무선 통신을 위한 진보된 키 합의 프로토콜*

유재길†, 윤은준, 유기영‡
경북대학교 컴퓨터공학과 정보보호연구실

Advanced Key Agreement Protocol for Wireless Communication

Jae-Gil Yu†, Eun-Jun Yoon, Kee-Young Yoo‡
Dept. of Computer Engineering, Graduate School, Kyoungpook National University, Deagu, South Korea

요 약

Diffie-Hellman기반 키 합의 프로토콜들은 비교적 고비용의 연산인 지수연산으로 인해, 유선 네트워크 환경에 비해 저전력이고 컴퓨팅 자원이 제한되어 있는 무선 네트워크 환경에서는 비효율적이고 구현하기 어려운 문제가 있다. 이에 Yang등은 대리서버(Proxy Server)를 이용하여 Diffie-Hellman방식을 적용하면서도 단말 무선 네트워크 사용자의 지수연산부담을 감소시키는 효율적인 키 합의 프로토콜(이하 SEKAP)을 제안하였다. 그러나 SEKAP는 재전송공격(Replay Attack), 알려지지 않은 키 공유 공격(Unknown Key Share Attack), 그리고 키 노출로 인한 위장공격(Key Compromised Impersonation Attack) 등에 취약하며 전방향 안전성(Forward Secrecy)을 제공하지 못한다. 본 논문에서는 SEKAP가 위 공격들에 대해 취약함을 보이고, 세션키의 상호인증을 추가한 개선된 프로토콜을 제안한다.

I. 서론

키 합의 프로토콜은 네트워크상에서 안전한 통신을 위해서 중요한 요소이다. 기존의 유선 네트워크 환경에서는 Diffie-Hellman 알고리즘을 이용하여 안전한 보안성을 제공하는 키 합의 프로토콜들이 제안되어져 왔다. 하지만 Diffie-Hellman기반 키 합의 프로토콜들은 고비용의 지수연산을 필요로 함으로, 제한적인 자원과 저전력 특징을 가지는 무선 네트워크 환경에는 비효율적이며 구현하기 어려운 문제가 있다. 이에 Yang등[1]은 무선 네트워크 환경에 적합하며, Diffie-Hellman 알고리즘을 적용하면서, 무선 단말 사용자의 지수연산 부담을 감소시킨 효율적인 키 합의 프로토콜을 제안하였다. SEKAP는 dhHybrid [2]의 개념을 이용하여 1024비트의 세션키를 512비트의 두 부분으로 나누고, 그 중 하나의 512비트 연산에 대해 대리서버를 이용함으로서 무선 단말 사용자의 Diffie-Hellman 지수연산 부담을 감소시킨다.

하지만 SEKAP에는 재전송공격, 알려지지 않은 키 공유 공격, 그리고 키 노출로 인한 위장공격 등에 대한 보안 취약점이 존재하며, 전방향 안전성을 제공하지 못한다. 본 논문에서는 SEKAP의 보안 취약점을 보이고, 세션키의 상호인증을 추가한 무선 네트워크 환경에서 효율적이고 안전한 보안성을 제공하는 키 합의 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구로 SEKAP를 살펴보고, 3장에서 SEKAP의 보안 취약점을 분석하며, 4장에서 향상된 키

* 본 연구는 한국과학재단 특정기초연구(R01-2006-000-10614-0)지원으로 수행되었음.
† 주저자 : eobiya@infosec.knu.ac.kr
‡ 교신저자 : yook@knu.ac.kr

합의 프로토콜을 제안하고, 5장에서 제안한 프로토콜의 안전성을 분석한다. 마지막으로 6장에서 결론을 맺는다.

II. 관련 연구

이 장에서는 본 논문에서 사용할 용어들을 정의하고, Yang등이 제안한 SEKAP를 소개한다.

