논문 14-39B-06-01 The Journal of Korea Information and Communications Society '14-06 Vol.39B No.06
http://dx.doi.org/10.7840/kics.2014.39B.6.333

안전한 휴대 저장장치로서의 스마트카드를 활용한 (ID, 패스워드) 쌍들의 안전한 저장 및 검색 기법

박 준 철◆°

A Scheme for Secure Storage and Retrieval of (ID, Password) Pairs Using Smart Cards as Secure and Portable Storages

Jun-Cheol Park◆°

요 약

많은 인터넷 사용자들은 다수의 인증 정보를 기억해야 하는 불편함 때문에, 보안상의 취약점에도 불구하고 동일하거나 매우 유사한 패스워드 및 ID를 여러 사이트에서 재사용하려는 성향을 보인다. 본 논문에서는 안전한 휴대 저장장치로 스마트카드를 활용하여, 랜덤하게 생성한 다수의 (ID, 패스워드) 쌍들을 사용자가 기억할 필요 없이 안전하게 저장 및 검색, 갱신할 수 있는 기법을 제안한다. 제안 기법은 사이트의 (ID, 패스워드) 쌍 정보를 스마트카드의 메모리 및 별도의 원격 서버에 분할하여 보관하기 때문에, 스마트카드의 분실 또는 도난에도 안전하다. 또한 제안 기법에는 원격 서버의 정보를 삭제하는 기능이 포함되어 있어서, 스마트카드 분실 및 메모리의 해킹이 의심되는 상황에서도 공격자가 스마트카드 소유자의 어떠한 (ID, 패스워드) 쌍도 구해내지 못하도록 막을 수 있다. 제안 기법을 적용할 경우 사용자는 (ID, 패스워드) 정보를 얻기 위해 접근을 원하는 사이트 정보와 패스프레이즈를 스마트카드에 입력해야 하나, 이 정도의 추가 부담은 ID와 패스워드를 자유롭게 선택할 수 있음에도 그것들을 전혀 기억할 필요가 없다는 장점을 고려할 때 감내할 수준이내라고 판단한다.

Key Words : authentication, password, smart card, secret sharing, security

ABSTRACT

Despite the security weakness of reusing passwords, many Internet users are likely to use a single ID and password on various sites to avoid the inconvenience of remembering multiple credentials. This paper proposes a scheme for securely storing, retrieving, and updating randomly chosen (ID, password) pairs by using smart cards as secure and portable storages. The scheme makes a user free from remembering her (ID, password) pairs for Internet accesses. By splitting and scattering the (ID, password) pairs of a user across the user's smart card memory and a remote server's storage, it can protect the logon credentials even from the theft or loss of the smart card. Also, a user, if deemed necessary, can issue and let the server to delete all information belonging to the user. Hence even an attacker who cracked the smart card memory would not be able to obtain any (ID, password) pair of the victim thereafter. The scheme requires a user to input a site information and pass-phrase to her smart card to obtain the logon credentials, but it should be an acceptable overhead considering the benefits of not remembering the freely chosen (ID, password) pairs at all.

※ 본 논문은 2013학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.

◆ First Author and Corresponding Author : Department of Computer Engineering, Hongik University, jcpark@hongik.ac.kr, 종신회원

논문번호 : KICS2014-03-092, Received March 18, 2014; Revised May 26, 2014; Accepted May 26, 2014

Ⅰ. 서 론

전통적 패스워드 기반의 인증[1,2]은 사용자가 자신의 ID와 패스워드를 인증 서버에게 보내고, 서버는 수신한 값을 자신이 관리하는 패스워드 테이블의 정보와 비교하여 사용자 인증 여부를 결정하는 방식으로, 현재에도 사용자 인증을 위해 인터넷 사이트 및 서버들에서 가장 널리 사용되고 있다. 최근에는 기존 패스워드 기반 인증 방식의 보안 수준을 높이기 위해 휴대성과 보안성이 뛰어난 스마트카드를 인증에 활용하려는 많은 시도들[3-7]이 발표되었다. 하지만 이러한 연구 결과들은 모두 서버 및 클라이언트 측이 따라야 하는 새로운 프로토콜을 제시하고 있는데, 인증 프로토콜의 변경은 클라이언트 측과 서버 측 모두의 인증 절차와 인증 데이터베이스 등의 수정을 요구하기 때문에, 그 비용과 변경 적용 및 안정화에 필요한 시간 등을 고려했을 때 즉각적 적용은 쉽지 않다. 결국 ID 및 패스워드만으로 사용자 인증을 처리하는 다수의 인터넷 사이트 및 서버들에서는 보안성 향상을 위해 SSL/TLS[8]를 이용한 서버 인증 및 통신 내용 보호, 서버에서 패스워드의 해쉬 후 저장[9] 등, 기존 프로토콜 자체를 유지하는 범위에서 추가 적용이 가능한 보안 기법들을 주로 채택하고 있다.

