Advanced Modification 공격에 안전한 패스워드 기반 키 동의 프로토콜

곽 진[†] · 오 수 현[††] · 양 형 규[†††] · 원 동 호[††††]

요 약

사용자의 인증기술로 패스워드를 기반으로 하는 메커니즘이 널리 사용되고 있다. 패스워드를 기반으로 하는 메커니즘은 사용자들이 기억하기 쉬운 패스워드를 선택하여 사용하는 경우가 대부분이므로, 패스워드 추측 공격(password guessing attack)에 취약하다는 문제점이 있다. 이러한 패스워드 추측 공격을 방지하기 위해 많은 키 분배 프로토콜이 제안되고 있으며, 최근 Seo-Sweeny는 패스워드를 기반으로 하는 인증 키 동의(SAKA : Simple Authenticated Key Agreement) 프로토콜을 제안하였다. 본 논문에서는 먼저, 패스워드를 기반으로 하는 SAKA 프로토콜과 이를 개선한 방식들의 키 설정 및 키 확인 과정을 살펴보고, 각각의 프로토콜이 본 논문에서 정의한 Advanced Modification 공격에 대해 취약함을 보인다. 그리고 Advanced Modification 공격에 대해 안전한 패스워드 기반 인증 키 동의 프로토콜을 제안한다.

Password-based Authenticated Key Agreement Protocol Secure Against Advanced Modification Attack

Jin Kwak[†] · Soo Hyun Oh[††] · Hyung Kyu Yang[†††] · Dong Ho Won[††††]

ABSTRACT

Password-based mechanism is widely used methods for user authentication. Password-based mechanisms are using memorable passwords(weak secrets), therefore password-based mechanisms are vulnerable to the password guessing attack. To overcome this problem, many password-based authenticated key exchange protocols have been proposed to resist password guessing attacks. Recently, Seo-Sweeny proposed password-based Simple Authenticated Key Agreement(SAKA) protocol. In this paper, first, we will examine the SAKA and authenticated key agreement protocols, and then we will show that the proposed simple authenticated key agreement protocols are still insecure against Advanced Modification Attack. And we propose a Password-based Simple Authenticated Key Agreement Protocol secure against Advanced Modification Attack.

키워드 : 패스워드 기반(Password-Based), Simple Authentication, 키 동의 프로토콜(Key Agreement Protocol), 키 확인(Key Confirmation), Advanced Modification 공격(Advanced Modification Attack)

1. 서 론

1976년 Diffie-Hellman[1]에 의해 공개키 암호 시스템의 개념이 발표된 이후, RSA[2], ElGamal[3] 등과 같은 여러 종류의 공개키 암호 시스템이 제안되었다. Diffie-Hellman 키 동의 프로토콜은 인증된 두 개체간에 안전하지 않은 채널을 사용하여 세션키(session key)를 분배하는 방법으로 널리 알려져 있다. 이 프로토콜의 안전성은 유한체 상에서의 이산대수 문제(DLP : Discrete Logarithm Problem)의 어려움에 의존한다. 하지만 Diffie-Hellman 키 동의 프로토콜의 가장 큰 문제점은 통신하는 두 개체 사이에서 공격자가 임의의 메시지를 삽입하거나 정당한 사용자로 위장하는 man-in-the-middle 공격에 취약하다는 것이다.

이러한 man-in-the-middle 공격을 방지하기 위한 방법으로는 키 교환 프로토콜에서 인증서(e.g. 전자서명)를 사용하는 방식과 패스워드를 기반으로 하는 방식 등이 있다.

먼저 인증서를 사용하는 방식은 공개키(public key)의 소유자를 확인하기 위해 신뢰할 수 있는 인증기관(CA : Certification Authority)으로부터 인증서(certificate)를 발급 받아 사용한다. 인증기관이 발급한 인증서는 통신하려는 두 개체의 공개키와 소유자에 대한 정보를 포함하고 있으므로 공격자가 두 개체 사이에서 상대방으로 위장하거나 다른 공개키를 삽입하는 등의 공격이 불가능하다. 하지만 인증서

※ This work was supported by the University IT Research Center Project funded by the Korean Ministry of Information and Communication.
† 준 회 원 : 성균관대학교 대학원 정보통신공학부
†† 정 회 원 : 호서대학교 컴퓨터공학부 교수
††† 정 회 원 : 강남대학교 컴퓨터미디어공학부 교수
†††† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2004년 1월 30일, 심사완료 : 2004년 5월 14일

를 사용하는 방식의 경우 사용자의 증가와 함께 인증서에 포함된 사용자들의 전자서명을 검증하는데 추가적인 계산이 필요하고, 인증서들을 저장하기 위해 큰 저장 시스템을 필요로 하게된다. 또한 인증기관이 손상을 입었을 경우 전체 시스템이 영향을 받게 되는 문제점을 가지고 있다.

패스워드를 기반으로 하는 방식은 사용자들이 선택한 패스워드를 사용하여 통신하려는 상대방의 신분을 확인하는 방식이다. 패스워드를 기반으로 하는 방법은 다른 기억장치가 필요하지 않고 사용자들이 자신의 패스워드를 쉽게 선택하고 기억할 수 있다는 장점을 가지고 있다. 하지만, 사용자들이 한정된 공간에서 기억하기 쉬운 패스워드를 선택하기 때문에 공격자에게 패스워드 추측 공격을 허용할 수 있다는 약점이 존재한다. 이러한 공격을 방지하기 위해 다양한 패스워드 기반의 인증 키 교환 프로토콜들이 제안되었다[4-8].

패스워드 기반 인증 키 교환 프로토콜은 통신하려는 두 개체가 오프라인을 통해 프로토콜 수행 이전에 미리 패스워드를 공유한다고 가정한다. 통신하고자 하는 두 개체는 미리 공유한 패스워드를 사용하여 공통의 세션키를 설정하고 세션키에 대한 키 확인 과정을 수행한다. 최근 Seo-Sweeny는 위와 같은 방식을 기반으로 간단하게 세션키를 공유하고 확인할 수 있는 SAKA(Simple Authenticated Key Agreement) 프로토콜을 제안하였다[9].