2.1 용어 정의

- A : 모바일 사용자
- B : 웹 서버
- PA : 대리 서버
- p : 큰 소수 ($2^{511} < p < 2^{512}$)
- g : 곱셈군 Z_p^*의 생성자
- $x_A, x_B / y_A, y_B$: A와 B의 비밀키/공개키 ($y_A = g^{x_A} \bmod p$, $y_B = g^{x_B} \bmod p$)
- x_{AP} : A가 PA에게 전달하는 위임키
- K_{AP} : A와 PA사이에 공유된 비밀키
- C : 세션키 생성을 위한 공유비밀값
- $SKDF(\cdot)$: 세션키 유도 함수
- SK : A와 B사이에 공유된 세션키
- Y_A / Y_B : 전방향 안전성을 제공하기 위해 A/B에 의해 생성되는 값
- N_P : PA가 생성한 난수
- $ReqSK$: 세션키 생성 요청메시지
- $E(\cdot)/D(\cdot)$: 대칭키 방식 암호화/복호화
- $Sig(\cdot)$: RSA방식 전자서명기법
- $H(\cdot)$: 암호학적 해쉬함수

2.2 SEKAP

SEKAP는 위임키 생성 및 전송 단계와 키 합의 단계로 구성되며, 키 합의 단계에서 사용하는 대리서명(Proxy Signature)기법은 Mambo [4]의 기법을 사용한다.

위임키 생성 및 전송 단계:

(1) $A \rightarrow PA$: x_{AP}, K

- A는 랜덤수 $r \in Z_{p-1} \backslash \{0\}$을 선택하고, 아래와 같이 K와 x_{AP}를 계산하여, PA에게 안전한 채널을 통해 전송한다.

$$K = g^r \bmod p$$

$$x_{AP} = x_A + r \cdot K \bmod p - 1$$

- PA는 아래 수식을 검증하여 만족할 경우 x_{AP}를 승인한다.

$$g^{x_{AP}} \stackrel{?}{=} y_A \cdot K^K \bmod p$$

키 합의 단계:

(1) $A \rightarrow PA$: $ReqSK$

- A는 PA에게 세션키 생성 요청메시지를 전송한다.

(2) $PA \rightarrow B$: y_{AP}, K, t_P, r, s

- PA는 두 랜덤수 $r_P, x \in Z_{p-1} \backslash \{0\}$를 선택하고, 아래와 같이 t_P, r, s를 계산하여 $\{y_{AP}, K, t_P, r, s\}$를 B에게 전송한다.

$$t_P = g^{r_P} \bmod p$$

$$r = H(g^x \bmod p, t_P)$$

$$s = x / (r + x_{AP}) \bmod p - 1$$

(3) $B \rightarrow PA$: y_B, t_B

- B는 아래의 수식을 검증하여, A의 요청으로 대리서버 PA가 세션키 생성을 요청하는 것을 확인한다.

$$y_{AP} \stackrel{?}{=} y_A \cdot K^K \bmod p$$

- B는 아래 두 연산을 통해 대리서버 PA가 올바른 대리서버인지 확인한다.

$$r' = (y_{AP} \cdot g^r)^s$$

$$r \stackrel{?}{=} H(r', t_P)$$

- 계산한 r이 $H(r', t_P)$와 동일할 경우, B는 랜덤수 $r_B \in Z_{p-1} \backslash \{0\}$를 선택하고, 아래와 같이 t_B를 계산하여, B의 공개키 y_B와 함께 PA에게 전송한다.

$$t_B = g^{r_B} \bmod p$$

- B는 아래와 같이 공유비밀값 C를 계산한다.

$$C = y_A^{x_B} \| t_p^{r_B} = g^{x_A x_B} \| g^{r_B r_P} \bmod p$$

(4) $PA \rightarrow A$: $E_{K_{AP}}(C')$

- PA는 t_B와 자신의 랜덤수 r_P를 이용하여 아래와 같이 C'를 계산한다.

$$C' = t_B^{r_P} \bmod p$$

· PA는 C'를 K_{AP}로 대칭키 암호화하여 $E_{K_{AP}}(C')$를 A에게 전송한다.