새로운 인증 프로토콜을 제시하는 연구들과 달리, 본 논문에서는 기존의 ID 및 패스워드 기반 인증 방식 하에서 사용자가 다수의 (ID, 패스워드) 쌍들을 어떻게 간편하고 안전하게 관리하고, 필요 시 즉각 추출하거나 폐기시킬 수 있을지의 문제를 다룬다. 인터넷 사이트마다 서로 다른 ID를 쓰는 것은 사용자의 프라이버시 보장에 유리하다는 장점을 가지며, 서로 다른 패스워드를 쓰는 것은 가장 보안이 취약한 사이트의 패스워드가 유출되더라도 다른 사이트에 동일 패스워드로 인증이 되는 것을 방지할 수 있다는 측면에서 매우 중요하다. 하지만 사용자에게 서로 연관되어 있지 않은 다수의 ID나 패스워드를 기억하도록 강제하는 것은 매우 힘들다는 것이 잘 알려져 있다[10-12]. 그 결과 대부분의 사용자는 기억하기 쉽다는 이유로 동일한 ID 및 패스워드를 여러 사이트에서 재사용하거나, 아니면 서로 다른 ID나 패스워드를 쓰기는 하지만 이들을 종이에 써 두고 필요할 때마다 참조하게 된다[10]. 이 ID 및 패스워드 목록은 공격자에게 유출되었을 경우 해당 개인의 모든 가입된 사이트에 대한 접근을 가능하게 하므로 심각한 문제를 야기할 수 있다. 결국 사용자 입장에서는 사용의 편의성(기억하기 용이함)과 높은 보안성의 둘 중 하나를, 다른 것을 위해 일부 희생하는 선택을 하는 것이 일반적이다[10].

사용의 편의성을 높이면서도 안전한 패스워드를 쓸 수 있도록 하는 다양한 연구 결과가 제시되었다. Password Manager[13]라 통칭하는 몇몇 프로그램들이 발표되었는데, 이는 주로 브라우저의 플러그인 형태로 제공되면서 사용자가 다수의 패스워드들을 모두 기억할 필요 없이, 사이트별 패스워드를 로그인 시마다 계산해내는 역할을 한다. 이 때 도출된 패스워드는 해쉬 함수 등의 도움으로 계산된 값으로 높은 안전성을 가지게 된다. 다만 마스터 패스워드 또는 사이트별 개인 패스워드(실제 패스워드와는 다름)는 사용자가 기억하고 있어야 하며, 사이트별 사용자 ID 또한 각 개인이 기억하고 있어야 하는 부담이 있다. 또한 Password Manager에 입력하는 사용자의 비밀 값이 공격자에게 노출되는 경우, 해당 사용자의 패스워드들이 자신도 모르는 채 공격자에 손에 들어갈 수 있다.

한편 사용자가 패스워드를 좀 더 쉽게 기억하도록 돕는 다양한 방법들도 제시되었는데, 연상기호(mnemonic) 문구를 이용하여 외우기 쉬우면서도 안전한 패스워드를 생성하는 기법[12], 그래픽 패스워드(텍스트 대신 이미지들을 활용하여 사용자가 암기 대신에 연상하는 방식을 이용해 생성한 패스워드)를 사용하는 기법[14], 사용자가 선택한 패스워드에 임의로 선택한 문자들을 무작위로 포함시켜서 섞은 결과를 패스워드로 사용하게 하는 방법[15], 암기 대신 연상을 적용하여 기억하기 쉬운 텍스트 기반 패스워드를 생성하는 기법[16], 사용자가 패스워드 선택 시에 여러 질문들에 대해 각각 답을 하도록 하고 그 결과를 조합하여 패스워드를 생성하게 하는 기법[17] 등을 들 수 있다. 이들 기억을 돕는 방식들은 사용자의 패스워드 선정 과정에 개입하여 더욱 안전한 패스워드를 사용자가 쓰도록 하고자 하는 것이며, 패스워드의 안전한 보관 및 관리 영역을 다루는 본 논문의 연구 결과와는 해결하고자 하는 문제 영역에서 차이를 가진다.

본 논문에서는 간편한 휴대 저장장치로서 안전성이 높은 스마트카드를 활용하여, 개인 사용자의 서로 다른 (ID, 패스워드) 쌍들을 안전하게 관리하는 기법을 제시한다. 이를 위해 스마트카드의 저장 및 대칭키 암호화 연산 기능을 활용하며, 스마트카드의 리더는 스마트카드를 내장하고 있는 모바일 컴퓨터로서 키보드 및 디스플레이 입출력 장치를 가지고 있다고 가정한다. 이 기준을 만족하는 대표적인 기기로는 USIM 형태의 스마트카드를 내장한 스마트폰을 들 수 있는데, 스마트폰은 휴대성이 뛰어나고, 사용자가 그 조작에 익숙하며, 다양한 입출력 인터페이스를 가진다는 특징

이 있다. 제안 기법을 사용하면 사용자가 각 인터넷 사이트의 (ID, 패스워드) 쌍을 다른 사이트의 (ID, 패스워드) 쌍과는 전혀 관련이 없도록 임의로 선택할 수 있게 하면서도, 사이트별 (ID, 패스워드) 쌍을 전혀 기억할 필요가 없게 된다. 이를 위해 제안 방식은 스마트카드에 모든 정보를 저장하지 않고, 외부의 인증정보관리 서버(Credentials Management 서버, 이하 CM 서버라 칭하며 2장 이하에서 상세 설명함)에 (ID, 패스워드)의 복원에 필요한 정보를 분할하여 저장한다. CM 서버는 사용자 인증 후 데이터 테이블 검색 및 레코드 접근 서비스를 제공하는 단순한 기능을 하며, 자체 서버를 쓰지 않고 클라우드 스토리지 서비스를 활용하여 구축할 수도 있다.

