通信情報保護學會學會誌
第 9 券 第 4 號, 1999. 12

패스워드를 이용한 인증 프로토콜들에 대한 고찰

박왕석*, 정종필*, 박창섭**,이동훈***

요 약

클라이언트 측의 사용자들이 응용서버(application server)로부터의 서비스를 받기위해서는 사용자간에 또는 사용자와 암호시스템 간에 인증은 필수적인 요구 조건이다. 이러한 인증과정은 현재 사용자가 통신을 통해 정보를 교환하고 있는 상대가 실제 의도한 상대인지를 확인하는 과정을 의미한다. 따라서 이러한 문제를 해결하기 위한 보다 효율적이고 안전한 프로토콜의 개발이 절실하며, 본 논문에서는 이를 위해 기존에 제시되어진 패스워드를 이용한 인증 프로토콜들을 평문등가(plaintext-equivalent) 프로토콜과 확인자 기반(verifier-based) 프로토콜로 나누어 고찰해 보고, 이들을 안전성, 그리고 연산속도, 메시지 전송 횟수를 기준으로 비교 분석해 보고자 한다.

Ⅰ. 서론

클라이언트 측의 사용자들이 응용서버(application sever)로부터의 서비스를 받기 위해서는 사용자간에 또는 사용자와 암호시스템 간에 인증은 필수적인 요구 조건이다. 이러한 인증 과정은 현재 사용자가 통신을 통해 정보를 교환하고 있는 상대가 실제 의도한 상대인지를 확인하는 과정을 의미한다. 따라서, 이러한 문제를 해결하기 위한 보다 효율적이고 안전한 프로토콜의 개발이 절실하다.

사용자 인증 프로토콜은 인증의 기반이 되는 요소가 무엇이냐에 따라 다음과 같이 세 가지로 분류된다.

① 사용자의 물리적인 특징을 이용한 인증(목소리 식별, 망막 검사 등)

② 사용자가 소유한 물건을 통한 인증(ID card, smart card 등)

③ 사용자가 알고 있는 지식을 통한 인증(password, PIN)

이러한 방법들 모두 개별적으로 인증 프로토콜을 구현하는 데에 쓰일 수 있으나, 보다 강력한 인증을 위해서는 두 가지 이상이 혼합되어 사용되기도 한다.

첫 번째 방식과 두 번째 방식은 강력한 보안을 위해 사용되기도 하지만 그에 따르는 부가적인 하드웨어 비용 또한 크다. 반면, 세 번째 방식은 별도로 필요로 하는 장비가 없기 때문에 큰 비용을 들이지 않고 쉽게 사용될 수 있는 방식으로 패스워드 프로토콜이 이에 해당한다. 그러나, 사용자들이 자신의 패스워드를 선택할 수 있도록 해주는 암호시스템에서 사용자들은 쉽게 기억할 수 있는 패스워드를 선택하는 경향이 있다.

만약, 이렇게 선택된 패스워드의 일방향 해쉬함

* 단국대학교 전자계산학과 정보보안 연구실(seclab.anseo.dankook.ac.kr)
** 단국대학교 전자계산학과 부교수
*** 고려대학교 전자계산학과 부교수

수 값이나, 패스워드를 대칭키로 사용한 암호문이 공격자에게 도청 당한다면, 패스워드 추측공격을 당할 수 있게 된다.[8]

이러한 점들을 보완하기 위하여 사용자들간 또는 사용자와 시스템간에 패스워드와 부수적인 매개변수를 이용한 프로토콜이 사용되고 있으며, 이러한 프로토콜은 크게 다음과 같이 두 가지로 구분할 수 있다. 첫 번째로 서버가 사용자 패스워드의 복사본을 저장하고 있어야만 하는 인증 프로토콜이 있는데 이를 평문등가(plaintext-equivalent) 프로토콜이라 한다. 두 번째로 서버가 오직 사용자의 패스워드에 대한 확인자(verifier)만을 저장하고 있으면 되는 인증 프로토콜이 있는데 이를 확인자 기반(verifier- based) 프로토콜이라 한다. 확인자란 공개키와 비슷한 수학적 특성을 가지고 있는 것으로서, 패스워드를 알고 있을 경우에는 패스워드로부터 쉽게 계산되어질 수 있지만 확인자로부터 패스워드를 알아내는 것은 계산적으로 불가능하며, 간단한 예로 패스워드에 대한 해쉬값을 생각할 수 있다.[13]

본 논문에서는 기존에 제시되어진 평문등가와 확인자 기반 프로토콜들에 대해 고찰해 보고 이들을 안전성, 그리고 연산속도, 메시지 전송 횟수를 기준으로 비교 분석하는데 목적이 있으며 논문의 구성은 다음과 같다. 2장에서는 본 논문을 이해하는데 필요한 지식과 기반이 되는 프로토콜에 대해 알아보고, 3장에서는 평문등가 프로토콜들과 확인자 기반 프로토콜들에 대해 논의하며, 4장에서는 확인자 기반 프로토콜들에 대한 안정성, 연산속도와 메시지 전송 횟수를 기준으로 효율성을 비교 분석해 보도록 한다.

II. 기반 이론과 용어

여기에서는 본 논문을 이해하는데 필요한 이론과 기반이 되는 프로토콜에 대해 알아본다.

2.1 패스워드 프로토콜에서의 대표적 공격방법

알려진 대표적인 공격방법에는 패스워드 추측 공격(password guessing attack)[4], Denning-Sacco 공격[1], 신분위장 공격(masquerade attack)[11] 등이 있다.

패스워드 추측 공격은 공격자가 사용자에 의해 자주 선택되는 패스워드들에 대한 사전(dictionary)을 가지고 있다고 할 때 두 가지 방법으로 나누어 질 수 있다. 첫 번째로 온라인 패스워드 추측 공격(on-line password guessing attack)이 있는데, 공격자는 사전으로부터 임의로 선택된 패스워드를 기반으로하여 정당한 사용자로 가장해본다. 만약 공격이 실패하면 공격자는 선택한 패스워드를 사전에서 제거하고, 다른 패스워드를 선택하여 이를 반복하게 된다. 두 번째로 오프라인 패스워드 추측 공격(off-line password guessing attack)은 공격자가 이전 통신에서 사용된 메시지들을 기록해 놓고, 이들을 이용해 자신이 추측한 패스워드에 대한 확인을 통해서 패스워드를 찾는 방법이다.

온라인 패스워드 추측 공격은 실패 회수를 제한함으로써 쉽게 막을 수 있지만, 오프라인 패스워드 추측 공격은 오프라인으로 수행되어지기 때문에 공격을 감지하기가 매우 어렵다. 물론 패스워드의 선택에 있어서 큰 키 공간으로부터 랜덤하게 선택되어지는 패스워드를 사용한다면 이러한 오프라인 추측 공격으로부터 안전해 질 수 있지만, 이 경우에는 사용자가 자신의 패스워드를 기억하기 위해 메모와 같은 부수적인 방법을 사용하여야 하고, 이것은 보안적 측면에 있어서 또 다른 문제들을 발생시킨다.

