情 報 保 護 學 會 誌
第 12 券 第 4 號, 2002. 8

패스워드 기반 키분배 프로토콜 표준화 동향

손 기 욱*, 서 인 석*, 원 동 호**

요 약

패스워드는 단순성, 편리성으로 인해 개인의 전자 거래 등에 있어 많이 사용되는 비밀 정보이다. 그러나 상대적으로 짧은 데이터의 사용이나 추축가능성 등의 보안상 취약한 부분도 내포하고 있다. 이에 대한 개선책으로 사전 공격이나 전수 조사 등의 공격 방법 등에 적용 가능한 프로토콜이 발표되었다. 본 논문에서는 패스워드 기반의 키 분배 프로토콜에 대한 기술에 대해 살펴보고자 한다.

Ⅰ. 서 론

사용자 인증을 위하여 많은 하드웨어 토큰과 생체 인식 기술이 개발되고 있지만 이러한 기술이 실용화 되기 위해서는 특별히 고안된 인터페이스 장치가 있어야 한다. 패스워드에 기반하는 인증은 특별한 외부 인터페이스 장치를 요구하지 않기 때문에 가장 널리 사용되고 있는 방법이다. 특히 현재의 웹 환경과 같이 어느 곳에서나 안전하게 서버와 접속하고자 하는 사용자에게는 패스워드를 이용하는 것이 가장 적절한 방법일 것이다.

그러나, 일반적으로 패스워드를 사용하는 암호 시스템은 오프라인 사전공격(off-line dictionary attack)에 대해 취약점을 가지고 있다. 1990년대 초반부터 오프라인 사전 공격에 대한 안전성을 갖는 방식들이 연구되었으며[1] 최근에는 이러한 연구 결과를 바탕으로 패스워드 기반 키 분배 방식의 표준화를 추진하고 있다[2].

패스워드를 사용한 키교환 프로토콜은 사용자가 단지 패스워드만을 사용하여 키교환을 수행할 수 있다는 점에서 최근에 매우 관심을 끌고 있는 프로토콜이다. 패스워드는 정보량적인 측면에서 낮은 엔트로피(불확실성)를 가지고 있기 때문에 패스워드에 대한 추측 공격에 약하다. 따라서 이러한 패스워드 추측 공격에 저항하기 위하여 비대칭 암호방식과 결합하여 사용되는 키분배 방식들이 Bellovin과 Merritt에 의해 처음 제안되었다[1]. 이러한 키분배 방식들은 보통 패스워드 자체를 이용하는 방식들과 패스워드를 인자로 갖는 어떤 함수를 이용하여 증명자와 검증자가 패스워드에 관해 다른 지식을 소유하게 만드는 방식으로 나눌 수 있다. 이때 증명자의 지식은 보통 패스워드 자체이고 검증자의 지식은 패스워드 자체는 알 수 없지만 패스워드로부터 나올 수 있는 값, 즉 식별자(verifier)라고 부른다. 보통, 개인식별 기능을 제공하기 위해서는 후자를 이용한다.

본 논문에서는 IEEE의 패스워드 기반 키분배 프로토콜의 표준화에 사용되는 공개키 기술 및 이를 기반으로 하는 프로토콜에 대한 기술동향을 소개하고자 한다. 2장에서는 IEEE P1363.2.에 기술된 표준에 관련된 내용을, 3장에서는 패스워드를 이용한 키 분배 프로토콜에 대한 내용을 시스템 구성 및 키 분배 과정에 대해 정리하고 마지막 4장에서는 결론을 맺고자 한다.

Ⅱ. IEEE의 패스워드 기반 키 분배 기술

IEEE P1363.2는 키 동의를 위한 패스워드 기반의 공개키 프로토콜 및 기반 기술을 정의하고 있다. P1363을 시작으로 논의된 패스워드 기반의 공개키 프로토콜은 P1363.a를 거쳐 2000년 후반부터 P1363.2를 통해 표준화를 정의하고 있다.[3]

P1363.2에서는 패스워드 기반의 키 분배 방식에

* 국가보안기술연구소({kiwook, isseo}@etri.re.kr)
** 성균관대학교 전기전자컴퓨터공학부(dhwon@dosan.skku.ac.kr)

서 사용되는 스킴 및 프로토콜에 대하여 관련 공개 키 기술과 더불어 표 1과 같이 정의하였다.

