일회용 패스워드 시스템의 표준화 연구[1)]

장 청 룡*○, 이 용 권*, 양 형 규**, 이 완 석***, 홍 기 웅***
경동대학교* 강남대학교**, 한국정보보호센터***

A Study on the Standardization of One-Time Password System

Chung-ryong Jang*○, Yong-kwon Lee*, Hyung-kyu Yang**, Wan-suk Yi***, Ki-Yoong Hong***
Kyungdong University*, Kangnam University**, Korea Information Security Agency***

< 요 약 >

본 논문에서는 먼저 인증에 대한 개념, 인증 서비스를 하기 위한 사용원칙, 그리고 인증 메커니즘의 분류를 살펴보고 특히, 일회용 패스워드를 이용한 인증 시스템의 현황과 기 제안 방식들의 특장점 등을 논의한다. 그리고, 이러한 논의를 바탕으로 국내 실정에 적합한 일회용 패스워드시스템의 표준화를 위한 표준화 방향을 검토하여 표준 시스템을 제안하고 이의 실용성과 안전성을 평가해 보기로 한다.

1. 서 론

'90년대 이후부터 전세계적으로 인터넷의 보급 확산으로 다양한 계층의 사용자가 망에 연결된 각 조직의 호스트급, 서버급 컴퓨터나 워크스테이션에 접속하여 원하는 정보를 공유하고 있다. 그러나, 일부에서는 특정 정보통신 또는 전산 자원들에 대하여 인증된 사용자만의 접근을 허용하여 관리하고 있다. 이를 위하여 특정 정보에 대한 접근이 제한적인 지역에서는 물리적 보안을 강화하고 원격지에서의 망을 통하여 접근할 때 정당한 사용자를 인증하기 위한 정적인 패스워드 또는 이들의 주기적인 갱신에 의한 방식이 일반적으로 사용되고 있다. 그러나, 이러한 방식은 특히 망을 통한 원격접속시 패스워드가 망을 통해 전달시 이를 검출하여 악의의 제3자가 이를 도용하므로써 정당한 사용자인양 재 사용할 가능성이 많아 최악의 경우 귀중한 정보자원의 유출은 물론 이의 손괴를 야기시킬 수 있을 것이다.

따라서, 이러한 정적인 패스워드를 사용할 때마다 동적으로 매번 다르게 사용하는 방식이 필요하게 되었으며 이를 구현하는 방법의 하나가 바로 일회용 패스워드를 이용하는 것이다. 최근 일회용 패스워드 방식은 물리적인 보안 시스템, 정보통신 및 전산시스템에서의 사용자인증, 특정 조직의 안전한 망운용을 위해 도입하는 방화벽과 같은 침입차단시스템에서 정당한 사용자의 인증 등을 위하여 폭 넓게 활용되고 있다. 그러나 이러한 인증방식을 적용하는 조직에서는 각자 자신의 환경에 적당한 다양한 벤더들의 제품을 이용하여 구축 운용함으로 인하여 상호운용성에 문제가 일으킬 수 있어 이에 대비한 일회용 패스워드에 대한 표준화가 시급히 요구되고 있다.

따라서, 본 논문에서는 먼저 인증에 대한 개념, 인증 서비스를 하기 위한 사용원칙, 그리고 인증 메커니즘의 분류를 살펴보고 특히, 일회용 패스워드를 이용한 인증 시스템의 현황과 기 제안 방식들의 특장점 등을 논의한다. 그리고, 이러한 논의를 바탕으로 국내 실정에 적합한 일회용 패스워드시스템의 표준화를 위한 표준화 방향을 검토하여 표준 제안 시스템을 실용성과 안전성을 평가해 보기로 한다.

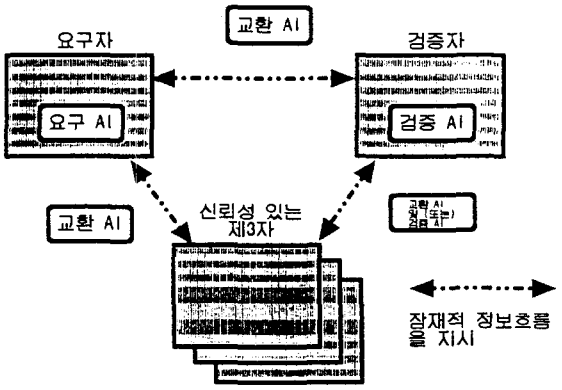
2. 일회용 패스워드 시스템

2.1 인증 서비스 개요

1. 본 연구는 1998년도 한국정보보호센터의 위탁과제 연구비 지원에 의한 결과의 일부입니다.

인증은 요구된 실체의 신원에 대한 보증 기능을 제공한다. 인증은 다음의 두 가지 유형으로 구별한다. 즉, 주체와 검증자간에 통신을 하는 관계인 실체 인증과 주체가 다른 실체에 유용한 데이터 항목의 발신처라 할 때인 데이터 발신처 인증으로 나눈다. 실체 인증은 주체의 신원 확인을 통신 과정을 통하여 수행되며 그 주체의 인증된 신원은 이 서비스가 시행 과정에만 보증된다. 본 논문에서 논의하는 일회용 패스워드를 이용한 실체인증에서는 신뢰적인 제3자가 배제된 경우만을 다루기로 한다[1,2].

(그림 1)은 요구자, 검증자, 신뢰성 있는 제3자 사이의 관계 및 인증 정보의 세 가지 유형을 나타낸다.



