스마트카드에 적용가능한 분산형 인증 및 키 교환 프로토콜☆

The Distributed Authentication and Key Exchange Protocols for Smartcard

오 흥 룡* 윤 호 선** 염 흥 열***
Heung-Ryongl Oh Ho-Sun Yoon Heung-Youl Youm

요 약

PAK(Password-Authenticated Key Exchange) [1,2] 프로토콜은 사용자가 암기하거나 휴대하기 쉬운 짧은 길이의 패스워드를 이용하여 통신 주체들간에 상호 인증하고, 결과적으로 통신주체간에 추후의 안전한 통신을 위한 충분히 큰 길이의 세션키를 분배하는 프로토콜이다. 본 논문에서는 MTI(Matsumoto, Takashima, Imai) 키 분배 프로토콜 [3] 과 PAK 프로토콜을 이용하고 스마트카드에 적용가능한 분산형 키 분배 프로토콜들을 제안한다. 그리고 단일 서버에 패스워드를 저장할 경우, 공격자가 서버와 타협하여 통신 주체들간에 비밀 정보를 알아낼 수 있으므로, 패스워드 검증 정보를 여러 서버들에 분산시켜 안전성을 향상시켰다. 본 논문의 목적은 사용자가 스마트카드의 비밀키와 패스워드를 이용하여 인증과 세션키 분배를 수행하는 분산형 키 분배방식을 제안하는 것에 있다. 또한 서버 타협 공격을 피하기 위하여 임계치 비밀 분산(threshold secret sharing) 기법을 적용하여 공격자가 서버와 타협이 이루어졌을 경우라도 비밀 정보를 유지할 수 있는 MTI 키 분배 프로토콜을 이용한 분산형 키 분배 프로토콜들을 제안한다. 그리고 제안된 분산형 키 분배 프로토콜의 안전성 분석 및 기존의 방식과 비교 분석한다.

Abstract

A PAK(Password-Authenticated Key Exchange) protocol is used as a protocol to provide both the mutual authentication and allow the communication entities to share the session key for the subsequent secure communication, using the human-memorable portable short-length password. In this paper, we propose distributed key exchange protocols applicable to a smartcard using the MTI(Matsumoto, Takashima, Imai) key distribution protocol and PAK protocol. If only one server keeps the password verification data which is used for password authentication protocol, then it could easily be compromised by an attacker, called the server-compromised attack, which results in impersonating either a user or a server. Therefore, these password verification data should be distributed among the many server using the secret sharing scheme. The object of this paper is to present a password-based key exchange protocol which is to allow user authentication and session key distribution, using the private key in a smartcard and a password typed by a user. Moreover, to avoid the server-compromised attack, we propose the distributed key exchange protocols using the MTI key distribution protocol. And we present the security analysis of the proposed key exchange protocol and compare the proposed protocols with the existing protocols.

☞ Keyword : 패스워드, PAK, MTI, Threshold, Secret Sharing

* 정 회 원 : 한국정보통신기술협회(TTA)
hroh@tta.or.kr(제 1저자)
** 정 회 원 : 한국전자통신연구원(ETRI)
yhs@etri.re.kr(공동저자)
*** 정 회 원 : 순천향대학교 정보보호학과 정교수
hyyoum@sch.ac.kr(공동저자)
☆ 본 연구는 정통부 ITRC 지원 사업에 의하여 수행되었음
[2004년/08/06 투고 - 2004/09/16 심사 - 2005/02/22 심사완료]

1. 서 론

통신 및 네트워크 기술의 발전으로 인터넷, 전자 상거래, 원격 사용자간의 통신, 응용 서버와 클라이언트의 통신 등 많은 서비스가 확대되고 있다. 특히 인터넷 통신을 이용한 원격 사용자간의 통신과 응용 서버와 클라이언트의 통신에서 통신 주체 사이에 안전한 통신 서비스를 제공하기 위하여 여러 가지 방법으로 보안 서비스를 제공하고 있는데, 통신 정보의 암호를 이용한 보안 서비스는 대표적인 방법이다. 이때 정보의 암호 서비스를 위해서는 통신주체 사이에 통신에서 사용될 키가 공유되어야 하고, 통신 주체는 상호 인증을 통해 통신을 하고 있는 주체가 올바른 주체인지를 확인하여야 한다. 이로 인하여 정보의 암호화 서비스에서 키 분배와 상호 인증은 반드시 필요한 기능이다. 또한, 단일 서버와 클라이언트 통신 모델에서 공격자와 서버의 타협으로 인해 서버와 클라이언트의 공유 비밀 정보인 패스워드가 노출된다면 클라이언트의 비밀 정보는 더 이상 보호될 수 없다. 그러므로 단일 서버에 저장된 패스워드 정보를 여러 서버에 분산하여 사용자의 비밀 정보를 보호해야 한다.

본 논문에서는 MTI 키 분배 프로토콜을 이용하여 통신 주체간에 비밀 통신이 가능하도록 키 분배 프로토콜을 제안하였으며, 이때 통신 주체간 인증을 위해서 PAK 프로토콜을 사용하였다. 또한 서버 타협 공격을 피하기 위하여 임계치 비밀 분산 기법을 적용하여 공격자가 서버와 타협이 이루어졌을 경우에도 안전하게 프로토콜을 수행할 수 있도록 하였다. 제안된 분산형 키 분배 프로토콜은 단일 서버형 키 분배 프로토콜에 비해서 서버측의 계산량 및 통신량은 증가하지만 클라이언트측의 계산량을 동일하게 설계함으로써 스마트카드와 같은 연산 능력이 떨어지는 클라이언트에 적합하도록 설계하였다. 또한 임계치 키 분배 프로토콜에 참여하는 참여자의 수가 5보다 작은 경우 기존에 제안된 방식에 비해서 계산량 측면에서 효율적이다.

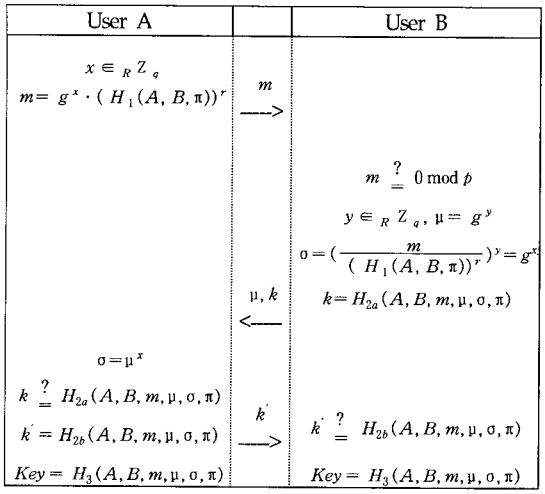
본 논문의 2장에서는 패스워드 기반의 인증 키 분배 프로토콜과 MTI 멱승보간법 프로토콜의 개념을 살펴보고, 3장에서는 MTI 프로토콜을 이용한 분산형 키 분배 프로토콜을 제안한다. 마지막으로 4장에서는 제안한 프로토콜의 안전성 및 효율성을 분석하고 기존의 프로토콜과 제안된 프로토콜을 비교 분석하며, 5장에서 결론을 맺는다.

