개선된 '간단한 인증키 동의 (Simple Authenticated Key Agreement)' 프로토콜☆

An Enhancement of Simple Authenticated Key Agreement Protocol

김 영 신* 김 윤 정** 황 준***
Young-Sin Kim Yoon-Jeong Kim Jun Hwang

요 약

Diffie-Hellman 키 교환 방법은 두 통신자간에 공통 세션키를 생성할 수 있으나, 중간자 공격 (man-in-the-middle attack)이 가능하다는 문제점을 안고 있다. 이러한 문제점을 해결하기 위하여 여러 가지 프로토콜들이 제안되었는데 Simple Authenticated Key Agreement (SAKA) 프로토콜도 그 중 하나이다. 이 프로토콜은 Seo-Sweeney, Tseng, Ku-Wang 등에 의하여 각각 제안된 바 있는데, 본 논문에서는 이들 프로토콜을 종합적으로 개선하여 안전하면서도 수행성능이 개선된 새로운 프로토콜을 제안한다. 기존 프로토콜들이 공통 세션키 생성단계와 검증단계를 구분하여 이루어지는 것에 비하여 본 논문에서 제안하는 프로토콜은 생성 단계와 검증 단계를 한꺼번에 처리함으로써, 수행시간 단축의 효과를 갖는다.

Abstract

The Diffie-Hellman Key Exchange scheme can produce a common session key between the two communicators, but its problem is that it makes a man-in-the middle attack possible. To solve problems like these, several protocols have been put forward, and the Simple Authenticated Key Agreement (SAKA) Protocol is among them. This protocol has been suggested by Seo-Sweeney, Tseng, and Ku-Wang, respectively. In this paper, we will put forward a new protocol that has been improved from all the original protocols mentioned above, but is still safe and quick to use. While the existing protocol divides the common session key production stage and the verification stage, the protocol suggested in this paper takes care of both of those stages simultaneously, therefore improving the processing performance.

☞ Keyword : Simple Authenticated Key Agreement Protocol, Diffie-Hellman Key Exchange

1. 서 론

두 통신자 간에 공통 세션키를 생성할 수 있는 방안으로 제안된 Diffie-Hellman 키 교환방법은, 사전 공유 정보 없이도 공통 세션키를 생성할 수 있는 획기적인 방법으로 평가받고 있다[1]. 그런데 이 방법은 중간자 공격(man-in-the-middle attack)이 가능하다는 단점을 안고 있다[2].

* 준 회 원 : 서울여자대학교 컴퓨터학과 박사과정
amaryllis@empal.com(공동저자)
** 정 회 원 : 서울여자대학교 정보통신공학부 조교수
yjkim@swu.ac.kr(제1저자)
*** 종신회원 : 서울여자대학교 정보통신공학부 교수
hjun@swu.ac.kr(공동저자)
☆ 이 논문은 2003년도 서울여자대학교 교내특별과제 연구비 지원을 받았음

이의 개선안의 일환으로 여러 방법들이 제안되었는데, 인증서 (certificate)를 사용하는 키 교환 프로토콜[3]과 두 명의 사용자 사이에 비밀 패스워드를 미리 공유하는 인증된 키 교환 프로토콜 (authenticated key exchange protocol) 등이 그것이다[4,5,6]. 전자의 경우 전송된 메시지의 무결성을 검증하기 위해 신뢰할 수 있는 제 3 의 인증기관을 필요로 한다. 이 시스템은 사용자 수가 증가하게 되면 사용자 인증서를 저장하기 위한 더 큰 용량의 저장소를 필요로 하게 되며, 많아진 사용자들의 전자서명을 검증하기 위해 높은 네트워크 대역폭을 필요로 하게 된다. 따라서 인증서를 사용하는 키 교환 프로토콜은 시스템의 확장이 어렵다. 게다가 인증기관이 손상되면 시스템 전체에

영향을 미친다는 단점을 가지고 있다.