〔표 1〕 키 분배 스킴 및 프로토콜

Scheme or Protocol	Primitive
balanced password-authenticated key agreement	
{DL,EC} BPKAS-PPK	({DL,EC}REDP-1 and {DL,EC}PEPKGP-PAK and {DL,EC}SVDP-PAK)
{DL,EC} BPKAS-PAK-ALICE	({DL,EC}REDP-1 and {DL,EC}PEPKGP-PAK and {DL,EC}SVDP-DH)
{DL,EC} BPKAS-PAK-BOB	({DL,EC}REDP-1 and {DL,EC}PKGP-DH and {DL,EC}SVDP-PAK)
{DL,EC} BPKAS-SPEKE	({DL,EC}REDP-1 or DLREDR-2) and DL,EC}PEPKGP-SPEKE and {DL,EC}SVDP-SPEKE)
augmented password-authenticated key agreement	
{DL,EC} APKAS-AMP-CLIENT	({DL,EC}PKGP-DH and {DL,EC}SVDP-AMP-CLIENT and {DL,EC}PVDGP-AMP)
{DL,EC} APKAS-AMP-SERVER	({DL,EC}PEPKGP-AMP-SERVER and {DL,EC}SVDP-AMP-SERVER)
{DL,EC} APKAS-BSPEKE1-CLIENT	({DL,EC}REDP-1 and {DL,EC}PEPKGP-SPEKE and {DL,EC}SVDP-SPEKE and {DL,EC}PVDGP-BPSPEKE1)
{DL,EC} APKAS-BSPEKE2-SERVER	({DL,EC}REDP-1 and {DL,EC}PEPKGP-SPEKE and {DL,EC}SVDP-SPEKE and {DL,EC}PVDGP-BPSPEKE1)
{DL,EC} APKAS-BSPEKE2-CLIENT	({DL,EC}REDP-1 and {DL,EC}PEPKGP-SPEKE and {DL,EC}SVDP-SPEKE and {DL,EC}PVDGP-BPSPEKE2)
{DL,EC} APKAS-BSPEKE2-SERVER	({DL,EC}REDP-1 and {DL,EC}PEPKGP-SPEKE and {DL,EC}SVDP-SPEKE and {DL,EC}PVDGP-BPSPEKE2)
{DL,EC} APKAS-PAKX-CLIENT	({DL,EC}REDP-1 and {DL,EC}PKGP-PAKX-CLIENT and {DL,EC}SVDP-DH and {DL,EC}PVDGP-PAKX)
{DL,EC} APKAS-PAKX-SERVER	({DL,EC}REDP-1 and {DL,EC}PKGP-PAKX-SERVER and {DL,EC}SVDP-DH and {DL,EC}PVDGP-PAKX)
DLAPKAS-SRP-CLIENT	(DLPKGP-SRP-CLIENT and DLSVDP-SRP-CLIENT and DLPVDGP-SRP)
DLAPKAS-SRP-SERVER	(DLPKGP-SRP-SERVER and DLSVDP-SRP-SERVER and DLPVDGP-SRP)
password-based key retrieval	
{DL,EC} PKRS-1-CLIENT	(DLREDP-1 and DLPEPKGP-1 and DLSVDP-1) or (ECRESP-1 and ECPEPKGP-1 and ECSVDP-1)
{DL,EC} PKRS-1-SERVER	(DLSVDG-1) or (ECSVDG-1)

Families

DL : Discrete Logarithm

EC : Eliptic Curve

Schemes
APKAS : augmented PKAS
BPKAS : balanced PKAS
PKAS : password-authenticated key agreement scheme
PKRS : password-authenticated key retrieval scheme

Primitives
GE2SVOSP : group element to secret value octet string(conversion) primitive
GE2SVFEP : group element to secret value field element(conversion) primitive
PEPKGP : password-entangled PKGP
SVDP : secret value derivation primitive
PKGP : public key generation primitive
PVDGP : password verification data generation primitive
REDP : random element derivation primitive

Ⅲ. 패스워드 기반 키 분배 프로토콜 기술

본 장에서는 패스워드를 이용하여 키 분배를 수행하는 프로토콜에 대해 살펴보도록 한다. 프로토콜의 동작은 시스템 구성 및 키 분배 과정을 중심으로 기술한다.

1. DH-EKE[1]

그림 1에 나타난 DH-EKE는 패스워드 P의 일방향 해쉬함수 값인 h(P)를 암호화 키로 사용하는 대칭암호 시스템과 DH 키동의 방식을 결합하여 세션키를 분배하는 프로토콜이다. DH 키동의 방식에서 사용되는 소수의 형태는 *Pohlig-Hellman*이 [4][PH78]에서 지적한 것과 같이 p-1이 적어도 하나의 큰 소수를 포함해야 한다. DH-EKE에서 시스템 파라미터는 소수 $p=2q+1$(q:큰 소수)와 원시원소 g를 가지며, 키분배 방식은 다음과 같다.

[시스템 구성]

사용자 A는 패스워드를 선택하고 사용자 B(혹은 호스트)는 그 패스워드의 식별자 h(P)를 가지고 있다. 소수 p는 2q+1의 형태를 가지고 g는 Z_p^*상의 원시원소이다 (단, p, q는 큰 소수). N_A, N_B는 일련번호이고 $E_{h(P)}(\cdot)$는 h(P)를 암호화 키로 사용하여 대칭 암호 방식으로 암호화된 값이다.