2. 연구배경

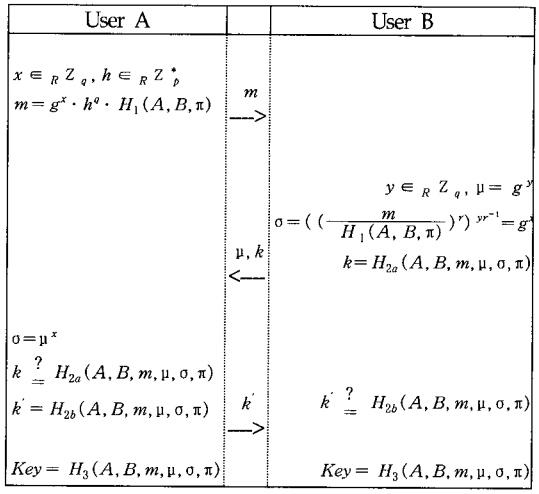
패스워드 기반의 인증 키 분배 프로토콜의 개념은 1992년 S. Bellovin 과 M. Merritt에 의해 제안되었다[11]. 그리고, 그 후 많은 변형 프로토콜과 다양한 환경에 적용 가능한 프로토콜이 제안되었다. MacKenzie는 2002년 처음으로 임계치 PAKE를 제안[23]하였고, Perlman과 Kaufman은 PAK 프로토콜을 이용한 가상 소프트 토큰(virtual soft token)의 아이디어를 제안하였다. 또한 Victor Boyko, Philip MacKenzie와 Sarvar Patel는 PAK 프로토콜과 여러 가지 PAK 변형 프로토콜들을 제안하였으며[2], Diffie-Hellman 키 분배 프로토콜 기반의 PAK 프로토콜, 서버 클라이언트 모델에서 클라이언트의 계산량 감소를 위해 제안된 PAK-R, PAK-EC 프로토콜, 상호 인증 과정을 암시적인 방법으로 최적화한 PPK 프로토콜, 서버 타협에 강한 PAK-X 프로토콜[1,2]등이 그것이다. 그리고 Ford and Kaliski는 처음으로 가상 소프트 토큰에 적용 가능한 다중 서버와 단일 클라이언트 모델을 제안하였다. 본 논문에서 PAK 프로토콜은 통신주체간에 인증을 위해 사용되어진다.

2.1 PAK 프로토콜

그림 1과 그림 2는 본 논문의 기본 빌딩 블록으로 이용된 MacKenzie 가 제안한 PAK, PAK-R 프로토콜을 나타내고 있다[1,2]. 사용자 A와 사용자 B는 사전에 개체 인증을 위한 패스워드를 공유하고 있다고 가정한다. PAK, PAK-R 프로토콜



〈그림 1〉 PAK 프로토콜



〈그림 2〉 PAK-R 프로토콜

의 시스템 파라미터는 다음과 같다.

- A : 사용자 A의 ID, B : 사용자 B의 ID
- π : 사용자 A, B의 패스워드
- $p = rq + 1$, $\gcd(r, q) = 1$
- p : 1024 비트의 소수, q : 160 비트의 소수
- g : Z_p^*의 서브그룹의 생성자, q의 원시근
- h : Z_r의 서브그룹의 생성자, $h^q = (g^r)^q$이며, $g^{qr} = g^{(p-1)} = 1$
- H_1, H_{2a}, H_{2b}, H_3 : 해쉬 기능의 함수
- H_1 : 출력 길이가 1024+160=1184 비트인 해쉬 함수[1]

$\alpha[A,B,\pi] \in_R Z_q$를 생성하여 저장하고, $h \in_R Z_p^*$와 $\beta \in_R Z_{\lfloor 2^\eta/p \rfloor}$를 생성한다. 그리고 $(h^q g^{\alpha[A,B,\pi]} \bmod p) + \beta p$를 반환한다. 그리고 이것은 $h^q g^{\alpha[A,B,\pi]} \bmod p$ 가 Z_p^*[8]로부터의 임의의 원소이고, $\frac{2^\eta \bmod p}{2^\eta}$는 무시할 정도이므로, 길이 η의 랜덤 비트 스트링과 분간할 수 없다. 결과적으로 $(H_1(A,B,\pi))^r$은 $(h^q g^{\alpha[A,B,\pi]})^r = h^{qr} g^{r \cdot \alpha[A,B,\pi]} = g^{r \cdot \alpha[A,B,\pi]}$ 이다.

- H_{2a}, H_{2b}, H_3: 출력 길이가 160비트인 해쉬 함수
- Key : 세션키

통신주체들(서버, 클라이언트)간에 동일한 패스워드를 공유하여 불안전한 네트워크 환경에서, 상호간에 인증하고 충분히 큰 세션키를 나누어 갖기 위한 패스워드 기반의 키 교환 프로토콜에 대한 많은 연구가 수행되었다. 이중 Philip MacKenzie가 제안한 PAK[1] 프로토콜은 오프라인 사전공격(off-line dictionary attack)을 이용하여 공격자가 올바른 패스워드를 결정하지 못하도록 하였으며, PFS(Perfect forword Secrecy), DDH(Decision Diffie-Hellman)에 대한 가정을 통해 안전성이 증명되었다. 여기서 DDH란 위수가 소수 q인 순환 덧셈군으로 (g, ga, gb, gab)와 (g, ga, gb, gc)를 구분하는 문제이며, $c \equiv ab$를 만족하는 경우 Diffie-Hellman tuple이라 한다.

이후 Philip MacKenzie에 의해 PAK 프로토콜에서 클라이언트 측면에 계산량을 감소시켜 구형 PC, 스마트카드, PDA에 적용 가능한 PAK-R[2] 프로토콜이 제안되었다. 기존의 PAK에서는 m을 구하기 위해 q비트의 지수승 1회, r비트의 지수승 1회, σ를 구하기 위한 q비트의 지수승 1회 등 총 3회의 연산을 하지만, PAK-R 프로토콜에서 m을 구하기 위해 q비트의 지수승 2번, σ를 구하기 위한 q비트의 지수승 1회 총 3회의 연산

을 계산한다(여기서, q : 160비트, r : 760비트). 그리고 PAK-R 프로토콜 또한 DDH에 대한 가정을 통해 안전성이 증명되었다.

