ElGamal 기반의 새로운 패스워드-인증 키 교환 프로토콜

심현정, 김락현, 염흥열

순천향대학교 정보보호학과

New Password-Authenticated Key Exchange Protocol based on ElGamal

Hyun-Jeung Sim, Rack-Hyun Kim, Heung-Youl Youm

Department of Information Security Soonchunhyang Univ.

요 약

본 논문에서는 ElGamal 암호 기반에 PAK(Password-Authenticated Key Exchange) 프로토콜을 적용하여, 새로운 패스워드-인증 키 분배 프로토콜을 제안하고자 한다. ElGamal 암호 기법에서 사용자와 서버의 공개키와 개인키로 미리 공유된 패스워드와 비밀 정보를 암호문으로 전송하고 복호함으로써 서버-클라이언트 상호 인증을 하고, 비밀 정보를 근거로 세션키를 나누어 갖는 것이 이 프로토콜의 목적이다. 또한 설계 시 패스워드-인증 키 교환 프로토콜에서의 보안 요구사항을 고려하여 패스워드 기반의 인증 키 분배 프로토콜에서 요구하는 보안 요구사항을 만족하고 있다.

I. 서론

패스워드는 일반 사용자가 암기하기 쉽다는 장점 때문에 클라이언트-서버 구조에서 가장 많이 이용되는 사용자 인증방법이다. 또한 패스워드를 사용한 패스워드-인증 키 분배 프로토콜은 사용자가 단지 패스워드만을 사용하여 키 교환을 수행할 수 있기 때문에 많은 분야에 적용되고 있다. 그러나 패스워드 인증절차, 즉 패스워드-인증 키 분배 프로토콜들은 여러 가지 편리한 점이 있으나 반면에 몇 가지 문제를 가지고 있다. 우선 클라이언트가 서버에게 패스워드를 노출하지 않고 임의의 방법으로 패스워드를 알고 있음을 서버에게 반드시 증명하여야 한다. 또한, 두 참여자의 통신 정보가 공격자에게 노출된다면 다양한 공격 방법을 통하여 패스워드 추측 공격을 당할 수 있게 되므로 안전성을 강화하기 위해 클라이언트와 서버에 추가적인 매개변수가 삽입된 프로토콜을 사용해야 한다.[1]

패스워드를 사용하는 암호 시스템은 오프라인 사전공격(offline dictionary attack)에 취약점을 가지고 있다. 이와 같은 패스워드 추측공격을 피하기 위하여 Bellovin 과 Merrit는 비대칭암호 방식과 결합하여 사용하는 패스워드-인증 키 분배 프로토콜 제안하였다. [2]

본 논문에서는 우선 패스워드-인증 키 분배 프토콜과 기존의 RSA와 ElGamal 암호알고리듬에 대하여 분석하고, 이전에 제시되었던 SNAPI(Secure Network Authentication With Password Information)에 대하여 살펴본다.[3] 그리고 ElGamal에 기반을 둔 새로운 PAK프로토콜을 제안하고자 한다.

II. 기존 프로토콜 분석

1. PAK 프로토콜

1) PAK에서의 파라미터 정의

- g : 원시근
- p : 1024 비트 소수, q = 160 비트 소수
- $p = rq+1$, $\gcd(r, q) = 1$
- H_1, H_{2a}, H_{2b}, H_3= 랜덤한 해쉬 함수

- H_1= output 1024+160 = 1184 비트
- $H_{2a}.H_{2b}.H_3$ =output 160비트
- π = 사용자 A,B의 공유 패스워드,
- K = 세션키
- A,B = 사용자 A,B 의 ID