[키동의 과정]

① 사용자 A는 랜덤수 r_A를 선택하여 자신의 IDA와 $E_{h(P)}(g^{r_A} \bmod p)$를 B에게 보낸다.

$$IDA,\ E_{h(P)}(g^{r_A} \bmod p)$$

② B는 사용자 A가 보내준 정보를 복호화하고 랜덤수 r_B를 선택하여 세션키를 구한다.

$$SK \equiv (g^{r_A})^{r_B} \equiv g^{r_A r_B} \bmod p$$

③ B는 다음과 같은 전송정보를 사용자 A에게 보낸다.

$$E_{h(P)}(g^{r_B} \bmod p),\ E_{SK}(N_B)$$

④ A는 B가 보내준 전송정보를 복호화하여 세션키를 계산하고, 세션키를 이용하여 N_B를 복호화한다.

$$SK \equiv (g^{r_B})^{r_A} \equiv g^{r_A r_B} \bmod p,$$
$$N_B = D_{SK}(E_{SK}(N_B))$$

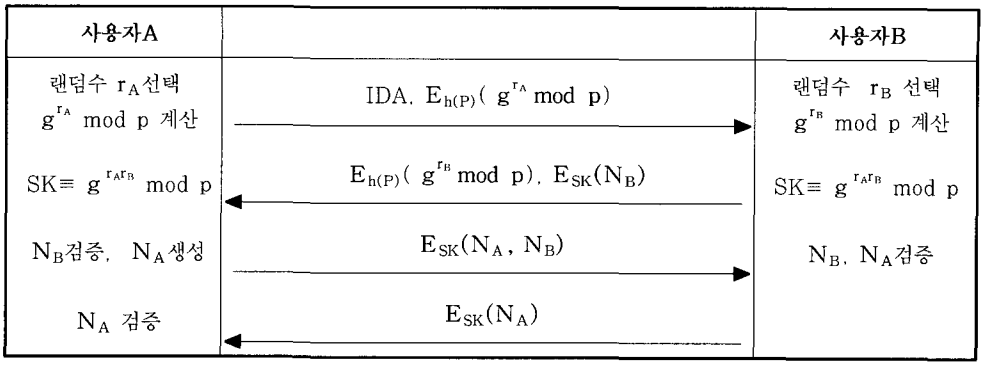
⑤ A는 키확인을 위해 N_A를 생성 후 다음과 같이 B에게 전송한다.

$$E_{SK}(N_A, N_B)$$

⑥ B는 $E_{SK}(N_A, N_B)$를 복호화하여 N_A, N_B를 검사하고, 키확인을 위해 다음을 A에게 전송한다.

$$E_{SK}(N_A)$$

이 프로토콜은 perfect forward secrecy를 제



〔그림 1〕 DH-EKE

공하며 known key 공격에도 안전하다[5-6]. 도청하는 공격자는 후보 패스워드 P′로써 $D_{h(P')}(E_{h(P)}(g^{r_A}(\bmod p))) \equiv g^{r_A'} \bmod p$를 구할 수 있지만 $g^{r_A'} \equiv g^{r_A} \bmod p$를 검증할 수 있는 유용한 정보가 없다. 또한 직접적으로 세션키 SK가 드러나지 않기 때문에 신분을 위장(impersonation)한 능동적인 공격자도 자신의 전송정보를 이용한 패스워드 추측공격을 성공적으로 수행할 수 없다. 하지만 Z_p^*상의 원시원소를 사용하지 않을 경우, 분할 공격(partition 공격)에 의해 패스워드가 노출될 가능성이 많아진다. 즉, 도청하는 공격자는 계산된 $g^{r_A'} \bmod p$를 가지고 패스워드의 유효성을 검증할 수 있다. 즉, Z_p^*상의 원소 g의 위수가 p-1이 아니라면 $g^{r_A'} \bmod p$가 가질 수 있는 값의 범위가 정해지고 따라서 이 범위를 벗어나는 값을 가지면 추측된 패스워드가 올바른 패스워드가 아니라는 것을 알 수 있게된다[1, 7].

DH-EKE는 B가 저장하고 있는 식별자가 패스워드 추측 공격에 강하다 하더라도 이 식별자를 아는 공격자는 A처럼 행동할 수 있다는 문제점이 있다.

즉, B는 항상 사용자 A처럼 행동할 수 있게 되는 것이다. 이와 같은 문제점을 해결하기 위하여 [8] 〔BM93b〕에서는 디지털 서명을 사용하는 방식을 제안하고 있다.

