情報保護學會論文誌
第16卷 第4號, 2006. 8

Signcryption을 이용한 안전한 인증된 키 교환 프로토콜 연구*

김 락 현,† 염 흥 열‡
순천향대학교

Secure Authenticated key Exchange Protocol using Signcryption Scheme*

Rack-hyun Kim,† Heung-Youl Youm‡
Soonchunhyang University

요 약

1997년 Yuliang Zheng에 의해 제안된 Signcryption은 서명과 암호 기법을 결합한 하이브리드 공개키 프리미티브로서 서명 기법 후 암호 기법을 각각 적용한 기법보다 계산 및 통신비용 측면에서 높은 효율성을 갖는 기법이다. 또한 PAK(Password-Authenticated Key Exchange) 프로토콜은 사용자가 암기하거나 휴대하기 쉬운 짧은 길이의 패스워드를 기반으로 통신 주체를 상호 인증하고, 결과적으로 안전한 통신을 위하여 충분히 큰 길이의 세션키를 분배하는 프로토콜이다. 본 논문에서는 PAK와 Signcryption의 특징을 이용하여 참여자의 비밀 정보를 이용한 상호 인증 및 안전한 통신을 위한 키 분배 프로토콜을 제안한다. 그리고 제안 프로토콜의 보안성을 증명하고 효율성을 비교한다.

ABSTRACT

A Signcryption proposed by Yuliang Zheng in 1997 is a hybrid public key primitive that combines a digital signature and a encryption. It provides more efficient method than a straightforward composition of an signature scheme with a encryption scheme. In a mobile communication environment, the authenticated key agreement protocol should be designed to have lower computational complexity and memory requirements. The password-based authenticated key exchange protocol is to authenticate a client and a server using an easily memorable password. This paper proposes an secure Authenticated Key Exchange protocol using Signcryption scheme. In Addition we also show that it is secure and a more efficient that other exiting authenticated key exchange protocol.

Keywords : Signcryption, PAK, Authentication Key Exchange

Ⅰ. 서 론

Signcryption 기법은 인증성을 제공하는 서명 과정 후 비밀성을 제공하는 암호화과정을 수행하는 하이브리드 기법으로 서명기법과 암호기법을 효율적으로 결합한 기법이다. Signcryption은 하나의 논리적인 단계에 인증과 비밀성

접수일: 2006년 6월 7일 ; 채택일: 2006년 7월 31일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음 (IITA-2005-(C1090-0502-0020))
† 주저자, rhkim@sch.ac.kr
‡ 교신저자, hyyoum@sch.ac.kr

두 가지 모두를 지원하는 기법으로 작은 계산량과 통신상에서의 작은 크기의 오버헤드를 유지하기 때문에, 많은 연구가 이루어지고 있다[1-3]. 또한, 패스워드 기반 인증 키 분배 프로토콜은 사용자가 암기하거나 휴대하기 쉬운 짧은 길이의 패스워드를 기반으로 하여 통신 주체를 상호 인증하고, 결과적으로 안전한 통신을 위하여 충분히 큰 길이의 세션키를 분배하는 프로토콜이다[4-7]. 그리고 패스워드 기반 인증 키 분배 프로토콜은 안전한 통신을 위하여 특별한 휴대품(인증서, ID-Card 등)이 필요가 없고, 이동이 용이하다는 장점 때문에 중요한 연구 대상이 되고 있다. 적용 가능한 통신 모델을 살펴보면 일반적인 두 사용자간의 통신 모델, 응용 서버와 클라이언트 통신 모델, 그리고 모바일단말기와 서버(기지국) 모델 등 다양한 통신 구조에 적용이 가능하다.

본 논문에서는 Signcryption을 이용하여 두 통신 당사자간의 패스워드를 기반으로 상호인증과 암호 기능을 제공하는 키 분배 프로토콜을 제안하고 안전성과 효율성을 분석한다.

본 논문의 구성은 다음과 같다. 먼저 Ⅱ장에서 관련연구로서 Zheng에 의해 제안된 Signcryption 기법 및 연구 동향을 설명하고, Ⅲ장에서는 패스워드 기반의 인증 키 분배 프로토콜에 관하여 기술한 후, Ⅳ장에서는 Signcryption을 이용한 패스워드 기반의 키 분배 프로토콜을 제안하고, Ⅴ장에서는 보안 요구 사항, 효율성 그리고 기능성을 분석한다. 마지막으로 Ⅵ장에서는 결론을 기술한다.