본 논문에서는 Advanced Modification 공격을 정의하고 SAKA 프로토콜이 본 공격에 취약함을 보인다. 그리고 본 논문에서 정의한 공격에 안전한 개선된 패스워드 기반 인증 키 동의 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 SAKA 프로토콜에 대하여 설명하고, 3장에서는 Advanced Modification 공격에 대한 정의와 이에 대한 SAKA 프로토콜의 취약점을 분석한다. 다음으로 4장에서는 Advanced Modification 공격에 안전한 패스워드를 이용하는 개선된 인증 키 동의 프로토콜을 제안하고, 5장에서는 제안한 프로토콜의 안전성과 효율성에 대하여 분석한다. 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

최근 Seo-Sweeny는 Diffie-Hellman 방식을 기반으로 패스워드를 사용하여 두 개체들의 인증과 비밀 세션키를 공유하는 SAKA 프로토콜을 제안하였다[9]. 그러나 SAKA 프로토콜은 Tseng[10]에 의해 재전송 공격에 대한 취약점이 발견된 이후, Ku-Wang[11], Lin[12] 등에 의해 패스워드 추측 공격과 modification 공격에 대한 취약점이 발견되었으며, 이를 개선한 프로토콜들이 각각 발표되었다. 본 절에서는 각각의 프로토콜들을 살펴보고 앞에서 언급한 취약점들에 대하여 설명한다.

SAKA 프로토콜과 본 논문에서 제안하는 프로토콜에서 사용하는 시스템 파라미터들은 다음과 같다.

[시스템 파라미터]

- Alice, Bob : 통신 개체
- Eve : 능동적 공격자 (active attacker)
- S : 공통의 비밀 패스워드
- p : 큰 소수
- g : Z_p 상의 원시원소 ($ord(g) = p-1$)
- a : Alice가 선택한 랜덤 수, $a \in_R Z_p^*$
- b : Bob이 선택한 랜덤 수, $b \in_R Z_p^*$
- $SK_{\{A,B\}}$: Alice와 Bob의 세션키
- H : 일방향 해쉬 함수 (one-way hash function)

2.1 Seo-Sweeny의 SAKA 프로토콜

SAKA 프로토콜은 두 개체가 통신을 시작하기 이전에 공통의 비밀 패스워드를 공유하고 있다는 것을 가정한다. SAKA 프로토콜의 키 설정 과정 그리고 키 확인 과정은 다음과 같다((그림 1) 참조).

Alice	공개 정보 p, g	Bob
[키 설정 과정] $a \in_R Z_P^*$ $X_1 = g^{aQ} \bmod p$ 계산	X_1 →	
	← Y_1	$b \in_R Z_p^*$ $Y_1 = g^{bQ} \bmod p$ 계산
$Y = (Y_1)^{Q^{-1}} = g^b \bmod p$ $SK_A = (Y)^a = g^{ab} \bmod p$		$X = (X_1)^{Q^{-1}} = g^a \bmod p$ $SK_B = (X)^a = g^{ab} \bmod p$
[키 확인 과정]	$(SK_A)^Q = g^{abQ} \bmod p$ →	
	← $(SK_B)^Q = g^{abQ} \bmod p$	
$((SK_B)^Q)^{Q^{-1}} \stackrel{?}{=} SK_A$		$((SK_A)^Q)^{Q^{-1}} \stackrel{?}{=} SK_B$

(그림 1) SAKA 프로토콜

[키 설정 과정]

① Alice와 Bob은 프로토콜을 시작하기 전에 각각 두 정수 $Q \bmod p-1$와 $Q^{-1} \bmod p-1$를 계산한다. 여기서 Q는 패스워드 S로부터 유도된 값으로, $(p-1)$과 서로 소인 값이어야 한다. 또한, 서로 다른 패스워드가 주어졌을 경우 동일한 Q 값이 나올 확률이 매우 낮아야 한다.

② Alice는 $a \in {}_R Z_p^*$를 선택하여 다음을 계산한 후 Bob에게 전송한다.

$$X_1 = g^{aQ} \bmod p$$

③ Bob은 랜덤 수 $b \in {}_R Z_p^*$를 선택하여 다음을 계산한 후 Alice에게 전송한다.

$$Y_1 = g^{bQ} \bmod p$$

④ Alice는 Bob으로부터 수신한 Y_1을 이용하여 다음을 계산한다.

$$Y = (Y_1)^{Q^{-1}} = g^b \bmod p\ ,\ SK_A = (Y)^a = g^{ab} \bmod p$$

⑤ Bob은 Alice로부터 수신한 X_1을 이용하여 다음을 계산한다.

$$X = (X_1)^{Q^{-1}} = g^a \bmod p\ ,\ SK_B = (X)^b = g^{ab} \bmod p$$

[키 확인 과정]

⑥ Alice는 다음을 계산하여 Bob에게 전송한다.

$$(SK_A)^Q = g^{abQ} \bmod p$$

⑦ Bob은 다음을 계산하여 Alice에게 전송한다.

$$(SK_B)^Q = g^{abQ} \bmod p$$

⑧ Alice와 Bob은 각각 수신한 값과 자신이 계산한 값을 비교하여 키 확인 과정을 수행한다.

$$\text{Alice}: ((SK_B)^Q)^{Q^{-1}} \stackrel{?}{=} SK_A$$

$$\text{Bob}: ((SK_A)^Q)^{Q^{-1}} \stackrel{?}{=} SK_B$$

Seo-Sweeny는 공격자 Eve가 키 설정 과정 ②에서 X_1을 수신하더라도, 이산대수 문제가 계산 불가능하고 비밀 패스워드 S를 알 수 없기 때문에 $g^a \bmod p$와 Q값을 추측할 수 없으므로 man-in-the-middle 공격에 대해 안전하다고 설명하였다.

