情 報 保 護 學 會 誌
第 12 券 第 4 號, 2002. 8

국외 키로밍 제품 개발 현황 분석

박 해 룡*, 권 현 조*, 김 지 연*, 김 승 주*

요 약

키로밍 기술은 사용자가 별도의 저장매체 없이도 인터넷이 연결되는 모든 단말기에서 키로밍서버로부터 자신의 개인키를 다운받아 암호서비스를 이용할 수 있는 기술로 사용자에게 편리한 이동성을 제공하여 현재 많은 주목을 받고 있는 기술이다. 본 논문에서는 국외 주요 정보보호업체 키로밍 제품 분석을 통하여 키로밍 기술동향을 살펴보고자 한다.

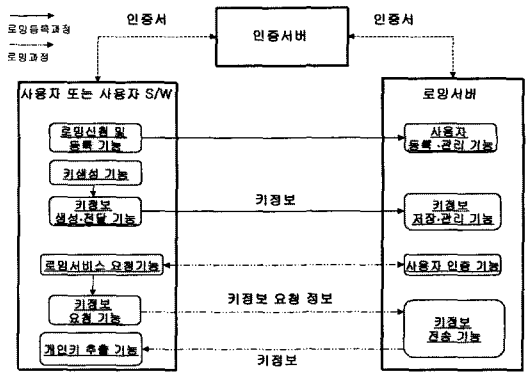
Ⅰ. 서 론

암호기술은 정보의 유출을 방지하고 상대의 신원확인을 가능하게 함으로서 인터넷 뱅킹, 보안전자메일 등과 같은 암호서비스를 가능하게 하는 등 많은 장점을 가지고 있다. 이러한 암호서비스에 사용되는 키는 사람이 쉽게 기억할 수 있는 형태가 아니기 때문에 일반적으로 자신의 PC의 하드디스크나 스마트카드 등의 저장매체에 저장된다. 그러나 키가 저장되어 있는 저장매체를 손·망실(損·亡失)한 경우나 자신의 개인키가 저장되어 있지 않는 PC에서는 암호서비스를 수행할 수 없는 불편함이 생길 수 있다. 키로밍(Key Roaming) 기술이란, 신뢰할 수 있는 서버에 자신의 암호서비스에 필요한 키를 저장한 후, 서버로부터 키를 다운받아 이용할 수 있도록 한 기술로, 저장매체의 손·망실 문제를 해결하고 사용자에게 이동성을 제공한다.

본 논문에서 주요 정보보호업체의 국외 키로밍 제품 을 살펴보도록 한다. 본 논문의 구성은 다음과 같다. 2장에서는 키로밍 기술을 일반화하여 기능중심의 키로밍 모델을 제시하고, 이에 대한 각 기능을 간략히 서술하였다. 3장에서는 국외 주요 키로밍 제품의 기능을 키로밍을 위한 등록과정과 키로밍 과정으로 구분하여 상세히 기술하고 이들 각 제품의 특징을 언급하였다. 4장에서는 3장에서 언급한 키로밍 제품들의 특징에 대한 내용을 표로 정리하였다. 마지막으로 5장에서는 결론을 맺도록 한다.

Ⅱ. 모 델

키로밍 기술을 일반화한 키로밍 모델은 다음과 같다.



〔그림 1〕 키로밍 모델

그림 1에서 키생성 기능 및 키정보 생성·전달 기능은 키로밍 제품에 따라 로밍서버 또는 인증서버에서 동작할 수 있으며, 개인키 추출기능도 제품에 따라 로밍서버에서 동작할 수 있다.

1. 구성 객체

- 사용자 또는 사용자 S/W : 키로밍서비스를 이용하는 사용자 또는 사용자가 이용하는 S/W

* 한국정보보호진흥원(KISA) ({hrpark, hckwon, jykim, skim}@kisa.or.kr)

○ 로밍서버 : 사용자의 키정보를 저장·관리하는 서버
○ 인증서버 : 사용자의 로밍되는 개인키에 대응되는 공개키의 인증서를 발행·관리하는 서버
○ 저장장소 : 로밍에 사용될 사용자의 키 또는 관련 정보를 저장하는 장소
  - DB : 로밍서버나 인증서버만이 접근할 수 있는 저장장소
  - 디렉토리 : 로밍서버나 인증서버뿐만 아니라 사용자도 접근할 수 있는 저장장소

2. 주요 기능

○ 사용자 등록 기능 : 사용자 정보(로밍서비스 요청시 사용자를 인증할 수 있는 정보 등을 포함)를 로밍서버에게 전달하는 기능
○ 사용자 등록·관리 기능 : 사용자 정보를 등록·관리하는 기능
○ 키생성 기능 : 사용자가 암호서비스(전자서명, 암호화 등)에 사용될 키를 생성하는 기능
○ 키정보 생성·전달 기능 : 사용자가 암호서비스에 사용할 정보(키정보)를 생성하여 로밍서버에게 전달하는 기능
○ 키정보 저장·관리 기능 : 키정보를 안전하게 저장·관리하는 기능
○ 로밍서비스 요청 기능 : 로밍서버에게 로밍서비스를 요청하는 기능
○ 사용자 인증 기능 : 저장된 사용자 정보를 이용하여, 로밍서비스를 요청한 자를 인증하는 기능
○ 키정보 요청 기능 : 로밍서버에 키정보를 요청하는 기능
○ 키정보 전송 기능 : 정당한 키정보 요청에 대해, 로밍서버가 키정보를 안전하게 전송하는 기능
○ 개인키 추출 기능 : 키정보로부터 암호서비스 키 등을 추출하는 기능

Ⅲ. 국외 키로밍 제품

본 장에서는 국외 키로밍 제품별 등록 과정 및 키로밍 과정을 살펴보도록 한다. 먼저 각 회사별 제품명은 표 1과 같다.

본 장에서 사용할 공통적인 시스템 파라메터(Parameter) 는 다음과 같다.

○ ID : 사용자의 ID
○ PWD : 사용자의 패스워드

〔표 1〕 회사별 제품명

제품 / 구성 개체	Hush Enterprise	UniCERT Roaming	Entrust Roaming	VeriSign Roaming Service	RSA Keon Web PassPort Server	EasySign	KISA의 Key Roaming Service
사용자 또는 사용자 소프트웨어	• New User Computer Program • Enabler Computer Program (특허[6]에 언급된 명칭임)	• UniCERT Roaming Applets	• Entrust /Entelligence 5.0	• PTA(Personal Trust Agent)	• Web PassPort Plug-in	언급없음	언급없음
로밍서버	• Hush Key Server	• UniCERT Roaming Server Administrator • UniCERT Roaming Server • UniCERT Roaming PEK (Protection Encryption Key) server	• Entrust /Profile Server	• Roaming Server • Roaming/Storage Server	• Web PassPort Server	• Signature Server	• 키서버i (i=1,2,…,n) • 응용서버
인증서버	• Hush CA	• Key and Certificate Administrator	• Entrust /Authority 5.0	• OnSite 4.5	• RSA Keon CA (OneStep module)	• Certificate Authority	언급없음
저장장소	• Private Key DB • Public Key DB • DB	• LDAP Server	• LDAP Directory	• Roaming DB • Roaming/Storage DB	• LDAP Directory	언급없음	• 응용서버의 DB • 키서버i의 DBi (i=1,2,…,n)

- o PRI, PUB : 사용자의 개인키/공개키 쌍
- o CertA : 사용자의 인증서
- o E : 대칭키 암호화 알고리즘
- o D : 대칭키 복호화 알고리즘
- o Γ : 공개키 암호화 알고리즘
- o Δ : 공개키 복호화 알고리즘
- o h : 해쉬 알고리즘

1. Hush사의 Hush Enterprise 제품

Hush사의 저장장소는 다음과 같이 세 가지 개체로 구성되어 있다.[5,6]

- Private Key DB : (Private alias, EPWD (PRI))가 저장되어 있는 개체
- Public Key DB : CertA가 저장되어 있는 개체
- DB : h(PWD)가 저장되어 있는 개체

1.1 등록 과정[3,5,6]

사용자와 로밍서버의 통신시 보안 채널로 SSL을 사용한다.