후자는 두 통신자 간에 미리 암호키를 공유하고 있는 방법으로, 시스템의 안전성이 제 3의 인증기관이 아닌, 각 개인에게 의존한다는 특성을 갖는다. 이 방식의 대표적인 기법인 EKE (Encrypted Key Exchange) 방식에서는, 임의의 공개키/비밀키 쌍을 생성하고 공개키를 미리 공유된 패스워드로 암호화하여 전송한다. 이 방식은 off-line 패스워드 공격 등에 강하다는 장점이 있으나, 알고리즘이 복잡하며, 또한 특허권을 가지고 있어 널리 사용되지 못한다는 단점이 있다.

한편, 최근에 패스워드 기반의 인증 방식 중 하나로, '간단한 인증 키 동의 프로토콜 (SAKA: Simple Authenticated Key Agreement protocol)'이 제안되었다[7,8,9]. 이 프로토콜은 Diffie-Hellman 키 교환 방법에 기반하고 있으나 중간자 공격에 대하여 안전하다는 특성을 갖는다.

본 논문에서는 이 '간단한 인증 키 동의 프로토콜' 에 대한 기존 연구들을 정리 분석하고 기존 제안 방법들과 안전성은 동일하면서도 수행성능이 개선된 새로운 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서는 이전 연구 내용들을 기반으로 기존 SAKA 프로토콜을 정리 분석한 내용을 기술한다. 3 장에서 새로이 제안하는 프로토콜을 기술하고 이의 안전성 분석과 성능 분석 내용을 4 장과 5 장에서 각각 기술한다.

2. 관련 연구

최근에 Seo와 Sweeney가 공동으로 Diffie-Hellman 프로토콜을 기반으로 하는 새로운 키 동의 프로토콜 (key agreement protocol)인 simple authenticated key agreement algorithm (SAKA)을 제안하였다. 이 SAKA에서는, 두 사용자가 데이터를 송, 수신하기 위해 미리 공유된 패스워드를 갖고 있으며 이를 이용한 패킷을 교환함으로써 공통 세션키 (session key)를 생성하고, 또한 서로를 확인한다[7]. 그러나 SAKA 프로토콜은 침입자가 세션키를 이용하여 사용자를 속일 수 있는 문제점이 있음을 Tseng이 지적하고, key 검증 단계를 수정함으로써 해결방안을 제시한 바 있다 [8]. 이 해결 방안 또한 두 가지 방법에 의해 공격을 받을 수 있음을 KuWang이 증명하였고, key 검증 단계에서 전달되는 메시지를 변경함으로써 약점을 보완하고자 하였다[9].

본 절에서는 이들 세 가지 방법들의 세부 내용을 정리 분석한 내용을 기술한다.

2.1 Seo-Sweeney의 simple key agreement 프로토콜 [7,8]

Simple key agreement 프로토콜은 사전에 공유된 패스워드 (pre-shared password) 기술을 기반으로 하고 있다. 따라서 프로토콜이 시작되기 전에 앨리스와 밥은 비밀 패스워드 P를 공유하고 있고, Diffie-Hellman 키 교환 방법처럼 공통 값, n과 g를 가지고 있다고 가정한다.

Simple key agreement 프로토콜의 세션 키 (session key) 생성 단계는 다음과 같다.

① 앨리스와 밥은 각각 패스워드 P로부터 두 정수 Q와 $Q^{-1} \bmod (n-1)$을 각각 계산한다. 이 때 Q의 값은 미리 지정된 방법을 이용하여 계산된다.

② 앨리스는 임의의 정수 a를 선택하고 다음 식의 값을 밥에게 보낸다.

$$X = g^{aQ} \bmod n$$

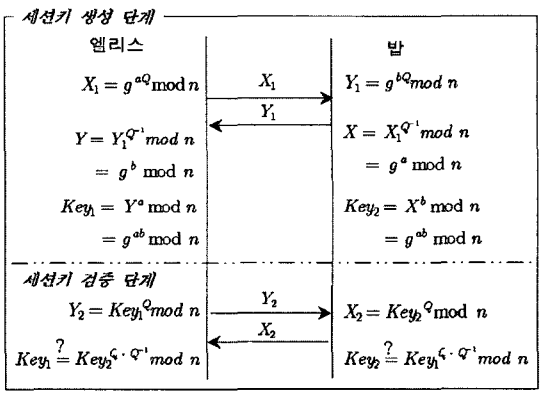
③ 밥 또한 임의의 정수 b를 선택한 후, 다음 식의 값을 앨리스에게 보낸다.