Ⅱ. 관련연구 1 (Signcryption)

1. Signcryption

Zheng은 서명과 대칭키 암호화를 논리적인 한 단계에서 수행하는 Z_p로 정의되는 곱셈군 기반 Signcryption을 1997에 제안하였다. 그리고 1998년에는 타원 곡선에서 덧셈군을 기반으로 한 기법을 제안하였다[1].

다음 표 1은 Zheng이 제안한 Signcryption에서 사용되는 파라미터를 정의한 것이고, 그림 1은 제안한 Signcryption과 Unsigncryption의 기본 프로토콜이다.

이 때, Zheng은 Alice의 Signcryption과정에서 ⑤의 s와 ⑧의 w의 계산방법에 따라 3가지 기법으로 나누어 소개하였다. 표 2는 3가지 기법을 비교한다.

표 1. Signcryption의 공개 및 비밀 파라미터

Alice의 키	공개	Bob의 키
x_a:개인키 y_a:공개키 $(y_a = g^{x_a} \bmod p)$	p:큰 소수 q:큰 소수 (factor of $p-1$) g:$0<g<p$이고, 차수 q mod p G, H: 일방향 해쉬 (E, D): 대칭키 암·복호 알고리즘	x_b:개인키 y_b:공개키 $(y_b = g^{x_b} \bmod p)$

Alice(Signcryption)	Bob(Unsigncryption)
① 랜덤선택 $x \in_R \{1, .., q-1\}$ ② $w = y_b^x \bmod p$ ③ $k = G(w)$ ④ $r = H(m, bind\_info, w)$ ⑤ $s = x/(r+x_a) \bmod q$ ⑥ $c = E_k(m)$ ⑦ (c, r, s) 전송	⑧ $w = (y_a \cdot g^r)^{s \cdot x_b} \bmod p$ ⑨ $k = G(w)$ ⑩ $m = D_k(c)$ ⑪ $if(r \stackrel{?}{=} H(m, bind\_info, w))$ m 수락

그림 1. Signcryption과 Unsigncryption 기본 프로토콜

표 2. s와 w의 계산과정에 따른 3가지 기법

Alice(Signcryption)	Bob(Unsigncryption)
1) $s = x/(r+x_a) \bmod q$	1) $w = (y_a \cdot g^r)^{s \cdot x_b} \bmod p$
2) $s = x/(1+x_a \cdot r) \bmod q$	2) $w = (g \cdot y_a^r)^{s \cdot x_b} \bmod p$
3) $s = (x - x_a \cdot r) \bmod q$	3) $w = (g^s \cdot y_a^r)^{x_b} \bmod p$

2. Signcryption을 이용한 키 교환 프로토콜

Zheng은 1998년 Signcryption을 이용하여, 두 사용자간에 위조 불가능하고 신선한 세션키 교환이 가능한 프로토콜을 제안하였다.[3] 제안한 논문에서 Zheng은 키를 교환하기 위해 두 가지 프로토콜을 소개하고 있다. 이 두 프로토콜은 세션키의 신성성을 위해 랜덤 수를 이용하는 방법(DKEUN)과 타임스탬프(DKEUTS)를 이용하는 방법으로 구분된다. 그림 2와 3은 Zheng의 키 교환 프로토콜을 나타낸다.