그러나 SAKA 프로토콜은 사용자의 패스워드 Q가 노출된 경우, 과거의 세션키 SK_A가 $((SK_A)^Q)^{Q^{-1}}$와 같은 간단한 계산을 통해 쉽게 계산이 가능하므로 Perfect Forward Secrecy를 제공하지 못한다. 또한, 사용자가 패스워드를 선택할 때, 충분하지 못한 패스워드(poorly chosen) Q를 선택할 경우, 공격자는 $((SK_A)^Q)Q = (X_1)^b \bmod p$와 같은 계산을 통해 Q를 구하는 것이 가능하다. 그러므로 SAKA 프로토콜은 사전공격에 안전하지 못하다.

[정의 1]

Perfect Forward Secrecy(PFS) : 사용자의 패스워드와 같은 비밀정보가 노출되더라도, 공격자가 두 사용자 사이에 설정된 과거의 세션키를 계산할 수 없는 경우에 PFS를 만족한다고 한다.

[정의 2]

사전공격(Dictionary Attack) : 공격자가 키 교환 과정 동안 수행한 개체들 사이의 전송정보를 이용하여 패스워드 또는 세션키를 구하려는 공격

2.2 Tseng의 Improved SAKA 프로토콜

Tseng은 SAKA 프로토콜에서 공격자 Eve가 재전송 공격에 의해 Bob으로 위장하는 것이 가능하다는 것을 보였으며, 이러한 취약점을 보완한 Improved SAKA 프로토콜을 제안하였다[10]. 공격자 Eve가 키 확인 과정에서 Alice가 계산하여 Bob에게 전송하는 $(SK_A)^Q$ 값을 Bob으로 위장하여 Alice에게 재전송 하는 경우 Bob으로 위장하는 것이 가능하다((그림 2) 참조).

그러므로, Alice가 잘못된 세션키를 계산하고 공격자 Eve가 Alice와 같은 세션키를 계산하지 못한다 하더라도, Eve는 Alice가 올바른 세션키를 가지고 있다고 믿게 할 수 있다. 이것은 SAKA 프로토콜이 올바른 키 확인 과정을 수행하지 못하고 있음을 보여준다.

[Tseng의 키 확인 과정]

Tseng은 이러한 취약점을 보완하기 위해서, 키 확인 과정을 다음과 같이 개선한 프로토콜을 제안하였다. Tseng의 키 설정과정은 SAKA 프로토콜과 동일하다((그림 3) 참조).

① Alice는 Bob으로부터 수신한 Y_1을 이용하여 다음과 같이 Y를 계산하여 Bob에게 전송한다.

$$Y = (Y_1)^{Q^{-1}} = g^b \bmod p$$

② Bob은 Alice로부터 수신한 X_1을 이용하여 다음과 같이 X를 계산하여 Alice에게 전송한다.

$$X = (X_1)^{Q^{-1}} = g^a \bmod p$$

③ Alice와 Bob은 각각 수신한 값과 자신이 계산한 값을 비교하여 올바른 값인지 확인한다. 올바른 값이면 세션키를 신뢰한다.

Alice		Eve
[키 확인 과정] $(SK_A)^Q = g^{abQ} \bmod p$ 계산	$(SK_A)^Q = g^{abQ} \bmod p$ →	
	← $(SK_A)^Q = g^{abQ} \bmod p$	$(SK_A)^Q = g^{abQ} \bmod p$ 수신 후 재전송
$((SK_A)^Q)^{Q^{-1}} \stackrel{?}{=} SK_A$		

(그림 2) SAKA 프로토콜의 키 확인 과정에 대한 재전송 공격

Alice		Bob
[키 확인 과정] $Y = (Y_1)^{Q^{-1}} = g^b \bmod p$	Y →	$X = (X_1)^{Q^{-1}} = g^a \bmod p$
$X \stackrel{?}{=} g^a \bmod p$	← X	$Y \stackrel{?}{=} g^b \bmod p$

(그림 3) Improved SAKA 프로토콜의 키 확인 과정

Tseng의 키 확인 과정에서 공격자 Eve가 Alice를 속이기 위해서는 Alice로부터 X_1값을 수신한 후 X값을 반드시 계산하여 Alice에게 전송해야 한다. 하지만, 공격자 Eve가 $g^a \bmod p$값과 Q값을 얻는 것은 이산대수 문제를 푸는 어려움과 비밀 패스워드로 인해 불가능하므로, Eve는 X_1값으로부터 정확한 X를 얻을 수 없다.

2.3 Ku-Wang의 Enhanced SAKA 프로토콜

Ku-Wang은 Tseng의 Improved SAKA 프로토콜이 Modification 공격에 안전하지 않음을 보이고, 이러한 취약점을 개선하여 Enhanced SAKA 프로토콜을 제안하였다[11].

키 설정 과정에서, 공격자 Eve는 Alice로부터 X_1이 전송될 때 X_1'으로 대체하여 Bob에게 전송하고, Bob은 Y_1을 Alice에게 전송한다. 키 확인과정에서, Alice는 수신한 Y_1값으로 Y를 계산하여 Bob에게 전송하고, Bob은 공격자 Eve가 전송한 X_1'값으로 계산한 $X'(=(X_1')^{Q^{-1}} \bmod p)$을 Alice에게 전송한다. 여기서 $X' \neq g^a \bmod p$이므로, Alice는 키 설정 과정을 통해 계산한 세션키를 신뢰하지 않지만, Bob은 $Y = g^b \bmod p$이므로 자신이 계산한 세션키를 신뢰하게 된다. 비록 공격자 Eve가 Bob의 세션키를 계산할 수는 없지만 Bob에게 잘못된 세션키를 올바른 것으로 속일 수 있게 된다((그림 4) 참조).