① 로밍서버 → 사용자 : 사용자는 로밍서버에 접속하여 사용자 S/W를 다운받는다.

② 사용자 : 사용자는 PRI, PUB, PWD를 생성한다.

③ 사용자 → 로밍서버 : 사용자는 ID, PUB와 함께 $E_{PWD}(PRI)$, h(*PWD*), Private alias를 다음과 같이 생성해서 로밍서버에게 전송하고 인증서 발행을 요청한다.

㉠ Private alias를 생성하는 과정은 다음과 같다.

단계1. $salt$ = **SHA1**(*ID*) mod 2^{64}

단계2. *sessionkey* = **S2K**(PWD, salt, 220)(단, S2K는 RFC 2440에 서술된 Iterated String To Key algorithm임) [30]

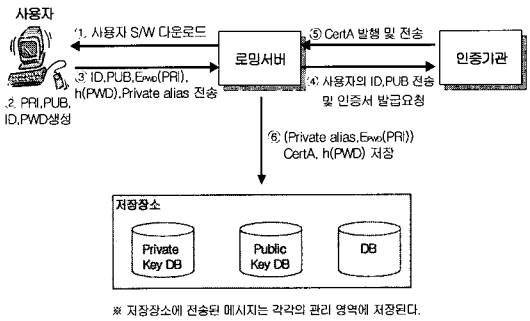
단계3. *private alias* = **SHA1**(*sessionkey*)

④ 로밍서버 → 인증서버 : 로밍서버는 인증서버에게 사용자의 ID와 PUB를 전송하고, 사용자의 인증서 발행을 요청한다.

⑤ 인증서버 → 로밍서버 : 인증서버는 CertA를 발행하고, CertA를 로밍서버에게 전송한다.

⑥ 로밍서버 → 저장장소 : 로밍서버는 (Private alias, EPWD(PRI))는 Private Key DB에, CertA는 Public Key DB에 h(PWD)는 자신의 DB에 각각 저장한다.

Hush Enterprise 제품의 키로밍을 위한 등록 과정 흐름도



※ 저장장소에 전송된 메시지는 각각의 관리 영역에 저장된다.

1.2 키로밍 과정[3,5,6]

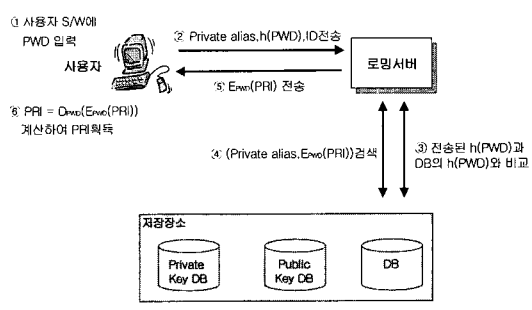
사용자와 로밍서버의 통신시 보안 채널로 SSL을 사용한다.

① 사용자 : 사용자는 다운로드된 사용자 S/W에 PWD를 입력한다.

② 사용자 → 로밍서버 : 사용자는 Private alias, h(*PWD*), ID를 로밍서버에게 전송하고 키로밍을 요청한다.

③ 저장장소 → 로밍서버 : 로밍서버는 사용자가 전송한 h(*PWD*)와 자신의 DB에 저장된 h(*PWD*)를 비교한다.

Hush Enterprise 제품의 키로밍 과정 흐름도



④ 저장장소 → 로밍서버 : ③단계를 통과하면,

로밍서버는 사용자가 전송한 Private alias를 이용하여 Private Key DB로부터 $E_{PWD}(PRI)$를 검색한다.

⑤ 로밍서버 → 사용자 : 로밍서버는 사용자에게 $E_{PWD}(PRI)$을 전송한다.

⑥ 사용자 : 사용자는 $PRI = D_{PWD}(E_{PWD}(PRI))$를 계산하여 자신의 PRI를 획득한다.

1.3 제품의 특징

Hush Enterprise 제품의 경우에 사용자의 ID와 PWD를 이용해서 생성한 "Private alias"를 암호화된 개인키를 검색하기 위한 인덱스로 사용하기 때문에 공격자는 Private Key DB로부터 개인키와 사용자간의 연결 정보(binding)를 직접적으로 알아 낼 수 없다. 그러나, 로밍서버만이 키로밍 권한을 가지고 있고, 로밍서버가 패스워드 확인자를 이용한 사용자처럼 행동할 수 있는 문제점이 있다. 또한 로밍시 별도의 보안 프로토콜이 없으면 전송되는 패스워드 확인자를 이용한 패스워드 추측 공격이 가능하다.

2. Baltimore사의 UniCERT Roaming 제품

UniCERT Roaming 제품은 로밍서버가 다음과 같이 세 가지로 구성되어 있다.

- 로밍서버 관리자(Roaming Server Administrator) : 사용자의 로밍 정보를 생성하는 서버
- 로밍서버(Roaming Server) : 사용자의 키 정보를 저장·관리하는 서버
- 로밍PEK서버(Roaming PEK(Protection Encryp -tion Key) Server) : 키로밍서비스 제공 권한을 가진 서버

[시스템 파라메터]

- .p12 : ($E_{PWD}(PRI)$, CertA)를 포함하는 파일
- CertP : 로밍PEK서버의 인증서
- PPUB : 로밍PEK서버의 공개키
- PPRI : 로밍PEK서버의 개인키
- RAK(Random Authentication Key) : 키로밍 요청시 사용되는 인증키로 해쉬한 패스워드 값을 이용하여 로밍서버 관리자가 생성
- REK(Random Encryption Key) : 각 사용자의 .p12 파일 암호화에 사용되는 키로 로밍서버 관리자가 생성
- ARN(Access Request Number) : 키로밍 서비스를 요청한 횟수
- OSK(One-time local Session Key) : 사용자가 로밍서버로부터 해당 키정보를 안전하게 다운받기 위한 세션키

2.1 등록 과정[7,8]

① 사용자 → 인증서버 : 사용자는 인증서버에게 사용자 등록을 신청한다.

② 인증서버 : 사용자의 PWD, PRI, PUB 및 CertA를 생성한다.

③ 인증서버 → 저장장소 : 인증서버는 CertA를 저장장소에 저장한다.

④ 인증서버 → 로밍서버 관리자 : 인증서버는 로밍서버 관리자에게 사용자의 .p12를 전송한다.

⑤ 로밍서버 관리자 ↔ 로밍서버 : 로밍서버 관리자는 로밍서버에게 CertP를 요청하고 전송받는다.

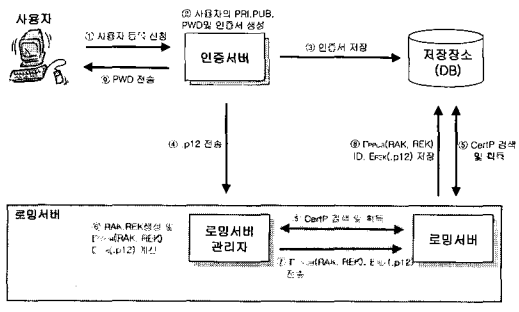
⑥ 로밍서버 관리자 : 로밍서버 관리자는 RAK와 REK를 생성하고, $\Gamma_{PPUB}(RAK, REK)$, $E_{REK}(.p12)$를 계산한다.

⑦ 로밍서버 관리자 → 로밍서버 : 로밍서버 관리자는 $\Gamma_{PPUB}(RAK, REK)$, $E_{REK}(.p12)$를 로밍서버에게 전송한다.