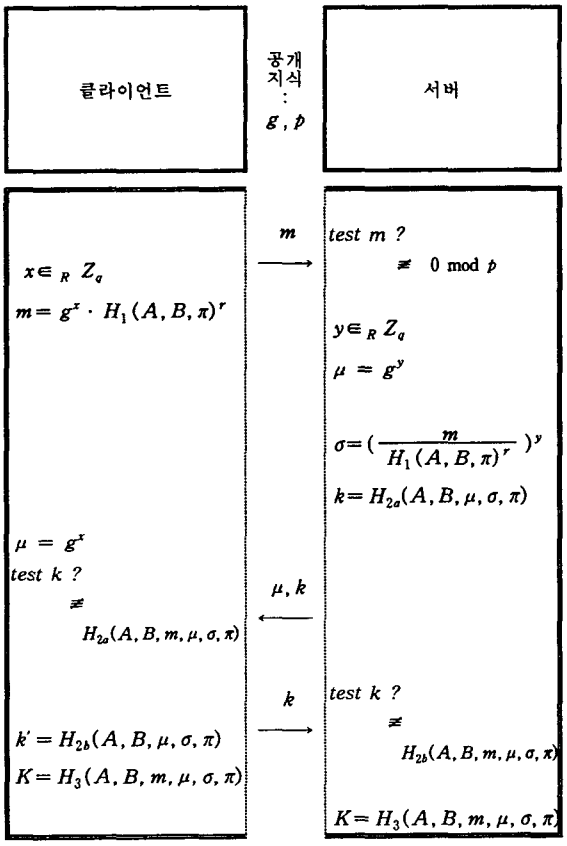


그림 1. PAK 프로토콜

클라이언트는 랜덤한 메시지(m)를 선택하여 서버에게 전송한다.[4] 서버는 랜덤 메시지 m 을 검사한 후, σ 를 계산한다. 이때 σ 는 상호 인증을 위한 파라미터이다. 서버는 μ 값을 계산하고 공유 정보를 해쉬하여 클라이언트에게 전송한다. 클라이언트는 수신한 μ 를 이용하여 σ 를 계산한 후, 수신한 해쉬값과 자신이 계산한 해쉬값을 비교하여 동일하면 공유 정보를 또 다른 해쉬 함수를 이용하여 계산, 그 값을 서버로 전송한다. 서버는 상호 인증의 마지막 단계인 해쉬값 비교를 하고 조건이 만족하면 서버와 클라이언트는 세션키 K 를 분배한다. 매 세션마다 서버와 클라이언트에서 선택하는 비밀 정보들을 랜덤하게 선택하므로 세션마다 새로운 세션키가 분배된다.

2. RSA 프로토콜

1) RSA 파라미터 정의

- p,q : 소수.
- n,e : 공개 암호화키, d :비밀 복호화키

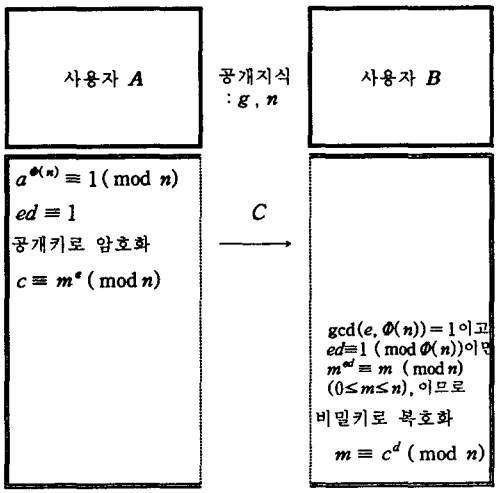


그림 2. RSA 프로토콜

RSA는큰 소수의 소인수 분해의 어려움에서 안전성을 얻는다.[5] 공개키와 비밀키는 한 쌍의 큰 소수이다. 메시지를 암/복호화를 위한 초기화 단계는 다음과 같다.

단계 1 : 두개의 큰소수 p 와 q 를 선정하여 합성수 $n=pq$ 를 법으로 한다.

단계 2 : n 를 공개하고, 서로소인 임의의 e 를 선택하여 공개키로 사용한다. n 이 두 소수의 곱일때 $\phi(n)=(p-1)(q-1)$ 이다.

