패스워드 기반 키동의 프로토콜의 동향☆

심 현 정** 김 배 완** 유 종 호** 염 흥 열*

◈ 목 차 ◈

1. 서 론
2. 패스워드 기반 인증된 키동의 방식의 연구 동향
3. 결론

1. 서 론

패스워드는 암기하기 쉬운 간편성, 편리성으로 인하여 주로 이용되고 있는 사용자 인증 방법이다. 개방형 네트워크에서 안전하게 서버와 접속하고자 할 때는 손쉽게 사용자의 패스워드를 이용할 수 있다. 그러나 패스워드는 인간의 짧은 기억을 통해 유지됨에 따라 다양한 추측공격에 노출된다.

패스워드 기반 인증된 키교환 프로토콜이란 기본적으로 패스워드를 기반으로 상호간에 사용자 인증이 이루어진 다음 서로간에 공유될 키를 설정하는 절차를 의미한다. 일반적으로 사용자들은 자신의 패스워드를 선택할 수 있도록 해주는 암호시스템에서 사용자들이 쉽게 기억할 수 있는 패스워드를 선택하는 경향이 있다. 만약 이렇게 선택된 패스워드의 일방향 함수값이나 패스워드를 대칭키로 사용한 암호문이 공격자에게 도청 당한다면, 패스워드 추측공격을 당할 가능성이 존재하게 된다. 따라서 이러한 점들은 보완하기 위하여 사용자들간 또는 사용자와 시스템 간에 패스워드와 공개키 암호방식을 이용한 프로토콜로 확장되고 있다.[1]

패스워드 기반 인증 프로토콜은 크게 다음과 같이 두 가지로 구분할 수 있다. 첫 번째로 서버가 사용자 패스워드와 동일한 복사본을 저장하는 형태로 이를 평문등가(plaintext equivalent) 프로토콜이라 한다. 두 번째로 서버가 오직 사용자 패스워드에 대한 검증자(verifier)만을 저장하는 검증자 기반(verifier-based) 프로토콜이 있다. 검증자란 공개키와 비슷한 수학적 특성을 가지고 있는 것으로써, 패스워드를 알고 있을 경우에는 패스워드로부터 쉽게 계산되어질 수 있지만 검증자로부터 패스워드를 알아내는 것은 계산적으로 불가능하다. 이에 대한 간단한 예로 패스워드에 대한 해쉬값을 고려 할 수 있다.[1,9]

1992년 Bellovin과 Merritt가 EKE(Encrypted Key Exchange)[6]로 알려진 논문을 발표한 것을 필두로 추측공격에 강하게 저항하도록 패스워드 정보에 공개키 암호 알고리즘인 DLP(Discrete Logarithm Problem) 기반의 DH(Diffie-Hellman), RSA, 타원곡선(elliptic curve) 알고리즘 및 램덤 오라클(random oracle) 모델인 일방향 해쉬함수 등을 추가하여 안전성을 향상시켜왔다.

SRP[1], Auth[3], PAK[4] 등에서는 패스워드-파일 타협에 저항하고 클라이언트와 서버가 서로간에 동일하지 않는 정보를 기억하는 검증자(verifier)-기반 프로토콜들이 제안하였다. 패스워드는 사용자의 기억으로만 한정되는 엔트로피(entropy)로 제한되기는 하지만 여전히 검증자만으로부터 패스워드를 유도하는 것은 계산적으로 불가능하다. 그러나 만일 서버의 파일이 타협된다면 검증자-기반 프로토콜조차 여전히 사전 추측공격(dictionary guessing attack)을 허용하게 된다.

☆ 본 연구는 인터넷침해대응기술연구센터 지원으로 수행하였습니다
* 순천향대학교 공과대학 정보보호학과 교수
** 순천향대학교 일반대학원 정보보호학과 박사과정
** 순천향대학교 일반대학원 정보보호학과 석사과정
** 순천향대학교 일반대학원 전기·전자공학과 박사

이 문제에 대한 해결책으로 가장 좋은 방법으로는 [13]에서와 같이 검증자를 서버의 비밀키로 암호화하여 보관하거나 또는 [2,5]에서와 같이 검증자를 임계치(threshold) 기법을 통해 분산하는 것이다.[13]

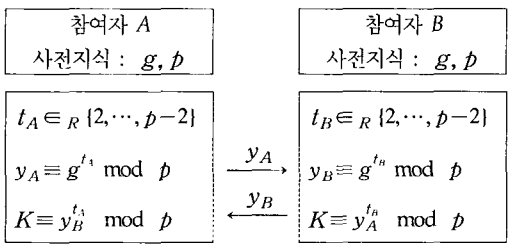
이와 같이 기존 알려진 사항을 토대로 본 논문에서는 패스워드 기반 키동의 프로토콜의 동향에 대하여 분석한다. 2장에서는 기존에 제안되었던 대표적인 방식들을 논하고 IEEE P1363.2[11]에서 추진중인 패스워드 기반 공개키 암호 기술 표준에 대하여 기술한 후, 3장에서 결론을 맺는다.

2. 패스워드 기반 인증된 키동의 방식의 연구 동향

지금까지 알려진 대표적인 대칭구조 패스워드 기반 인증 키분배 방식으로 EKE(Encrypted Key Exchange), SPEKE(Strong Password only Authenticated Key Exchange), DH-EKE(Diffie-Hellman EKE) 등이 있다.[6,7] EKE와 SPEKE는 기본적으로 평문등가 형태의 인증 프로토콜이기 때문에 서버와 클라이언트는 동일한 정보(패스워드)를 지니게 된다. 이것은 만일 서버가 내부/외부 공격자에 의해 타협된다면 패스워드가 노출될 위험이 따르게 된다[9].

2.1 Diffie-Hellman 키 교환 프로토콜

DH(Diffie-Hellman) 키교환 프로토콜은 두 사용자간 또는 두 시스템간에 공통된 비밀키를 공유하기 위한 목적으로 사용된다. 그림 1은 DH 키교환을 도시한 것이다.[8]



(그림 1) Diffie-Hellman 키교환

참여자 A와 참여자 B는 각각 자신의 공개키와 개인키를 가지며 상호간에 공개키를 교환함으로써 공통된 비밀키를 공유할 수 있다.

일단 참여자 A와 참여자 B간에 공통된 비밀키가 공유되면 DES(Data Encryption Standard)나 3-DES, IDEA(International Data Encyption Algorithm) 등과 같은 대칭형 암호에 적용하여 메시지를 암호화할 수 있게 된다.

