http://dx.doi.org/10.7840/kics.2015.40.8.1597

속성 기반 암호화 방식을 이용한 다중 서버 패스워드 인증 키 교환

박 민 경◆°, 조 은 상*, 권 태 경*

Multi Server Password Authenticated Key Exchange Using Attribute-Based Encryption

Minkyung Park◆°, Eunsang Cho*, Ted "Taekyoung" Kwon*

요 약

패스워드 인증 키 교환 프로토콜(Password Authenticated Key Exchange: PAKE)은 서버와 클라이언트가 서로 인증하고 키를 교환하는 알고리즘이다. 패스워드를 여러 개의 서버에 나누어 저장해서, 모든 서버가 손상되지 않으면 패스워드나 키가 유출되지 않는 알고리즘은 다중 서버 PAKE다. 속성 기반 암호화 방식에서는 암호화 하는 주체가 원하는 속성을 모두 만족하여야 복호화가 가능한 특징이 있다. 본 논문에서는 속성 기반 암호화 방식의 속성 값을 패스워드로 보아, 공개키/개인키를 별도로 생성하지 않고 공개키 기반 암호화가 가능한 다중 서버 PAKE 프로토콜을 제안한다. 제안한 프로토콜은 서버 당 한 번의 메시지 교환이 필요하며 사전(dictionary) 공격에 안전하다. 또한 사전 공격에 대한 위협 모델을 제시하고 보안 분석을 통하여 안전성을 검증하였으며, 사용한 암호 알고리즘의 수행시간 측정을 통해 제안한 프로토콜의 실현가능성(feasibility)을 검토한다.

Key Words : Password Authenticated Key Exchange, Attribute-based Encryption, Dictionary Attack

ABSTRACT

Password authenticated key exchange (PAKE) is a protocol that a client stores its password to a server, authenticates itself using its password and shares a session key with the server. In multi-server PAKE, a client splits its password and stores them to several servers separately. Unless all the servers are compromised, client's password will not be disclosed in the multi-server setting. In attribute-based encryption (ABE), a sender encrypts a message M using a set of attributes and then a receiver decrypts it using the same set of attributes. In this paper, we introduce multi-server PAKE protocol that utilizes a set of attributes of ABE as a client's password. In the protocol, the client and servers do not need to create additional public/private key pairs because the password is used as a set of public keys. Also, the client and the servers exchange only one round-trip message per server. The protocol is secure against dictionary attacks. We prove our system is secure in a proposed threat model. Finally we show feasibility through evaluating the execution time of the protocol.

※ 본 논문은 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구 결과물임을 밝힙니다.(No.2013R1A2A2A01016562)

◆° First and Corresponding Author : Seoul National University Department of Computer Science and Engineering, mkpark@mmlab.snu.ac.kr, 학생회원

* Seoul National University Department of Computer Science and Engineering, escho@mmlab.snu.ac.kr, tkkwon@snu.ac.kr

논문번호 : KICS2015-06-164, Received June 1, 2015; Revised August 11, 2015; Accepted August 11, 2015

Ⅰ. 서 론

개체가 스스로를 인증하려고 할 때(예를 들어, 사람들이 인터넷으로 웹 서버에 로그인을 하거나, 스마트폰에서 무선랜에 접속하려고 할 때, 허가된 단말인지 인증하는 등) 사용되는 것은 주로 패스워드이다. 패스워드 기반 인증 키 교환 프로토콜(Password Authenticated Key Exchange: PAKE)은 서버에 저장되어 있는 패스워드를 이용해서 서버와 클라이언트가 서로를 인증하고 공유 키를 교환하는 알고리즘이다. 패스워드는 인증서나, 생체 정보 등 다른 인증 방식에 비해 상대적으로 기억하기 쉽고, 사용하기도 쉽다는 장점 때문에 많이 사용되고 있다. 그러나 이러한 패스워드의 장점이 단점으로도 작용할 수 있다. 보안에 취약하고 낮은 엔트로피를 가진 패스워드는 공격자가 추측하기 쉽고, 또한 보안에 취약한 서버는 공격자의 공격에 인증정보를 노출시킬 수도 있다. 같은 패스워드를 여러 사이트에서 사용하기 때문에 한 서버의 패스워드가 유출이 되면 다른 사이트의 패스워드도 더 이상 안전하지 않다[1].

PAKE는 서버의 수에 따라 2가지로 분류할 수 있다. 첫 번째 방식은 패스워드를 저장하는 서버가 1개 존재하는 단일 서버 방식이다. 해당 서버의 패스워드나 개인키가 탈취되면 더 이상 안전하지 않다는 단점이 있다. 이를 극복한 방식이 두 번째 방식인 다중 서버 PAKE이다. 다중 서버 PAKE는 클라이언트의 패스워드를 여러 개의 서버(N개)에 저장한다. N개의 서버 중 N-1개의 서버가 손상되어도 클라이언트의 패스워드 일부만 유출이 되기 때문에 상대적으로 안전하다. 또한 보안이 뛰어난 제 3의 서버를 패스워드 저장 서버로 이용하면 다른 서버에서 패스워드가 탈취되어도 보안이 뛰어난 서버의 개인키가 탈취될 확률이 낮다.