2.2 MTI 키 분배 프로토콜

MTI 키 분배 프로토콜[3]은 1986년 일본에서 Matsumoto, Takashima, 그리고 Imai 가 제안한 키 분배 프로토콜이다. 이 프로토콜은 Diffie-Hellman 키 분배 프로토콜의 변형으로서 두 개의 교환 메시지를 교환함으로써 수동 공격의 취약점을 보완하여, 장기간 비밀키가 누설되더라도 단기간 세션키를 복구할 수 없는 PFS(Perfect Forward Secrecy) 특성을 제공하며 안전한 통신을 위한 세션키를 분배한다. 이 프로토콜은 Diffie-Hellman 문제와 유한 필드에서 이산대수 문제를 이용하여 설계되었다. MTI 키 분배 프로토콜의 시스템 파라미터는 다음과 같다.

- A,B,p,q,g : PAK 프로토콜 파라미터와 동일함
- X_A,X_B : 사용자 A, B의 고정 비밀키 ; $X_A,X_B \in R[1..q-1]$
- r_A,r_B : 사용자 A, B의 임시 비밀키 ; $r_A,r_B \in R[1..q-1]$
- y_A,y_B : 사용자 A, B의 고정 공개키 ; $y_A = g^{X_A} \bmod p, y_B = g^{X_B} \bmod p$
- T_A,T_B : 사용자 A, B의 임시 공개키 ; $T_A = g^{r_A} \bmod p, T_B = g^{r_B} \bmod p$
- Key : 세션키

User A $y_A = g^{X_A} \bmod p$	공유정보 p, g, y_A, y_B	User B $y_B = g^{X_B} \bmod p$
$r_A \in_R Z_p$ $T_A = g^{r_A} \bmod p$	T_A ——> T_B <——	$r_B \in_R Z_p$ $T_B = g^{r_B} \bmod p$
$Key = T_B^{X_A} y_B^{r_A} \bmod p$ $= g^{r_B X_A + r_A X_B} \bmod p$		$Key = T_A^{X_B} y_A^{r_B} \bmod p$ $= g^{r_A X_B + r_B X_A} \bmod p$

〈그림 3〉 MTI 키 분배 프로토콜

MTI 키 분배 프로토콜은 키의 멱승 부분이 곱셈 형태로 되는 것과 덧셈의 형태로 되는 것이 있다. 본 논문에서는 그림 3과 같이 덧셈 형태의 MTI 키 분배 프로토콜만을 이용하여 제안하였다. 이는 곱의 형태로 설계하였을 때 보다 클라이언트의 계산량이 감소되기 때문이다.

2.3 PAK 기반의 MTI 키 분배 프로토콜

본 절에서는 PAK 프로토콜 기반의 MTI 키 분배 프로토콜을 설명하며, 패스워드를 이용한 인증을 위해서 PAK 프로토콜을 이용한다. 본 프로토콜은 3장에서 제안될 분산형 키 분배 프로토콜을 제안하기 위한 기반 프로토콜로 이용된다. 본 논문에서 제안하는 PAK 프로토콜을 위한 시스템 파라미터는 다음과 같다. PAK 프로토콜은 기본적으로 서버와 클라이언트간의 프로토콜이므로, 약어를 서버에 적합하도록 설정하였다.

- C : 클라이언트의 ID, S : 서버의 ID
- π : 사용자 C와 서버 S의 사전에 공유하는 패스워드
- p,q,g : PAK 시스템 파라미터와 같음
- X_C,X_S : C와 S의 고정 비밀키 ; $X_C,X_S \in R[1..q-1]$
- r_C,r_S : C와 S의 임시 비밀키 ; $r_C,r_S \in R[1..q-1]$
- y_C,y_S : C와 S의 고정 공개키 ; $y_C = g^{X_C} \bmod p, y_S = g^{X_S} \bmod p$
- H_1, H_{2a}, H_{2b} : 해쉬 기능의 함수
- H_1 : 출력 길이가 1024+160=1184 비트인 해쉬 함수[1]. 2.1 절 참조
- H_{2a}, H_{2b} : 출력 길이가 160 비트인 해쉬 함수
- H_3 : 세션키를 계산하기 위한 HMAC-SHA와 같은 해쉬 함수
- Key : 세션키

제안 프로토콜은 토큰으로 스마트카드를 이용하고 키 분배 방식으로 공개키 암호 방식을 이용하

며, 스마트카드를 사용하기 위하여 패스워드를 사용자가 이용하는 경우에 적용될 수 있다. 기존의 스마트카드에 기반을 둔 키 분배 방식은 대부분 공개키를 이용하나, 본 논문에서 제안하는 키 분배방식은 스마트카드에서 사용되는 사용자 인증을 위하여 패스워드를 키 분배에 이용함으로써, 사용자 인증 기능을 수행하고 추후의 암호 통신을 위한 세션키를 동시에 분배할 수 있는 프로토콜이다. 여기서 클라이언트는 스마트카드에 자신의 공개키에 대응되는 비밀키를 저장하고, 클라이언트를 구동하는 사용자는 클라이언트에게 자신의 패스워드를 입력하며, 스마트카드는 사용자가 입력하는 패스워드와 자신이 저장하는 패스워드 검증 정보를 이용하여 사용자를 인증하며, 스마트카드는 사용자가 입력한 패스워드를 추후의 서버와의 안전한 통신을 위한 세션키를 생성하고, 클라이언트가 서버에게 인증받는 개체 인증을 수행한다. 서버와 클라이언트는 사전에 안전한 방법으로 제안 프로토콜을 수행하기 위한 패스워드 검증 정보를 교환함을 가정한다. 제안 프로토콜을 수행함으로써, 클라이언트는 서버에게 자신을 인증함과 동시에 추후에 서버와 클라이언트 간에 암호통신을 위한 세션키를 교환할 수 있다. 그림 4는 단일 서버 환경에서 제안된 PAK 기반의 MTI 키 분배 프로토콜을 나타내고 있다.

Step 1, 클라이언트가 랜덤하게 $r_C \in_R Z_p$를 선택하고 $m = g^{r_C} H_1(C,S,\pi)^r$을 계산한다. 클라이언트는 m을 서버에게 전송한다. 또한 서버는 m이 0인지 확인한다. 0 이 아니면 서버는 $r_S \in_R Z_p$를 랜덤하게 선택한다.

Step 2, 서버는 자신이 저장하고 있는 패스워드 검증 데이터를 이용하여 $\sigma = (\frac{m}{H_1(C,S,\pi)^r})^{X_S}$ $y_C^{r_S} = g^{X_C r_S} g^{X_S r_C} \bmod p$ 을 계산한다.

Step 3, 서버는 $\mu = g^{r_S}$, $k = H_{2a}(y_C y_S m, \mu, \sigma, \pi)$를 계산하고 이를 클라이언트에게 전송한다.

Step 4, 클라이언트는 수신된 μ와 서버의 공개키를 이용하여 σ 를 계산하고, 서버에서 수신한 k와 자신이 계산한 k 값이 동일한지를 검사한다.