· A는 $E_{K_{AP}}(C')$를 복호화하고, 공유비밀값 C를 계산한다.

$$C = y_B^{x_A} \| C' = g^{x_A x_B} \| g^{r_B r_P} \bmod p$$

(5) A, B: 세션키 생성

· 공유비밀값 C를 이용하여 아래와 같이 세션키를 생성한다.

$$SK = SKDF(C)$$

III. SEKAP 보안 취약점

이 장에서는 SEKAP의 보안 취약점들을 분석한다.

(1) 재전송 공격

공격자가 PA에서 B로 전송되는 메시지 $\{y_{AP}, K, t_P, r, s\}$를 가로채고, 이전 세션에 전송된 메시지 $\{y_{AP}', K', t_P', r', s'\}$를 B에게 전송하면, B는 잘못된 세션키를 공유하게 된다. 또한 공격자가 PA에서 A로 전송되는 메시지 $\{E_{K_{AP}}(C')\}$를 가로채고, 이전 세션에 전송된 메시지를 B에게 전송하면, A는 이전 세션키를 생성하게 되어, B와 올바른 통신을 할 수 없다. 따라서 SEKAP는 재전송 공격에 취약하다.

(2) 알려지지 않은 키 공유 공격

공격자가 B에서 PA로 전송되는 메시지 (y_B, t_B)를 가로채고, t_B를 임의로 선택한 값으로 변경할 경우, A는 잘못된 세션키를 공유하게 되어, B와 올바른 통신을 할 수 없다. 따라서 SEKAP는 알려지지 않은 키 공유 공격에 취약하다.

(3) 키 노출로 인한 위장공격

A의 비밀키 x_A가 노출 되었을 때, 공격자는 B로 위장할 수 있으며, A와 정상적인 세션키를 생성할 수 있다. 왜냐하면 B에서 PA로 전송되는 메시지 $\{y_B, t_B\}$에는 B의 신원에 대한 아무런 인증이 없기 때문에, 공격자가 임의의 $r_B' \in Z_{p-1} \backslash \{0\}$를 선택하여 아래와 같이 t_B'를 계산하여 메시지 $\{y_B, t_B'\}$를 B에게 전송하더라도, PA는 아무런 의심 없이 다음 단계의 절차를 수행하게 된다.

$$t_B' = g^{r_B'} \bmod p$$

결국 공격자는 아래와 같이 공유비밀값 C를 계산하고, A와 동일한 세션키 SK를 공유하게 된다.

$$C = y_B^{x_A} \| t_p^{r_B} \bmod p$$

따라서 SEKAP는 키 노출로 인한 위장공격에 취약하다.

(4) 전방향 안전성

비밀키 x_A와 K_{AP}가 노출 되었을때, 공격자는 현재 및 이전의 모든 세션키를 계산할 수 있다. 즉, 비밀키 K_{AP}를 통해 C'를 구하고, C'와 x_A, y_B를 이용하여 아래와 같이 공유비밀값 C를 계산할 수 있다.

$$C = y_B^{x_A} \| C' \bmod p$$

따라서 SEKAP는 전방향 안전성을 제공하지 못한다.

IV. 제안하는 키 합의 프로토콜

본 논문에서 제안하는 키 합의 프로토콜은, SEKAP를 기반으로 재전송공격, 알려지지 않은 키 공유 공격, 그리고 키 노출로 인한 위장공격 등을 막기 위해 난수와 전자서명기법, 그리고 한 개의 모듈러 지수연산을 추가로 사용한다. 또한 K_{AP}를 이용한 대칭키 암호화를 x_{AP}를 이용한 XOR연산으로 대체하여 장기(long-term) 비밀키를 하나 제거하고, 세션키의 상호인증을 위해 한 라운드를 추가하여 보안성을 강화한다.

위임키 생성 및 전송 단계:

제안하는 프로토콜의 위임키 생성 및 전송 단계는 SEKAP와 동일하다.