⑧ 로밍서버 → 저장장소 : 로밍서버는 저장장소에 (ID, $\Gamma_{PPUB}(RAK, REK)$, $E_{REK}(.p12)$)를 저장한다.

⑨ 인증서버 → 사용자 : 인증서버는 PWD를 사용자에게 전송한다.

UniCERT Roaming 제품의 키로밍을 위한 등록 과정 흐름도



2.2 키로밍 과정[7,8]

① 사용자 : 사용자는 사용자 S/W에 ID와 PWD를 입력한다.

② 사용자 → 로밍서버 : 사용자는 로밍서버에게 h(*PWD*)와 키로밍 요청서를 전송한다.

③ 로밍서버 : 로밍서버는 h(*PWD*)와 키로밍 요청서를 확인하고, ARN를 생성한다.

④ 로밍서버 → 저장장소 : 로밍서버는 저장장소에 사용자의 ID를 이용하여 해당 정보를 검색한다.

⑤ 저장장소 → 로밍서버 : 로밍서버는 저장장소로부터 $\Gamma_{PPUB}(RAK, REK)$, $E_{REK}(.p12)$, CertP, 로밍PEK서버의 URL을 검색한다.

⑥ 로밍서버 → 사용자 : 로밍서버는 사용자에게 $\Gamma_{PPUB}(RAK, REK)$, $E_{REK}(.p12)$, CertP, 로밍PEK 서버의 URL 및 ARN을 전송한다.

⑦ 사용자 : 사용자는 OSK를 생성하고, $\Gamma_{PPUB}(OSK)$를 계산한다.

⑧ 사용자 → 로밍PEK서버 : 사용자는 로밍PEK서버에게 $\Gamma_{PPUB}(OSK)$, $\Gamma_{PPUB}(RAK, REK)$, ARN, $E_{REK}(.p12)$를 전송한다.

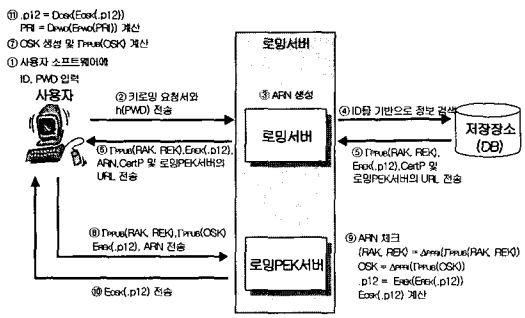
⑨ 로밍PEK서버 : 로밍PEK서버는 전송된 ARN이 이전 ARN보다 크면, 다음과 같이 계산하여 RAK, REK, OSK, .p12를 획득하고 $E_{OSK}(.p12)$를 계산한다.

㉠ $(RAK, REK) = \Delta_{PPRI}(\Gamma_{PPUB}(RAK, REK))$

㉡ $OSK = \Delta_{PPRI}(\Gamma_{PPUB}(OSK))$

㉢ $.p12 = \Delta_{REK}(E_{REK}(.p12))$

UniCERT Roaming 제품의 키로밍 과정 흐름도



⑩ 로밍PEK서버 → 사용자 : 로밍PEK서버는 사용자에게 $E_{OSK}(.p12)$를 전송한다.

⑪ 사용자 : 사용자는 다음과 같이 계산하여 *.p12*를 획득한다.

$$.p12 = D_{OSK}(E_{OSK}(.p12))$$

2.3 제품의 특징

Baltimore사의 UniCERT Roaming 제품의 경우에 .p12를 암호화한 REK가 로밍PEK서버의 공개키로만 암호화되어 있으므로 키로밍 권한 분산 기능을 제공하지 못하고, 로밍서버가 사용자의 패스워드 확인자를 이용해서 사용자처럼 행동할 수 있는 문제점이 발생한다. 그리고 등록 과정 및 키로밍 과정시, 보안 채널에 관해서는 문서에서 언급하지 않고 있지만, 전송되는 패스워드 확인자에 대해 패스워드 추측 공격이 가능하다.

3. Entrust사의 Entrust Roaming 제품

[시스템 파라메터]

- Profile : $E_{PWD}(PRI)$, PUB를 포함하는 파일
- K : Profile을 암호화 하기 위한 대칭키
- RSK : 로밍서버의 비밀키
- x : 사용자가 생성한 랜덤 수
- y : 로밍서버가 생성한 랜덤 수
- S1 : SPEKE 기법[12]을 이용해 생성한 세션키

3.1 등록 과정[10,11]

사용자와 로밍서버의 통신시 보안 채널을 형성하는 자사가 개발한 Entrust/Session[11]을 사용한다.

① 사용자 → 인증서버 : 사용자는 인증서버에게 사용자 등록 신청 및 Profile을 발급 요청을 한다.

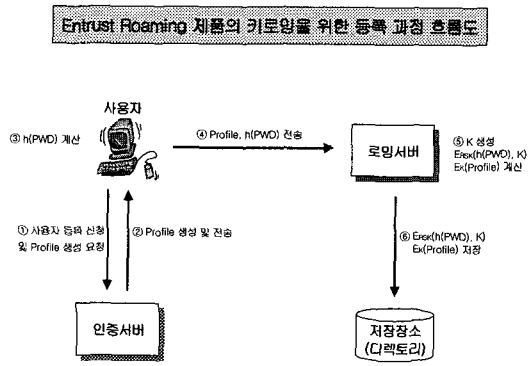
② 인증서버 → 사용자 : 인증서버는 사용자의 Profile을 발급하고, 사용자에게 전송한다.

③ 사용자 : 사용자는 h(*PWD*)를 계산한다.

④ 사용자 → 로밍서버 : 사용자는 로밍서버에게 Profile, $\mathbf{h}(PWD)$를 전송한다.

⑤ 로밍서버 : 로밍서버는 비밀키 K를 생성하고, $\mathbf{E}_K(Profile)$, $\mathbf{E}_{RSK}(\mathrm{h}(PWD), K)$을 계산한다.

⑥ 로밍서버 → 저장장소 : 로밍서버는 저장장소에 $\mathbf{E}_K(\mathrm{Profile})$, $\mathbf{E}_{RSK}(\mathrm{h}(PWD), K)$을 저장한다.



3.2 키로밍 과정[10,11]

사용자와 로밍서버의 통신시 보안 채널을 형성하기 위해서 SPEKE방식[12]을 사용한다.

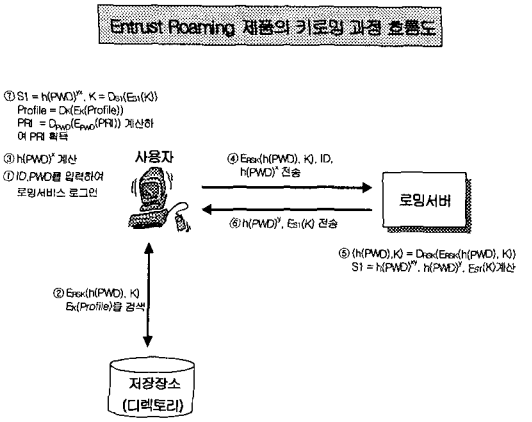
① 사용자 : ID와 PWD를 입력하여 로밍서비스 요청을 위한 로그인한다.

② 사용자 ↔ 저장장소 : 사용자는 저장장소에 저장되어 있는 자신의 키정보 ($\mathbf{E}_{RSK}(h(PWD), K)$, $\mathbf{E}_K(Profile)$)를 검색한다.

③ 사용자 : 사용자는 $h(PWD)^x$를 계산한다.

④ 사용자 → 로밍서버 : 사용자는 로밍서버에게 $\mathbf{E}_{RSK}(h(PWD), K)$, ID, $\mathbf{h}(PWD)^x$를 전송한다

⑤ 로밍서버 : 로밍서버는 $(h(PWD), K) = D_{RSK}(\mathbf{E}_{RSK}(h(PWD), K), h(PWD)^y$, $S1 = h(PWD)^{xy}$, $\mathbf{E}_{S1}(K)$를 계산한다.