Step 5, 만약 동일하면, 클라이언트는 해쉬 함

	Client X_C, π	공유정보 $y_C = g^{X_C} \bmod p$, $y_S = g^{X_S} \bmod p$	Server $X_S, H_1(C, S, \pi)$
1	$r_C \in {}_R Z_p$ $m = g^{r_C} H_1(C, S, \pi)^r$	m —>	$Test \quad m \stackrel{?}{=} 0 \bmod p$ $r_S \in {}_R Z_p$
2			$\sigma = (\frac{m}{H_1(C, S, \pi)^r})^{X_S} y_C^{r_S}$ $= g^{X_C r_S} g^{X_S r_C} \bmod p$
3		μ, k <—	$\mu = g^{r_S}$ 계산 $k = H_{2a}(y_C, y_S, m, \mu, \sigma)$ 계산
4	$\sigma = \mu^{X_C} y_S^{r_C}$ $= g^{X_C r_S + X_S r_C} \bmod p$ $k \stackrel{?}{=} H_{2a}(y_C, y_S, m, \mu, \sigma)$		
5	$k' = H_{2b}(y_C, y_S, m, \mu, \sigma)$ $Key = H_3(y_C, y_S, m, \mu, \sigma)$	k' —>	$k' \stackrel{?}{=} H_{2b}(y_C, y_S, m, \mu, \sigma)$ $Key = H_3(y_C, y_S, m, \mu, \sigma)$

〈그림 4〉 PAK 기반의 MTI 키 분배 프로토콜

수 H_{2b} 를 사용하여 $k' = H_{2b}(y_C y_S m, \mu, \sigma, \pi)$를 계산하여 서버에게 전송한다. 서버는 자신이 계산한 k' 값과 서버에게 받은 k' 이 동일한지를 검사한다.

결과적으로, 클라이언트와 서버는 또 다른 해쉬함수의 일종인 H_3 를 이용하여 세션키 Key를 공유한다.

이 프로토콜은 만약 해커가 서버를 타협하여 패스워드 검증 데이터를 알아내면, 그 후 해커는 다른 사용자에 대하여 서버를 가장할 수 있게 되는 취약점이 있다. 따라서 일반적으로 패스워드 검증 데이터는 안전한 방법으로 저장되고 관리되어야 한다고 가정하고 있다. 그러나 이러한 가정은 비현실적이다. 따라서 패스워드 검증 데이터를 여러 개의 서버들에 분산하여 저장하고, 사용자를 인증할 때 정해진 개수 이상의 서버들이 프로토콜에 개입되어 사용자를 인증하는 분산형 PAK 프로토콜들이 제안되었다.

2.4 임계치 비밀분산 기법

Shamir는 임계치 기법의 구성을 위해 유한체의 다항식을 사용하였다. (t, n) 임계치 비밀 공유 기법은 n 참여자들에게 비밀에 관한 부분 정보를 분배하는 딜러(Dealer)가 있는 n 참여자들 사이의 프로토콜이다. 이러한 비밀 공유 기법은 t보다 작은 참여자들의 그룹이 비밀에 관한 어떠한 정보도 얻을 수 없으며, 적어도 t 참여자들의 그룹은 다항 시간(Polynomial Time)에 비밀을 계산할 수 있다[15]. Shamir의 방식과는 다르게 딜러 없이 각 참여자가 임의의 난수를 공유하기 위해서 연합 난수 비밀 공유 기법을 이용한다[18]. 하지만 이러한 프로토콜은 모든 참여자들이 정직하다는 가정하에서 유효하며, 실제로 악의 있는 적을 고려할 필요가 있다. 이러한 문제점을 해결하기 위해서 Feldman과 Pedersen의 검증 가능한 비밀 공유 방식을 이용할 수 있다[16,17].

분배된 비밀 정보를 보간하기 위해서 Lagrange의 보간 다항식(interpolation polynomial)을 이용할 수 있으며, 분배된 멱승 정보를 보간하기 위해서 멱승 보간법을 이용할 수 있다. 멱승 보간법은 $(a_1,\ldots,a_n) \xleftrightarrow{(t,n)} a \bmod q$인 a_i가 각 참여자들에게 분배되고, 각 참여자들이 $g^{a_i} \bmod p$로부터 $g^a \bmod p$를 계산하기 위한 방법이다[18,19].

또한 제안된 프로토콜에서는 참여자들 사이에 공유된 주어진 두 비밀 u와 v가 있을 때, 이 두 비밀을 밝히지 않고 두 비밀의 곱 $\mu \times \nu$를 계산하기 위한 프로토콜이 사용된다. 각각 t 차수의 다항식에 의해 공유된 주어진 μ, ν는 각 참여자들이 자신들의 μ, ν공유를 지역적으로 곱한다. 결과는 $2t$ 차수 다항식의 $\mu \times \nu$ 공유가 될 것이다. 따라서, 값 $\mu \times \nu$는 $2t+1$의 정당한 공유들의 집합으로부터 재구성된다[18].

이와 같은 비밀 분산 기법은 3장에서 제안되는 분산형 키 분배 프로토콜에 적용된다.

3. 제안하는 키 교환 프로토콜

본 장에서는 제안하는 분산형 키 교환 프로토콜에서 사용될 파라미터를 설명하고, PAK 기반의 MTI 키 분배 프로토콜들에 비밀 분산 방식을 적용한 프로토콜들을 제안한다. 제안하는 분산형 키 분배 프로토콜은 단일 서버 형태의 키 분배 프로토콜에서 요구되는 클라이언트측의 계산량과 동일한 수준으로 클라이언트측의 계산량을 요구한다. 즉, 서버측의 계산량 및 서버간 통신량은 증가하지만 클라이언트측의 계산량 및 통신량은 단일 서버 형태의 키 분배 프로토콜과 동일하다. 일반적으로 서버는 계산 능력이 뛰어난 시스템을 많이 사용하며, 클라이언트는 이동성이 편리하면서 계산 능력이 다소 떨어지는 스마트카드와 같은 시스템을 사용하므로 제안하는 프로토콜이 응용될 분야는 많을 것으로 생각된다.

본 논문에서 제안하는 프로토콜은 두 개의 비밀에 대한 곱 정보를 복구하기 위해서 $2t+1$개

이상의 서버들이 참여해야만 하며, 클라이언트가 프로토콜에 참여하는 서버들 각각과 세션키를 공유하기 위해서 클라이언트는 각 서버들과 해당 프로토콜을 반복해야만 한다.