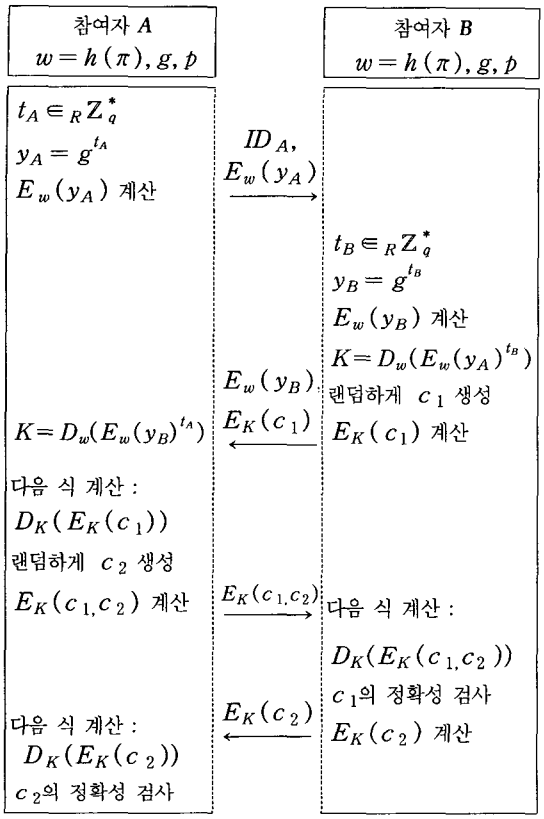
DH 키교환 프로토콜의 안전성은 이산대수 문제에 기반을 둔다. 이산대수 문제에 기반 둔 암호 시스템은 법(modulus) p의 조건에 의해 안전성이 결정된다. 따라서 DH 키교환 프로토콜에서의 각 사용자들은 공개키와 개인키를 계산함에 있어 적당히 큰 소수 p, 위수 $p-1$, 그리고 곱셈군 $\mathbb{Z}_p^*$의 생성원(generator) g를 사전에 규정해야 한다.[8]

2.2 DH-EKE 프로토콜

DH-EKE(Diffie-Hellman Encrypted Key Exchange)[6]는 패스워드 π의 일방향 해쉬함수 값인 $h(\pi)$를 암호화 키로 사용하는 대칭형 암호 시스템과 DH 키동의 방식을 결합하여 세션키를 분배하는 프로토콜이다.

DH-EKE 프로토콜은 세션키 설정단계에서 사용자가 생성한 공개키가 사용되고, 전송되는 메시지들은 사용자의 패스워드 w로 암호화된다. 그림 2는 DH-EKE 프로토콜을 도시한 것이다. 그림 2에서 $E_w()$ 및 $D_w()$은 각각 키 w로 대칭형 암호화/복호화 동작됨을 의미하고 $h(): \{0,1\}^* \rightarrow \{0,1\}^l$은 해쉬함수를 나타낸다. 참여자 A와 B는 소수 $p=2q+1$(p와 q는 큰 소수)와 위수 q를 프로토콜 수행 전에 조정하고, 또한 $\mathbb{Z}_p^*$상의 원시근(primitive root)과 합동인 임의의 생성원(generator) g를 설정해야 한다.

이 프로토콜은 전방향 안전성(perfect forward secrecy)를 제공하며 알려진 키(known key)공격에도 안전하다.[6,10] 도청자는 후보 패스워드 π'을 통해 $D_{h(\pi')}(g^{t_A}) = g^{t_A'}$을 구할 수 있지만, $g^{t_A} = g^{t_A'}$을 검증할 수 있는 유용한 정보를 갖지 못한다. 또한 직



(그림 2) DH-EKE 프로토콜

접적으로 세션키 K가 드러나지 않기 때문에 신분을 위장(impersonation)한 능동적인 공격자도 자신의 전송 정보를 이용한 패스워드 추측공격을 성공적으로 수행할 수 없다.

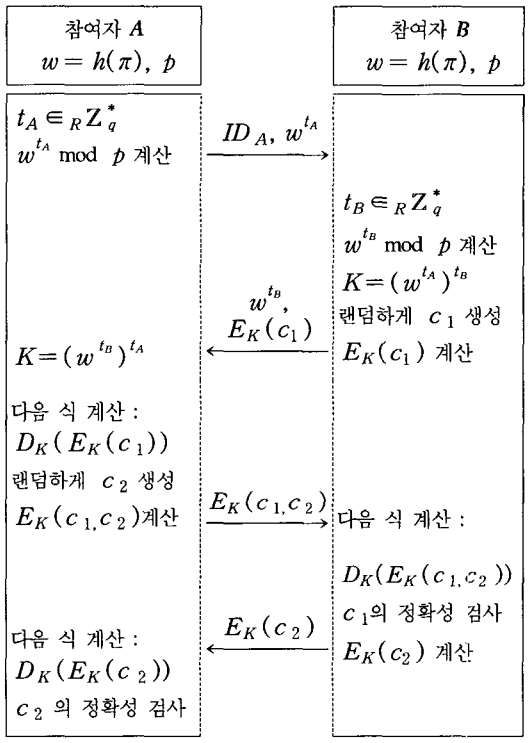
다만 Z_p^*상의 원시원소를 사용하지 않을 경우 분할 공격(partition attack)에 의한 패스워드 노출 가능성이 높아진다. 분할 공격은 오프라인(off-line) 분석으로서 오라클(oracle)에게 질문(ask)하는 것을 통해 패스워드들의 대수적 집합을 감소시키는 위한 공격이다. 즉, Z_p^*상의 원소 g의 위수가 $p-1$이 아니라면 $g^{t_A'}$가 가질 수 있는 값의 범위가 정해지고 따라서 이 범위를 벗어나는 값을 가지면 추측된 패스워드가 올바른 패스워드가 아니라는 것을 알 수 있게 된다. 이점은 도청자가 계산된 $g^{t_A'}$를 가지고 패스워드의 유효성을 검증할 수 있음을 의미한다.[7]

DH-EKE는 참여자 B의 검증자가 패스워드 추측공격에 강하다 할지라도 이 검증자를 아는 공격자는 참여자 A와 같은 행동을 할 수 있다는 문제점이 제기되어 있다. 이와 같은 문제점을 해결하기 위하여 [7]에서는 전자서명을 사용하는 방식을 제안하고 있다.

