상호 인증 키 교환 프로토콜의 안전성 분석

이재민[0], 류은경*, 김기원*, 이형목*, 유기영*
[0]경북대학교 정보통신학과
*경북대학교 컴퓨터공학과
(a1035[0], ekryu, kwkim, hmhl01, yook)@purple.knu.ac.kr

Security analysis for authenticated key agreement protocol

Jae-Min Lee[0], Eun-Kyung Ryu*, Kee-Won Kim*, Hyung-Mok Lee*, Kee-Young Yoo*
[0]Dept. of Information and Communication at Kyungpook National University
*Dept. of Computer Engineering at Kyungpook National University

요 약

Seo와 Sweeny는 통신 당사자간의 직접적인 세션키(session key) 교환을 위해 SAKA(Simple Authenticated Key Agreement Algorithm)를 제안했다. SAKA는 패스워드(password)를 사용하여 사용자 인증 기능을 제공하는 변형된 Diffie-Hellman 키 교환 프로토콜로써, 키 생성 및 사용자 인증 시 요구되는 계산량과 메시지 전송량을 고려할 때 효율적인 프로토콜이다. 그러나, 최근에 Lin은 SAKA의 안전성에 취약점이 있음을 지적하고 개선된 프로토콜을 제안하였다. 본 논문에서는 개선된 프로토콜이 여전히 재전송 공격 (replay attack)에 안전 하지 않기 때문에 사용자 인증을 제공 할 수 없음 을 보인다.

1. 서론

공개된 통신 채널을 기반으로 안전한 통신을 하기 위해서는 전송될 정보의 암호화가 요구된다. 이를 위해서는 먼저 통신 당사자간에 공통으로 사용할 수 있는 키의 공유가 필요하다. 대표적인 키 공유 방법으로는 유한필드(finite field) 상에 이산 대수 문제(discrete logarithm problem)를 기반으로 하는 Diffie-Hellman 키 교환 프로토콜(key exchange protocol)이 있다[1]. Diffie-Hellman 키 교환 프로토콜은 세션 키(session key)를 생성하기 위해 요구되는 메시지 전송량이 매우 작다는 장점이 있으나, 사용자 인증을 제공할 수 없기 때문에 중간자 공격(man-in-the-middle attack)에 취약하다는 단점이 있다.

사용자 인증(user authentication)문제를 해결하기 위한 기존의 연구는 인증서(certificate)를 이용하는 방법과 패스워드(password)를 기반으로 하는 방법으로 분류된다. 인증서를 이용하는 하는 방법은 CA(Certifying Authority)로부터 발행된 인증서를 통해 사용자의 공개 키(public key)를 인증함으로써 사용자 인증기능을 제공한다. 그러나 인증서 기반의 키 교환 프로토콜은 시스템 확장이 어렵고, 인증서를 보관하기 위한 대용량의 저장 장치와 사용자가 증가함에 따라 각각의 사용자 인증을 위해 더 큰 대역폭(band-width)을 필요로 한다. 또한 시스템의 보안이 전적으로 CA에게 의존한다. 패스워드를 기반으로 하는 방법은 통신 당사자간에 미리 공유된 패스워드를 사용하여 세션 키를 생성함으로써 사용자 인증을 제공한다. 이 방법은 시스템의 보안이 CA에 의존되지 않으며, 또한 중간자 공격을 막을 수 있다. 이러한 장점 때문에 현재까지 패스워드를 기반으로 키 교환 프로토콜에 관한 많은 연구가 이루어져 왔다.[2-8]

특히, Seo와 Sweeny[9]는 패스워드를 기반으로 하는 SAKA (Simple Authenticated Key Agreement Algorithm)를 제안하였다. SAKA는 패스워드를 사용하여 사용자 인증 기능을 추가한 변형된 Diffie-Hellman 키 교환 프로토콜로써, 키 생성 및 사용자 인증 시 요구되는 계산량과 메시지 전송량을 고려할 때 효율적인 프로토콜이다. 그러나 이후에 Sun[10]에 의해 SAKA의 안전성에 문제점이 있음이 밝혀졌다. 그러한 문제점을 해결하기 위해서 Lin[11]은 개선된 상호 인증 키 교환 프로토콜을 제안하였다. 본 논문에서는 개선된 프로토콜이 여전히 재 전송 공격(replay attack)에 안전 하지 않기 때문에 사용자 인증을 제공 할 수 없음을 보인다.