3.1 사전 수행 절차

본 프로토콜은 인증이 이루어지는 클라이언트와 서버들간에 사전에 오프라인으로 안전하게 수행된다. 클라이언트는 패스워드 π, 클라이언트의 ID인 C, 그리고 서버의 ID인 S를 이용해서 $\nu=\alpha[C,S,\pi]$을 계산하고, 계산된 v를 이용하여 각 서버들에 대한 Shamir의 비밀 공유 방식을 이용하여 부분정보인 ν_i를 계산하고, 이 부분정보를 멱승한 값인 g^{ν_i}을 안전한 방법으로 프로토콜에 참여하고 있는 대표 서버에게 전달한다. 또한 클라이언트는 공개 정보인 $y_C=g^{X_C}\bmod p$를 계산하고 클라이언트의 공개키를 공개한다. 여기서 $\alpha[C,S,\pi]$는 입력되는 값이 C와 S라는 점을 제외하고 2.1절에서 기술한 것과 동일하다.

서버들은 연합 난수 비밀 방식을 이용해서 서버의 비밀키인 X_S를 공유하고, 프로토콜에 참여하는 서버 i들은 대표 서버 i에게 자신의 부분정보의 멱승값을 $g^{X_{Si}}\bmod p$를 안전하게 전달한다. 서버 i는 수신된 정보를 바탕으로 멱승 보간법을 이용하여 서버들의 공개키인 $y_S=g^{X_S}\bmod p$를 계산하고 이를 공개한다. 이 절에서 기술한 내용은 본 논문에서 제안하는 2가지 프로토콜들에 적용된다.

3.2 분산형 키 분배 프로토콜 제안(1)

본 절에서는 그림 5와 같이 분산 서버 환경에 적용 가능한 PAK 기반의 MTI 분산형 키 분배 프로토콜을 제안한다.

Step 1, 클라이언트가 랜덤하게 $r_C\in_R Z_p$를 선택하고 $m=g^{r_C}\cdot(H_1(C,S,\pi))^r$을 계산한다. 클라이언트는 m을 서버 i에게 전송하고, 서버 i는 각각의 서버들에게 m을 전달한다.

Step 2, 서버 i는 다음과 같은 과정으로 $\omega=g^{r_c a}\bmod p$을 계산한다.

① a는 연합난수 비밀공유 기법으로 계산된다. 각각의 서버들로부터 $(a_1,a_2,\cdots,a_n)\xrightarrow{(t,n)}a\bmod p$을 계산한다. 여기서 a는 각각의 각 서버들이 모르는 랜덤한 값이고, 서버 j는 a에 대한 부분정보인 a_j를 소지한다.

② 각각의 서버 j는 $m^{a_j}\bmod p$을 계산하고, 서버 i에게 보낸다. 멱승보간법을 이용하여 서버 i는 $m^a\bmod p$을 계산한다.

③ 각각의 서버 j는 $(g^{\nu_j})^{a_j}\bmod p$을 계산하여 서버 i에 전달한다. 멱승보간법을 이용하여 서버 i는 $g^{a\nu}\bmod p$을 계산한다.

④ 서버 i는 $H_1(C,S,\pi)^{ra}(=g^{\nu ra}\bmod p)$, $\omega=m^a/H_1(C,S,\pi)^{ra}\bmod p=g^{r_C a}$를 계산하고, 각각의 서버들에게 $\omega=g^{r_C a}\bmod p$을 전달한다.

Step 3, 각각의 서버j는 $\omega^{X_{S_j}}$를 계산하고, 대표 서버 i에게 이를 전달한다. 서버 i는 각 서버들로부터 $\omega^{X_{S_j}}$을 수신하여 멱승보간법을 이용하여 $(\omega^{X_{S_1}},\omega^{X_{S_2}},\cdots,\omega^{X_{S_n}})\xrightarrow{(t,n)}\tau=\omega^{X_S}\bmod p$ 을 계산하고, 계산된 τ을 각각의 서버들에게 전달한다.

Step 4, $\sigma=g^{X_s r_C+X_C r_S}\bmod p$는 다음과 같은 과정을 통해서 계산된다.

① 각 서버들은 난수 b를 연합난수 비밀공유 기법을 이용하여 계산한다. 각 서버들은 $(b_1,b_2,\cdots,b_n)\xrightarrow{(t,n)}b\bmod p$을 공유하고, 서버 i에게 자신의 부분 비밀값 τ^{b_j} 를 전달한다. 또한 각 서버들은 $a_j b_j\bmod p$을 계산하여 서버 i에게 전달한다.

② 서버 i는 멱승보간법을 이용하여 $(\tau^{b_1},\tau^{b_2},\cdots,\tau^{b_n})\xrightarrow{(t,n)}\tau^b\bmod p$을 계산하고,$ab\bmod p$, $(ab)^{-1}\bmod p$와

$(\tau^b)^{(ab)^{-1}} \bmod p = g^{r_C X_S} \bmod p$을 계산한다.

③ 각 서버들은 연합난수 비밀공유 기법을 이용하여 임의의 난수 r_S $(r_{S_1}, r_{S_2}, \cdots, r_{S_n}) \xrightarrow{(t,n)} r_S \bmod p$을 공유한다. 그리고 각 서버들은 $g^{r_{S_i}}$ 와 $y^{r_{S_j}}_{\bmod pC}$ 을 계산하여 서버 i에게 전달한다.

④ 서버 i는 $\sigma = g^{X_S r_C + X_C r_S} \bmod p$을 계산한다.

Step 5, 서버 i는 다음과 같은 과정으로 μ, k를 계산하여 클라이언트에게 전달한다.

① 각각의 서버들은 $g^{r_{S_i}}$을 계산하여 서버 i에게 전달한다.

② 서버 i는 $\mu = g^{r_S} \bmod p$, $k = H_{2a}(y_C, y_S, m, \mu, \sigma)$을 계산하여 클라이언트에게 전달한다.

Step 6, 클라이언트는 σ와 $H_{2a}(y_C, y_S, m, \mu, \sigma)$을 계산하고, 서버 i로부터 받은 값과 자신이 계산한 값이 일치한가를 검증한다.

Step 7, 클라이언트는 $k' = H_{2b}(y_C, y_S, m, \mu, \sigma)$ 을 계산하여 서버 i에게 전달하고, 서버 i는 클라이언트로부터 받은 값을 검증한다. 결과적으로, 클라이언트와 서버 i는 암호학적으로 강한 세션키 Key를 공유한다.

그림 5에서 나타난 프로토콜에서 보는 바와 같이 $(H_1(C,S,\pi))^r$ 정보가 각 서버에 공개되는 것을 방지하기 위해서 임의의 난수 a를 이용하여 이를 은폐시켰다. 또한 클라이언트와 서버 i간에 인증 및 동일한 세션키가 공유되었음을 확인하기 위해 해쉬 함수를 이용하고 있으며, 이를 이용하여 세션키 확신 기능을 제공한다.