2.3 SPEKE 프로토콜

SPEKE(Strong Password Encrypted Key Exchange)[7] 프로토콜은 세션키 설정단계에서 DH 키교환 프로토콜을 사용한다. 이때 DH 키교환 프로토콜의 밑수(base)은 $w=h(\pi)$값이 사용된다.

DH-EKE에서는 분할 공격을 막기 위해 Z_p^*상의 원시원소를 사용하는 것이 중요한 반면 SPEKE는 소수 위수 부분군(prime order subgroup)을 사용하기 때문에 소수의 형태가 매우 중요하다. SPEKE에서 소수는 $p=2q+1$의 형태를 갖는다.



(그림 3) SPEKE 프로토콜

DH-EKE와 마찬가지로 참여자 A와 참여자 B의 비밀정보가 동일하므로 참여자 B는 다른 사용자에게 참여자 A인척 흉내내는 것이 가능하다. 세션키 인증 단계에서는 EKE와 마찬가지로 시도-응답 방법이 사용된다.[7,9] 그림 3은 SPEKE 프로토콜을 도시한 것이다.

SPEKE도 패스워드를 사용하여 암호화하기 때문에 만일 서버가 타협된다면 패스워드 추측공격이 가능하다.

2.4 SRP 프로토콜

Thomas Wu는 검증자 기반 SRP(Secure Remote Password)[1]프로토콜을 제안하였다.

이 프로토콜은 두 참여자의 키교환 설정단계에서 이산대수 문제를 이용하여 구성하고, 두 참여자간의 상호인증은 해쉬함수를 이용하여 구성된다.

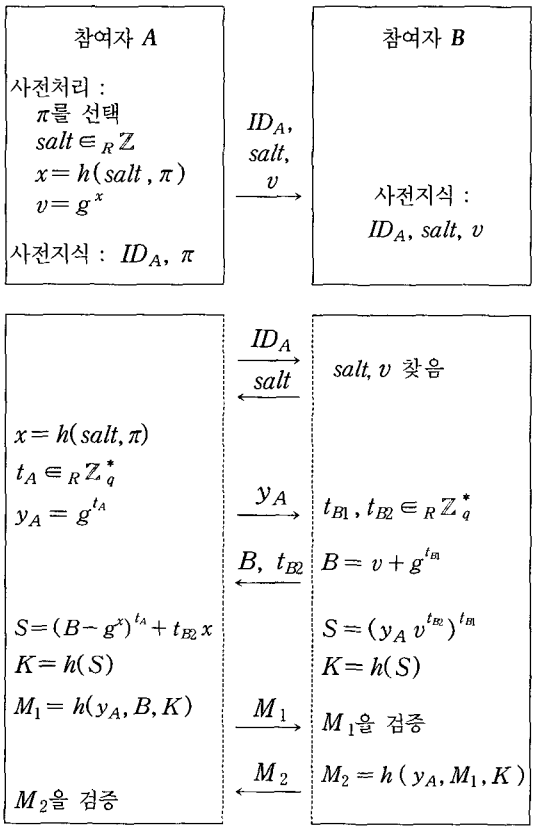
SRP는 비대칭형(검증자, verifier) 기반 메커니즘의 키분배 프로토콜로 분할 공격(partition attack)에 안전하고 부분군 제한 공격에도 강한 특성을 지닌다.

Thomas Wu는 안전한 소수 $p=2q+1$와 생성원 g를 사용하도록 구성하였다. SRP는 참여자 B가 검증자 v를 알고 있고 있는가에 대한 확인과정과 참여자 A가 패스워드를 알고 있는가에 대한 확인과정이 시도-응답 방법으로 처리된다. 이에 대한 상세 사항이 그림 4에 도시되어 있다.

참여자 A은 $x=h(salt,\pi)$와 $v=g^x$을 계산하기 위하여 난수값 $salt$을 선택한다. 참여자 B는 참여자 A의 패스워드 검증자(verifier)와 salt값로서 v 와 $salt$을 저장한다. x는 사용자의 패스워드와 $salt$로부터 유추된 개인키(private key)이지만 패스워드 π로부터 유도가 가능하기 때문에 사전처리 후 버려지게 된다.

SRP의 안전성은 참여자 A와 참여자 B의 역할을 수행하는 공격자 참여자 A'혹은 참여자 B'을 가정함으로써 좀 더 자세히 살펴 볼 수 있다.

참여자 A'는 공개정보와 추측한 패스워드 π'을 이용하여 x'을 구할 수 있고 이것으로부터 식별자 v'을 계산 할 수 있다. 하지만 참여자 B로부터 전송



(그림 4) SRP 프로토콜

되는 정보에 참여자 A'가 추측한 v'을 검증할 수 있는 정보가 없기 때문에 추측할 수 있는 세션키 $K'=(y_A(v')^{t_{B2}})^{t_{B1}}$가 올바른 것인가에 대한 답을 구할 수 없다. 또한 v를 아는 참여자 A의 역할을 수행하는 공격자 A'의 공격을 막기 위해 위의 프로토콜에서 참여자 B는 난수 t_{B2}를 이용한다. t_{B2}가 없는 경우에 공격자 A'는 $y_{A'}=y_A v^{-1}$을 보내어 세션값 $S=(y_{A'}v)^{t_{B1}}=g^{t_A t_{B1}}$ 를 쉽게 만들 수 있다.[1]

