2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

공개키와 패스워드 기반의 사용자 직접 인증 프로토콜에 관한 연구

김찬오*, 최은정**, 송주석*
*연세대학교 컴퓨터과학과
**연세대학교 수학과
e-mail:captin33, jssong@emeral.yonse.ac.kr

Study on Public Key Cryptosystem and Password Based Direct Authentication Protocol for Remote User Access

Chan-oh Kim*, Eun-jeong Choi**, Joo-Seok Song*
*Dept. of Computer Science, Yonsei University
**Dept. of Mathematics, Yonsei University

요 약

신뢰할 수 없는 네트워크를 통한 패스워드 기반의 원거리 사용자 인증은 패스워드의 선택범위와 길이가 사용자의 기억력에 제한되는 낮은 안전성 때문에 오프라인 사전공격에 취약하다. 본 논문은 이산대수 문제 해결의 어려움에 기반한 Diffie-Hellman 키 교환과 블록암호화 알고리듬 및 MAC을 이용하여 패스워드 기반 인증 및 키 협상 프로토콜을 제안한다. 제안된 프로토콜은 오프라인 사전공격을 예방할 수 있으며, 세션키와 패스워드 검증정보가 독립적이므로 공격자에게 패스워드가 노출되더라도 이전 세션의 복호화에 영향을 미치지 않는 전향적 보안성을 제공한다. 또한 세션키의 노출이 패스워드에 대한 정보를 노출시키지 않으며, 암호화 횟수와 메시지 크기를 최소화 하여 효율성을 극대화 하였다. 따라서 웹을 통한 홈뱅킹이나, 모바일 환경이 요구되는 셀룰러 폰에서의 사용자 인증처럼 제3의 신뢰기관을 이용하지 않는 단순 직접 인증에 적합하다

1. 서론

공개 네트워크를 통한 안전한 통신을 위해서 사용자는 기밀성과 무결성이 보장된 상태에서 자신이 정당한 사용자임을 증명하는 인증절차가 필요하며, 인증은 다음과 같은 3가지 형태의 검증정보를 기반으로 이루어진다.

- 개인의 지식 (Knowledge based : 패스워드, PIN...)
- 개인의 소유 (Token based : 스마트카드, 토큰...)
- 신체적 특성 (Biometric : 지문, 홍채, 목소리...)

개인이나, 조직체에서 허용하는 신뢰와 안정성 수준에 따라 네트워크 보안의 강·약 레벨은 다양하게 구현될 수 있으나, 2가지 형태 이상의 인증 정보가 결합되어 강한 인증 서비스가 제공되어야 한다. 패스워드와 같은 지식기반 인증은 단순성, 편리성, 이동성의 장점 때문에 광범위하게 사용된다. 사용자 인증을 위한 패스워드 기반 인증 프로토콜에서 사용자는 패스워드를 이용하여 사용자 인증과 세션 암호화를 위한 키를 분배한다. 그러나, 패스워드의 선택범위와 길이는 사용자의 기억력에 의해 제한되는 낮은 엔트로피(entropy)를 가지므로 공격자가 패스워드로 유추되는 단어들을 사전화하고, 오프라인에서 이 단어들을 차례로 대입하여 정당성을 확인하는 오프라인 사전 공격에 취약하다. [1][2]

본 논문에서는 사용자 인증과 세션키 교환을 위해 두 당사자만이 참여하는 직접 인증에서 명시적 키교환을 통하여 상호인증을 제공하는 프로토콜을 제안한다. 제안한 프로토콜은 오프라인 패스워드 유추공격을 예방할 수 있고, 공격자가 패스워드를 획득하더라도 이전 세션키의 복호화가 불가능한 전향적 보안성(forward secrecy)을 제공하고, 공격자가 패스워드 검증 파일을 획득하더라고 사용자의 패스워드를

직접 획득할 수 없으며, 정당한 사용자로 가장할 수 없다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 프로토콜의 수행과정을 설명하고, 3장에서는 이 프로토콜의 안전성 측면에서 분석하고, 4장에서는 성능면에서 분석하고, 5장에서는 결론과 향후 필요한 연구방향을 언급한다.

2. 제안하는 패스워드 인증 모델

2.1 표기법

C, S :	클라이언트(Client), 서버(Server)
P_w :	클라이언트의 패스워드
P_w' :	공격자가 추측한 패스워드
$V(P_w)$:	패스워드 검증값, $(= g^{H(P_w \| ID_C \| ID_S)})$
K :	세션키
$MAC_K[M]$:	메시지(M)의 키(K) 인증코드
$E_K[M]$:	메시지(M)의 키(K) 대칭키 암호문
$Nonce$:	반복되지 않는 임의의 난수
Y_C, Y_S :	클라이언트와 서버의 공개키
x, y :	클라이언트와 서버의 비밀키
g :	생성자, 유한체 $GF(p)$의 원시근
p :	큰 소수

2.2 가정

가. C는 P_w를 기억하고, S는 $V(P_w)$를 저장하는 검증자 기반 인증 프로토콜이다.