Zheng의 키 분배 프로토콜의 특징 중 첫 번째는

Direct Key Exchange Using a Nonce (Protocol DKEUN)		
Alice (A)		Bob(B)
	$\Leftarrow NC_b \Leftarrow$	$NC_b \in_R \{0,1\}^{l_2}$
$key \in_R \{0,1\}^{l_1}$ $x \in_R [1, \ldots, q-1]$ $(k_1, k_2) = hash(y_b^x \bmod p)$ $c = E_{k_1}(key)$ $r = KH_{k_2}(key, NC_b, etc)$ $s = (x/(x+x_a)) \bmod q$	$\Rightarrow c,r,s \Rightarrow$	$(k_1, k_2) = hash((y_a \cdot g^r)^{x_b} \bmod p)$ $key = D_{k_1}(c)$ Accept key only if $KH_{k_2}(key, NC_b, etc) = r$
$(k_1^*, k_2^*) = hash((y_b \cdot g^{r^*})^{x_a} \bmod p)$ $key^* = D_{k_1^*}(c^*)$ Accept key^* only if $KH_{k_2^*}(key^*, key, etc) = r^*$	$\Leftarrow c^*,r^*,s^* \Leftarrow$	$key^* \in_R \{0,1\}^{l_1}$ $x^* \in_R [1, \ldots, q-1]$ $(k_1^*, k_2^*) = hash(y_b^{x^*} \bmod p)$ $c^* = E_{k_1^*}(key^*)$ $r^* = KH_{k_2^*}(key^*, key, etc)$ $s^* = (x^* / (r^* + x_b)) \bmod q$
$tag = MAC_{key@key^*}(NC_b)$	$\Rightarrow tag \Rightarrow$ optional	Verify whether $tag = MAC_{key@key^*}(NC_b)$

그림 2. Nonce를 이용한 직접 키 교환 프로토콜

Direct Key Exchange Using a Time-Stamp (Protocol DKEUTS)		
Alice (A)		Bob(B)
$key \in_R \{0,1\}^{l_1}$ $x \in_R [1, \ldots, q-1]$ $(k_1, k_2) = hash(y_b^x \bmod p)$ Get a current time-stamp TS $c = E_{k_1}(key, TS)$ $r = KH_{k_2}(key, TS, etc)$ $s = (x/(x+x_a)) \bmod q$	$\Rightarrow c,r,s \Rightarrow$	$(k_1, k_2) = hash((y_a \cdot g^r)^{x_b} \bmod p)$ $(key, TS) = D_{k_1}(c)$ Accept key only if TS is fresh and $KH_{k_2}(key, TS) = r$
$(k_1^*, k_2^*) = hash((y_b \cdot g^{r^*})^{x_a} \bmod p)$ $(key^*, TS) = D_{k_1^*}(c^*)$ Accept key^* only if TS^* is fresh and $KH_{k_2^*}(key^*, TS^*, key, etc) = r^*$	$\Leftarrow c^*,r^*,s^* \Leftarrow$	$key^* \in_R \{0,1\}^{l_1}$ $x^* \in_R [1, \ldots, q-1]$ $(k_1^*, k_2^*) = hash(y_b^{x^*} \bmod p)$ Get a current time-stamp TS $c^* = E_{k_1^*}(key^*, TS^*)$ $r^* = KH_{k_2^*}(key^*, TS^*, key, etc)$ $s^* = (x^* / (r^* + x_b)) \bmod q$
$tag = MAC_{key@key^*}(TS)$	$\Rightarrow tag \Rightarrow$ optional	Verify whether $tag = MAC_{key@key^*}(TS)$

그림 3. Time-Stamp를 이용한 직접 키 교환 프로토콜

상호 인증을 위해 Signcryption을 양방향으로 두 번 사용한다는 것과 두 번째는 키의 신성성 유지와 위조를 불가능하게 하기 위해 랜덤수와 타임스탬프를 이용한다는 것이다. 최종 두 사용자가 공유하게 되는 세션키는 Nonce와 타임스탬프를 입력으로, 키를 이용한 해쉬를 통해 생성한다.

Ⅲ. 관련 연구 2(패스워드 기반 인증 키 분배 프로토콜)

패스워드 기반의 인증 키 분배 프로토콜의 개념은 1992년 S. Bellovin과 M. Merritt에 의해 제안되었다. PAK는 패스워드를 기반으로 상호 인증 및 키 교환을 수행할 수 있는 프로토콜로써, PKI와 같은 인프라가 필요 없는 장점이 있다. 또한, Victor Boyko, Philip MacKenzie와 Sarvar Patel은 PAK 프로토콜과 여러 가지 PAK 변형 프로토콜을 설계하였는데[2], Diffie-Hellman 멱승보간법 프로토콜 기반의 PAK 프로토콜, 서버 클라이언트 모델에서 클라이언트의 계산량 감소를 위해 설계된 PAK-R, PAK-EC 프로토콜, 상호 인증 과정을 암시적인 방법으로 최적화한 PPK 프로토콜, 서버 타협에 강한 PAK-X 프로토콜 등이 그것이다. 또한, 이를 기반으로 ITU-T SG17 Question 5에서 표준화로 진행 중인 PAK(이후 ITU-PAK)가 있다.[6] ITU-PAK는 MacKenzie의 PAK 프로토콜을 연산 부하량에서 최적화한 프로토콜로서 패스워드를 기반으로 상호 인증 및 키 교환을 수행할 수 있는 프로토콜이며 PKI와 같은 인프라가 필요 없는 공통적인 특징이 있다.

