기밀성을 제공하는 상호 인증 일회용 패스워드 메커니즘 설계

박희운, 이임영
순천향대학교 정보기술공학부
e-mail:heeun@ai-cse.sch.ac.kr

A Design of Mutual Authentication One Time Password Mechanism for Confidentiality

Hee-Un Park, Im-Yeong Lee
Division of Information Technology Eng,
Soonchunhyang University

요약

컴퓨터 및 네트워크의 급속한 발전은 우리의 생활 전반에 다양성과 편리성을 제공하고 있다. 그러나 이러한 발전 속에서 자신의 정보를 보호하고 정당한 사용자의 접근을 보장하기 위해 접근 통제가 필수적으로 요구되고 있다. 그러나 단순한 접근 통제는 상호 인증 및 기밀성 제공에 있어 한계가 있다.

본 고에서는 기존 방식들의 특징 및 문제점들을 고찰하고, 이들이 고려하지 안았던 상호 인증 및 기밀성을 제공하는 새로운 일회용 패스워드 방식을 제안한다.

1. 서론

정보화 사회의 급속한 발전을 통해 새로운 인프라가 구축되고 있다. 특히 컴퓨터의 발전과 네트워크의 확장은 우리의 일반 환경에 있어 많은 변화를 가져왔다. 인터넷을 통해 일반 사용자들도 필요한 정보를 쉽게 얻고 있으며, 전자적인 문서의 교류는 여러 회사간에 문서 결재를 가능하게 하였다. 그 외에도 이들의 응용 범위는 매우 광범위하다.

이러한 네트워크 및 컴퓨터의 발전 속에서 우선적으로 고려해야 할 사항은 시스템 접속 요구자의 정당성을 확인하는 인증 과정이 필요하며, 동시에 안전하게 메시지를 전송하기 위한 암호 키 분배가 필요하게 된다.

사용자 인증을 위한 접근 통제 방법으로서 가장 대중화 된 방법은 패스워드 방식이다. 그러나, 이 방식은 패스워드가 암호화 과정 없이 네트워크를 통해 전송되므로 도청과 같은 불법행위에 대해 매우 취약하다. 뿐만 아니라 한번 정해진 패스워드를 계속적으로 사용하므로, 사용자 또는 서버에 등록되어 있는 패스워드가 도난 당할 경우 새로이 키를 생성해야하는 문제점이 발생한다.

이러한 문제점을 해결하기 위해서 최근에는 일회용 패스워드 방식이 사용되고 있다. 이 방식은 사용자 인증시 사용되는 패스워드를 해쉬를 이용하여 한번만 사용하게 하고 있다. 하지만 이 방식은 패스워드 사용의 한계가 있고, 안전하게 메시지를 전송하기 위해서는 별도의 암호화 과정을 통한 키 분배가 필요하게 된다.

본 고에서는 기존의 접근 통제 기법들 중 몇몇을 살펴보고, 이에 대한 문제점을 지적할 것이다. 또한 일회용 패스워드의 문제점으로 지적된 사항들을 극

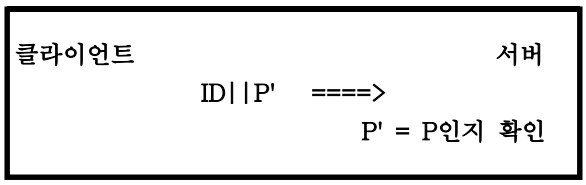
본 연구는 한국과학재단 특정기초연구과제(과제 번호:97-01-00-06-01-3)연구비 지원에 의해 수행되었음.

복하면서 동시에 키 분배가 가능한 방식을 제안하고자 한다.

2. 기존의 방식들 고찰

2.1 패스워드 인증 방식

본 방식은 서버 - 클라이언트 모델에서 일반적으로 가장 많이 사용하고 있는 방식으로 접근 통제를 위해 사용자 자신의 ID에 대해 패스워드 P'을 입력함으로서 인증을 받게된다.