2.5 IEEE 패스워드 기반 공개키 암호 키동의 방식 표준

본 절에서는 IEEE P1363.2 [11]에 기술된 키동의를 위한 패스워드 기반 공개키 암호 방식 및 기반 기술

(표 1) IEEE1393.2 패스워드 기반 공개키 암호 키동의 방식

방 식	구체적인 세부 동작	추가사항
Balanced Password-authenticated Key Agreement Schemes		
{DL,EC}BPKAS-PPK-CLIENT, {DL,EC}BPKAS-PPK-SERVER	({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-PAK, {DL,EC}SVDP-PAK2	KDF1, KDF2
{DL,EC}BPKAS-PAK-CLIENT	({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-PAK, {DL,EC}SVDP-PAK1-CLIENT	KDF1, KDF2
{DL,EC} BPKAS-PAK-SERVER	({DL,EC}REDP-1 or REDP-2), {DL,EC}PKGP-DH, {DL,EC}SVDP-PAK2	KDF1, KDF2
{DL,EC}BPKAS-SPEKE-CLIENT, {DL,EC}BPKAS-SPEKE-SERVER	({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-SPEKE, {DL,EC}SVDP-SPEKE	KDF1, KDF2
Augmented Password-authenticated Key Agreement Schemes		
{DL,EC}APKAS-AMP-CLIENT	{DL,EC}PKGP-DH, {DL,EC}SVDP-AMP-CLIENT	KDF1, KDF2
{DL,EC}APKAS-AMP-SERVER	{DL,EC}PVDGP-AMP, {DL,EC}PEPKGP-AMP-SERVER, {DL,EC}SVDP-AMP-SERVER	KDF1, KDF2
{DL,EC}APKAS-BSPEKE2-CLIENT	({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-SPEKE, {DL,EC}SVDP-SPEKEK	KDF1, KDF2
{DL,EC}APKAS-BSPEKE2-SERVER	{DL,EC}PVDGP-BSPEKE2, ({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-SPEKE, {DL,EC}SVDP-SPEKEK	KDF1, KDF2
{DL,EC}APKAS-WSPEKE-CLIENT	({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-SPEKE, {DL,EC}SVDP-WSPEKE-CLIENT	KDF1, KDF2
{DL,EC}APKAS-PAKZ-CLIENT	{DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-PAK, {DL,EC}SVDP-PAK1-CLIENT	KDF1, KDF2
{DL,EC}APKAS-PAKZ-SERVER	{DL,EC}PVDGP-PAKZ, ({DL,EC}RE에-1 or REDP-2), {DL,EC}PKGP-DH, {DL,EC}SVDP-PAK2	KDF1, KDF2
DLAPKAS-SRP3-CLIENT	DLPKGP-SRP3-CLIENT, DLPVDGP-SRP3, DLSVDP-SRP3-CLIENT, DLPVDGP-SRP3	KDF1, KDF2
DLAPKAS-SRP3-SERVER	DLPVDGP-SRP3, DLPEPKGP-SRP3-SERVER, DLSVDP-SRP3-SERVER	KDF1, KDF2
ECAPKAS-SRP5-CLIENT	ECPVDGP-SRP5, ECPKGP-DH, ECSVDP-SRP5-CLIENT	KDF1, KDF2
ECAPKAS-SRP5-SERVER	ECPVDGP-SRP5, ECPEPKGP-SRP5-SERVER, ECSVDP-SRP5-SERVER	KDF1, KDF2
DLAPKAS-SRP6-CLIENT	DLPVDGP-SRP6, DLPKGP-SRP6-CLIENT, DLSVDP-SRP6-CLIENT	KDF1, KDF2
DLAPKAS-SRP6-SERVER	DLPVDGP-SRP6, DLPEPKGP-SRP6-SERVER, DLSVDP-SRP6-SERVER	KDF1, KDF2
Password-authenticated Key Retrieval Schemes		
{DL,EC}PKRS-1-CLIENT	({DL,EC}REDP-1 or REDP-2), {DL,EC}PEPKGP-1, {DL,EC}SVDP-1	KDF1, KDF2
{DL,EC}PKRS-1-SERVER	{DL,EC}SVDG-1	KDF1, KDF2

에 대하여 논한다. P1363을 시작으로 논의된 패스워드 기반 공개키 암호 키동의 프로토콜은 P1363.a를 거쳐 2000년 후반부터 P136.2를 통해 표준화로 정의되고 있다[9,11]. IEEE P1363.2에서는 논의된 패스워드 기반 키동의에 사용된 방식 및 구체적인 세부사항을 정리한 것이 표 1에 기술되어 있다.

표 1에서 첫 번째 분류인 BPKAS(Balanced Password-authenticated Key Agreement Schemes)는 평문등가 구조로 이루어진 방식들을 지칭한다. 앞서 설명한 바와 같이 이 방식들은 두 참여자(클라이언트와 서버)가 공통의 패스워드를 공유한 이후, 각기 다른 참여자에게서 그 패스워드에 대하여 알고 있음을 증명함으로서 인증이 이루어진다. 이에 따라 서버와 클라이언트의 세부 동작이 일치할 수도 있다. 두 번째 분류 APKAS (Augmented Password-authenticated Key Agreement Schemes)는 검증자 기반 구조로 이루어진 방식들을 지칭한다. 이 방식에 있어서 서버는 패스워드로부터 유도된 패스워드 검증데이터를 기억하게 된다. 세 번째 분류 PKRS(Password-authenticated Key Retrieval Schemes)는 패스워드 검증데이터를 분할한 후에 정족수(threshold)를 만족시키지 못한 경우엔 어떠한 정보도 노출되지 않도록 하는 방법으로 활용된다.

2.5.1 표에 사용된 약어에 대한 용어와 동작

① DL : discrete logarithm, 이산대수 기반 공개키 암호
② EC : elliptic curve, 타원곡선 기반 공개키 암호
③ CLIENT와 SERVER : 통상적인 클라이언트와 서버를 지칭한다. 단 표에서는 각 참여자가 수행하여야 하는 동작을 구별하여 준다.
④ REDP : random element derivation primitive, 기본 파라메터들과 랜덤하게 선택된 값을 입력으로 군(group)의 원소를 유도하는 함수이다.
⑤ PEPKGP : password-entangled public-key generation primitives, 이 동작은 패스워드가 지수부에 첨부된 상태에서 랜덤하게 공개키를 생성하는 함수이다.
⑥ SVDP : secret value derivation primitive, 이 동작은 인증이 이루어진 이후에 임의의 세션 비밀값을 생성하는 함수이며 BPKAS, APKAS, 그리고 PKRS 모두 공통적으로 수행하는 사항이다.
⑦ PVDGP : password verification data generation primitive, 이 동작은 패스워드 값을 입력받아 패스워드 검증 데이트를 생성하는 함수로서 APKAS에만 적용된다.
⑧ PKGP : public key generation primitive, 이 동작은 랜덤한 임시 공개키를 생성하는 함수이다.
⑨ KDF : key derivation function, 키유도 함수.