또한, PAK 프로토콜은 Diffie-Hellman 교환을 수행함으로써 perfect forward secrecy를 수립하는 동안 두 참여자가 상호 인증을 하도록 허용한다. 인증은 off-line 사전 공격을 예방하고 도청자로부터 보호하는 사전에 공유한 비밀정보에 의존한다.[6]

표 3은 PAK 프로토콜에서 사용되는 파라미터를 정의한 것이며, 그림 4와 그림 5는 PAK 프로토콜과 PAK-R 프로토콜의 절차를 나타내고, ITU- PAK 프로토콜은 그림 6에서 설명한다.

표 3. PAK 프로토콜 파라미터

- 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 비트 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^*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

User A		User B
$x \in_R Z_q$ $m = g^x \cdot H_1(A, B, \pi)^r$	m $\rightarrow$	
		$m \stackrel{?}{=} 0 \bmod p$ $y \in_R Z_q$, $\mu = g^y$
	$\leftarrow \mu, k$	$\sigma = (\frac{m}{H_1(A, B, \pi)^r}) = g^{xy}$ $k = H_{2a}(A, B, m, \mu, \sigma, \pi)$
$\sigma = \mu^x$ $k \stackrel{?}{=} H_{2a}(A, B, m, \mu, \sigma, \pi)$ $k' = H_{2b}(A, B, m, \mu, \sigma, \pi)$	$k' \rightarrow$	$k' \stackrel{?}{=} H_{2b}(A, B, m, \mu, \sigma, \pi)$
$Key = H_3(A, B, m, \mu, \sigma, \pi)$		$Key = H_3(A, B, m, \mu, \sigma, \pi)$

그림 4. PAK 프로토콜

Alice (A) : Client		Bob(B) : Server
$x \in_R Z_q$, $h \in_R Z_q^*$ $m = g^x \bullet h^q \bullet H_1(A, B, \pi) \bmod p$	$m \rightarrow$	$y \in_R Z_q$ $\mu = g^y \bmod p$ $\sigma = ((\frac{m}{H_1(A, B, \pi)})^r)^{y r^{-1}} = g^{xy} \bmod p$
$\sigma = \mu^x = g^{xy} \bmod p$ $k \stackrel{?}{=} H_{2a}(A, B, m, \mu, \sigma, \pi)$ $k' = H_{2b}(A, B, m, \mu, \sigma, \pi)$	$\leftarrow \mu, k$ $k' \rightarrow$	$k = H_{2a}(A, B, m, \mu, \sigma, \pi)$ $k' \stackrel{?}{=} H_{2b}(A, B, m, \mu, \sigma, \pi)$
$Key = H_3(A, B, m, \mu, \sigma, \pi)$		$Key = H_3(A, B, m, \mu, \sigma, \pi)$

그림 5. PAK-R 프로토콜

Alice (A) : Client		Bob(B) : Server
Select $R_A \in_R Z_q$ $m = H(P) \bullet (g^{R_A} \bmod p)$	$m \rightarrow$	$m \stackrel{?}{=} 0 \bmod p$ $\frac{H(P) \bullet (g^{R_A} \bmod p)}{H(P)} = g^{R_A}$ Select $R_B \in_R Z_q$
$S_1' = H("1" \mid P \mid g^{R_A} \bmod p \mid g^{R_B} \bmod p \mid g^{R_A R_B} \bmod p)$	$\leftarrow m', S_1$	$S_1 = H("1" \mid P \mid g^{R_A} \bmod p \mid g^{R_B} \bmod p \mid g^{R_A R_B} \bmod p)$ $m' = H(P) \bullet (g^{R_A} \bmod p)$
$S_1' \stackrel{?}{=} S_1$ $S_2 = H("2" \mid P \mid g^{R_A} \bmod p \mid g^{R_B} \bmod p \mid g^{R_A R_B} \bmod p)$	$S_2 \rightarrow$	$S_2' = H("2" \mid P \mid g^{R_A} \bmod p \mid g^{R_B} \bmod p \mid g^{R_A R_B} \bmod p)$ $S_2' \stackrel{?}{=} S_2$
$K = H("3" \mid P \mid g^{R_A} \bmod p \mid g^{R_B} \bmod p \mid g^{R_A R_B} \bmod p)$		$K = H("3" \mid P \mid g^{R_A} \bmod p \mid g^{R_B} \bmod p \mid g^{R_A R_B} \bmod p)$