⑥ 로밍서버 → 사용자 : 로밍서버는 사용자에게 $\mathbf{h}(PWD)^y$, $\mathbf{E}_{S1}(K)$를 전송한다.

⑦ 사용자 : 사용자는 $S1 = h(PWD)^{xy}$, $\mathbf{K} = D_{S1}(\mathbf{E}_{S1}(K))$, $Profile = D_K(\mathbf{E}_K(Profile))$, $PRI = D_{PWD}(\mathbf{E}_{PWD}(PRI)$을 계산하여 PRI를 획득한다.

3.3 제품의 특징

Entrust Roaming 제품의 경우에 키정보를 안전하게 다운받기 위해 SPEKE(Simple Password Exponential Key Exchange) 방식[12]을 사용한다. 로밍서버만이 키로밍 권한을 가지고 있고, 로밍서버가 패스워드 확인자를 이용하여 사용자처럼 행동할 수 있는 문제점이 있다. 그리고 [10]과 [11]에 따르면 Entrust Roaming 제품에 탑재된 SPEKE는 [12]와 [36]에서 언급된 공격에 의해 패스워드 추측 공격이 가능하다. 그러나 실제 제품에 탑재된 SPEKE가 이러한 문제점을 해결하여 구현했는지에 대해서는 알 수 없었다.

4. VeriSign사의 VeriSign Roaming Service 제품

[시스템 파라메터]

- W : 로밍요청정보를 생성하기 위하여 패스워드(PWD)를 변환시킨 값(=$\mathbf{f}(PWD)$)
- f : 패스워드를 Z_P^*상의 곱 연산에 대한 위수가 q인 원소로 대응시키는 함수
- M : 로밍요청정보(=W^a mod P, 단 P는 큰 소수로 P = 2q + 1이며, 1 < a < q-1)
- $C_{(i)}$: 로밍응답정보
- b(i) : 로밍서버($RS_{(i)}$)의 로밍비밀정보(1 < b(i) < q-1)
- $K_{(i)}$: 로밍용 암호키 조각
- K : 패스워드와 로밍서버의 로밍비밀정보로 생성한 로밍용 암호키
- KDF : 로밍용 암호키 조각을 조합하여 로밍용 암호키를 생성하는 함수
- EPD(Encrypted Private Data) : (PRI, CertA)를 K로 암호화한 키정보
- v(i) : 로밍검증정보
- a : 사용자에 의해서 생성된 랜덤 수(1 < a < q-1)

VeriSign Roaming Service 제품은 VeriSign

OnSite 제품의 선택 추가사항으로 제공되며 동작과정은 다음과 같다.

4.1 등록 과정[13,15,16,20]

사용자와 로밍서버의 통신시 보안 채널로 SSL을 사용한다.

① 사용자 : 사용자는 인증서 발급 신청 및 키로밍 등록 신청을 위한 웹페이지에 접속하여 사용자 S/W를 다운받아 설치한다.

② 사용자 : 사용자의 PRI, PUB을 생성한다.[21]

③ 사용자 → 인증서버 : 사용자는 사용자 등록정보,인증서발급요청서, 키로밍등록신청서를 전송한다.[14, 17]

※ 설정된 로밍정책에 따라 키로밍 여부를 사용자가 직접 선택할 수 있게 할 수 있음.[13,14,17]

④ 인증서버 → 사용자 : 인증서버는 사용자의 인증서를 발급하여 전송한다.

⑤ 사용자 : 자신의 ID, PWD를 입력한다.

⑥ 사용자 ↔ 로밍서버 : 사용자는 PWD로부터 a 및 M을 다음과 같이 생성하여, 사용자의 ID와 함께 M을 각 $RS_{(i)}$에게 전송한 후, 이에 대한 $C_{(i)}$를 전송 받는다.

㉠ 사용자는 a(1 < a < q-1)를 랜덤하게 선택한 후, 다음과 같이 M을 계산한다.

$$M = W^a \bmod P$$

㉡ 2개의 $RS_{(i)}$는 각각

$$C_{(i)} = M^{b(i)} \bmod P \ (i=1,2)$$

를 계산하여 사용자 S/W에게 전송한 후, 사용자의 ID와 b(i)를 자신의 DB에 저장한다.

㉢ 사용자는 수신한 $C_{(i)}$로부터

$$K_{(i)} = C_{(i)}^{1/a} \bmod P \ (i=1,2)$$

을 계산한 후, 이들의 조합으로 K를 생성한다.

$$K = KDF(K_1, K_2)$$

※ 로밍용 암호키 조각을 조합하는 경우 장애허용기능을 제공할 수 있도록 임계치 비밀분산 기법(threshold secret sharing)을 이용할 수도 있음[15]

⑦ 사용자 → 로밍서버 : 사용자는 v(i)를 계산하여 각 $RS_{(i)}$에 전송한다.[15]

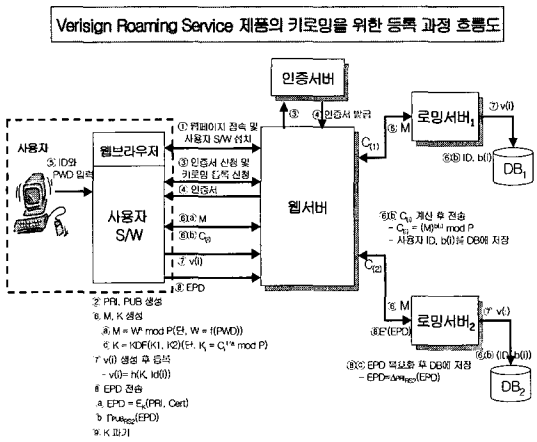
$$v(i) = h(K, Id(i)), \ (i=1,2)$$

(Id(i): 로밍서버 i의 ID)

⑧ 사용자 → 로밍서버 #2 : EPD를 $RS_{(2)}$에게 전송한다. 즉, $PUB_{RS(2)}$로 암호화하여 전송한다.[21] 그리고, K를 파기한다.

$$\underline{EPD} = EK(PRI, CertA)$$

⑨ 로밍서버 #2 → 저장장소 #2 : $RS_{(2)}$는 암호화된 키정보를 자신의 개인키로 복호화한 후, 자신의 DB #2에 EPD를 저장한다.[21]



4.2 키로밍 과정[13,15,16,20]

① 사용자 : 자신의 ID, PWD를 입력한다.

② 사용자 ↔ 로밍서버 : 사용자 S/W는 PWD로부터 a와 M을 다음과 같이 생성한다. 그리고 사용자의 ID와 함께 M을 각 $RS_{(i)}$에게 안전하게(SSL) 전송한 후, 이에 대한 $C_{(i)}$를 안전하게(SSL) 전송 받는다.

㉠ 사용자는 a(1 < a < q-1)를 랜덤하게 선택한 후, M을 계산한다.

$$M = W^a \bmod P$$

㉡ 2개의 $RS_{(i)}$는 사용자의 ID에 해당하는 b(i)를 검색한 후,

$$C_{(i)} = M^{b(i)} \bmod P \ (i=1,2)$$

를 계산하여 사용자에게 안전하게(SSL) 전송한다.