키 합의 단계:

(1) $A \rightarrow PA$: $ReqSK$, Y_A

· A는 랜덤수 $r_A \in Z_{p-1} \backslash \{0\}$를 선택하여 아래와 같이 Y_A를 계산하고, 세션키 생

성 요청메시지 $ReqSK$와 함께 PA에게 전송한다.

$$Y_A = y_A^{r_A} \bmod p$$

(2) $PA \rightarrow B$: $y_{AP}, K, t_P, r, s, N_P, Y_A$

· PA는 난수 N_P를 생성하고, 두 랜덤수 r_P, $x \in Z_{p-1} \backslash \{0\}$를 선택하여, 아래와 같이 t_P, r, s를 계산하여,

$$t_P = g^{r_P} \bmod p$$

$$r = H(g^x \bmod p, t_P, N_P)$$

$$s = x/(r + x_{AP}) \bmod p - 1$$

$\{y_{AP}, K, t_P, r, s, N_P, Y_A\}$를 B에게 전송한다.

(3) $B \rightarrow PA$: Y_B, H, t_B, S

· B는 아래의 수식을 검증하여, A의 요청으로 대리서버 PA가 세션키 생성을 요청하는 것을 확인한다.

$$y_{AP} \stackrel{?}{=} y_A \cdot K^K \bmod p$$

· B는 아래 두 연산을 통해 대리서버 PA가 올바른 대리서버인지 확인한다.

$$r' = (y_{AP} \cdot g^r)^s$$

$$r \stackrel{?}{=} H(r', t_P, N_P)$$

· 계산결과 r이 $H(r', t_P, N_P)$와 동일하면, B는 랜덤수 $r_B \in Z_{p-1} \backslash \{0\}$를 선택하고, 아래와 같이 t_B와 Y_B, C를 계산한다.

$$t_B = g^{r_B} \bmod p$$

$$Y_B = y_B^{r_B} \bmod p$$

$$C = Y_A^{x_B r_B} \| t_p^{r_B} = g^{x_A x_B r_A r_B} \| g^{r_B r_P} \bmod p$$

· B는 A가 B의 세션키를 확인하는데 사용하는 $H(H = H(C))$를 계산한다.

· 다음으로 B는 전자서명 값 S를 아래와 같이 계산한다.

$$S = Sig_{x_B}(t_B, t_P)$$

· B는 생성한 t_B와 S를 Y_B, H와 함께 PA에게 전송한다.

(4) $PA \rightarrow A$: $x_{AP} \oplus C'$, Y_B, H

· PA는 B로부터 받은 전자서명 값 S를 이용하여 t_B의 위조여부와 t_P가 B에게 정확히 전달되었는지 확인한다.

· PA는 r_P와 t_B를 이용하여 C'를 구한다.

$$C' = t_B^{r_P} \bmod p$$

· 다음으로 PA는 위임키 x_{AP}와 C'를 XOR하여, B로부터 받은 Y_B, H를 함께 A에게 전송한다.

(5) $A \rightarrow B$: $E_{SK}(H(C))$

· A는 아래와 같이 C'와 C, SK를 계산한다.

$$C' = (x_{AP} \oplus C') \oplus x_{AP}$$

$$C = Y_B^{x_A r_A} \| C' = g^{x_A x_B r_A r_B} \| g^{r_A r_B} \bmod p$$

$$SK = SKDF(C)$$

· A는 아래의 수식을 통해 B가 정확히 세션키를 생성하였는지 확인한다.

$$H \stackrel{?}{=} H(C)$$

· A는 $H(C)$를 세션키 SK로 암호화하여 $\{E_{SK}(H(C))\}$를 B에게 전송한다.

· 마지막으로 B는 자신이 생성한 SK로 $\{E_{SK}(H(C))\}$를 복호화하여 $H(C)$를 확인하고, A가 정확한 SK를 생성하였음을 확인하면, 세션키의 상호인증은 완료된다.

V. 안전성 분석

이 장에서는 제안한 프로토콜의 안전성을 분석한다.