그림 6. ITU-PAK 프로토콜

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[4] 프로토콜은 오프라인 사전공격(off-line dictionary attack)을 이용하여 공격자가 올바른 패스워드를 결정하지 못하도록 하였으며, PFS(Perfact forward Sececy), 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[5] 프로토콜이 제안되었다. 기존의 PAK에서는 m 을 계산하기 위해 q 비트의 지수승 1회, r 비트의 지수승 1회, σ를 계산하기 위한 q 비트의 지수승 1회 등 총 3회의 연산을 하지만, PAK-R 프로토콜에서 m 을 계산하기 위해 q 비트의 지수승 2번, σ 를 계산하기 위한 q 비트의 지수승 1회 총 3회의 연산을 계산한다(여기서, q : 160비트, r : 760비트). 그리고 PAK-R 프로토콜 또한 DDH에 대한 가정을 통해 안전성이 증명되었다. 그러나 ITU-T SG17 Q.5에서 표준화 하고 있는 PAK 프로토콜은 Philip MacKenzie의 보안 기능을 모두 만족하면서 서버와 클라이언트측의 연산에서 H_1 해쉬 함수를 일반 해쉬를 사용함으로써 지수승 연산을 각각 1회 이상 감소시키는 효과가 있다.

Ⅳ. Signcryption을 이용한 PAK 기반의 키 분배 프로토콜 제안

1. 제안 프로토콜

본 절에서는 Signcryption의 서명기법과 암호기법을 이용하여 PAK 프로토콜 기반의 키 분배 프로토콜을 제안하고자 한다. 제안한 키 분배 프로토콜은 PAK를 기반으로 클라이언트 측면에서 계산량을 감소시키기 위해 [4]에서 제안한 PAK-R 프로토콜을 변형 적용 하였다. 이때 PAK-R 프로토콜에서 H_1 함수(1024bit + 160비트)를 연산하기 위해 사용한 h^q를 제거하기 위해 ITU-U PAK의 H_1 함수를 출력 160비트의 일반 해쉬 함수를 이용하여 클라이언트에서 계산량 감소를 위해 제안된 PAK-R프로토콜보다 클라이언트에서의 연산량을 월등히 감소시켰다. 또한, PAK 프로토콜에서의 전제인 참여자간 공유하

는 패스워드는 모바일 환경에 적합한 전제가 되지 못한다. 이에 참여자 패스워드를 대체 할 수 있는 세션 임시 비밀키를 이용하여 상호 인증을 할 수 있도록 설계하였다.

다음 표 4는 제안한 프로토콜에서 사용되는 파라미터를 정의한 것이고, 그림 7은 제안한 Signcryption을 이용한 패스워드 기반의 키 분배 프로토콜이다.

표 4. Signcryption의 공개 및 비밀 파라미터

Alice의 키	공개	Bob의 키
A:사용자 A의 ID x_a:개인키 y_a:공개키 $(y_a = g^{x_a} \bmod p)$ K_1, K_2 : 대칭키	p:큰 소수 q:큰 소수 (factor of $p-1$) g:$0<g<p$이고 차수 q mod p H_1, H_2, H_3: 일방향 해쉬 KH : Key-ed 1-way 해쉬	B:사용자 B의 ID x_b:개인키 y_b:공개키 $(y_b = g^{x_b} \bmod p)$ K_1, K_2 : 대칭키

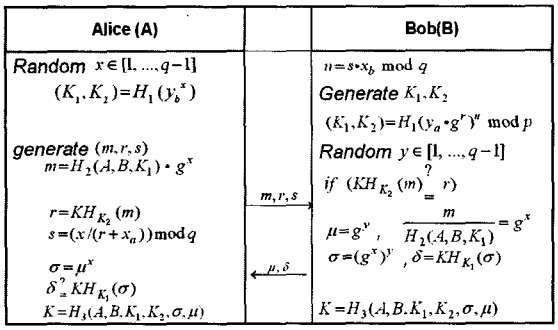

그림 7. Signcryption을 이용한 패스워드 기반의 키 분배 프로토콜

Alice에 의한 Signcryption은 다음 절차를 따라 수행된다.