㉢ 사용자 S/W는 수신한 $C_{(i)}$로부터

$$K_{(i)} = C_{(i)}^{1/a} \bmod P \ (i=1,2)$$

계산한 후, 이들 $K_{(i)}$의 조합으로 K를 재생성한다.

$$K = \mathbf{KDF}(K_1, K_2)$$

③ 사용자 → 로밍서버 : 사용자는 검증정보(v'(i))를 계산하여 각 $RS_{(i)}$에게 안전하게(SSL) 전송한다.

$$v'(i) = h(K, Id(i))(i=1,2)$$

(Id(i): 로밍서버 i의 ID)

④ 저장장소 → 로밍서버 : $RS_{(i)}$는 자신의 DB에 저장된 검증정보 v(i)와 지금 전송받은 v′(i)와 일치하는 지를 검증함으로써 사용자를 인증한다.[15]

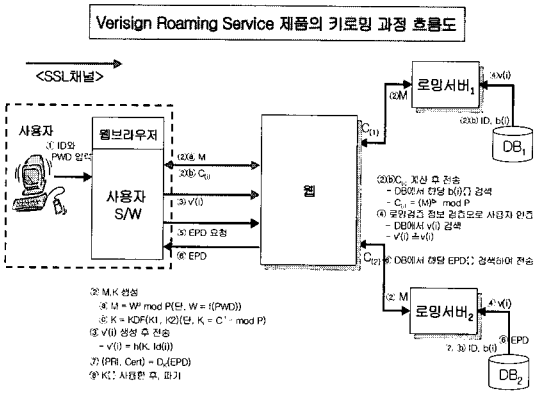
⑤ 사용자 → 로밍서버 #2 : 사용자 인증과정이 성공적으로 완료되면, 사용자는 EPD를 요청한다.

⑥ 로밍서버 #2 → 사용자 : $RS_{(2)}$는 EPD를 검색하여 사용자에게 전송한다.

⑦ 사용자 : 재생성한 K로 EPD를 복호화하여 자신의 로밍프로파일을 얻음으로써 개인키를 로밍받는다.

$$(PRI,\ CertA) = D_K(EPD)$$

⑧ 사용자 : 로밍프로파일을 사용한 후에, 로밍프로파일과 K를 파기한다.



4.3 제품의 특징

VeriSign사의 Roming Service 제품은 다음 3가지의 모델을 제공하고 있어, 응용환경에 따른 다양한 서비스 형태를 제공한다[13].

- Enterprise Roaming : 모든 로밍서버를 기업 내에서 운영·관리하는 모델 구조
- Split Hosting : 한 개의 로밍서버는 Veri Sign등과 같은 로밍서비스 제공기관에서 운영하고, 또 다른 로밍/저장 서버는 기업에서 운영·관리하는 모델 구조
- Outsourced Roaming : 모든 로밍서버/저장 서버를 VeriSign에서 운영하는 모델구조

이 모델구조 중에서 Split Hosting 구조만이 권한분산, 저장장소의 장애허용성 기능 및 서버의 집중 공격에 대한 대처 기능을 제공할 수 있다. 또한 사용자 S/W를 사용자 시스템에서 동작할 수 있는 모드(PTA 모드 : Personal Service Trust mode)와 VeriSign 서버에서 동작할 수 있는 모드(PTS 모드 : Personal Trust Service mode)의 2 가지 모드를 제공하고 있어 조직의 로밍정책에 따라 다양한 로밍서비스 제공이 가능하다.[14, 18, 19, 20] 즉, PTS모드에서는 Verisign 서버가 사용자를 대신하여 키정보 등록, 키로밍, 전자서명 등의 기능을 수행한다.

VeriSign사의 Roaming Service 제품은 서버 공격에 취약한 기존 로밍서비스 모델의 문제점을 해결하기 위하여 사용자의 패스워드와 각각의 로밍서버에서 생성한 2개의 값들이 모아져야만 로밍용 암호키를 생성할 수 있도록 하였다. 이로써, 공격자가 사용자의 로밍용 암호키를 얻기 위해서는 여러 대의 서버를 동시에 공격해야하기 때문에 중앙집중공격에 대처할 수 있다. 또한 물리적으로 분리된 별도에 장소에 위치한 각각의 로밍서버는 별도의 관리자가 독립적으로 운영하며, 각 로밍서버가 관리하는 데이터베이스에 로밍용 암호키 관련정보를 저장함으로써 권한분산 기능을 제공한다[20]. 그러나 SSL을 사용하지 않는 경우에 [29]에서 언급한 것과 같이 패스워드 추측 공격에 취약하다.

5. RSA사의 Keon Web PassPort 제품

사용자와 로밍서버의 통신시 서버로부터 인증 쿠키(Authentication Cookie)를 이용하여 보안 채널을 형성한다.

5.1 동작 과정[23,24]

RSA사의 로밍서버(RSA Keon Web PassPort Sever)는 다음의 서브시스템으로 구성되어 있다.

- 사용자 인증서버 중개자(AB : Authentication Broker) : 로밍정책에 따른 사용자 인증기법(패스워드 기반/RSA Secure ID-이중(2factor) 인증)을 이용하여 사용자를 인증해주는 서버

  ※ RSA Secure ID-이중 인증 : 사용자가 알고 있는 PWD와 RSA SecureID 인증기(Authentication)를 통해 생성된 코드를 이용하는 인증 방식[35]

- 검색서버(CS : Credential Server) : LDAP에 저장되어 있는 virtual card를 검색하여 사용자에게 전송하거나 사용자의 virtual card를 LDAP에 저장하는 서버
- S/W 다운로드 서버(DS : Web PassPort Plug-in)을 안전하게 사용자에게 다운로드해 주는 서버
- URL 필터(URL filter) : 정해진 인증정책에 따라 조직의 웹 응용서버에 대한 사용자의 접근을 통제하는 서버

① 사용자 → 로밍서버 : 사전에 등록된 사용자는 로밍서버에 접속하여 원하는 서비스(예, 회사 문서 접근 요청, 은행계좌 접근 요청 등)를 요청한다.

② 로밍서버 → 사용자 : 조직의 로밍정책에 따라 사용자 인증이 요구되는 지를 검사한다. 사용자 인증이 요구되어지면 로밍서버에 로그인할 수 있는 웹화면(Web PassPort login options page)을 사용자에게 전송한다.

③ 사용자 → 로밍서버 : 미리 정해진 로밍정책의 인증방법에 따라 사용자는 해당하는 인증정보를 전송한다.

④ 로밍서버 → 사용자 : 로밍서버는 전송한 사용자 인증정보를 검증한다. 사용자 인증이 성공적으로 완료되면, 성공메시지와 함께 인증쿠키를 전송해준다.

⑤ 사용자 → 로밍서버 : 사용자는 전송 받은 인증쿠키로 원하는 서비스를 제공하는 응용서버 URL에 접근한다.

⑥ 로밍서버(URL filter → CS) : 서비스를 요청한 사용자가 virtual card(개인키와 인증서를 암호화하여 저장하고 있음)를 소유하고 있는지를 검사한다. virtual card가 없는 경우 a)과정을 거치고, virtual card가 있는 경우 b)과정을 거치게 된다.

a) virtual card가 없는 경우

㉠ 사용자 ↔ 로밍서버 : 사용자는 사용자 S/W를 다운받을 수 있는 URL을 요청한다. 로밍서버(URL filter)는 사용자의 인증쿠키를 이용하여 사용자가 사용자 S/W를 다운로드받을 수 있는 권한이 있는지 확인한 후, 로밍서버(Web PassPort Download Service)가 사용자 S/W를 다운로드한다. 사용자는 다운로드받은 S/W를 설치한다.

㉡ 사용자 → 로밍서버 : 인증서를 발급받기 위한 사용자 정보를 등록한다.

㉢ 로밍서버 ↔ 인증서버 : 사용자 정보와 함께 인증서 발급 요청서를 인증서버에게 전송한 후, 인증서버는 사용자의 인증서와 이에 해당하는 개인키를 로밍서버에게 전송한다.

㉣ 로밍서버 → 저장장소 : 로밍서버는 사용자의 개인키와 인증서를 전송받은 즉시, 사용자의 virtual card에 이들을 다음과 같이 설치한 후,
- $\mathbf{E}_K(PRI)$ (**E** : 3DES-CBC모드 K:112비트), CertA
- $\mathbf{E}_{PUK}(K)$ (PUK : PIN Unlock key로 랜덤하게 생성한 128비트 RC4 대칭키) virtual card를 사용자의 ID 및 PWD와 함께 안전하게 저장장소에 저장한다.