[Ku-Wang의 키 확인 과정]

Tseng의 Improved SAKA 프로토콜은 키 확인 과정에서 Bob이 올바른 세션키가 생성되었는지 판단할 수 없다는 문제점을 가지고 있다. 그러므로 Ku-Wang은 Tseng의 키 확인 과정을 다음과 같이 개선하였다((그림 5) 참조).

① Tseng의 키 확인 과정에서 Alice는 Y_2를 계산하여 Bob에게 전송한다.

$$Y_2 = (SK_A)^Q \bmod p = g^{abQ} \bmod p$$

② Bob은 Alice로부터 수신한 Y_2값을 이용하여 다음을 계산한다.

$$(Y_2)^{Q^{-1}} \bmod p \stackrel{?}{=} SK_B$$

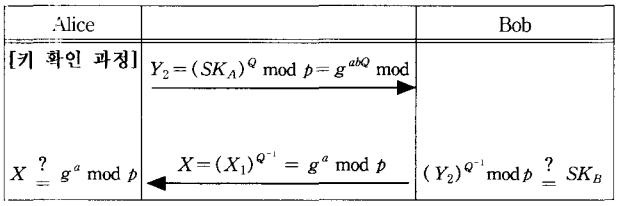
계산한 값이 올바른 값이면, 자신이 Alice로부터 수신한 X_1값과 자신이 전송한 Y_1값을 Alice가 정확하게 수신한 것으로 신뢰한다. Bob은 자신이 계산한 세션키가 유효한 것으로 신뢰하고 X값을 Alice에서 전송한다.

Alice		Eve		Bob
[키 설정 과정] $a \in Z_p^*$ $X_1 = g^{aQ} \bmod p$	X_1 →	X_1'	X_1' →	
	← Y_1		← Y_1	$b \in Z_p^*$ $Y_1 = g^{bQ} \bmod p$
$Y = (Y_1)^{Q^{-1}} = g^b \bmod p$ $SK_A = (Y)^a = g^{ab} \bmod p$				$X = (X_1')^{Q^{-1}} \bmod p$ $SK_B = (X')^b \bmod p$
[키 확인 과정] $Y = (Y_1)^{Q^{-1}} = g^b \bmod p$	Y →		Y →	
$X' \neq g^a \bmod p$	← X		← X	$Y \stackrel{?}{=} g^b \bmod p$

(그림 4) Improved SAKA 프로토콜에 대한 Modification 공격

③ Alice는 Bob으로부터 수신한 X값을 이용하여 다음을 계산하고, 정확한 값이면 자신이 수신한 Y_1값과 Bob이 X_1값을 올바르게 수신한 것으로 신뢰한다. 즉, Alice는 자신의 세션키가 유효한 것으로 신뢰한다.

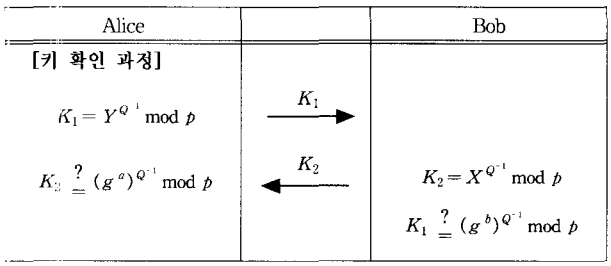
$$X \stackrel{?}{=} g^a \bmod p$$



(그림 5) Ku-Wang의 Enhanced SAKA 프로토콜의 키 확인 과정

2.4 Lin 등의 Enhanced SAKA 프로토콜

Lin 등은 SAKA 프로토콜이 사용자의 신원확인 기능과 perfect forward secrecy 기능을 제공하지 못하며 사전공격에 취약하다는 것을 지적하고, 이를 개선한 프로토콜을 제안하였다[12]. Lin 등이 제안한 Enhanced 프로토콜의 키 설정 과정은 SAKA 프로토콜과 동일하다((그림 6) 참조).



(그림 6) Lin 등이 제안한 Enhanced SAKA 프로토콜의 키 확인 과정

[Lin의 키 확인 과정]

① 키 설정 과정이 끝난 후, Alice는 다음을 계산하여 Bob에게 전송한다.

$$K_1=Y^{Q^{-1}} \bmod p$$

② Bob은 다음을 계산하여 Alice에게 전송한다.

$$K_2=X^{Q^{-1}} \bmod p$$

③ Alice와 Bob은 각각 다음을 계산하여 키 확인과정을 수행한다. 만약 올바른 값이면, 각각 자신들이 계산한 세션키가 올바른 것으로 신뢰한다.

$$\text{Alice}: K_2 \stackrel{?}{=} (g^a)^{Q^{-1}} \bmod p$$
$$\text{Bob}: K_1 \stackrel{?}{=} (g^b)^{Q^{-1}} \bmod p$$

Lin 등이 제안한 프로토콜은 공격자가 K_1과 K_2를 생성할 수 없기 때문에 정당한 사용자로 위장하려는 공격을 막을 수 있으며, $K_1 \neq K_2$이고 랜덤 수 a, b, Q, Q^{-1}이 비밀로 유지되기 때문에 사전공격을 막을 수 있다. 만약 Q가 노출되더라도 이산대수 문제의 어려움에 의해 랜덤 수 a, b를 구하는 것이 불가능하고, 공격자 Eve는 이전의 세션키를 계산할 수 없으므로 perfect forward secrecy를 제공한다.

3. Advanced Modification 공격

SAKA 프로토콜은 위에서 살펴본 바와 같이, Tseng, Ku-Wang, 그리고 Lin 등에 의해 취약점을 개선하며 발전하였다. 본 절에서는 위에서 설명한 Ku-Wang과 Lin 등이 제안한 프로토콜이 본 논문에서 정의한 Advanced Modification 공격에 안전하지 못함을 증명하고, 제4장에서 본 공격에 대해 안전한 프로토콜을 제안한다.