(1) 랜덤 정수 $x \in \{1, .., q-1\}$를 선택하고, 다음 식에 의해 대칭키 K_1, K_2를 계산한다.

$$(K_1, K_2) = H_1(y_b{}^x)$$

(2) 메시지 m과 r, s을 생성한다.

$$m = H_2(A, B, K_1) \cdot g^x$$
$$r = KH_{K_2}(m)$$
$$s = (x/(r + x_a)) \bmod q$$

(3) Bob에게 생성된 (m, r, s)를 전송한다.

(m, r, s)를 수신한 Bob은 다음과 같이 Unsigncryption 절차를 수행한다.

(1) 수신된 s를 이용하여 세션정보 u를 다음과 같이 계산한다.

$$u = (s \cdot x_b) \bmod q$$

(2) 세션 정보를 이용하여 세션키 K_1, K_2를 다음과 같이 계산한다.

$$(K_1, K_2) = H_1(y_a \cdot g^r)^u \bmod p$$

만일, $r \stackrel{?}{=} KH_{K_2}(m)$ 이 아니면, 세션이 종료되고, 조건에 만족하면 Bob은 Alice에게 인증 받기 위한 과정을 수행한다.

(1) Bob은 자신의 비밀정보를 $\mu = g^y$와 같이 계산하고, 수신된 m으로부터 Alice의 비밀정보를 계산하여 다음 수식에 의하여 (μ, σ)를 계산한다. 이때 y는 랜덤 정수 $y \in \{1, .., q-1\}$이다.

$$\mu = g^y$$
$$\frac{m}{H_2(A, B, K_1)} = g^x$$
$$\sigma = (g^x)^y$$
$$\delta = KH_{K_1}(\sigma)$$

(2) Alice에게 (μ, δ)를 전송한다. Bob은 자신의 패스워드 정보와 Alice의 비밀 정보를 이용하여 생성한 δ를 이용하여 세션 키 K를 생성한다.

Alice는 Bob으로부터 수신한 (μ, σ)를 이용하여 자신의 비밀정보를 이용한 Bob의 메시지인 것을 확인 후, 대칭키를 다음과 같이 생성한다.

(1) $\sigma = \mu^x$

(2) 만일 수신된 δ와 Alice 자신이 다음 계산 값이 일치하면 Bob을 인증하고 세션키를 생성한다.

$$\delta \stackrel{?}{=} KH_{K_1}(\sigma)$$

(3) 마지막으로 Alice와 Bob은 다음 식을 수행하여 세션키를 생성한다.

$$K = H_3(A, B, K_1, K_2, \sigma, \mu)$$

V. 안전성 및 성능 분석

본 논문에서 제안한 프로토콜은 다음과 같은 보안 요구사항을 만족한다.

(1) 수동적 도청 공격(Passive Eavesdropping)

- 공격자는 두 참여자 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 불가능 하다. 이 공격은 DLP의 어려움을 근거로 해결 가능하다. 본 제안 프로토콜에서 공격자가 공개정보 p, q, g, y_a, y_b를 알고, Alice에서 Bob으로 전송되는 m, r, s를 알고 있다 하더라도, 공격자가 g^x, g^y를 구하는 것은 DLP를 풀어내는 것만큼 어렵다. 또한 통신 주체간에 인증을 위한 파라미터와 해쉬값은 전송되지 않고 통신 주체가 직접 계산하므로 도청 공격을 통해서는 알 수 없다. 그리고 r, s로부터 x_a를 구하는 것은 Signcryption 안전성에 의존한다.