나.모든 연산은 $GF(p)$에서의 모듈러 연산이다.

다. C와 S는 $Nonce$를 생성하는 랜덤비트 생성기를 가지며, 공격자는 랜덤 비트 패턴과 삭제 정보를 이용하여 $Nonce$값을 유추할 수 없어야 한다.

2.3 프로토콜 묘사

[1단계]

C는 프로토콜 수행전에 P_w를 이용하여 $V(P_w)$를 계산하고, 소수 p로 구성되는 Z_p에서 $x \in_R Z_{p-1}$을 선정해서 $g^x \bmod p$를 계산하여 두 값을 $\oplus$ 연산하여 Y_C를 생성한 후 $H(g^x)$와 $Nonce_1$을 $\oplus$연산하여 S에 인증을 요청한다.

$$C \rightarrow S : ID_C \| Y_C \| H(g^x) \oplus Nonce_1$$

$$Y_C = g^x \oplus V(P_w) \bmod p \ , \ V(P_w) = g^{H(P_w \| ID_C \| ID_S)}$$

[2단계]

S는 Y_C가 $1 < Y_C < p$ 인지를 확인하고, Y_C에서 자신이 저장하는 $V(P_w)$와 $\oplus$ 연산하여 $g^x \bmod p$ 를 구한 후 $g^x \bmod p$의 해쉬값을 이용하여 $Nonce_1$을 구한다. 또한 Z_p상에서 y를 선정하여 K를 생성하여, 이를 키로 하고, C가 생성한 $Nonce_1$를 입력으로 하는 MAC을 생성하여 C에 전송한다.

$$K = (Y_C \oplus V(P_w))^y$$

$$S \rightarrow C : Y_S \| Nonce_1 \oplus Nonce_2 \| MAC_K[Nonce_1]$$

[3단계]

C는 Y_S와 x를 이용하여 K를 생성한 후 자신이 생성했던 $Nonce_1$을 이용하여 $Nonce_2$를 생성하고, $Nonce_2$를 K의 키로 하는 MAC을 이용하여 K의 무결성을 확인하여 $g^x \bmod p$가 K 생성에 정확하게 사용되었는지를 확인하고, S가 패스워드 검증자를 소유하였는지를 확인할 수 있다. 무결성이 확인된 K는 C가 기억하고 있는 패스워드와 신분을 연결한 값을 해쉬한 값과 $Nonce_2$를 $\oplus$ 연산을 수행하여 이 값을 입력으로 하는 블록암호화 알고리듬의 키로 사용되어 암호화 되어 S에 전송한다.

$$C \rightarrow S : ID_C \| E_K[H(P_w \| ID_C \| ID_S) \oplus Nonce_2]$$

S는 K를 이용하여 암호화된 메시지를 복호화하고, $Nonce_2$로서 $H(P_W \| ID_C \| ID_S)$를 구하고, 이 값을 지수로 하는 법 연산을 수행한 결과가 패스워드의 검증자 $V(P_w)$와 일치할 경우 상호 인증 프로토콜이 완료된다.

3. 제안한 인증 모델의 안전성 분석

이 프로토콜의 안전성은 분산 계산 모델에서 형식적인 방법으로 증명되지 않지만, 경험적이고, 수학적 가정에서 다음과 같은 안전성을 설명할 수 있다.

3.1 오프라인 사전공격

패스워드 기반 인증 프로토콜의 가장 큰 취약점은 패스워드에 대한 사전공격이며, 따라서 인증 수행간에 패스워드에 대한 어떠한 정보도 노출되어서는 안된다. 따라서, 제안한 프로토콜에서는 검증 불가능한 난수 메시지를 암호화하기 위해 랜덤한 세션키를 사용하고자 한다.

제안된 프로토콜에서 전송되는 메시지들이 C와 S의 신분(Identification)을 제외하고 암호화 및 MAC이 수행된 난수이므로 도청등의 수동적 공격을 예방할 수 있으며, 공격자가 추측한 패스워드 P_w'에 대하여 검증할 수 있는 노출된 정보가 없으므로 능동적 공격을 예방할 수 있다. 만약 공격자가 [1단계]에서 전송되는 메시지를 도청하여 P_w'를 이용하여 Y_C'와 $g^{x'}$을 구하더라도, $Nonce_1$으로 $H(g^x)$가 마스크되어 있고, 이 $Nonce_1$은 다시 2단계에서 $Nonce_2$로 마스크된 후 세션키로 암호화 되어 있으므로 사전공격이 불가능하다. 또한 [3단계]에서 C가 블록 암호화 알고리듬의 세션키를 생성할 때 $g^x \bmod p$의 무결성을 재 확인하므로 사전공격을 탐지할 수 있다.