본 논문의 구성은 다음과 같다. 2장에서는 제안 기법의 적용을 위한 구성 요소 및 구조를 설명하고, 3장에서는 CM 서버에 분할된 정보 등록과 이후의 (ID, 패스워드) 추출 및 갱신 프로토콜을 제시한다. 이후 4장에서는 스마트카드와 그 패스프레이즈의 유출 시, CM 서버의 정보 삭제를 통한 (ID, 패스워드) 유출 방지 프로토콜 및 스마트카드 패스프레이즈 분실 시의 갱신 프로토콜을 제안한다. 5장에서는 발생 가능한 공격 유형에 대하여, 제안 방식이 자유롭게 선택된 서로 다른 (ID, 패스워드) 쌍들의 기밀성을 어떻게 보장하는지 설명한다. 마지막으로 6장에서는 제안 방식의 의미와 적용 범위, 향후 연구 방향을 언급하면서 결론을 맺는다.

Ⅱ. 제안 기법의 구성 요소 및 구조

스마트카드 소유자는 어떤 사이트의 (ID, 패스워드) 쌍을 추출하기 위해 스마트카드에 저장된 비밀 값 및 CM 서버로부터 수신한 해당 사이트의 정보를 결합하는 과정을 거친다. 그 선행 작업으로, 우선 다수의 인터넷 사이트 각각에 대해 서로 다른 (ID, 패스워드) 쌍을 선택 후, 해당 정보를 분할 저장하는 것이 필요한데 3장에서 그 자세한 내용을 설명한다. 선행 작업이 완료되었다 가정할 때, 다음의 그림 1은 스마트카드 소유자인 Alice가 자신이 가입된 한 인터넷 사이트에 제시할 (ID, 패스워드) 쌍을 추출해내기 위해, CM 서버에 질의를 보내고 응답을 받아, 이를 통해 실제의 (ID, 패스워드)를 계산하는 과정의 데이터 흐름을 보여준다. 이 때 Alice의 스마트카드에는 암호화된 S가 저장되어 있는데, 이 S 값은 CM 서버에 제시되어 인증 및 Alice를 위한 테이블에 접근하는 인덱스 키의 역할을 한다. 우선 Alice는 기억하고 있는 패스프레이

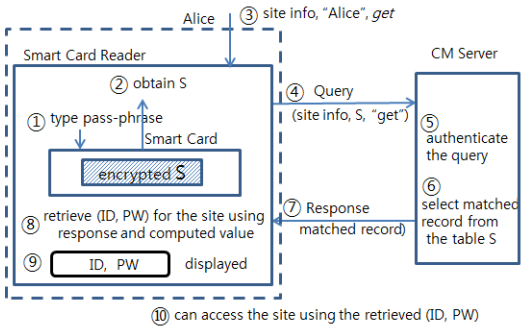

그림 1. 올바른 (ID, PW) 쌍을 추출하기 위한 구조 및 제어/데이터 흐름도
Fig. 1. Structure and Control/Data Flow Diagram for Retrieving the Appropriate (ID, PW) Pair

즈를 스마트카드에 입력하고, 스마트카드는 이 패스프레이즈로부터 복호화에 사용할 키를 생성하고 이를 사용하여 원래의 값 S를 추출한다. 그림 1의 과정을 포함하여, 앞으로 제시되는 사용자 Alice와 CM 서버 사이의 모든 통신은 SSL 프로토콜에 의해 보호된다고 가정한다. 또한 CM 서버는 제안 프로토콜을 규정된 대로 실행하지만, CM 서버 스스로가 공격자가 되어 저장하고 있는 개인별 데이터를 읽거나 이로부터 어떤 값을 유추하려는 시도를 할 수 있다고 가정한다.

앞으로, $E_K(M)$ 및 $D_K(M)$ 은 M을 키 K로 대칭키 암호화(예: AES) 및 복호화 시킴을 의미하고, $h()$는 암호화 해쉬 함수(예: SHA-256), $\oplus$는 비트단위의 XOR 연산, $\|$는 연접(concatenation) 연산, $|seq|_B$는 seq의 길이(단위: 바이트)를 의미하는 표기법으로 사용한다.

Ⅲ. CM 서버에 분할된 정보 등록과 (ID,패스워드) 추출 및 갱신 프로토콜

2장에서 설명한 데이터 흐름에 따라 Alice가 자신의 (ID, 패스워드) 쌍을 검색하려면, 우선 Alice의 스마트카드 메모리에 적절한 정보가 저장되고, CM 서버에도 사이트별 인증을 위한 분할 정보가 Alice에 의해 등록되어야 한다.