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

④ 앨리스는 세션키인 Key_1 다음과 같이 계산한다.

$$Y = Y_1^{Q^{-1}} \bmod n = g^b \bmod n$$
$$Key_1 = Y^a \bmod n = g^{ab} \bmod n$$

⑤ 밥은 세션키인 Key_2를 다음과 같이 계산한다.



(그림 1) S대-Sweeney 프로토콜의 세션키 생성 및 검증 단계

$$X = X_1^{Q^{-1}} \bmod n = g^a \bmod n$$
$$Key_2 = X^b \bmod n = g^{ab} \bmod n$$

그림 1의 윗부분에 Seo-Sweeney 방법의 세션키 생성 단계가 그림으로 나타나 있다. 앨리스와 밥은 각자 X_1과 Y_1을 패스워드 P를 이용하여 계산된 값 Q와 공통값 n과 g를 이용하여 계산한다. 그리고, 계산된 X_1과 Y_1을 서로 교환하고 이로부터 X, Y값을 구하고 이 값을 이용하여 공통 세션키 Key_1과 Key_2를 계산한다. 이 때 Key_1과 Key_2는 그림 1에서 볼 수 있듯이 $g^{ab} \bmod n$으로 값이 동일함을 알 수 있다.

세션키의 타당성을 검증하기 위해(상대방이 올바른 사용자임을 검증), 그림 1의 아래 부분과 같이, 앨리스와 밥은 다음의 세션키 검증 단계를 수행한다.

① 앨리스는 $Key_1^Q \bmod n$을 계산하고, 밥에게 보낸다.
② 밥은 $Key_2^Q \bmod n$을 계산하고 앨리스에게 보낸다.
③ 앨리스와 밥은 Q^{-1}을 이용하여 각각 전송받은 메시지에서 Key를 계산해내고, 자신의 세션키와 계산해 낸 Key를 비교한다.

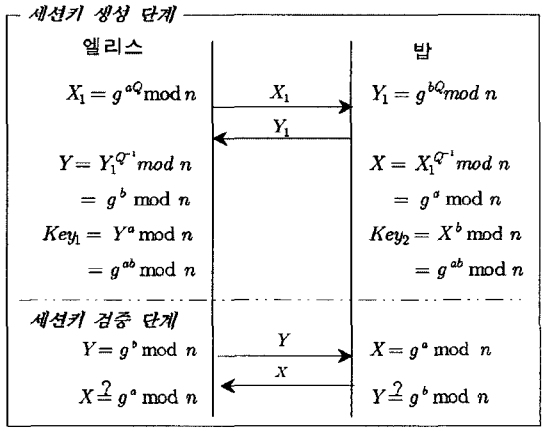
Seo-Sweeney 프로토콜의 세션키 검증 단계상에서의 취약점은 다음과 같다 [8]. 침입자 이브는 앨리스가 보낸 메시지 $Key_1^Q \bmod n$을 가로 챌 수 있으며, 그것을 밥이 보낸 정보처럼 다시 앨리스에게 보낼 수 있다. 그러면, 앨리스는 Q^{-1}을 이용해 $(Key_1^Q)^{Q^{-1}} \bmod n$을 계산하여 세션키를 계산한다. $Q \cdot Q^{-1} \equiv 1 \bmod (n-1)$이기 때문에 계산된 값은 Key_1이며, 따라서 앨리스는 침입자 이브를 밥으로 인지하게 된다.

2.2 Tseng 프로토콜 [8,9]

Seo-Sweeney 프로토콜의 약점을 극복하기 위해서 Tseng은 세션키 검증 단계를 그림 2의 아래 부분처럼 다음과 같이 수정했다.