단계 3 : $ed \equiv 1 (\mathrm{mod}\ \Phi(n))$ 이 되는 d 를 계산하여 비밀키로 사용한다. 사용자 A 는 메시지를 B 의 공개키로 암호화한다. 메시지 m 은 e 의 역원인 d 로만 복호가 가능하다. 공격자가 암호문을 도청하였다 할지라도 사용자 B 외에는 메시지를 알 수가 없어 기밀성 보장과 암호문을 사용자 A 가 보냈다는 것을 알 수 있는 사용자인증이 이루어진다.

3. ElGamal 프로토콜

1) 파라미터 정의

- g :원시근, p : 소수
- x : 사용자 A 의 개인키, y : 공개키
- k : $p-1$ 과 서로소인 랜덤한 값 선택
- m : 메시지

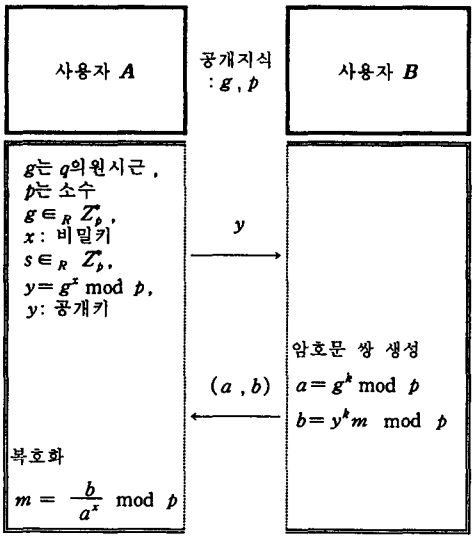


그림 3. ElGamal 프로토콜

이 암호 방식은 이산대수의 어려움으로부터 안전성을 얻는다. 키 쌍을 생성하기 위해서 우선 소수 p 를 선택하고 p 보다 작은 두개의 랜덤수 g 와 a 를 선택하여 사용한다. [6]

사용자 A 가 공개키를 보내주면 그 정보를 가지로 사용자 B 는 메시지 m 을 암호하여 암호문 쌍으로 보내준다. 사용자 A 는 암호문을 받기 전에 자신이 생성한 공개키 정보를 가지고 사용자 B 가 암호하여 보내준 것이므로 공격자가 도청을 하여도 오직 사용자 A 만이 메시지를 복호화 할 수 있다. 이로써 메시지의 기밀성과 사용자 암호문을 B 가 보냈다는 것을 알 수 있는 B 사용자 인증이 이루어진다.

4. SNAPI 프로토콜

1) SNAPI 파라미터 정의

- A : 사용자 A (또는 클라이언트)
- B : 사용자 B (또는 서버)
- π : 사용자 A , B 의 공유 패스워드
- $((e,N),(d,N))$: RSA 키 쌍
- $h,h',h'':\{0,1\}^*\rightarrow\{0,1\}^k$, $H:\{0,1\}^*\rightarrow\{0,1\}^\eta$:
- 해쉬 함수
- 조건 : $(\eta \geq l+k)$,
- h,h',h'',H : 독립적인 랜덤 함수로 가정
- $S_N=\{p: p\leq 2^\eta-(2^\eta-N) \text{and } \gcd(p,N)=1\}$
  - p는 1024 비트의 소수
  - q는 160 비트의 소수
  - g는 Z_p^*의 서브그룹의 생성자, q의 원시근
- K : 공유된 세션키

SNAPI 프로토콜은 RSA 기반에서 PAK 프로토콜을 적용한 방식이다. 클라이언트의 패스워드 정보를 이용하여 RSA 암호 방식을 통해 상호 인증과 서버 클라이언트간의 비밀 정보를 공유하고 이를 근거로 세션키를 분배하는 프로토콜이다. 다음은 SNAPI 프로토콜의 흐름을 설명한 것이다. [3]

