韓國 OA學會 論文誌
第6卷 第4號, 2001. 12.

2001-6-4-3-12

SHA-1 방식을 이용한 제한된 웹 페이지에 접근하기 위한 서버 독립적인 패스워드 인증 방안

하 창 승*, 조 익 성**

A Server-Independent Password Authentication Method for Access-Controlled Web Pages Using the SHA-1 Algorithm

Chang-Seung Ha*, Ik-Sung Cho**

요 약

본 논문에서는 웹 페이지 운영자가 웹 서버에 패스워드로 보호된 웹 영역을 배치시키는데 편리함을 제공하는 목적으로 새로운 패스워드 인증 방안을 제안한다. 이 방법에 따라 웹 영역은 접근 제한된 웹 페이지가 저장된 웹서버의 보안 디렉토리에 연결된다. 패스워드는 보안 디렉토리의 이름을 생성하는데 사용되며, 자바스크립트 코드는 보안 디렉토리 영역 외부의 인증 웹 페이지에 포함된다. 이 스크립트 코드는 사용자가 입력한 패스워드를 디렉토리 이름으로 변환시키고, 보안 디렉토리 내부의 접근 제한된 웹 페이지를 지시하는 완전한 URL을 형성한다. 따라서, 패스워드를 아는 사용자만이 유효한 URL을 구성할 수 있고, 접근제한된 웹 페이지를 검색할 수 있다. 이 방법에 따라, 웹 페이지 운영자는 서버 독립적인 방식으로 패스워드로 보호된 웹 영역을 배치할 수 있다.

Abstract

A new password authentication method is proposed in this paper for the purpose of providing web page authors the convenience in deploying password-protected Web realms at a web server. According to this method, a web realm is mapped to a secret directory at the web server, in which access-controlled web pages are stored. A password is used to construct the name of the secret directory. A javaScript code is embedded in a sign-in web pageoutside the secret directory, which converts the user-entered password into the directory name and forms a complete URL pointing to an access-controlled web page inside the secret directory. Thus, only users knowing the password can compose a valid URL and retrieve the access-controlled web page. Using this method, web page authors can deploy password-protected web realms in a server-independent manner.

* 동명대학 정보통신계열 조교수
** 동명대학 정보통신계열 전임강사

Ⅰ. 서론

처음 웹서비스가 대두되기 시작하면서 지향한 목표는 정보의 효율적인 공유였다. 또한 이런 원칙에 따라 웹 페이지는 외부에 모두 공개되는 것이 원칙이었다. 이처럼 정보의 효율적인 공유에 초점을 두었기 때문에 웹 상에서의 사용자 인증에 대한 비중은 무척 낮다. 웹에 제공된 자료는 모든 사용자에게 개방되었으며 사용자는 서버에 로그인하는 과정 없이 곧바로 임의의 자원에 접근할 수 있도록 구성되는 것이 원칙이었기 때문이다. 이 영향으로 자원은 하나의 서버에 국한되지 않고 하이퍼텍스트의 링크들을 따라 광범위한 그물망의 형태로 연결될 수 있었다. 이에 반하여, 웹의 기술을 그대로 기업의 정보 시스템 구축에 사용하려는 인트라넷 시스템 혹은 특정그룹 단위의 사용자들에게는 사용자의 인증이 중요한 문제로 대두되었다. 웹 페이지의 접근을 제한하는 가장 단순한 방법은 패스워드를 이용하는 것이며, 패스워드를 아는 사용자만이 접근이 제한된 웹 페이지에 접근할 수 있도록 하는 것이다. 더 기능적인 구현은 패스워드를 담고 있는 웹 페이지를 만드는 것이다.

사용자는 웹 영역에 접근하기 전에 패스워드를 반드시 입력해야 한다. 패스워드가 웹서버에 전달되고 난후, CGI(Common Gateway Interface)와 같은 서버측 프로그램 또는 웹서버 자체에 의해 인증되어야 한다. 비록 CGI 프로그램이 널리 사용되고 있지만, 개인적인 웹 페이지 운영자들은 웹서버에 그들 자신의 CGI 프로그램을 구동시킬 권한이 없을 경우도 있다.