Denning-Sacco 공격은 세션키가 누출되었을 때, 공격자가 그 동안 도청한 정보들을 기반으로 사용자의 패스워드에 대한 정보나 앞으로 진행될 세션에서 사용되어 질 세션키에 대한 정보를 얻고자 하는 공격이다.

신분위장 공격은 정당한 사용자가 아닌 공격자가 시스템을 이용하기 위해 정당한 사용자로 위장을 하는 공격이다.

2.2 기지평문과 추정가능문

기지평문(known plaintext)이라는 개념은 지금까지 암호학을 연구하는 학자들의 주된 관심사였다. 암호분석가가 주어진 암호문을 복호화하지 않고도 암호문에 대한 평문의 일부 또는 전부를 예측할 수 있을 때, 이러한 암호문은 기지평문을 포함하고 있다고 한다. 반면에, 암호분석가가 주어진 암호문

을 자신이 추측한 암호화키 K'를 사용하여 복호화 했을 경우에 자신이 추측한 암호화키 K'가 올바른지를 추측할 수 있는 내용, 즉 평문이 가질 수 있다고 생각되어지는 범위 내에 포함되는 내용을 찾을 수 있다면 이러한 암호문은 추정가능문(verifiable text)을 포함하고 있다고 한다. 만약 암호문이 기지 평문이나 추정가능문을 포함하고 있다면 공격자로부터 암호화키에 대한 오프라인 패스워드 추측 공격을 받을 수 있다고 알려져 있다.[5]

기지평문의 예로써 만약, 공격자가 평문 m, n을 대칭키 K로 암호화 한 $K(m, n)$을 도청했고, 평문이 값 m을 포함한다는 것을 알고 있다고 가정하자.그렇다면 공격자는 대칭키 K'을 추측한 후 이를 사용해 암호문 $K(m, n)$을 복호화하여, 결과 값에 m의 포함 유무를 검사하면 자신이 추측한 키 K'이 올바른지를 검사할 수 있다.

추정가능문의 경우 Kerberos 인증 시스템을 예로 들어 살펴보면[5], 사용자의 요청에 대한 서버의 초기 응답은 시각표인 *timestamp*와 네트웍 서비스 이름인 S 또는 주소와 같은 인식할 수 있는 정보를 포함하고 있으며, 이는 사용자의 패스워드로부터 유추된 암호화키 K를 사용하여 암호화되어진 후 사용자에게 전달된다. 공격자가 암호문 $K(S, timestamp, \cdots)$을 도청했다면, 평문이 *timestamp*와 S를 포함한다는 것을 알고있기 때문에, 공격자는 키 K'을 추측한 후 암호문 $K(S, timestamp, \cdots)$을 복호화하여 결과 값이 *timestamp* 형식의 값과 S의 올바른 형식을 포함하고 있는지를 검사하면 자신이 추측한 키 K'이 올바른지를 확인할 수 있다. 따라서, 암호문은 기지 평문이나 추정가능문을 포함하여서는 안된다.

2.3 평문등가 방식과 확인자 기반 방식

패스워드를 이용한 인증 프로토콜은 세션키 설정단계와 세션키 인증단계(세션키 확인 및 상호 인증)로 구분되어 진다. 먼저 세션키 설정단계에서는 사용자와 서버가 공유하고 있는 정보(패스워드 또는 확인자)를 기반으로 세션키를 생성하고, 세션키 인증단계에서는 서로가 공유한 세션키가 올바른지 그리고 정당한 사용자와 서버인지를 인증 하게 된다. 만약 사용자와 서버가 공통된 정보를 가지고 있었다면 동일한 세션키를 생성하게 될 것이고, 또한 공유한 세션키가 올바르다면 상대방을 인증할 수 있게된다.

이때 세션키 설정단계에서 사용되어지는 정보, 즉 서로가 공유하고 있는 정보에 따라 패스워드 프로토콜을 평문등가(plaintext-equivalent) 방식과 확인자 기반(verifier-based) 방식으로 구분할 수 있다. 사용자와 서버가 직접적인 패스워드를 사용하는 프로토콜은 평문등가 방식으로 구분되며, 사용자는 직접적인 패스워드를 이용하고 서버는 패스워드로부터 유추되어지는 확인자를 사용하는 프로토콜은 확인자 기반 방식으로 구분된다.[13]

2.4 패스워드를 사용하는 인증 프로토콜의 보안 요구사항

① 안전한 패스워드 프로토콜을 설계할 때 가장 문제가 되는 것은 일반적으로 패스워드의 키 공간이 좁기 때문에 랜덤하게 선택된 암호화키에 비해서 공격이 쉽다는 것이다. 이러한 이유 때문에 앞에서 설명했던 오프라인 패스워드 추측 공격이 가능하게 되며, 이러한 패스워드가 암호화 함수의 키로서 사용되어진다면 안전한 암호화 함수까지도 취약하게 만들 수 있다.[8]

② 양자간에 설정된 세션키는 패스워드에 대한 어떠한 정보도 가지고 있지 않아야 한다. 만약 세션키가 패스워드에 관한 정보를 가지고 있다면 세션키를 알아낸 공격자에 의해 사용자의 패스워드가 공격을 당할 수 있다.

③ 패스워드가 누출되어도 이전에 사용되었던 세션키가 누출되지 않아야 한다. 즉, perfect forward secrecy[11] 특성을 가져야 한다.

2.5 Diffie-Hellman 키 교환 프로토콜

Diffie-Hellman 키 교환 프로토콜[14]은 두 사용자간 또는, 두 시스템간에 공통된 비밀키를 공유하기 위한 목적으로 사용된다. 예를 들어 Alice와 Bob사이에 공통된 비밀키를 공유하고자 한다고 하자. Alice와 Bob은 각각 자신의 공개키와 개인키를 가지며 상호간에 공개키를 교환함으로써 공통된

비밀키를 공유할 수 있게 된다. Diffie-Hellman 키 교환 프로토콜이 안전하게 운영되기 위한 필요조건은 비밀키 설정을 위해서 교환되는 공개키들을 공격자가 도청한다고 할지라도 Alice나 Bob의 개인키나 또는 이들간에 합의되는 비밀키를 도출하는 것이 계산적으로 불가능해야 한다는 것이다. Diffie-Hellman 키 교환 프로토콜의 안전성은, 이산대수 문제의 어려움에 그 기반을 두고 있다. 일단, Alice와 Bob간에 공통된 비밀키가 공유되면 DES나 3-DES, IDEA등의 대칭형 암호화를 이용하여 메시지를 암호화할 수 있게 된다.