2. A-EKE[8]

A-EKE 프로토콜은 DH-EKE의 문제점을 개선하고 이를 확장시킨 방식이다. 키인증 과정에 디지털 서명을 사용한다는 점에서 DH-EKE와 구별되며 식별자를 저장하고 있는 사용자 B(혹은 호스트)는 A로 가장할 수 없다. A-EKE는 디지털 서명을 사용함으로써 식별자 기반 메커니즘의 방법 2를 이용하고 있다(〔그림 2〕 참조).

〔시스템 구성〕

사용자 A는 패스워드를 이용하여 디지털 서명의 비밀키와 공개키 쌍 (S_P, V_P)을 생성하고 B의 저장량을 줄이기 위해 V_P를 DH-EKE의 h(P)로써 사용한다.

〔키동의 과정〕

① 사용자 A는 랜덤수 r_A를 선택하고, 자신의 개인정보 IDA와 $g^{r_A} \bmod p$를 공개키 V_p로 암호화하여 B에게 보낸다.

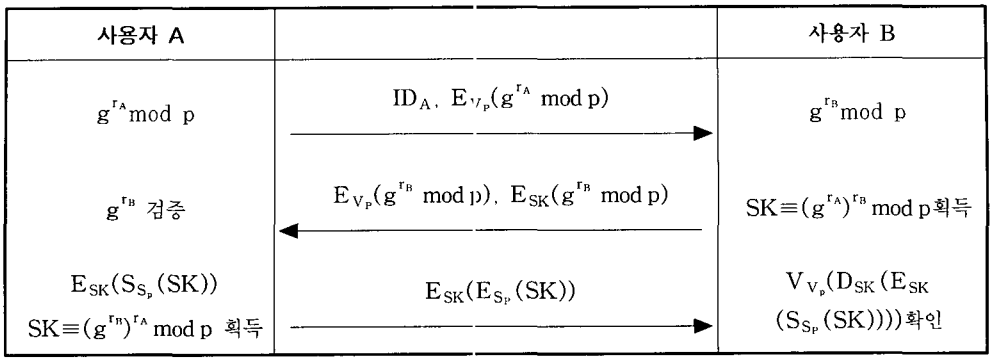
$$IDA, E_{V_p}(g^{r_A} \bmod p)$$

② 사용자 B는 랜덤수 r_B를 선택하고, A가 보내준 전송정보를 복호화하여 $g^{r_A} \bmod p$를 구하여 세션키를 계산한다.

$$g^{r_A} = D_{V_p}(E_{V_p}(g^{r_A} \bmod p))$$
$$SK \equiv (g^{r_A})^{r_B} \bmod p$$

③ 사용자 B는 $g^{r_B} \bmod p$를 암호화하여 사용자 A에게 전송한다.

$$E_{V_p}(g^{r_B} \bmod p), \quad E_{SK}(g^{r_B} \bmod p)$$



〔그림 2〕 A-EKE

④ A는 $D_{V_P}(E_{V_P}(g^{r_B} \bmod p))$를 계산하여 $g^{r_B} \bmod p$를 얻고 다음과 같이 세션키를 계산한다.

$$SK \equiv (g^{r_B})^{r_A} \bmod p$$

⑤ A는 얻어진 세션키를 이용하여 $E_{SK}(g^{r_B} \bmod p)$를 복호화하고 $g^{r_B} \bmod p$인지를 확인하여 B에 대한 인증을 수행하며, S_P를 이용하여 서명하고, 이를 다시 세션키 SK로 암호화하여 B에게 전송한다.

$$E_{SK}(E_{S_P}(SK))$$

⑥ B는 A가 보내온 정보를 SK로 복호화 하고 V_P로 검증한다.

사용자 A는 $E_{SK}(g^{r_B} \bmod p)$를 통해 B가 V_P를 알고 있는지 검증할 수 있고(원 논문에서는 Challenge-Response 프로토콜로 사용자 B가 SK를 알고 있다는 사실을 검증한다.)

B는 $E_{SK}(S_{S_P}(SK))$를 검증함으로써 A가 정말로 패스워드 P를 알고 있는 사람인지를 알 수 있다. 그러나 known key 공격 하에서는 S_P가 패스워드의 함수에 의해 생성되므로, 현재의 세션키 SK를 알고 있는 공격자가 $E_{SK}(S_{S_P}(SK))$를 이용해 패스워드 추측공격을 할 수 있다는 문제점이 있다[9][6][10]. 따라서 강한 패스워드의 사용이 필요하다.

3. SPEKE[7]

SPEKE는 A-EKE와는 달리 식별자 기반 메커니즘의 방법 1에 해당한다. DH-EKE에서는 분할 공격을 막기 위해 소수의 형태보다는 Z_p^*상의 원시원소를 사용하는 것이 중요한 반면에 SPEKE는 소수 위수 서브그룹(prime order subgroup)을 사용하기 때문에 소수의 형태가 매우 중요하다. 즉, 이 프로토콜에서 소수 p는 2q+1(q:큰 소수)의 형태를 가진다. 또한, 대칭암호 시스템을 사용하지 않고 패스워드를 통한 인증된 키분배 방식을 제공해 준다. 하지만 DH-EKE와 마찬가지로 A와 B의 비밀정보가 동일하므로 B는 다른 사용자(혹은 호스트)에게 A인척 할 수 있다. 여기에서 위수 q를 갖는 기저들로 나타낼 수 있는 그룹을 G_q로 표시한다(〔그림 3〕 참조).