웹 영역에 패스워드를 구성하는 방법은 HTTP 프로토콜에 명시되어 있지 않다. 그것은 웹서버 구현에 달려 있으며, 여러 다른 해결책이 제안되어왔다. 아파치와 같은 가장 유명한 웹서버는 접근 액세스의 "directives"에 의존한다. 그런 "directives"는 웹서버의 관리자에 의해 접근할 수 있는 ".access" 파일이나, 개인 웹페이지 소유자에 의해 소유되는 ".htaccess" 파일에 존재한다[1]. 마이크로소프트 IIS는 개인 소유자들이 공유 파일이 관리되는 것과 같은 방식으로 모든 파일과 디렉토리에 관련된 접근 제한 리스트를 관리할 수 있도록 윈도우 기반의 인터페이스를 제공한다. 또한 웹 운영자의 접근 제한된 웹 페이지를 검색하기에 적합한 사용자는 웹서버 시스템에 등록된 사용자여야만 한다[2].

넷스케이프 엔터프라이즈 서버는 접근 제한된 리스트를 제공하기 위해 LDAP(Lightweigh Directory Access Protocol) 서버를 이용한다[3]. 그것은 관리를 위한 브라우저 기반의 인터페이스를 제공한다. 패스워드 인증을 위해 CGI 프로그램을 구동시킬 수 없는 웹 페이지 운영자들은 패스워드로 보호된 웹 영역을 배치시키기 위해 웹서버에 전적으로 의존한다. 무엇보다도 웹서버 관리자의 도움이 요구된다. 즉 보안 정보(사용자 ID와 패스워드)가 웹서버 관리자에 의해 노출되며, 웹 서버 모델이 변화될 때나 소유자가 다른 웹사이트로 접근 제한된 웹 페이지를 옮기면, 접근제한 리스트와 패스워드는 모두 다시 재구성해야 한다. 이 문제를 해결하기 위해 본 논문에서는 새로운 패스워드 인증 방안을 제안한다. 새로운 패스워드 인증 방안은 웹 서버에 디렉토리 이름을 감추고, 보안정보가 웹서버 관리자에 의해 노출되는 것을 방지하기 위해 클라이언트측 자바 스크립트 코드를 이용하여 디렉토리를 생성시키며, SHA-1 알고리즘을 이용하여 보안 정보를 강화시키는데 목적을 두고 있다. 웹페이지 운영자가 디렉토리 구조를 변화시키지 않는 한, 이 패스워드 인증 방안은 서버 독립적이다. 본 논문의 구성은 다음과 같다. 2장에서는 관련기술로서 웹에서 적용되는 인증 기술과 해쉬 알고리즘인 SHA-1, 웹서버 인증시 문제점을 검토한 후, 3장에서는 이들을 해결할 수 있는 SHA-1 방식을 이용한 패스워드 인증 시스템에 대하여, 4장에서는 실험 결과 및 고찰에 대하여, 5장에서는 결론 및 향후 과제를 제시한다.

Ⅱ. 관련연구

웹 페이지 인증은 공개된 인터넷의 가상공간에서 이루어지기 때문에 전과정의 철저한 보안을 요구하므로 암호화는 필수적이다. 따라서 본장에서는 웹에서 적용되는 인증 기술과 로그인 단계에서 강력한 인증을 수행할 수 있는 SHA-1 해쉬 알고리즘, 웹서버 인증시 문제점을 고찰한다.

1. 웹기반 패스워드 인증 시스템

웹기반 패스워드 인증 시스템은 웹서버 자체에 의한 인증과, CGI와 같은 서버측 프로그램에 의한 인증으로 나눌 수 있다. 웹브라우저가 기본기능 외에 아직 구현되지 않은 여러 자료들을 처리할 때 흔히 플러그인이나 헬퍼(Helper) 응용 프로그램을 사용하는 것을 알 수 있다. 결국에는 웹브라우저와 웹서버의 기본기능 외에 별도의 기능을 추가할 때 웹서버에는 CGI 프로그램을, 웹브라우저에는 플러그인이나 헬퍼 응용 프로그램을 사용하는 것을 알 수 있다[4,5]. 그림 1에서 보는바와 같이 웹브라우저와 웹서버 자체를 통해 이뤄지는 접근 인증 과정을 나타내면 다음과 같이 4단계로 구성된다.

① 웹브라우저가 어떤 자료의 URL을 요청하면, 웹브라우저는 요청 헤더를 발생시켜 웹서버에 전달한다.

② 웹서버는 웹브라우저로 요구된 페이지의 접근 인증이 필요한지 여부를 확인한다.

③ 접근 인증이 필요한 경우 웹서버는 응답 메세지에 401 상태 코드를 삽입하고 웹브라우저는 사용자로부터 ID와 패스워드를 입력받아 인코딩된 스트링을 헤더에 기록한 후 웹서버에게 넘긴다.