(그림 1) 일반적인 패스워드 인증 방식

그러나, 이 방식은 다음과 같은 사항들을 통해 문제점을 발생시키고 있다 [1].

. 패스워드 전송 노출 : 사용자가 전송하는 ID와 패스워드 P'은 어떠한 안전 조치 없이 전송되므로 도청과 같은 불법 행위로부터 P'를 쉽게 얻을 수 있다. 이를 통해 P'의 노출이 가능하다.

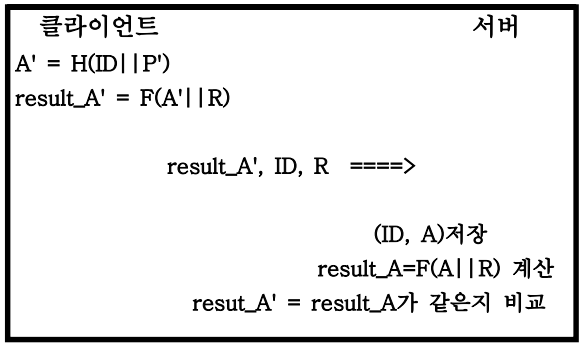
. 패스워드 재전송 : 현재 사용하는 패스워드 P'은 다음 인증시에도 그대로 사용되므로 전송시 패스워드 정보를 복사한 경우, 쉽게 위장이 가능하다.

. 서버 인증 정보 공격 : 서버가 보관하고 있는 각 사용자의 인증 정보가 노출될 경우 쉽게 패스워드를 획득할 수 있고, 전송되는 사용자 인증 정보를 가로채어 서버로 가장할 수 있는 문제점이 발생한다.

. 별도의 키 분배 방식 필요 : 만약 사용자와 비밀 통신을 수행하려 할 경우 별도의 키를 생성해야 하는 번거로움이 발생한다.

2.2 변형 패스워드 인증 방식

이 방식은 패스워드 방식의 문제점을 해결하기 위하여 2개의 일방향 해쉬 함수 H, F를 도입했다. 즉 사용자의 인증 정보 (ID, P')을 해쉬 함수 H를 이용하여 전송함으로서, 제 3자에 의한 도청을 방지함으로서 전송 노출을 막고 있다. 또한 패스워드 재전송을 방지하기 위해 랜덤 값 R을 사용하고, 서버 인증 정보 공격을 막기위해 해쉬된 정보를 그대로 저장함으로서 안전성을 획득하고 있다.



(그림 2) 변형 패스워드 인증 방식

그러나, 이 방식은 각 ID에 대해 하나의 패스워드를 소유하고 있다. 또한 해쉬 함수 H와 F가 공개되어 있다고 가정한다. 따라서 사용자 인증 정보 전송시 및 서버 인증시 동시적인 공격자의 공격이 가능할 경우, 공격자는 사용자의 ID와 랜덤값 R을 알게되고 저장중인 A를 알게 되므로 result_A를 계산 할 수 있다. 이를 통해 사용자 위장이 가능하게 된다는 문제점을 안고 있다.

2.3 일회용 패스워드 인증 방식

이 방식은 상기 방식과 비슷하게 일방향 해쉬 함수 H를 사용한다[3][4]. 그러나, 패스워드 생성에 있어 seed 값을 생성하고 이를 기초로 매 접속시 새로운 패스워드로 인증 받는다는 것이 다른 점이다. 따라서 매회 새로운 패스워드로 인증이 수행되기 때문에 상기 방식의 문제점을 극복하는 방식이라 하겠다. 다음은 RFC 1760 표준을 기초로 일회용 패스워드 방식을 기술한다.

2.3.1 시스템 계수

다음은 이 방식에서 사용되는 시스템 계수를 서술한 것이다.