본 논문에서 제안하는 Advanced Modification 공격에서는 키 설정 과정과 키 확인 과정에서 Alice와 Bob을 속이기 위해 능동적 공격을 수행하는 능동적 공격자(active attacker) Eve가 자신이 선택한 랜덤 수 u를 이용하여, Alice와 Bob 사이의 통신에서 메시지의 삽입 및 변조가 가능하다고 가정한다((그림 7) 참조).

[키 설정 과정에 대한 Advanced Modification 공격]

키 설정 과정에 대한 Advanced Modification 공격 과정에서 Eve는 능동적 공격자를 의미하고, u는 $\gcd(p-1, u)=1$인 랜덤 수이다.

[정의 3]

능동적 공격자(Active Attacker) : 단순히 참가자들의 통신 내용을 도청하는 것뿐만 아니라 전송되는 메시지를 위·변조하거나 새로운 메시지를 삽입하는 등의 실제 통신에 참여하는 공격자

① Eve는 Alice가 전송한 X_1값을 다음과 같이 X_1'으로 대체하여 Bob에게 X_1인 것처럼 전송한다.

$$X_1'=(X_1)^u \bmod p = g^{auQ} \bmod p$$

② Eve는 Bob이 전송한 Y_1값을 Alice에게 전송한다.

$$Y_1=g^{bQ} \bmod p$$

③ Alice는 Y값과 세션키 SK_A를 다음과 같이 계산한다.

$$Y=(Y_1)^{Q^{-1}}=g^b \bmod p$$
$$SK_A=(Y)^a=g^{ab} \bmod p$$

Alice		Eve		Bob
[키 설정 과정] $a \in Z_p^*$ $X_1 = g^{aQ} \bmod p$	X_1 →	$X_1' = (X_1)^u \bmod p$ $= g^{auQ} \bmod p$	X_1' →	$b \in Z_p^*$ $Y_1 = g^{bQ} \bmod p$
	← Y_1		← Y_1	$X = (X_1')^{Q^{-1}}$ $= g^{au} \bmod p$
$Y = (Y_1)^{Q^{-1}} = g^b \bmod p$ $SK_A = (Y)^a = g^{ab} \bmod p$				$SK_B = (X)^b$ $= g^{abu} \bmod p$
[키 확인 과정] **1. Ku-Wang**	Y_2 →	$Y_2' = (Y_2)^u = g^{abuQ} \bmod p$	Y_2' →	
	← X'	$X' = (X)^{u^{-1}} = g^a \bmod p$	← X	
$X' \stackrel{?}{=} g^a \bmod p$				$(Y_2')^{Q^{-1}} \stackrel{?}{=} SK_B$
2. Lin *et al*	K_1 →	$K_1 = Y^{Q^{-1}} \bmod p$	K_1 →	
	← K_2'	$K_2' = (K_2)^{u^{-1}} \bmod p$ $= g^{aQ^{-1}} \bmod p$	← K_2	
$K_2' \stackrel{?}{=} g^{aQ^{-1}} \bmod p$				$K_1 \stackrel{?}{=} (g^b)^{Q^{-1}} \bmod p$

(그림 7) Enhanced 프로토콜에 대한 Advanced Modification 공격

④ Bob은 X값과 SK_B를 다음과 같이 계산한다.

$$X = (X_1')^{Q^{-1}} = g^{au} \bmod p$$
$$SK_B = (X)^b = g^{abu} \bmod p$$

능동적 공격자 Eve가 위와 같이 공격을 수행할 경우, Alice와 Bob은 키 확인 과정을 수행하기 전까지 그들이 설정한 세션키가 올바른 것인지 알 수 없게 된다.

[키 확인 과정에 대한 Advanced Modification 공격]

● Ku-Wang의 Enhanced 프로토콜에 대한 공격

⑤ Eve는 Alice가 전송한 Y_2값을 Y_2'로 대체하여 Bob에게 Y_2인 것처럼 전송한다.

$$Y_2' = (Y_2)^u = g^{abuQ} \bmod p$$

⑥ Eve는 Bob이 전송한 X값을 X'로 대체하여 Alice에게 X인 것처럼 전송한다.

$$X' = (X)^{u^{-1}} = g^a \bmod p$$

⑦ Alice와 Bob은 각각 X'과 Y'을 계산하여 확인한다. 만약 계산한 값이 정확하다면, 각각 자신들이 계산한 세션키가 정당하다고 신뢰한다.

$$\text{Alice}: X' \stackrel{?}{=} g^a \bmod p$$
$$\text{Bob}: (Y_2')^{Q^{-1}} \stackrel{?}{=} SK_B$$

Alice와 Bob은 키 확인 과정을 수행하여도 정당치 않은 세션키가 생성된 것을 알지 못하고 올바른 세션키로 신뢰하게 된다.

● Lin의 Enhanced 프로토콜에 대한 공격

① Eve는 Alice가 전송한 K_1을 Bob에게 그대로 전송한다.

$$K_1 = Y^{Q^{-1}} \bmod p$$

② Eve는 Bob이 전송한 K_2값을 K_2'으로 대체하여 K_2인 것처럼 Alice에게 전송한다.

$$K_2' = (K_2)^{u^{-1}} \bmod p = g^{aQ^{-1}} \bmod p$$

③ Alice와 Bob은 각각 다음을 계산한다.

$$\text{Alice}: K_2' \stackrel{?}{=} g^{aQ^{-1}} \bmod p$$
$$\text{Bob}: K_1 \stackrel{?}{=} (g^b)^{Q^{-1}} \bmod p$$

만약 계산한 값이 정확하다면, 각각 자신들이 계산한 세

선키가 정당하다고 신뢰한다. Alice와 Bob은 키 확인 과정을 수행하여도 올바른 값이 계산되기 때문에, 그들이 설정한 세션키를 정당한 것으로 신뢰하게 된다.