KDFs는 ANSI X9.63에 기술된 사항을 토대로 정의된 새로운 키유도 함수이며 IEEE Std 1363-2000에서의 새로운 서명 및 암호화 방식을 지원하기 위해 정의된 함수이다. 키유도 함수는 키동의 방식, 암호화 방식, 그리고 엔코딩(encoding) 방식에 대한 기본 지침으로서의 표준화 항목이다. 키유도 함수의 주요 목적은 두 참여자간에 동의된 비밀값 및 사전에 알려진 파라메터들로부터 하나 이상의 공유된 비밀키를 계산하는 것에 있다. 이 유도된 비밀키는 일반적으로 대칭형 암호알고리즘의 비밀키로 사용된다.[12]

보안 프로토콜 수행시 부적당한 키유도 함수의 사용은 보안상에 있어서 타협(compromise)을 유발할 수 있다. 따라서 IEEE Std 1363-2000 표준에서는 IEEE P1363a 제시된 KDF1 및 KDF2 방식을 사용하도록 강하게 권고하고 있다. 물론 다른 엔코딩 방식 및 다른 키유도 함수가 보안 프로토콜에 적용될 수도 있다. 그러나 이와 같은 경우에 있어서 키유도 함수는 또 다른 보안 분석을 요구하기 때문에 IEEE Std 1363-2000에서는 확실히 권장하지 않았다.[12]

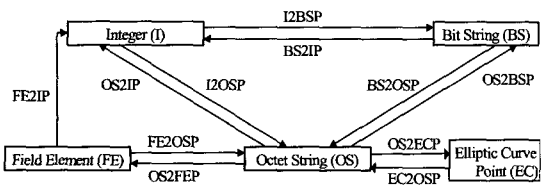
- KDF1 : IEEE P1363.2의 KDF1은 일반적인 키유도 함수로 동의된 비밀값과 파라메터를 입력으로 해쉬함수(*Hash*)의 길이만큼의 출력한다. 동의된 비밀값은 *zLen* 옥텟(octet) 길이의 옥텟 스트링 Z 또는 이에 해당되는 *zBits* 비트 길이의 비트 스트링 ZB로 표현된다(단 $zBits = 8zLen$). 입력된 파라메터 역시 *pLen* 옥텟 길이의 옥텟 스트링 P 또는 이에 해당되는 *pBits* 비트 길이의 비트 스트링 PB로 표현된다. 출력된 공유 비밀키는 다음과 같다: $K = h(ZB \| PB)$.

해쉬함수로는 FIPS PUB 180-2에 정의된 SHA-1, SHA-256, SHA-384, SHA-512, 및 RIPEMD-160을 이용한다.

- KDF2 : KDF2는 ANSI X.9.42:2001과 ANSI X9.63에서의 구성을 바탕으로 하고 있으며, DL/EC 키동의 방식과 DL/EC 암호화 방식을 사용함에 IEEE Std 1363-2000에서 권고되는 방식이다. KDF2는 KDF1과는 달리 출력 길이에 대한 정보 *oBits*를 입력으로 받아서 그 길이만큼 출력하는 함수이다. 주어진 입력값(동의된 비밀값, 32비트의 출력 길이 정보, 파라미터)을 해쉬하여 구하지만 카운터가 포함되어 해쉬함수의 출력길이 이상의 공유 비밀키를 만들어 낼 수 있다. 여기에서 CB_i는 정수 i를 32비트로 표현한 비트열이다.

$$HB_i = h(ZB \| CB_i \| PB),\ K = HB_1 \| \cdots$$

⑩ 데이터 타입 변환 프리미티브(primitive).[12]
IEEE P1363.2에서 사용된 데이터 타입 변환은 다음과 같이 요약된다.
더불어 GE2OSP-X(Group Elements to Octet Strings)는 FE2OSP 또는 EC2OSP를 의미한다.



⑪ KCF1 : key confirmation function, 키확신 함수.[14]
KCF는 동의된 비밀값 및 관련 데이터를 다른 참여자에게 확신시키기 위한 메시지 생성에 그 목적을 둔다. 구성된 데이터에는 키동의 과정에서 교환된 공개키 값 및 사전-공유된 패스워드 데이터 등을 주요 입력값으로 취한다. 정확한 입력 정보로는 키유도 파라메터 옥텟 스트링 P, 클라이언트 및 서버의 공개키 w_A/w_B, 공유된 키 옥텟 스트링 Z, 공유된 패스워드 옥텟 스트링 o_π을 받아들여, 출력값 $o = h(P \| o_A \| o_B \| o_\pi)$을 계산한다. 여기에서 o_A/o_B는 각각 GE2OSP(w_A)/GE2OSP(w_B)이다.

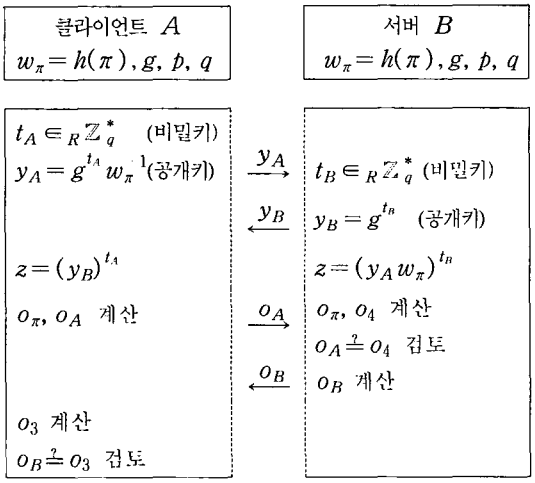
⑫ AMP, PAK, PPK, SPEKE, SRP : 각각 패스워드 기반 인증 프로토콜로써 제안된 방식이며, 상세 사항은 [11]을 참조

다음 절에서는 IEEE P1363.2에 기술된 키동의를 위한 패스워드 기반 공개키 암호 방식 중에서 3가지 분류(BPKAS, APKAS, PKRS)에 대한 대표적인 방식 3가지에 대하여 간략히 설명한다. 설명에 앞서 사용된 용어 및 기호들에 대하여 다음과 같이 정의한다.