Step	Client		Server i (server 1, ···, server n)
1	$y_C = g^{X_C} \bmod p$ $r_C \in {}_R Z_p$ $m = g^{r_c} \cdot (H_1(C,S,\pi))^r$	m --->	
2			$a \in {}_R Z_p$ 을 연합난수 공유기법을 이용하여 분배함 멱승 보간법을 이용하여 $m^a \bmod p$ 계산 $H_1(C,S,\pi)^{ra}$ 계산 $\omega = g^{ra} \bmod p$ 계산하여 공유함
3			$\tau = g^{r_c a X_S} \bmod p$ 계산
4			$\sigma = g^{X_S r_C + X_C r_S} \bmod p$계산
5		μ, k <---	$\mu = g^{r_S}$ 계산 $k = H_{2a}(y_C, y_S, m, \mu, \sigma)$ 계산
6	$\sigma = \mu^{X_C} y_S^{r_C}$ $= g^{X_C r_S + X_S r_C} \bmod p$ $k \stackrel{?}{=} H_{2a}(y_C, y_S, m, \mu, \sigma)$		
7	$k' = H_{2b}(y_C, y_S, m, \mu, \sigma)$ $Key = g^{X_C r_S + X_S r_C} \bmod p$	k' --->	$k' \stackrel{?}{=} H_{2b}(y_C, y_S, m, \mu, \sigma)$ $Key = g^{X_C r_S + X_S r_C} \bmod p$

〈그림 5〉 분산형 키 분배 프로토콜 제안(1)

3.3 분산형 키 분배 프로토콜 제안(2)

이 절에서는 PAK-R 기반의 MTI 분산형 키 분배 프로토콜을 제안한다. 이 프로토콜의 기본 절차는 3.2절의 분산형 키 분배 프로토콜 제안(1)과 비슷하지만 다음과 같은 점이 다르다.

Step 1, 클라이언트는 난수 r_c 와 h 를 이용하여 $m=g^{r_C}\cdot h^q\cdot H_1(C,S,\pi)$을 서버 i에게 전송하고, 서버 i는 각각의 서버들에게 m을 전달한다.

Step 2, 서버 i는 다음과 같은 과정으로 $\omega=g^{r_i a}\bmod p$을 계산한다.

①, ②, ③은 3.2절의 분산형 키 분배 프로토콜 제안(1) Step 2의 ①, ②, ③ 과정과 동일하다.

④ 서버 i는 $H_1(C,S,\pi)^{ra}(=g^{\nu ra}\bmod p)$, $\omega=(m^{ra}/H_1(C,S,\pi)^{ra})^{r_i^{-1}}\bmod p$를 계산하고, 각각의 서버들에게 $\omega=g^{r_i a}\bmod p$을 전달한다.

Step 3에서 Step 7까지의 과정은 3.2절의 분산형 키 분배 프로토콜 제안(1) Step 3에서 Step 7과 동일하다.

제안된 프로토콜에서와 같이 클라이언트가 서버에게 전달하는 메시지 m에서 PAK 프로토콜과는 다르게 $H_1(C,S,\pi)^r$을 계산하지 않고, h^q를 계산한다. q는 160 비트의 소수이므로 클라이언트에서 계산되는 계산량이 감소하게 된다. 이러한 방식은 무선 단말기나 스마트카드와 같은 계산 능력이 부족한 이동식 장비에 적용 가능하다.

4. 안전성 및 성능 분석

4.1 안전성 분석

본 논문에서 제안된 클라이언트와 다중 서버를 이용한 인증 및 키 분배 프로토콜은 다음과 같은 공격에 강한 안정성을 가지고 있다.

① 오프라인 패스워드 유추공격

패스워드 기반 인증 프로토콜에서의 가장 큰 취약점은 패스워드에 대한 사전공격(dictionary attack)이다. 이 공격은 다시 오프라인 공격과 온라인 공격으로 구분될 수 있는데, 오프라인 공격은 클라이언트와 서버 사이에서 정당한 인증이 수행되는 동안에 전송되는 메시지를 가로채어 오프라인 상태에서 공격자가 추측한 패스워드 사전의 목록을 차례로 대입하는 전사적공격(brute-force attack) 방법으로 패스워드를 비교하는 방법이다. 이러한 공격은 공격자의 컴퓨터 사양이 뛰어나거나 여러대의 컴퓨터를 동시에 이용할 경우 성공률이 상당히 높다. 온라인 공격은 사용자가 각기 다른 패스워드를 사용하여 서버에 온라인 상태에서 인증을 시도하는 공격인데, 이는 서버에서 인증 횟수를 제한하는 것에 의해서 쉽게 해결할 수 있다[24].

제안된 분산형 키 분배I 프로토콜에서 공격자가 전송되는 메시지 m,μ을 얻어도 DHP를 해결하지 못하므로 정확한 임의의 난수 r_C,r_S를 알 수 없다. 따라서 전사적공격으로 사용자의 패스워드를 구할 수 없을 것이다. 만약 공격자가 임의의 난수 r_C', 임의의 패스워드 π'와 공개정보를 통해 메시지 m'을 만들어 서버에게 전송하여도 사전에 공유된 패스워드가 아님을 서버가 알게 된다. 또한 여기서 전송되는 m,μ,k는 해쉬값 (H_1,H_{2a},H_{2b})으로 인증되기 때문에 정확한 해쉬값을 모르면 이를 해결하지 못한다.

② 수동적 공격에 대한 안전성

제안된 프로토콜에서 X_C,X_S는 유한체 $GF(p)$에서 생성원 g가 있을 때, $GF(p)$의 원소 y_C,y_S는 $0<y_C,y_S<p-1$의 범위를 가진 정수이며, g^{X_C},g^{X_S}에서 X_C,X_S를 y_C,y_S의 차수(degree)라고 하며, y_C,y_S는 g를 X_C,X_S번 곱한 결과이다. 이 때, X_C,X_S를 알면 y_C,y_S를 구하기 쉬우나, y_C,y_S를 알아도 X_C,X_S을 구하는 것은 이산대수 문제로 구하기 어렵다. 따라서 클라이언트와 서버의 공개 정보 y_C,y_S를 공개하여도 공격자는 정확한 X_C,X_S을 구하는 것은 어렵다. 또한 이

값을 모르면 프로토콜에서 정확한 τ, σ값을 구할 수 없다.

③ PFS(Perfect Forward Secrecy)의 만족

PFS는 통신 주체들간에 장기간 비밀키가 노출되더라도, 공격자가 통신 주체들간의 과거 세션키를 계산할 수 없는 경우를 PFS를 만족한다고 정의한다. 제안된 분산형 키 분배 프로토콜은 임의의 난수 r_C, r_S와 사전에 공유된 패스워드 π를 모르면 정확한 m, μ를 구할 수 없다. 그리고 매 세션마다 사용되는 세션키에 세션마다 다른 난수(a, b, r_C, r_S)를 사용하므로 정확한 세션키와 이전에 사용된 세션키를 알 수 없다. 또한 제안 조건에서 비밀분산기법을 이용하여 비밀정보를 다중 서버에게 분산하여 저장하였기 때문에 일부 서버와의 결탁으로 공격이 이루어질 수 없다.