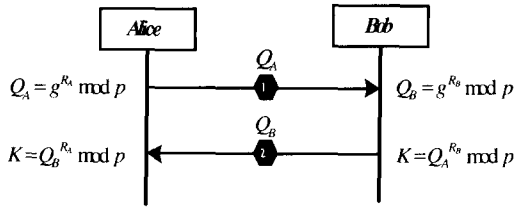

그림 1. Diffie-Hellman 키 교환 프로토콜

Diffie-Hellman 키 교환 프로토콜에서는 각 사용자들의 공개키와 개인키를 도출하는 데에는 매우 큰 소수 p와 위수가 $p-1$인 곱셈상의 군 Z_p^*의 생성자 g가 요구된다. Alice와 Bob은 임의의 난수 R_A와 R_B를 생성하여 자신의 개인키로 보관하고 또한 각각의 공개키 $Q_A = g^{R_A} \bmod p$와 $Q_B = g^{R_B} \bmod p$를 계산하여 상호간에 교환한다. Alice는 Bob의 공개키 Q_B에 자신의 비밀키 R_A를 적용하여 공통된 비밀키 $K = Q_B^{R_A} \bmod p = g^{R_A R_B} \bmod p$를 계산하고, Bob 역시 Alice의 공개키 Q_A에 자신의 비밀키 R_B를 적용하여 공통된 비밀키 $K = Q_A^{R_B} \bmod p = g^{R_A R_B} \bmod p$를 생성할 수 있게 된다.

공격자가 Alice와 Bob간에 교환되는 공개키 Q_A와 Q_B를 도청하여 Q_A와 Q_B로부터 각각 R_A와 R_B를 도출해낼 수만 있다면 Diffie-Hellman 키 교환 프로토콜의 안전성은 깨지게 된다. 하지만 결국 공격자가 이산대수 문제를 풀어야 하기 때문에 이는 불가능하게 된다. 이러한 이산대수를 이용한 시스템의 안정성의 정도는 사용되는 소수 p의 조건에 따라 결정된다. 시스템에 사용되어지는 소수 p를 선택함에 있어 p가 충분히 크며 $p-1$이 큰 소인수를 갖는 등의 strong prime[11]의 성질을 만족한다면 이 시스템은 기존에 알려진 많은 공격방법으로부터 안전하게 된다.

따라서, 본 논문에서 사용되어지는 소수 p는 strong prime의 성질을 만족한다고 가정한다.

Ⅲ. 프로토콜 분석

이번 장에서는 기존에 제안되어졌던 프로토콜들에 대해서 알아보는데, 각 프로토콜의 동작 방식과 동작 원리에 대해서 먼저 알아보고 프로토콜의 안전성에 대해 논하도록 한다.

3.1절에서는 프로토콜을 설명하는데 사용되는 notation에 대해 설명하고, 3.2절에서는 평문등가(plaintext-equivalent) 방식의 프로토콜인 EKE와 SPEKE, DH-EKE에 대해서 알아본다. 3.3절에서는 확인자 기반(verifier-based) 방식의 프로토콜인 A-EKE, B-SPEKE, SRP에 대해 알아본다.

3.1 표기정의(Notation)

- P : 사용자 Alice의 패스워드
- v : 사용자 Alice의 패스워드로부터 유추되어지는 확인자(verifier).
- ID_A, ID_B : Alice와 Bob의 identity.
- $K(m)$: 대칭키 K를 이용한 메시지 m에 대한 암호문.
- $K^{-1}(c)$: 대칭키 K를 이용한 암호문 c에 대한 복호문.
- $E_K(m)$: 공개키 K를 이용한 메시지 m에 대한 암호문.
- $D_K(c)$: 개인키 K를 이용한 암호문 c에 대한 복호문.
- $h(m)$: 메시지 m에 대한 일방향 해쉬함수 값.

- $h(m_1)(m_2)$: 메시지 m_1에 대한 일방향 해쉬 함수 값을 대칭키로 이용한 메시지 m_2의 암호문
- C_A, C_B : Alice와 Bob이 선택한 임의의 시도 값(challenge).
- R_A, R_B : Alice와 Bob이 임의로 선택한 임시 비밀키.
- s : salt 값.
- g : 군 Z_p^*의 생성자.

3.2 평문등가 방식의 프로토콜

이번 장에서는 평문등가 방식의 프로토콜들 중에서 EKE, DH-EKE, SP-EKE 프로토콜에 대해 살펴본다.

1) EKE(Encrypted Key Exchange)

EKE 프로토콜[4]은 세션키 설정단계에서 사용자가 생성한 공개키가 사용되어지고, 전송되는 메시지들은 사용자의 패스워드 P로 암호화된다. 세션키 인증단계에서는 시도-응답(challenge-respon se) 방법[11]을 사용한다.

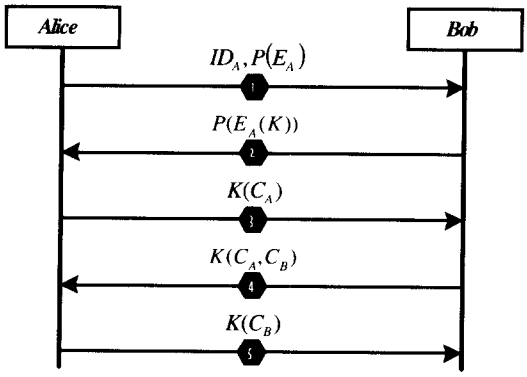

그림 2. EKE 프로토콜

① Alice는 공개키 (E_A)와 개인키 (D_A)를 생성한 후, 공개키를 패스워드 P를 사용하여 대칭키 암호화 시켜서 $P(E_A)$를 계산한다. 이를 자신의 identity와 함께 Bob에게 전달한다.

② Bob은 전달받은 $P(E_A)$로부터 공유하고 있는 패스워드 P를 사용하여 $P^{-1}(P(E_A))=E_A$를 얻는다. 그런 후 세션키 K를 랜덤하게 선택한 후, 이를 E_A를 사용하여 공개키 암호화시키고, 이를 다시 패스워드 P를 사용하여 대칭키 암호화 시켜서 $P(E_A(K))$를 생성한다. 그리고 이를 Alice에게 전달한다.

③ Alice는 전달받은 $P(E_A(K))$를 자신의 패스워드 P를 사용하여 $E_A(K)$를 구하고, 다시 개인키 D_A를 사용하여 세션키 K를 얻을 수 있다. 그런 후, C_A를 랜덤하게 선택하고 이를 전달받은 세션키 K를 사용하여 암호화하여 Bob에게 전달한다.

④ Bob은 세션키 K를 사용해서 $K(C_A)$로부터 C_A를 얻고 C_B를 랜덤하게 선택한 후, 이들을 세션키 K를 사용하여 암호화하여 Alice에게 전달한다.