- π : 패스워드, 또는 패스워드로부터 유도된 값
- p : 체(field)의 크기
- q : $p-1$을 나누어 떨어뜨리는 소수인 위수(order)
- g : 원시근, $GF(p)$에서 위수 q의 원소
- $h()$: 해쉬함수
- t_A, t_B : 클라이언트 A 및 서버 B의 임시 개인키
- y_A, y_B : 클라이언트 A 및 서버 B의 임시 공개키
- z, z_1, z_2 : 공유된 비밀 값, $GF(q)$의 원소들, 원시 비밀 그룹 원소에서 유도된 비밀값
- Z, Z_1, Z_2 : 공유된 비밀값, 키동의에서 수행된 옥텟 스트링
- o_A, o_B : 클라이언트의 키확신에서 수행된 옥텟 스트링, 서버의 키 확신에서 수행된 옥텟 스트링
- $K_1, K_2, \cdots$: 프로토콜 수행 후 공유된 비밀키
- $P_1, P_2, \cdots$: 동의된 키를 유도한 키유도 파라메터 (key derivation parameter)

2.5.2 BPKAS-PAK

BPKAS-PAK은 [4]에 제안된 방식에 바탕을 두고 있으며, EC/DL 방식을 사용한다.[11] 프로토콜 수행이 앞서 두 참여자 클라이언트 및 서버는 패스워드 옥텟 스트링 π을 공유하여야 하며, EC/DL 방식에 필요한 파라메터들을 사전에 정의하여야 한다. 또한 키유도 함수 KDF 및 REDP 함수를 결정하여야 한다.



(그림 5) BPKAS-PAK 프로토콜

BPKAS-PAK는 사전등록 단계, (클라이언트/서버)키동의 단계, 그리고 (클라이언트/서버)에서의 키확신 단계로 구별되며 각각의 상세 사항은 다음과 같은 절차로 처리된다. BPKAS-PAK에 대한 상세 프로토콜은 그림 5에 도시되어 있다.

○ 사전등록 단계 : BPKAS-PAK는 평문등가 형태의 패스워드 기반 인증 프로토콜로써 클라이언트와 서버 간 사전 등록단계를 거친 후 기억하는 지식은 동일한 형태이다.

1. 원 패스워드(pwd)를 그대로 사용하지 않고 가공된 패스워드($\pi = salt||pwd||ID_s$)로 변형한다.
2. 패스워드를 해쉬, $w_m = h(\pi)$

○ 클라이언트에서의 키동의
({DL,EC}PEPKGP-PAK)
1. 비밀값 t_A를 랜덤하게 생성
2. 클라이언트의 공개키 $y_A = g^{t_A} w_\pi^{-1}$ 계산
3. 서버에게 y_A를 송신

({DL,EC}SVDP-PAK1-CLIENT)
4. 서버로부터 서버의 공개키 y_B를 수신
5. 만일 서버의 공개키 y_B가 정의된 군(group)의 원소가 아니면 프로토콜 중지
6. (t_A, w_m)을 이용하여 체의 원소 z를 계산
7. $Z = FE2OSP(z)$, 유한체 원소(finite field element)를 옥텟 스트링으로 변환
8. 각 키유도 파라메터 P_i 및 공유된 비밀 옥텟 스트링 Z을 통해 공유된 비밀키 $K_i = KDF(Z, P_i)$을 계산
9. 비밀키는 $K_1, K_2, \cdots, K_t$가 됨

○ 서버에서의 키동의
({DL,EC}PKGP-DH)
1. 비밀값 t_B를 랜덤하게 생성
2. 서버의 공개키 $y_B = g^{t_B}$ 계산
3. 서버에게 y_B를 송신

({DL,EC}SVDP-PAK2)
4. 클라이언트로부터 y_A를 수신
5. 만일 y_A가 정의된 군(group)의 원소가 아니면 프로토콜 중지
6. (t_B, w_m, y_A)을 이용하여 체의 원소 z를 계산
7. $Z = FE2OSP(z)$, 유한체 원소(finite field element)를 옥텟 스트링으로 변환
8. 각 키유도 파라메터 P_i 및 공유된 비밀 옥텟 스트링 Z을 통해 공유된 비밀키 $K_i = KDF(Z, P_i)$을 계산
9. 비밀키는 $K_1, K_2, \cdots, K_t$가 됨

○ 클라이언트에서의 키확신
1. 서버로부터 옥텟 스트링 o_B를 수신
2. $o_\pi = GE2OSP-X(w_m)$를 계산
3. $o_3 = KCF1(hex(03), y_A, y_B, o_\pi)$를 계산
4. 서버로부터 수신한 o_B값과 클라이언트가 계산한 o_3값을 비교. 만일 $o_3 \neq o_B$라면 키확신 프로토콜 중지
5. $o_A = KCF1(hex(04), y_A, y_B, o_\pi)$를 계산

6. 서버에게 옥텟 스트링 o_A를 송신

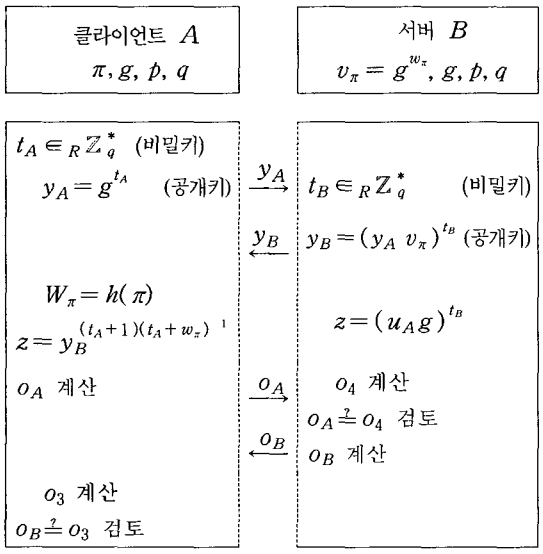
○ 서버에서의 키확신

1. $o_\pi = GE2OSP - X(w_m)$를 계산
2. $o_B = KCF1(hex(03), y_A, y_B, Z, o_\pi)$를 계산
3. 클라이언트에게 o_B송신
4. 클라이언트로부터 o_A를 수신
5. $o_4 = KCF1(hex(04), y_A, y_B, Z, o_\pi)$를 계산
6. 클라이언트로부터 받은 o_A와 서버가 계산한 o_4 값을 비교, 만일 $o_4 \neq o_A$라면 키확신 프로토콜 중지

2.5.3 APKAS-AMP

APKAS-AMP(Augmented Password-Authenticated Key Agreement Scheme)는 [13]에서 제안된 방식에 기반을 두고 있으며, EC/DL 기술을 사용한다.[11]

{DL,EC}APKAS-AMP-SERVER은 패스워드 기반 공개키 v_π를 사용하며, 이 값은 워드 π를 입력값으로 넣어 유도된(계산된) 값인 패스워드 검증데이터이다. APKAS-AMP에 대한 상세 프로토콜은 그림 6에 도시되어 있다.



