情報保護學會論文誌
第18卷 第5號, 2008. 10

서로 다른 패스워드를 사용하는 두 사용자를 위한 경량 패스워드 기반 키 교환 프로토콜*

권 정 옥†, 김 기 탁, 정 익 래, 이 동 훈‡
고려대학교 정보경영공학전문대학원

Light-Weight Password-Based Authenticated Key Exchange for Two Users using Different Passwords*

Jeong Ok Kwon†, Ki Tak Kim, Ik Rae Jeong, Dong Hoon Lee‡
Graduate School of Information Management & Security CIST, Korea University

요 약

본 논문에서는 두 사용자들 사이에 패스워드를 공유하고 있지 않은 환경에서 세션 키(session key)를 공유할 수 있는 패스워드 기반 키 교환 프로토콜을 제안한다. 제안 프로토콜에서 두 사용자들은 서버에 자신의 패스워드를 등록한 후, 서버의 도움을 받아 동일한 세션 키를 공유하게 된다. 제안 프로토콜은 랜덤오라클(random oracle)을 사용하지 않고 전방향 안전성(forward secrecy)을 만족하는 프로토콜로써, 기존 랜덤오라클을 사용하는 프로토콜과 비교했을 때 효율성면에서 큰 차이가 없다. 제안 프로토콜에서는 인간이 기억하기 쉬운 패스워드만을 사용하고 프로토콜을 수행하는데 필요한 다른 모든 정보는 공개된 정보이다.

ABSTRACT

In the paper, we consider password-based authenticated key exchange with different passwords, where the users do not share a password between themselves, but only with the server. The users make a session key using their different passwords with the help of the server. We propose an efficient password-based authenticated key exchange protocol with different passwords which achieves forward secrecy without random oracles. In fact this amount of computation and the number of rounds are comparable to the most efficient password-based authenticated key exchange protocol in the random oracle model. The protocol requires a client only to memorize a human-memorable password, and all other information necessary to run the protocol is made public.

Keywords : Password-based key exchange, Dictionary attacks, Forward secrecy, Known-key secrecy

접수일 : 2008년 3월 12일; 채택일 : 2008년 7월 18일
* 이 연구에 참여한 연구자 중 일부는 '2단계 BK21사업'의 지원비를 받았으며 다른 일부는 '지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업 (IITA-2008-(C1090-0801-0025))'의 지원비를 받았음.
† 주저자, pitapat@korea.ac.kr
‡ 교신저자, donghlee@korea.ac.kr

Ⅰ. 서 론

안전하지 않은 인터넷과 같이 공개된 네트워크에서 안전하게 통신을 하기 위해서는 세션 키(session key)를 안전하게 교환하는 것이 필수적이라 할 수 있다. 그 후에 공유된 세션 키는 기밀성(confidentiality) 또는 데이

터 무결성(integrity)과 같은 암호학적인 안전성 목적을 위해 사용되어진다.

두 개체(two-party) 간 안전한 통신을 위한 효율적인 PAKE 프로토콜은 지난 몇 년 동안 광범위하게 연구되어져 왔다. 기존 제안되었던 대부분의 이자 간 PAKE 프로토콜들은 두 명의 참여자(흔히 사용자)가 사전에 공통된 패스워드를 공유한다고 가정한다. 따라서 이러한 모델에서 사용자에게는 통신하고자 하는 개체의 수만큼의 패스워드를 기억하여야하는 어려움이 따른다.

서로 다른 패스워드를 사용하는 두 사용자간 PAKE 프로토콜은 앞서 언급한 이자간 PAKE 프로토콜의 문제점을 해결하기 위해 기 제안된 모델이다. 이러한 프로토콜 모델에서 사용자는 신뢰할 수 있는 서버에 패스워드를 등록하게 된다. 그러면 신뢰할 수 있는 서버는 두 참여자를 인증하고, 서로 다른 패스워드를 가지고 있는 두 사용자들이 공통된 세션 키를 공유할 수 있도록 도와주는 역할을 한다. 즉, 각 사용자는 신뢰할 수 있는 서버와 단 한 개의 패스워드만 공유하면 된다. 프로토콜에 세 개체가 참여한다는 의미에서 이러한 패스워드 기반 키 교환 프로토콜을 흔히 삼자(three-party) 간 PAKE 프로토콜이라 부른다. 삼자 간 PAKE 프로토콜은 사용자가 기억해야 하는 패스워드의 수를 줄여주기 때문에 불특정 개체들끼리 안전한 통신을 가능하게 한다는 장점이 있다. 그러나 두 명의 참여자가 세션 키를 공유하게 하기 위해서는 서버가 프로토콜에 반드시 참여해야 한다.

1.1 삼자간 PAKE 프로토콜의 안전성 요구 조건

패스워드 추측공격 : PAKE 프로토콜의 안전성 모델에서는 온라인과 오프라인 패스워드 추측공격(on-line/off-line password guessing attack)에 대한 안전성을 반드시 고려해야 한다. 삼자 간 PAKE 프로토콜과 같이 서버의 도움을 받아야 하는 PAKE 프로토콜에서는 온라인 패스워드 추측 공격을 더욱 주의하여 고려해야 한다. 왜냐하면 악의를 가진 사용자가 서버를 상대 사용자의 패스워드를 검증하기 위한 오라클(oracle)로 사용하여 온라인 패스워드 추측 공격을 시도할 수 있기 때문이다. 만약 악의를 가진 사용자에 의한 상대 사용자의 패스워드에 대한 추측 공격이 서버에 의해 탐지될 수 없고 기록될 수 없다면 이러한 공격을 탐지할 수 없는 온라인 패스워드 추측 공격(undetectable password guessing attack)이라고 부른다[11]. 탐지할 수 없는 온라인 패스워드 추측 공격을 방어하기 위해서 서버의 도움을 받는 PAKE 프로토콜은 프로토콜이 악의적으로 수행되는지 올바르게 수행되는지를 서버 측에서 구별할 수 있는 방법을 제공해야 한다. PAKE 기법의 주된 안전성의 목표는 공격자가 탐지할 수 있는 온라인 패스워드 추측 공격만을 할 수 있도록 제한하는 것이다. 왜냐하면 이러한 공격은 패스워드 입력 횟수를 제한함으로써 쉽게 방어될 수 있기 때문이다.

키 기밀성 : 키 교환 프로토콜의 가장 기본적인 안전성 요구사항은 키 기밀성(key secrecy)이다. 공격자는 정직한 개체 간의 통신을 도청하거나, 메시지를 위조하거나 변조하여 전송 할 수 있다. 키 기밀성은 이러한 공격자가 세션 키에 대한 어떠한 정보도 얻을 수 없어야 함을 의미한다. 키 교환 프로토콜에 요구되는 다른 안전성은 전방향 안전성(forward secrecy)과 기지 키 공격에 대한 안전성(known-key secrecy)이다. 이에 대한 형식적인 정의(formal definition)는 Ⅱ장에서 다룬다.

서버의 도움이 요구되는 삼자 간 PAKE 프로토콜에서 추가적으로 요구되는 안전성은 호기심 많은 서버(curious server) 또는 악의적인 서버(malicious server)에 대한 키 기밀성이다. 이것은 서버가 두 사용자들이 공통된 세션 키를 공유하도록 도와주는 역할을 하지만, 두 사용자들 간의 통신을 도청할 경우에 세션 키에 대한 어떠한 정보도 없을 수 없어야 함을 의미한다. 따라서 서버에 대한 키 기밀성이 제공된다면, 서버가 도청을 통해 사용자들의 세션 키를 알아낼 수 있는 경우보다 더 강한 안전성을 제공하게 되고, 사용자들의 서버에 대한 신뢰 의존도를 낮출 수 있다. 서버는 사용자들의 패스워드를 가지고 있기 때문에 물론 정당한 사용자를 가장할 수 있다. 따라서 이 모델에서는 서버를 수동적 공격자(passive adversary)로 가정한다.

1.2 기존 연구 및 연구 결과

삼자 간 PAKE 프로토콜은 지난 몇 년 동안 광범위하게 연구되어 왔다[3,5,6,17,19]. 그 중에서 C2C-PAKE 프로토콜은 정형화 된 안전성 증명(formal

security proof) 없이 제안되었다[5]. 이 프로토콜은 블록 암호(block cipher)를 이상적인 암호(ideal cipher)로 가정하였을 때 안전하다고 여겨진다. 그러나 C2C-PAKE는 탐지할 수 없는 온라인 패스워드 추측 공격에 안전하지 않다[16].

Abdalla와 그 외는 삼자 간 PAKE 프로토콜에 대한 정형화된 안전성 모델(security model)과 안전한 이자 간 PAKE 프로토콜을 삼자 간 PAKE 프로토콜로 변환할 수 있는 일반적인(generic) 설계방법인 GPAKE를 제안하였다[3]. GPAKE는 이자 간 PAKE 프로토콜과 삼자 간 키 분배 프로토콜을 하위 프로토콜(sub-protocol)로 이용하여 삼자 간 PAKE 프로토콜을 만든다. GPAKE의 안전성은 표준모델(standard model)에서 증명되었지만, 전방향 안전성(forward secrecy)에 대한 정형적인 증명은 제시되지 않았다. 또한 GPAKE는 하위 프로토콜들이 블랙박스(black box)처럼 모듈(module) 단위로 다루어지기 때문에 GPAKE의 구체적인 예시 프로토콜은 라운드(round)와 계산량 복잡도 관점에서 볼 때 최적화된 것은 아니다. 예를 들면, GPAKE로부터 만들어진 가장 효율적인 프로토콜이라도 6 라운드와 각 사용자 마다 17번 이상의 모듈로(modulo) 지수승 연산을 필요로 한다. 모바일 기기는 작은 양의 계산과 통신, 작은 수의 라운드로 세션 키를 만들 수 있어야 하기 때문에 GPAKE에 의해 만들어진 프로토콜은 계산능력이 제한된 모바일 기기에 사용하기에 적합하지는 않다.