서버 자체의 안정성 이외에도 PAKE 프로토콜이 안전해야 한다. PAKE는 사전 공격(dictionary attack)에 취약하기 때문이다. 사전 공격은 추측 공격의 일종으로, 성공할 가능성이 있는 패스워드들을 추측해서(사전을 만들어서) 패스워드를 알아내는 방법이다. 사전 공격은 크게 온라인 사전 공격과 오프라인 사전 공격으로 나뉜다[2]. 이 공격은 프로토콜에 따라서 패스워드가 쉽게 노출될 수도, 노출되지 않을 수도 있기 때문에 PAKE 프로토콜은 이에 안전해야 한다.

패스워드를 안전하게 전달하기 위해 주로 비대칭키 기반 암호화를 한다. 비대칭키 기반 암호화 방식은 키 쌍이 필요하다. 서버와 패스워드 간에 미리 키 쌍을 나누거나 키 쌍을 생성해내는 방법을 나누는 등의 양쪽의 합의가 있어야 한다. 또한 다중 서버를 사용하는 경우에는 클라이언트와 각 서버가 다른 키 쌍을 공유해야 한다.

본 논문에서는 패스워드를 비대칭키로 사용하는 다중 서버 PAKE를 제안하여 별도의 키(비대칭키)를 생성하지 않도록 한다. 이를 가능하게 하는 것은 속성 기반 암호화 방식(Attribute-based encryption: ABE)이다. 속성 기반 암호화 방식은 Goyal 등[3]에 의해 고안된 암호화 방식이다. 메시지를 암호화하는 주체는 이를 복호화는 주체의 속성을 이용해 암호화한다. 암호화에 사용된 속성을 모두 가지고 있어야 암호문을 복호화 할 수 있다. 제안하는 프로토콜에서는 패스워드를 속성으로 보았다. 패스워드를 가진 서버와 클라이언트만 복호화가 가능하다. 속성(패스워드)을 가지고 공개키 기반 암호화를 하기 때문에 별도의 키 쌍이 필요하지 않다. 암호화는 속성과 난수를 이용해 암호화마다 새로운 키 쌍을 만든다. ABE의 자세한 설명은 2장에서 설명하겠다.

남은 장은 다음과 같이 구성되어 있다. 2장에서는 관련 연구와 ABE에 대한 자세한 방법을 설명하고, 논문에서 제안하는 프로토콜의 기본이 되는 모델에 대해 정의한다. 3장에서는 제안하는 PAKE 프로토콜을 자세하게 설명한다. 4장에서는 제안하는 PAKE 프로토콜의 안전성을 점검한다. 5장에서는 성능을 간단히 보이고, 6장에서 결론을 맺는다.

Ⅱ. Background & Definition

2.1 PAKE Algorithms

Bellovin과 Merritt[4]는 RSA나 El gamal를 이용한 사전 공격에 안전한 PAKE 알고리즘을 발표하였다. 그 이후로 인증서를 동시에 사용하는 PKI 기반 PAKE[5], ID를 동시에 사용하는 ID기반 PAKE[6], Smart card를 이용한 방식 등이 발표되었다[7]. 비교적 최근에는 생체 정보를 이용한 방식 등 다양한 인증 정보와 패스워드를 함께 사용하는 프로토콜에 대한 연구가 있다[8,9].

다중 서버 방식 PAKE도 다양한 프로토콜이 고안되었다. 다중 서버 방식도 세부적으로 2개의 서버를 사용한 방식과 3개 이상의 서버를 사용한 방식으로 나눌 수 있다. 2개의 서버에 패스워드를 나눠 저장하고 두 서버 모두 패스워드 인증 과정에 참여하는 PAKE가 발표되었다[2,10]. 패스워드를 3개 이상 서버에 저장하는 방식으로는, N개의 서버에 패스워드를

저장하여 그 중 K개의 서버와 모두 인증을 성공해야 사용자 인증이 가능한 threshold PAKE[11]가 있다. 그러나 위의 다중 서버 PAKE 프로토콜들은 메시지 교환이 많이 발생하거나, 제한된 수의 서버에만 적용할 수 있다는 한계가 있다.

2.2 Attribute-based Encryption

속성 기반 암호화 방식(ABE)[3]은 신원 기반 암호화 방식(ID-based encryption: IBE)[12]으로 부터 파생되었다. IBE는 전자 우편 주소와 같은 고유한 ID를 공개키로 사용하는 공개키 암호 시스템으로 Shamir에 의해 처음 제안되었다. 그 후에 일반적인 IBE에서 진화한 Fuzzy Identity-Based Encryption(FIBE)이 Sahai와 Waters[13]에 의해 제안되었다. 이 방식에서는 identity를 여러 가지 속성의 집합으로 정의한다. 암호화할 때 사용된 공개키의 집합과 복호화 하려는 개인키의 집합의 distance metric의 차이가 어느 threshold 보다 작으면 복호화가 가능하다.

이에서 한 층 더 진화한 것이 ABE이다. ABE는 접근 제어를 위해 Goyal 등[3]에 의해 고안되었다. 사용자의 데이터는 서버에 암호화되어 저장된다. 따라서 서버가 공격당해도 암호화 된 데이터는 복호화 되기 쉽지 않기 때문에 기밀성이 유지된다. 데이터를 암호화해서 저장하면, 데이터를 소비하고자 하는 소비자는 이를 복호화하기 위한 키가 필요하다. 소비자에 따라 데이터마다 다른 접근 제어가 필요한 상황에서는 다른 키 쌍을 생성해야 하고, 이러한 키 쌍을 효율적으로 생성/관리 하는 것은 중요한 일이다. 사용자 속성에 따른 유연성 있는 접근 제어를 ABE에서 가능하게 했다.