(그림 6) APKAS-AMP 프로토콜

v_π는 {DL,EC}APKAS-AMP-CLIENT에서 사용된 패스워드 기반의 옥텟(octet string)값에 대응된다.

APKAS-AMP는 사전등록 단계, (클라이언트/서버)키동의 단계, 그리고 (클라이언트/서버)에서의 키확신 단계로 구별되면 각각의 상세 사항은 다음과 같은 절차로 처리된다.

○ 사전등록 단계 : 클라이언트와 서버 간 사전 등록 단계는 다음과 같다.

(PVDGP-AMP)

1. 패스워드를 해쉬, $w_\pi = h(\pi)$
   평문의 패스워드(pwd)를 사용하지 않고 가공된 패스워드($\pi = salt \| pwd \| ID_s$)를 이용
2. 패스워드 검증데이터를 생성. $v_\pi = g^{w_\pi}$
3. 패스워드 검증데이터를 서버에게 안전하게 전송

○ 클라이언트에서의 키동의

({DL,EC}PKGP-DH)

1. 비밀값 t_A를 랜덤하게 생성
2. 클라이언트의 공개키 $y_A = g^{t_A}$ 계산
3. 서버에게 y_A를 송신
4. 서버로부터 서버의 공개키 y_B를 수신

({DL,EC}PVDGP-AMP)

5. 패스워드 기반 개인키 w_π를 계산

({DL,EC}SVDP-AMP-CLIENT)

6. (t_A, w_π, y_B)을 이용하여 체의 원소 z를 계산
7. $Z = FE2OSP(z)$, 유한체 원소를 옥텟 스트링으로 변환
8. $K_i = KDF(Z, P_i)$
9. 출력으로 유도된 키는 $K_1, K_2, \cdots, K_t$가 됨

○ 서버에서의 키동의

1. 비밀값 t_B를 랜덤하게 생성
2. 클라이언트로부터 y_A를 수신

({DL,EC} PEPKGP-AMP-SERVER)

3. 패스워드와 통합된 서버의 공개키 y_B를 계산
4. y_B를 클라이언트에게 송신

({DL,EC} SVDP-AMP-SERVER)

5. (t_B, y_A)을 이용하여 체의 원소 z를 계산
6. $K_i = KDF(Z, P_i)$
7. 출력으로 유도된 키는 $K_1, K_2, \cdots, K_t$ 가 됨

○ 클라이언트에서의 키확신

1. $o_A = KCF1(hex(04), y_A, y_B, Z, \text{“ ”})$를 계산
2. 서버에게 옥텟 스트링 o_A를 송신
3. 서버로부터 옥텟 스트링 o_B를 수신
4. $o_3 = KCF1(hex(03), y_A, y_B, Z, \text{“ ”})$를 계산
5. 서버로부터 수신한 o_B값과 클라이언트가 계산한 o_3값을 비교, 만일 $o_3 \neq o_B$라면 키확신 프로토콜 중지

○ 서버에서의 키확신

1. 클라이언트로부터 o_A를 수신
2. $o_4 = KCF1(hex(04), y_A, y_B, Z, \text{“ ”})$를 계산
3. 클라이언트로부터 받은 o_A와 서버가 계산한 o_4 값을 비교, 만일 $o_4 \neq o_A$라면 키확신 프로토콜 중지
4. $o_B = KCF1(hex(03), y_A, y_B, Z, \text{“ ”})$ 계산
5. 클라이언트에게 o_B송신

APKAS-AMP 방식에서는 서버가 클라이언트의 패스워드 검증데이터를 가지고 있으며, 서버는 검증 데이터를 가지고 클라이언트와 키동의 과정을 수행하고 상호인증절차를 수행한다. 키동의 과정에서 서버와 클라이언트는 {DL,EC}PVDGP-AMP 형태로 수행하고 검증된 데이터를 이용하여 PEKGP-AMP-SERVER 형태의 공개키를 사용하므로 오프라인 공격이나 패스워드 추측공격에 저항할 수 있다.

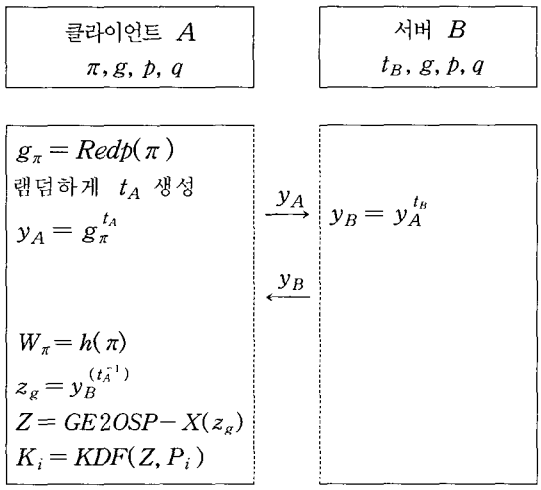
패스워드로부터 유도된 랜덤한 비밀값을 사용하는 SVDP-AMP-CLIENT, SVDP-AMP-SERVER 형태를 사용하기 때문에 세션키가 노출되었다 하더라도, 공격자가 그 동안 도청한 정보들을 기반으로 사용자의 패스워드에 대한 정보나 이후 진행될 세션키에 대한 어떠한 정보도 얻을 수 없다.

2.5.4 PKRS-1

{DL,EC}PKRS-1{CLIENT,SERVER}은 인증된 패스워드 키복구 기술(Password-Authenticated Key Retrieval Scheme) version 1이며 [15]에서 제안한 방식에 바탕을 두고 있다.[11]

PKRS는 임의의 참여자가 패스워드에 대응되는 비밀키를 소유하고, 다른 참여자의 도움을 받아 패스워드를 이용하여 고정된 키를 협상하여 새로운 고정된 키를 생성한다.

PKRS에 대한 상세 사항이 그림 7에 도시되어 있다.



(그림 7) PKRS-1 프로토콜

○ 클라이언트에서의 사전등록 단계

PKRS는 두 참여자간에 키 협상이 이루어지기 전에 패스워드 기반 옥텟스트링에 대한 사전등록단계를 거친다.

1. 원래의 패스워드(pwd)를 그대로 사용하지 않고 가공된 패스워드($\pi = salt \| pwd \| ID_s$)를 구하고

패스워드 옥텟 스트링을 클라이언트를 클라이언트에 사전 등록한다.

