암호 알고리즘을 이용한 일회용 패스워드 메커니즘 개발

박정식*, 김영길*, 백규태*, 백기영**, 류재철**

*한국통신 멀티미디어연구소, **충남대학교 컴퓨터과학과

Development of a One-time Password Mechanism using Cryptographic Algorithms

Jung-Sik Park*, Young-Gil Kim*, Gyu-Tae Baek*, Ki-Young Baek**, Jae-Cheol Ryou**

*Korea Telecom Multimedia Lab.
** Dept. of Computer Science, Chungnam National Univ.

불법적인 컴퓨터 사용이나 허가되지 않은 자료 접근 등, 컴퓨터 시스템에 대한 외부의 위협 문제들을 해결하기 위해서 사용자 인증 메커니즘(user authentication mechanism)과 같은 보호 메커니즘이 개발되고 있다. 그러나, 기술이 발전해 갈수록 패스워드와 같이 단순한 인증 메커니즘만으로는 이러한 문제를 해결하는 것이 불가능해지게 되었다. 또한, 이러한 문제를 해결하기 위해 지금까지 개발된 일회용 패스워드 메커니즘들도 대부분 특정 하드웨어 장치를 사용해야 하는 방식으로 개발됨으로써 개발품의 단가를 높이고 이를 적용한 시스템의 유지 보수를 까다롭게 해왔다.

이 논문에서는 이러한 점들을 고려하여 기존 일회용 패스워드 생성 메커니즘을 개선함으로써 스마트 카드와 같은 범용 저장 장치를 이용하여 운영하기 적합하도록 인증 메커니즘을 설계하고 구현하였다.

1. 서 론

특수한 분야에만 이용되던 컴퓨터의 사용이 점차 사무환경으로 확산되면서 이제는 사무환경에서 컴퓨터는 중요한 위치를 차지하게 되었고, 컴퓨터 없는 사무환경은 생각 조차 할 수 없는 시대에 이르렀다. 이에 따라 예전에 종이에 작성되던 모든 문서가 이제는 컴퓨터를 이용하여 작성되어 컴퓨터의 하드 디스크에 저장되고 있다. 종이에 문서를 작성하던 시절에는 문서를 금고에 보관해 놓고 문서에 접근할 수 있는 몇몇 사람들만이 열쇠를 가지고 문서에 접근하여 열람하거나 수정할 수 있는 권한을 갖게 하였으나, 모든 문서들이 컴퓨터에 저장되면서부터 컴퓨터에 저장된 문서에 접근할 수 있는 권한을 설정하는 일이 문제가 되기 시작했다.

이러한 문제들을 해결하기 위하여 컴퓨터 사용자의 정당성을 확인하는 사용자 인증 메커니즘과 같은 보호 메커니즘이 개발되고 있다. 특히, 사용자 인증 메커니즘은 컴퓨터 시스템에 대한 접근 제어를 목적으로 하는 기반 기술로서 패스워드 메커니즘 같은 것이 이에 해당한다.

그러나, 기술이 발전해 갈수록 패스워드와 같이 단순한 인증 메커니즘만으로는 컴퓨터 시스템 자원의 불법적인 사용을 막는 것이 불가능해지게 되었다. 기존의 패스워드 방식에서는 인증 요구자가 검증자로 인증을 요구할 때에 항상 고정된 값의 패스워드를 인증 검증자로 전송함으로써 네트워크 상에서 패스워드 값이 노출될 위험성이 있어 보안상 취약하게 되기 때문이다. 그래서 네트워크 상에서 기존 방식의 패스워드 노출 취약점 등을 보완하기 위해 일회용 패스워드 방식이 제시되었다. 일회용 패스

워드 방식은 기존 방식과는 달리 네트워크상에서 검증자로 전달되는 인증 요구자의 인증용 패스워드가 매번 다른 값을 갖도록 해줌으로써 패스워드 노출 취약성과 같은 문제점을 개선하였다.