① 앨리스가 Y를 밥에게 보낸다.
② 밥은 앨리스에게 X를 보낸다.
③ 앨리스와 밥은 각각 $X = g^a \bmod n$과 $Y = g^b \bmod n$ 인지를 검증한다.

따라서 수정된 프로토콜을 사용할 경우 침입자 이브가 앨리스로부터 $X_1 (= g^{aQ} \bmod n)$을 가로채더라도, 침입자 이브는 세션키 검증 단계에서



(그림 2) Tseng 프로토콜 세션키 생성과 검증 단계

$X = X_1^{Q^{-1}} \bmod n = g^a \bmod n$을 계산할 수 있어야 밥인 것처럼 앨리스를 속일 수 있다. 그러나 $g^a \bmod n$과 Q를 획득한다는 것은 불가능하므로, 침입자 이브는 X_1과 Y_1만을 가지고 정확한 X와 Y를 계산할 수 없다.

또한, Seo-Sweeney 프로토콜의 세션키 검증 단계에서는 $Key_1^Q \bmod n$과 $Key_2^Q \bmod n$을 계산하나, Tseng 프로토콜에서는 키 생성 단계에서 이미 계산된 X, Y를 검증 단계에서 이용하므로 $Key_1^Q \bmod n$과 $Key_2^Q \bmod n$을 계산할 필요가 없다. 따라서 Tseng 프로토콜은 Seo-Sweeney 프로토콜보다 계산량이 적어 계산 시간이 단축된다.

그런데, Tseng 프로토콜도 역시 공격을 받을 수 있는 약점을 가지고 있는데, 그것은 키 확인 메시지 두 개의 값이 $X_1 = Y_1$로 같을 수 있다는 데 있다. 이를 침입지가 데이터를 수정하지 않고 공격하는 경우와 데이터를 수정하여 공격하는 경우의 두 가지로 나누어 설명하면 다음과 같다[9].

우선, 데이터 수정이 없는 공격은 아래와 같다. 앨리스가 밥에게 X_1을 보낼 때, 침입자 이브는 X_1을 가로챈 후에 밥이 Y_1를 앨리스에게 보내는 것처럼 X_1을 그대로 밥에게 보낸다. 그 결과로, 앨리스는 다음을 계산한다.

$$Y = Y_1^{Q^{-1}} \bmod n = X_1^{Q^{-1}} \bmod n = g^a \bmod n$$

$$Key_1 = Y^a \bmod n = g^{a^2} \bmod n$$

그리고 앨리스는 밥에게 Y를 보내고, 침입자 이브는 밥인 것처럼 위장해서 앨리스에게 Y를 돌려보낸다. $Y = g^a \bmod n$을 확인한 뒤에, 앨리스는 이브를 밥으로 인정하게 되고 결과적으로 잘못된 세션키 Key_1을 믿게 된다.

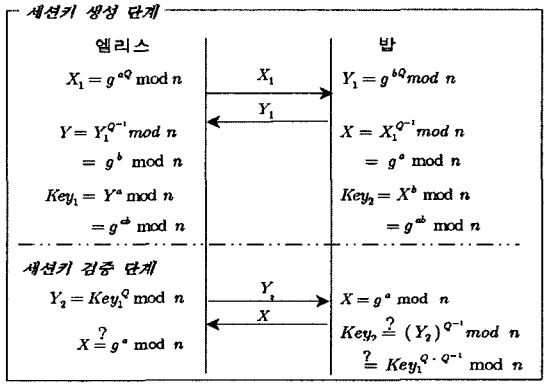
데이터를 수정하여 공격하는 방법은 다음과 같다. 앨리스에 의해 보내진 X_1이 침입자 이브에 의해 수정되어 X_1'으로 밥에게 전송된다 (이 때, X_1'은 $[1, n-1]$에 속하는 임의의 숫자이다). 밥은 앨리스에게 Y_1을 보낸다. 그 후, 앨리스는 대응하는 응답 Y를 밥에게 보낸다. 그러면, 밥은 X를 앨리스에게 보낼 것이다. X는 $(X_1')^{Q^{-1}} \bmod n$이고, $X' \neq g^a \bmod n$이기 때문에, 앨리스는 Key_1'을 믿지 않을 것이다. 그러나 밥은 $Y = g^b \bmod n$을 확인하고, $Key_2' = (X_1')^{Q^{-1}b} \bmod n$을 세션키로 믿게된다. 비록 침입자 이브가 Key_2'를 계산할 수 없다 하더라도, 침입자 이브는 이 잘못된 세션키를 믿게 함으로써 밥을 속일 수 있다. 또한, 침입자 이브가 밥에 의해 보내진 Y_1을 수정한 Y_1'을 앨리스에게 보낸다면, 이브는 앨리스도 마찬가지 방법으로 속일 수 있다.