㉤ 로밍서버 → 사용자 : 로밍서버는 virtual card를 사용자에게 전송한다.

㉥ 사용자 → 응용서버 : 사용자는 virtual card를 이용하여 응용서버에게 접근한 후, 원하는 서비스를 제공받는다.

b) virtual card가 있는 경우

㉠ 로밍서버 → 저장장소 : 로밍서버는 다음과 같이 사용자의 개인키와 인증서가 저장되어 있는 virtual card를 저장장소에서 검색한다. 이 경우 검색인덱스로 사용자의 ID와 PWD를 이용한다.
- $\mathbf{E}_K(PRI)$ (**E** : 3DES-CBC모드 K:112비트), CertA
- $\mathbf{E}_{PUK}(K)$ (PUK : PIN Unlock key로 랜덤하게 생성한 128비트 RC4 대칭키)

㉡ 로밍서버 → 사용자 : 로밍서버는 virtual card를 사용자에게 전송한다.

㉢ 사용자 → 응용서버 : 사용자는 virtual card를 이용하여 응용서버에게 접근한 후, 원하는 서비스를 제공받는다.

5.2 제품의 특징

키정보(개인키와 인증서)를 안전하게 저장하기 위한 수단으로 virtual card를 이용하며, vitual

card는 로밍서버에서 저장·관리한다. virtual card는 S/W 스마트카드의 형태로 virtual card와의 인터페이스를 위해 PKCS#11(Cryptographic token interface standard) 및 MS사의 CAPI를 준용한다. 또한 virtual card를 생성하기 위한 인증서 및 개인키는 자동등록 및 인증서 자동발급(OneStep Module)과정을 통해 한번에 이루어진다. 그리고, 패스워드를 이용한 사용자 인증, RSA SecureID-이중 인증 기법 등 다양한 인증방법을 제공한다.

응용환경에 따라 다양하게 구현할 수 있도록 등록과정에서 수동모드와 자동모드 2 가지 모드를 제공한다. 즉, 수동모드일 경우 인증서 및 개인키는 사용자에게 다운로드되고, 사용자 웹 브라우저에서 인증서 및 개인키를 virtual card에 설치한다. 자동모드인 경우 인증서버는 로밍서버에게 인증서와 개인키를 내려주고, 로밍서버는 virtual card에 인증서와 개인키를 설치한다.

6. Cryptomathic사의 EasySign 제품

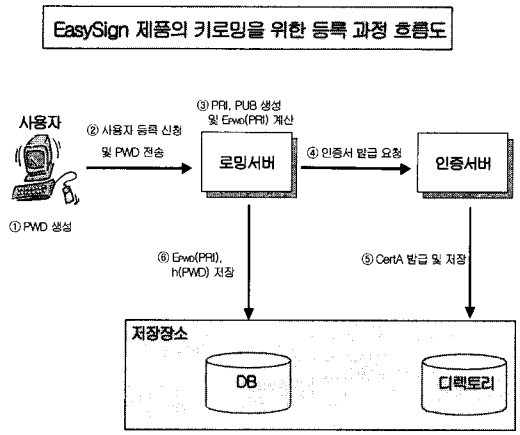
[시스템 파라메터]

- M : 서명할 메세지
- OTPWD : 일회용 패스워드(One-time Password)
- Sign : 전자서명 알고리즘

6.1 등록 과정[25,26]

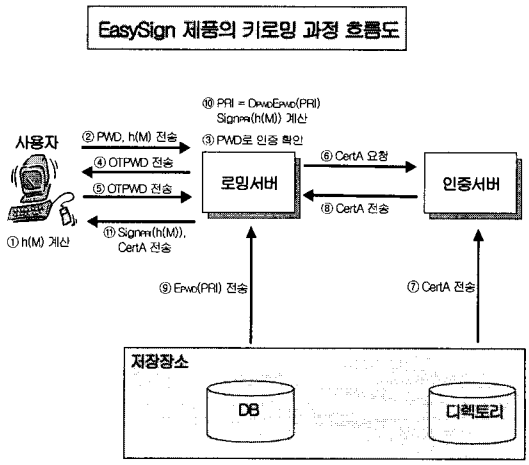
사용자와 로밍서버의 통신시 보안 채널로 SSL을 사용한다.

① 사용자 : 사용자는 PWD를 생성한다.
② 사용자 → 로밍서버 : 사용자는 사용자 등록 신청 및 PWD를 전송한다.
③ 로밍서버 : 로밍서버는 사용자의 PRI, PUB를 생성한다.
④ 로밍서버 → 인증서버 : 로밍서버는 사용자의 인증서 발급 요청을 한다.
⑤ 인증서버 → 저장장소 : 인증서버는 CertA를 발급하고 저장장소(디렉토리)에 CertA를 저장한다.
⑥ 로밍서버 : $E_{PWD}(PRI)$, h(*PWD*)를 계산한다.
⑦ 로밍서버 → 저장장소 : 로밍서버는 저장장소(DB)에 $E_{PWD}(PRI)$와 h(*PWD*)를 저장한다.



6.2 키로밍 과정[25,26]

사용자와 로밍서버의 통신시 보안 채널로 SSL을 사용한다.



① 사용자 : 사용자는 h(M)을 계산한다.
② 사용자 → 로밍서버 : 사용자는 PWD, h(*M*)을 전송한다.
③ 로밍서버 : 전송된 PWD를 이용한 h(*PWD*)와 저장된 h(*PWD*)를 비교
④ 로밍서버 → 사용자 : ③단계를 통과하면, 로밍서버는 사용자에게 OTPWD를 전송한다.
⑤ 사용자 → 로밍서버 : 사용자는 로밍서버에게 OTPWD를 전송한다.
⑥ 로밍서버 → 인증서버 : 로밍서버는 인증서버

에게 CertA를 요청한다.

⑦ 저장장소 → 인증서버 : 인증서버는 저장장소(디렉토리)에서 CertA를 가져온다.

⑧ 인증서버 → 로밍서버 : 인증서버는 로밍서버에게 CertA를 전송한다.

⑨ 저장장소 → 로밍서버 : 로밍서버는 저장장소(DB)에서 $\mathbf{E}_{PWD}(PRI)$를 가져온다.

⑩ 로밍서버 : 로밍서버는 $PRI = \mathbf{D}_{PWD}(\mathbf{E}_{PWD}(PRI))$, $\mathbf{Sign}_{PRI}(h(M))$을 계산한다.

⑪ 로밍서버 → 사용자 : 로밍서버는 사용자에게 $\mathbf{Sign}_{PRI}(h(M))$와 CertA를 전송한다.

6.3 제품의 특징

EasySign 제품의 경우에 키로밍 요청시 보안 채널을 통해 사용자 패스워드가 로밍서버에게 전송되므로 로밍서버가 사용자처럼 행동할 수 있다. 또한 사용자는 로밍서버를 신뢰해서 암호서비스 수행을 전적으로 위임하는 특징을 갖는다.

7. KISA의 Key Roaming Service

〔시스템 파라메터〕

- KS_i : i 번째 키서버 ($1 \le i \le n$)
- AS_i : i 번째 응용서버 ($1 \le i \le m$)
- p : p는 소수이며, $p=2q+1$
- g : $g^q \equiv 1 \bmod p$
- x_i : 키서버i의 PRI ($1 \le i \le n$)
- f : PWD를 Z_p^*상의 곱연산에 대한 위수가 q인 원소로 대응시키는 함수
- **KDF** : 인증키 계산 함수
- **OWF** : 일방향 함수(One-Way Function)
- K : 인증키
- K_I : 응용서버에 대한 사용자 인증키
- y : 키서버들의 그룹 공개키로 다음과 같이 생성된다.