예를 들어 인사과, 정보과 등 부서에 따라서 데이터 접근 권한이 달라질 수 있고, 사장이나 부장 등 직책이나 맡은 역할에 따라서도 접근 권한이 달라질 수 있다. ABE에서는 인사과이면서 동시에 부장이거나 혹은 사장에게만 접근 권한이 있도록 데이터를 암호화할 수도 있고, 정보과이면서 동시에 부장에게만 접근 권한이 있도록 데이터를 암호화 할 수 도 있다. 첫 번째 경우에는 인사과 키와 부장 키를 동시에 가진 소비자 혹은 사장 키를 가진 소비자만 복호화가 가능하고, 두 번째 경우에는 정보과 키와 부장 키를 가진 소비자만이 복호화가 가능하다.

Goyal 등이 고안한 ABE는 쌍(Pairing)기반 암호화 방식이다. **소수 위수(prime order)**가 p인 2개의 순환 군(cycle group) G_1, G_2에 대해 bilinear map $e : G_1 \times G_2 \rightarrow G_t$는 다음과 같은 특별한 성질을 갖는다는 점을 이용한다.

1) $\forall\, a, b \in F_p^*, e(a \times g, b \times g) = e(g,g)^{ab}$ (1)

2) $e(P,Q) \neq 1$ (2)

3) e를 효과적으로 계산하는 알고리즘이 존재한다.

위의 특징을 이용하여, ABE 알고리즘을 설명할 수 있다. ABE는 Setup, Encryption, Key Generation, Decryption 4단계로 구성된다. 간단히 각 단계를 설명하면 다음과 같다.

Setup 기본 public parameter를 초기화 한다. 위의 예에서, 정보부, 인사부와 같은 부서나 직책 같은 속성을 나타내는 속성 집합 $v = \{1, ..., n\}$를 규정하고, 각 v의 원소에 대한 속성 특징값 $t_i (i \in v)$와 y 값을 랜덤으로 선택한다.

public parameter는 $g^{t_1}, ..., g^{t_n}, e(g,g)^y$ 이다.

master key(비밀값)는 $t_1, ..., t_n, y$ 이다.

Encryption (M, γ, PK) 메시지 $M \in G_2$을 속성 γ로 암호화하기 위해 랜덤인 수 $s \in Z_p$를 선택한다. 이 때, 암호문 E는 다음과 같다.

$$E = \left(\gamma, Me(g,g)^{sy}, g^{t_i s}\right) \; where \; i \in \gamma \quad (3)$$

Key generation 암호문을 복호화하기 위해, 풀어낼 키를 생성한다. 암호문 E로 부터 받은 $g^{t_i s}$를 이용해 개인 키를 만든다.

$$g^{y_i / t_i} (i \in \gamma) \quad (4)$$

Decryption(E, D) Key generation 과정에서 생성한 키를 이용하여 암호문 E의 E'을 복호화 한다.

$$E' / e(g,g)^{sy} = M \quad (5)$$

2.3 Model & Definition

이 절에서는 보안 모델에 대해 설명하고 앞으로 사용할 용어에 대해 정의할 것이다. 사용할 모델은 다른 많은 PAKE 알고리즘 [2,10,11]에서 사용한 모델을 기반으로 하였다.

제안하는 PAKE 프로토콜은 클라이언트와 N개의

서버, 2종류의 사용자가 참여한다. 확률적 다항시간 공격자(Probabilistic Polynomial Time: PPT)는 A로 표기한다. 사용자 U는 공격자 A를 제외한 클라이언트 C와 서버$S_k(1 \le k \le N)$를 총칭하고, U로 표기한다. 사용자가 프로토콜에서 대화하는 상대방을 파트너라고 한다. 서버는 N개가 있다고 가정하며 $\{S_1, S_2, ..., S_N\}$으로 표현한다. N개의 서버 중 적어도 1개는 안전하다고 가정한다. 하나의 클라이언트의 패스워드 $pw_C \in Z_p$는 나눠져서 N개의 서버에 저장된다. 각 서버에 저장된 패스워드는 $\{pw_1, ..., pw_N\}$으로 표현되고 $pw_C = pw_1 + ... + pw_N$의 특징을 만족한다. 초기 패스워드의 분배는 공격자의 간섭 없이 안전하게 이루어진다고 가정한다.

각 사용자는 프로토콜을 다른 사용자와 여러 번 수행할 수 있다. 매번 수행할 때마다, 사용자의 새로운 인스턴스가 프로토콜에 참여하며, 사용자 U의 인스턴스 i를 U_i라고 표기한다.

- U_i는 다양한 변수(variables)를 가지고 있다. 파트너의 아이디를 나타는 pid, 세션 아이디를 나타내는 sid 그리고 사용자 U가 파트너와 공유한 세션 키인 sk가 있다. 파트너의 아이디란 U_i가 프로토콜을 통해 통신하고 있는 사용자의 아이디를 뜻한다. 세션 아이디는 프로토콜의 수행을 구별해 주는 아이디이다. 세션 키는 프로토콜이 종료된 후 파트너와 맺게 되는 키를 뜻하는데, 클라이언트 C에게는 $(sk^i_{C,S_1}, sk^i_{C,S_2}, ..., sk^i_{C,S_N})$의 전체 또는 일부를 뜻하고, 서버 S_k에게는 sk^i_{C,S_k}를 뜻한다.
- acc, term는 각각 U_i가 성공적으로 프로토콜이 다 수행된 후 세션 키가 의도한 세션 키인지, 프로토콜이 전부 수행되었는지를 나타내는 변수이다. 따라서 클라이언트와 서버 $S_k(k \in \{1, ..., N\})$가 각각 같은 세션 키를 공유한 다는 것을 의미한다.