3.1 스마트카드 메모리 내장 정보 및 별도 저장이 요구되는 정보

앞으로의 설명에서, Alice의 스마트카드의 메모리에는 비밀 값 S가 암호화된 상태, 즉 $E_K(S)$,로 저장되어 있다고 가정한다. 여기서 암호화에 사용된 키는 $K=h(P)$이며, P는 Alice가 기억하고 있는 패스프

레이즈이다. 비밀 값 S는 사용자마다 임의로 선택한 랜덤한 값 S^*로부터 $h(S^*)=S$ 에 의해 계산된다. 이 비밀 값 S^*는 그림 1의 사이트별 (ID, 패스워드)를 추출하는 과정에서는 불필요하나, 스마트카드의 분실/도난이나 패스프레이즈를 잊었을 때, CM 서버에 정보 삭제 요청을 보내거나 S를 스마트카드에 재설정할 때 사용된다. 이 S^* 값은, 만약 유출된다면 S가 유출된 것이나 마찬가지 이므로 안전하게 저장되어야 하고, 필요 시 즉각 얻을 수 있어야 한다. 더불어 패스프레이즈의 재설정을 위해 $K=h(P)$ 역시 안전하게 보관할 필요가 있다. 사용자 Alice가 자신의 S^* 및 K값을 안전하게 보관하는 방법은 다음과 같다. (1) S^* 및 K 값을 Alice의 public key로 암호화 시킨 후, 가용성 및 무결성이 보장되는 장소에 저장하고, 필요 시 Alice의 private key를 이용하여 얻음, 또는 (2) 서로 독립적인 두 서버에 각각 $E_R(S^*,K)$, R을 저장하고, 필요 시 $E_R(S^*,K)$를 Alice의 이메일주소1로 보내고, R을 Alice의 이메일주소2로 보내도록 한 후, 이메일 수신 값들로부터 $D_R(E_R(S^*,K))=S^*,K$를 계산함.

3.2 CM 서버에 분할된 정보 초기 등록

Alice가 인터넷 사이트 X의 ID인 id_X와 패스워드(또는 PW)인 pw_X를 추출해내기 위해서, CM 서버에 어떤 값을 등록하는지 설명한다. Alice는 위에서 설명한 것처럼, 랜덤하게 S^* 를 선택하고, 이 S^*로부터 $h(S^*)=S$ 를 얻어, 이에 접근할 수 있다. Alice의 스마트카드는 S를 이용하여 $A_X=h(X\|S\|cnt)\|h(cnt\|S\|X)$를 계산한다. 이 때, cnt는 counter 값으로, 초기 레코드 등록 시에는 숫자 0이 사용된다. 해쉬 함수 $h()$는 그 출력 값의 길이가 ID 또는 PW의 가능한 최대 길이(바이트 수)보다 1 바이트만큼 큰 것으로 선택한다. 본 논문에서 ID 및 PW는 각각 최대 31 문자를 가질 수 있다고 가정하여, 해쉬 함수의 출력 값은 (31+1) 바이트, 즉, 256 비트가 되도록 해쉬 함수로는 SHA-256을 사용한다. 다음으로, ID 및 PW의 길이(바이트 수)를 각각 m, $n(m,n\le 31)$이라고 할 때, T_X를 다음과 같이 정의한다. $T_X=m\|id_X\|n\|pw_X\|pat$ (단, $|T_X|_B=|A_X|_B$, $m=|id_X|_B$, $n=|pw_X|_B$, $|m|_B=|n|_B=1$, $|pat|_B\ge 0$, 여기서, pat는 T_X와 A_X의 길이를 같게 하기 위해 도입된 임의의 패턴(비트 열)으로, 실제 ID와 PW의 추출 시에는 사용되지 않을 부분이므로 어떤 값이 들어가도 좋음). 이후, Alice의 스마트카드 리더가 CM 서버에 사이트 X를 위한 등록 정보로 다음 메시지를 전송한다.

$$put, S, h(X\|\text{“Alice”}), cnt, T_X\oplus A_X$$

여기서 S는 CM 서버에서 인증과 더불어 인덱스 키의 역할을 한다. 즉, CM 서버에 테이블 S가 등록되어 있는 경우에만 CM 서버는 이후의 프로토콜을 계속 진행한다. 이때 S에는 사용자 식별정보가 전혀 포함되지 않기 때문에 CM 서버는 이 스마트카드의 소유주에 대해 어떤 추정도 할 수 없다. put은 '저장'을 의미하는 명령어이다. cnt는 counter 값으로 이 요청이 첫 번째 등록(put)이므로 0을 사용한다. 추후 X의 PW (또는 ID)가 변경되면 변경된 PW(또는 ID)를 반영한 정보를 가지도록 레코드가 갱신되며, 이 때 cnt 는 1 만큼 증가한 값이 사용된다.

초기 등록 요청을 수신한 CM 서버는 테이블 S를 생성하고(아직 생성되지 않았다면), 이 테이블에 레코드 $\langle h(X\|\text{“Alice”}), cnt, T_X\oplus A_X\rangle$를 삽입한다. 이 테이블에서 원하는 레코드를 찾을 때 사용하는 기본 (primary) 키는 $h(X\|\text{“Alice”})$가 된다. 위와 같은 방식을 반복하여, Alice는 X 이외의 각 인터넷 사이트에 대해서도 임의로 선택한 ID 및 PW의 추출을 위한 정보를 CM 서버의 S 테이블에 저장한다.

3.3 (ID, 패스워드) 추출 및 갱신 프로토콜

Alice는 인터넷 사이트 X의 ID와 PW를 얻기 위해 스마트카드를 카드 리더에 넣고, 아래와 같은 추출 프로토콜을 실행한다.