GPAKE의 효율성을 개선하기 위해, Abdalla와 그 외는 효율적인 AP 프로토콜을 제안하였다[6]. AP 프로토콜은 결정적 디피-헬만(decisional Diffie-Hellman) 문제의 어려움에 기반하여 이상적인 해쉬모델(ideal hash model), 흔히 랜덤 오라클(random oracle) 모델로 불리는 모델에서 안전성이 증명되었지만, 전방향 안전성에 대한 정형화된 증명은 제시되지 않았다. AP 프로토콜은 C2C-PAKE 프로토콜에서와 같이 서버가 온라인 패스워드 추측 공격을 알아차릴 수 없기 때문에 탐지할 수 없는 온라인 패스워드 추측 공격에 대하여 안전하지 않다. 게다가, 만약 이상적인 암호나 이상적인 해쉬 함수와 같은 이상화된 함수 대신에 실제 함수를 사용하면 이상화된 암호 또는 해쉬 모델에서는 안전한 스킴이 실제 환경에서는 안전하지 않다고 잘 알려진 경우들이 있다[7,9,10,13,18]. 따라서 이상화된 함수의 사용이 없이 표준모델에서 안전하면서도 효율적인 삼자 간 PAKE 프로토콜의 설계가 요구된다.

본 논문에서는 통신량과 계산량 그리고 라운드의 복잡도 관점에서 랜덤 오라클을 사용하지 않는 실용적인 삼자 간 PAKE 프로토콜인 KEDP(Key Exchange with Different Passwords)를 제안한다. KEDP는 표준모델에서 안전하고 효율적이다. [표 1]에서 안전성이 증명된 기존 삼자 간 PAKE 프로토콜과 제안 프로토콜의 효율성과 안전성을 비교한다. GPAKE에 의해 만들어진 프로토콜의 효율성은 하위 프로토콜로 사용되는 이자 간 PAKE 프로토콜과 삼자 간 키 분배 프로토콜의 효율성에 의존한다. [3]에서 제시된 GPAKE의 구체적인 예시 프로토콜은 Katz와 그 외가 제안한 표준모델에서 안전한 이자 간 PAKE 프로토콜[15]과 Bellare와 그 외가 제안한 삼자 간 키 분배 기법[8]을 이용하였기 때문에 비교표에서는 이 두 프로토콜을 사용한 GPAKE의 예시 프로토콜과 비교한다. 사실, GPAKE의 예시 프로토콜의 비효율성은 하위 이자 간 PAKE 프로토콜의 비효율성에 기인한다. 표준모델에서 안전성이 증명된 이자 간 PAKE 프로토콜들[12,14,15]이 존재하지만, 이러한 PAKE 프로토콜은 모바일 기기에 사용하기에는 매우 비효율적이다. 그중에서도 GPAKE에 사용된 Katz와 그 외의 프로토콜[15]이 표준모델에서 가장 효율적인 이자 간 PAKE 프로토콜로 알려져 있다.

〔표 1〕 증명 가능한 안전성을 가지는 삼자 간 PAKE 프로토콜의 비교

스킴	라운드	모듈로 지수승		탐지할 수 없는 온라인 패스워드 추측 공격에 대한 안전성	서버에 대한 키 기밀성	전방향 안전성	기지 키 공격에 대한 안전성	모델
		사용자	서버					
GPAKE[3]	6	≥ 17	≥ 17	-	○	–	○	표준
AP[6]	2	2	2	×	○	–	○	랜덤 오라클
KEDP(제안 기법)	2	4	2	-	○	○	○	표준

* 제안 기법에서 지수승의 수는 프로토콜 내부에서 사용되는 암호 기법을 DHIES[1,2]로 이용한 경우이다.
* 제공 안전성에 대한 증명이 되지 않은 경우 '-' 기호로 표시한다.

Ⅱ. 안전성 모델

이 장에서는 세션 키 교환을 원하는 두 명의 사용자가 서로 다른 패스워드를 사용하는 PAKE 프로토콜을 위한 안전성 모델을 정의한다. 여기서 정의하는 안전성 모델은 키 교환 프로토콜을 분석하는데 광범위하게 사용되는 Bellare와 그 외[7]에 의해 확립된 모델을 따르는 삼자 간 PAKE에 대한 Abdalla와 그 외[3]의 모델에 기반 한다.

P_i는 사용자의 식별자(identity)이고, S는 서버의 식별자이다. S는 서로 다른 패스워드를 갖고 있는 두 사용자 P_i와 P_j가 공통의 세션 키를 공유할 수 있도록 도와준다. P_i의 패스워드는 pw_i이다. pw_i가 패스워드 사전(dictionary) 내에 존재하고, 사전의 크기를 PW라고 가정한다. 본 논문에서 고려하는 모델은 서버 S가 각 사용자 P_i의 패스워드 pw_i를 가지고 있는 대칭적인 (symmetric) 모델이다.

파트너 관계(partnered)에 대한 정의는 세션 식별자(session identifier)의 개념을 사용하는 정의[7]를 따른다. P_i는 동시에 여러 개의 세션을 수행할 수 있기 때문에 많은 인스턴스(instance)들을 가질 수 있다. 세션에서의 인스턴스를 구별하기 위해서 P_i^k를 P_i의 k 번째 인스턴스라고 하고, S^ℓ을 S의 ℓ 번째 인스턴스라고 한다. P_i^k의 세션 식별자 sid_i^k는 세션을 유일하게 식별하며 메시지의 소유자의 식별자의 사전 편찬 순으로 정렬된 프로토콜 메시지의 접합으로 정의한다. 본 논문에서는 브로드캐스트 채널을 가정하고 참여자는 메시지를 동시 다발적으로 전송할 수 있다고 가정한다. 또한 사용자들은 그들의 식별자 순으로 정렬(예를 들면, 사전 편찬 법)될 수 있다고 가정하고 $P_i < P_j$는 이러한 정렬에 의한 순서를 나타낸다.

P_i^k의 파트너 식별자인 πd_i^k는 P_i^k가 세션 키 SK_i^k를 확립하고자 하는 상대 참여자의 식별자를 나타낸다. 만약 다음의 조건을 만족한다면 오라클 P_i^k와 $P_j^{k'}$은 파트너 관계에 있다 :

(1) $sid_i^k = sid_j^{k'}$; (2) $\pi d_i^k = P_j$; (3) $\pi d_j^{k'} = P_i$.

공격자는 정당한 사용자를 제외한 외부 공격자(outside attacker), 호기심 많은 서버(malicious server)와 악의적인 내부 사용자(inside attacker)의 세 가지 유형으로 구분된다. 외부 공격자는 세션 키에 대한 키 기밀성을 깨는 것을 목적으로 한다. 호기심 많은 서버는 수동적인 공격(passive attack)을 수행하지만 정직한 사용자 사이에 공유된 세션 키에 대한 키 기밀성을 깨는 것을 목적으로 한다. 악의적인 내부 사용자는 상대 사용자의 패스워드를 알아내는 것을 목적으로 한다.

2.1 외부 공격자를 위한 쿼리

A를 공격자라고 하자. A는 모든 통신을 통제할 수 있고 오라클에게 다음과 같은 쿼리를 할 수 있다.

- $Send(P_i^k, m)$ 또는 $Send(S^\ell, m)$: 이 쿼리를 통해 A는 인스턴스 P_i^k 또는 S^ℓ에게 메시지 m을 보내고 이에 대한 응답을 얻는다. 이 쿼리는 공격자 A의 메시지의 위·변조와 같은 능동적인 공격(active attack)을 모델링 한다. 공격자는 $Send_0(P_i^k, S, P_j)$ 쿼리를 통해 P_i^k가 S, P_j와 키 교환 프로토콜을 시작하도록 할 수 있다. P_i^k는 $Send_0(P_i^k, S, P_j)$에 대한 응답으로 프로토콜의 첫 번째 메시지를 보낸다.

- $Execute(P_i^k, S^\ell, P_j^{k'})$: 이 쿼리는 도청과 같은 공격자 A의 수동적 공격(passive attack)을 모델링 한다. 이 쿼리를 통해 공격자는 S^ℓ, P_i^k와 $P_j^{k'}$ 간에 프로토콜의 올바른 수행에 의해서 교환된 메시지들을 얻는다. (*Send* 오라클 쿼리를 반복적으로 실행하여 *Execute* 쿼리를 수행할 수 있지만 *Execute* 쿼리는 수동적 공격과 능동적 공격을 구별하기 위해 필요하다.)

- $Reveal(P_i^k)$: 이 쿼리는 세션 키가 노출되는 경우를 모델링 한다. 즉, 이 쿼리는 실제 환경에서 기지 키 공격(known-key attack)을 모델링 한 것이다. 만약 세션 키 SK_i^k가 P_i^k에 의해 만들어진 것이라면 공격자는 이 쿼리를 통해 P_i^k의 세션 키 SK_i^k를 얻을 수 있다. 더 자세히 설명하면, 기지 키 공격에 대한 안전성은 다수의 세션들(반드시 기밀성이 보장되어야 하는 세션이외의 세션들)의 세션 키가 노출되는 경우를 포함하는 "Denning-Sacco" 공격에 대한 안전성이다. 패스워드 기반 키 교환 프로토콜에서 기지 키 공격에 대한 안전성은 공격자가 정직한 참

여자들 사이에 성공적으로 확립된 세션 키를 이용해서 패스워드에 대한 오프라인 추측 공격을 통해 패스워드를 알아낼 수 없어야 한다는 것을 포함한다.