공격자 A는 프로토콜에서 사용되는 모든 환경을 제어 할 수 있다고 가정한다. 공격자는 오라클 모델에 접근하여, 프로토콜 환경을 제어한다. 여기서 사용되는 오라클 모델의 타입은 다음과 같이 정의한다.

- send(U, i, M): 사용자U의 인스턴스 i인 U_i에게 메시지 M을 전송한다. U_i는 상태를 업데이트 하고, 그에 대한 메시지를 출력하는데, 이 상태나 메시지는 A가 알 수 있다.
- execute(C, i, $\{(S_1, l_1), ..., (S_N, l_N)\}$): 클라이언트 C의 인스턴스 i는 서버 S_k의 인스턴스 l_k $(k \in \{1, ..., N\})$와 전체 프로토콜을 수행한다. 수행 시에 오고 가는 메시지나 상태를 공격자A는 모두 볼 수 있다. 또한 개인 키가 유출된 서버가 있다면, 유출된 서버의 내부 상태도 A에게 주어진다. 이 오라클은 메시지를 엿듣는 공격을 하는 공격자를 표현한다.
- corrupt(U): 사용자 U의 개인 키가 유출될 상황을 모델링한다. 이 오라클은 일부 서버나 클라이언트가 해킹당한 상황을 나타낸다.
- reveal(U, U', i): 현재 세션 키 $sk^i_{U,U'}$을 A에게 알린다. 이 오라클은 세션 키가 유출이 된 상황을 나타낸다.
- test(U, U', i): 이 오라클은 실제 상황을 나타내는 모델이 아니라, 보안성을 검증하기 위해 사용한다. U_i가 랜덤하게 비트 b를 생성한다. 만약 b=1이면 세션 키 $sk^i_{U,U'}$를 A에게 알려준다. b=1이 아니면, 랜덤한 세션 키를 A에게 알려준다. 이 오라클은 프로토콜 수행 도중 아무 때나 실행 할 수 있고, 단 한 번만 가능하다.

U와 U'의 개인 키가 유출되지 않았고, 공격자가 reveal(U, U', i)과 reveal(U', U, j)를 질의하지 않았을 때 세션 키는 신선하다(fresh)고 한다.

앞서 설명한 정의를 바탕으로, 보안성을 정의할 것이다. 공격자 A가 acc^i_U=TRUE이고, 신선한 U_i에 Test(U, U', i) 질의한 아웃풋 b와 공격자 A가 선택한 b'의 비트가 같으면, 즉 b=b'인 경우 '공격자 A의 공격은 성공했다'고 하자. 이러한 상황을 Suc이라고 표기한다. 공격자 A가 프로토콜 P에서 얻을 수 있는 이득을 Adv^P_A라고 표기하고 $Adv^P_A = 2 \times \Pr[Suc] - 1$라고 정의한다.

Ⅲ. Proposed protocol

이번 장에서는 제안하는 PAKE 프로토콜에 대해 설명한다. 이번 장에서 언급하는 암호화/복호화는 ABE를 이용한 암호화/복호화다.

프로토콜이 시작되기 이전에 패스워드와 ABE의 public parameter를 분배하는 과정이 필요하다. 클라이언트 C는 패스워드 $pw_C \in Z_p$를 생성한다. $pw_C = pw_1 + ... + pw_N$의 조건에 맞게 서버 S_k

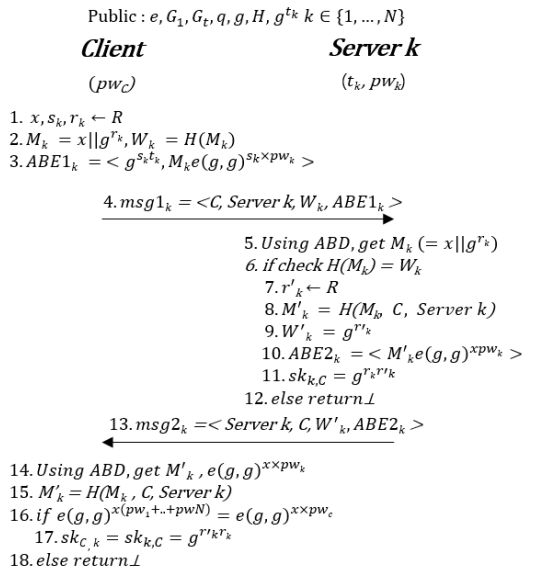

그림 1. 프로토콜의 수행 과정
Fig. 1. An execution of the protocol

$(k \in \{1, 2, ..., N\})$에 pw_k를 저장한다. pw_k는 안전한 경로로 분배한다고 가정한다. 각각의 서버에서는 다음과 같은 초기 과정을 수행한다. 서버 S_k는 비밀값 $t_k \in G_1$를 생성한다. 이 값을 이용해 g^{t_k}를 생성하며, 이 값은 public parameter로 공개하는 정보가 된다.