2.3 Ku-Wang 프로토콜 [9]

Tseng 방법의 문제점을 개선하기 위하여, Ku와 Wang은 그림 3과 같이 새로운 키 검증 단계를 다음과 같이 제안하였다 [9].

① $Y_2 = (Key_1)^Q \bmod n = g^{abQ} \bmod n$을 앨리스가 계산한다. 그 후, 밥에게 Y_2를 보낸다.

② 밥은 $Y_2^{Q^{-1}} \bmod n = Key_2$를 만족하는지 여부를 검사한다. 만약 그렇다면 자신이 받은



(그림 3) Ku-Wang 프로토콜의 세션키 생성 및 검증 단계

X_1이 올바른 것이고 앨리스도 Y_1을 올바르게 받았다는 것을 알게 된다. 즉, 밥은 Key_2를 검증한 것이다. 그런 후에, 밥은 앨리스에게 X를 보낸다.

③ 앨리스는 $X = g^a \bmod n$를 계산한 값이 X와 동일한지를 검사한다. 만약, 동일하다면, 앨리스는 정확한 Y_1을 획득했고, 밥 또한 정확한 X_1을 획득했다는 것을 믿게 된다. 즉, 앨리스는 Key_1을 검증하게 된다.

이상과 같이 Ku-Wang은, Tseng의 프로토콜에서 전송되는 메시지가 대칭구조를 이루기 때문에 안전상의 취약점이 발생함을 밝히고, 검증단계에서 앨리스는 $Y_2 = (Key_1)^e \bmod n$를 밥에게 전송하고, 밥은 앨리스에게 $X = g^a \bmod n$를 전송하는 비대칭구조를 제시하였다.

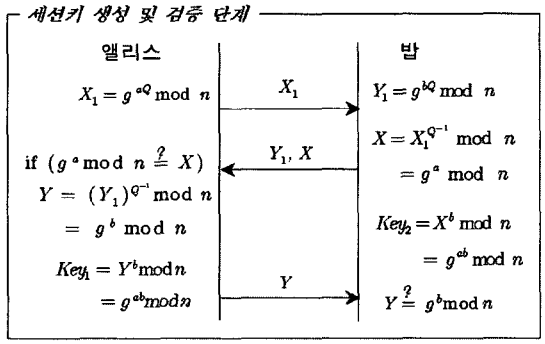
3. 새로운 SAKA 프로토콜

새로이 제안하는 프로토콜은 Tseng 프로토콜을 수정하여 구성되었다. 2장에서 제시된 Tseng 프로토콜의 안전성 문제는 세션키 생성단계에서 앨리스가 밥에게 전송 받은 Y_1이 변경되지 않았다는 것을 확인 할 수 있다면 해결될 수 있다. 따라서, 제안하는 프로토콜에서는 밥이 앨리스에게 Y_1을 보낼 때 X를 함께 보내어 검증을 먼저 수행하도록 하였다.

즉, 아래의 순서로 진행하는데, 이들은 그림 4에 나타나 있다.

① 앨리스와 밥은 각각 패스워드 P로부터 두 정수 Q와 $Q^{-1} \bmod (n-1)$ 을 각각 계산한다. 이 때 Q의 값은 미리 지정된 방법을 이용하여 계산된다.