Step 1. 클라이언트

- 임의의 메시지 $m \in_R \{0,1\}^k$ 를 생성
- 사용자 ID와 메시지 그리고 공개키 정보를 서버에게 전송

Step 2. 서버

- 수신된 메시지와 공개키 정보 조건 검사
- 검사에 성공하면, 인증 과정에 사용될 $\mu \in_R \{0,1\}^k$ 와 서버의 비밀 정보 $a \in_R Z_N^*$ 를 선택
- 공유정보를 해쉬한 후, 임의로 선택한 서버의 비밀 정보를 이용하여 암호화

$$p = H(N,e,m,\mu,A,B,\pi)$$

$$q \equiv pa^e \bmod N$$

- 선택한 임의의 μ 와 해쉬의 암호값 q 를 클라이언트에게 전송

Step 3. 클라이언트

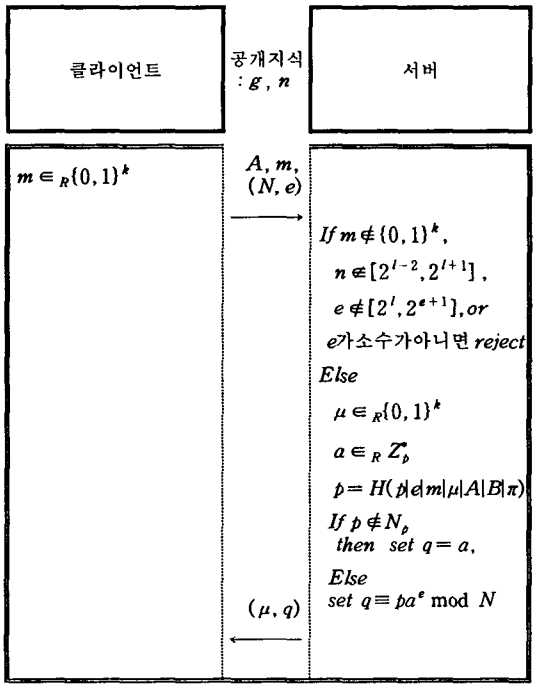
- 서버의 난수값 μ 와 q 가 조건에 만족한지 검사
- 공유 정보를 해쉬하여 암호문으로 전달된 메시지 복호
- 이때 공유 정보와 복호한 서버의 비밀 정보 a 를 포함하여 해쉬 값 r 계산
- r 을 서버에게 전송

Step 4. 서버

- 수신된 해쉬값을 검사
- 수신된 공유 정보를 또 다른 해쉬 함수를 이용하여 해쉬
- 해쉬값을 클라이언트에게 전송하고
- 공유 정보를 가지고 세션키 K 를 계산

Step 5. 클라이언트

- 수신된 해쉬값을 검사
- 최종 수신된 공유 정보를 포함한 세션키 K 를 계산.