⑤ Alice는 세션키를 사용해서 $K(C_A, C_B)$를 복호화 하여 C_B를 얻고, 이를 다시 세션키 K를 사용하여 암호화한 후 Bob에게 전달한다.

EKE에서 패스워드에 의해 암호화되는 메시지는 $P(E_A)$와 $P(E_A(K))$이다. 여기서 E_A와 $E_A(K)$는 추정가능문이 아니기 때문에 패스워드 추측 공격으로부터 보호되어진다.

세션키 K가 누출된 경우, 공격자는 추측한 패스워드 P'를 사용하여 $P(E_A)$를 복호화해서 $P'^{-1}(P(E_A))=E_A'$을 얻은 후 $E_A'(K)=P'^{-1}(P(E_A(K)))$를 계산하여 자신이 추측한 패스워드 P'이 올바른지 확인할 수 있게 된다. 따라서 Denning-Sacco 공격을 당할 수 있는데, 이는 compounder[5]를 사용하여 보완할 수 있다. Compounder란 공개키로 암호화된 추정가능문을 패스워드 추측 공격으로부터 보호하기 위해 사용되어지는 임의의 난수를 말하며, 프로토콜의 두 번째 단계에서 $P(E_A(K))$를 $P(E_A(K, c))$로 바꾸어

주면 공격자는 자신이 추측한 패스워드 P'을 사용하여 얻은 E_A'이 올바른지 확인하기 위해서 compounder c도 함께 추측을 해야만 하므로 공격이 불가능해 진다.

2) DH-EKE(Diffie-Hellman EKE)

DH-EKE 프로토콜[8]은 세션키 설정단계에 Diffie -Hellman 키 교환 프로토콜을 사용하고, 전송되는 메시지들은 사용자의 패스워드 P로 암호화된다. 세션키 인증단계에서는 시도-응답 방법을 사용한다.

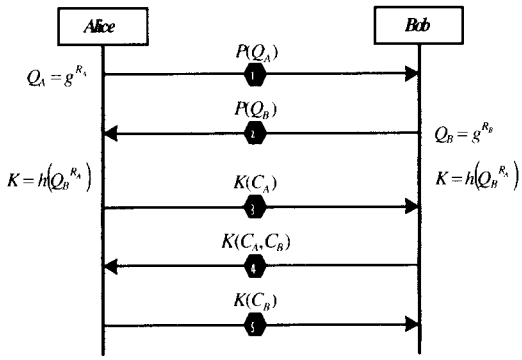

그림 3. DH-EKE 프로토콜

① Alice는 R_A를 선택한 후 Q_A를 계산하여 이를 패스워드 P로 암호화하여 Bob에게 전달한다.

② Bob은 R_B를 선택한 후 Q_B를 계산하여 이를 패스워드 P로 암호화하여 Alice에게 전달한다. 그러면, Alice는 전달받은 $P(Q_B)$로부터 패스워드 P를 사용하여 Q_B를 구할 수 있으며, 이를 사용하여 세션키 $K = h(Q_B^{R_A})$를 생성하고, Bob 역시 같은 방법으로 세션키 $K = h(Q_A^{R_B})$를 생성하게 된다.

③ Alice는 C_A를 선택하고 이를 생성한 세션키 K를 사용하여 암호화하여 Bob에게 전달한다.

④ Bob은 세션키 K를 사용해서 $K(C_A)$로부터 C_A를 얻고 C_B를 선택한 후, 이들을 세션키 K를 사용하여 암호화하여 Alice에게 전달한다.

⑤ Alice는 세션키 K를 사용해서 $K(C_A, C_B)$로부터 C_B를 얻고, 이를 다시 세션키 K를 사용하여 암호화한 후 Bob에게 전달한다.

세션키 K가 누출되었다고 해도 $K = h(Q_B^{R_A})$나 $K = h(Q_A^{R_B})$로부터 R_A와 R_B를 구해내는 것은 이산대수 문제이기 때문에 Denning-Sacco 공격은 불가능하다. 위의 과정중 ①과 ②에서 Q_A와 Q_B를 모두 패스워드 P로 암호화하여 전송하고 있다. 이 둘 중에서 하나는 암호화가 생략되어질 수 있으나 만약 생략한 측에서 먼저 세션키 K의 인증에 대한 시도(challenge)를 보내왔을 때에는 패스워드 P에 대한 패스워드 추측 공격을 받을 수 있다.[8]

3) SPEKE

SPEKE 프로토콜[8]은 세션키 설정단계에서 Diffie-Hellman 키 교환 프로토콜을 사용한다. 이 때 Diffie-Hellman 키 교환 프로토콜의 기저(base)는 아래에서 설명되는 함수 $f(P)$의 값이 사용된다. 세션키 인증단계에서는 시도-응답 방법을 사용한다.

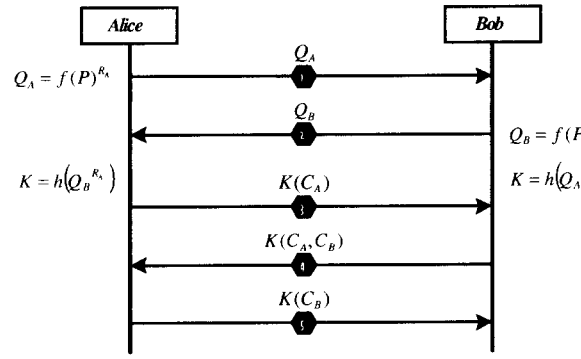

그림 4. SPEKE 프로토콜

① Alice는 R_A를 선택한 후, Q_A를 계산하여 이를 Bob에게 전달한다.

② Bob은 R_B를 선택한 후, Q_B를 계산하여 이를 Alice에게 전달한다. 그러면, Alice는 전달

받은 Q_A를 사용해서 세션키 $K=h(Q_B^{R_A})$를 생성하고, Bob 역시 전달받은 Q_B를 사용해서 세션키 $K=h(Q_A^{R_B})$를 생성한다.

③ Alice는 C_A를 선택하고 이를 생성한 세션키 K를 사용하여 암호화하여 Bob에게 전달한다.

④ Bob은 세션키 K를 사용해서 $K(C_A, C_B)$로부터 C_B를 얻고, 이를 다시 세션키 K를 사용해서 암호화한 후 Bob에게 전달한다.

세션키 K가 누출되었다고 해도 $K=h(Q_B^{R_A})$나 $K=h(Q_A^{R_B})$로부터 R_A와 R_B를 구해내는 것은 이산대수 문제이기 때문에 Denning-Sacco 공격은 불가능하다. SPEKE의 이산대수 공격(discrete log attack)에 대한 안정성은 다음과 같은 함수 $f(P)$를 사용함으로써 DH-EKE 보다 더 향상될 수 있다.