그렇지만 지금까지 개발된 일회용 패스워드 방식은 특정 하드웨어를 기반으로 개발되었기 때문에, 일회용 패스워드 방식의 인증을 사용하려면 하드웨어를 구입해야 하는 추가적인 부담이 따르게 되고, 이러한 인증 방식을 지원하기 위해서는 전체 컴퓨터 환경을 재구성해야 함으로써 시스템의 구성을 복잡하게 하고 다양한 방면에 적용하기 힘들다는 단점이 존재한다.

최근에는 카드 내에 프로세서 칩(processor chip)을 장착한 스마트 카드(smart card)를 인증 메커니즘 구현에 사용하고 있다. 스마트 카드에 의해 제공되는 인증 메커니즘은 사용자 인증과 개체 인증 같은 상호 인증 기능을 포함하고 있다[1][2][3][4]. 그러나, 스마트 카드는 보호 수단으로서의 기본적인 기능만을 제공하기 때문에 스마트 카드 기반의 보호 체계 개발 시에는 시스템의 보다 안전한 운영과 보호를 위해서 전체 시스템의 운용 환경을 고려하여 이에 적합한 보안 설계와 대책이 고려돼야 한다. 이를 위해서 스마트 카드, 카드 사용자, 컴퓨터 시스템 간의 관계를 고려하여 보호 체계를 적절히 구현해야 한다.[5]

그러므로, 이 논문에서는 이러한 스마트 카드의 특성 및 전체 시스템의 운용 환경 등을 고려하여, 보호 체계 구축에 쉽게 적용할 수 있는 스마트 카드에 적합하며 특정 하드웨어에 제한되지 않고 다양한 환경에 적용할 수 있는 일회용 패스워드 인증 메커니즘을 설계하게 되었다.

여기에서 제안한 인증 메커니즘의 활용분야는 인증이 필요한 어느 곳이나 사용될 수 있는데, 특히 현재 인터넷에서 대두되고 있는 LDAP(Lightweight Directory Access Protocol)과 같은 경우 LDAP 서버에서는 접속하는 사용자를 인증하여 사용자에 해당하는 권한을 허가해 주는 역할을 하고 있으며, LDAP 과 같은 경우 서버에 접근하는 사용자 인증만이 필요하며, 서버에 대한 인증이 필요 없어 제안한 인증 메커니즘에 적당하다고 생각한다.

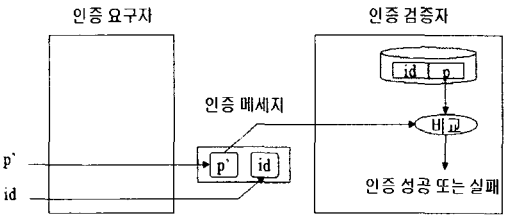
또한, LDAP 버전 3 에서는 SASL(Simple Authentication and Security Layer) 메커니즘을 이용하여 다양한 인증 방법들이 사용될 수 있게 하였는데, 기존의 패스워드 방식을 비롯하여 커버로스와 같은 널리 알려진 인증방법들이 사용되게 된다. 따라서 패스워드와 같이 취약한 인증방법이나 커버로스와 같이 구성이 복잡한 인증 방법외에 이 논문에서 설계된 인증 방법을 사용하여 LDAP 서버와 사용자간의 인증을 수행할 수 있다.

2. 패스워드 인증 메커니즘 고찰

많은 컴퓨터 시스템에서 사용자 인증 방식으로 패스워드 메커니즘을 사용하고 있다[6][7][8][9][10]. 패스워드 메커니즘은 [그림. 1]에서 보는 바와 같이 인증 검증자가 패스워드를 사용하여 인증 요구자를 인증하는 메커니즘이다. 인증 요구자는 식별자 id 와 패스워드 p'을 이용하여 인증 메시지를 생성하고, 네트워크를 통해 인증 메시지를 인증 검증자에게 보낸다. 인증 검증자는 인증 요구자가 보내온 인증 메시지에 포함된 패스워드 p'을 자신이 저장하고 있는 패스워드 p 와 비교하여 같으면 인증 요구자를 인증해준다.

그러나, 위와 같은 패스워드 인증 메커니즘은 다음과 같은 문제점들을 갖고 있다[11].