(그림1) 인증 실체의 구성 및 인증 정보의 유형

2.2 인증에 사용되는 원칙

일반적으로 특정한 인증 방법은 하나 이상의 원칙에 관련된 일련의 기대나 가정에 의존한다.

그 원칙은 다음과 같은 것들을 포함하여 사용된다[1,2,3].

1) 알고 있는 것 (SYK: Something You Know; 예, 비밀번호)
2) 소유하고 있는 것 (SYH: Something You Have; 예, 자기 카드 또는 스마트 카드)
3) 불변의 특성 (SYA: Something You Are; 예, 생체학적 식별자)
4) 제3의 실체(신뢰성 있는 제3자)가 인증을 설정하였음을 인지
5) 구문 (예, 주체의 주소)

이 모든 원칙에는 고유의 약점이 있음을 주의해야 한다. 예를 들면 소유하고 있는 것에 의한 인증은 종종 소유자를 인증하다기보다 소유물에 대한 인증이다. 어떤 경우에 그 약점은 몇 가지 원칙의 조합으로 극복된다. 예를 들면 스마트 카드(소유하고 있는 것)가 사용될 때 그 약점을 카드에 그 소유자를 인증시키기 위해 PIN(알고 있는 것)을 추가하여 극복할 수 있다. 더욱이 원칙 5)는 특히 약하여 항상 다른 원칙과 조합하여 사용되는 것이 보편적이다.

2.3 일회용 패스워드의 유형

주지하다시피 고정 패스워드를 이용하는 경우는 많은 안전성의 문제를 내포하고 있음을 알 수 있었다. 한편 몇몇 문제는 어떠한 패스워드 기반의 시스템에서도 고유한 것이며, 그 밖의 문제는 구현 시스템의 내부 처리 과정에서 보안의 취약성으로부터 야기되는 것이다. 예를 들면, 어떤 고정 패스워드 인증 메커니즘을 사용하더라도 추측과 사전공격에 대하여 강한 패스워드를 선택하여야하는 것은 대단히 의미가 있다. 그러나, 안전한 패스워드라도 안전하지 못한 정보통신 채널을 통하여 평문으로 입력된다면 이 역시 망에서의 도청에 취약할 것이다.

이러한 공격을 방지하기 위한 대안적 기법이 바로 일회용 패스워드이다. 고정 패스워드에 기반을 둔 인증 메커니즘과는 달리, 일회용 패스워드에 기반을 둔 것은 평문형태의 매번 다른 패스워드 입력에 의해 서도 결코 취약하지는 않을 것이다.

본 절에서는 표준화 측면에 강조되는 실용성과 안전성을 고려하여 현재 인터넷 부문에서 표준화가 추진중인 IETF RFC 2289 일회용 패스워드 시스템, 그리고 이의 원조인 Bellcore의 S/KEY, 공개키에 기반을 둔 서명고리 방식 그리고 실용성을 강조한 SecuPass II를 소개하고 이들에 대한 안전성 문제를 검토해 보기로 한다[3,4,5,6].

2.3.1 일회용 패스워드 인증 방식의 유형

일회용 패스워드 인증 방식의 분류는 그 분류 기준에 따라 다양하게 나눌 수 있다. 본 논문에서는 먼저 시간 동기에 의한 분류 기준으로 분류하고 이를 계속해서 비동기 방식에서 암호학적 기법의 적용여부에 의한 분류 방식으로 나누어 정리하기로 한다.

이러한 방식의 분류와 각 방식에 대한 특장점과 문제점이 (그림 2)와 <표 1>에 도시되어 있다.

2.3.2 일회용 패스워드 운용 사례

가. IETF RFC 1760 S/KEY 방식 [5]

IETF RFC 1760에서 규정하는 Bellcore의 S/KEY 일회용 패스워드 시스템의 아이디어는 Leslie Lamport (1981)에 의해 제안된 방식을 채택하고 있으며 Phil Karn에 의해 UNIX시스템 상에서 구현되었다. 이의 일차적인 목표는 안전하지 못한 네트워크 상에서의 안전한 패스워드 기반의 인증을 제공하자는 것이다. S/KEY에서는 인증과정에서 단 한번만 사용하는 일련의 패스워드를 알고리즘적으로 생산하는 사용자의 비밀 패스워드를 이용하여 이 목표를 달성하였다. 표준 UNIX 패스워드에서와 같이 어떤 일회용 S/KEY 패스워드도 서버 시스템에는 평문형태로 저장되지는 않는다. 비밀의 패스워드는 항상 이를 아는 사람들이 비밀로 간직하며 표준 UNIX 패스워드와는 달리 통신망을 통해 결코 전달되지는 않는다.

S/KEY 운용의 핵심은 한 방향으로의 계산은 쉽게되지만 그의 역 계산은 계산적으로 어려운 일방향 해쉬함수를 사용한다는 것이다.

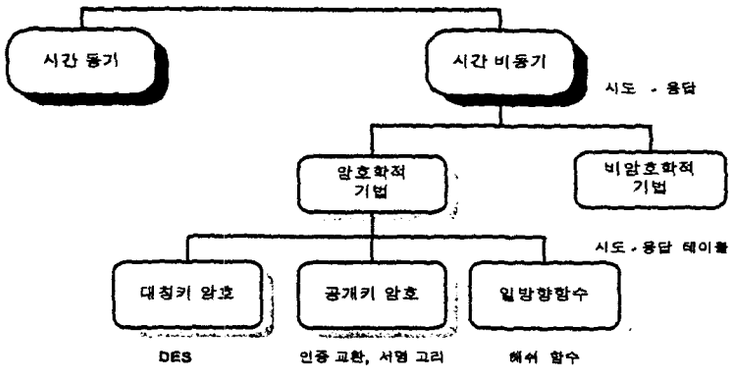
사용자 i가 일회용 패스워드를 n회 적용하여 로그-인을 할 수 있도록 하기 위하여 S/KEY 시스템에서는 먼저 사용자 i의 패스워드를 요구한다. 편의상 n=3이라 하면, S/KEY는 비밀 패스워드를 MD4에 3회 적용시킨다. 이 결과인 $f(f(f(s)))) = f^3(s)$ 가 S/KEY 패스워드 데이터베이스에 저장된다.