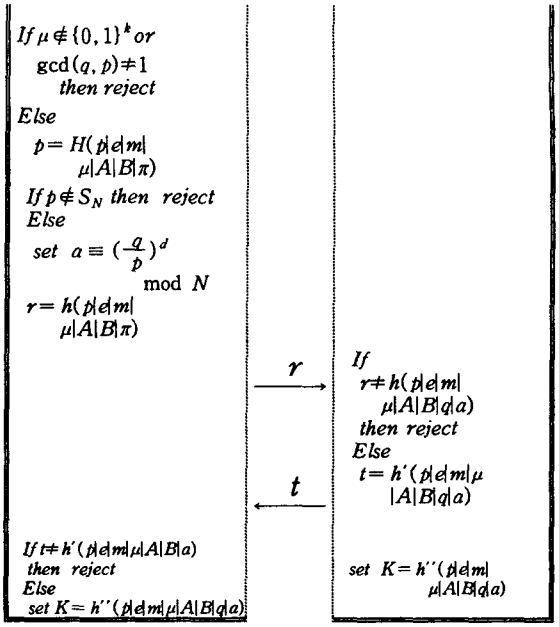


그림 4. SNAPI 프로토콜

5. ElGamal 기반 PAK 프로토콜

1) ElGamal 기반 PAK 파라미터 정의

- A : 사용자 A (또는 클라이언트)
- B : 사용자 B (또는 서버)
- π : 사용자 A , B 의 공유 패스워드
- a : 사용자 A의 비밀키
- z : p-1과 서로소인 랜덤한 값 선택
- w : 임의의 랜덤한 값. 서버의 비밀정보
- C_1, C_2 : ElGamal 암호문 키 쌍
- $h, h', h'' : \{0,1\}^* \rightarrow \{0,1\}^k$
  $H : \{0,1\}^* \rightarrow \{0,1\}^\eta$: 해쉬 함수

  조건 : $(\eta \geq l+k)$, h, h', h'', H : 랜덤 함수
- $\gcd(S, p) = 1$
  - p는 1024 비트의 소수
  - q는 160 비트의 소수
  - g는 Z_p^*의 서브그룹의 생성자, q의 원시근
- K : 공유된 세션키

2) ElGamal 기반 PAK 프로토콜

PAK+ElGamal 프로토콜은 ElGamal 기반에서 PAK 프로토콜을 적용한 방식이다. 클라이언트의 패스워드 정보를 이용하여 ElGamal 암호 방식을 통해 클라이언트와 서버간의 상호 인증과 비밀 정보를 공유하고 이를 근거로 세션키를 분배하는 프로토콜이다. 다음은 PAK+ElGamal 프로토콜의 흐름을 설명한 것이다.

Step 1. 클라이언트

- 임의의 메시지 $m \in_R \{0,1\}^k$ 를 생성
- 클라이언트는 개인키로 지수승하여 자신의 공개 정보 y를 보냄.
- 사용자 ID와 메시지 그리고 공개키 정보를 서버에게 전송

Step 2. 서버

- 수신된 메시지와 공개키 정보 조건 검사
- 검사에 성공하면, 인증 과정에 사용될 $\mu \in_R \{0,1\}^k$ 와 서버의 비밀 정보 $s \in_R Z_p^*$ 를 선택
- 공유정보를 해쉬한 후, 임의로 선택한 서버의 비밀 정보를 이용하여 암호화

$$S = H(y, m, \mu, A, B, \pi)$$

$$C_1 = g^z \bmod p$$
$$C_2 = (S \cdot y^z \cdot w) \bmod p ,$$

- 선택한 임의의 μ 와 해쉬의 암호값 C_1, C_2 를 클라이언트에게 전송

Step 3. 클라이언트

- 서버의 난수값 μ 가 조건에 만족한지 검사
- 공유 정보를 해쉬하여 암호문으로 전달된 메시지 복호
- 이때 공유 정보와 복호한 서버의 비밀 정보 w 를 포함하여 해쉬 값 r 계산
- r 을 서버에게 전송한다.

Step 4. 서버

- 수신된 해쉬값을 검사
- 수신된 공유 정보를 또 다른 해쉬 함수를 이용하여 해쉬
- 해쉬값을 클라이언트에게 전송하고 공유 정보를 가지고 세션키 K 를 계산