. seed : 사용자 및 서버에서 패스워드 생성시 사용하는 초기 공유 값

. H : 안전한 일방향 해쉬 함수(RFC 1760에서는 MD4를 사용)

. X_n : seed 값을 n번 해쉬한 값

. ID_A : 사용자 A의 식별 ID

2.3.2 프로토콜

(1) 초기화 단계

: 사용자 A는 자신의 ID_A를 생성해 log-in 한다.

: 서버는 seed 값을 생성한 다음, 해쉬할 횟수 n을 선택하여 안전하게 사용자 A에게 전송한다.

(2) 패스워드 생성 단계

: 사용자는 seed 값과 password 값을 연접해 해쉬 함수 H를 이용하여 n회 해쉬한 다음, X_1에서 X_n까지 그 결과값을 저장한다.

: 사용자 A는 password를 서버에게 전송한다. 서버는 이를 수신한 다음 사용자와 마찬가지로 H를 이용하여 X_1'에서 X_n'까지 그 결과값을 저장한다.

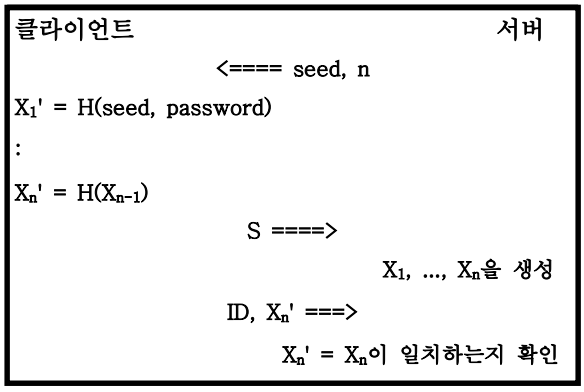
(3) 인증 과정

: 사용자는 서버에게 접속하기 위하여 ID_A, X_n'을 서버에게 전송한다.

: 서버는 저장되어 있는 X_n과 X_n'을 비교하여 일치하면 인증이 된 것이다.

: 사용자가 차후 인증을 위해서는 ID_A, X_{n-1}을 전송해 인증을 받게된다.

: 이렇게 하여 사용자 인증을 위해서는 한번에 단 하나의 패스워드를 사용하게 된다.



(그림 3) 일방향 패스워드 인증 방식

2.3.3 일회용 패스워드 방식 분석

일회용 패스워드 방식은 다음과 같은 특징을 갖고 있다[5].

. 이 방식은 구조가 단순하기 때문에 인증 시스템에 쉽게 적용 가능한 방식이다.

. 패스워드가 매 인증시 새로이 사용되므로, 상기 변형 패스워드 방식의 문제점을 해결하고 있다.

. 그러나 이 방식은 최초 생성했던 n에 의해 사용 횟수가 제한되고, 안전성이 해쉬 알고리즘에 의존한다는 문제점을 안고 있다. 따라서 사용자는 인증에 사용되는 일회용 패스워드를 미리 생성해야 하는 부담을 가지고 있다.

. 또한 별도의 기밀성 서비스를 제공하기 위해서는 키 분배 과정이 따로 필요하다는 문제점이 있다.

3. 기밀성 제공 상호 인증 일회용 패스워드 방식 제안

본 장에서는 새로운 일회용 패스워드 상호 인증 키 분배 메커니즘을 제안한다. 본 방식은 일회용 패스워드 방식을 통해 사용자와 서버 사이에 상호 인증이 가능하며, 이를 기초로 매 통신시 사용되는 비밀키 생성이 가능하므로 매우 유용한 메커니즘이라 할 수 있다. 또한 인증을 위한 패스워드 생성의 사용 횟수 제약이 없다는 특징을 가지고 있다.

3.1 설정 환경

본 방식은 일회용 패스워드 방식을 이용해 상호 인증을 수행하고, 또한 기밀성 제공을 위해 비밀키 암호 방식을 적용한다. 이를 위해 다음과 같은 환경을 설정한다.