그러면 사용자 i가 처음 시스템에 로그-인을 할 때, 그는 평문 형태의 $f^2(s)$를 서버에게 전달하게 된다. 서버 측의 S/KEY 로그-인 프로그램은 이것을 접수하여 한번더 해쉬를 취한다. 즉, $f(f^2(s)) = f^3(s)$을 계산한다. 이 값이 패스워드 데이터베이스에 저장된 것과 일치하면 그 사용자는 사용자 i이거나 아니면 그의 비밀 패스워드를 아는 사람일 것이다.

그리고 나면, 패스워드 데이터베이스에 있던 $f^3(s)$ 는 $f^2(s)$ 로 갱신된다. 그후 사용자 i가 마지막 번째의 로그-인을 하면, $f(s)$를 제시하여 인증을 받게된다.

여기서 몇 가지 유의 사항이 있다. 먼저, S/KEY 패스워드 데이터베이스에는 어떤 비밀 패스워드로도 저장되어 있지 않다는 것이다. 여기에는 비록 선택된 비밀 패스워드에 대한 사전공격이 가능할지라도 공격자에게 거의 이용가치가 없는 해쉬가 적용된 $f^N(s)$을 간직하게 된다. 두 번째로, 연속적인 일회용 패스워드는 그들 자체로부터 현실적으로 유도할 수 없으므로 이러한 일회용 패스워드들은 평문으로 전달하여도 안전하게 된다. 세 번째로, 일단 검증이 되면 현재 사용한 일회용 패스워드는 다음번 것을 검증하기 위한 도구로 이용하게 된다.

그러나 이 방식은 일회용 패스워드로서 구현된 최초의 간단하고 실용적인 것이지만 안전성 측면에서 고정된 seed의 사용으로 특정 통신과정에서 replay공격에 취약하다.



(그림 2) 일회용 패스워드 인증 방식의 분류

< 표 1> 일회용 패스워드 인증 기법의 특장점

구 분	이용 기법		적용례	특 징	문제점	비 고
시간 동기			휴대용 난수 생성기	시간에 대한 함수값(난수) 생성	-시간동기 -시간편차(장치의 노후화) -패스워드(난수)의 유효성	
시간 비동기	암호학적 기법	대칭키	DES	-간단, 안전	-시도-응답 절차로 다소 시간 소요 -키 관리 -서버의 신뢰	
		공개키	인증교환	-공개키 인증교환 -안전	-구현 문제 -이식성 문제	
			서명 고리	-공개키 서명 -안전	-레이스 공격 -서비스거부 문제 -비용 문제(고가)	
		일방향 함수	해쉬함수	-간단, 안전	-레이스 공격 -스푸핑 공격	-고정 씨드를 난수로 하여 공격 대응
	비암호학적 기법		시도-응답 테이블	-간단	-안전성 결여(모든 공격에 취약)	

나. IETF RFC 2289 일회용 패스워드 시스템 [6]

RFC 2289 방식은 당초 Bellcore의 S/KEY를 기초로한 RFC 1760의 내용 및 체제를 보완시켜 RFC 1938로 제안한 것을 좀더 포괄적인 일회용 패스워드 시스템의 인증 서비스를 위하여 최근 RFC 2289의 형태로 발전시켜 이를 표준화하고 있는 단계에 있다.

이 방식에서의 인증 시스템은 일련의 일회용 패스워드를 생성하기 위하여 기존의 S/KEY 방식과 같이 비밀의 패스-프레이즈(pass-phrase)를 이용한다. 이 시스템을 이용하면, 사용자의 비밀 패스-프레이즈는 인증 과정 동안 혹은 패스-프레이즈의 변경동안에는 언제든지 통신망을 가로질러 전송할 필요는 결

코 없게 된다. 따라서, 이것은 재사용 공격에 결코 취약하지 않게 된다. 부가된 보안성은 서버가 적절히 보호된다는 것을 포함하여 어떤 비밀 정보도 어떤 시스템에 저장될 필요가 없다는 성질에 의해 제공된다.

이 방식은 인증 서브 시스템에 대한 외부로부터의 수동적 공격을 보호하지만 망을 통한 도청자로 하여금 비공개 정보의 접근을 방지하지는 못하며 또한 " social engineering " 혹은 능동적 공격에 대한 보호를 하지 못한다.

일회용 패스워드 시스템의 운용을 위하여 두개의 실체인 생성기와 서버로 구성된다. 생성기는 사용자의 비밀 패스-프레이즈와 서버로부터 수신한 시도(challenge)에 포함된 정보로부터 적절한 일회용 패스워드를 생성해야 한다. 또한, 서버는 생성 파라미터를 포함한 시도를 해당 생성기로 보내야 하고, 수신한 일회용 패스워드를 검증해야 하며, 수신한 바로 이전의 유효한 일회용 패스워드를 저장해야 하며, 마지막으로 그에 상응하는 일회용 패스워드의 순차번호를 저장해야 한다. 사용자의 비밀 패스-프레이즈의 변경을 안전한 방법으로 용이하게 처리하여야 한다.