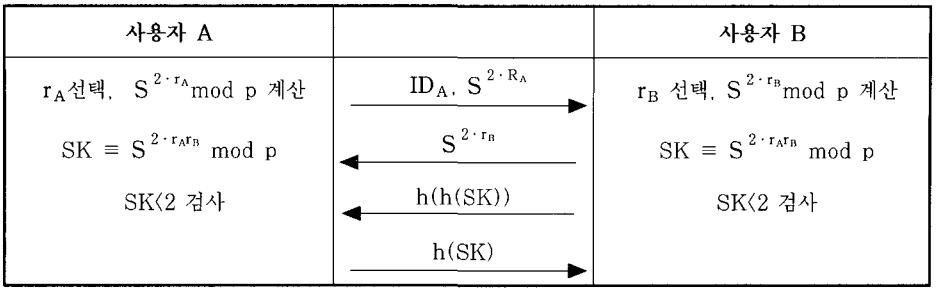
〔시스템 구성〕

사용자 A, B는 소수 p=2q+1(p,q는 큰 소수)와 위수 q를 갖는 Z_p^*의 기저(base)로 $S^{\frac{p-1}{q}} = S^2$를 사용한다. 여기에서 S=h(P)이다.

〔키동의 과정〕

① 사용자 A는 랜덤수 r_A를 선택하고 자신의 개인정보 IDA와 $S^{2 \cdot r_A} \bmod p$를 사용자 B에게 보낸다.

② 사용자 B는 랜덤수 r_B를 선택, $SK \equiv S^{2 \cdot r_A \cdot r_B}$



〔그림 3〕 SPEKE

mod p를 계산하여 SK=1이면 프로토콜을 실패로 끝내고, 그렇지 않으면 $S^{2 \cdot r_B} \bmod p$를 A에게 보낸다.

③ 사용자 A는 세션키 $S^{2 \cdot r_A \cdot r_B} \bmod p$를 계산하고 SK=1이면 프로토콜은 실패로 끝난다.

④ 사용자 B는 h(h(SK))를 사용자 A에게 보낸다.

⑤ 사용자 A는 h(SK)를 사용자 B에게 보낸다.

세션키 SK와 r_A가 노출되었을 경우를 대비해 세션키 SK′=h(SK)를 사용하고 키확인 과정인 ④에서 h(h(h(SK))), ⑤에서 h(h(SK))를 사용할 수도 있다. SPEKE의 패스워드에 대한 추측공격은 분할공격과 서브그룹 제한 공격으로 나누어 살펴볼 수 있다.

SPEKE에 대한 분할 공격은 추측한 패스워드 P′로 검사한다. 즉, 공격자는 $S^{2 \cdot r_{A'}}$나 $S^{2 \cdot r_{B'}}$와 같은 전송정보를 만들어서 그 값이 Z_p^*상의 원시근이면 기저도 원시근이라는 사실을 알 수 있고 이 사실에 근거하여 공격자는 S^2이 원시근이면 추측된 패스워드 P′가 올바른 패스워드가 아님을 알 수 있다. 이와 같은 분할 공격을 막기 위해서 사용자는 DH-EKE에서와 같이 원시원소 g를 사용해야 하지만 아래에 언급되는 서브그룹 제한 공격에 의해 소수 위수 서브그룹(prime order subgroup)의 사용도 요구된다[11][12]. 따라서, SPEKE에서는 이러한 요구를 절충한 소수 p=2q+1(q:큰 소수)를 사용하며, 최소한 분할 공격에 의해 1비트의 정보유출이 일어난다.