클라이언트 C는 $k \in \{1, ..., N\}$인 서버 S_k의 공개 값 g^{t_k}과 pw_k값을 이용해 메시지를 암호화 할 수 있고, t_k를 알고 있는 S_k는 복호화가 가능하다. 각 서버는 pw_k와 랜덤한 수 x를 이용해 메시지를 암호화 한다. 클라이언트는 pw_C를 알고 있으므로 $e(g,g)^{pw_C}$ 를 알 수 있다. 또한, 클라이언트가 x값을 알고 있다면 $e(g,g)^{x \times pw_C}$ 값을 알 수 있다. 따라서 각 서버가 생성한 암호문의 $e(g,g)^{x \times pw_k}$ 를 이용해 $e(g,g)^{x \times pw_C}$ 를 구할 수 있게 된다.

이러한 원리를 이용해 인증 프로토콜이 수행 된다. 그림 1에 프로토콜의 수행 과정을 명시하였다. H는 충돌할 확률이 무시할 정도로 작은 해시 함수를 나타내고 R은 랜덤한 수를 생성하는 생성기를 나타낸다. $s \leftarrow R$은 랜덤한 수 s를 생성함을 의미한다. ABD는 복호화 과정인 Attribute-based Decryption을 의미한다.

클라이언트는 ABE 암호화를 통해 자기 자신을 서버에게 인증한다. 클라이언트는 x값과 세션 키 생성을 위한 Diffie-Hellman 값(g^{r_k})을 암호화 한다(그림 1의 1~3번). ABE 암호화과정에서 비밀 값 s_k를 매번 생성하기 때문에 다른 서버의 메시지나 s_k을 클라이언트 이외에는 알아낼 수 없고, $s_k \times pw_k$를 일회용 비밀번호처럼 사용하여 서버에게 보내기 때문에 pw_k가 노출되지 않는다. 서버가 메시지를 복호화 하려면 $g^{s_k \times pw_k}$값을 알아야 한다. 이 값은 $g^{s_k \times t_k}$를 이용해 계산하므로, s_k값이 노출되지 않고 M_k를 얻을 수 있다(그림 1의 5번). 클라이언트가 생성한 W_k의 값과 복호화로 얻은 메시지 M_k의 해시 값이 일치하게 되면 정상적으로 복호화가 된 것으로 간주하고, 클라이언트 인증이 끝난다.

클라이언트 인증이 끝나면, 서버는 자신을 클라이언트에게 인증한다. 클라이언트는 메시지 M_k에 모든 서버가 공통으로 사용할 비밀 값 x와 세션 키를 만들어 내는데 사용하는 g^{r_k}값을 임의로 생성해서 전송했다. 비밀 값 x를 이용해 서버 S_k는 자신의 패스워드 pw_k를 가지고 암호화를 하여 클라이언트에게 전송한다(그림 1의 7~13번). 클라이언트는 서버로 부터 $e(g,g)^{x \times pw_k}$를 받게 된다. 이렇게 받은 값을 모두 곱하게 되면 $e(g,g)^{x \times pw_c}$를 얻게 된다. 이를 정확하게 도출해 내면 클라이언트는 서버가 정확한 패스워드를 가지고 있음을 동시에 인증할 수 있다. 양방향 인증이 성공하면 클라이언트와 각 서버는 서로 공통적인 세션 키 $g^{r'_k x}$를 갖게 된다.

Ⅳ. Security analysis

프로토콜이 사전 공격에 안전하다는 것을 다음과 같이 정의한다. [2,10,11]

Definition 1. 사전의 크기를 Dic이라고 하자. (pw_c가 생성될 수 있는 공간이며 G_1의 크기와도 같다.) Q_p는 공격자가 온라인 사전공격을 시도한 횟수이고, α는 무시할 정도로 작은 수이다. PPT 공격자 A가 최대 Q_p번의 온라인 사전공격을 한다고 할 때, 프로토콜 P에 대해

$$Adv_A^P \leq Q_p / Dic + \alpha \qquad (6)$$

라면 P는 사전공격에 안전하다.

Definition 1에서 Q_p가 충분히 작고, Dic이 충분히 크다면 Adv_A^P는 무시할 정도로 작다. 따라서 온라인 사전 공격에 안전하다고 볼 수 있다. 2장에서 설명한 모델을 바탕으로, 제안하는 프로토콜의 Definition 1에 따라 사전공격에 안전함을 검증할 것이다.

프로토콜을 수행하는 환경을 시뮬레이터 S라고 하자. 서버는 최대 N-1개의 서버($k \in \{2, 3, ..., N\}$)에 corrupt(S_k)를 질의하였다고 가정한다. 제안한 프로토콜을 P0라고 하고, 시뮬레이터에서 P0의 몇 가지 변화 상황에 대해 공격자의 이익 Adv_A^P를 측정한다. 이를 바탕으로 최종으로 제안하는 프로토콜이 안전함을 보인다.

프로토콜 P1: 시뮬레이터는 P0에서 두 가지 상황을 제외하고 똑같이 동작한다. (1) 메시지($msg1_k$, $msg2_k$)는 오라클로부터 중복 생성될 수 있다. 중복 생성되면 프로토콜을 중단한다. (2) 해시 함수 H에서 충돌이 발생하면, 프로토콜을 중단한다.