4. 제안하는 프로토콜

본 절에서는 위에서 정의한 Advanced Modification 공격에 대해 안전한 패스워드 기반 키 동의 프로토콜을 제안한다((그림 8) 참조).

[키 설정 과정]

① Alice와 Bob은 프로토콜을 시작하기 전에 각각 두 정수 $Q \bmod p$와 $Q^{-1} \bmod p$를 계산한다. 여기서 Q는 패스워드 S로부터 유도된 값으로, $(p-1)$과 서로 소인 값이어야 한다. 또한, 서로 다른 패스워드가 주어졌을 경우 동일한 Q값이 나올 확률이 매우 낮아야 한다.

② Alice는 랜덤 수 a를 선택하여 다음을 계산한 후 Bob에게 전송한다.

$$X_1 = g^{aQ} \bmod p$$

③ Bob은 랜덤 수 b를 선택하여 다음을 계산한 후 Alice에게 전송한다.

$$Y_1 = g^{bQ} \bmod p$$

④ Alice는 Bob으로부터 수신한 Y_1을 이용하여 다음을 계산한다.

$$SK_A = ((Y_1)^{aQ^{-1}}) = g^{ab} \bmod p$$

⑤ Bob은 Alice로부터 수신한 X_1을 이용하여 다음을 계산한다.

$$SK_B = ((X_1)^{bQ^{-1}}) = g^{ab} \bmod p$$

[키 확인 과정]

⑥ Alice는 다음을 계산하여 Bob에게 전송한다.

$$k_1 = H(SK_A, Q)$$

⑦ Bob은 다음을 계산하여 Alice에게 전송한다.

$$k_2 = H(SK_B, Q^{-1})$$

⑧ Alice와 Bob은 각각 다음 식을 이용하여 자신들이 계산한 세션키를 검증한다.

$$\text{Alice} : H(SK_B, Q^{-1}) \stackrel{?}{=} H(SK_A, Q^{-1})$$

$$\text{Bob} : H(SK_A, Q) \stackrel{?}{=} H(SK_B, Q)$$

Ku-Wany과 Lin등이 제안한 프로토콜에서는 공격자 Eve에 의한 Advanced Modification 공격을 통해, Alice와 Bob이 키 확인 과정을 수행한다 할지라도 잘못된 세션키를 신뢰하도록 할 수 있다. 이것은 공격자 Eve가 프로토콜에서 X_1 또는 Y_1을 계산하지 못하고, Q 또는 Q^{-1} 값을 알지 못한다 하더라도 Alice와 Bob을 속일 수 있음을 보여준다. 그러나 본 논문에서 제안하는 프로토콜에서는 키 확인 과정에서 Eve의 공격이 성공하려면 반드시 $g^{abu} \bmod p$값을 계산하여 Bob에게 전송할 수 있어야 한다. 그러나 이산대수 문제를 푸는 것이 계산상 불가능하고 패스워드가 비밀리에 보관되므로 이 값을 계산하는 것은 불가능하다. 그러므로 Eve는 Alice와 Bob을 속일 수 없게 된다.

5. 안전성 분석

본 논문에서 제안한 Advanced Modification 공격에 안전한 패스워드 기반 키 동의 프로토콜의 안전성은 이산대수 문제의 어려움과 비밀 패스워드 그리고 해쉬 함수의 일 방

Alice	공개정보 p, g	Bob
[키 설정 과정] $a \in Z_p^*$ $X_1 = g^{aQ} \bmod p$	X_1 → ← Y_1	$b \in Z_p^*$ $Y_1 = g^{bQ} \bmod p$
$SK_A = ((Y_1)^{aQ^{-1}}) = g^{ab} \bmod p$		$SK_B = ((X_1)^{bQ^{-1}}) = g^{ab} \bmod p$
[키 확인 과정]	$k_1 = H(SK_A, Q)$ → ← $k_2 = H(SK_B, Q^{-1})$	
$H(SK_B, Q^{-1}) \stackrel{?}{=} H(SK_A, Q^{-1})$		$H(SK_A, Q) \stackrel{?}{=} H(SK_B, Q)$

(그림 8) Advanced Modification 공격에 안전한 패스워드 기반 키 동의 프로토콜

향성에 의존한다.

● Advanced Modification 공격에 대한 안전성

본 논문에서 정의한 Advanced Modification 공격은 지금까지 제안된 모든 SAKA 프로토콜에 대해 유효하다. 본 공격에서 능동적 공격자 Eve는 두 개체 사이에서 전송되는 정보를 수신하여, 통신하는 두 개체가 잘못된 세션키를 설정하여 키 확인 과정을 수행하더라도 올바른 세션키를 생성한 것으로 신뢰하도록 하는 공격을 수행한다.

본 논문에서 제안한 프로토콜에서 Eve의 공격이 성공하기 위해서는 키 확인 과정에서 이산대수의 문제의 해와 해쉬 함수의 역을 구해야 한다. 그러나 이산대수의 문제를 푸는 것이 계산상 불가능하고 안전한 일방향 해쉬 함수를 사용한다면, Eve의 Advanced Modification 공격은 불가능하다. 따라서, 제안하는 프로토콜은 Advanced Modification 공격에 대해 안전하다.

● man-in-the-middle 공격에 대한 안전성

Man-in-the-middle 공격은 공격자가 통신하고자 하는 두 개체에게 정당한 사용자로 위장하여 공격하는 방식이다. 제안하는 프로토콜에서 Alice가 랜덤 수를 가지고 계산한 값을 Bob에게 전송하는 경우, 공격자가 이를 수신하여 임의의 값으로 대체하여 Bob에게 전송한다. 이러한 경우에, Bob이 랜덤 수를 사용하여 계산한 값을 전송하는 경우 패스워드를 알지 못하기 때문에 정당한 세션키를 계산 할 수 없다. 또한, 패스워드로부터 유도된 Q 또는 Q^{-1} 값을 모르기 때문에 정당한 사용자로의 위장이 불가능하다. 따라서, 제안하는 방식은 Man-in-the-middle 공격에 대해 안전하다.