3.2 Denning-Sacco 공격에 대한 안전성

공격자가 이전 세션키를 획득하고 하더라도 매 세션마다 C, S가 x, y를 랜덤하게 선택하고, 세션키의 값이 매 세션마다 상이하므로 x, y를 모를 경우에 $K = g^{xy}$를 구할 수 없다. 이는 이산대수 문제 해결의 어려움에 기반한다. 즉 이전 세션키들 성립과 이후 세션키 성립이 패스워드 정보와 독립적으로 결정되므로 이후 세션키에 대한 정보를 획득할 수 없으며, 또한 세션키의 노출은 패스워드에 대한 정보도 노출시키지 않는다.

3.3 전향적 보안성 (Forward Secrecy)

C, S 사이에 공유하는 패스워드가 노출되더라도 이전에 사용된 세션키를 공격자는 알 수 없다. 즉 공격자가 도청을 통해 공개정보 Y_C, Y_S를 획득하고, 현재 C의 P_w를 획득하더라도, x, y가 패스워드와는 독립적인 정수값이므로 이전 세션키를 구할 수 없다. 이러한 전향적 보안성을 제공하기 위해 패스워드 기반의 세션키 교환에 $Diffie-Hellman$ 키 협상 프로토콜이 사용된다.

3.4 서버 손상시 안전성 (Server Compromise)

제안한 프로토콜은 검증자(Verifier) 기반인증을 제공하므로 서버가 트로이 목마와 같은 악성 프로그램에 감염되어 패스워드 파일의 노출시 프로토콜의 안전성 손상을 최소화 할 수 있다. 서버의 검증자 정보 $V(P_w)$를 획득한 공격자는 검증자 정보와 $Nonce$를 생성하므로서 정당한 C로 위장을 시도하려 할 것이다. 만약 [3단계]에서 세션키가 메시지를 복호화하여 $Nonce$가 일치하고, $H(P_W \| ID_C \| ID_S)$를 이용한 법 연산의 값이 일치하지 않는다면, 패스워드의 검증자 값이 노출되었음을 확인할 수 있다. 또한 검증자 정보가 노출되더라도 공격자는 C의 정확한 패스워드를 알아내기 위하여 P_w'를 이용한 오프라인 사전공격이 수행되어야 한다.

3.5 명시적 키 인증성(Explicit Key Authentication)

원거리 사용자 인증에서 중간자 공격은 공격자가 사용자에게는 서버처럼 서버에게는 사용자처럼 위장하는 공격이다[4]. 이러한 공격에 대한 대응책으로는 간접인증에서는 PKI 인증서가 사용될 수 있고, 직접인증에서는 인터락 프로토콜과 전자 서명된 키를 교환하는 방법 등이 있으나, 가장 손쉽게 상호인증을 통해 예방할 수 있다[5]. 상호인증이란 두 당사자들이 서로 간에 패스워드를 알고 있는지를 확인하는 과정이다. 1,2단계에서 C, S 자신이 생성하여 전송한 $Nonce$ 값이 검증 절차를 거쳐 정확하게 응답($Response$) 됨을 확인하므로서 명시적 키 교환을 통한 상호인증이 이루어진다.

4. 제안한 인증 모델의 성능 분석

4.1 메시지 교환횟수를 최소화

본 프로토콜에서 상호간에 안전하게 동일한 세션키를 소유하고 있음을 확인하기 위해 3단계의 메시지 흐름으로 구성되었으며, 메시지에 불필요한 내용이 삽입되지 않도록 하였다. 인증을 요청하는 C와 S가 정확하게 패스워드 검증자를 공유하고, 세션키를 누구도 생성할 수 없다고 확신하는 함축적 키 인증은 2단계만으로도 가능하다. 즉[2단계]가 수행된 후 C는 S가 정당한 패스워드 검증자를 소유하고 있음을 확인할 수 있다. 따라서 [3단계]는 S의 검증자 정보 노출 여부를 확인하는 과정일 뿐만 아니라 S와

C가 상호간에 정확하게 세션키를 생성했음을 확인하는 과정이다.