- *Corrupt*(P_i) 또는 *Corrupt*(S) : 이것은 P_i 또는 S 각각이 소유하고 있는 롱텀 키(long-term key)가 노출되는 경우를 모델링 한다. 즉, 이것은 전방향 안전성(forward secrecy)을 모델링한 것이다. 공격자는 이 쿼리를 통해 P_i 또는 S의 롱텀 키를 얻을 수 있다. 전방향 안전성을 만족하는 키 교환 프로토콜에서는 롱텀 키가 노출되기 이전에 세션 키가 위·변조와 같은 공격자의 적극적 공격에 의해 방해를 받아 만들어졌다고 하여도 세션 키의 기밀성이 유지된다.

- *Test*(P_i^k) : 이 쿼리는 공격자의 이득(advantage)을 정의하기 위해서 사용된다. 공격자 A는 프레쉬 오라클(fresh oracle)에 대해서만 단 한 번 이 쿼리를 할 수 있다. 프레쉬 오라클은 이후에 정의한다. 이 쿼리에서 동전 b가 던져진다. 만약 $b=1$이면 P_i^k가 가지고 있는 세션 키 SK_i^k가 공격자에게 반환된다. 만약 $b=0$이면 θ가 안전성 파라미터일 때 $\{0,1\}^\theta$에서 임의로 선택된 난수가 반환된다.

2.2 외부 공격자에 대한 프레쉬니스(freshness)

자명한(trivial) 공격을 제외하기 위해 전방향 안전성을 고려한 *Test* 쿼리에 대한 프레쉬니스의 개념을 정의한다. 만약 다음의 조건을 만족한다면 오라클 P_i^k는 프레쉬(fresh) 하다고 한다 :

(1) P_i^k에게 *Reveal* 쿼리가 요청되지 않았다.
(2) 만약 P_i^k와 $P_j^{k'}$가 파트너 관계라면 $P_j^{k'}$에게 *Reveal* 쿼리가 요청되지 않았다.
(3) P_j^k가 P_i^k의 파트너일 때 어떠한 *Send*(P_i^k, *) 쿼리 이전에 공격자에 의해 *Corrupt*(S) 쿼리와 *Corrupt*(P_j)쿼리 모두 요청되지 않았다. S 또는 P_j에 *Corrupt* 쿼리를 요청하여 S 또는 P_j의 롱텀 키를 얻은 후에는 *Send*(P_i^k, *)를 이용하여 특정한 세션에서 공격자 A가 P_j를 항상 가장할 수 있다. 이러한 경우 A는 자명하게 이 세션의 세션 키를 알아낼 수 있다. 이러한 자명한 경우를 제외하기 위해 이 세 번째 조건이 필요하다.

2.3 키 기밀성을 깨려고 하는 호기심 많은 서버를 위한 쿼리

본 논문에서는 서버가 모든 사용자에 대한 패스워드를 알고 있고 정직하게 행동하지만 사용자들이 공유한 세션 키를 알아내려고 하는 호기심 많은 서버 즉, 악의적인 서버를 가정한다. 이러한 이유에서 제안 프로토콜에서는 키 기밀성을 깨려고 하는 서버는 *Execute* 오라클과 *Send*(P_i^k, m) 오라클에 여러 번의 쿼리를 할 수 있지만, *Send*(S^t, m) 오라클과 *Reveal* 오라클은 패스워드와 비밀키를 사용하여 각각 쉽게 모의(simulation) 할 수 있기 때문에 이들 오라클에는 쿼리할 수 없다.

2.4 호기심 많은 서버에 대한 프레쉬니스(freshness)

만약 다음 조건들을 만족한다면 오라클 P_i^k는 프레쉬(fresh) 하다고 한다.

(1) P_i^k와 파트너 관계인 인스턴스 $P_j^{k'}$이 존재한다.
(2) P_i^k는 *NLL*이 아닌 세션 키 SK_i^k를 계산하였고, P_i^k와 $P_j^{k'}$ 모두에게 *Reveal* 쿼리를 요청하지 않았다.

정의 **1.** 만약 다음 세 가지 성질을 만족한다면 프로토콜 P는 안전하다고 말한다.

- 유효성(validity) : 만약 세션의 모든 오라클들이 파트너 되었다면(partnered) 모든 오라클들의 세션 키는 동일하다.
- 외부 공격자에 대한 키 기밀성 : 공격자 A는 위에서 설명한 외부 공격자를 위한 오라클들에 쿼리를 하고 그에 대한 응답을 받는다. 게임(game)의 어떤 시점에서 A는 외부 공격자에 대한 프레쉬 오라클에 *Test* 쿼리를 한다. A는 *Test* 쿼리를 한 이후에도 다른 쿼리를 할 수도 있다. A는 *Test* 오라클에서 사용된 비트 b에 대한 추측한 결과로 b'을 출력한다. 공격자 A의 이득(advantage)은 반드시 안전성 파라미터 θ에 의한 값으로 측정되어야 하며 다음과 같이 정의 된다 :

$$Adv_{P,A}^{outAtt}(\theta)=\Pr[A(\ )=1|b=1] -\Pr[A(\ )=1|b=0].$$

이득 함수(advantage function)는 다음과 같이 정의 된다 :

$$Adv_{P}^{outAtt}(\theta,t)=\max_{A}\{Adv_{P,A}^{outAtt}(\theta)\},$$

이 때 A는 θ에 대한 다항 함수로 표현되는 t를 시간 복잡도로 갖는 공격자이다. $Adv_{P}^{outAtt}(\theta,t)$는 $\epsilon(\theta)$가 네글리져블(negligible)하고, q_{se}가 *Send* 쿼리의 횟수이고, PW가 패스워드 공간의 크기일 때, $\frac{q_{se}}{PW}+\epsilon(\theta)$로 제한되어(bounded)야 한다.

- 호기심 많은 서버에 대한 키 기밀성 : 서버는 위에서 설명한 호기심 많은 서버를 위한 오라클들에 쿼리를 하고 그에 대한 응답을 받는다. 그리고 위에서 설명한 서버에 대한 프레쉬 오라클에 *Test* 쿼리를 한다. $\epsilon(\theta)$가 네글리져블(negligible)할 때 $Adv_{P}^{curSvr}(\theta,t)$는 $\epsilon(\theta)$로 제한되어(bounded)야 한다.

Ⅲ. 제안 프로토콜

본 장에서는 서로 다른 패스워드를 사용하는 두 사용자를 위한 PAKE 프로토콜을 위한 제안 프로토콜인 KEDP를 설명한다. G를 소수 q를 위수로 하는 순환 군이라고 하고, g를 G의 생성자라고 하자. M을 강력한 위조 불가능성을 지닌 MAC 알고리즘이라고 하고, H를 $\{0,1\}^{*}\rightarrow\{0,1\}^{\theta}$인 해쉬 함수라고 하자.

초기화. 서버는 비대칭 암호 스킴 $E=(E.key, E.enc, E.dec)$에 대한 비밀키 sk와 공개키 pk를 가진다. 사용자 P_i는 패스워드 pw_i를 선택한다. P_i와 서버 S는 안전한 방법으로 pw_i를 공유한다. 공개 파라미터인 (G, q, g, H, E, M, pk)는 모든 개체에게 접근 가능한 정보이다. 여기서 공개 파라미터는 사전에 모든 사용자 디바이스에 심는 방법 등을 사용하여 안전하게 분배되어 짐으로 공개키 pk에 대한 인증서는 필요 없다 (공개키 기반의 두 개체 간 키 교환의 경우에는 키 교환을 하는 대상이 불특정하기 때문에 공개키에 대한 인증서가 반드시 필요하지만, 제안 프로토콜과 같이 삼자 간 키 교환 모델에서는 서버가 고정되어 있기 때문에 공개키를 사용자에게 공개 정보로 놓는 것에 큰 무리가 없다). P_i와 P_j가 세션 키를 확립하고자 한다고 가정하고 P_i의 관점에서 프로토콜을 설명한다. P_j는 P_i와 동일하게 프로토콜을 수행한다.

KEDP

라운드 **1.** P_i는 $[1,q]$ 내에서 임의로 x_i를 선택하고, $\{0,1\}^{\theta}$ 내에서 임의로 k_i를 선택한다. P_i는 $X_i=g^{x_i}$를 계산하고, S에게 $(P_i, c_i=E.enc_{pk}(pw_i\|k_i\|X_i\|P_j))$를 전송한다.

라운드 **2.** $(P_i, \tilde{c_i})$와 $(P_j, \tilde{c_j})$를 받고 난 후 S는 $\tilde{c_i}$와 $\tilde{c_j}$을 복호화 한다. 이를 통해 $(\widetilde{pw_i}\|\tilde{k_i}\|\tilde{X_i}\|P_j)=E.dec_{sk}(\tilde{c_i})$와 $(\widetilde{pw_j}\|\tilde{k_j}\|\tilde{X_j}\|P_i)=E.dec_{sk}(\tilde{c_j})$를 얻는다. 만약 $\widetilde{pw_i}\neq pw_i$ 또는 $\widetilde{pw_j}\neq pw_j$ 이면, S는 프로토콜을 종료한다. 그렇지 않으면, S는 $\tau_i=Mac_{\tilde{k_i}}(P_i\|P_j\|\tilde{X_i}\|\tilde{X_j})$와 $\tau_j=Mac_{\tilde{k_j}}(P_j\|P_i\|\tilde{X_j}\|\tilde{X_i})$를 계산한다. S는 P_i에게 $(P_i, \tilde{X_j}, \tau_i)$를 보내고 P_j에게 $(P_j, \tilde{X_i}, \tau_j)$를 보낸다.