첫 번째 상황은 무시할 수 있을 정도의 확률로 발생한다. 두 번째 상황은 전제 조건에서 해시 함수에서는 충돌이 무시할 정도로 작은 확률로 일어난다고 가정한다. 따라서 $|Adv_A^{P0} - Adv_A^{P1}|$은 무시할 정도로 작다.

프로토콜 P2: 시뮬레이터는 P1과 한 가지 상황을 제외하고 똑같이 동작한다. 공격자가 execute(C, i, $\{(S_1, l_1), ..., (S_N, l_N)\}$)를 질의할 때, corrupt($S_k$)가 질의 되지 않은 S_1이 생성하는 평문 M_k는 임의의 수가 된다.

메시지 M_k는 매번 임의로 클라이언트로부터 생성되고, 서버별로 다른 값을 갖게 된다. 따라서 프로토콜 P1과 P2의 차이점은 공격자 A가 ABE를 키 없이 복호화 할 수 있느냐에 관련된 문제이다. ABE에서 평문 M_k를 추측할 수 없으므로 따라서 $|Adv_A^{P1} - Adv_A^{P2}|$는 무시할 정도로 작다.

프로토콜 P3: 시뮬레이터는 P1과 한 가지 상황을 제외하고 똑같이 동작한다. 공격자가 execute(C, i, $\{(S_1, l_1), ..., (S_N, l_N)\}$)를 질의할 때, corrupt($S_k$)가 질의된 적이 없는 서버 S_1에서 생성하는 $sk_{S_1,C}$, $sk_{C_1,S}$는 임의의 수가 된다.

세션 키는 Diffie-Hellman 알고리즘에 의해 생성된다. 따라서, 프로토콜 P2와의 차이점은 Diffie-Hellman 문제를 풀 수 있는지에 관련된 문제이다. 이는 어렵다(hard)고 알려져 있으므로, $|Adv_A^{P3} - Adv_A^{P2}|$는 무시할 정도로 작다.

프로토콜 P4: 시뮬레이터가 send(S_k, j, $msg1_k$)와 send(C,i,$msg2_k$)가 질의되었을 때의 응답을 다음과 같이 바꾼다.

$msg1_k$가 공격자에 의해 생성된 경우(오라클에 의해 생성되지 않은 경우), send(S_k, j, $msg1_k$) 쿼리는 $acc_{S_k}^j$=TRUE일 때만, $msg1_k$가 유효하다. $msg2_k$도 마찬가지로 공격자에 의해 생성된 경우, send(C, i, $msg2_k$)는 acc_C^i=TRUE일 때만 $msg2_k$가 유효하다. $msg1_k$(혹은 $msg2_k$)가 공격자에 의해 생성되었고, $acc_{S_k}^j$(혹은 acc_C^i)가 TRUE라면, $acc_{S_k}^j = \nabla$ ($acc_C^i = \nabla$)라고 하자. P4에서는 공격자의 성공(Suc)을 다르게 정의할 것이다. 만약 공격자 A가 send를 신선한 서버 인스턴스 Sj에게 질의했을 때, $acc_C^i = \nabla$이거나, 공격자A가 send(S_k,j,$msg2_k$)를 신선한 클라이언트 인스턴스 C_i에게 질의했을 때, $acc_{S_k}^j = \nabla$이면 adversary는 성공했다고 간주한다. 위에서 나열한 상황이 아닌 경우의 Suc의 정의는 P3와 같다. 따라서 $Adv_A^{P3} \leq Adv_A^{P4}$가 된다.

프로토콜 P5: 시뮬레이터는 두 가지 상황을 제외하고 똑같이 동작한다. 공격자가 send(S_k, j, $msg1_k$)를 질의할 때, corrupt(S_k)가 질의된 적이 없는 서버 S_1에서 생성하는 평문 M_1는 임의의 수가 된다. 또한 S_1에서 발생하는 send(C, i, $msg2_k$)의 평문 M'_k또한 임의의 수가 된다.

메시지 M_k는 매번 임의로 클라이언트로부터 생성되고, 서버별로 다른 값을 갖게 된다. 또한 M'_k도 M_k를 연산한 결과물이기 때문에 매 프로토콜 수행시마다 달라진다. P2와 마찬가지로 ABE를 복호화 할 수 있는 가능성에 관한 문제로, $|Adv_A^{P4} - Adv_A^{P5}|$는 무시할 정도로 작다.

프로토콜 P5에서, $msg1_k$와 $msg2_k$는 pw_C와 독립적이다. 따라서 오프라인 사전 공격은 성공하지 못한다. 매 번 다른 s와 x를 생성할 뿐 아니라 서버 N개에 나누어져 있기 때문에 pw_C가 그대로 사용되는 경우도 없다. 온라인 공격을 하는 공격자 A는 프로토

콜 P5에서 3가지 경우에 성공(Suc)할 수 있다. (1) 공격자가 유효한 $msg1_k$를 생성하여 send(S_k, j, $msg1_k$)를 신선한 서버 S_k에게 질의하는 경우이다. (즉, $acc^j_{S_k}$=▽) 이 경우를 Suc1이라고 표기한다. (2) 공격자가 유효한 $msg2_k$를 생성하여 send(C, i, $msg2_k$)를 신선한 C에게 질의하는 경우이다.(즉, acc^i_C=▽) 이 경우를 Suc2라고 표기한다 (3) (1)의 경우도 아니고, (2)의 경우도 아닐 때, Ci나 S_{kj}에게 Test를 질의해서 $b = b'$ 이 되는 경우이다.