(1) $f(P)=g_q^s \bmod p$ 이때 g_q는 위수(order)를 q로 가진 원소이다.

(2) $f(P)=P^{(p-1)/q} \bmod p$

위의 식들은 위수로 q를 가지는 군 Z_p^*의 부분군(subgroup)의 생성자(generator)를 이용하는 것이다. 이러한 부분군을 사용할 경우에도 p가 strong prime의 특성을 만족한다면 암호시스템은 안전하다.

3.3 확인자 기반(verifier-based) 프로토콜

이번 장에서는 확인자 기반 프로토콜들 중 A-EKE, B-SPEKE, SRP 프로토콜에 대해 살펴본다.

1) A-EKE(Augmented EKE)

A-EKE 프로토콜[6]은 설정단계에서는 Diffie-Hellman 키 교환 프로토콜을 사용하고, 전송되는 메시지들은 사용자 패스워드 P의 확인자 $h(P)$로 암호화된다. 세션키 인증단계에서는 시도-응답 방법을 사용하며 사용자에 대한 인증 방법으로 디지털 서명이 일반적으로 사용된다.

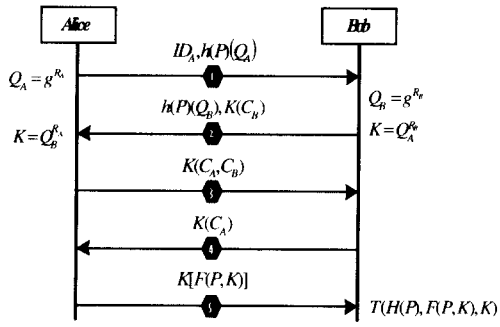

그림 5. A-EKE 프로토콜

① Alice는 임시 개인키 R_A를 생성하고 $Q_A=g^{R_A}$를 계산하여 확인자(verifier) $h(P)$로 암호화하여, 자신의 identity와 함께 Bob에게 전달한다.

② Bob은 전달받은 $h(P)(Q_A)$를 공유하고 있는 확인자 $h(P)$를 사용하여 복호화 하고 임시 개인키 R_B를 생성하여 $Q_B=g^{R_B}$와 세션키 $K=Q_A^{R_B} \bmod p$를 계산한다. 그 후, 랜덤하게 C_B를 선택하여, 세션키 K와 가지고 있는 확인자 $h(P)$를 이용해 $h(P)(Q_A)$와 $K(C_B)$를 Alice에게 보낸다.

③ Alice는 전달 받은 $h(P)(Q_B)$를 $h(P)$를 복호화하여 Q_B를 구하고 세션키 $K=Q_B^{R_A}$를 계한하여 C_B를 계산한다. 그런 후, C_A를 랜덤하게 선택하고 계산한 C_B와 함께 세션키 K를 사용해서 암호화하여 Bob에게 전달한다.

④ Bob은 세션키 K를 사용해서 $K(C_A, C_B)$로부터 C_A와 C_B를 구하고 자신이 보낸 C_B임을 확인하고 다시 K를 사용하여 C_A를 암호화하여 Alice에게 전달한다.

⑤ Alice는 전달 받은 $K(C_A)$를 세션키 K로

복호화 하여 자신이 보낸 C_A인가를 확인한다.

함수 $F(P, K)$는 패스워드 P와 세션키 K에 의해 값이 결정되는 일방향 함수이다. Alice는 자신의 패스워드 P와 설정된 세션키 K를 사용하여 얻은 값을 Bob에게 보내주게 되며, Bob은 다음과 같이 올바른 패스워드 P와 세션키 K에 대해서만 참 값을 돌려주는 인증함수 T를 사용하여 사용자가 올바른 패스워드 P를 사용하였는지를 검사하게 된다.

$$T(h(P),\ F(P,\ K),\ K)$$

인증함수 $T(X, Y, Z)$은 $X = h(P)$이고 $Y = F(P,K)$일 때 참이 되며, 프로토콜의 요구조건을 만족하는 안전한 $h()$, $F()$, $T()$에는 디지털 서명을 사용하는 방법이 있다.

A-EKE에서 확인자 $h(P)$에 의해 암호화되는 메시지는 $h(P)(Q_A)$와 $h(P)(Q_B)$이다. 여기서 Q_A와 Q_B는 추정가능문이 아니기 때문에 패스워드 추측 공격으로부터 보호되어 진다.

세션키 K가 누출되었다고 해도 $K = Q_B^{R_A}$나 $K = Q_A^{R_B}$로부터 R_A와 R_B를 구해내는 것은 이산대수 문제이기 때문에 Denning-Sacco 공격은 불가능하다.

2) B-SPEKE

B-SPEKE 프로토콜[10]은 세션키 설정단계에서는 Diffie-Hellman 키 교환 프로토콜을 사용하고, 세션키 인증단계와 사용자에 대한 인증 방법으로는 HMAC(hash-based MAC)[9][11]이 일반적으로 사용된다.

① Alice는 자신의 identity를 Bob에게 전달한다.

② Bob은 R_B를 선택하고 Alice의 확인자(verifier) S를 사용하여 Q_B를 계산한 후 이를 Alice에게 전달한다.

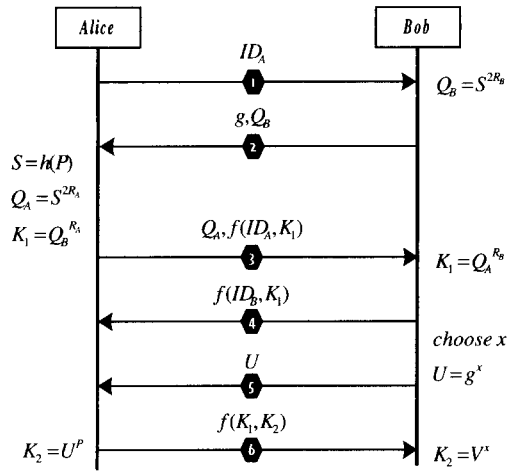

그림 6. B-SPEKE 프로토콜

③ Alice는 자신의 패스워드 P를 사용하여 S를 계산하고 R_A를 선택한 후 Q_A를 계산한다. 그리고, R_A와 Bob으로부터 전달받은 Q_B를 사용하여 세션키 K_1을 계산한다. 그런 후, Q_A와 $f(ID_A, K_1)$을 Bob에게 전달한다. 여기에서 함수 $f()$는 ID_A와 세션키 K_1에 대한 증명함수이다.

④ Bob은 전달받은 Q_A와 자신이 가지고 있는 R_B를 사용하여 세션키 K_1을 계산하고, 이를 증명함수를 통해 Alice로부터 전달받은 값과 비교함으로써 Alice가 올바른 확인자를 가지고 있는지를 확인한다. 그런 후, $f(ID_B, K_1)$을 계산하여 Alice에게 전달한다.