키 계산. $(P_i, \hat{X_j}, \hat{\tau_i})$를 전송 받은 후, P_i는 $Vfy_{k_i}(P_i\|P_j\|X_i\|\hat{X_j}, \hat{\tau_i})=1$인지 확인한다. 만약 검증이 성공하면, P_i는 세션 키 $SK=H(\hat{X_j}^{x_i})$를 계산한다.

3.1 완전성(Completeness)

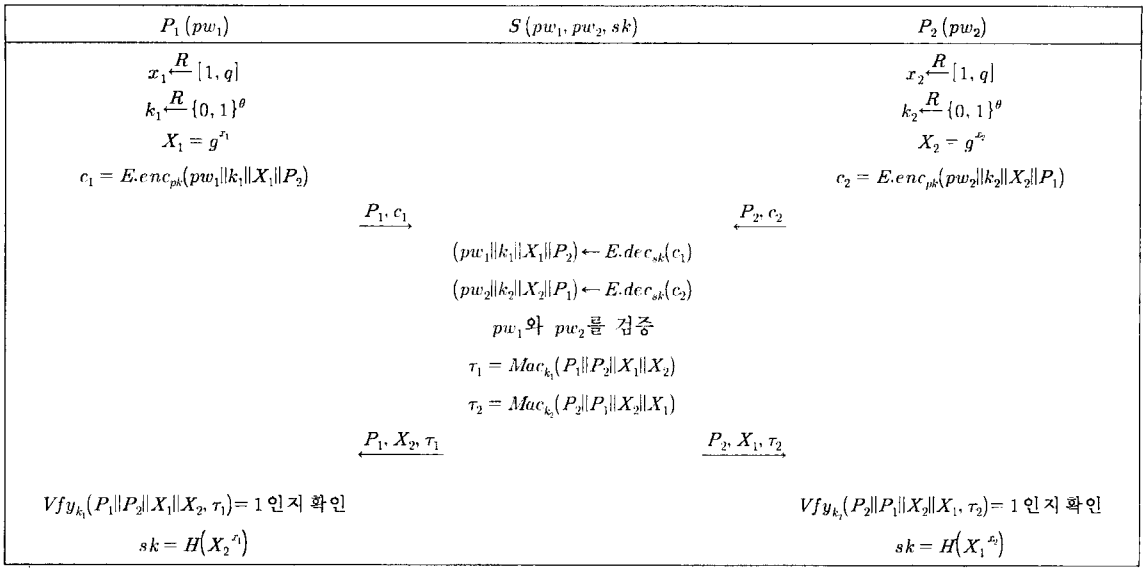
프로토콜이 모두 정상적으로 수행되면 사용자 P_i와 P_j는 동일한 세션 키 $SK=H(g^{x_ix_j})$를 계산하게 된다. [그림 1]은 KEDP 프로토콜의 실행 예이다.

3.2 효율성

KEDP의 효율성은 프로토콜에서 사용되는 비대칭 암호 스킴인 E에 의존한다. 현재까지 제안된 비대칭 암호 스킴 중 랜덤 오라클(random oracle)을 사용하지 않는 환경에서 IND-CCA에 안전한 가장 효율적인 비대칭 암호 스킴은 DHIES이다[1,2]. DHIES에서 메시지를 암호화 하는 과정에서는 두 번의 지수승 연산을 필요로 하고, 암호문을 복호화 하는 과정에서는 한 번의 지수승 연산이 필요하다. 그러므로 DHIES를 사용하는 KEDP의 사용자는 네 번의 지수승 연산을 해야 하고, 서버는 두 번의 지수승 연산을 해야 한다.

3.3 탐지할 수 없는 온라인 패스워드 추측 공격에 대한 안전성

KEDP는 탐지할 수 없는 온라인 패스워드 추측 공격



〔그림 1〕 KEDP 프로토콜 실행의 예

에 안전하도록 설계되었다. 만약 온라인 패스워드 추측 공격이 서버에 의해 탐지 가능하다면, 이러한 공격들은 더 이상 중대한 위협이 되지 못할 것이다. KEDP에서 서버는 사용자로부터 받은 암호문을 복호화한 후 복호한 패스워드가 올바른지 확인함으로써 사용자 중 누구의 패스워드가 탐지할 수 없는 온라인 패스워드 추측 공격에 대상이 되고 있음을 알아차릴 수 있다. 만약 기존 정해진 실패 제한 횟수를 초과하는 경우 서버는 공격 대상이 되는 사용자에게 더 이상 패스워드를 사용하지 말고 패스워드를 교체하도록 한다. 검증 과정을 통과하기 위해서는 공격자는 단 한 번에 또는 실패 제한 횟수를 초과하지 않는 횟수 내에 올바른 패스워드를 성공적으로 추측해 내야 한다. 그러나 이것은 패스워드 공간의 크기 때문에 성공할 확률이 일반적으로 매우 낮다.

3.4 외부 공격자에 대한 안전성

다음 정리는 KEDP가 외부 공격자에 대한 오프라인 패스워드 추측 공격에 안전하고, 전방향 안전성과 기지 키 공격에 대한 안전성을 제공함을 보인다.

정리 **1.** G를 HDH 가정이 만족되는 그룹이라고 하고, M을 강력한 위조 불가능성(SUF)을 지닌 MAC 알고리즘이라고 하자. 그러면 KEDP는 외부 공격자에 대한 키 기밀성을 제공한다. 구체적으로,

$$Adv^{outAtt}_{KEDP}(\theta, t, q_{se}) \le (2+3Nq_s) \cdot Adv^{IND\ \ CCA}_{E} + Nq_s \cdot Adv^{SUF}_{M} + (Nq_s)^2 \cdot Adv^{HDH}_{H,GG} + \frac{q_{se}}{PW} + \frac{q_{se}}{2^\theta}.$$

이때 t는 공격자의 수행 시간을 포함한 최대 총 게임의 시간이다. 공격자는 q_{se} 개의 *Send* 쿼리를 만든다. N은 전체 사용자의 수이고, q_s는 공격자가 만든 세션 수의 상한 값이다. PW는 패스워드 공간의 크기이다.

정리 **1**의 증명. 본 증명에서는 공격자가 취할 수 있는 최고의 전략이 온라인 패스워드 추측 공격을 수행하는 것임을 증명한다. KEDP의 안전성을 무시할 수 없는 확률로 깨는 외부 공격자 A를 가정한다. 이는 A가 온라인 패스워드 추측 공격을 통해 올바른 패스워드를 찾기 전에 무시할 수 없는 확률로 KEDP의 안전성을 깨야함을 의미한다. 따라서 본 증명에서는 올바른 패스워드로 만들어진 첫 번째 라운드의 메시지를 전송하지 않는 공격자의 이득(advantage)을 측정한다. 이를 측정하기 위해 우선 다음과 같은 게임들을 정의 한다 :

$$\begin{pmatrix} game_{KEDP}, game_{0}, game_{1,1}, \ldots, game_{1,q_s}, \ldots, \\ game_{N,1}, \ldots, game_{N,q_s} \end{pmatrix}$$

$game_{KEDP}$에서는 3.1에서 정의한 외부 공격자를 위한

오라클 쿼리들($Send(P_i^k, m)$, $Execute(S^\ell, P_i^k, P_j^{k'})$, $Reveal(P_i^k)$, $Corrupt(P_i)$, $Test(P_i^k)$)에 대한 응답은 KEDP 프로토콜에서와 동일하게 만들어진다. $game_0$은 다음을 제외하고는 $game_{KEDP}$와 동일하다 :

(1) 시뮬레이터는 P_i의 패스워드로 pw_i'을 $\{0,1\}^\theta$에서 임의로 선택한다.

(2) S^ℓ이 첫 번째 라운드 메시지로 $(P_i, \tilde{c_i})$를 받았다고 가정하자. $\tilde{c_i}$를 복호화 한 것이 $(\widetilde{pw_i}\|\tilde{k_i}\|\tilde{X_i}\|\tilde{P_j})$라고 하자. 만약 $\widetilde{pw_i} \in \{pw_i, pw_i'\}$이면, 시뮬레이터는 $\tilde{c_i}$를 올바른 것으로 보고, S^ℓ의 두 번째 라운드 메시지를 전송한다. 그렇지 않다면 시뮬레이터는 두 번째 라운드에서 아무 메시지도 보내지 않는다.

게임 $game_{i,k}(1 \le i \le N, 1 \le k \le q_s)$는 다음을 제외하고 이전의 게임과 동일하다 : 시뮬레이터는 P_i^k의 첫 번째 라운드 메시지로 $(P_i, c_i = E.enc_{pk}(pw_i'\|k_i\|X_i\|P_j))$을 전송한다. 여기서 P_j는 P_i^k의 파트너이다. 여기서 pw_i 대신에 $pw_i' \in \{0,1\}^\theta$가 사용되는 것을 주의하자.

$Adv^{outAtt}_{game_X}$는 $game_X$에서 Adv^{outAtt}을 나타내고, $Adv^{outAtt}_{game_X,A}$는 $game_X$에서 Adv^{outAtt}_A을 나타낸다.