Step 5. 클라이언트

- 수신된 해쉬값을 검사
- 최종 수신된 공유 정보를 포함한 세션키 K 를 계산

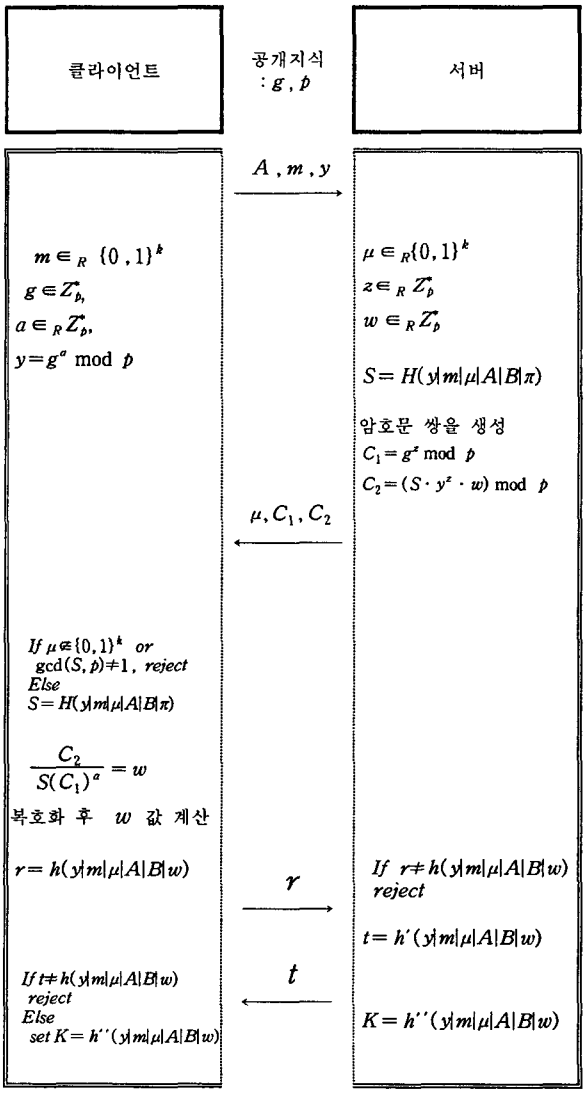


그림 5. 제안된 ElGamal 기반 PAK 프로토콜

6. 보안 요구사항 분석

1) 도청 공격

- 메시지 m 은 세션마다 임의의 난수를 근거로 생성하고, 비밀 정보는 ElGamal 암호 기법을 이용하여 보호한다.

2) 재전송 공격

- 단계별로 전송되는 메시지가 연속적으로 전송되지 않고, 인증을 위해 암호 기법과 해쉬 기법을 혼용하여 사용한다.

3) 중간자 공격

- Diffie-Hellman과 이산대수 문제의 어려움에 근거하여 설계하였다.

4) 사전 공격

- 해쉬 값에 의한 개인 비밀 정보는 암호기법으로 보호되고, 상호 인증과 키 분배를 위해 비 대칭적인 3개의 서로 다른 해쉬를 이용한다.

5) PFS(perfect forward secrecy)의 만족

- 세션키의 생성은 두 사용자의 비밀 정보와 난수를 이용하여 세션마다 갱신되므로 사용자의 패스워드가 공개되더라도 이전 세션의 세션키 값을 알수 없다

III. 결론

본 논문에서는 서버와 클라이언트 공유 패스워드와 비밀 정보들을 이용하여 상호 인증과정을 수행한 후, 세션키를 분배하는 ElGamal 기반의 PAK 프로토콜을 제안하였다. 제안한 프로토콜은 기존의 SNAPI 프로토콜의 특성과 효율성 측면의 계산량을 유지하면서 보안요구사항을 만족하는 프로토콜로서 다양한 시스템에 적용 가능한 프로토콜이다.

참고문헌

[1] M.Bellare and P.Rogaway, "The AuthA Protocol for Password-Based Autheticated Key Exchange", Comtribution to the IEEE P1363 study group, March 14, 2000

[2] S.M.Bellovin and M.Merritt. "Encrypted key exchange : Password-based protocols secure against dictionary attacks." In Proceedings of the IEEE Symposium on Research in Security and Privacy, pages 72-84, 1992

[3] Philip.MacKenzie and Sarvar Patel and Ram Swaminathan, "Password-Authenticated Key Exchange based RSA" , 2000

[4] V.Boyko, P.MacKenzie, and S.Patel. "Provably-secure password authentication and key exchange using Diffie-Hellman". In EUROCRYPT2000[EUR00]

[5] 이만영, 송유진, 김지홍, 염홍열, "전자상거래보안 기술". 생능출판사, 1999.

[6] Bruce Schneier, "Applied Cryptography" , p476-477 John Willey & Sons, Inc.second edition, 1992.