⑤ Bob은 임의의 x를 선택하여 U를 계산하고 이를 Alice에게 전달한다.

⑥ Alice는 전달받은 U와 자신의 패스워드 P를 사용하여 K_2를 계산한 후, K_1과 K_2를 이용해 증명함수 $f(K_1, K_2)$을 계산하여 Bob에게 전달한다. Bob은 확인자 $V = g^P$를 이용해 K_2를 계산하여 전달 받은 $f(K_1, K_2)$가 올바른지를 확인하여 Alice를 인증한다. 두 개의 키가 서로에 대한 정보를 제공하지 않을 때 즉, 두 개의 키 중 하나의 키가 누출이 되어도 누출된 키를 이용해 다른 한 키에 대한 정보를 알 수 없을 경우 두 키

간에 정보 은닉 성질(information-hiding property)[10]을 만족한다고 한다. Bob이 Alice가 정당한 패스워드 소유자인지를 알아보기 위해 사용되는 증명함수 $f()$는 K_1과 K_2 간의 정보은닉 성질을 만족해야 하며, 증명함수 $f()$는 HMAC이 사용될 수 있다.

B-SPEKE에서 세션키 K_1이 누출되었을 경우에도 공격자는 R_A 또는 R_B를 알지 못하므로 확인자 S를 알아낼 수 없다. 프로토콜에 사용되어지는 증명함수 $f()$가 K_1과 K_2간의 정보은닉 성질을 만족시키므로 K_1으로부터 K_2를 도출해 낼 수 없다. 따라서 Denning-Sacco 공격에 대해 안전하다.

3) SRP(Secure Remote Password Protocol)

SRP 프로토콜[13]은 세션키 설정단계에서는 이산대수 문제와 수학적 함수를 사용하고, 세션키 인증단계와 사용자에 대한 인증 방법으로는 해쉬함수를 사용한다.

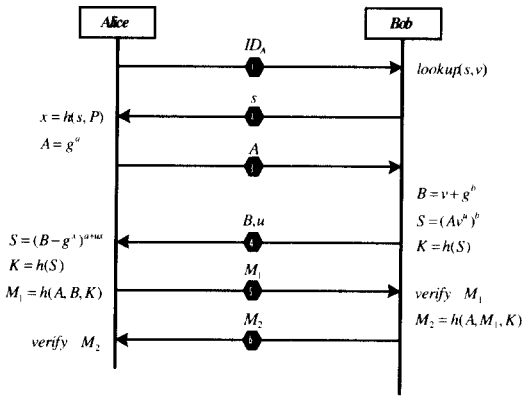

그림 7. SRP 프로토콜

① Alice는 자신의 identity를 Bob에게 전달한다.

② Bob은 Alice의 확인자 $v(v = g^x)$와 salt s를 찾은 후 s를 Alice에게 전달한다.

③ Alice는 전달받은 salt s와 자신의 패스워드 P를 사용하여 확인자 x를 계산한다. 그리고 임시 개인키 a를 선택하고 이에 대한 임시 공개키 A를 계산하여 Bob에게 전달한다.

④ Bob은 임시 개인키 b를 선택하고 이에 대한 임시 공개키 B를 계산한다. 그리고 임의의 u를 선택하고 Alice에게서 전달받은 A를 사용해서 $S = (Av^u)^b$를 계산한 후 이 값에 일방향 해쉬함수를 적용하여 세션키 K를 계산한다. 그런 후 이때 사용된 B와 u를 Alice에게 전달한다.

⑤ Alice는 전달 받은 B와 u, 그리고 자신의 임시 개인키 a를 사용하여 $S = (B - g^x)^{a+ux}$를 계산한 후 이 값에 일방향 해쉬함수를 적용하여 세션키 K를 계산한다. 그런 후, 계산된 세션키 K와 자신의 임시 공개키 A, 그리고 Bob의 임시 공개키 B에 대한 일방향 해쉬 값을 계산하여 Bob에게 전달한다.

⑥ Bob은 세션키 K, Alice의 임시 공개키 A, 그리고 자신의 임시 공개키 B를 사용하여 메시지 M_1을 검증하여 Alice를 인증한다. 그런 후 $M_2 = h(A, M_1, K)$를 계산하여 Alice에게 전달하면 Alice는 메시지 M_2가 올바른지 검사한 후 Bob을 인증하게 된다.

공격자가 세션키 K를 알아냈다고 하더라도 M_1, M_2는 공개된 정보를 사용하여 계산되어지는 것이기 때문에 K, M_1, M_2로부터 새로운 정보를 얻지 못한다. 그리고 세션키 K는 패스워드 P에 대한 어떠한 정보도 가지고 있지 않기 때문에 Denning-Sacco 공격으로부터 보호되어진다.

Ⅳ. 비교 분석

지금까지 알아보았던 프로토콜들은 평문등가 방식과 확인자 기반방식으로 구분될 수 있다. 평문등가 방식은 서버가 사용자의 패스워드나 비밀키를 저장

하고 있는 패스워드 데이터베이스를 관리하게 된다. 따라서 서버가 공격당하면 패스워드 데이터베이스에 저장되어 있는 모든 사용자의 패스워드가 누출되어지게 되고, 모든 사용자들이 피해를 당하게 된다.

반면에 확인자 기반방식은 서버가 사용자의 패스워드나 비밀키 대신 이로부터 유추되어지는 확인자를 저장하고 있게 된다. 그리고, 확인자 기반방식은 사용자가 올바른 패스워드를 직접적으로 사용하여 인증을 받는 단계가 추가되어 있다. 그러므로 서버가 공격을 당해 확인자를 저장하고 있는 데이터베이스가 공격자에게 누출되어진다고 하더라도 확인자를 사용한 직접적인 공격은 불가능하다.[13]

따라서 평문등가 방식보다는 확인자 기반방식이 보다 더 안전하다고 할 수 있으며, 여기에서는 확인자 기반방식 프로토콜인 A-EKE, B-SPEKE, SRP 프로토콜에 대해서 안정성 그리고 사용자와 서버측면에서의 연산속도, 메시지 전송 횟수를 중심으로 비교 분석해 보고자 한다.

아래에 설명되는 각 프로토콜의 메시지 전송 횟수에 따른 최적화 모델을 기반으로 안전성, 그리고 사용자와 서버측면에서의 연산속도, 메시지 전송의 횟수를 살펴보도록 한다.