- 패스워드 노출 : 네트워크를 통해 패스워드가 보호되지 않은 상태로 전달됨
- 패스워드 재연 : 일정한 값을 갖는 패스워드가 반복적으로 사용됨
- 검증자 침해 : 인증 검증자가 저장 관리하고 있는 인증 정보가 노출되었을 때 이것을 분석함으로써 패스워드를 추측할 수 있음

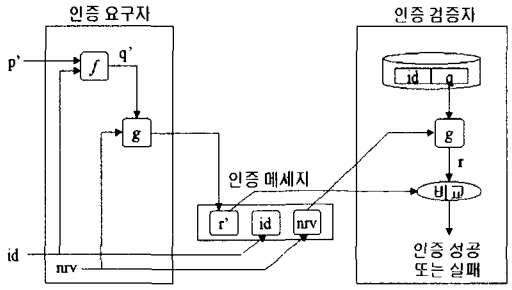


[그림. 1] 기본적인 패스워드 인증 메커니즘

기본적인 패스워드 인증 메커니즘이 갖고 있는 이러한 문제점들을 해결하기 위하여 [그림. 2] 및 [그림. 3]과 같이 개선되거나 변형된

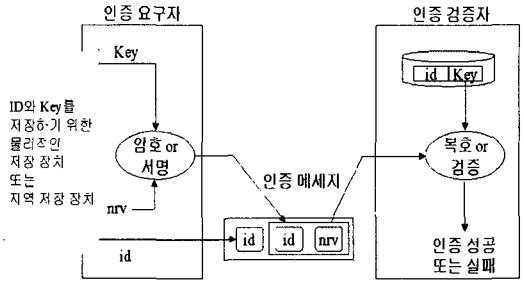
인증 메커니즘이 제시되었다.

[그림. 2]에 제시된 인증 메커니즘은 일방향 함수를 사용하여 구성한 패스워드 메커니즘으로써 인증 요구자로부터 인증 검증자로 전달되는 패스워드는 일방향 함수 f를 수행한 값을 사용하도록 하여 패스워드가 노출되는 것을 막았다. 또한, 인증 검증자에 의해 저장 관리되는 사용자 인증 정보도 패스워드 값 자체를 사용하는 것이 아니라 패스워드의 일방향 함수 g를 수행한 값을 사용함으로써 검증자 침해를 방지한다. 마지막으로, 인증 요구자가 인증 메시지를 구성할 때에는 반복되지 않는 값 nrv를 사용하여 메시지를 구성하게 함으로써 패스워드 재연을 방지한다.



[그림. 2] 검증자 침해 및 패스워드 노출/재연 방지를 고려한 패스워드 인증 메커니즘

[그림. 3]에 제시된 인증 메커니즘은, 패스워드 인증 메커니즘은 아니지만, 패스워드와 같은 역할을 하는 변형된 개념의 키와 암호 알고리즘을 사용하여 구성한 메커니즘으로써 인증 요구자로부터 인증 검증자로 전달되는 인증 메시지를 암호 또는 서명하여 인증을 수행하게 된다. 여기에서는 암호 또는 서명에 사용되는 키가 노출되지 않기 때문에 키 내지는 패스워드 노출과 같은 문제점은 없다. 반면에 인증 검증자 측에서 복호 또는 검증에 사용되는 키 값을 저장해 두기 때문에 검증자 침해에 대비하여 키 값 자체를 다른 암호 알고리즘 또는 다른 키 값을 이용하여 보호함으로써 침해를 방지해야 한다. 여기에서도, 앞서의 패스워드 인증 메커니즘에서와 마찬가지로, 인증 요구자가 인증 메시지를 구성할 때에는 반복되지 않는 값 nrv를 사용하여 메시지를 구성하게 함으로써 인증 메시지 재연을 방지한다.



[그림. 3] 간단한 암호화 기반의 인증 메커니즘

일반적으로, 인증 메커니즘의 동작 또는 성립 원칙을 살펴 보면, 대부분의 인증 메커니즘에 적용되는 두 가지 원칙이 다음과 같다는 것을 알 수 있다[11][12][13][14].