본 논문의 구성은 다음과 같다. 2장에서 SAKA를 간략히 기술한 후 SAKA프로토콜의 문제점과 그 문제점을 개선한 Lin의 상호 인증 키 교환 프로토콜에 대해서 설명한다. 3장에서는 앞장에서 기술한 Lin의 상호 인증 키 프로토콜의 안전성을 분석한다. 마지막으로 결론을 맺는다.

2. 관련 연구

2.1 SAKA(Simple Authenticate Key Agreement Algorithm)

SAKA는 미리 공유된 패스워드를 사용하여 세션 키를 생성하여 상호인증 기능을 제공하는 프로토콜로써, Diffie-Hellman 키 교환 프로토콜을 기반으로 한다.

본 논문에서 Alice와 Bob은 안전한 통신을 위해 세션 키를 교환하고자 하는 통신 당사자이고, Eve는 통신을 방해하는 공격자라 가정한다. 시스템 매개변수는 p와 g로 구성된다. 이때, p는 소수이고 g는 $GF(p)$ 상의 원시 근(primitive root)로서 공개된 값이다. SAKA는 키 생성과정과 키 검증과정으로 이루어지며, 각각의 과정은 다음과 같다.

[키 생성 과정]

(e.1) 프로토콜을 시작하기 전에 Alice와 Bob은 사전에 패스워드 S를 공유하고 있다고 가정한다. 또한 통신을 시작하기 전에 미리 정해 놓은 방법을 사용해서, S로부터 Q와 $Q^{-1} \bmod p-1$을 계산한다. 예를 들어, $Q = S + c \bmod p-1$ 과 같이 S로부터 Q를 계산한다.

(e.2) Alice는 임의의 정수 a를 선택하여, X_1을 계산한다.

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

그리고, X_1을 Bob에게 전송한다.

(e.3) Bob은 임의의 정수 b를 선택하여, Y_1을 계산한다.

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

그리고, Y_1을 Alice에게 전송한다.

(e.4) Alice는 Y_1을 전송 받은 후, key_1을 계산한다.

$$Y = Y_1^{Q^{-1}} \bmod p$$

$$key_1 = Y^a \bmod p$$

(e.5) Bob은 X_1을 전송 받은 후, key_2를 계산한다.

$$X = X_1^{Q^{-1}} \bmod p$$

$$key_2 = X^b \bmod p$$

[키 검증 단계]

(v.1) Alice는 $key_1^{\,Q}$를 Bob에게 전송한다.

(v.2) Bob은 $key_2^{\,Q}$를 Alice에게 전송한다.

(v.3) Alice는 전송 받은 값에 Q^{-1}을 사용해서 key_1와 동일한지를 확인한다

$$(key_2^{\,Q})^{Q^{-1}} \bmod p = g^{ab} \bmod p$$

(v.4) Bob은 전송 받은 값에 Q^{-1}을 사용해서 key_2와 동일한지 확인하다.

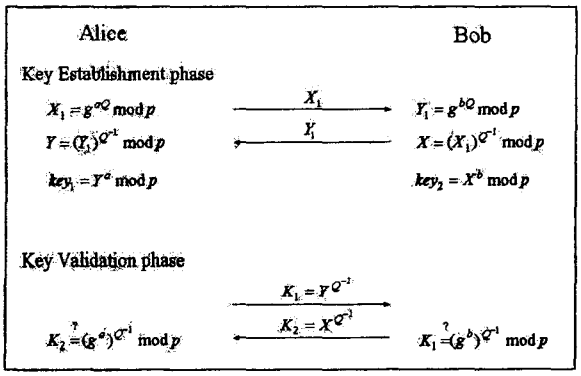
$$(key_1^{\,Q})^{Q^{-1}} \bmod p = g^{ab} \bmod p$$

Alice와 Bob은 (v.3), (v.4) 단계에서 전송 받은 키와 자신이 생성한 세션 키의 값을 확인 함으로서 상대방이 인증된 사용자인지를 서로 확인할 수 있다.