안전성 측면 중 2.4절에서 논의했던 perfect forward secrecy 특성은 Diffie-Hellman 키 교환 프로토콜에서 Diffie-Hellman 지수로 임시개인키 (R_A, R_B)를 사용해 세션키를 생성함으로써 제공될 수 있다.[11] A-EKE와 B-SPEKE에서는 세션키 생성에 있어 Diffie-Hellman 키 생성 프로토콜에 기반을 두고 있으므로 perfect forward secrecy 특성을 갖는다. 그리고 SRP에서는 세션키를 생성할 때마다 매번 선택되어 사용되어지는 임시개인키 (a, b)로 인해, perfect forward secrecy 특성을 갖는다.

사용자와 서버측면에서의 연산속도에 있어서 가장 큰 영향을 주는 것은 군에서의 모듈러 지수승 연산이다. 이를 제외한 연산들은 모듈러 지수승 연산에 비해 적은 부분을 차지하므로 프로토콜의 연산속도에 영향을 주지 않는 것으로 가정한다. 각각의 프로토콜에 대한 모듈러 지수승 연산을 비교하기 위해 다음과 같은 표기를 정의한다.[13]

- t_g : 기저 g가 작은 모듈러 지수승 연산에 드는 시간
- t_e : 지수 e가 작은 모듈러 지수승 연산에 드는 시간
- t_b : 기저 g와 지수 e가 모두 큰 모듈러 지수승 연산에 드는 시간

먼저 최적화된 A-EKE 프로토콜[6]에 대해 살펴본다.

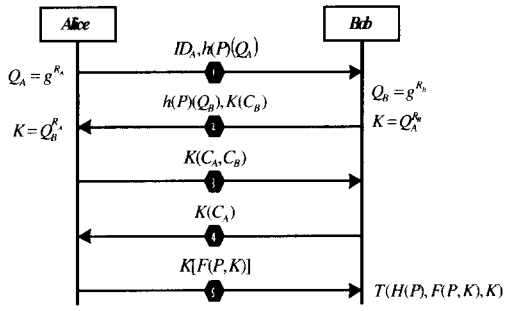

그림 8. 최적화된 A-EKE 프로토콜

안전성 측면을 살펴보면, 세션키 K가 누출되었다고 해도 $K = h(Q_B^{R_A})$나 $K = h(Q_A^{R_B})$로부터 R_A와 R_B를 구해내는 것은 이산대수 문제이기 때문에 Denning-Sacco 공격은 불가능하다. 패스워드의 확인자 $h(P)$로 암호화된 메시지인 $h(P)(Q_A)$와 $h(P)(Q_B)$는 추정가능문이 아니기 때문에 패스워드 추측 공격은 불가능하다. 또한 ⑤번 메시지에서 사용자가 올바른 패스워드를 알고 있는지를 검사하기 때문에 신분위장 공격 역시 불가능하다.

연산속도에 대해 살펴보면, 사용자 측면에서는 ①번 메시지에서 t_g의 시간이 걸리고, 세션키 K를 생성할 때 t_b의 시간이 걸린다. 그리고 디지털 서명을 할 때 $t_g + t_e$의 시간이 걸리므로, 총 $2t_g + t_e + t_b$의 계산시간이 소요된다. 서버측면에서는 ②번 메시지에서 t_g의 시간이 걸리고, 세션키 K를 생성할 때 t_b의 시간이 걸린다. 그리고 디지털 서명을 확인하는데 $t_g + t_b$의 시간이 걸리므로, 총 $2t_g + 2t_b$의 시간이 소요된다. 그리고 메시지 전송의 횟수는 그림 8 에서 보는바와 같이 5회이다.

공격방법	A-EKE	B-SPEKE	SRP
Denning-Sacco 공격	안전	안전	안전
패스워드 추측 공격	안전	안전	안전
신분위장 공격	안전	안전	안전

표 1. 프로토콜들의 안정성 비교

프로토콜	메시지 전송 횟수	연산 속도	
		사용자	서버
A-EKE	5 회	$2t_g + t_e + t_b$	$2t_g + 2t_b$
B-SPEKE	4 회	$3t_b$	$3t_b + t_g$
SRP	4 회	$2t_g + t_b$	$t_b + t_e + t_g$

표 2. 프로토콜들의 연산속도 및 메시지 패스 횟수 비교

최적화된 B-SPEKE 프로토콜[10]을 살펴본다.

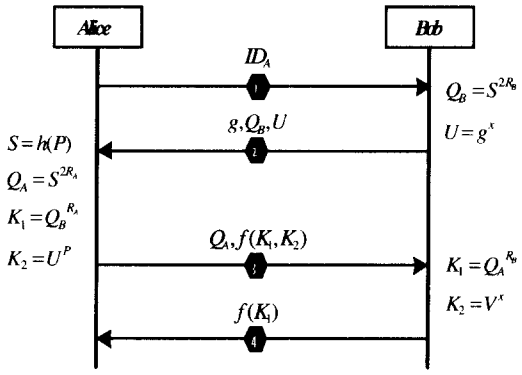

그림 9. 최적화된 B-SPEKE 프로토콜

안전성 측면을 살펴보면, 세션키 K_1이 누출되었을 경우에도 공격자는 R_A 또는 R_B를 알지 못하므로 verifier S를 알아낼 수 없다. 그리고 증명함수 $f()$가 K_1과 K_2간의 정보은닉 성질을 만족시켜 줌으로 K_1으로부터 K_2를 도출해 낼 수 없다. 따라서 Denning-Sacco 공격에 대해 안전하다. 그리고 패스워드와 관련된 정보를 가지는 메시지는 전송되지 않기 때문에 패스워드 추측 공격은 불가능하다. 또한 ④번 메시지에서 사용자가 올바른 패스워드를 알고 있는지를 검사하기 때문에 신분위장 공격 역시 불가능하다.

연산속도를 살펴보면, 사용자 측면에서는 Q_A를 계산하는데 t_b의 시간이 걸리고, K_1을 계산하는데 t_b의 시간이 걸린다. 그리고 K_2를 계산하는데 t_b의 시간이 걸리므로, 총 $3t_b$의 계산시간이 소요된다. 서버 측면에서는 Q_B를 계산하는데 t_b의 시간이 걸리고 U를 계산하는데 t_g의 시간이 걸린다. 그리고 K_1과 K_2를 계산하는데 각각 t_b의 시간이 걸리므로, 총 $3t_b + t_g$의 계산시간이 소요된다. 그리고 메시지 전송의 횟수는 그림 9 에서 보는바와 같이 4회이다.

최적화된 SRP 프로토콜[13]에 대해 살펴본다.