② 앨리스는 임의의 정수 a를 선택하고 다음 식의 값을 밥에게 보낸다.



(그림 4) 제안 프로토콜의 세션키 생성과 검증 단계

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

③ 밥은 임의의 정수 b를 선택하여 $Y_1 = g^{bQ} \bmod n$을 계산한다. 그리고 $X = X_1^{Q^{-1}} = g^a \bmod n$을 계산하고, Y과 X를 앨리스에게 보낸다.

④ 앨리스는 $X = g^a \bmod n$ 가 맞는지를 검증한다.

⑤ ④의 검증결과가 맞으면 $Y = (Y_1)^{Q^{-1}} \bmod n = g^b \bmod n$을 계산하고 이로부터 $Key_1 = g^{ab} \bmod n$을 계산한다. 그리고, 밥에게 Y를 전송한다.

⑥ ④의 검증결과가 맞지 않으면 즉, $X \neq g^a \bmod n$인 경우 앨리스는 프로토콜을 더 이상 진행하지 않는다.

⑦ 밥은 앨리스로부터 Y를 받으면, $Y = g^b \bmod n$ 인지를 검증한다. 밥이 앨리스로부터 Y를 전송받지 못하는 경우에는 (⑥의 경우로, 앨리스가 검증결과가 맞지 않아 밥에게 Y를 전달하지 않음), 밥은 세션키를 검증할 수 없다.

4. 제안 SAKA 프로토콜 안전성 분석

본 절에서는 제안하는 프로토콜의 안전성에 대한 분석 내용을 기술한다. 안전성 분석은, 제안 프로토콜이 Tseng 프로토콜의 안전에 대한 취약

성을 개선하여 구성된 관계로, 우선 Tseng 프로토콜을 공격하는 기법에 대하여 강함을 보인다. 다음으로, Perfect forward secrecy에 대한 안전성 분석 내용을 기술한다.

Tseng 프로토콜에 대한 데이터 수정이 없는 공격을 제안 프로토콜에 적용해 보자. 제안 프로토콜에서, 앨리스가 밥에게 X_1을 보낼 때 (단계 ②), 침입자 이브는 X_1을 가로챈 후에 밥이 Y_1을 앨리스에게 보내는 것처럼 X_1을 그대로 밥에게 보낼 수 있다 (단계 ③). 그런데, 단계 ③에서는 X_1과 함께 X도 전송되야하고, 단계 ④에서 앨리스는 X를 검증한다. 침입자 이브는 패스워드로부터 도출된 Q 값을 모르므로 올바른 X 값을 전송할 수 없다. 따라서, 단계 ④의 검증 과정에서 공격이 실패한다.

Tseng 프로토콜에 대한 데이터를 수정한 공격을 제안 프로토콜에 적용한 결과는 다음과 같다. 앨리스에 의해 보내진 X_1이 침입자 이브에 의해 수정되어 X_1'으로 밥에게 전송된다 (단계 ②). 밥은 앨리스에게 Y_1과 X를 보낸다 (단계 ③). 단계 ④에서 앨래스는 X를 검증한다. 이 검증이 실패하면 앨리스는 밥을 올바른 밥으로 인정하지 않고 또한, 밥에게 Y를 전송하는 작업도 수행하지 않는다. 밥은 앨리스로부터 Y를 전송받지 못하므로 단계 ⑦을 진행할 수 없고 따라서 밥도 침입자 이브가 관여된 상대방을 인증하지 않게 된다. 즉, 앨리스와 밥 모두 침입자 이브의 공격에 대하여 취약하지 않다.

Perfect forward secrecy 특성은 패스워드가 손상되더라도, 이전의 세션키 Key_1의 값이 누출되지 않아야 한다는 성질이다 [10]. Seo-Sweeney 프로토콜과 Ku-Wang 프로토콜의 경우 Key_1^Q 또는 Key_2^Q 값이 네트워크 상에 공개되므로 알려진 패스워드를 이용하여 $Key_1=(Key_1^Q)^{Q^{-1}}$ 또는 $Key_2=(Key_2^Q)^{Q^{-1}}$를 계산할 수 있었다. 그러나, 제안 프로토콜에서는 Key_1^Q 또는 Key_2^Q 값이 네트워크상에 전송되지 않는다. 따라서, 패스워드가 손상되더라도 세션키의 값을 유추할 수 없다. 즉, 제안 프로토콜은 perfect forward secrecy를 만족한다.