① 각 키서버i(단, $1 \le i \le n$)는 난수 $r_i \in_R Z_q$를 선택한 후, $y_i = g^{ri} \bmod P$를 계산하여 공개한다.

② 각 키서버$_i$는 $f_i(0) = r_i$인 t-1차의 다항식 f_i를 Z_q상에서 랜덤하게 선택한다. 즉,

$$f_i(x) = r_i + a_{i,1} \cdot x + a_{i,2} \cdot x^2 + \cdots + a_{i,t-1} \cdot x_{t-1} \bmod q$$

(단, $a_{i,1}, a_{i,2}, \cdots, a_{i,t-1} \in_R Z_q$)

키서버$_i$는 $f_i(j) \bmod q$ (단, $\forall j \neq i$, $1 \le j \le n$)를 계산하여 키서버$_j$에게 비밀리에 전송한다. $g^{ai,1} \bmod P$, $g^{ai,2} \bmod P$, $\cdots$, $g^{ai,t-1} \bmod P$를 계산하여 공개한다.

③ 각 키서버$_i$는 전송 받은 $f_j(i)$(단, $\forall j \neq i$, $1 \le j \le n$)를 이용하여 다음 식으로 그 정당성을 검증한다

$$g^{f_j(i)} \stackrel{?}{=} y_j \cdot (g^{a_{j,1}})^{i^1} \cdot \cdots \cdot (g^{a_{j,t-1}})^{i^{t-1}} \bmod P$$

(단, $\forall j \neq i$, $1 \le j \le n$)

④ $H \equiv \{$키서버i | 키서버$_i$는 과정 ③을 통과한 모든 정직한 키서버$\}$이라 하자. 이제 각 키서버$_i$는 자신의 개인키 $x_i = \sum_{j \in H} f_j(i)$를 계산하여 비밀리에 보관한다.

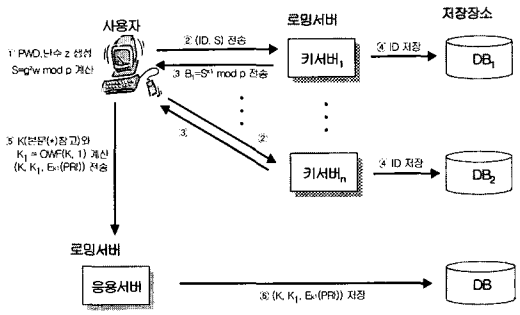
⑤ 키서버들은 자신들의 그룹 공개키 y를 다음과 같이 계산하여 공개한다.

$$y = \prod_{j \in H} y_j$$

7.1 등록 과정[29]

사용자와 로밍서버의 통신시 보안 채널이 필요하다.

KISA의 Key Roaming Service의 키로밍을 위한 등록 과정 흐름도



① 사용자 : 사용자는 PWD와 난수 z를 생성하고, w와 S를 다음과 같이 계산한다.

㉠ $w = f(PWD)$

㉡ $S = g^z w \bmod p$

② 사용자 → 키서버i : 사용자는 (ID, S)를 키서버에게 전송한다. (i = 1, 2, $\cdots$, n)

③ 키서버$_i$ → 사용자 : 키서버$_i$는 $B_i = S^{xi} \bmod p$를 사용자에게 전송한다.

④ 키서버$_i$ → 저장장소 : 키서버$_i$는 저장장소(DB_i)에 ID를 저장한다.

⑤ 사용자 → 응용서버 : 사용자는 인증키 K, K_1,를 다음과 같이 계산하여 (ID, K_1, $\mathbf{E}_{K1}(PRI)$)를 응용서버에게 전송한다.

㉠ $K = KDF((\prod_{1 \le i \le t} B_i^{\prod_{1 \le j \le t, j \ne i} j/(j-i)})/y^z \bmod p)$ (*)

㉡ K1 = OWF(K, 1)

⑥ 응용서버 → 저장장소 : 응용서버는 저장장소 (DB)에 (ID, K_1, $\mathbf{E}_{K1}(PRI)$)를 저장한다.

7.2 키로밍 과정[29]

사용자와 로밍서버간의 통신시 SSL과 같은 별도의 보안 채널을 필요로 하지 않는다.

① 사용자 : 사용자는 PWD와 난수 z를 이용해서 w와 S를 다음과 같이 계산한다.

㉠ $w = f(PWD)$

㉡ $S = g^z w \bmod p$

② 사용자 → 키서버$_i$: 사용자는 (ID, S)를 키서버에게 전송한다.(i = 1, 2, …, n)

③ 키서버$_i$ → 저장장소 : 키서버$_i$는 저장장소 (DB_i)를 검색하여 등록된 ID를 확인한다.

④ 키서버$_i$ → 사용자 : 키서버$_i$는 사용자에게 $B_i = S^{xi} \bmod p$를 계산해서 전송한다.

⑤ 사용자 → 응용서버 : 사용자는 인증키 K, K_1,를 다음과 같이 계산하고, 응용서버에게 키로밍을 요청한다.

㉠ $K = \mathbf{KDF}((\prod_{1 \le i \le t} B_i^{\prod_{1 \le j \le t, j \ne i} j/(j-i)})/y^z \bmod p)$ (*)

㉡ $K_1 = \mathbf{OWF}(K, 1)$

⑥ 응용서버 → 사용자 : 응용서버$_j$는 챌린지 값 C_1 ($1 \le C_1 \le q-1$)을 생성하여 사용자에게 전송한다.

⑦ 사용자 → 응용서버 : 사용자는 챌린지 C_1의 응답으로 $\mathbf{OWF}(K_1, C_1)$를 계산하여 응용서버에게 전송한다.

⑧ 응용서버 → 저장장소 : 응용서버는 저장장소 (DB)에서 K_1를 검색하고 $\mathbf{OWF}(K_1, C_1)$를 계산한 후, 사용자가 전송한 $\mathbf{OWF}(K_1, C_1)$를 비교하여 사용자 인증하고, 사용자에게 $E_{K1}(PRI)$를 전송한다.

⑨ 응용서버 → 사용자 : 응용서버는 사용자에게 $\mathbf{E}_{K1}(PRI)$를 전송한다.

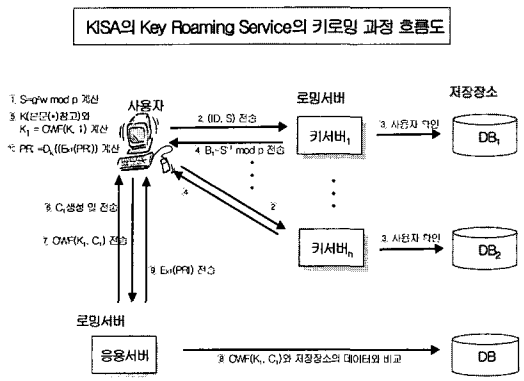
⑩ 사용자 : 사용자는 PRI = $\mathbf{D}_{K1}(\mathbf{E}_{K1}(PRI))$를 계산하여 PRI를 획득한다.