(2) 재전송공격(Replay attack)

- 재생 공격은 공격자가 Alice의 메시지 m을 Bob에게 재전송하여 이미 정상적인 Alice에 의해 생성된 이전키(old session key)를 다시 생성하기 위한 공격이다. 그러나 모든 통신 메시지들은 매 세션마다 균일한 확률 분포에서 랜덤하게 생성되어짐을 가정하기 때문에 이 공격에 대한 공격자의 성공 확률은 무시할 만하다.

(3) 중간자 공격(Man-in-the-middle attack)

- 공격자가 두 참여자 사이에서 합법적으로 가장하거나 전송되는 메시지를 개로 챈 다음, 공격자와 Alice, 공격자와 Bob 사이에서 각각의 별도의 세션값을 만들어 내는 공격이다. 그러나 본 제안 프로토콜은 참여자의 비밀 정보를 유한 필드상에서의 D-H 문제와 이산대수 문제의 어려움에 근거하여 설계하였기 때문에, 공격자는 프로토콜내의 모든 대화 내용을 이용하더라도 참여자의 비밀 정보를 알아 낼 수 없다.

(4) 제안된 프로토콜은 오프라인(off-line) 사전 추측 공격에 대한 저항성을 갖는다.

- 제안된 프로토콜에서 공격자는 전송되는 메시지 m, r, s, μ 그리고 δ를 얻어도 DHP를 해결하지 못하므로 정확한 임의의 난수 x, y를 알 수 없다. 또한 여기서 전송되는 m, δ, K는 해쉬를 이용하여 인증되기 때문에 정확한 해쉬값을 모르면 이를 해결하지 못한다.

(5) PFS(Perfect Forward Secrecy)의 만족

- PFS는 통신 주체들간에 장기간 개인키가 노출되더라도, 공격자가 통신 주체들간의 과거 세션키를 계산할 수 없는 경우를 PFS를 만족한다고 정의한다. 본 제안 프로토콜은 각 세션마다 두 참여자의 비밀 정보와 난수를 이용하여 세션마다 세션키를 갱신하기 때문에 현재의 세션키와 이전의 세션키를 알 수 없다.

(6) 키의 신성성과 위조 불가능성

- 세션마다 사용자는 새로운 키 요소를 난수에 의하여 생성하고 이를 기반으로 세션키를 생성하므로 키의 신성성이 보장되고 이를 위조 하는 것은 불가능하다. 그리고 참여자의 개인키 정보를 이용하여 상호 인증을 수행하고 이를 기반으로 세션키를 생성한다.

표 5와 6은 Signcryption 프로토콜과 인증된 패스워드 키 분배 프로토콜, 그리고 제안한 Signcryption을 이용한 패스워드 키 분배 프로토콜의 계산적 비용과 기능면에서 성능을 비교 분석한다.

표 5. 계산량의 정량적 비교

기법 \ 연산량		계산적 비용						
		멱승	해쉬	암호	복호	곱셈	나눗셈	난수
PAK & ITU-U PAK	Alice	3	4	-	-	3		2
	Bob	3	4	-	-	2	1	2
DKEUN	Alice	3	5	1	1	2	3	2
	Bob	3	5	1	1	2	3	1
DKEUTS	Alice	3	5	1	-	2	3	1
	Bob	3	5	-	1	2	3	1
제안 프로토콜	Alice	3	5	-	-	1	1	1
	Bob	3	5	-	-	1	1	1

표 6. 기능적 비교

기법	상호 인증	통신량	키분배	Alice 키 요소	Bob 키 요소
PAK & ITU-U PAK	○	3-way	○	○	○
DKEUN	○	3-way (+ 1 opt)	○	○	×
DKEUTS	○	2-way (+ 1 opt)	○	○	×
제안 프로토콜	○	2-way	○	○	○