주장 **1.** $Adv^{outAtt}_{game_{KEDP}} \le Adv^{outAtt}_{game_0}$.

주장 **2.** $Adv^{outAtt}_{game_0} - Adv^{outAtt}_{game_{Nq_s}} \le 2Nq_s \cdot Adv^{IND-CCA}_E$.

주장 **3.** $Adv^{outAtt}_{game_{Nq_s}} \le (2+Nq_s) \cdot Adv^{IND-CCA}_{E,B} + Nq_s \cdot Adv^{SUF}_M + (Nq_s)^2 \cdot Adv^{HDH}_{G,H} + \frac{q_{se}}{PW} + \frac{q_{se}}{2^\theta}$.

위의 주장 1, 2와 3으로부터 정리 1이 증명된다.

주장 **1**의 증명. $game_0$에서 P_i에 대해 두 개의 패스워드 pw_i와 pw_i'가 존재한다. 그러므로 $game_0$에서 공격자의 이득(advantage)은 $game_{KEDP}$에서의 공격자의 이득보다 큰 것은 명백하다.

주장 **2**의 증명. 본 증명에서는 만약 두 인접한 게임 $game_{i',k'}$과 $game_{i^*,k^*}$에서 A의 이득의 차이가 네글리져블(negligible)하지 않으면, 즉 무시할 수 없는 값이면 A를 이용하여 E의 IND-CCA 안전성을 깨는 알고리즘 B를 설계할 수 있음을 보인다. 즉 다음을 보인다.

$$Adv^{outAtt}_{game_{i,k},A} - Adv^{outAtt}_{game_{i',k'},A} \le 2 \cdot Adv^{IND-CCA}_{E,B}.$$

B에게는 공개 키 pk와 복호화 오라클 $E.dec_{sk}(\cdot)$이 주어지고, B는 A에게 다음과 같이 KEDP를 시뮬레이션 해준다 :

(1) A의 쿼리들에 대해서, B는 다음을 제외하고 $game_{i',k'}$에서와 같이 응답 한다 : 만약 B가 $P_{i^*}^{k^*}$의 첫 번째 라운드 메시지를 생성해야 한다면 B는 IND-CCA 게임의 $find$ 단계에서 $(m_0 = pw_{i^*}'\|k_{i^*}\|X_{i^*}\|P_j,\ m_1 = pw_{i^*}\|k_{i^*}\|X_{i^*}\|P_j)$을 출력한다. 여기서 P_j는 $P_{i^*}^{k^*}$의 파트너이다. B는 도전 암호문(challenge cipher) c^*를 받는다. B는 $P_{i^*}^{k^*}$의 첫 번째 라운드 메시지로 A에게 $(P_i, c_i = c^*)$를 전송한다. 만약 B가 암호문을 복호해야 한다면 B는 복호화 오라클 $E.dec_{sk}(\cdot)$를 사용한다.

(2) B는 $Test$ 쿼리에서 동전 b을 사용한다고 가정하고, A는 b'을 출력한다고 가정한다. 만약 $b=b'$이면 B는 1을 출력하고 종료한다. 그렇지 않으면 B는 0을 출력하고 종료한다.

만약 $c^* = E.enc_{pk}(pw_{i^*}\|k_{i^*}\|X_{i^*}\|P_j)$이면 B는 $game_{i',k'}$을 시뮬레이션 한다. 만약 $c^* = E.enc_{pk}(pw_{i^*}'\|k_{i^*}\|X_{i^*}\|P_j)$이면 B는 $game_{i^*,k^*}$를 시뮬레이션 한다. 그러므로 다음의 수식이 성립한다 :

$$\begin{aligned} Adv^{IND-CCA}_{E,B} &= \Pr[B(\ )=1|d=1] - \Pr[B(\ )=1|d=0] \\ &\ge \Pr_A[b=b' in\ game_{i',k'}] \\ &\quad - \Pr_A[b=b' in\ game_{i^*,k^*}] \\ &\ge \frac{Adv^{outAtt}_{game_{i,k},A}+1}{2} - \frac{Adv^{outAtt}_{game_{i',k'},A}+1}{2}. \end{aligned}$$

하이브리드 논법(hybrid argument)을 사용하면 다음의 수식이 성립한다 :

$$Adv^{outAtt}_{game_0} - Adv^{outAtt}_{game_{Nq_s}} \le 2Nq_s \cdot Adv^{IND-CCA}_E.$$

주장 **3**의 증명. $\tilde{c_i} = E.enc_{pk}(pw_i'\|*\|*\|*)$이고 P_i가 전송한 적이 없는 $(P_i, \tilde{c_i})$를 A가 S에게 보내는 사건

을 $forgeCipher$라고 정의한다.

$game_{N,q_s}$에서 A의 이득(advantage)은 다음의 경우로 나뉜다.

(경우 1) 최소 한 번의 $forgeCipher$ 사건이 발생한다.

(경우 2) $forgeCipher$ 사건이 발생하지 않는다.

위이 각 경우에 대한 이득(advantage)은 다음의 주장과 같다.

주장 **3.1.** $Adv^{outAtt.forgeCipher}_{game_{Nq_s},A}$

$$\leq Nq_s \cdot Adv^{IND-CCA}_{E.B} + \frac{q_{sr}}{2^{\theta}}.$$

주장 **3.2.** $Adv^{outAtt.\overline{forgeCipher}}_{game_{Nq_s},A}$

$$\leq 2 \cdot Adv^{IND\ \ CCA}_{E.B} + Nq_s \cdot Adv^{SUF}_{MI} + (Nq_s)^2 \cdot Adv^{HDH}_{H.GG} + \frac{q_{se}}{PW}.$$

주장 3.1과 3.2로부터 주장 3이 증명된다.

주장 **3.1**의 증명. $\tilde{c_i} = E.enc_{pk}(pw_i' \| * \| * \| *)$이고 $game_{N,q_s}$에서 P_i로부터 전송된 적이 없는 $(P_i, \tilde{c_i})$를 A가 S에게 보내는 사건을 $forgeCipher$라고 정의한다. $forgeCipher$ 사건에서의 이득(advantage)인 $Adv^{outAtt.forgeCipher}_{game_{Nq_s},A}$를 구하기 위해서 다음과 같은 게임들을 정의한다 :

$$\begin{pmatrix} game_{N+1,1}, \cdots, game_{N+1,q_s}, \cdots \\ game_{2N,1}, \cdots, game_{2N,q_s} \end{pmatrix}.$$

게임 $game_{i,k}(N+1 \leq i \leq 2N, 1 \leq k \leq q_s)$는 다음 사항을 제외하고 이전의 게임과 동일하다 :

(1) 시뮬레이터는 P_i의 패스워드 pw_i''을 $\{0,1\}^{\theta}$에서 임의로 선택한다.

(2) 시뮬레이터는 P_i^k의 첫 번째 라운드 메시지 $(P_i, c_i = E.enc_{pk}(pw_i'' \| k_i \| X_i \| P_j))$을 전송한다. 여기서 P_j는 P_i^k의 파트너이다. pw_i' 대신에 pw_i''가 사용된다는 점에 주의하자.

(3) S^f은 첫 번째 라운드 메시지로 $(P_i, \tilde{c_i})$를 받았다고 가정한다. $\tilde{c_i}$를 복호화 한 메시지를 $(\widetilde{pw_i} \| \tilde{k_i} \| \tilde{X_i} \| \tilde{P_j})$이라고 하자. 만약 $\widetilde{pw_i} \notin \{pw_i, pw_i', pw_i''\}$이면 시뮬레이터는 두 번째 라운드에서 아무런 메시지도 전송하지 않는다.

$game_{2N,q_s}$에서 P_i의 메시지들은 pw_i'와 독립적이므로 $forgeCipher$가 일어날 확률은 $\frac{q_{sr}}{2^{\theta}}$이라는 것은 명백하다. 만약 인접한 게임 $game_{i',k'}$과 $game_{i^*,k^*}$에서 사건 $forgeCipher$가 일어날 확률의 차이가 네글리저블(negligible)하지 않다면 즉, 무시할 수 없다면, A를 이용하여 E의 IND-CCA 안전성을 깨는 알고리즘 B를 설계할 수 있다. 그러므로 사건 $forgeCipher$로부터 A의 이득(advantage)을 하이브리드 논법(hybrid argument)을 사용하여 다음과 같이 구할 수 있다.

$$Adv^{outAtt.forgeCipher}_{game_{Nq_s},A} \leq Nq_s \cdot Adv^{IND-CCA}_{E.B} + \frac{q_{sr}}{2^{\theta}}.$$

공개 키 pk와 복호화 오라클 $E.dec_{sk}(\cdot)$가 B에게 주어지고, B는 A에게 KEDP를 다음과 같이 시뮬레이션 해 준다 :

(1) A의 오라클 쿼리들에 대해 B는 다음의 경우를 제외하고 $game_{i',k'}$에서처럼 응답한다 : 만약 B가 $P_{i^*}^{k^*}$의 첫 번째 라운드 메시지를 생성해야 한다면, B는 $(m_0 = pw_{i^*}'' \| k_{i^*} \| X_{i^*} \| P_j,\ m_1 = pw_{i^*}' \| k_{i^*} \| X_{i^*} \| P_j)$를 출력한다. 여기서 P_j는 $P_{i^*}^{k^*}$의 파트너이다. B는 IND-CCA 게임에서의 도전 암호문 c^*를 받는다. B는 A에게 $P_{i^*}^{k^*}$의 첫 번째 라운드 메시지로 $(P_i, c_i = c^*)$을 보낸다. 만약 B가 암호문을 복호해야 한다면 B는 복호화 오라클 $E.dec_{sk}(\cdot)$을 사용한다.