(1) Alice → 스마트카드: P 입력
(2) 스마트카드: 키 $K=h(P)$를 생성하여, $D_K(E_K(S))=S$를 얻고 리더에 전달함; P,K,S를 모두 삭제함
(3) Alice → 스마트카드 리더: X,“Alice”, get
(4) 스마트카드 리더 → CM 서버: $get, S, h(X\|\text{“Alice”})$
(5) CM 서버:
  (i) 테이블 S 가 존재하는지 확인(인증 및 테이블 선택), 존재하지 않으면 중지
  (ii) 테이블 S 에서 $h(X\|\text{“Alice”})$를 기본 키로 하여 레코드 검색, 레코드 존재하지 않으면 중지; 검색된 레코드: $\langle h(X\|\text{“Alice”}), cnt, T_X\oplus A_X\rangle$

(6) CM 서버 → 스마트카드 리더: $cnt, T_X \oplus A_X$

(7) 스마트카드 리더:

(i) 수신한 cnt 값을 이용하여, $\overline{A_X} = h(X \| S \| cnt) \| h(cnt \| S \| X)$를 계산함

(ii) 수신한 $T_X \oplus A_X$를 이용하여, $\overline{T_X} = (T_X \oplus A_X) \oplus \overline{A_X}$를 계산함

(iii) $\overline{T_X} = \overline{m} \| \overline{ID} \| \overline{n} \| \overline{PW} \| \overline{pat}$의 $(|\overline{m}|_B + 1)$번째 바이트부터 $\overline{m}$ 바이트만큼 취해 id_X로, $(|\overline{m} \| \overline{ID} \| \overline{n}|_B + 1)$번째 바이트부터 $\overline{n}$ 바이트만큼을 취해 pw_X로 삼음

(iv) 획득한 (id_X, pw_X)를 디스플레이 함

(v) S, (id_X, pw_X), X, "Alice" 및 중간 계산 값 전부를 삭제함

Alice가 입력한 패스프레이즈 P가 올바른 것이었고, 스마트카드의 메모리에 저장된 $E_K(S)$ 및 CM 서버의 레코드 $\langle h(X \| \text{"Alice"}), cnt, T_X \oplus A_X \rangle$가 변경되지 않았다면, $A_X = \overline{A_X}$이 성립되므로 $T_X = \overline{T_X}$ 가 되어 결국 Alice는 정확한 ID 및 PW를 위 과정을 통해 추출할 수 있다.

이제 사이트 X의 PW(또는 ID, 또는 ID와 PW 모두)를 변경하고자 할 때 실행하는 갱신 프로토콜을 제시한다. 새로운 ID 및 PW를 각각 id_X^* 와 pw_X^* 라고 하고, $m^* = |id_X^*|_B$, $n^* = |pw_X^*|_B$라고 하자.

(1) Alice → 스마트카드: P 입력

(2) 스마트카드: 키 $K = h(P)$를 생성하여, $D_K(E_K(S)) = S$ 를 얻고 리더에 전달함; P, K, S를 모두 삭제함

(3) Alice → 스마트카드 리더: X, "Alice", $update$

(4) 스마트카드 리더 → CM 서버: $update, S, h(X \| \text{"Alice"})$

(5) CM 서버:

(i) 테이블 S가 존재하는지 확인(인증 및 테이블 선택), 존재하지 않으면 중지

(ii) 테이블 S에서 $h(X \| \text{"Alice"})$를 기본 키로 하여 레코드 검색, 레코드 존재하지 않으면 중지; 검색된 레코드: $\langle h(X \| \text{"Alice"}), cnt, T_X \oplus A_X \rangle$

(iii) $cnt \leftarrow cnt + 1$ // 갱신되는 레코드는 1이 // 증가된, 새로운 cnt 값 이용해 계산함

(6) CM 서버 → 스마트카드 리더: cnt // 갱신된 cnt 값이 전달됨

(7) 스마트카드 리더:

(i) 수신한 cnt 값을 이용하여, $\overline{A_X} = h(X \| S \| cnt) \| h(cnt \| S \| X)$를 계산함

(ii) id_X^*와 pw_X^*를 이용하여 $T_X^* = m^* \| id_X^* \| n^* \| pw_X^* \| pat$을 계산함

(8) 스마트카드 리더 → CM 서버: $put, S, h(X \| \text{"Alice"}), cnt, T_X^* \oplus \overline{A_X}$

(9) 스마트카드 리더: S, (id_X^*, pw_X^*), X, "Alice" 및 중간 계산 값 전부를 삭제함

(10) CM 서버: 테이블 S에서 기본 키 $h(X \| \text{"Alice"})$의 레코드를 $\langle h(X \| \text{"Alice"}), cnt, T_X^* \oplus \overline{A_X} \rangle$로 교체함

위의 프로토콜들을 사용하면, Alice는 사이트별로 임의의 ID와 보안성이 높은 PW를 마음대로 선택(예: ID: xy72abz8q , PW: 4!A3&*faBy2)하면서도 이를 기억할 필요가 없고, 원하면 언제든지 ID 및 PW를 변경할 수 있다. Alice가 기억해야 할 것은 스마트카드에 입력할 패스프레이즈 뿐이며, 패스프레이즈로는 자신이 기억하기는 용이하지만 타인이 쉽게 추측하기 힘든 것(예: let-2T-go:fr.oZEN])을 선택하면 된다.