Seo-Sweeney 방법, Tseng 방법, Ku-Wang 방법, 그리고 본 논문에서 제안하는 방법의 안전성을 비교한 결과가 표 2의 2번 째 열에 나타나 있다. Seo-Sweeney 방법과 Tseng 방법은 2.1절과 2.2절에서 기술한 것처럼 알려진 취약점을 갖고 있으며, Ku-Wang 방법과 제안 방법은 취약점이 없다.

5. 제안 SAKA 프로토콜 성능 분석

제안 프로토콜의 성능 분석을 위하여, 메시지 전송 횟수와 프로토콜 수행시 필요한 연산 시간을 계산하였다. 연산 시간은 모듈라 멱승 (exponentiation) 연산의 수행 시간이 비교연산 등의 수행시간에 비하여 월등히 큰 관계로, 필요한 연산 시간은 모듈라 멱승 연산의 횟수를 계산함으로써 수행하였다.

제안 프로토콜은 그림 4에서 보이듯, 일반적인 경우 총 3 회의 메시지 전송이 발생한다. 침입자 이브가 중간자 공격을 시도하는 경우에는 앨리스가 이름 감지하고 더 이상의 프로토콜 진행을 하지 않음으로써 2 회의 메시지 전송이 수행된다.

제안 프로토콜에서 수행하는 모듈라 연산은 표 1과 같이 총 8회 수행된다. 제안 프로토콜의 단계 ④에서 $X=g^a \bmod n$을 검증할 때는 단계 ①에서 계산한 $g^a \bmod n$을 이용하므로, $g^a \bmod n$ 연산의 횟수는 한번만 수행되면 된다. 마찬가지로 단계 ③과 단계 ⑦에서 필요한 $g^b \bmod n$ 연산도 한 번만 수행되면 된다. 즉, 일반적인 경우 모듈라 멱승 연산은 총 8회 수행된다. 침입자 이브가 중간자 공격을 수행하는 경우 단계 ⑤가 수행되지 않으므로, $Y_1^{Q^{-1}}$과 $Y^b \bmod n$ 연산은 수행되지 않는다 (표 1의 6,7 번). 즉, 앨리스가 공격을 감지하는 경우에는 모듈라 멱승 연산이 총 6회 수행된다.

Seo-Sweeney 방법, Tseng 방법, Ku-Wang 방법,

(표 1) 제안프로토콜에서 수행되는 모듈라 멱승 연산

번호	모듈라 멱승 연산	비고 (수행위치)
1	$g^a \bmod n$	앨리스
2	$(g^a)^Q \bmod n$	앨리스
3	$g^b \bmod n$	밥
4	$(g^b)^Q \bmod n$	밥
5	$X_1^{Q'} \bmod n$	밥
6	$Y_1^{Q'} \bmod n$	앨리스
7	$Y^b \bmod n$	앨리스
8	$X^b \bmod n$	밥

그리고 본 논문에서 제안하는 방법의 성능을 비교한 내용이 표 2의 3번째, 4번째 열에 나타나 있다. 각각의 프로토콜에 대하여 메시지 전송 횟수와 모듈라 멱승 연산의 횟수를 계산하였는데, 제안 방법이 두 가지 경우 모두에 있어서 효율적임을 알 수 있다.

6. 결 론

불안전한 통신 채널을 통하여 두 사용자간에 비밀키를 확보하는 알고리즘으로 잘 알려진 Diffie-Hellman 키 교환 방법은 중간자 공격에 약하다는 단점을 가지고 있다. 본 논문에서는, 중간자 공격에 대한 문제를 해결하기 위해 제안된 키 분배 알고리즘인 SAKA (Secure Authenticated Key Agreement) 알고리즘에 대한 내용들을 정리 분석하였다. 그리고, SAKA 중 가장 안전하다고 알려진 프로토콜과 동일한 안전성을 유지하면서도 기존 방법들보다 성능은 효율적인 새로운 방법을 제안하였다.