2.2 SAKA의 문제점 및 개선된 알고리즘

SAKA는 패스워드를 기반으로 사용자 인증기능을 제공하는 키 교환 프로토콜로서, 키 교환과 사용자 인증시 요구되는 계산량과 메시지 전송량을 고려할 때 효율적인 프로토콜이지만, 다음과 같은 문제점들이 있다.[10] 첫째, 사용자의 신원(identity)을 확인할 수 없다. 둘째, 사전 공격(dictionary attack)에 안전하지 못 하다. 셋째, PFS(perfect forward secrecy)를 제공하지 못한다.

Lin[11]은 이러한 문제점들을 해결하기 위해 다음과 같이 SAKA의 키 검증 단계를 개선하였다. 프로토콜에서 사용되는 시스템 매개변수와 키 생성 과정은 SAKA와 동일하며, Lin의 상호 인증 키 교환 프로토콜은 <그림 1>과 같다.



<그림.1> Lin의 상호인증 키 교환 프로토콜

[개선된 키 검증 단계]

(i.1) Alice는 $K_1 = Y^{Q^{-1}} \bmod p$를 계산해서 Bob에게 전송한다.

(i.2) Bob은 $K_2 = X^{Q^{-1}} \bmod p$를 계산해서 Alice에게 전송한다.

(i.3) Alice는 $K_2 = X^{Q^{-1}} \bmod p$를 전송 받은 후 다음을 확인한다.

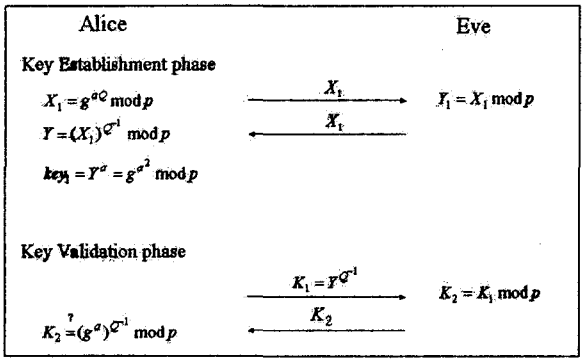
$$K_2 = (g^a)^{Q^{-1}} \bmod p$$

(i.4) Bob은 $K_1 = Y^{Q^{-1}} \bmod p$를 전송 받은 후 다음을 확인한다.

$$K_1 = (g^b)^{Q^{-1}} \bmod p$$

3. Lin의 상호 인증 키 교환 프로토콜에 대한 안전성 분석

Lin은 제안한 프로토콜이 SAKA의 문제점을 해결할 뿐만 아니라, 보다 안전한 상호 인증 키 교환 프로토콜임을 주장하였다. 그러나, Lin의 프로토콜에서 통신 채널상에 노출되는 값 X_1을 가로챌 수 있는 공격자 Eve는 <그림 2>와 같이 재전송 공격(replay attack)이 가능하다.



<그림.2> 재전송 공격

먼저, 프로토콜의 키 생성과정 (e.2)에서 Alice가 Bob에게 키 생성을 위해 전송한 메시지 X_1 을 Eve가 가로채어 X_1을 (e.3)에서 Alice에게 재 전송한다. 이에 대응하여, Alice는 재전송된 X_1사용하여 공유 키 key_1과 검증 메시지 K_1을 계산한다. 또한 Alice는 생성한 세션 키를 사용하기 전에 프로토콜의 키 검증 단계 (i.1)에서 검증 메시지 K_1을 전송하고, Eve는 Alice가 전송한 K_1을 Alice에게 재 전송한다. Alice는 전송 받은 K_1 값이 (i.3)식을 만족하는지 확인하여 통신 상대자를 검증하게 된다. Lin의 프로토콜에서 공격자 Eve에 의한 재전송 공격 과정의 세부 단계를 설명하면 다음과 같다.