- 알고 있는 것(패스워드, ...)
- 소유하고 있는 것(신용카드, 신분증, ...)

앞에서 기술한 [그림. 2]와 같은 패스워드 인증 메커니즘은 "알고 있는 것"을 이용하여 사용자를 인증하는 방식으로 "소유하고 있는 것"을 인증에 포함하지 않았다. 인증 메커니즘이 "소유하고 있는 것"을 통해서 사용자를 인증하도록 지원하기 위해서는 물리적인 저장 장치(이하 토큰)를 사용해야만 한다. 반면에, [그림. 3]과 같은 암호화 기반의 인증 메커니즘에서는 물리적인 토큰에 해당하는 "소유하고 있는 것"을 인증에 포함하고는 있으나, 패스워드와 같이 "알고 있는 것"을 포함하고 있지 않다. 그러므로, 암호 알고리즘에 사용되는 키와 더불어 추가적으로 사용자 패스워드를 기억하고 비교하는 능력이 있는 스마트 카드와 같은 저장 장치를 물리적인 토큰으로 사용해야만 한다.

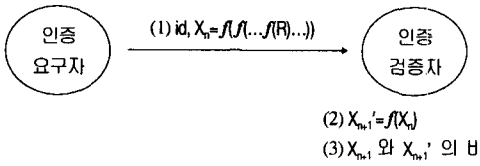
일반적인 방식으로는, 사용자가 패스워드를 사용하여 먼저 자신을 인증하고, 사용자를 대신하는 장비는 암호 알고리즘을 사용하여 궁극적인 인증 검증자에게 사용자를 인증하게 한다. 실제로 이와 같은 방식은 여러 가지 조합으로 구현되지만, 중요한 것은 사용자가 "알고 있는" 패스워드와 사용자가 "소유하고 있는" 스마트 카드를 사용한다는 것이다.

그러므로, 이 논문에서도 앞서 살펴 본 패스워드 메커니즘과 스마트 카드와 같은 물리적인 토큰을 사용하는 암호 기반의 인증 메커니즘을 사용함으로써 "알고 있는 것"과 "소유하고 있는 것"을 통한 사용자 인증이 이루어지도

록 인증 메커니즘을 구성하였다.

3. 일회용 패스워드 인증 메커니즘 설계

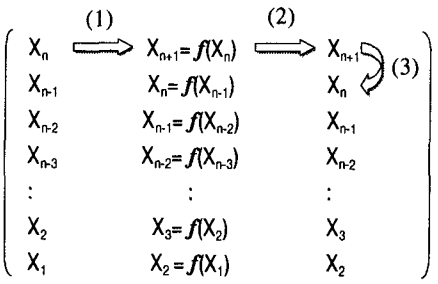
기존에 제안된 S/KEY 일회용 패스워드 생성 메커니즘은 [그림. 4]에서 나타난 바와 같이 일방향 함수를 사용하여 패스워드를 생성하게 되어 있다. 여기에서는 맨 처음 난수 R 값을 생성하고 이 값에 대해 $X_{n+1}=f(f(\ldots f(f(R))\ldots))$ 이 되도록 일방향 함수 f 를 n+1 번 수행하여 X_{n+1} 을 구한다. 그리고, R 과 X_{n+1} 을 각각 인증 요구자와 검증자에게 시스템 최초 설정 시 전달하여 저장해 둔다[10].



[그림. 4] 일방향 함수를 사용한 기존 일회용 패스워드 메커니즘

이렇게 설정된 상태에서 인증 요구자가 검증자에게 인증을 받을 필요가 생기면, (1)에서처럼 자신의 사용자 번호인 *id* 와 일방향 함수 f 를 n 번 수행한 X_n을 인증 정보로 인증 검증자에게 전달하게 된다. (2)에서 인증 검증자는 전달 받은 인증 정보의 X_n을 이용하여 일방향 함수 f 를 1 회 더 계산함으로써 $X_{n+1}'=f(X_n)$ 을 구한다. (3)에서 인증 검증자는 앞서 계산된 X_{n+1}'값을 시스템 설정 시 저장한 X_{n+1} 값과 비교하여 같으면 인증 요구자를 인증하게 된다. 인증이 성공적으로 이뤄지게 되면 인증 검증자는 X_n 을 다시 저장한다.