④ 수동적 도청 공격(Passive Eavesdropping)

수동적 도청은 공격자가 클라이언트와 서버 사이에서 교환되는 메시지를 도청할 수 있고, 이들 사이에서 공유된 비밀 정보와 통신 내용, 세션키를 구하려고 시도한다. 그러나 수동적 도청 공격자는 임의의 메시지를 바꾸거나, 삭제 또는 삽입하는 것은 불가능하다. 이 공격은 DLP의 어려움을 근거로 해결 가능하다.

제안된 분산형 키 분배 프로토콜에서 공격자가 공개 정보 p, q, g, y_C, y_S를 알고, 클라이언트와 서버 사이에서 교환되는 m, μ, k를 알고 있다하더라도, 공격자가 g^{X_C}, g^{X_S}를 구하는 것은 DLP를 풀어내는 것만큼 어렵다. 또한 통신 주체간에 인증을 위한 파라미터와 해쉬값(H_1, H_{2a}, H_{2b})은 전송되지 않고 통신 주체가 직접 계산하므로 도청 공격을 통해서는 알 수 없다.

⑤ 재전송공격(Replay attack)

재전송공격은 공격자가 클라이언트의 메시지 m을 서버에게 재전송하여 이미 정상적인 클라이언트에 의해 생성된 이전키(old session key)를 다시 생성하기 위함이다. 그러나 모든 통신 메시지들은 매 세션마다 균일한 확률 분포에서 랜덤하게 생성되므로 이 공격에 대한 공격자의 성공 확률은 무시할 수 있다. 또한 서버와 클라이언트는 3개의 서로 다른 해쉬 함수(H_1, H_{2a}, H_{2b})를 사용하므로 메시지와 비밀정보가 보호된다.

⑥ 능동적 중간자 공격(Positive Man-in-the-middle attack)

능동적 중간자 공격은 공격자가 클라이언트와 서버 사이에서 합법적으로 가장하거나 전송되는 메시지를 가로챈 다음, 공격자와 클라이언트, 공격자와 서버 사이에 각각의 별도의 세션값을 만들어내는 공격이다. 공격자는 프로토콜 내의 모든 대화내용을 이용하더라도 사전에 공유된 패스워드 π를 모른다면 제안된 분산형 키 분배 프로토콜에서 $k \stackrel{?}{=} H_{2a}(y_C y_S m, \mu, \sigma)$, $k \stackrel{?}{=} H_{2b}(y_C y_S m, \mu, \sigma)$ 검사를 통과하지 못한다.

⑦ 서버-타협 공격에 면역성 제공

기본적으로 제안된 프로토콜은 패스워드 검증 정보와 서버의 공개키에 대한 비밀키를 비밀분산 기법을 이용하여 여러 서버들이 공유하고 있다. 따라서 공격자가 하나의 서버를 공격해도 사용자를 인증하거나 서버를 흉내 낼 수 있는 패스워드 검증 데이터에 대한 부분정보만을 얻을 수 있다. 만약 t개 이상의 서버를 타협하면 가능하나, 이러한 가능성은 매우 낮다. 따라서 제안된 프로토콜은 서버-타협 공격에 면역성이 있음을 알 수 있다.

4.2 성능 분석

본 성능 분석에서의 t는 2.4절의 Shamir 임계치 기법으로 유한체의 다항식에서, n명의 참여자들에게 비밀에 관한 부분 정보를 분배하고 n명의 참여자들 중에 t보다 작은 참여자들의 그룹이 비밀에 관한 어떠한 정보도 얻을 수 없으며, 적어도 t 참여자들의 그룹은 다항 시간(Polynomial Time)에 비밀을 계산할 수 있음을 의미한다. 표 1에서

〈표 1〉 기존 프로토콜과의 계산량 비교

프로토콜	라운드	S_i		클라이언트	
		라운드	멱승	라운드	멱승
MacKenzie의 임계치 PAKE 프로토콜	6t	6t	22+34t	3t	15+t
분산형 키 분배 프로토콜 제안(1)	11t	11t	15t	3t	4t
분산형 키 분배 프로토콜 제안(2)	11t	11t	13t	3t	4t

는 MacKenzie 등이 제안하는 임계치 PAKE 프로토콜과 제안 프로토콜들과의 계산량을 비교한다 [23]. 제안하는 프로토콜들은 전체 라운드 수가 MacKenzie의 임계치 PAKE 프로토콜에 비해서 월등히 많다. 이것은 제안하는 프로토콜들이 서버들간에 연합 난수 비밀 공유와 같은 비밀 공유 기법이 사용됨으로써 라운드 수가 크게 증가한 것이다. 이것은 전체적으로 서버들간에 교환되는 메시지의 양이 많음을 의미한다. 하지만 클라이언트가 관여하는 라운드 수는 MacKenzie의 임계치 PAKE 프로토콜과 동일하다. 또한 클라이언트에서 수행하는 멱승 연산의 수는 t가 5보다 작으면 제안하는 프로토콜이 MacKenzie의 임계치 PAKE 프로토콜 보다 동일하거나 작게 된다. 단 t는 실제 구현에 있어 5보다 작은 값을 사용하는 것이 일반적이다. 반면에 서버간에 수행되는 멱승 연산의 횟수는 제안하는 프로토콜이 효율적이다.

제안 방식(1), (2) 이외에 MTI의 다른 방법과 PAK, PAK-R를 이용하여 프로토콜을 설계할 수 있지만, 이들 4가지 프로토콜을 서버측 및 클라이언트측 계산량을 종합적으로 비교해 보면 제안방식 (2)이 가장 효율적이다.

5. 결 론

본 논문에서는 PAK, MTI 키 분배 프로토콜을 이용하여 분산 서버와 단일 클라이언트 모델에 적용 가능한 분산형 키 분배 프로토콜을 제안하였다. 지금까지 대부분의 PAK 프로토콜은 DH 키 분배 프로토콜에 기반을 두고 설계되었다. 본 논문에서는 DH 키 분배 프로토콜의 변형인 MTI 키 분배 방식을 이용하고 통신주체간에 인증을 위해 PAK 프로토콜을 이용하였으며, 이를 서버 타협 공격을 피하기 위하여 분산형 키 분배 프로토콜을 제안하였다. 제안하는 분산형 키 분배 프로토콜들은 클라이언트측의 계산량 및 통신량을 단일 서버 형태의 키 분배 프로토콜과 동일한 수준으로 낮추었으며, 기존의 방식에 비해서 연산량 측면에서 효율적이다.