[재 전송 공격 과정]

(a.1) Alice는 키 생성을 위해 메시지 X_1을 Bob에게 전송한다.

(a.2) Eve는 Bob에게 전송된 X_1을 가로채어 Alice에게 재 전송한다.

(a.3) Alice는 X_1 을 전송 받은 후 key_1과 K_1을 계산한다.

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

(a.4) Eve는 가로챈 $K_1 = g^{aQ^{-1}} \bmod p$을 Alice 에게 재전송한다.

(a.5) Alice는 Eve로부터 전송 받은 K_1이 다음의 식을 만족하는지 확인함으로써 공격자 Eve를 인증한다.

$$K_1 = (g^a)^{Q^{-1}} \bmod p$$

이와 같이,프로토콜에서 공격자 Eve는 통신 당사자간에 미리 공유된 패스워드에 관계없이 키 교환과 인증과정에서 노출되는 메시지를 단순히 재전송 함으로써 인증된다. 따라서, Lin의 상호 인증 키 교환 프로토콜은 사용자 인증 기능을 제공 할 수 없다.

4. 결론

본 논문에서는 Lin이 제안한 상호 인증 키 교환 프로토콜의 안전성을 분석 하였다. 분석결과, Lin의 상호인증 키 교환 프로토콜은 재전송 공격(replay attack)에 안전하지 않아 사용자 인증 기능을 제공할 수 없음을 알 수 있었다.

5. 참고문헌

[1] Diffie. W. and M. Hellman., New directions in cryptography, *IEEE Trans. on Information Theory*, IT-22, pp.644 654, 1976.

[2] S. Bellovin and M. Merrit., Encrypted key exchange: password-based protocols secure against dictionary attacks, *Proceedings of IEEE Comp. Society Symp. on Research in Security and Privacy*, pp.72-84, May, 1992.

[3] S. Bellovin and M. Merrit., Augmented encrypted key exchange: password file compromise, *Proceedings of the 1st ACM conference on Computer and communications Security*, pp.244-250, 1993.

[4] L. Gong , M.Lomas, R.Needham, and J.Saltzer. , Protecting poorly chosen secrets form guessing attacks, *IEEE Journal on Selected Areas in Communications*, vol.11, no.5, pp.648-656, 1993.

[5] D. Jablon., Strong password-only authenticated key exchange, *ACM Comput. Commun. Rev.*, vol.20, no.5, pp.5-26, October, 1996.

[6]D.Jablon., Extended password key exchange protocol, *Proceeding of WETICE Workshop on Enterprise Security*, June, 1997.

[7] S. Lucks., Open key exchange: How to defeat dictionary attacks without encrypting public keys, *Proceeding of the Security Protocol Workshop*'97, pp.7-9, April, 1997.

[8] T. Wu. , Secure remote password protocol, *Proceeding of Internet Society Network and Distributed System Security Symp.*, pp.97-111, May, 1998.

[9] D. Seo and P. Sweeny, Simple authenticated key agreement algorithm, *Electronics Letters*, vol.13, no.35, pp.1073-1074, June, 1999.

[10]H. Sun., On the security of simple authenticated key agreement algorithm, *Proceedings of the Management Theory Workshop*'2000, 2000.

[11]Iuon-Chang Lin, Chin-Chen Chang and Min-Shiang Hwang, Security Enhancement for the Simple Authentication Key Agreement Algorithm, *Computer Software and Applications Conference.*, pp.113-115, 2000.

[12] KU. W. C and WANG.D., Cryptanalysis of modified authenticated key agreement protocol, *Electronic Letters*, vol.36, no.21, pp.1770 -1771, October, 2000.

[13]TSENG.Y.M., Weakness in simple authenticated key agreement protocol, *Electronic Letters*, vol.36, no.6, pp.48-49, January, 2000.

[14] B. T. HSIEH, H. M. SUN and T. HWANG, Cryptanalysis of enhancement for simple authentication key agreement algorithm, *Electronic Letters*, vol.38, no.3, pp.20-21, January, 2002.