[그림. 5]는 이러한 S/KEY 일회용 패스워드 생성 메커니즘에서 인증 정보를 생성하기 위해 인증 요구자와 검증자가 각각 일방향 함수를 어떻게 사용하는지 구체적으로 보여준다. 최초 시스템 설정 시에 인증 요구자와 검증자는 R 값에 대해서 $f(f(\ldots f(f(R))\ldots))$ 이 되도록 일방향 함수 f 를 n 번, n+1 번 수행하여 각각 X_n, X_{n+1} 을 갖고 있게 된다. 이때 R 값은 인증 요구자가 알고 있는 값으로 인증 요구자는 이 값으로부터 일방향 함수 f 를 수행하여 임의의 n 에 대하여 X_n을 계산하는 것이 가능하지만, 인증 검증자는 요구자가 전달해준 X_n에 대해 일방향 함수 f 를 수행하여 $X_{n+1}=f(X_n)$ 을 계산하는 것만 가능하다.



[그림. 5] 기존 일회용 패스워드 메커니즘에서의 일방향 함수 운용 개념

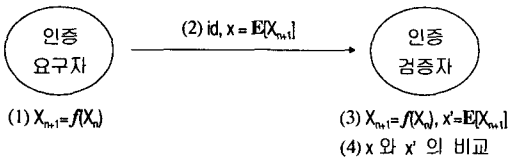
(1)에서 인증 요구자가 R 값에 대해 일방향 함수 f 를 n 번 수행하여 X_n 을 인증 검증자에게 전달한다. (2)에서 인증 검증자는 전달 받은 X_n에 대해 일방향 함수 f 를 수행하여 $X_{n+1}=f(X_n)$ 을 계산한다. 그 다음, (3)에서 시스템 설정 시에 저장되어 있던 X_{n+1} 과 (2)의 계산 결과를 비교하여 같으면 인증 요구자를 인증하게 되고 요구자로부터 전달 받은 X_n 을 저장하여 다음 인증 과정에서 사용한다.
다음 인증 과정이 수행되면, (1)에서 인증 요구자는 다시 R 값에 대해 일방향 함수 f 를 n-1 번 수행하여 X_{n-1} 을 인증 검증자에게 전달한다. (2)에서 인증 검증자는 전달 받은 X_{n-1}에 대해 일방향 함수 f 를 수행하여 $X_n=f(X_{n-1})$을 계산한다. 그 다음, (3)에서 저장되어 있던 X_n 과 (2)의 계산 결과를 비교하여 같으면 인증 요구자를 인증하게 되고 요구자로부터 전달 받은 X_{n-1} 을 저장하여 다음 인증 과정에서 사용한다. 이와 같은 과정이 X_1 일 때까지 반복된다.

그렇지만, 위와 같은 일회용 패스워드 생성 메커니즘에는 다음과 같은 문제점이 존재하게 된다.

① 사용 횟수에 대한 제한이 있다.
② 인증 요구자는 인증에 사용되는 일회용 패스워드를 미리 생성하거나 매번 필요한 횟수만큼 함수를 수행하여 계산해야 한다.
③ 메커니즘의 안전성이 일방향 함수의 특성에 의존한다.

④ 인증 요구자에 대한 인증만을 수행함으로써 인증 검증자와 요구자간에 비밀통신을 위해서는 다시 키 분배 과정을 수행해야 한다.

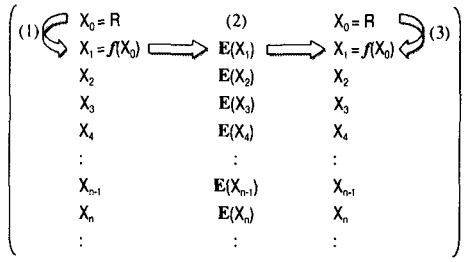
이 논문에서는 이러한 문제점을 갖는 S/KEY 방식을 개선하여 일방향 함수(one-way function) 및 암호 알고리즘(encryption algorithms)을 이용한 일회용 패스워드(one-time password) 메커니즘을 제안하였다. [그림. 6]은 이 논문에서 제안한 일회용 패스워드 생성 메커니즘에 대해 나타내고 있다.