표 5와 표 6에서 보는 바와 같이 Signcryption, PAK, [3]에서 Zheng이 제안한 키 분배 프로토콜 그리고 제안한 Signcryption을 이용한 패스워드 기반의 키 분배 프로토콜의 계산 부하량과 기능적 관점에서 비교 하였다. 제안한 프로토콜은 PAK 프로토콜에 비해 멱승의 연산량은 동일하나 해쉬 연산량에서 참여자 각각 1번씩의 연산이 증가된 것을 볼 수 있다. 그러나 암·복호, 곱셈과 나눗셈 그리고 난수생성에 있어서 연산량이 현저히 줄어든 것을 볼 수 있다. 통신량 또한 기본 3-way의 PAK 프로토콜에 비해 2-way의 통신으로 상호 인증이 가능하다. Zheng의 키 분배 프로토콜과 비교하면 멱승과 곱셈 연산에서는 동일한 연산량을 유지하면서 암·복호, 곱셈과 나눗셈 그리고 난수생성에 있어서 연산량이 현저히 줄어든 것을 볼 수 있다. 통신량 면에서 Zheng의 Signcryption을 이용한 키 분배 프로토콜은 상호 인증을 위해 DKEUN는 4-way 통신량과 DKEUTS는 3-way 통신량을 보이는 반면, 제안한 프로토콜은 2-way로써 상호 인증과 키 분배까지 가능하다.

Ⅵ. 결 론

본 논문에서는 Signcryption을 이용하여 패스워드 기반의 인증 키 분배 프로토콜을 제안하였다. 기존의 패스워드 키 분배 프로토콜보다 연산량과 통신량을 감소시키고, Signcryption을 이용하여 통신 당사자간의 인증과 패스워드 요소의 비밀성을 유지하였다. 향후 사용자 인증이 가능하고 사용자 패스워드에 대한 비밀성을 유지하면서 패스워드 기반의 세션키를 세션마다 갱신하므로 도메인에서(하나 또는 다중 도메인에서) 하나의 패스워드로 안전하게 통신을 할 수 있는 "다중 도메인에서 단일 패스워드를 이용한 인증 및 키 분배 프로토콜"로 연구가 가능할 것으로 생각한다.

참 고 문 헌

[1] Y. Zheng, "Digital signcryption or how to achieve cost (signature and encryption) << cost(signature) + cost(encryption)", Advances in Cryptology, Proceedings of CRYPTO'97, LNCS Vol. 1294, Springer-Verlag, pp. 165-179, 1997

[2] Y. Zheng, "Updates on Signcryption" IEEE P1363, UCSB, 2002, 8

[3] Y. Zheng, "Shortened Digital Signature, Signcryption and Compact and Unforgeable Key Agreement Schemes" IEEE P1363a:Standard Specifications for Public-key Cryptography : Additional Techiques, 1998

[4] V. Boyko and S. Patel, "Provably Secure Password Authentication and key Exchange Using Diffie-Hellman" EuroCryp 2000, pp.156-171, 2000.

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

[6] Alec Brusilovsky, Igor Faynberg, Sarvar Patel, Zachary Zeltsan, "Password-Authenticated Key Exchange(PAK) Protocol", STUDY GROUP 17, COM 17-D121-E, Jan, 16, 2006

[7] 김락현, 염흥열 "MTI 기반의 새로운 PAK 프로토콜 제안", 한국정보보호학회 하계정보보호학술대회 논문집, Vol. 13 No1. pp.126-132, 2003

[8] Rack-Hyun Kim, Ho-Sun Yoon, and Heung-Youl Youm, "New Password Authenticated Key Exchange Protocols for Mobile Network," The 8th International Conference on Cellular and Intelligent Communications (CIC2003), pp.451-466, 2003.10, Seoul Korea.

〈著者紹介〉

김 락 현 (Rack-Hyun Kim) 학생회원
1997년 2월 : 순천향대학교 전자공학과 졸업
1999년 8월 : 순천향대학교 일반대학원 전기·전자공학과 석사 졸업
2001년 2월~현재 : 순천향대학교 일반대학원 정보보호학과 박사과정
〈관심분야〉 암호 이론, 공개키 기반구조, 네트워크 보안, 보안 프로토콜, 이동통신보안

염 흥 열 (Heung-Youl Youm) 종신회원
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~2006년 2월 학교 산학연컨소시엄센터 소장
1997년 3월~현재 : 한국정보보호학회 총무이사, 학술이사, 교육이사, 현 총무이사
2004년 1월~현재 : 한국인터넷정보학회 이사, 논문지 편집위원
2004년 1월~현재 : OSIA 이사
2003년 9월~2004년 3월 : ITU-T SG17/Q10, Associate Rapporteur
2004년 3월~현재 : ITU-T SG17/Q9 Rapporteur
〈관심분야〉 네트워크보안, 전자상거래보안, 공개키 기반구조, 부호이론, 이동통신보안