$\Pr[Suc1]$과 $\Pr[Suc2]$를 측정하기 위해, 공격자 A가 corrupt(S_l)하였다고 가정하고, ($l \in \{2, ..., N\}$) 4가지 경우로 나눈다. 아래 4가지 경우에 corrupt가 질의되지 않은 서버를 S_k라 하자.

Case 1. 공격자 A는 $msg1_k$ <C, Server k, W_k, $ABE1_k$>를 바꾼다. 하지만 ABE는 선택된 암호문 공격(Chosen Ciphertext Attack: CCA)에 안전하다. 따라서 $\Pr[Suc1]$은 무시할 정도로 작다.

Case 2. 공격자 A는 $msg1_k$'<C, Server k, W_k, $ABE1_k$>를 새로 생성한다. 공격자가 생각하는 암호 pw_A를 이용해, M_k를 암호화하고, W_k, $ABE1_k$를 생성한다. 이 경우에 $\Pr[Suc1]$은 Q_p/Dic+α이다.

Case 3. 공격자 A는 $msg2_k$ <C, Server k, W'_k, $ABE2_k$>를 바꾼다. 이 경우도 Case1과 마찬가지로 $\Pr[Suc2]$은 무시할 정도로 작다.

Case 4. 공격자 A는 $msg2_k$'<C, Server k, W'_k, $ABE2_k$>를 새로 생성한다. 공격자가 생각하는 암호 pw_A를 이용해 M'_k를 암호화하고, W'_k, $ABE2_k$를 생성한다. 이 경우에 $\Pr[Suc2]$는 Q_p/Dic+α다.

위의 4가지를 고려해 보았을 때, $\Pr[Suc1 \cup Suc2]$ = Q_p/Dic+α가 된다.

$$\Pr_A^{P5}[Suc] \leq 1/2(1 + Q_p/dic + \alpha) \quad (7)$$

가 되고,

$$\begin{aligned} Adv_A^{P5} &= 2\Pr_A^{P5}[Suc] - 1 \\ &\leq Q_p/Dic + \alpha \end{aligned} \quad (8)$$

가 된다. 따라서 사전 공격에 안전하다.

V. 실 험

이번 장에서는 제안한 프로토콜에서 사용하는 ABE의 수행 시간과 그 프로토콜의 수행 시간을 측정한 결과를 분석하였다. JPBC library[1]를 이용하여 Java로 알고리즘을 구현하였다. 실험을 수행한 환경은 다음과 같다. Intel i7-4770 CPU, 8GB RAM PC 1대에서 측정하였다. 같은 보안 레벨에서 측정하기 위해 ABE는 160bit, RSA는 1024bit의 키 길이를 이용하였다[14].

먼저 ABE와 RSA의 암호화, 복호화 속도를 1000회 측정한 평균을 Table 1에 나타내었다. 암호화, 복호화에 사용한 메시지 길이는 동일하게 20byte로 하였다. Table 1에서 볼 수 있듯이, ABE가 RSA에 비해 암호화에는 약 150배, 복호화에는 약 7배가 느리다. ABE는 bilinear pairing연산을 사용하기 때문에 다른 공개키 기반 암호화 방식인 RSA나 ElGamal에 비해 속도가 느리다. 하지만, pairing 연산이나 ABE의 성능을 향상시키려는 연구가 지속적으로 있고[15, 16], 이에 따라 ABE 알고리즘의 성능 향상을 기대할 수 있다.

두 번째로 측정한 것은 제안한 프로토콜의 평균 수행 시간이다. 제안한 알고리즘은 서버의 개수가 증가해도, 증가하는 서버의 개수와 1번의 메시지 교환만 하기 때문에 성능의 차이가 크지 않다. 이를 보이기 위해, 서버의 수를 1개부터 10개까지 변화시키면서 프로토콜 수행 시간을 5번씩 측정하여 평균을 구하였다. 측정 결과를 그림 2에서 나타내었다. 서버와 클라이언트에서 수행하는 수행 속도는 선형적인 양상을 보인다. 서버를 여러 개 사용하는 다른 프로토콜의 경우[11] 서버의 수 N에 따라 메시지 교환이 N^2배 일어나게 된다. 이에 비해 서버의 개수가 많은 상황이라면 더 나은 성능을 기대할 수 있다. 또한 서버와 클라이언트가 같은 머신이 아닌 개별 머신에서 수행한 결과를 측정한다면, 성능 향상이 예상된다. 또한, 프로토콜에서 서버와 클라이언트 당 ABE 알고리즘을 2번씩 수행하는데, 그림 2의 결과에 따르면 대부분의 프로토콜 수행 시간이 암호화, 복호화에 소요된다고 예상할

표 1. ABE, RSA 수행 시간 평균
Table 1. The average time for running RSA and ABE

	Encryption	Decryption
ABE	35,076 us	7,126 us
RSA	203 us	1,035 us

1) JPBC library 2.0.0 (http://gas.dia.unisa.it/projects/jpbc)

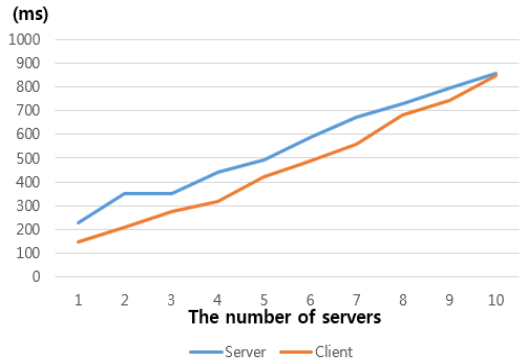

그림 2. 서버의 수에 따른 프로토콜 수행 시간
Fig. 2. Comparison of the protocol execution time

수 있다. 따라서 ABE의 알고리즘 성능이 향상되면, 제안하는 프로토콜의 성능이 크게 개선될 것으로 보인다.