즉, 생성기는 시도의 일부로서 서버로부터 수신한 씨드 값과 사용자의 패스-프레이즈를 함께 안전한 해쉬함수에 입력시켜 복수회의 반복연산에 의해 하나의 일회용 패스워드를 생성한다. 매번 인증을 성공한 후에는 안전한 해쉬함수의 반복 회수는 하나씩 감소하게 된다. 따라서, 하나의 유일한 패스워드 순차번호가 생성된다.

서버는 생성기로부터 수신한 일회용 패스워드를 안전한 해쉬함수에 한번 적용시켜 연산을 한 후 그 결과를 이전에 접수한 일회용 패스워드와 비교함으로써 검증을 하게 된다.

이와 같은 RFC 2289 방식은 기존 S/KEY 방식이 UNIX 기반 시스템에 응용으로 제안된 것을 보다 다양한 응용에 포괄적으로 적용할 수 있도록 제안되었다. 즉, 클라이언트와 호스트간의 인증을 보다 확대하여 인증에 사용되는 패스-프레이즈 생성기와 서버간의 인증으로 적용범위를 휴대용 인증기와 이에 대응하는 단말간의 인증으로부터 일반적인 전산 환경의 클라이언트 서버간의 인증까지를 포함한다. 또한, 구현의 용이성 등 실용적 측면에서 일방향 함수의 적용하고 있으며 안전한 해쉬함수의 수용 범위에 SHA1을 추가하였으며 아울러 이러한 유사한 알고리즘에 대한 인터페이스의 확대 적용이 가능하도록 하였다. 한편, 운용절차는 S/KEY 방식과 같으나 구현상에 따른 안전성에 대한 검토로서 레이스공격에 대한 방어 수단의 언급을 추가하였으며 세션하이제킹과 같은 공격에 대하여는 관련 IP 보안 기술의 이용을 권고하고 있다.

다. 서명 고리 방식 [4]

서명 고리 개념을 이용한 공개키 방식의 일회용 패스워드는 사용자 인증을 위하여 씨드(seed)에 서명한 sign(seed)를 사용하고 이 sign(seed)를 다음번 씨드로 대체하여 사용한다. 이의 동작 절차는 사용자 등록 과정과 인증과정으로 구분한다. 즉 사용자 i의 등록 과정에서는 먼저, 사용자 i 는 호스트 측에 등록을 신청한다. 그러면 호스트에서는 사용자 i 의 (공개키 Pi, 비밀키 Si) 쌍을 계산하고, 랜덤 초기값 씨드를 생성한다. 호스트는 사용자 i에 대응하는 (IDi, Pi, $seedi$)를 저장한 후, 사용자 i 에게 (Si, $seedi$)를 분배한다. 이렇게 등록을 마치면 인증을 할 수 있게된다. 즉, m번째 인증과정에서 사용자 i 는 $sign^{m-1}(seedi)$에 대한 서명값, $sign^{m}(seedi)$를 계산한 후, 이 서명 값을 호스트 쪽에 전송한다. 그러면 사용자 i는 자신의 비밀 정보인 (Si , $sign^{m-1}(seedi)$)를 (Si , $sign^{m}(seedi)$)로 갱신한다. 이를 이용하여 호스트에서는 사용자 i의 공개키를 이용하여 서명 값을 확인하고 이것이 맞으면 호스트도 (IDi, Pi, $sign^{m-1}(seedi)$)를 (IDi, Pi, $sign^{m}(seedi)$)로 갱신한다.

이 방식은 기존의 S/KEY 근간의 방식들이 비밀정보의 갱신에 따른 일회용 패스워드의 생명주기의 사용횟수에 제약을 받는다는 점을 개선하였다. 이는 기존의 공개키를 이용한 인증교환 방식에서와 같이 매번 다른 응답을 생성할 수 있으며 사용자 측에서 호스트 측으로 일회의 통신으로 인증절차를 마칠 수

있는 간단한 일회용 패스워드 시스템이다. 그러나, 이 방식은 레이스공격, 서비스 거부와 같은 안전성 문제를 내포하며 실용성 측면에서 기존의 공개키 시스템에서의 인증 시스템에 비교하여 IC 카드 등과 같은 인증토큰의 이용시 사용자 정보의 갱신을 위하여 카드의 write 기능이 추가되어 비용측면에서 다소 부담이 된다.

라. SecuPass II [4]

이 방식은 해쉬함수의 특성과 난수 생성 방법을 이용하여 통신채널을 통하여 단 한번의 유효성을 갖는 패스워드만이 사용된다. 이 방식은 S/KEY 방식과 마찬가지로 도청 공격이나 재시도 공격과 같은 수동적에 안전하며 더욱이 씨드대신에 난수를 사용하므로써 매번 다른 시도와 이에 대응하는 응답을 처리하여 레이스 공격을 저지할 수 있다.

한편 초기화 과정에서 사용자의 비밀 패스워드를 서버 측에 임의의 수 N 만큼 해쉬한 값을 저장하여 역으로 사용자의 패스워드 유추가 불가능하며 해쉬되어 저장되는 정보도 비밀 정보화 처리하여 내부 및 외부의 공격에 안전하도록 한다.

더욱이, 시도-응답에 의한 인증절차로 사용자와 서버들이 상호 위장 공격에 대처할 수 있는 상호 인증 기능을 제공한다.