● 사전 공격(dictionary attack)에 대한 안전성

사전 공격은 공격자가 키 교환 과정 동안 수행한 개체들 사이의 전송정보를 이용하여 패스워드 또는 세션키를 구하려는 공격이다. 제안하는 프로토콜의 경우, 공격자가 사전 공격을 성공하기 위해서는 키 확인 과정에서 세션키를 계산하여야 한다. 그러나 랜덤 수 a, b와 Q, Q^{-1}가 비밀로 유지되고 키 확인 과정에서 검증 정보가 비밀 정보를 포함하고 있으므로, 공격자가 사전 공격을 성공하기 위해서는 해쉬 함수의 역함수와 이산대수 문제를 해결해야 한다. 그러므로 제안하는 프로토콜은 사전 공격에 대해 안전하다.

● 패스워드가 노출된 경우의 안전성

만약 사전에 공유한 비밀 패스워드가 노출된다 하더라도, Alice와 Bob이 선택한 랜덤 수를 알지 못하기 때문에 공격자가 과거의 세션키를 계산하는 것은 불가능하다. 이것은 이산대수 문제를 푸는 어려움과 동치이므로 제안하는 프로토콜은 perfect forward secrecy를 제공한다.

본 논문에서 제안하는 프로토콜은 기존의 프로토콜과 상호 교환되는 통신량은 같으나, 키 확인과정을 수행하면서 필요로 하는 지수연산 과정을 사용하지 않고, 해쉬함수의 일방향성을 이용하여 키 확인과정을 수행하므로 지수연산을 수행하여 키 확인과정을 수행하는 것에 비해 계산량의 효율성을 제공한다. 다음 <표 1>은 제안하는 프로토콜과 기존의 SAKA 프로토콜들의 계산량과 상호 교환되는 통신량을 비교한 것이다.

〈표 1〉 각 프로토콜의 계산량 및 통신량 비교

구 분	계산량(지수연산)		통 신 량		비 고
	키 설정과정	키 확인과정	키 설정과정	키 확인과정	해쉬연산
SAKA	6	2	2		×
Tseng's 프로토콜	6	4	2		×
Ku-Wang's 프로토콜	6	2	2		×
Lin et al's 프로토콜	6	4	2		×
제안하는 프로토콜	4	0	2		○

PAK[7]프로토콜은 대표적인 패스워드 기반 키분배 프로토콜 중의 하나로써, 제안하는 프로토콜과 마찬가지로 이산대수분제와 해쉬함수의 안전성에 기반하여 제안된 프로토콜이다. 본 논문에서는 명시적 키 인증을 제공하는 프로토콜인 PAK 프로토콜과 제안하는 프로토콜의 특징을 분석하기 위해, 키설정 과정과 키 확인 과정에서의 지수연산량과 해쉬연산량, 그리고 상호 교환되는 통신량을 비교한다. 다음 <표 2>는 PAK 프로토콜과 제안하는 프로토콜을 비교한 것이다.

〈표 2〉 PAK 프로토콜과 제안하는 프로토콜 비교

구 분	계 산 량		통신량
	지수연산	해쉬연산	
PAK	4	7	3
제안하는 프로토콜	4	2	4

PAK 프로토콜과 제안하는 프로토콜은, 프로토콜의 수행에 있어서, 동일한 지수연산을 포함하고 있으나, PAK 프로토콜의 경우, 키 설정과 키 확인을 위해서는 제안하는 프로토콜보다 많은 해쉬연산을 필요로 한다. 상호 교환되는 통신량에 있어서는 PAK 프로토콜이 3번, 제안하는 프로토콜이 4번의 통신량을 가진다. PAK 프로토콜은 통신량의 측면에서, 그리고 제안하는 프로토콜은 해쉬연산의 측면에서 효율적인 면을 가지고 있으나, 기본적으로 수행하는 지수연산량이 같으므로, 실제 시스템에서 사용될 경우, 각 구현 환경에 적합한 프로토콜을 선택하여 사용할 수 있을 것으로 기대된다. 제안하는 프로토콜과 기존의 SAKA 프로토콜들의 경우, 키 설정 과정과 키 확인 과정이 구분되어 수행되는 반면, PAK 프로토콜의 경우 키 설정 과정과 키 확인 과정을 구분하지 않음으로써 통신량을 줄였으며, 제안하는

프로토콜은 키 확인 과정에서만 해쉬연산을 사용하여 적은 계산량을 가지는 장단점이 있다.

6. 결 론

최근 Seo-Sweeny는 Diffie-Hellman 방식을 기반으로 패스워드를 사용하여 두 개체들의 인증과 비밀 세션키를 공유하는 SAKA 프로토콜을 제안하였다. 이후, SAKA 프로토콜은 Tseng에 의해 재전송 공격에 대한 취약점이 발견되었고, Ku-Wang, Lin 등에 의해 패스워드 추측 공격과 Modification 공격에 대한 취약점이 제시되었으며, 이를 개선한 프로토콜들이 각각 발표되었다.

본 논문에서는 먼저 Electronic Letters에 발표된 SAKA 프로토콜과 이를 개선한 방식들에 대하여 살펴보았으며, Ku-Wang과 Lin 등이 제안한 프로토콜이 본 논문에서 정의한 Advanced Modification 공격에 대해 안전하지 못함을 보였다. 그리고 Advanced Modification 공격에 대해 안전한 키 확인 과정을 포함하는 개선된 프로토콜을 제안하였다. 본 논문에서 제안하는 패스워드 기반 키 동의 프로토콜은 man-in-the-middle 공격과 사전 공격에 대해 안전하고 perfect forward secrecy를 제공하며, 본 논문에서 정의한 Advanced Modification 공격에 대해서도 안전하다는 장점이 있다.