[그림. 6] 일방향 함수 및 암호 알고리즘을 사용하여 제안한 일회용 패스워드 메커니즘

제안한 메커니즘에서는 맨 처음 시스템 설정 시에 난수 R 값을 생성하여 이 값을 각각 인증 요구자와 검증자에 저장해 둔다. 이 상태에서 인증 요구자가 검증자로부터 인증을 받기 위해 (1)에서 $X_{n+1}=f(X_n)$를 계산한다. 그리고, (2)에서 자신의 사용자 번호인 id 와 $E[X_{n+1}]$를 인증 정보로 인증 검증자에게 전달하게 된다. (여기에 사용된 암호 알고리즘 **E** 는 임의의 키를 사용하여 데이터 암호화 기능을 수행한다.) (3)에서 인증 검증자도 (1)에서와 마찬가지로 $X_{n+1}=f(X_n)$를 계산하고 이를 암호화하여 $E[X_{n+1}]$을 구한다. (4)에서 인증 검증자는 (2)에서 보내온 x 값과 (3)에서 구한 x'값을 비교하여 같으면 인증 요구자를 인증하게 된다. 인증이 성공적으로 이뤄지게 되면 인증 검증자는 X_{n+1} 값을 저장해 둔다.

[그림. 7]은 제안한 일회용 패스워드 생성 메커니즘에서 인증 정보를 생성하기 위해 인증 요구자와 검증자가 각각 일방향 함수와 암호 알고리즘을 어떻게 사용하는지 구체적으로 보여준다. 여기에서는 최초 시스템 설정 시에 난수 R 값을 각각 인증 요구자와 검증자에 저장한다. 그러므로, 인증 요구자와 검증자는 $X_0=R$ 로부터 $X_{n+1}=f(X_n)$ 을 구하는 것이 가능하며, 여기에서 생성되는 일련의 값들은 외부로 노출되지 않도록 안전하게 각자가 잘 보관하여 관리해야 한다.



[그림. 7] 제안한 일회용 패스워드 메커니즘에서의 일방향 함수 및 암호 알고리즘의 운용 개념

(1)에서 인증 요구자는 $X_0=R$ 값에 대해 일방향 함수 f 를 수행하여 $X_1=f(X_0)$ 을 생성한다. (2)에서 인증 요구자는 X_1 값을 **E** 로 암호화하여 $E[X_1]$ 값을 인증 검증자로 전달한다. (3)에서 인증 검증자는 자신이 저장하고 있는 $X_0=R$ 로부터 $X_1=f(X_0)$을 구하여 이를 **E** 로 암호화하여 $E[X_1]$ 값을 구한 뒤에 (2)에서 전달 받은 값과 비교하게 되고, 값이 같으면 인증 요구자를 인증하게 된다. 새로 계산된 $X_1=f(X_0)$ 값은 인증 요구자와 검증자가 각각 저장하여 다음 인증 과정에서 사용된다.

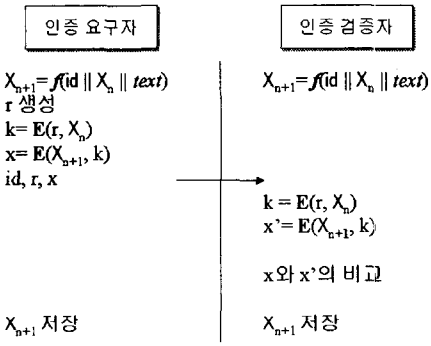
다시, 다음 인증 과정이 수행되면, (1)에서 인증 요구자는 저장된 X_1 값에 대해 일방향 함수 f 를 수행하여 $X_2=f(X_1)$ 을 생성한다. (2)에서 인증 요구자는 X_2 값을 **E** 로 암호화 하여 $E[X_2]$ 값을 인증 검증자로 전달한다. (3)에서 인증 검증자도 자신이 저장하고 있는 X_1 값으로부터 $X_2=f(X_1)$을 구하고 이를 **E** 로 암호화 하여 $E[X_2]$ 값을 구한 뒤에 (2)에서 전달 받은 값과 비교하게 되고, 값이 같으면 인증 요구자를 인증하게 된다. 인증 요구자와 검증자는 새로 계산된 $X_2=f(X_1)$ 값을 저장하여 다음 인증 과정에서 사용한다. 이와 같은 과정이 계속 반복된다.