이 방식은 사용자 등록과정인 초기화 과정과 인증과정으로 구분되며 초기화 과정은 일반적인 인터넷 부문의 과정과 유사하므로 생략하고 여기서는 사용자와 서버간의 인증과정만을 약술하기로 한다.

먼저 사용자가 사용자의 ID를 가지고 접근 요구를 서버 측에 보내게 된다. 서버 측에서는 난수 R을 생성한 후 사용자의 ID를 보고 이에 해당하는 N과 X_{N+1}을 이용해서 ($N \| R \oplus X_{N+1} \| H(R \oplus X_{N+1})$)을 시도(challenge) 값으로 사용자 측으로 보낸다. 여기서 $H(R \oplus X_{N+1})$은 난수 R과 X_{N+1}을 입력으로 하여 해쉬함수를 수행하는 것이다.

도전 값을 받은 사용자 측은 사용자가 가지고 있던 비밀키 P를 도전 값의 N 만큼 해쉬함수를 반복 수행해서 X_N과 X_{N+1}를 계산한다. 이것의 X_{N+1}을 도전 값의 $R \oplus X_{N+1}$와 $\oplus$ 연산을 수행하면 난수 R만 남게 된다. 여기서 얻은 난수 R과 X_{N+1}값으로 해쉬함수를 수행하는 $H(R \oplus X_{N+1})$의 결과를 사용자 측에서 계산한 $H(R \oplus X_{N+1})$과 비교하여 일치하게 되면, 서버 측의 소유정보인 X_{N+1}과 난수 R을 확인할 수 있어 서버 인증이 된다.

사용자 측은 다시 X_N을 난수 R과 $\oplus$ 연산을 수행한 결과인 $(R \oplus X_N)$를 응답(response) 값을 서버 측으로 보낸다. 서버 측에서는 이 응답 값을 수신하여 서버 측에서 생성한 난수 R을 이용하여 $H(R \oplus (R \oplus X_N))$의 결과와 X_{N+1}을 비교하여 일치하면 사용자의 신분을 확인할 수 있게 되므로 사용자와 서버 간의 상호 인증이 된다.

이 후에 서버 측에서는 사용자에 관한 저장정보를 N+1에서 N으로, X_{N+1}에서 $H_N(P)$ 측, X_N으로 갱신하면 모든 절차는 끝나게 된다. 이후 사용자 측으로부터의 인증요구가 다시 발생하면, 상기의 절차를 반복 수행하면 된다.

2.4 일회용 패스워드 시스템의 보안 취약점 분석 [3,4,6]

2.4.1 사전(Dictionary) 공격

일회용 패스워드 시스템중 S/KEY의 취약점은 8자리 이상의 패스-프레이즈를 포함한 모든 인증 정보가 평문(plaintext)으로 전달되어진다는 것에 있다. 이것은 시도와 응답을 악의의 제3자가 알 수 있다는 것을 의미하며, 이 정보들을 가지고 사전의 단어들에 적용한 시도의 결과와 비교하는 것이다. 그러나 이를 보완한 RFC 2289에서는 이러한 사전 또는 전수 공격에 대비하여 10자리 이상 63자리까지의 패스-프레이즈를 반드시 지원하도록 하고 있다.

또한, 기존의 S/KEY 패키지가 클라이언트나 서버 어디서도 사용자 패스워드 선택시에 보안성 검사

를 하지 않는다는 중요한 결점을 가지고 있다는 것이다.

사전 공격에 대한 또 다른 문제점은 /etc/skeykeys 파일에 있다. 아주 놀랍게도 skey를 사용하고 있는 호스트들이 이 파일에 부적절한 permission을 갖고 있다는 것이다. 이 파일은 누구나 읽기 가능하도록 하지 말아야 한다. skeykeys 파일의 구조는 아래와 같다.

```
root    0072  k113357      12afaa8be65f0502     Jun 29,1995 12:40:48
jdoe    0099  k113355      c7f42dfd84914af3     May 30,1995 16:20:19
[etc...]
```

여기서 우리는 사용자 이름, iteration counter, 씨드, 그리고 다섯 단어 응답의 16진수 표현을 볼 수 있다. 나머지 세 필드는 단순히 날짜 시간 정보들이다. 이 파일의 정보들을 가지고 앞서 언급한 사전공격 방식 그대로 해킹이 가능하다.

2.4.2 스푸핑(Spoofing)공격

S/KEY와 RFC 2289 방식에서는 iteration counter가 씨드와 함께 클라이언트 측으로 전송이 되어지기 때문에, 서버로 위장하는 공격 가능성을 가지게 된다. 이 공격 방법은 가짜 게이트웨이를 설정함으로 수행되어진다. 다음의 시나리오를 살펴보자.

```
login : jdoe      /*jdoe가 telnet으로 호스트에 접속 시도한 후*/
s/key 55 k113355   /*jdoe는 98번째의 시도 대신에 55번째의 시도를 받았다.*/
password :
password : MY SPIT SOFT HEAD REAR
 /*jdoe의 계산기는 그의 패스워드를 이용하여 55 k113355에 대한 응답을 생성한다.*/

Login incorrect
login : /* 가짜 호스트는 login이 틀렸다고 말하고, jdoe가 실제로 원하는 호스트로 다음 연결을 하게 한다. */
```

여기서 55 k113355 시도로부터 얻어진 응답을 가지고 스니퍼는 해쉬함수를 이용하여 나머지 응답들을 알아낼 수 있게 된다. 예를 들면, 지금 가지고 있는 정보를 이용해서 60 k113355에 대한 응답을 알고 싶다면, 해쉬함수를 다섯 번만 수행하면 원하는 응답을 얻을 수 있게 되는 것이다.