4.2 전송 비트 수를 최소화

프로토콜의 각 단계에 사용된 해쉬 함수, MAC, $\oplus$ 연산은 무결성 및 메시지 마스크 뿐만 아니라 전송되는 비트 수를 최소화한다. Diffie-hellman 키 교환은 모듈러 p의 크기와 특징에 의존하며, $m=[\log_2 p]$일 경우 m의 값이 충분히 큰 소수 ($1024bit$이상)일 경우 이산대수 문제 해결이 거의 불가능하다. $GF(p)$에서의 이산대수 계산의 복잡성을 최대화하기 위해 p의 값은 $p = 2q+1$, q=소수인 안전소수(safe prime)가 권장되며, 개인키는 160bit 이상이 권장된다.

해쉬 함수는 메세지 다이제스트(message digest)라고도 불리며, 입력 데이터의 길이에 관계없이 고정된 출력 데이터의 길이(128bit, 160bit)를 생성하며, 해쉬함수의 안전성은 주어진 출력데이터에 대하여 입력 데이터를 찾아내는 것이 계산적으로 불가능한 일방향성과 동일한 출력을 가지는 서로 다른 입력쌍을 찾는 것이 계산상으로 불가능한 충돌회피성에 의존한다. MAC(Message authentication code)은 키 종속적인 일방향 해쉬 함수이며 개체간에 메시지의 변조 여부를 확인하기 위해 사용된다. 일 방향 해쉬 함수를 MAC으로 바꾸는 방법은 해쉬 값을 대칭 알고리즘으로 암호화 하는 것이다.

4.3 산술적 계산 및 암호화 횟수의 최소화

프로토콜에서 임의의 난수 $\oplus$ 연산은 블록 암호화 연산보다 수행시간에서 효율적이며, $\oplus$ 연산은 비밀정보에 대한 길이만을 공격자에게 제공한다. C는 비밀키 x를 선택하여 $g^x \bmod p$ 구하는 연산, Y_C 생성을 위한 $\oplus$ 연산, $g^x \bmod p$의 해쉬연산, 세션키 생성을 위한 $(Y_S)^x \bmod p$ 연산, 블록 암호화 1회가 사용되며, S측에서는 비밀키 y를 선택하여 $g^y \bmod p$ 연산, Y_C를 구하기 위한 $\oplus$ 연산 및 $H(g^x)$의 무결성 확인을 위한 $g^x \bmod p$의 해쉬연산, 세션키 생성을 위한 $(Y_C)^y \bmod p$ 연산 및 MAC 연산, 블록 암호화 알고리듬의 복호화가 사용된다. 실제 MAC 연산과 해쉬연산, $\oplus$ 연산은 효율적인 계산시간을 제공하므로 계산량에 크게 영향을 미치지 않으며, 블록 암호화 알고리듬의 입력값 및 출력값 또한 해쉬값이 $\oplus$ 연산을 수행하므로 입력값이 160bit를 초과하지 않고, $H(P_W \| ID_C \| ID_S)$는 C측에서 사전에 계산 가능하다.

5. 결론 및 향후과제

본 논문에서는 $Diffie-Hellman$ 키 협상 프로토콜에 기반하여 인증 수행간에 패스워드에 대한 정보를 노출시키지 않으며, 공격자의 오프라인 사전공격을 예방할 수 있는 새로운 인증 프로토콜을 제안하였으며, MAC과 $\oplus$ 연산을 이용하였다. 이 프로토콜은 중간자 공격이나 패스워드의 손상과 같은 내부자 공격에 취약성 등을 보완할 수 있으며, 전자인증서 방식의 단점인 개인키 보관의 어려움, 처리 속도의 문제, PKI 연동의 복잡성 등을 극복할 수 있다. 제안하는 프로토콜은 TTP와 같은 신뢰기관을 이용할 수 없는 휴대폰과 같은 소형장치에서 직접 단순 인증에 적합하기 위해 프로토콜 흐름 단계, 메시지 크기 및 암호화 횟수를 최소화하며, 타원곡선 암호를 이용하여 효율적인 수행속도 개선이 가능하다.

추가적인 연구 방향으로는 $B-R$ 모델에서 형식화된 수행속도 비교 및 안전성 검증이 이루어져야 한다.

참고문헌

[1] Thomas Wu. "The Secure Remote Password Protocol". 1999 Internet Society NDSS, Mar.,1998

[2]David Jablon, "Public Key Methods for Shared Secret Authentication", RSA'98,January 14, 1998

[3]Bellovin&Merritt, "Encrypted Key Exchange", I.E.E.E on Security and Privacy, May ,1992.

[4]Halevi, "public-key cryptography & password protocols", ACM Trans. on ISS, Vol.2, Aug.,1999.

[5]Taekyoung Kwon, "Authentication and key agreement via memorable password", IEEE, P1363, Aug. 20, 2000