일방향 함수와 암호 알고리즘을 적용하여 제안한 일회용 패스워드 메커니즘을 이용하여 인증 프로토콜을 수행할 때에 인증 요구자 및 검증자가 수행해야 하는 연산 절차와 인증 정보 전달 과정이 [그림. 8]에 나타나 있다.

인증 요구자는 자신의 사용자 번호인 id, 저장되어 있는 X_n, 그리고 기타 사용자관련 정보인 *text* 를 이용하여 $X_{n+1}=f(\mathrm{id} \| X_n \| \mathit{text})$를 계산한다. 다음으로 인증 정보를 암호화하기 위한 키 생성을 위해 난수 r 을 생성하여 암호 알고

리즘 E를 통해 키 $k= E(r, X_n)$를 계산한다. 그리고, 앞에서 계산된 X_{n+1}, k 로부터 $x= E(X_{n+1}, k)$를 계산한다. 마지막 단계로 인증 요구자는 검증자에게 id, r, x 를 전달하게 된다.

인증 검증자는 id, r, x 를 전달 받은 뒤에 먼저 $X_{n+1}= f(id \| X_n \| text)$를 계산한다. 그리고, $k = E(r, X_n)$을 계산하여 인증 정보 암호화에 사용된 키 값을 계산한다. 그 다음에 앞서 계산된 k, X_n을 이용하여 $x'= E(X_{n+1}, k)$를 계산한다. 여기에서 계산된 x'을 인증 요구자가 보내온 x 와 비교하여 같으면 요구자를 인증하게 된다. 인증이 성공적으로 이뤄지면 인증 요구자와 검증자는 각각 X_{n+1}를 저장하여 다음 인증 프로토콜이 수행될 때 사용하게 된다.



[그림. 8] 제안한 일회용 패스워드 메커니즘을 이용한 인증 프로토콜

위에서 제안한 일회용 패스워드 메커니즘에서는 일회용 패스워드가 $X_0=R$, $X_1= f(X_0)$, $X_2= f(X_1)$, ... 형태로 생성되기 때문에 사용 횟수에 대한 제한이 없을 뿐 아니라 이를 미리 생성하거나 매번 필요한 횟수만큼 반복하여 함수를 계산하지 않아도 된다. 또한, 위 메커니즘은 메커니즘의 안전성이 일방향 함수 및 암호 알고리즘의 특성에 의존한다. 그리고 인증 과정 수행이 완료된 후에는 인증 검증자와 요구자간에 비밀 통신을 위한 통신 세션 키 분배가 이루어 진다.

4. 결론

이 논문에서 제안하는 암호알고리즘을 이용한 일회용 패스워드 메커니즘은 기존 패스워드나 암호 인증 메커니즘에 비해 암호 알고리즘 사용에 따른 인증요구자인 클라이언트상의 사용자 인증용 키 관리가 용이하고 클라이언트와 인증검증자인 서버간에 인증 및 키 분배를 실시할 수 있게 해준다. 그 외 다른 항목에 대한 기존의 S/KEY 인증 메커니즘과 제안한 일회용 패스워드 메커니즘간의 비교 결과는 표 1 과 같다.