. 비밀키 암호 방식을 이용한 키 분배를 위해 2-pass 방식을 적용하며, 상호 인증성을 제공하기 위해 사용자 및 서버 양자가 키 구성 정보를 생성한다.

. 접근 인증 패스워드는 필요한 만큼 사용한다.

. 인증 및 키 분배 방식을 통합함으로서 일회용 패스워드 방식의 문제점을 개선한다.

. 서버는 어떠한 특정 비밀 정보도 자신의 DB에 보관하지 않는다.

3.2 프로토콜 제안

3.2.1 시스템 계수

. seed : 일회용 패스워드 설정 초기 값
. EK_{AS}: 대칭키
. T_A, T_S : Time Stamp
. N_A, N_S : 랜덤 수
. H : 안전한 일발향 해쉬 함수
. F_A, F_S : 키 구성 및 일회용 인증 요소
. ID_A : 요구자 A의 ID
. X_n : n 번째 일회용 패스워드

3.2.2 프로토콜

(1) 초기화 단계

: 서버는 스마트 카드와 같은 안전한 채널을 이용해 seed 값을 생성한 다음, 사용자 A에게 전송한다.

(2) 패스워드 생성 단계

: 사용자 A는 seed 값과 자신의 ID를 연접해 해쉬 함수 H를 이용하여 해쉬를 수행하여 X_0을 만든다[6]. 서버 역시 동일한 방법으로 X_0를 만든다.

(3) 사용자의 인증 및 키 구성 정보 생성 단계

: 사용자 A는 인증을 위해 다음과 같이 X_{n+1}을 생성한다.

$$X_{n+1} = H(ID_A||X_n||T_A)$$

: 사용자 A는 키 분배 요소 F_A를 생성한 다음 F_A를 통해 인증 정보를 암호화하여 자신의 ID와 Time Stamp 및 N_A를 서버에게 전송한다.

$$F_A = E_{Xn}(N_A)$$
$$A_M = E_{FA}(X_{n+1})$$

(4) 서버의 인증 수행 및 키 생성 단계

: 서버는 수신된 정보를 기초로 다음을 생성한다.

$$X_{n+1} = H(ID_A||X_n||T_A)$$
$$F_A = E_{Xn}(N_A)$$
$$A_M' = E_{FA}(X_{n+1})$$

: 서버는 수신된 A_M과 A_M'이 동일한지 확인한다. 동일하다면 정당한 사용자로서 A는 인증받게 된다.

: 서버는 다음과 같은 키 생성 정보를 생성하여 세션키 K_{AS}를 다음과 같이 생성한다.

$$S_e = ID_A \oplus X_n$$
$$F_S = E_{Xn}(S_e||N_B)$$
$$K_{AS} = H(F_A||F_S)$$

: 서버는 생성된 키 K_{AS}와 A의 인증 정보 X_{n+1}를 연접한 다음, 해쉬를 취해 N_S와 함께 사용자 A에게 전송한다.

$$S_M = E_{FS}(K_{AS}||X_{n+1})$$

(5) 사용자의 서버 인증 및 키 생성

: 사용자 A는 수신된 정보를 통해 다음을 생성한다.

$$S_e = ID_A \oplus X_n$$
$$F_S = E_{Xn}(S_e||N_S)$$

: 다음 값을 생성하여 각각이 S_M및 K_{AS}과 일치한다면 서버는 인증되고 사용자 A는 정확한 키를 얻게된다.