〔표 2〕 키로밍 제품별 특징 분석표

기능 \ 제품	Hush Enterprise	UniCERT Roaming	Entrust Roaming	VeriSign Roaming Service	RSA Keon Web PassPort	EasySign	KISA의 Key Roaming Service
사용자의 키생성 주체	사용자[6]	인증서버[7]	인증서버[11]	사용자[21]	인증서버[23]	로밍서버[27]	사용자[29,31]
로밍될 키용도	암호화용, 전자서명용[4]	전자서명용[8]	암호화용, 전자서명용[9]	전자서명용[13]	암호화용, 전자서명용[23]	전자서명용[26]	암호화용[29,31]
키로밍 권한 분산	×	×	×	○	×	×	○
로밍서버의 저장된 패스워드 또는 패스워드 확인자를 이용한 사용자 위장 공격에 안전	×	×	×	○	-1)	×	○
별도의 보안 프로토콜 없이도 패스워드 추측 공격에 안전	×	×	△2)	× [29]	-1)	×	○

1) 제품 관련 문서에 상세한 내용이 기술되어 있지 않음

2) 실제 제품에 탑재된 SPEKE가 [12]과 [36]에서 언급된 공격 기법을 고려하여 구현되었는지의 여부는 제품 관련 문서에 기술되어 있지 않음



7.3 제품의 특징

Verisign Roaming Service의 인증 방식은 사용자가 인증키 계산시에 (n+1)번의(n은 로밍서버의 개수) 지수승 연산이 필요하나 KISA의 Key Roaming Service의 인증 방식은 2번의 지수승 연산과 (n+1)번의 곱셈 연산만을 필요로 하기 때문에 로밍서버가 증가하더라도 인증키를 효율적으로 계산할 수 있다. KISA의 Key Roaming Service의 경우에 인증키를 생성하기 위해서 복수개의 키서버들을 두며, 복수개의 키서버들의 개인키와 그룹 공개키를 이용하여 인증키를 발생시키므로, 키서버는 사용자의 ID 이외에 어떠한 정보도 저장하지 않는다. 그 결과, SSL과 같은 별도의 보안 채널 없이도 위장 서버의 사용자 패스워드에 대한 사전 탐색 공격을 차단할 수 있다. 특히, 키로밍 요청시 (t, n)-비밀분산기법[32-34]을 이용해서 인증키를 생성하기 때문에 키서버들의 장애허용성 기능이 제공된다.

Ⅳ. 국외 키로밍 제품들의 특징

3장에서 살펴본 국외 키로밍 제품들은 로밍된 키의 사용이 종료됨과 동시에 로밍된 단말기의 임시 메모리에서 삭제되는 공통된 특징이 있다. 사용자의 키생성 주체, 로밍될 키용도, 키로밍 권한 분산, 로밍서버의 패스워드 확인자를 이용한 사용자 위장 공격에 안전성 및 별도의 보안 프로토콜 없이도 패스워드 추측 공격에 안전성 여부에 관한 특징은 표 2에서 살펴보기로 한다.

Ⅴ. 결 론

사용자가 별도의 저장매체 없이도 인터넷이 연결되는 모든 컴퓨터에서 키로밍서버로부터 자신의 개인키를 다운받아 암호서비스를 할 수 있는 키로밍 기술은 사용자에게 편리한 이동성을 제공하여 현재 많은 주목을 받는 기술이다. 이에 따라 본 논문에서는 국외 키로밍 제품들에 대한 기능 및 특징을 위주로 비교 분석하였다. 현재 많은 국외 주요 정보보호 업체에서 키로밍 기술에 대한 제품 및 특허를 개발한 상태이며, 국내에서도 키로밍 기술 연구 및 제품 개발이 주요 정보보호업체를 중심으로 활발히 진행되고 있다.

참고문헌

[1] http://www.hush.com/services/key_server_network/user_technology.shtml/

[2] Hushmail Private Label (document)

[3] http://www.hush.com/our_technology/how _it _works/encryption.shtml/

[4] Hush Encryption Engine White Paper

[5] http://www.hush.com/services/key_server_network/

[6] Austin, Cliff A. Baltzley, Tex, Public Key Cryptosystem With Roaming User Capability, *US. Patent*, [US Patent Number 6,154,543] 2000. 11

[7] http://www.Baltimore.com/unicert/technology/roaming.asp

[8] Company confidential roaming info document

[9] http://www.entrust.com/authority/roaming/technology

[10] The Entrust Roaming Solution (document)

[11] http://csrc.nist.gov/pki/twg/y2000/twg00-9.html

[12] D. Jablon, Strong password-only auth- enticated key exchange, *ACM Computer Communications Review*, vol 26, no 5, 1996

[13] VeriSign Inc., Roaming Service Administra- tor's Guide, 2001. 7

[14] VeriSign Inc., Go Secure! for Web Appli- cations Administrator's Guide, 2001.10

[15] Burton S. Kaliski JR, Wellesley,

Server-Assisted Regeneration of a Strong Secret From a Weak Secret, *US. Patent*, [US Patent Number 09/804,460] 2001.12
[16] Warwick Ford, Burton S. Kaliski JR, Server-Assisted Generation of a Strong Secret form a Password, *Proceedings of the 5th International Workshop on Enterprise Security*, IEEE, 2000
[17] VeriSign Inc., OnSite Introduction /Rev3, 2000
[18] VeriSign Inc., Go Secure! Web Applications 2.6 Release Notes, 2001.10
[19] VeriSign Inc., Sopport and Service-Overview, 2002.1
[20] http://www.verisign.com/products/roaming
[21] VeriSign Korea Inc.(CrossCert Inc.), Private communication, 2002
[22] http://www.rsasecurity.com/
[23] RSA Keon Web PassPort Technical Overview
[24] RSA Keon Web PassPort (document)
[26] http://www.cryptomathic.com/news/signature-internetbanking.html
[26] EasySign - New Approach to Digital Signatuares
[27] EASYSIGN Technical White Paper Version 2.0
[28] Cryptomatic Case Study - Home Banking Leaves Home
[29] 김지연, 김승주, 권현조, 박해룡, 김홍근, 서버의 사전탐색을 고려한 패스워드 기반의 사용자 인증 프로토콜, *국내 특허 출원* [국내 특허 출원번호 10-2002-0010313] 2002. 2
[30] OpenPGP Message Formet, RFC 2440
[31] 한국정보보호진흥원, "암호키관리기반구조 모델 및 구축기술 개발", 최종 연구개발 결과보고서
[32] Y. Desmedt, Y.Frankel, Threshold cryptosystems, *Advanced in Cryptology - Crypto'89, Springer-Verlag, LNCS 435*, 1990.
[33] T. P. Pedersen, A threshold cryptosystem without a trusted party, *Advanced in Cryptology - Eurocrypt'91, Springer-Verlag*, LNCS 547, 1991
[34] T. P. Pedersen, Distributed provers with applications to undeniable sig-natures, Advanced in Cryptology-Euro'91, Springer-Verlag, LNCS 547, 1991
[35] http://www.rassecurity.com/products/securid/ index.html
[36] P. C. VanOorschot, M. J. Wiener, On Diffie- Hellman Key Agreement with Short Exponent, *in Advances in Cryptology-Eurocrypt'96, Spriger-Verlag*, LNCS 1070, 1996

〈著者紹介〉

박 해 룡(Haeryong Park)
1999년 2월 : 전남대학교 수학과 이학사
2001년 2월 : 서울대학교 대학원 수학과 이학석사
2000년 12월~현재 : 한국정보보호 진흥원(KISA) 연구원
관심분야 : 키관리, 암호프로토콜, 전자화폐 등

권 현 조(Kwon Hyun Jo)
1997년 2월 : 성균관대학교 정보공학과 공학사
2000년 8월 : 성균관대학교 정보통신대학원 공학석사
1997년 1월~1997년 7월 : (주) 나라계전 기술연구소 연구원
1997년 7월~현재 : 한국정보보호진흥원(KISA) 연구원
관심분야 : 키관리, 암호프로토콜, 스마트카드, 정보보호시스템 평가체계, 정보보호 기술 표준화

김 지 연(Jeeyeon Kim)
종신회원
1995년 2월 : 성균관대학교 정보공학과 공학사
1997년 2월 : 성균관대학교 대학원 정보공학과 공학석사
1996년 12월~현재 : 한국정보보호진흥원(KISA) 연구원

관심분야 : 암호프로토콜, 키관리 등

김 승 주(Seungjoo Kim)
종신회원
본호의 "암호라이브러리 및 암호 API 개발현황" 저자 소개 참조.