본 논문에서 제안한 방식은 스마트카드에 적용할 수 있으며, 스마트카드에 있는 비밀키와 사용자가 인증시에 입력하는 패스워드를 이용하여 키 분배와 사용자 인증을 수행할 수 있다. 따라서, 스마트카드가 분실되더라도 스마트카드 습득자는 패스워드를 알지 못하기 때문에 서버와 PAK 프로토콜을 성공적으로 수행할 수 없다는 특성이 있다. 본 방식은 인터넷, 무선통신 그리고 모바일 통신 환경에서 서버의 개수가 여러 개인 경우에 적용될 수 있는 분산형 키 분배 프로토콜이다. 앞으로 애드혹 망과 같은 모바일 통신 환경에서 매우 유용하게 활용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] Victor Boyko, Philip MacKenzie and Sarvar Patel, "Provably Secure Password Authentication and key Exchange Using Diffie-Hellman(extended abstract)," Euro Crypt 2000, pp.156-171, 2000.

[2] P. MacKenzie, "More Efficient Password-Authenticated Key Exchange," RSA Conference, Cryptographer's Track, pp. 361-377, 2001.

[3] T. Matsmoto, Y. Takashima and H. Imai, "On Seeking Smart Public-key Distribution Systems," The Transaction of the IECE of Japan, E69, pp.99-106, 1986.

[4] William Stalling, Cryptographic and Netwokk Security, Prentice Hall, 1999.

[5] S.Blake-Wilson, and A. Menezes, "Authenticated Diffie-Hellman Key Agreement Protocol," Selected Areas in Cryptography-SAC '98 Proceeding, pp. 339-361, 1999.

[6] Taekyoung Kwon, Myeongho Kang, Sangjoon Jung and Jooseok Song, "An Improvement of the Password-based Authentication Protocol(K1P) on Security against Replay Attacks",IEICE Transactions on Communications, vol.E82-B, no.7, July 1999, pp.991-997

[7] T.Kwon and J.Song, "A Secure Agreement Scheme for g^xy via Password Authentication", Electronics Letters, vol.35, no. 11, pp.892-893, 27th May 1999

[8] IEEE. IEEE 1363, "Standard Specifications for Public Key Cryptography," 2000.

[9] M. Bellare, R. Canetti, and H. Krawczyk, "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols,"Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pp.419-428. 1998.

[10] Jonathan Katz, Rafail Ostrovsky, Moti Yung: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. EUROCRYPT 2001: 475-494

[11] S. M. Bellovin and M. Merritt. "Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks," In Proceedings of IEEE Security and Privacy, pp. 72-84, 1992.

[12] D. Jablon. "Strong Password-only Authenticated Key Exchange." ACM Computer Communication Review. ACM SIGCOMM, 26(5):5-20, 1996

[13] D. Jablon. "Extended Password Key Exchange Protocols immune to Dictionary Attack," In WETICE'97 Workshop on Enterprise Security, 1997.

[14] Jonathan Katz, Moti Yung: Threshold Cryptosystems Based on Factoring. ASIACRYPT 2002: 192-205

[15] A. Shamir, "How to Share a Secret," Communications of the ACM, 1979

[16] P. Feldman, "A Practical Scheme for Non-interactive Verifiable Secret Sharing", In Proc of the 28th IEEE Symposium on the Foundations of Computer Science, pp 427-437, 1987

[17] T.Pedersen, "Non-interactive and Information-theoretic Secure verifiable Secret Sharing," Proc. CRYPTO 91, pp129-140, 1991

[18] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, "Robust Threshold DSS Signature," Advances in Cryptology Proc of Eurocrypt '96, pp.354-371, 1996.

[19] Rosario Gennaro, Michael Rabin, Tal Rabin, "Simplified VSS and Fast-Track Multiparty Computations with Applications to Threshold cryptography, "http://theory.lcs.mit.edu/~rosario/research.html, preprint, 1998

[20] V. Shoup. "Practical threshold signatures," In advance in Cryptology - EUROCRYPT 2000, pp.207-220, May 2000.

[21] X. Wang, "Intrusion-Tolerant Password-

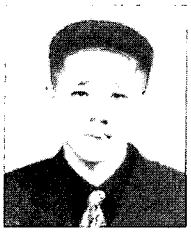
Enabled PKI," 2nd Annual PKI Research Workshop,

[22] X. Wang, "A Distributed Password-Authenticated Key Exchange Protocol Secure Against Server Compromise-Based Dictionary Attacks," Department of Computer Science, James Madison University, Harrisonburg, VA, USA, January 2003.

[23] Philip MacKenzie, Thomas Shrimpton, Markus Jakobsson, "Threshold Password-Authenticated Key Exchange(extended abstract)," Advances in Cryptology Proc of CRYPTO 2002, pp.385-400, 2002.

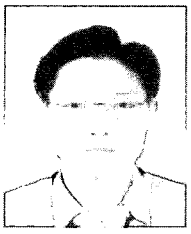
[24] Shai Halevi, Hugo Krawczyk, "public-key cryptography and password protocols", ACM Transactions on Information and System Security, Vol.2, No.3, pp.203-268 August 1999.

◐ 저 자 소 개 ◑



오 흥 룡

2002년 2월 순천향대학교 전자공학과 졸업(학사)
2004년 2월 순천향대학교 대학원 정보보호학과 졸업(석사)
2004년 2월~현재 한국정보통신기술협회(TTA) 표준화본부
관심분야 : 정보보호 표준, 보안 프로토콜
E-mail : hroh@tta.or.kr



윤 호 선

1997년 2월 순천향대학교 전자공학과 졸업(학사)
1999년 2월 순천향대학교 대학원 전기·전자공학과 졸업(석사)
1999년 3월~2000년 2월 순천향대학교 산업기술연구소
2000년~현재 한국전자통신연구원 광대역통합망연구단
관심분야 : 네트워크 보안, VPN 기술, 보안 프로토콜
E-mail : yhs@etri.re.kr

염 흥 열

1981년 2월 한양대학교 전자공학과 졸업(학사)
1983년 2월 한양대학교 대학원 전자공학과 졸업(석사)
1990년 2월 한양대학교 대학원 전자공학과 졸업(박사)
1982년 12월~1990년 9월 한국전자통신연구소 선임연구원
1990년 9월~현재 순천향대학교 공과대학 정보보호학과 교수
1997년 3월~2000년 3월 순천향대학교 산업기술연구소 소장
2000년 4월~현재 순천향대학교 산학연컨소시엄센터 소장
1997년 3월~현재 한국정보보호학회 총무이사, 학술이사, 교육이사
2004년 1월~현재 한국인터넷정보학회 이사, 논문지 편집위원
2003년 9월~2004년 3월 ITU-T SG17/Q10, Associate Rapporteur
2004년 3월~현재 ITU-T SG17/QL Rapporteur
관심분야 : 네트워크보안, 전자상거래보안, 공개키 기반구조, 부호이론, 이동통신보안
E-mail : hyyoum@sch.ac.kr