2. *Redp*는 {DL,EC}REDP-1, {DL,EC}REDP-2를 사용하는 REDP 함수를 정의

○ 서버에서의 사전등록 단계

1. 클라이언트의 π에 대응하는 서버의 개인키 u값을 등록
2. 서버의 개인키 t_B를 랜덤하게 생성

○ 클라이언트에서의 키 협상단계

1. $g_\pi = Redp(\pi)$ 계산
2. 개인키 t_A를 랜덤하게 생성 ({DL, EC}KRBP-1)
3. Blind 패스워드 기반 공개키 {DL, EC}KRBP-1 (t_A, g_π)을 사용하여 $y_A = g_\pi^{t_A}$를 계산
4. y_A를 서버에 송신
5. 서버로부터 치환된 Blind 공개키 y_B를 수신 ({DL, EC}KRUP-1)
6. 치환된 비밀 키 $z_g = y_B^{(t_A^{-1})}$를 계산
7. 옥텟 스트링 $Z = GE2OSP - X(z_g)$를 계산
8. 각 키유도 파라메터 P_i 및 공유된 비밀 옥텟 스트링 Z을 통해 공유된 비밀키 $K_i = KDF(Z, P_i)$을 계산
9. 협상된 비밀키는 $K_1, K_2, \cdots, K_t$가 됨

○ 서버에서의 키 협상단계

1. 클라이언트로부터 Blind 패스워드기반 공개키 y_A를 수신 ({DL,EC}KRPP-1)
2. {DL,EC}KRPP-1 (u, y_A)를 이용하여 Blinded 공개키 y_B를 계산
3. 클라이언트에게 y_B를 송신

3. 결론 및 연구 방향

본 논문에서는 패스워드 기반 키교환 프로토콜에 대한 소개, 패스워드 기반 인증 프로토콜의 여러 방법들 중 몇 가지를 분석 및 IEEE.1363 표준안 관련 현황 분석을 하였다. 패스워드를 이용한 인증방법은 사용자가 가장 편리하게 자신을 증명할 수 있는 방법으로 가장 많이 사용되고 있다. 패스워드 인증은 하드웨어 토큰 및 생체 인식 기술을 이용한 방법이 완전하게 자리 잡기 전까지 다양한 환경에 적합한 방식이라고 할 수 있다. 앞으로 키 복구 기능을 포함한 프로토콜에 대한 연구와 병행하여 키교환 과정에 적용될 수 있기를 기대한다.

참고문헌

[1] T. Wu, "Secure remote password protocol," Internet Society Symposium on Network and Distributed System Security, 1998

[2] P. MacKenzie, T. Shrimpton, and M. Jakobsson, "Threshold Password-Authenticated Key Exchange," CRYPTO 2002

[3] M. Bellare and P. Rogaway, "The AuthA protocol for password-based authenticated key exchange," 사이트 http://grouper.ieee.org/groups/1363/passwdPK/

[4] V. Boyko, P. MacKenzie, and S. Patel, "Provably Secure Password Authenticated Key Exchange Using Diffie-Hellman," Eurocrypt 2000

[5] Xunhua Wang, "Intrusion-Tolerant Password Enabled PKI," 2nd annual PKI Research Workshop, 2002

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

[7] David P. Jablon, "Strong Password Only Authenticated Key Exchange", Computer Communication Review, ACM SIGOMM, vol. 26, no. 5, PP. 5-26, 1996

[8] A. Menezes, P. van Oorschot, S. Vanston, "Handbook of applied cryptography," CRC Press, Inc., pp 618, 1997

[9] 손기욱, 서인석, 원동호, "패스워드 기반 키분배 프로토콜 표준화 동향", 한국정보보호학회 학회지 제12권 제4호, pp. 46-54, 2002. 8

[10] 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 Labs, 1994.

[11] IEEE P1363.2(Draft 13), "Standard Specifications for Password-Based Public-Key Cryptographic Techniques", 2004. 3, 사이트 http://grouper.ieee.org/groups/1363/.

[12] IEEE 1363-2000, "Standard Specifications For Public Key Cryptography", 2004. 3, 사이트 http://grouper.ieee.org/groups/1363/.

[13] Taekyoung Kwon, "Authentication via Memorable Password-Revised Submission to IEEE P1363.2", Submission to the P1363 Working Group, October 3, 2002

[14] P. Mackenzie, "The PAK suite: Protocols for Password-Authenticated Key Exchange", a P1363.2 submission to the IEEE P1363 Working Group, April, 2002

[15] W. Ford & Kaliski, "Server-Assisted Generation of a Strong Secret from a Password", proceedings of the IEEE 9th International Workshops on Enabling Technologies : Infrastructure for Collaborative Enterprises, NIST, Gaithersburg MD, June 14-16, 2000

◐ 저 자 소 개 ◑

심 현 정

1999년 공주대학교 산업정보학과 졸업
2002년 순천향대학교 산업정보대학원 전자상거래학과 석사
2002~현재 : 순천향대학교 일반대학원 정보보호학과 박사과정
관심분야 : 암호이론, 공개키 기반구조, 보안프로토콜

김 배 완

2002년 순천향대학교 전자공학과 졸업
2002년 3월~현재 : 순천향대학교 일반대학원 정보보호학과 석사과정
관심분야 : 공개키 기반구조, 보안프로토콜

류 종 호

1998년 순천향대학교 전자공학과 졸업
2000년 순천향대학교 일반대학원 전기·전자공학과 석사
2004년 순천향대학교 일반대학원 전기·전자공학과 박사
2004년~현재 인터넷침해대응기술 연구센터(IIRTRC) 연구원
관심분야 : 암호이론, 공개키 기반구조, 보안 프로토콜

엄 흥 열

1981년 한양대학교 전자공학과 졸업
1983년 한양대학교 대학원 전자공학과 석사
1990년 한양대학교 대학원 전자공학과 박사
1982년~1990년 한국전자통신연구소 선임연구원
1990년~현재 : 순천향대학교 공과대학 정보보호학과 교수
1997년~2000년 : 순천향대학교 산업기술연구소 소장
2000년~현재 : 순천향대 산학연컨소시엄센터 소장
1997년~현재 : 한국정보보호학회 총무이사, 학술이사, 교육이사
관심분야 : 네트워크 보안, 전자상거래 보안, 공개키 기반 구조, 부호이론, 이동통신보안