Ⅵ. 결 론

본 논문에서는 속성 기반 암호화 방식을 이용하여 클라이언트의 패스워드로부터 각 서버의 개인키를 생성하여 사용할 수 있는 다중 서버 PAKE 프로토콜을 제안하였다. 기존 PAKE 프로토콜은 공개키 기반 암호화 방식을 사용하기 위해 공개키, 개인키 쌍을 패스워드와 별도로 생성 및 관리해야 하고, 다수의 메시지 교환이 필요한 부담이 있다. 더욱이 기존 다중 서버 PAKE 프로토콜은 메시지 교환이 기존 PAKE 프로토콜보다 더 많이 필요하다. 제안하는 프로토콜은 이러한 기존 프로토콜의 단점을 완화하여 별도의 공개키, 개인키 쌍의 관리 부담을 해소하고, 서버 당 1회의 메시지 교환으로 인증과 키 교환이 가능하다. 또한 서버의 개수에 상관없이 적용 가능하며, 서버의 개수가 N개라고 할 때 N-1개의 서버가 손상되어도 사전공격에 안전함을 보였다.

References

[1] J.-C. Park, "A scheme for secure storage and retrieval of (ID, password) pairs using smart cards as secure and portable storages," *J. KICS*, vol. 39B, no. 06, pp. 333-340, Jun. 2014.

[2] X. Yi, et al., "ID-Based two-server password-authenticated key exchange," *Springer: Computer Security -ESORICS 2014,* vol. 8713, pp. 257-276, Sept. 2014.

[3] V. Goyal, et al., "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Security(CCS '06),* pp. 89-98, VA, USA, Oct. 2006.

[4] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocol secure against dictionary attack," *IEEE Symp. Research in Security and Privacy*, pp. 72-84, Oakland, CA, May 1992.

[5] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Trans. Inf. Syst. Security*, vol. 2, no. 3, pp. 230-268, 1999.

[6] X. Yi, et al., "Identity-based password-authenticated key exchange for client/server model," *SECRYPT 2012,* pp. 45-54, Rome, Italy, Jul. 2012.

[7] J. Xu, et al., "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723-728, Jun. 2008.

[8] B. Cho and J. Park, "Technology review on multimodal biometric authentication," *J. KICS*, vol. 40, no. 1, Jan. 2015.

[9] J. Park, et al., "QR-code based mutual authentication system for web service," *J. KICS*, vol. 39B, no. 4, pp. 207-215, Apr. 2014.

[10] J. Katz, et al., "Two-server password-only authenticated key exchange," *ACNS*, vol. 3531 of LNCS, pp. 1-16, NY, USA, Jun. 2005.

[11] P. MacKenzie, et al., "Threshold password-authenticated key exchange," *Crypto 2002,* pp. 385-400, Califoria, USA, Aug. 2002.

[12] A. Shamir, "Identity based cryptosystems and signature schemes in advances in cryptology," *CRYPTO 84*, vol. 196 of LNCS, pp. 37-53, 1984.

[13] A. Sahai, et al., "Fuzzy identity based encryption in advances in cryptology," *Eurocrypt*, vol. 3494 of LNCS, pp. 457–473, 2005.

[14] E. Barker, et al., NIST Special Publication 80 0-57 Part3 Rev. 1: Recommendation for Key

Management - Part1: General (Revision3), (2012), Retrived May 31, 2015, from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57Pt3r1.pdf

[15] S. Kwon, “Efficient tate pairing computation for elliptic curves over binary fields,” *ACISP*, vol. 3574 of LNCS, pp. 134-145, Barisbane, Australia, Jul. 2005.

[16] K. Javeed, et al., “Efficient montgomery multiplier for pairing and elliptic curve based cryptography,” *9th Int. Commun. Syst., Netw. & Digital Signal Process. (CSNDSP)*, pp. 255-260, Manchester, Jul. 2014.

박 민 경 (Minkyung Park)

2014년 2월 : 한국 항공대학교 컴퓨터 공학 학사

2014년 3월~현재 : 서울대학교 컴퓨터 공학 석박사 통합과정

<관심분야> 인터넷 보안, 인증, 프라이버시

조 은 상 (Eunsang Cho)

2008년 2월 : 서울대학교 컴퓨터공학 학사

2008년 3월~현재 : 서울대학교 컴퓨터공학 석박사 통합과정

<관심분야> 네트워크 구조, 네트워크 보안

권 태 경 (Ted “Taekyoung” Kwon)

1993년 : 서울대학교 컴퓨터 공학 학사

1995년 : 서울대학교 컴퓨터 공학 석사

2000년 : 서울대학교 컴퓨터 공학 박사

2004년~현재 : 서울대학교 컴퓨터 공학부 교수

<관심분야> 미래인터넷, IoT, Web/SNS Analysis, 인터넷보안, 무선네트워크