또한 제안하는 방식은 위에서 설명한 바와 같이, 기존의 프로토콜들과 키 설정과정에서의 계산량과 상호 교환되는 통신량은 동일하나, 키 확인과정에서 기존이 프로토콜들에서 필요로 하는 지수연산 대신 해쉬함수의 일방향성을 사용하여 확인 과정을 수행하므로 지수연산을 수행하는 것에 비해 계산효율을 제공할 수 있다.

본 논문에세 제안하고 있는 프로토콜의 활용분야로는, 현재 활발하게 연구되고 있는 유비쿼터스 환경에서의 RFID 태그 인식 기술분야에 적용할 수 있을 것으로 기대된다. RFID (Radio-Frequency Identification) 태그는 바코드(bar-code)를 대체하기 위해 개발된 방식으로 바코드 대신 상품에 태그를 부착하여 물류관리와 창고의 제고관리 등의 SCM(Supply-Chain Management) 분야에서 그 활용가치와 적용을 위한 연구가 활발히 진행되고 있다. 그러나 이러한 RFID 태그의 경우, 태그가 가지고 있는 계산능력이 현저히 낮으며 바코드를 대체하기 위해 소형화 저가격화 위주로 연구가 진행되고 있는 형편이므로, 본 논문에서 제안하는 프로토콜을 적용하여, 태그의 안전한 통신을 위한 연구를 진행할 수 있을 것으로 기대된다.

참 고 문 헌

[1] W. Diffie and M. E. Hellman, "New Directions in cryptography," *IEEE Transaction on Information Theory*, IT-22, 6, pp.644-654, 1976.

[2] R. Rivest, A. Shamir and L. Adleman, "A method of obtaining digital signature and public key cryptosystem," *ACM Communication*, Vol.21, No.2, pp.120-126, 1978.

[3] T. ElGamal, "A public key crytosystem and a signature scheme based on discrete logarithms," *IEEE Transaction on Information Theory*, Vol.31, pp.469-472, 1985.

[4] S. Bellovin and M. Meritt, "Encrypted key exchange : password-based protocols secure against dictionary attacks," *IEEE Symposium on Research in Security and Privacy*, pp.72-84, 1992.

[5] S. Bellovin and M. Meritt, "Augmented encrypted key exchange : a password-based protocol secure against dictionary attacks and password-file compromised," *ACM Conf. on Computer and Communicatons Security*, pp. 244-250, 1993.

[6] M. Boyarsky, "Public-key cryptography and password protocols : the multi user case," *ACM Conf. on Computer and Communicatons Security*, 1999.

[7] V. Boyko, P. MacKenzie and S. Patel, "Provably secure password authenticated key exchange using Diffie-Hellman," *Eurocrypt 00*, pp.156-171, 2000.

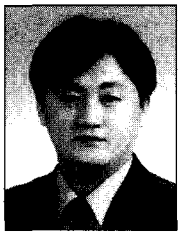
[8] M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated Key Exchange secure against Dictionary Attacks," *Eurocrypt 00*, pp.139-155, 2000.

[9] D. H. Seo and P. Sweeny, "Simple authenticated key agreement algorithm," *Electronics Letters*, Vol.35, No.13, pp.1073-1074, 1999.

[10] Y. M. Tseng, "Weakness in simple authen- ticated key agreement protocol," *Electronics Letters*, Vol.36, No.1, pp. 48-49, 2000.

[11] W. C. Ku and S. D. Wang, "Cryptanaysis of modified authenticated key agreement protocol," *Electronics Letters*, Vol.36, No.21, pp.1770-1771, 2000.

[12] I. C. Lin, C. C. Chang and M. S. Hwang, "Security enhancement for the simple authentication key agreement algorithm," *24th Annual International Computer Software and Application Conference*, pp.113-115, 2000.



곽 진

e-mail : jkwak@dosan.skku.ac.kr

2000년 성균관대학교 생물기전공학과 (공학사)

2003년 성균관대학교 대학원 전기전자 및 컴퓨터공학과(공학석사)

2003년~현재 성균관대학교 대학원 정보통신공학부 박사과정

관심분야 : 암호 프로토콜, 유비쿼터스 보안 등

오 수 현

e-mail : shoh@office.hoseo.ac.kr

1998년 성균관대학교 정보공학과(공학사)

2001년 성균관대학교 대학원 전기전자 및 컴퓨터공학과(공학석사)

2003년 성균관대학교 대학원 전기전자 및 컴퓨터공학과(공학석사)

2004년~현재 호서대학교 컴퓨터공학부 정보보호전공 전임강사

관심분야 : 암호 키 분배 프로토콜, 유비쿼터스 보안 등

양 형 규

e-mail : hkyang@kangnam.ac.kr

1983년 성균관대학교 전자공학과(공학사)

1985년 성균관대학교 대학원 전자공학과 (공학석사)

1995년 성균관대학교 대학원 정보공학과 (공학박사)

1984년~1991년 삼성전자 컴퓨터부문 선임연구원

1995년~현재 강남대학교 컴퓨터미디어 공학부 부교수

관심분야 : 암호 프로토콜, 네트워크 보안 등

원 동 호

e-mail : dhwon@dosan.skku.ac.kr

성균관대학교 전자공학과(학사, 석사, 박사)

1978년~1980년 한국전자통신연구원 전임 연구원

1985년~1986년 일본 동경공업대 객원 연구원

1988년~1999년 성균관대학교 교학처장, 전기전자 및 컴퓨터 공학부장, 정보통신대학원장, 정보통신기술연구소장

1996년~1998년 국무총리실 정보화추진위원회 자문위원

2002년~2003년 한국정보보호학회장

2003년~2004년 성균관대학교 연구처장

1982년~현재 성균관대학교 정보통신공학부 교수, 정보통신부 지정 정보보호인증기술연구센터장

관심분야 : 암호 프로토콜, 정보 보안 등