Ⅳ. (ID, 패스워드) 폐기 및 패스프레이즈 갱신 프로토콜

스마트카드는 간섭-저항(tamper-resistant) 성질을 가지고 있어, 분실이나 도난에 의해 공격자에 손에 들어가더라도 그 저장하고 있는 비밀 값이 쉽게 노출되지 않는다. 하지만 충분한 시간과 장비, 지식이 있는 공격자라면 공격에 성공할 수도 있기 때문에, 제안 방식에서는 스마트카드 메모리의 $E_K(S)$ 및 키 값 K 또는 K를 유도할 수 있는 패스프레이즈 P가 노출된 상황에서 (ID, 패스워드) 쌍들을 보호하기 위한 방법 역시 제공한다.

어떤 이유에서든 자신의 스마트카드 관련 정보가 유출되었다고 판단하면, Alice는 즉각 3장에서 설명된 방법에 따라 안전하게 저장되어 있던 S^* 값을 복원해

낸다. 이후, 아래와 같은 (ID, 패스워드) 폐기 프로토콜을 실행한다.

(1) Alice의 컴퓨터 → CM 서버: $revoke, S^*$
// $h(S^*) = S$
(2) CM 서버:
(i) 테이블 $h(S^*)$가 존재하는지 확인(인증 및 테이블 확인), 존재하지 않으면 중지
(ii) 테이블 $h(S^*)$의 모든 레코드를 삭제

이 폐기 프로토콜의 실행으로, 공격자는 더 이상 CM 서버에서 분할된 정보를 얻을 수 없기 때문에 Alice의 어떤 (ID, 패스워드) 쌍도 얻어낼 수 없다.

한편, Alice는 패스프레이즈 P를 잊었거나 P가 누군가에게 유출되었을 가능성이 있을 때, 즉각 안전하게 저장되어 있던 S^* 값 및 K 값을 복원한 후, 스마트카드 리더를 통해 아래의 패스프레이즈 갱신 프로토콜을 실행한다.

(1) 스마트카드 리더 → 스마트카드:
$replace, S^*, K, P^*$
// P^*가 새로운 패스프레이즈임
(2) 스마트카드:
(i) $h(S^*)$와 $D_K(E_K(S))$를 각각 계산한 후 두 값이 같은지 확인, 다르면 중지 (단, $E_K(S)$는 스마트카드에 저장되어 있던 값)
(ii) $E_K(S)$는 삭제, $E_{h(P^*)}(h(S^*))$를 저장함

V. 보안성 분석

제안 기법은 사용자의 (ID, 패스워드) 쌍을 스마트카드의 메모리 및 CM 서버에 분할 저장하는 방식을 취하고 있어, 어느 한 쪽에 대한 공격 및 정보 유출을 통해서는 (ID, 패스워드) 정보가 유출되지 않는다. 가능한 공격 방식에 대해 제안 기법이 어떻게 대응하고 있는지를 설명한다. 단, 스마트카드 소유자와 CM 서버 사이의 통신은 SSL로 보호되므로, 통신 내용의 기밀성과 무결성이 보장되며, 통신 내용의 재생 공격 등은 가능하지 않다고 간주한다.

5.1 스마트카드의 분실/도난과 그 저장된 비밀 값 유출

스마트카드를 손에 넣은 공격자가 적절한 장비와 노하우를 가지고 충분한 시간을 투자한다면, 스마트카드 역시 절대로 안전하다고 할 수는 없다. 어떤 공격자가 Alice의 인터넷 사이트 X에 대한 (ID, 패스워드)를 알고 싶다고 하자. 이 공격자가 Alice의 스마트카드를 손에 넣고, 그 패스프레이즈 P 또는 $K = h(P)$를 알아냈으며, 메모리의 $E_K(S)$에의 접근까지 성공한 최악의 시나리오가 발생한다면, 공격자는 $S = D_K(E_K(S))$를 CM 서버에 제시함으로써 인터넷 사이트 X에 대한 Alice의 (ID, 패스워드)를 구할 수 있다. 이러한 공격이 실제 발생하더라도, 제안 기법은 스마트카드의 분실/도난을 인지한 즉시, 소유자 Alice가 CM 서버에 S^* ($h(S^*) = S$)를 제시함으로써 자신의 정보를 폐기시킬 수 있다. CM 서버에 저장된 분할 정보 없이는 Alice의 어떤 인터넷 사이트의 (ID, 패스워드) 쌍도 추출할 수 없다. 또한 공격자가 CM 서버에 Alice 관련 정보의 삭제 요청을 보내는 것 역시 불가능하다. 왜냐하면 해쉬 함수의 일방향성에 의해 S로부터 $h(S^*) = S$인 S^*를 알아내는 것은 출력 크기가 충분히 큰 SHA-256을 쓰는 경우 실질적으로 불가능하기 때문이다. 그러므로 공격자가 S 값을 알아내더라도, Alice가 이후 자신의 (ID, 패스워드) 쌍을 추출하고자 하는 것까지 아예 막는, 일종의 서비스 거부 공격을 시도할 가능성은 무시할 수 있을 정도로 낮다. 또한 공격자가 스마트카드의 패스프레이즈를 임의의 값으로 변경하려는 시도 역시, K, S^*에 대한 지식이 필요하기 때문에 그 성공 가능성은 무시할 정도로 낮다.