2.4.3 Race 공격

이것은 같은 키를 가지고 동시에 login을 시도하는 두 프로세스를 허용하는 S/Key, RFC 2289, 서명 고리 방식의 문제점에 관한 내용이다. 만약 공격자가 지나가는 jdoe의 응답들을 획득할 수 있다면, 같은 호스트에 다른 telnet 세션을 열 수 있고, iteration counter가 줄어들기 이전의 같은 시도를 얻을 수 있다. 그후 jdoe의 응답을 가지고 login을 시도하여 운 좋게 locking problem이 발생하게 된다면, 둘다 같은 시도와 응답을 가지고 login에 성공하게 된다. 이것은 동시에 인증 세션을 개시하지 않도록 하는 대안으로 해당 방식의 소스 코드를 수정하여 쉽게 고쳐질 수 있는 문제이다.

3. 일회용 패스워드 시스템의 표준화

3.1 표준화 고려사항

일회용 패스워드 시스템의 표준화를 추진하기 위하여 여러 가지의 고려 사항이 있겠으나 인증서비스의 관점에서의 안전성, 지금까지의 이용 방식의 편의성, 구현 및 운용의 용이성 등을 고려한 실용성을 중심으로 어떠한 방식이 표준화에 적합한가를 검토해 보기로 한다. 이러한 검토를 위하여 앞서 정리한 일회용 패스워드 방식의 분류 형식을 이용하기로 한다.

먼저 안전성 측면에서는 시간 비동기 방식의 암호학적 변환 함수를 이용하는 방식이 비교적 안전할 것이다. 실용적 측면에서는 사용 알고리즘의 단순성, 구현의 용이성, 적용환경에의 이식성, 키 관리 , 비밀 정보의 관리, 지적소유권, 비용 문제 등이 고려 대상이 될 것이다.

실용성 검토에 대한 고려사항에 대한 분석은 <표 2>에 정리된 바와 같이 해쉬함수를 이용하는 간단한 방식이 대체로 우수한 것으로 분석되어 국내 표준화에 안전한 해쉬함수를 이용하는 방식이 적절한 것으로 생각된다.

< 표 2> 암호학적 함수를 이용한 일회용 패스워드 방식의 실용성 검토

범례 : 양호 ○ : 보통:△

이용 기법	적용례	구현 용이성	이식성	키관리	비밀 정보	보안 공격	지적 소유권	비 고
대칭키 암호	DES	○	○	△	△	○	○	보통
공개키 암호	인증교환, 서명 고리	△	△	△	△	○	△	보통
일방향함수	해쉬함수	○	○	해당 없음	△	○	○	우수

3.2 표준화 요구조건

3.2.1 체제

한국정보통신표준(안)으로 상정하기 위하여 이의 체제로 작성한다.

3.2.2 구성

가. 일반사항 : 한국정보통신표준의 구성을 준용하여 작성한다.

나. 세부사항

- 안전한 해쉬함수: 국내 표준(안)의 개발을 목표로 하므로 안전한 해쉬함수의 채택은 현재 한국정보통신기술협회의 관련 위원회에서 표준화중인 HAS (Hash Algorithm Standard) - 160을 포함시키고 아울러 기존 일회용 패스워드 시스템과의 연동을 고려하여 IETF RFC 2289에서 규정하는 MD4, MD5 및 SHA1을 적절히 수용한다.
- 일회용 패스워드 시스템의 운용 : 인증서비스의 프레임워크에서 규정하는 운용관련 기능 또는 절차에 맞추며 그 내용은 IETF RFC 2289에서 규정하는 내용을 수용하지만 안전성 문제에 대한 것은 3.3.2절에서 논의한 표준 제안 일회용 패스워드 시스템의 운용 절차에 따라 보완하여 규정한다.
- 부기 : HAS-160을 포함한 안전한 해쉬 알고리즘에 관한 인터페이스를 정리한다. 이러한 해쉬함수를 이용한 일회용 패스워드 시스템의 검증 예도 HAS-160을 포함하여 정리한다. 대안적 사전 알고리즘과 6-단어와 2진 포맷간의 변환 사전은 IETF RFC 2289를 수용한다.
- IETF RFC 2289의 수용에 따른 copyright 문제의 처리 : 참조 표준 및 권고 항에서 인터넷 그룹의 IETF RFC 2289를 수록하며 이를 각주로 처리하고 여기서 copyright notice를 언급한다.

3.3 표준화 주요 사항

본 절에서는 RFC 2289 수준의 인증 즉, 서버 측과 생성기 측의 상호 인증보다는 생성기 측만에 대한 인증과 관련된 일회용 패스워드 시스템 표준화를 위하여 요구되는 주요 기술적 사항을 소개하며 이들이 앞서 정리한 표준화 고려 사항과 요구조건에 적합한가를 평가해 보기로 한다.

특히, 실용성의 측면에 구현 비용, 이용의 편의성을 고려할 때 키를 사용하는 암호학적 기법보다는 비교적 구현 및 처리속도가 우수하며 키관리를 별도로 요구하지 않는 안전한 일방향 해쉬함수를 사용하는 방식을 채택하기로 한다.

3.3.1 안전한 해쉬함수

일회용 패스워드 시스템의 안전성을 유지하는 주요 알고리즘으로 현재 국내 표준화 중인 Hash Algorithm Standard (HAS-160)을 이용하기로 한다.