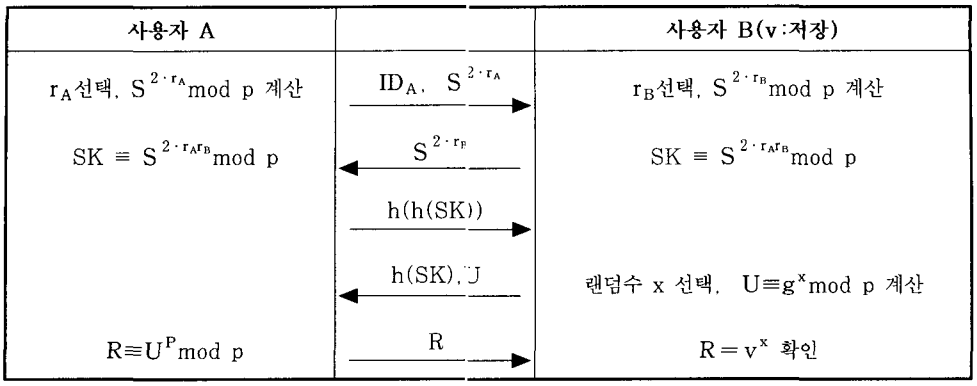
SPEKE에 대한 서브그룹 제한 공격은 소수 위수 서브그룹을 사용하지 않을 경우, 능동적인 공격자가 middle-person 공격에 의해 통신의 중간에서 메시지를 조정하거나 그렇지 않으면 정당한 사용자로 가장함으로써 가능하다. 예로써, 소수 p의 형태가 p=2qw+1(w:작은 소수, q:큰 소수)의 형태를 가진다면 통신의 중간에 위치하는 능동적인 공격자는 전송정보 $S^{2 \cdot r_A}$와 $S^{2 \cdot r_B}$에 q승을 함으로써 세션키 ($SK = S^{2 \cdot r_A r_B}$)를 서브그룹 G_w상의 원소로 만들 것이다. 따라서 공격자는 세션키를 1/w의 확률로 추측할 수 있다. 또한, A로 가장한 공격자는 작은 위수를 가지는 원소에 임의의 지수 승을 하여 전송함으로써 B의 세션키를 추측하기 쉬운 형태로 바꿀 것이다. 이러한 공격방법은 〔Ja96〕 〔OW96〕에 잘 나타나 있다[7][11].

3. B-SPEKE[13]

B-SPEKE는 B라는 확장명을 사용하며 디지털 서명을 사용하는 A-EKE와 구별한다. 이것은 A-EKE와 마찬가지로 식별자 기반 메커니즘 방법 2에 해당하며 DH방식을 사용한다. DH-EKE에 B방식을 적용한 것을 B-EKE라 하며 본 절에서는 SPEKE에 B방식을 적용한 B-SPEKE를 소개한다(〔그림 4〕 참조).

〔시스템 구성〕

사용자 A와 B는 S=h(P)를 비밀리에 공유하고 패스워드 P는 A만이 알고 있다. A는 식별자 $v = g^P$를 계산하여 B에게 비밀리에 준다. 여기에서 g는 Z_p^*상의 원시원소이고 소수의 형태는 p=2q+1(q:큰 소수)이다.



〔그림 4〕 B-SPEKE

〔키동의 과정〕

① 사용자 A는 랜덤수 r_A를 선택하고 자신의 개인정보 IDA와 $S^{2 \cdot r_A}$ mod p를 사용자 B에게 보낸다.

② 사용자 B는 랜덤수 r_B를 선택, $SK \equiv S^{2 \cdot r_A \cdot r_B}$ mod p를 계산하여 SK=1이면 프로토콜을 끝내고 그렇지 않으면 $S^{2 \cdot r_B}$를 사용자 A에게 보낸다.

③ 사용자 A는 $SK \equiv S^{2 \cdot r_A \cdot r_B}$ mod p를 계산하고 SK=1이면 프로토콜을 끝낸다.

④ B는 h(h(SK))를 A에게 보낸다.

⑤ A는 h(SK)를 B에게 보낸다.

⑥ B는 랜덤수 x를 선택하여 $U \equiv g^x$ mod p를 계산하고 U를 A에게 보낸다

⑦ A는 $R \equiv U^P$ mod p를 B에게 보낸다.

⑧ B는 R과 v^x가 같은지 검증한다.

$$R \triangleq v^x \equiv (g^P)^x \bmod p$$

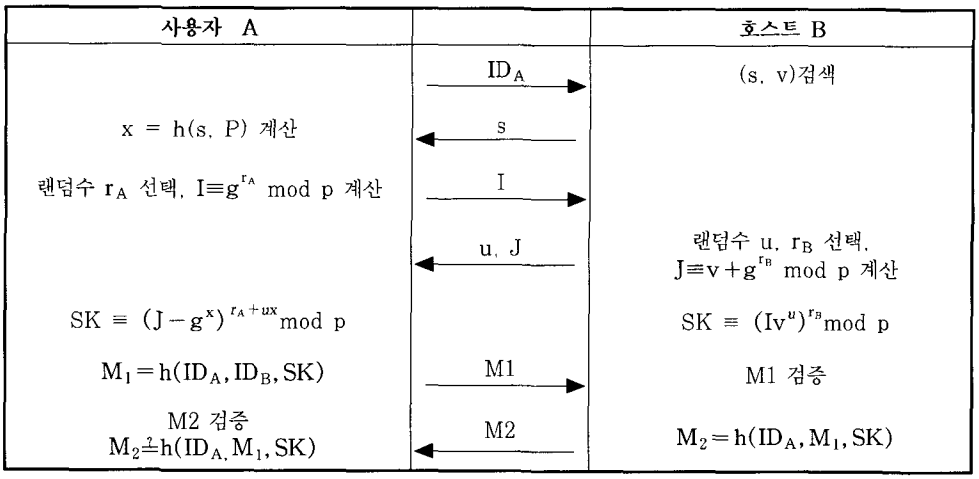
이 방식에서 A와 B는 K를 알고 있다는 사실을 h(h(SK))와 h(SK)로써 증명한다. 이것은 또한 두 사용자가 식별자 v를 알고 있는가에 대한 확인 절차이기도 하다. 특히, 이 방식에서 사용자 A는 A-EKE와는 다르게 그의 유일한 비밀정보 P(패스워드)에 대한 지식을 DH방식에 의해 증명하고 있다. 따라서, v를 아는 공격자도 A-EKE와 마찬가지로 다른 사용자(또는 호스트)에게 A인척 할 수 없다. 물론, v를 모르는 공격자에 대해서는 불확실성 정도가 낮은 약한 패스워드를 사용하더라도 시스템은 패스워드 추측공격에 대해 안전하다.