$$S_M' = E_{FS}(K_{AS}||X_{n+1})$$
$$K_{AS}' = H(F_A||F_S)$$

사용자A		서버
$X_{n+1} = H(ID_A\|\|X_n\|\|T_A)$		
$F_A = E_{Xn}(N_A)$	ID_A, T_A, N_A, A_M====>	
$A_M=E_{FA}(X_{n+1})$		
		$X_{n+1} = H(ID_A\|\|X_n\|\|T_A)$
		$F_A = E_{Xn}(N_A)$
		$A_M' = E_{FA}(X_{n+1}) \Leftrightarrow A_M$ 확인
<====N_S, S		$S_e = ID_A \oplus X_n$
		$F_S = E_{Xn}(S_e\|\|N_S)$
		$K_{AS} = H(F_A\|\|F_S)$
		$S_M = E_{FS}(K_{AS}\|\|X_{n+1})$
$S_e = ID_A \oplus X_n$		
$F_S = E_{Xn}(S_e\|\|N_S)$		
$S_M' = E_{FS}(K_{AS}\|\|X_{n+1}) \Leftrightarrow S_M$ 확인		
$K_{AS}' = H(F_A\|\|F_S) \Leftrightarrow K_{AS}$ 확인		

(그림 4) 제안 방식 흐름도

3.2.3 제안 방식 분석

제안 방식은 일회용 패스워드 방식을 이용하여 상호 인증과 기밀성 제공을 위한 키 분배를 동시에 가능하게 하는 방식이다. 또한 안전성이 암호 알고리즘에 의존하므로 일회용 패스워드의 문제점을 완전

하게 개선하고 있다. 다음은 기존의 방식과 제안 방식을 비교 분석한 결과이다.

<표 1> 각방식별 비교 분석

항목 \ 방식	패스워드 방식	변형 패스워드 방식	일회용 패스워드 방식	제안 방식
사용 횟수	제한 없음	제한 없음	n회	제한 없음
패스워드 노출 방어	X	O	O	O
패스워드 재전송 방어	X	O	O	O
서버 인증 정보 공격 방어	X	O	O	O
위장 공격 방어	X	X	O	O
안전성	X	해쉬 함수에 근거	해쉬 함수에 근거	암호알고리즘에 근거
기밀성 제공 유무	X	X	X	O
상호 인증 유무	X	X	X	O

3.결론

정보화 사회의 발전을 통해 개인 및 단체의 고유 정보를 보호하고, 허가된 사용자에게만 접근을 허락하는 접근 통제 방식은 없어서는 안될 매우 중요한 요소가 되었다. 동시에 이러한 인증 방식과 더불어 안전한 정보 전송을 위한 키 분배 방식 역시 고려되어야만 할 사항이다.

본 고에서는 현재 널리 사용되고 있는 기존의 몇몇 방식들을 살펴보았다. 일반 패스워드 방식은 패스워드 보호 측면에서 매우 취약했으며, 변형 패스워드 방식은 위장 공격이 가능했었다. 또한 일회용 패스워드 방식은 사용 횟수의 제한과 함께, 안전성이 해쉬 함수에 근거하고 있다. 또한 기밀성 및 상호 인증을 제공 못하는 단점이 있었다.

본 고에서 제안한 방식은 상기 방식들의 문제점들을 모두 해결하고 있으며, 특히 일회용 패스워드를 발전시켜 암호 알고리즘에 안전성을 둔 상호 인증 및 기밀성을 제공하고 있다. 향후 정보 인프라의 발전을 고려할 경우 더욱 안전하고 효율적인 방식을 위한 광범위한 연구가 진행되어야 할 것이다.

참고문헌

[1] W. Ford, "Computer Communication Security", Prentice Hall, pp.109-148, 1994.
[2] "전산망에서의 패스워드 누출 방지 기술 개발 보고서“, 한국정보보호센터, 1997. 12.
[3] N. Haller, "The S/Key One-Time Password System", RFC 1760, 1995.
[4] R. Rivest, "The MD4 Message-Digest algorithm", RFC 1320, April 1992.
[5] Mudge, "Vuleravilities in the S/Key one time password system", http://10pht. cp/~modge/skey_white_paper.html
[6] R. Rivest, "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
[7] D. Feldmeier and P. Karn, "UNIX Password Security-Ten Years Later", CRYPTO Proceedings, Summer 1989.
[8] B. Schneier, Applied Cryptography, Second Edition, John Wiley & Sons, 1996.