5.2 CM 서버의 분할 정보 유출 또는 CM 서버에 의한 (ID, 패스워드) 추측 공격

CM 서버의 Alice의 인증을 위한 분할 정보, 예로 레코드 $\langle h(X \| \text{“Alice”}), cnt, T_X \oplus A_X \rangle$,에 공격자가 접근할 수 있었다고 가정하자. 공격자는 추가의 정보가 없는 한, $h(X \| \text{“Alice”})$로부터 이 레코드의 소유자가 누구인지, 이 레코드가 어떤 인터넷 사이트에 대한 정보를 포함하고 있는지 알아내기 힘들다. 또한 공격자는 $T_X \oplus A_X$로부터 T_X, 즉 실제의 (ID, 패스워드) 쌍을 포함하는 값,을 얻을 수 없다. 왜냐하면 A_X는 Alice의 스마트카드에 보관된 S 값이 없이는 구할 수 없으며, CM 서버의 어떤 저장된 값으로부터도 S를 유추할 수 없기 때문이다.

한편 CM 서버는 스스로 공격자의 역할을 하여 Alice의 (ID, 패스워드) 쌍들을 알아내려고 할 수 있다. 이런 공격에 대해서 제안 기법은 CM 서버에 제공하는 값에 (ID, 패스워드) 쌍을 유추하는데 필요한 어

떠한 비밀 정보도 포함시키지 않는 것으로 대응하고 있다. 우선 CM 서버에 저장되는 테이블의 이름 S는 임의의 랜덤한 값(해쉬 함수의 결과)으로서 그 소유주 Alice를 유추할 수 있는 어떤 값도 포함되지 않는다. 또한 테이블 S의 각 레코드에서는 기본 키로 X가 아닌 $h(X\|\text{“Alice”})$를 사용함으로써, CM 서버가 이 레코드가 어떤 인터넷 사이트에 대한 것인지를 알지 못하도록 한다. 또한 레코드의 cnt는 counter 값으로 초기 값 0에서 시작하며, 추후 X의 PW(또는 ID)가 변경되면 이 cnt는 1씩 증가한다. 만약 A_X에서 cnt가 포함되지 않는다면, PW가 $pw \rightarrow \widehat{pw}$ 식으로 변경되었을 때, CM 서버는 $(T_X \oplus A_X) \oplus (\widehat{T_X} \oplus A_X) = T_X \oplus \widehat{T_X}$를 계산하여 $pw \oplus \widehat{pw}$를 구할 수 있다. 만약 다른 경로로 pw를 알게 되었다면, CM 서버는 새로운 PW인 $\widehat{pw}$를 얻을 수 있게 된다. 물론 CM 서버는 여전히 이 PW가 어떤 사용자의, 어떤 사이트에 대한 것인지 알아야 하지만, 제안 기법은 cnt의 도입을 통해 $pw \oplus \widehat{pw}$를 구할 가능성 자체를 제거한다.

5.3 CM 서버에 대한 서비스 거부 공격

Alice가 어떤 사이트에 대한 자신의 ID 및 패스워드를 얻고자 할 때 반드시 CM 서버에 접근해야 한다. 따라서 CM 서버에 대한 서비스 거부 공격이 가해져 CM 서버의 가용성에 문제가 생길 것에 대비하여 둘 이상의 CM 서버에 동일한 정보를 중복 저장케 함으로써 모든 CM 서버가 동시에 문제가 발생하는 극히 예외적인 경우를 제외하고는 항상 인증 서비스 제공이 가능하도록 제안 방식을 확장할 수 있다. CM 서버의 레코드 하나의 크기는 cnt의 크기를 16 비트(ID/PW 변경이 65,535번 가능)라 할 때, (256+16+512)=784비트= $98B$에 해당한다. 사용자당 최대 20개의 인터넷 사이트에 가입되어 있다면, CM 서버의 사용자별 테이블의 크기는 $98B \times 20 \fallingdotseq 1.914KB$에 불과하다. 따라서 최근의 저장장치 기술의 발전에 따른 용량 확장 및 가격 하락을 고려할 때, 인증 정보를 중복하여 저장, 관리하는 것이 사용자 입장에서 큰 부담이 되지 않으리라 판단된다. 만약 클라우드 스토리지 서비스를 이용해 CM 서버들을 구축한다면 데이터의 분할 및 중복 저장 기법[18]을 활용하여 가용성을 높이면서도 중복 저장되는 데이터의 양을 줄일 수 있다.

다만 둘 이상의 CM 서버를 운용할 때 서버에 저장된 내용을 수정해야 하는 경우는 중복된 CM 서버 모두에 갱신 절차를 진행함으로써 저장된 데이터의 일관성을 유지할 필요가 있다. 그런데 이런 갱신 작업은 ID/PW의 변경, 특정 사용자 테이블의 삭제 등과 같이 비정기적인 사건이므로, 사용자가 자신의 인증 정보를 획득하려고 임의의 CM 서버에 접근하는 통상의 작업에는 전혀 성능 측면의 악영향을 주지 않는다. 사용자는 단일 CM 서버의 경우와 마찬가지로, 중복된 CM 서버 중 임의의 하나를 선택하여 자신의 ID 및 패스워드 복원에 필요한 정보를 획득하는 절차를 진행한다.

Ⅵ. 결 론

본 논문은 패스워드 기반의 인증 방식에서 사용자 편의성과 보안성을 동시에 제고하는 새로운 접근 방식으로, 인증 정보의 분할 저장을 통해 (ID, 패스워드) 쌍의 추출, 갱신, 폐기의 안전한 실행을 보장하는 기법을 제시하였다.