정보처리시스템 및 정보통신망 환경에서 인증, 무결성, 부인봉쇄 서비스를 제공하고자 하는 모든 정보통신서비스에 보조적 수단으로 사용할 수 있는 알고리즘인 HAS-160은 특히 인증서비스에 적용하기 위한 일회용 패스워드 시스템에서 일방향 함수로 활용할 수 있다. 이 알고리즘은 임의의 길이를 가지는 입력 메시지를 512비트의 블록 단위로 입력 처리하여 해쉬코드(출력길이)가 160비트로 압축시킨다. 또한 기존의 각종 암호학적 공격으로부터 안전하게 설계되었으며, 같은 출력을 내는 SHA-1(Secure Hash Algorithm, FIPS PUB 180-1)이나 RIPEMD-160보다 효율적으로 설계되어 안전성과 실용성 측면에서 우수한 것으로 평가되고 있다[7, 8].

가. HAS-160 일회용 패스워드 인터페이스

HAS-160 해쉬알고리즘을 사용하여 일회용 패스워드 시스템 구현을 구현할 수 있다. 이의 기본 구조는 RFC 2289과 유사한 인증 처리 과정을 취한다. 그러나, 스푸핑을 통한 위장 공격에 대응하여 SecuPass II 과 유사한 묵시적 상호 인증을 할 수 있는 기능도 지원한다.

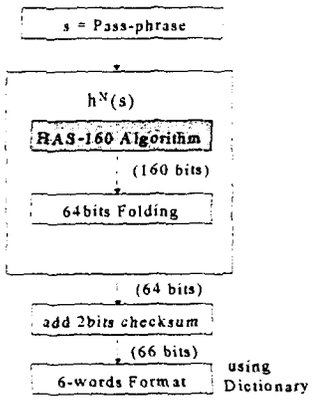
일회용 패스워드 생성을 위한 처리 단계에서는 (그림 3)에서와 같이 MD계열과 동일하게 처리한다. 즉, HAS-160에서 생성된 해쉬코드 출력길이가 160비트이므로 일회용 패스워드 시스템에서 요구하는 일회용 패스워드 출력 형식인 64비트로 출력시킨다. 그리고, 이러한 비트를 인증과정에서의 응답인 패스-프레이즈로 처리하기 위하여 인간이 짧고 쉽게 입력시킬 수 있는 단어로 부호화 처리한다. 이를 위하여 64비트의 출력과 이들 각 비트의 쌍에 대한 검사합 2비트를 추가하여 66비트로 처리한다. 그리고 나서 2048개로 구성된 표준사전을 통하여 인간 사용자가 읽을 수 있는 6워드 형태로 출력시킨다.

3.3.2 운용 절차

제안 일회용 패스워드 시스템의 운용절차는 (그림 4)에 보여진 바와 같이 초기화 과정과 인증과정으로 구성된다.

먼저 초기화 과정은 사용자가 자신의 패스워드와 같은 비밀정보(S)를 이용하여 이를 서버 측과 동일한 일방향 해쉬함수인 HAS-160 알고리즘에 적용시켜 $H_0 = h(S)$를 얻는다. 이 결과를 자신의 ID와 함께 서버에 안전한 채널을 통하여 등록을 한다. 그러면 서버 측에서는 H_0를 임의의 수 N과 그 만큼 해쉬한 값인 H_N만을 저장하여 역으로 사용자의 패스워드 유추가 불가능하도록 하며 해쉬되어 저장되는 정보도 비밀 정보화 처리하여 내부 및 외부의 공격에 안전하도록 한다.

다음으로 인증과정에서는 시도-응답에 의한 인증절차로 사용자와 서버들이 상호 위장 공격에 대처할 수 있는 상호 인증 기능을 제공한다. 즉 서버로의 위장 공격에 대하여는 사용자의 응답으로부터 곧바로 H_i (H_i 는 임의의 횟수만큼의 해쉬함수 적용 결과)를 얻어 질 수 없도록 단독으로 사용하지 않도록 한다. 그리고, 레이스 공격에 대하여는 RFC 2289에서의 고정된 씨드의 사용 대신에 일반적인 공개키 기반의 인증교환에서와 같이 매 로긴 마다 각각 다른 난수를 사용하도록 한다. 이에 대한 절차를 다음과 같다.

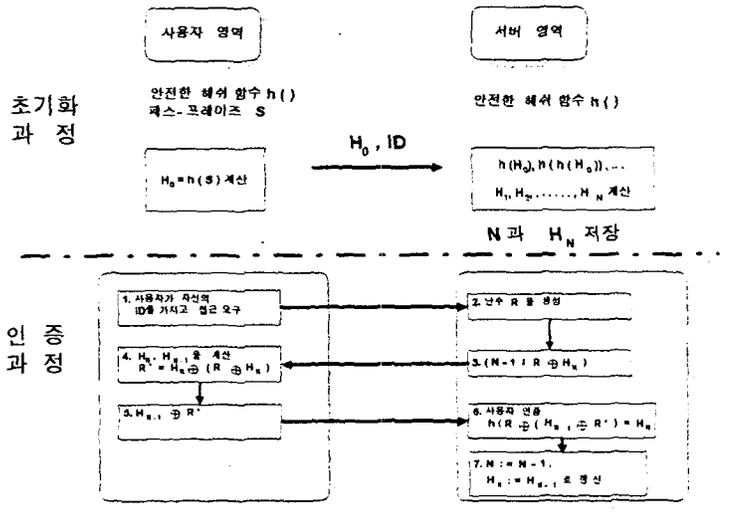


(그림 3) HAS-160 일회용 패스워드 인터페이스의 응답 생성

먼저 사용자가 자신의 ID를 가지고 접근 요구를 서버 측에 보내게 된다. 서버 측에서는 난수 R을 생성한 후 사용자의 ID를 보고 이에 해당하는 H_N을 이용해서($(N-1) \| R \oplus H_N$)을 시도(challenge) 값으로 사용자 측으로 보낸다.