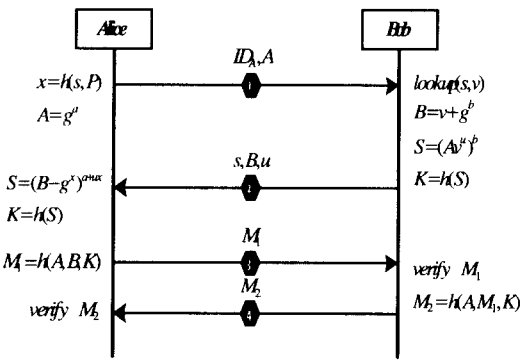

그림 10. 최적화된 SRP 프로토콜

안전성 측면을 살펴보면, 공격자가 세션키 K를 알아냈다고 하더라도 M_1, M_2는 공개된 정보를 사용하여 계산되어지는 것이기 때문에 K, M_1, M_2로부터 새로운 정보를 얻지 못하고, 세션키 K는 패스워드 P에 대한 어떠한 정보도 가지고 있지 않기 때문에 Denning-Sacco 공격으로부터 보호되어진다. 그리고 패스워드와 관련된 정보를 가지는 메시지는 전송되지 않기 때문에 패스워드 추측공격은 불가능하다. 또한 사용자는 매 세션마다 자신의 패스워드 P를 사용하여 확인자를 계산하는데 사용 되는 변수인 x를 계산하여야 하므로 신분위장공격은 불가능하다.

연산속도를 살펴보면, 사용자 측면에서는 A를 계산하는데 t_g의 시간이 걸리고, S를 계산하는데 $t_g + t_b$의 시간이 걸리므로, 총 $2t_g + t_b$의 계산시간이 소요된다. 서버 측면에서는 B를 계산하는데 t_g의 시간이 걸리고 S를 계산하는데 $t_b + t_e$의 시간이 걸리므로, 총 $t_b + t_e + t_g$의 계산시간이 소요된다. 그리고 메시지 전송의 횟수는 그림 10 에서 보는바와 같이 4회이다.

V. 결론

사용자가 쉽게 기억할 수 있는 패스워드를 이용한 인증 프로토콜은, 별도의 하드웨어 비용이 들지 않는다는 장점을 가지고 있다. 이러한 패스워드를 이용한 인증 프로토콜 중 확인자 기반 방법은 서버의 공격 위험성을 고려할 때, 평문등가 방법에 비해 안전성의 측면에서 우수함을 알 수 있다. 본 논문에서는 패스워드를 이용한 인증 프로토콜들을 살펴 보았으며, 그 중 안전성의 측면에서 우수한 확인자 기반 방법의 프로토콜들을 안전성, 그리고 연산속도, 메시지 전송 횟수를 기준으로 비교 분석해 보았다.

이들 모두 이미 알려져 있는 공격방법에 대해서는 안전한 것으로 나타났으며, 따라서 연산속도의 향상과 메시지 전송 횟수를 줄이는 방법에 대한 연구가 필요하다. 이중, 연산 속도는 일반적인 군에서의 모듈러 연산 대신, 타원곡선(elliptic curve)[15]에서의 곱셈연산을 사용함으로써 향상시킬 수 있을 것으로 기대되며, 메시지 전송 횟수에 대해서는 보다 많은 연구가 필요하다.

참고문헌

[1] D. Denning and G.Sacco, "Timestamps in Key Distribution Systems", *Communications of the ACM*, 1981

[2] L. Gong, "A Note on Redundancy in Encrypted Messages", *ACM Computer Communication Review*, vol. 20, no.1, pp.18 - 22, 1990

[3] W. Diffie, P. C. Van Oorschot and M.Wiener, "Authentication and Authenticated Key Exchanges", *Designs Codes and Cryptography*, vol. 2, pp.107 - 125, 1992

[4] Steven M. Bellovin and Michael Merritt, "Encrypted Key Exchange : Password-Based Protocols Secure Against Dictionary Attacks", *In Proc. IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, pp. 72-84, 1992.

[5] Li Gong, T. Mark A. Lomas, Roger M. Needham, and Jerome H. Saltzer, "Protecting poorly Chosen Secrets from Guessing Attack", *IEEE Journal on Selected Areas in Communications*, vol.11, no.5, pp. 648-656, 1993

[6] Steven M. Bellovin and Michael Merritt, "Augmented Encrypted Key Exchange : Password-Based Protocols Secure Against Dictionary Attacks and Password File Compromise", *Technical report, AT&T Bell Laboratories*, 1994.

[7] Michael Steiner, Gene Tsudik and Michael Waidner, "Refinement and Extension of Encrypted Key

Exchange" *Operationg Systems Review*, vol. 29, Iss. 3, pp.22-30, 1995.
[8] David P. Jablon "Strong Password-Only Authenticated Key Exchange", *Computer Communication Review, ACM SIGCOMM*, vol. 26, no. 5, pp. 5-26, 1996
[9] M.Bellare, R.Canetti and H.Krawczyk, "Keying hash functions for message authentication", *Advances in Cryptology-CRYPTO '96*, 1-15, 1996.
[10] David P. Jablon, "Extended Password Key Exchange Protocols Immune to Dictionary Attacks", *Proceedings of the Sixth Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET-ICE '97)*, IEEE Computer Society, Cambridge, MA, pp. 248-255, 1997
[11] A. Menezes, P. Van Oorschot and S. Vanstone, "Handbook of Applied Cryptography", *CRC Press*, 1997
[12] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols", *Proceedings of the Fifth ACM Conference on Computer and Communications Security*, 1998
[13] Thomas Wu, "The Secure Remote Password Protocol", *Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium*, San Diego. MA. pp. 97-111. 1998
[14] 박창섭, "암호이론과 보안", *대영사*, 1999.
[15] Certicom Corp., "Elliptic Curve Cryptosystem Tutorials and WhitePapers", *http://www.certicom.ca*

著者紹介

박 왕 석(Wang-suk Park) 학생회원

1998년 2월 : 단국대학교 수학과 졸업
1998년 8월 ~ 현재 : 단국대학교 전자계산학과 석사과정

정 종 필(Jong-pil Jung) 학생회원

1997년 2월 : 단국대학교 전자계산학과 졸업
1998년 8월 ~ 현재 : 단국대학교 전자계산학과 석사과정

박 창 섭(Chang-seop Park) 정회원

1983년 : 연세대학교 경제학 석사
1983년 : 한국 IBM System Administrator 근무
1987년 : LEHIGH Univ. 전자계산학 석사
1990년 : LEHIGH Univ. 전자계산학 박사
1990년 : 단국대학교 전자계산학과 조교수
1994년 ~ 현재 : 단국대학교 전자계산학과 부교수
〈관심분야〉 부호이론, 암호학

이 동 훈(Dong-hoon Lee) 정회원

1983년 8월 : 고려대학교 경제학사
1987년 12월 : Oklahoma Univ. 전산학 석사
1992년 5월 : Oklahoma Univ. 전산학 박사
1993년 2월 : 단국대학교 전자계산학과 전임강사
1997년 2월 : 고려대학교 전자계산학과 조교수
1997년 3월 ~ 현재 : 고려대학교 전자계산학과 부교수
〈관심분야〉 암호이론, 계산이론