 시스템을 개발하기 위해서는 더 많은 연구가 이루어져야 할 것으로 본다.

참고 문헌

[1] 통신정보보호학회지 제8권 제 3호 "PKI 특집", 1998. 9. 30
[2] 한국정보보호센터, "국외 공개키 기반 주조 추진 체계 분석", 김지원, 1998.7
[3] NIST, "Public Key Infrastructure Study" Final Report, Gaithersburg, MD, April, 1994
[4] 이임영, 채승철, "Key Recovery 시스템에 관한 고찰", 한국통신정보보호학회, Vol. 7, No. 4, pp. 45~58, 1997. 12.
[5] Hal. Abelson, *et al.*, "The Risks of Key Escrow and Trusted Third-Party Encryption," *http://www.cdt.org/crypto /risks98/risks98.pdf*, 1997.5.
[6] NIST, "Requirements For Key Recovery Products," Final Report, *http://csrc.nist.gov/key- recovery/*, Nov. 1998.
[7] 최용락, 소우영, 이재광, 이임영, "통신망 정보 보호", 그린 출판사 출판번호 8-161호, 1996. 2. 20
[8] 한국정보보호센터, 성균관대학교, "전자결재용 서명 고속화를 위한 32비트 고속연산 S/W 개발", 최종연구보고서 97-02, 1997. 9