표 1. 알고리즘 비교 분석

	S/KEY 알고리즘	개발 알고리즘
사용 횟수	r.	∞
키 저장	r.	1
안전성	One-way Function에 의존	One-way Function과 암호 알고리즘에 의존
키 분배	×	o

제안한 인증메커니즘에서는 S/KEY 알고리즘의 2가지 단점, 즉 일방향함수를 이용하여 사전에 n개의 키를 생성하여 사용함으로써 발생하는 사용 횟수 제한점과 키를 저장해 놓아야 한다는 점을 해결하였으며, 암호 알고리즘을 사용함으로써 안전성을 향상시켰고 인증과 더불어 서버와 클라이언트간에 세션키의 분배도 가능하게 했다. 기존의 인증 메커니즘을 이용할 경우 인증이 끝난 후에 상호간에 안전한 암호화 통신을 위해 세션키를 분배하는 부가적인 과정이 추가된다.

또한, 스마트 카드를 기반으로 한 일회용 패스워드 인증 메커니즘을 사용함으로써 기존 패스워드 메커니즘을 채택하고 있는 모든 응용과 개념적으로 호환성을 유지할 수가 있고, 사용자 인증 과정이 보다 더 안전하게 이뤄지고 인증 정보의 보호 및 관리가 용이하게 이뤄질 수 있게 하였다. 그리고, 클라이언트에서 서버로의 인증 과정이 일방향성을 갖게 함으로써 네트워크 상에서 트래픽 부담을 줄일 수 있는 형태로 개발했다.

지금까지 일회용 패스워드 메커니즘을 적용하여 개발된 제품들은 대부분 일회용 패스워드 생성에 특정 하드웨어 장치를 사용해야 하는 방식으로 개발됨으로써 개발품의 단가를 높이고 이를 적용한 시스템의 유지 보수를 까다롭게 해왔다. 그러나 제안한 인증 메커니즘은 특정 하드웨어 장치에 제한되어 개발되지 않았기 때문에 인증이 필요한 다양한 분야에 적용이

가능하다. 예를 들어 LDAP 버전 3 에서 LDAP 서버와 클라이언트 사이의 인증을 위해 다양한 인증 메커니즘을 사용할 수 있게 해 주는 SASL 를 이용하여, 제안한 인증 메커니즘을 LDAP 서버와 클라이언트 사이의 인증에 적용하는 것이 가능하다.

이 논문에서는 이러한 점들을 고려하여 기존 일회용 패스워드 생성 메커니즘을 개선하여 여러 분야에 적용이 가능하도록 하였으며, 스마트 카드와 같은 범용 저장 장치에 적합하도록 인증 메커니즘을 설계하였다.

참고 문헌

[1] Gemplus, *GPS120 Application Programmer's Guide*, 12. 1993.

[2] Gemplus, *GPS120 Reference Guide*, 06. 1993.

[3] Gemplus, *GPS120 User's Guide*, 09. 1993.

[4] 현대전자(주), *HYUNDAI COS(HYC 201/802) User's Guide*, 1997.

[5] 신진원, 권태경, 송주석, "스마트 카드 시스템의 보안 기능 분석 및 설계에 관한 고찰", 한국통신정보보호학회 종합학술발표회 논문집, Vol 5, No. 1, pp265-74, 11. 1995.

[6] "OnceID and Oasis," http://www.softforum.co.kr:4040/product/OnceID_Oasis.html.

[7] "One-Time Passcode Software for User Authentication," http://www.securitydynamics.com/solutions/pro-ducts/sofidata.html.

[8] "SecurID Tokens Datasheet," http://www.securitydynamics.com/solutions/products/tokens.html.

[9] Neil M. Haller and Philip R. Karn, "Description of The S/KEY One-Time Password System," http://www-staff.lboro.ac.uk/~ccgpg/skey.html.

[10] "패스워드 누출방지 기술," http://www.kisa.or.kr/K_tech/exp/netsec/password.html

[11] Warwick Ford, *Computer Communication Security*, Prentice Hall, pp.109-148, 1994

[12] 최용락, 소우영, 이재광, 이임영, *통신망 정보 보호(Network and Internetwork Security Principles and Practice)*, 도서출판 그린, 02. 1996.

[13] Bruce Schneier, *Applied Cryptography*, Second Edition, John Wiley & Sons, 1996.

[14] William Stallings, *Network and Internetwork Security*, Prentice-Hall, 1995.