마. SRP[14]

SRP는 식별자 기반 메커니즘의 방법 2에 의한 키분배 프로토콜로 분할 공격(partition 공격)에 안전하고 서브그룹 제한 공격에 강하다. 〔Wu98〕에서 Thomas Wu는 안전한 소수 p=2q+1(q:큰소수)와 원시원소 g의 사용을 권장한다. 또한, 세션키 SK를 얻은 능동/수동 공격자도 패스워드에 대한 어떤 정보도 알 수 없는 안전성을 가진다[5]. 또한 앞에서 언급된 프로토콜과는 달리 B가 식별자 v를 알고있는가에 대한 확인과정과 A가 패스워드 P를 알고 있는 가에 대한 확인과정이 h(h(K))와 h(K)의 challenge-response에 함께 포함되어 있다(〔그림 5〕 참조).

〔시스템 구성〕

사용자 A, B는 소수 p=2p'+1와 원시원소 g, 해쉬함수 h를 공유한다. A는 식별자를 랜덤하게 하기 위한 랜덤수 s에 대해 x=h(s, P)를 계산하고 이 x를 이용해 식별자 $v = g^x$를 얻는다. A는 계산된 s와 v를 B에게 비밀리에 전송하고 x를 폐기한다. B는 이 s와 v를 저장한다.



〔그림 5〕 SRP

〔키동의 과정〕

① 사용자 A는 사용자 B에게 자신의 식별정보 IDA를 보낸다.

② 사용자 B는 사용자 A에 해당하는 랜덤수 s와 v를 찾고, s를 A에게 보낸다.

③ 사용자 A는 h(s, P)를 계산하고 $I \equiv g^{r_A} \bmod p$를 B에게 보낸다.

④ 사용자 B는 $J \equiv v + g^{r_B} \bmod p$를 계산하여 랜덤수 u와 함께 A에게 보낸다. 그리고 세션키 $SK \equiv (Iv^u)^{r_B} \bmod p$를 계산한다.

⑤ 사용자 A는 세션키 $SK \equiv (J - g^x)^{r_A + ux} \bmod p$를 계산하고, $M_1 = h(ID_A, ID_B, SK)$를 B에게 보낸다.

⑥ 사용자 B는 A로부터 전송된 M_1을 검증하고 $h(ID_A, M_1, SK)$를 A에게 보낸다.

⑦ A는 B로부터 전송된 $h(ID_A, M_1, SK)$를 검증한다.

$$M_2 \stackrel{?}{=} h(ID_A, M_1, SK)$$

이 프로토콜에서 사용자 A는 A-EKE와 B-SPEKE에서처럼 패스워드 P를 알고 있다는 사실을 B에게 증명하고 있다. 따라서, 단지 식별자만을 아는 공격자는 A인척 할 수 없다. 위 프로토콜에서 분할공격을 시도할 수 있는 전송정보는 J이지만 이것은 분할을 위한 어떤 정보도 제공하고 있지 않다[14]. 또한, 이 프로토콜의 안전성은 A와 B의 역할을 수행하는 공격자 A′, B′를 가정함으로써 좀 더 자세히 살펴볼 수 있다. A′는 공개정보와 추측한 패스워드 P′를 이용하여 x′를 구할 수 있고 이것으로부터 식별자 v에 대한 추측 v′를 계산할 수 있다. 하지만 B로부터 전송되는 정보에 A′가 추측한 v′를 검증할 수 있는 정보가 없기 때문에 추측할 수 있는 세션키 $K' = (Iv'^u)^{r_B}$가 올바른 것인가에 대한 답을 구할 수 없다. 또한 v를 아는 A의 역할을 수행하는 공격자 A′의 공격을 막기 위해 위의 프로토콜에서 B는 랜덤수 u를 이용한다. u가 없는 경우에 A′는 전송정보 $I = g^{r_A} v^{-1}$를 보내어 세션키 $SK = (Iv)^{r_B} = g^{r_A r_B}$를 쉽게 만들 수 있다. 물론 v를 아는 공격자에 대해서는 패스워드 추측공격에 강한 패스워드를 사용해야 할 것이다.