제안 기법에서는 (ID, 패스워드) 쌍의 분할 저장을 통하여, 소유자의 스마트카드 메모리의 비밀 값 및 외부 CM 서버에 저장된 인증 관련 분할된 정보 양 쪽을 모두 이용하여 (ID, 패스워드) 쌍을 추출한다. 이 추출 과정전체에서 스마트카드 소유자는 패스프레이즈만을 기억하면 된다. 제안 기법은 스마트카드의 분실 또는 도난 후 그 메모리에 대한 공격 가능성에 대비하여, 스마트카드 소유자가 스마트카드 없이도 CM 서버에 저장된 자신의 모든 인증 정보를 폐기 요청하는 방식을 포함한다. 제안 기법을 적용하면 CM 서버의 어떤 사용자 관련 인증 정보가 공격자에 유출되거나 또는 CM 서버가 공격자의 역할을 하는 경우에도, 해당 사용자 스마트카드 메모리의 암호화된 내용을 동시에 획득하지 않는 한, (ID, 패스워드) 정보는 물론 해당 사용자가 누구인지, 어떤 인터넷 사이트에 계정을 가지고 있는지도 알아낼 수 없다. 아울러, 제안 기법은 ID나 패스워드의 변경과 패스프레이즈의 재설정이 종종 필요함을 고려하여 이러한 변경 역시 안전하게 실행되도록 설계되었다. 제안 기법사용 시, 사용자는 (ID, 패스워드) 정보를 얻기 위해 인터넷 사이트 정보와 패스프레이즈를 스마트카드에 입력해야 하나, 이 정도의 추가 부담은 다수의, 임의로 선택할 수 있는 (ID, 패스워드) 쌍들을 전혀 기억할 필요가 없다는 장점을 고려할 때 감내할 수준 이내라고 판단한다.

향후, 스마트카드를 종단으로 하면서 SSL에 의존하지 않고 제안 프로토콜을 통하여 통신 과정이 보호되도록 하는 것과 보안성을 저해하지 않는 범위에서 사용자의 사용 편의성이 향상되도록 제안 기법을 개선하는 부분에 대한 추가 연구가 필요하리라 판단한다.

References

[1] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no 11, pp. 770-772, 1981.

[2] N. Haller, The S/KEY one-time password system, *RFC 1760*, Feb. 1995.

[3] X. Li, J. Niu, M.K. Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *J. Netw. Comput. Appl.*, vol. 36, pp. 1365-1371, 2013.

[4] M. Kim, "Security analysis and enhancement of Tsai et al.'s smart-card based authentication scheme," *J. KICS*, vol. 39B, no. 1, pp. 29-37, Jan. 2014.

[5] J. Qiuyan, K. Lee, and D. Won, "Cryptanalysis of a secure remote user authentication scheme," *J. KICS*, vol. 37C, no. 8, Aug. 2012.

[6] R. Madhusudhan and R. C. Mittal, "Dynamic ID-based remote user password authentication schemes using smart cards: a review," *J. Netw. Comput. Appl.*, vol. 35, pp. 1235-1248, 2012.

[7] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, pp. 321-325, 2010.

[8] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," *RFC 5246*, Aug. 2008.

[9] M. Stamp, *Information Security: Principles and Practice*, 2nd Ed., pp. 229-254, NY: John Wiley & Sons, 2011.

[10] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memorability and security: empirical results," *IEEE Security and Privacy*, vol. 2, no. 5, pp. 25-31, Sept. 2004.

[11] S. Chiasson, A. Forget, E. Stobert, P.C. van Oorschot, and R. Biddle, "Multiple password interference in text passwords and click-based graphical passwords," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2009.

[12] C. Kuo, S. Romanosky, and L.F. Cranor, "Human selection of mnemonic phrase-based passwords," in *Proc. Symp. Usable Privacy and Security (SOUPS)*, 2006.

[13] S. Chiasson and P. C. van Oorschot, and R. Biddle "A usability study and critique of two password managers," in *Proc. Conf. USENIX Security Symp.(USENIX-SS)*, vol. 15, 2006.

[14] R. Biddle, S. Chiasson, and P.C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Computing Surveys*, vol. 44, no. 4, pp. 19:1-19:44, Sept. 2012.

[15] A. Forget, S. Chiasson, P.C. van Oorschot, and R. Biddle, "Improving text passwords through persuasion," in *Proc. Symp. Usable Privacy and Security (SOUPS)*, pp. 1-12, Jul. 2008.

[16] N. Wright, A.S. Patrick, and R. Biddle, "Do you see your password? applying recognition to textual passwords," in *Proc. Symp. Usable Privacy and Security (SOUPS)*, Jul. 2012.

[17] S. Maqsood, Text password authentication using cued text passwords, Honours Project, School of Computer Science, Carleton University, Dec. 2013.

[18] J. C. Park, "Improving data availability by data partitioning and partial overlapping on multiple cloud storages," *J. KICS*, vol. 36B, no. 12, pp. 1498-1508, Dec. 2011.

박 준 철 (Jun-Cheol Park)

1986년 2월 : 서울대학교 계산통계학과 졸업

1988년 2월 : KAIST 전산학과 석사

1998년 12월 : U. of Maryland, College Park, 전산학 박사

2001년 9월~현재 : 홍익대학교 컴퓨터공학과 교수

<관심분야> 소프트웨어 보안, 네트워크 보안