시도 값을 받은 사용자 측은 사용자가 가지고 있던 H_0를 시도 값의 N과 N-1만큼 각각 해쉬함수를 반복 수행해서 H_N과 H_{N-1}를 계산한다. 이것의 H_N을 시도 값의 $R \oplus H_N$와 $\oplus$ 연산을 수행하면 난수 R만 남게 된다.

사용자 측은 다시 H_{N-1}을 난수 R과 $\oplus$ 연산을 수행한 결과인 $(R \oplus H_{N-1})$를 응답(response) 값으로 서버 측에 보낸다. 서버 측에서는 이 응답 값을 수신하여 서버 측에서 생성한 난수 R을 이용하여 $H(R \oplus (R \oplus H_{N-1}))$의 결과와 H_N을 비교하여 일치하면 사용자의 신분을 확인할 수 있게 되므로 사용자를 인증이 된다. 물론 여기서 서버에 대한 인증 절차는 명확히 없지만 올바른 H_N를 계산할 수 있는 주체는 결국 정당한 서버만이 할 수 있으므로 묵시적인 인증을 실현할 수 있게 된다.



(그림 4) 표준 제안 일회용 패스워드 시스템 구조

이 후에 서버 측에서는 사용자에 관한 저장정보를 N에서 N-1로, H_N에서 H_{N-1} 로 갱신하면 모든 절차는 끝나게 된다. 이후 사용자 측으로부터의 인증요구가 다시 발생하면, 상기의 절차를 반복 수행하면 된다.

본 제안 일회용 패스워드 시스템은 RFC 2289에서 안전성 문제로 제시된 레이스 공격과 서버로의 위장을 하는 스푸핑 공격을 회피할 수 있는 SecuPass II 방식과 동일한 안전성을 유지하면서 클라이언트에서의 서버 인증을 위한 연산을 생략하더라도 묵시적으로 서버의 정당성을 확인할 수 있는 안전하고 실용적인 시스템으로 평가된다. 그러나, 사용자의 정보를 얻기 위한 정당한 서버로의 위장 공격에 대하여는 SecuPass II 방식에서와 같은 상호 인증을 선택적 사항으로 수용함이 바람직할 것이다.

4. 결 론

정보처리 전산시스템의 다양한 인증 환경 하에서 기존의 정적인 패스워드를 사용함에 있어 사용자의 패스워드와 같은 비밀정보가 외부로 노출되어 악의의 제3자가 그 사용자가 이용하는 서버급 전산시스템에 접근하므로써 해킹 및 시스템 손괴를 야기시킬 수 있는 가능성 있다. 이에 대한 방지 연구와 실용화의 하나로서 제안된 것이 일회용 패스워드이며 IETF의 인터넷을 중심으로 표준화가 추진되어 정보통신 분야에 널리 보급되고 있다. 그러나, 국내에는 인증 정보보호 서비스에 대한 수요 증가와 폭발적인 인터넷 이용 증가에도 불구하고 이의 표준화가 정립되지 못하여 이를 이용하는 사용자는 이용비용의 증가, 이용방식의 비호환성 등으로 많은 불편을 감수해야 하며 이는 국가 전체의 경쟁력 저하의 요인으로 작용할 것이다.

이를 위하여 본 논문에서는 패스워드 누출방지 소프트웨어(SecuPass), IETF의 일회용 패스워드 시스템 등에 관한 선행 연구결과를 분석하여 안전성과 실용성 측면에서 국내 표준화를 위한 일회용 패스워드의 방식 및 기능적 요구조건을 도출하였다. 특히, 사용자 인증만을 요구하는 일반적인 일회용 패스워드 시스템에 대하여는 안전한 해쉬함수로서 HAS-160을 수용하고 공개키 기반의 인증 교환 방식과 SecuPass II에서의 난수이용 방법을 S/KEY 와 RFC 2289 방식의 일회용 패스워드 생성 알고리즘에 적용하는 씨드대신 사용하므로써 스푸핑 공격과 레이스 공격에 대응할 수 있도록 하였다.

[참고문헌]

[1] ITU-T, "Information Technology - OSI - Security Frameworks for Open Systems ; Authentication Framework", ITU-T X.811, 1995

[2] ISO/IEC, "Information Technology - OSI - Security Frameworks for Open Systems ; Authentication Framework", ISO/IEC 10181-2, 1996

[3] Hughes, L j, "Actually Useful Internet Security Techniques, Chapter 3 and 4", pp. 67- 125, New Riders, 1995

[4] 이홍섭 외, "전산망에서의 패스워드 누출방지 기술 개발 보고서", 한국정보보호센터, 1997.12

[5] Haller, N, et. al.,"The S/KEY One-Time Password System", IETF RFC 1760, Feb. 1995

[6] Haller, N, et. al.,"A One-Time Password System", IETF RFC 2289, Feb. 1998

[7] Kohl, J. and Newman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, Sept. 1993.

[8] 한국정보통신기술협회, "해쉬알고리즘 표준(안)", 98.3

[9] 임채훈 외,"한국형 디지털 서명을 위한 해쉬함수의 제안 : PMD-160", http://crypt.future.co.kr /~chlim/pub/has160.ps