(1) 재전송 공격

공격자가 PA에서 B로 전송하는 메시지 $\{y_{AP}, K, t_P, r, s, N_P, Y_A\}$를 이전 세션에 전송된 것으로 변경하더라도, B는 N_P를 통해 메시지의 재전송 유무를 확인할 수 있다. 또한 공격자가 N_P를 임의로 선택한 N_P'로 변경하더라도, B의 아래와 같은 r과 s 검증연산을 통해 N_P의 위조여부를 확인할 수 있기 때문에 재전송공격에 대해 안전하다.

$$r' = (y_{AP} \cdot g^r)^s$$

$$r \stackrel{?}{=} H(r', t_P, N_P)$$

또한 공격자가 PA에서 A로 전송되는 메시지$\{x_{AP} \oplus C', H\}$를 가로채고, 이전 세션에 전송된 메시지를 A에게 전송하더라도, 공유

비밀값 계산에서 랜덤수 r_A가 포함되기 때문에, 아래와 같은 H의 검증결과를 통해 잘못된 세션키를 생성하지 않는다.

$$H \stackrel{?}{=} H(C)$$

따라서 제안하는 프로토콜은 재전송 공격에 안전하다.

(2) 알려지지 않은 키 공유 공격

B에서 PA로의 전송메시지 $\{Y_B, H, t_B, S\}$에는 전자서명기법이 사용되었기 때문에, 공격자가 t_B를 위조할 수 없으며, 알려지지 않은 키 공유 공격을 할 수 없다. 따라서 제안하는 프로토콜은 알려지지 않은 키 공유 공격에 안전하다.

(3) 키 노출로 인한 위장공격

A의 비밀키인 x_A가 노출 되더라도, 공격자는 B로 위장할 수 없다. 이유는 B가 PA에게 보내는 메시지 $\{Y_B, H, t_B, S\}$에는 전자서명기법이 사용되어, 공격자에 의해 t_B의 위조가 불가능하기 때문이다. 또한 공격자는 비밀키 x_A를 알더라도, 랜덤수 r_A와 r_B를 알 수 없기 때문에 세션키를 계산할 수 없다. 따라서 제안하는 프로토콜은 키 노출로 인한 위장공격에 안전하다.

(4) 전방향 안전성

장기 비밀키 x_A와 x_B, x_{AP}가 모두 노출되어도, 공격자는 현재 및 이전의 모든 세션키를 계산할 수 없다. 이유는 비밀키 x_{AP}를 통해 C'를 구하고, x_A, x_B를 알고 있다 하더라도, 아래와 같이 공유비밀값 C의 계산에는 일시적인(ephemeral) 키인 랜덤수 r_A와 r_B가 필요하기 때문이다.

$$C = Y_A^{x_B r_B} \| C' = Y_B^{x_A r_A} \| C' \bmod p$$

따라서 제안하는 프로토콜은 전방향 안전성을 제공한다.

VI. 결론 및 향후과제

본 논문에서는 2005년에 Yang등이 제안한 SEKAP의 보안 취약점들을 분석하고, 이를 개선한 키 합의 프로토콜을 제안하였다. 제안한 프로토콜은 재전송공격, 알려지지 않은 키 공유 공격, 그리고 키 노출로 인한 위장공격 등에 안전하고 전방향 안전성을 제공하며, 세션키의 상호인증을 통해 보안성을 강화하였다.

본 논문에서는 전방향 안전성을 제공하기 위해 모듈러 지수연산을 하나 추가하였는데, 이것은 무선 네트워크 환경에서 고비용의 연산이다. 따라서 향후과제로 전방향 안전성을 제공하기 위한 저비용의 효율적인 기법에 관한 연구가 필요하다.

[참고문헌]

[1] H.K.Yang, Y.H.An and J.H.CHoi, Secure and Efficient Key Agreement Protocols for Wireless Communication, Asia-Pacific Conference on Communications, Perth, Western Australia, pp.520-524, 3-5 October 2005.

[2] ANSI X9.42, Agreement of Symmetric Key on Using Diffie-Hellman Cryptography, 2001.

[3] W.Diffie and M.E.Hellman, New Directions in Cryptography, IEEE Transaction of Information Theory, IT-22, 6, pp.644-654, 1976.

[4] M.Mambo, K.Usuda and E.Okamoto, Proxy Signatures for Delegating Signing Operation, Proc. Third ACM Conference on Computer and Communications Security, pp.48-57, 1996.