참고문헌

[1] W. Diffie, M. Hellman, "New directions in Cryptography", IEEE Trans. on Information Theory, IT-22(6):644-654, November 1976.

[2] Bmce Schneier, Applied Cryptography-Protocols, Algorithms, and Source Code in C, 2nd edi., John Wiley & Sons, Inc., 1995.

[3] W. Diffie, P.C. Van Oorschot, M.J. Wiener, "Authentication and authenticated key exchanges", Design, Codes and Cryptography, Vol. 2, pp. 107~125, 1992.

[4] S. Bellovin, M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks", Proc. of IEEE Conf. on Research in Security and Privacy, pp. 72~84, May 1992.

[5] 박왕석, 정종필, 박창섭, 이동훈, "패스워드를 이용한 인증 프로토콜들에 대한 고찰", 한국정보보호학회지 제9권 제4호 pp. 51~63, 1992.

[6] 서승현, 조태남, 이상호, "OPT-EKE: 원-타임 패스워드 기반의 키 교환 프로토콜", 한국정보과학회 논문지 제 29권 5호 pp. 291~298, 2002.

[7] Dong Hwi Seo and P. Sweeney, "Simple authenticated key agreement algorithm", Electronics Letters, Vol. 35, No. 13, June, 1999.

(표 2) SAKA 프로토콜들의 안정성 및 성능 비교

프로토콜	안정성	메시지 전송 횟수			계산 속도 (멱승연산 회수)		
		세션키 생성단계	세션키 검증단계	총 회수	세션키 생성단계	세션키 검증단계	총 회수
Seo-Sweeney 방법	취약점 있음	2	2	4	8	2	10
Tseng 방법	취약점 있음	2	2	4	8	0	8
Ku-Wang 방법	발견된 취약점 없음	2	2	4	8	1	9
제안 방법	발견된 취약점 없음	3 (2)*		3 (2)	8 (6)		8 (6)

*() 침입을 감지한 경우의 횟수이다

[8] Yuh-Min Tseng, "Weakness in simple authenticated key agreement protocol", Electronics Letters, Vol. 36, No. 1, Jan, 2000.

[9] Wei-Chi Ku and Sheng-De Wang, "Cryptanalysis of modified authenticated key agreement protocol", Electronics Letters, Vol. 36, No. 21, Oct, 2000.

[10] Iuon-Chang Lin, Chin-Chen Chang, Min-Shiang Hwang, "Security Enhancement for the simple authentication key agreement algorithm", Proceedings of the 24th Annual International Computer Software and Application Conference, pp. 113~115, 2000.

◐ 저 자 소 개 ◑

김 영 신
1999년 서울여자대학교 컴퓨터학과 졸업(학사)
2002년 서울여자대학교 대학원 컴퓨터학과 졸업(석사)
2002년~현재 : 서울여자대학교 대학원 컴퓨터학과 박사과정
관심분야 : 그리드 컴퓨팅, 암호학
E-mail : amaryllis@empal.com

김 윤 정
1991년 서울대학교 컴퓨터공학과 졸업(학사)
1993년 서울대학교 대학원 컴퓨터학과 졸업(석사)
2000년 서울대학교 대학원 전기컴퓨터공학부 졸업(박사)
2002년~현재 : 서울여자대학교 정보통신대학 정보통신공학부 조교수
관심분야 : 암호학, 시스템 보안, 암호 응용
E-mail : yjkim@swu.ac.kr

황 준
1985년 중앙대학교 컴퓨터공학과 졸업(학사)
1987년 중앙대학교 대학원 컴퓨터공학과 졸업(석사)
1991년 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
1992년~현재 : 서울여자대학교 정보통신공학부 교수
관심분야 : 그리드 컴퓨팅, 분산 처리
E-mail : hjun@swu.ac.kr