Ⅳ. 결 론

본 논문에서는 패스워드를 이용한 키 공유 프로토콜에 대한 IEEE의 표준화 관련 현황과 몇 개의 프로토콜을 소개하였다. 패스워드는 현재까지 사용자가 가장 간편하게 자신을 증명할 수 있는 방법으로 많이 사용되고 있으며, 하드웨어 토큰 및 생체 인식 기술을 이용한 방법이 완전하게 자리 잡기 전까지는

다양한 환경에 적합한 방식으로 볼 수 있다. 또한 키 복구 기능을 포함하는 프로토콜에 대한 연구도 병행되어 사용자와 서버 사이의 키 교환 과정에 적용하려는 노력도 계속 진행되고 있다.

참고문헌

[1] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange : Password-Based Protocols Secure Against Dictionary Attacks", Proceedings of the IEEE Symposium on Research in Security and Privacy. Oakland, May 1992.

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

[3] IEEE P1363.2, "Standard Specifications for Public Key Cryptography : Password-based Techniques", 2002.

[4] S. C. Pohlig, M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", IEEE Trans. Inform. Theory, IT-24(1), 1978, pp.106-110.

[5] D. E. Denning, G. M. Sacco, "Timestamps in key distribution protocols", Communications of the ACM, 1981.

[6] Y. Yacobi, "A key distribution paradox", Advances in Cryptology-Crypto'90, Springer-verlag, LNCS 537, pp. 245-255, 1990

[7] D. P. Jablon, "Strong Password-Only Authenticated Key Exchange", Computer Communication Review, ACM SIGCOMM, vol.26, no.5, pp5-26, October 1996.

[8] S. M. Bellovin, M. Merritt, "Augmented Encrypted Key Exchange : a Password -Based Protocol Secure Against Dictionary Attacks and Password File Compromise", Proceedings of the First ACM Conference on Computer and Communications Security, 1993.

[9] M. Steiner, G. Tsudik, M. Waidner "Refinement and Extension of Encrypted Key Exchange" ACM Operating Systems Review, 29(3), July 1995.

[10] D. E. Denning, G. M. Sacco, "Timestamps in key distribution protocols", Communi- cations of the ACM, 1981.

[11] P. C. van Oorschot, M. J. Wiener, "On Diffie-Hellman Key Agreement with Short Exponents", In Advances in Cryptology-Eurocrypt'96, Springer-verlag, LNCS 1070, pp.332-343, 1996

[12] R. Anderson, S. Vaudenay, "Minding your p's and q's", Advances in Cryptology-Asiacrypt'96, Springer-verlag, LNCS 1163, pp.15-25, 1996

[13] D. P. Jablon, "Extended Password Key exchange Protocols Immune to Dictionary Attack" In WETICE '97 Enterprise Security Workshop, Cambridge, MA, June 1997.

[14] T. Wu. "The Secure Remote Password Protocol." Internet Society Symposium on Network and Distributed System Security, 1998.

〈著者紹介〉

손 기 욱 (Kiwook Sohn)
정회원
1990년 2월 : 성균관대학교 정보공학과 졸업
1992년 2월 : 성균관대학교 정보공학과 석사
2002년 8월 : 성균관대학교 전기전자 및 컴퓨터공학부 박사
1992년~1999년 : 한국전자통신연구원 선임연구원
2000년~현재 : 국가보안기술연구소 선임연구원
관심분야 : 키분배 프로토콜, 공개키기반구조

서 인 석 (Inseog Seo)
정회원
1976년 2월 : 고려대학교 전자공학과 졸업
1988년 2월 : 충남대학교 전자계산학과 석사

1976년~2000년 : 국방과학연구소 중앙전산실장, 책임연구원
2000년~현재 : 국가보안기술연구소 키관리센터장, 책임연구원
관심분야 : 전산망 보안, 공개키기반구조

원 동 호 (Dongho Won)
종신회원
1949년 9월 23일생
성균관대학교 전자공학과 (학사, 석사, 박사)
한국전자통신연구소(ETRI) 전임연구원
일본 동경공대 객원연구원
성균관대학교 전산소장, 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장
국무총리실 국가정보화 추진위원회 자문위원
한국정보보호학회 이사, 부회장, 수석부회장
현재:성균관대학교 정보통신공학부 교수
한국정보보호학회 회장
정통부 지정 정보호호인증기술연구센터 센터장
성균관대학교 연구처장