(2) 만약 $forgeCipher$ 사건이 일어나면 B는 1을 출력하고 종료한다.

만약 $c^* = E.enc_{pk}(pw_{i^*}' \| k_{i^*} \| X_{i^*} \| P_j)$이면, B는 $game_{i',k'}$을 시뮬레이션 한다. 만약 $c^* = E.enc_{pk}(pw_{i^*}'' \| k_{i^*} \| X_{i^*} \| P_j)$이면, B는 $game_{i^*,k^*}$을 시뮬레이션 한다. 그러므로 다음의 수식이 성립한다 :

$$\begin{aligned} Adv^{IND-CCA}_{E.B} &= \Pr[B(\ ) = 1 | b = 1] - \Pr[B(\ ) = 1 | b = 0] \\ &\geq \Pr[forgeCipher\ in\ game_{i',k'}] \\ &\quad - \Pr[forgeCipher\ in\ game_{i^*,k^*}]. \end{aligned}$$

주장 **3.2**의 증명. $\overline{forgeCipher}$의 이득(advantage)인

$Adv^{outAtt,\overline{forgeCipher}}_{game_{N,q_s},A}$를 구하기 위해서 다음과 같은 게임들을 정의한다 :

$$\begin{pmatrix} \widehat{game}_{N+1,1}, \cdots, \widehat{game}_{N+1,q_s}, \cdots, \\ \widehat{game}_{2N,1}, \cdots, \widehat{game}_{2N,q_s} \end{pmatrix}.$$

게임 $\widehat{game}_{i,k}(N+1 \le i \le 2N, 1 \le k \le q_s)$는 다음 사항을 제외하고 주장 2에서의 이전의 게임과 동일하다 :

(1) S^ℓ이 첫 번째 라운드 메시지인 $(P_i, \tilde{c}_i)$를 받았다고 가정하자. $\tilde{c}_i$를 복호화 한 메시지를 $(\widetilde{pw}_i \| \tilde{k}_i \| \tilde{X}_i \| P_j)$라고 하자. 만약 $\widetilde{pw}_i \notin \{pw_i, pw_i'\}$이면 시뮬레이터는 두 번째 라운드에서 아무런 메시지도 전송하지 않는다. 만약 $\tilde{k}_i = k_i$이면, 시뮬레이터는 임의의 값 k_i'을 선택하고, P_j에 의해 S^ℓ에게 보내지는 값이 $\tilde{X}_j$ 일 때, 두 번째 라운드에서 P_i에게 $(P_i, \tilde{X}_j, \tau_i = Mac_{k_i'}(P_i \| P_j \| \tilde{X}_i \| \tilde{X}_j))$을 전송한다.

(2) P_i^k가 두 번째 라운드의 메시지 $(P_i, \tilde{X}_j, \tilde{\tau}_i)$을 받았다고 가정하자. 시뮬레이터는 $Vfy_{k_i'}(P_i \| P_j \| X_i \| \tilde{X}_j, \tilde{\tau}_i) = 1$인지 확인한다. 만약 검증이 성공하면, 시뮬레이터는 세션 키 $SK = H(\tilde{X}_j^{x_i})$를 계산한다.

주장 **3.2.1.** $Adv^{outAtt,\overline{forgeCipher}}_{\widehat{game}_{i,k},A} - Adv^{outAtt,\overline{forgeCipher}}_{\widehat{game}_{i^*,k^*},A} \le 2 \cdot Adv^{IND-CCA}_{E,B}$.

주장 **3.2.2.** $Adv^{outAtt,\overline{forgeCipher}}_{\widehat{game}_{2N,q_s},A} \le Nq_s \cdot Adv^{SUF}_{M} + (Nq_s)^2 \cdot Adv^{HDH}_{H,GG} + \frac{q_{se}}{PW}$.

주장 3.2.1과 주장 3.2.2에 의해 주장 3.2가 증명 된다

주장 **3.2.1**의 증명. 만약 인접한 두 게임 $\widehat{game}_{i',k'}$와 $\widehat{game}_{i^*,k^*}$에서 *forgeCipher* 사건이 발생하지 않았을 때 A의 이득(advantage)의 차이가 무시할 수 없는 값이면 A을 사용하여 E의 IND-CCA 안전성을 깨는 알고리즘 B를 설계할 수 있다. 두 게임에서 A의 이득(advantage)의 차이는 다음과 같다.

$$Adv^{outAtt,\overline{forgeCipher}}_{\widehat{game}_{i,k},A} - Adv^{outAtt,\overline{forgeCipher}}_{\widehat{game}_{i^*,k^*},A} \le 2 \cdot Adv^{IND-CCA}_{E,B}.$$

공개 키 pk와 복호화 오라클 $E.dec_{sk}(\cdot)$이 주어진 B는 A에게 KEDP를 다음과 같이 시뮬레이션 해 준다 :

(1) A의 오라클 쿼리에 대해 B는 다음의 경우를 제외하고 $\widehat{game}_{i',k'}$에서처럼 응답한다 : 만약 B가 $P_{i^*}^{k^*}$의 첫 번째 라운드 메시지를 생성해야 한다면 B는 IND-CCA 게임의 *find* 단계에서 $(m_0 = pw_{i^*}' \| k_{i^*}' \| X_{i^*} \| P_j, m_1 = pw_{i^*}' \| k_{i^*} \| X_{i^*} \| P_j)$를 출력한다. 여기서 P_j는 $P_{i^*}^{k^*}$의 파트너이다. B는 도전 암호문(challenge cipher) c^*를 받는다. B는 A에게 $P_{i^*}^{k^*}$의 첫 번째 라운드 메시지로 $(P_i, c_i = c^*)$를 보낸다. 만약 B가 암호문을 복호해야 한다면 B는 복호화 오라클 $E.dec_{sk}(\cdot)$을 사용한다.

(2) 만약 B가 $P_{i^*}^{k^*}$에 대한 MAC을 검증해야 한다면, B는 k_{i^*}을 사용한다.

(3) B가 *Test* 쿼리에서 동전 b를 사용했고 A는 b'을 출력한다고 가정한다. 만약 $b = b'$이면 B는 1을 출력하고 종료한다. 그렇지 않으면 B는 0을 출력하고 종료한다.

만약 $c^* = E.enc_{pk}(pw_{i^*}' \| k_{i^*} \| X_{i^*} \| P_j)$이면, B는 $\widehat{game}_{i',k'}$을 시뮬레이션 한다. 만약 $c^* = E.enc_{pk}(pw_{i^*}' \| k_{i^*}' \| X_{i^*} \| P_j)$이면, B는 $\widehat{game}_{i^*,k^*}$을 시뮬레이션 한다. 그러므로 다음의 수식이 성립한다 :

$$\begin{aligned} Adv^{IND-CCA}_{E,B} &= \Pr[B(\ ) = 1 | d = 1] - \Pr[B(\ ) = 1 | d = 0] \\ &\ge \Pr_A[b = b' \ in \ \widehat{game}_{i',k'}] \\ &\quad - \Pr_A[b = b' \ in \ \widehat{game}_{i^*,k^*}] \\ &\ge \frac{\left(Adv^{outAtt,\overline{forgeCipher}}_{A} \ in \ \widehat{game}_{i',k'}\right) + 1}{2} \\ &\quad - \frac{\left(Adv^{outAtt,\overline{forgeCipher}}_{A} \ in \ \widehat{game}_{i^*,k^*}\right) + 1}{2}. \end{aligned}$$

주장 **3.2.2**의 증명. MAC의 검증이 성공하고 S에 의해 전송된 적이 없는 $(P_i, \tilde{X}_j, \tilde{\tau}_i)$을 A가 P_i에게 전송하는 사건을 *forgeMac*이라고 정의한다.

$\widehat{game}_{2N,q_s}$에서 A의 이득(advantage)을 다음의 두 가지 경우로 나눈다 :

(경우 1) 최소 한 번의 *forgeMac* 사건이 발생한다.

(경우 2) *forgeMac* 사건이 발생하지 않는다.

다음에서 위의 주장들의 각 경우에 대한 이득(advantage)을 구한다.

주장 **3.2.2.1.** $Adv^{outAtt,forgeMac}_{\widehat{game}_{2N,q_s},A} \le Nq_s \cdot Adv^{SUF}_{M}$.

주장 **3.2.2.2.** $Adv^{outAtt,\overline{forgeMac}}_{\widehat{game}_{2N,q_s},A}$

$$\le (Nq_s)^2 \cdot Adv^{HDH}_{G,H} + \frac{q_{se}}{PW}.$$

주장 3.2.2.1과 주장 3.2.2.2에 의해 주장 3.2.2이 증명 된다.

주장 **3.2.2.1**의 증명. 만약 공격자 A가 $\widehat{game}_{2N,q_s}$에서 MAC을 위조하였다면 A를 이용하여 프로토콜에서 사용하는 MAC 알고리즘을 깨는 알고리즘 B를 설계할 수 있다. MAC 생성 오라클인 $Mac_k(\cdot)$과 MAC 검증 오라클인 $Vfy_k(\cdot,\cdot)$이 주어진 B는 A에게 KEDP를 다음과 같이 시뮬레이션 해 준다 :

(1) B는 $i^* \leftarrow [1,N]$와 $t^* \leftarrow [1,q_s]$를 선택한다.

(2) A의 오라클 쿼리들에 대해, B는 다음을 제외하고 $\widehat{game}_{2N,q_s}$와 동일하게 응답한다 : 만약 B가 P_{i^*}의 t^* 번째 인스턴스에 대한 MAC을 생성하거나 검증해야 한다면, B는 $Mac_k(\cdot)$ 또는 $Vfy_k(\cdot,\cdot)$을 사용한다.

(3) 만약 P_{i^*}의 t^*번째 인스턴스에 대한 위조된 메시지-태그 쌍인 (M^*,τ^*)이 시뮬레이션 중에 발견한다면, B는 (M^*,τ^*)를 출력하고 종료한다.

B의 성공 확률은 $forgeMac$ 사건의 발생 여부와 B가 i^*와 t^*를 올바르게 추측했는지에 따라 결정된다. 만약 B의 추측이 맞는다면, B는 A에게 KEDP를 완벽히 시뮬레이션 해 주게 된다. 그러므로 다음의 수식이 성립한다 :

$$Adv^{SUF}_{M,B} \ge \frac{1}{Nq_s} \cdot Adv^{outAtt,forgeMac}_{\widehat{game}_{2N,q_s},A}.$$

주장 **3.2.2.2**의 증명. $\widehat{game}_{2N,q_s}$에서 사용되는 사용자의 패스워드는 실제 프로토콜에서 사용되는 사용자의 메시지와 독립적이다. 따라서 무한한 능력을 지닌 공격자라 하더라도 프로토콜의 실행을 수동적으로 관찰하는 것으로는 사용자의 패스워드에 대한 정보를 얻을 수 없다. 공격자는 사용자의 패스워드에 대한 정보를 얻기 위해서 자신이 추측한 사용자의 패스워드에 대한 검증을 위해 온라인 패스워드 추측 공격을 수행할 수도 있다. 온라인 패스워드 추측 공격을 수행함에 의해 공격자의 이득(advantage)이 증가하는 것은 명백하다. 그러나 $\widehat{game}_{2N,q_s}$에서 무한한 능력을 지닌 공격자라 할지라도 그 이득은 $\frac{q_{se}}{PW}$을 넘을 수 없다. 반면에 $forgeCipher$와 $forgeMac$ 사건이 발생하지 않았을 때 공격자가 P_i의 세션 키를 올바르게 추측하기 위해서는 $X_i = g^{x_i}$와 $X_j = g^{x_j}$을 이용해 $H(g^{x_i x_j})$를 계산해야 한다. 여기서 X_i는 P_i에 의해 만들어지고 X_j는 P_j에 의해 만들어진 것이다. 그러므로 A를 이용하여 HDH 가정을 깨는 공격자 B을 설계할 수 있다. B는 해쉬 디피-헬만 문제의 실험에서 (G, q, g, U_1, U_2, W)을 입력 받고 프로토콜의 메시지 속에 이 값들을 심는다. B에 대한 자세한 설명은 다음과 같다 :

1. B는 (G, q, g, U_1, U_2, W)를 입력 받는다. B는 $[1,N]$에서 i^*, j^*을 임의로 선택하고, $[1,q_s]$에서 t_1, t_2을 임의로 선택한다.
2. A의 각 오라클 쿼리에 대해, B는 다음의 경우를 제외하고 $\widehat{game}_{2N,q_s}$와 같이 응답한다 :
   - B는 $P^{t_1}_{i^*}$의 첫 번째 라운드 메시지로 $X_{i^*} = U_1$을 사용하고, $P^{t_2}_{j^*}$의 첫 번째 라운드 메시지로 $X_{j^*} = U_2$를 사용한다.
   - 만약 A가 $P^{t_1}_{i^*}$에 $Reveal$ 쿼리를 하고 $P^{t_1}_{i^*}$이 X_{j^*}를 받았다면, B는 시뮬레이션 실패로 종료한다. 만약 A가 $P^{t_1}_{i^*}$에 $Reveal$ 쿼리를 하고 $P^{t_1}_{i^*}$이 $X_k = g^{x_k}$를 받았다면, B는 $U_1^{x_k}$를 반환한다. $forgeCipher$와 $forgeMac$ 사건이 발생하지 않았으므로 A는 시뮬레이터에 의해 선택된 P_k의 인스턴스를 위한 $X_k = g^{x_k}$를 사용하여야 한다는 것에 유의하자.
   - 만약 A가 $P^{t_2}_{j^*}$에 $Reveal$ 쿼리를 하고 $P^{t_2}_{j^*}$가 X_{i^*}를 받았다면, B는 시뮬레이션 실패로 종료한다. 만약 A가 $P^{t_2}_{j^*}$에 $Reveal$ 쿼리를 하고 $P^{t_2}_{j^*}$가 $X_k = g^{x_k}$를 받았다면, B는 $U_2^{x_k}$를 반환한다. $forgeCipher$와 $forgeMac$ 사건이 발생하지 않았으므로 A는 시뮬레이터에 의해 선택된 P_k의 인스턴스를 위한 $X_k = g^{x_k}$를 사용

하여야 한다.

- 만약 A가 $P_{i*}^{t_1}$에 *Test* 쿼리를 하고 $P_{i*}^{t_1}$가 X_{j*}를 받았다면, B는 A에게 W를 반환한다. 만약 A가 $P_{j*}^{t_2}$에 *Test* 쿼리를 하고 $P_{j*}^{t_2}$가 X_{i*}를 받았다면, B는 A에게 W을 반환한다. 그렇지 않은 경우 B는 시뮬레이션 실패로 종료한다.

3. A가 b'을 출력하고 종료한다고 가정한다. 그러면 B는 b'을 출력하고 종료한다.

만약 B가 i^*, j^*, t_1과 t_2을 올바르게 추측했다면, B는 $U_1 = g^{u_1}$이고 $U_2 = g^{u_2}$일 때 $W = H(g^{u_1 u_2})$의 여부에 따라 실제 세션 키 또는 임의의 문자열을 반환한다. 따라서 다음의 부등식이 성립한다 :

$$
\begin{aligned}
&Adv_B^{HDH} \\
&= \Pr\left[B(U_1, U_2, W) = 1 \mid U_1 = g^{u_1}, U_2 = g^{u_2}, W = H(g^{u_1 u_2})\right] \\
&\quad - \Pr\left[B(U_1, U_2, W) = 1 \mid U_1 = g^{u_1}, U_2 = g^{u_2}, W = \{0,1\}^{\theta}\right] \\
&\geq \frac{1}{(Nq_s)^2}\left(\Pr\left[A(\,) = 1 \mid \begin{array}{l}\textit{Test} \text{ 쿼리에 대한 응답으로} \\ A\text{가 실제 세션키를 받음}\end{array}\right]\right. \\
&\qquad \left. - \Pr\left[A(\,) = 1 \mid \begin{array}{l}\textit{Test} \text{ 쿼리에 대한 응답으로} \\ A\text{가 임의의 문자열을 받음}\end{array}\right]\right) \\
&= \frac{1}{(Nq_s)^2} \cdot Adv_{game_{2Nq_s}, A}^{outAtt, \overline{forgeMac}}.
\end{aligned}
$$

이로써 주장이 증명된다.

3.5 키 기밀성을 깨려고 하는 서버에 대한 안전성

다음의 정리는 KEDP가 세션 키의 기밀성을 깨려고 하는 서버에 대하여 안전함을 보여준다. 즉, 서버가 사용자의 패스워드를 알고 있다고 하더라도 도청으로 인해 서버는 사용자 사이의 세션 키를 알 수 없다.

정리 2. G을 HDH 가정이 만족되는 그룹이라고 하자. 그러면 KEDP는 세션 키를 알아내려고 하는 호기심 많은 서버에 대하여 키 기밀성을 제공한다. 구체적으로 호기심 많은 서버에 대한 이득(advantage)는 다음과 같다.

$$Adv_{KEDP}^{curSvr}(\theta, t) \leq (Nq_s)^2 \cdot Adv_{H,GG}^{HDH},$$

이때 t는 공격자의 수행 시간을 포함한 최대 총 게임 시간이다. N은 사용자의 수이고 q_s는 공격자가 만들 수 있는 세션 수의 상한치이다.

정리 2의 증명. S는 *Execute*, $Send(P_i^k, m)$와 *Test* 오라클을 사용할 수 있다. 호기심 많은 서버 S는 P_i와 P_j 사이의 프로토콜 실행을 방해하지 않고 P_i의 세션 키를 올바르게 추측하기 위해는 $X_i = g^{x_i}$가 P_i에 의해 만들어진 것이고 $X_j = g^{x_j}$가 P_j에 의해 만들어진 것일 때, X_i와 X_j를 이용하여 $H(g^{x_i x_j})$를 계산해야 한다. 다음에서 S를 이용하여 HDH 가정을 깨는 알고리즘 B를 설계한다. 해쉬 디피-헬만 실험의 입력 값 (G, q, g, U_1, U_2, W)를 입력 받은 B는 프로토콜의 메시지 속에 이 값들을 심는다. B에 대한 자세한 설명은 다음과 같다 :

1. B는 (G, q, g, U_1, U_2, W)를 입력받는다. S는 B에게 공개키 pk를 준다. B는 $P_i (1 \leq i \leq N)$에 대한 패스워드 pw_i를 선택한다. B는 $pw_i (1 \leq i \leq N)$와 공개 파라미터인 (G, q, g, H, E, M, pk)을 S와 공유한다. B는 $[1, N]$에서 i^*, j^*를 임의로 선택하고 $[1, q_s]$에서 t_1, t_2을 임의로 선택한다.
2. S의 각 오라클 쿼리에 대해 B는 KEDP를 다음과 같이 시뮬레이션 해 준다 :
   - B는 $P_{i*}^{t_1}$의 첫 번째 라운드 메시지로 $X_{i*} = U_1$을 사용하고, $P_{j*}^{t_2}$의 첫 번째 라운드 메시지로 $X_{j*} = U_2$를 사용한다.
   - 만약 S가 $P_{i*}^{t_1}$에 *Test* 쿼리를 하고 $P_{i*}^{t_1}$가 X_{j*}를 받았다면, B는 S에게 W를 반환한다. 만약 S가 $P_{j*}^{t_2}$에 *Test* 쿼리를 하고 $P_{j*}^{t_2}$가 X_{i*}를 받았다면, B는 S에게 W을 반환한다. 그렇지 않은 경우 B는 시뮬레이션 실패로 종료한다.

만약 B가 i^*, j^*, t_1와 t_2을 올바르게 추측했다면, B는 $U_1 = g^{u_1}$이고 $U_2 = g^{u_2}$일 때 $W = H(g^{u_1 u_2})$의 여부에 따라 실제 세션 키 또는 임의의 문자열을 반환한다. 따라서 다음의 부등식이 성립한다 :

$$
\begin{aligned}
&Adv_B^{HDH} \\
&= \Pr\left[B(U_1, U_2, W) = 1 \mid U_1 = g^{u_1}, U_2 = g^{u_2}, W = H(g^{u_1 u_2})\right] \\
&\quad - \Pr\left[B(U_1, U_2, W) = 1 \mid U_1 = g^{u_1}, U_2 = g^{u_2}, W = \{0,1\}^{\theta}\right] \\
&\geq \frac{1}{(Nq_s)^2}\left(\Pr\left[S(\,) = 1 \mid \begin{array}{l}\textit{Test} \text{ 쿼리에 대한 응답으로} \\ S\text{가 실제 세션키를 받음}\end{array}\right]\right. \\
&\qquad \left. - \Pr\left[S(\,) = 1 \mid \begin{array}{l}\textit{Test} \text{ 쿼리에 대한 응답으로} \\ S\text{가 임의의 문자열을 받음}\end{array}\right]\right) \\
&= \frac{1}{(Nq_s)^2} \cdot Adv_S^{curSvr}.
\end{aligned}
$$

이로써 주장이 증명된다.

Ⅳ. 결 론

본 논문에서는 삼자 간 패스워드 기반 키 교환 프로토콜에 대해 살펴보았다. 기존 제안된 삼자 간 패스워드 기반 키 교환 프로토콜 중 효율적인 프로토콜은 랜덤 오라클을 사용하였다. 그러나 랜덤 오라클과 같이 이상적인 함수 대신에 실제 함수를 사용하면 랜덤 오라클을 사용하는 모델에서는 안전한 스킴이 실제 환경에서는 안전하지 않다고 알려진 경우들이 많이 있다. 따라서 이에 대한 개선이 요구된다. 본 논문에서는 통신량과 계산량 그리고 라운드의 복잡도 관점에서 랜덤 오라클을 사용하지 않는 실용적인 삼자 간 패스워드 기반 키 교환 프로토콜을 제안하였다. 제안 프로토콜은 온라인 추측 공격과 오프라인 추측 공격에 안전하도록 설계되었으며 2 번의 라운드와 각 사용자 당 4 번의 지수승 연산만을 요구한다. 또한 제안 프로토콜은 전방향 안전성과 기지 키 공격에 대한 안전성을 제공한다.

참고문헌

[1] M. Abdalla, M. Bellare and P. Rogaway. "DHAES : an encryption scheme based on the Diffie-Hellman problem," Submission to IEEE P1363, 1998.

[2] M. Abdalla, M. Bellare and P. Rogaway. "The oracle Diffie-Hellman assumption and an analysis of DHIES," CT-RSA01, pp. 143-158, 2001.

[3] M. Abdalla, P.-A. Fouque, D. Pointcheval. "Password-Based Authenticated Key Exchange in the Three-Party Setting," In PKC'05, LNCS 3386, pp. 65-84, 2005.

[4] M. Abdalla and D. Pointcheval. "Interactive Diffie-Hellman assumptions with applications to password-based authentication," In Proc. of Financial Cryptography 2005, LNCS 3570, pp. 341-356, Springer-Verlag, 2005.

[5] J. W. Byun, I. R. Jeong, D. H. Lee, and C.-S. Park. "Password-Authenticated Key Exchange between Clients with Different Passwords," In ICICS'02, LNCS 2513, pp. 134-146, 2002.

[6] M. Bellare and P. Rogaway. "Random oracles are practical : a paradigm for designing efficient protocols," In Proc. of 1st Conference on Computer and Communications Security, pp. 62-73, ACM, 1993.

[7] M. Bellare, D. Pointcheval and P. Rogaway. "Authenticated key exchange secure against dictionary attack," In Eurocrypt 00, LNCS 1807, pp. 139-155, Springer-Verlag, 2000.

[8] M. Bellare and P. Rogaway. "Provably secure session key distribution - the three party case," In Pro. of the 28th Annual ACM Symposium on Theory of Computing, pp. 57-66, 1996.

[9] R. Canetti, O. Goldreich, and S. Halevi. "The random oracle methodology, revisited," In Pro. of the 32nd Annual ACM Symposium on Theory of Computing, pp. 209-218, 1998.

[10] R. Canetti, O. Goldreich and S. Halevi. "On the Random-Oracle Methodology as Applied to Length-Restricted Signature Schemes," In Pro. of 1st Theory of Cryptography Conference (TCC), LNCS 2951, pp. 40-57, 2004.

[11] Y. Ding, P. Horster. "Undetectable on-line password guessing attacks," ACM Operating Systems Review 29 (4) : 77-86 (1995).

[12] O. Goldreich and Y. Lindell. "Session-Key Generation using Human Passwords Only," In Pro. of Crypto '01, LNCS 2139, pp. 408-432. Springer-Verlag, 2001.

[13] S. Goldwasser and Y. Taumen. "On the (in)security of the Fiat-Shamir Paradigm," In Proc. of STOC '03, pp. 102-115, IEEE Computer Society, 2003.

[14] J. Katz, R. Ostrovsky, and M. Yung. "Forward secrecy in Password-only Key Exchange Protocols," In Pro. of SCN '02, LNCS 2576, pp. 29-44, Springer-Verlag, 2002.

[15] J. Katz, R. Ostrovsky, and M. Yung. "Efficient password-authenticated key exchange using human-memorable passwords," In Pro. of Eurocrypt'01, LNCS 2045, pp. 475-494, 2001.

[16] J. O. Kwon, I. R. Jeong, K. Sakurai, D. H. Lee,

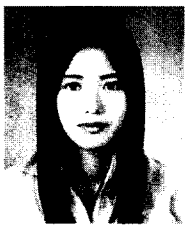
"Efficient Verifier-Based Password-Authenticated Key Exchange in the Three-Party Setting", Computer Standards & Interfaces, vol. 29 (5), pp. 513-520, 2007.

[17] C.-L. Lin, H.-M. Sun, M. Steiner, T. Hwang. "Three-party encrypted key exchange without server public-keys," IEEE Communication Letters 5 (12) : 497-499 (2001).

[18] J. B. Nielsen. "Separating Random Oracle Proofs from Complexity Theoretic Proofs : The Non-Committing Encryption Case," In Proc. of CRYPTO'02, LNCS 2442, pp. 111- 126, 2002.

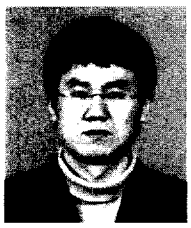
[19] M. Steiner, G. Tsudik, and M. Waidner. "Refinement and extension of encrypted key exchange," ACM SIGOPS Operating Systems Review, 29(3) : 22-30, July 1995.

〈著者紹介〉



권 정 옥 (Jeong Ok Kwon) 정회원
2000년 8월 : 동덕여자대학교 전자계산학과 졸업
2003년 2월 : 고려대학교 정보보호기술협동과정 석사 졸업
2007년 2월 : 고려대학교 정보경영공학전문대학원 박사 졸업
2007년 3월~2007년 8월 : 고려대학교 정보보호기술연구센터 박사후연구원
2007년 9월~현재 : 고려대학교 BK21 유비쿼터스 정보보호 사업단 연구교수
<관심분야> 암호프로토콜, 암호이론

김 기 탁 (Ki Tak Kim) 학생회원
2006년 8월 : 고려대학교 수학과 학사 졸업
2006년 9월~현재 : 고려대학교 정보경영공학전문대학원 석사 과정
<관심분야> 암호프로토콜, 암호이론



정 익 래 (Ik Rae Jeong) 정회원
1998년 2월 : 고려대학교 전산학과 학사 졸업
2000년 2월 : 고려대학교 전산학과 석사 졸업
2004년 8월 : 고려대학교 정보보호대학원 박사 졸업
2006년 6월~2008년 2월 : 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월~현재 : 고려대학교 정보경영공학부 조교수
<관심분야> 암호프로토콜, 암호이론, 계산이론

이 동 훈 (Dong Hoon Lee) 종신회원
1983년 8월 : 고려대학교 경제학과 학사 졸업
1987년 12월 : Oklahoma University 전산학과 석사 졸업
1992년 5월 : Oklahoma University 전산학과 박사 졸업
1993년 3월~1997년 2월 : 고려대학교 전산학과 조교수
1997년 3월~2001년 2월 : 고려대학교 전산학과 부교수
2001년 2월~현재 : 고려대학교 정보경영공학전문대학원 교수
<관심분야> 암호프로토콜, RFID/USN 보안, 프라이버시 보호기술