④ 웹서버는 사용자 ID와 패스워드를 인증하고, 만약 제시된 패스워드가 유효하다면 200 OK 상태 코드로서 웹브라우저에 접근 제한된 웹페이지를 나타낸다.

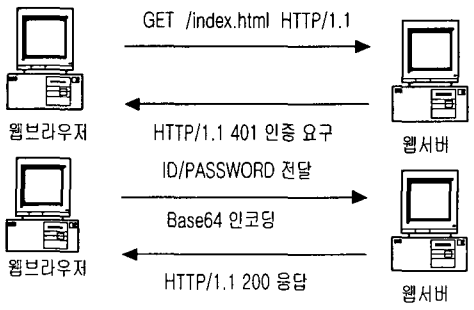

그림 1. 웹서버 패스워드 인증 시스템
Fig. 1 Web-server password authentication system

CGI 패스워드 인증 시스템의 동작절차는 그림 2와 같이 6단계로 구성된다.

① 웹브라우저가 어떤 자료의 URL을 요청하면, 웹브라우저는 요청 헤더를 발생시켜 웹서버에 전달한다.

② 웹서버는 웹브라우저로 요구된 페이지의 접근 인증이 필요한지 여부를 확인한다.

③ 접근 인증이 필요한 경우 웹서버는 응답 메세지에 401 상태 코드를 삽입하고 웹브라우저는 사용자로부터 ID와 패스워드를 입력받아 인코딩된 스트링을 헤더에 기록한 후 웹서버에게 넘긴다.

④ 웹서버는 헤더의 스트링을 임시 파일에 기록하고 보안 CGI를 호출한다.

⑤ CGI는 데이타베이스에 등록된 ID와 패스워드에 관계된 필드를 열어 임시 파일의 내용과 비교해 웹서버를 호출한다.

⑥ 웹서버는 패스워드가 유효하다면 200 성공 상태 코드로서 웹브라우저에 접근 제한된 웹페이지를 나타내며, 패스워드가 유효하지 않다면 401 코드를 재전송한다.

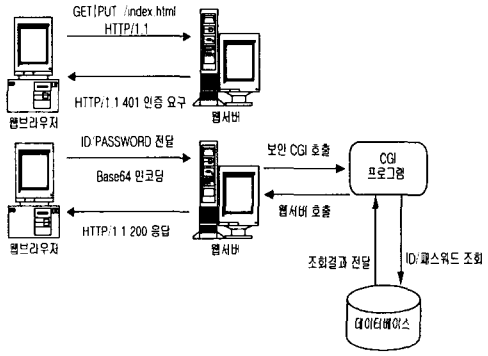

그림 2. CGI 패스워드 인증 시스템
Fig. 2 CGI password authentication system

CGI 프로그램은 TCP/IP 전송 프로토콜의 상위 계층인 응용 프로토콜 계층에서 웹서버와 메시지를 송수신하게 되므로, 그 메시지 처리에 있어 CGI 프로그램과 웹서버간의 오버헤드가 많이 발생한다. 또한 웹서버에서는 보통 하이퍼 미디어 데이터를 다루어주는 디렉토리와 CGI 프로그램을 다루어주는 디렉토리로 나누어진다. 그리고 이 디렉토리들에 대해서 별도의 보안 옵션을 주어 요청할 때 사용자 인증을 받게 할 경우 보안 문제를 발생하기 쉽다[6,7].

2. SHA-1(Secure Hash Algorithm-1)

현재 많은 주목을 받고 있는 메시지 인증 코드에 대한 변형은 단방향 해쉬 함수이다. 메시지 인증 코드에서와 같이 해쉬 함수는 다양한 크기의 입력 메시지를 받아 메시지 다이제스트(Digest)라고 하는 고정된 크기의 해쉬

코드 h(M)을 만든다. 해쉬 코드는 메시지에 대한 모든 비트들의 함수이고, 디지털 서명의 실용성 향상을 위한 서명문 압축과 에러 탐색 능력 그리고 전송 정보의 무결성 확보를 위해 사용한다. 즉, 메시지에 있어서 하나의 비트 또는 비트들의 변화는 해쉬 코드의 변화를 가져오게 된다. 그리고 해쉬값 H 는 다음과 같이 해쉬 함수(h)에 의해서 만들어진다.

H=h(M)

위 식에서 M은 가변길이의 메시지이고, h(M)은 고정길이의 해쉬값이다. 이 해쉬값은 메시지가 정확한 것으로 판단될 때 송신측에서 메시지를 추가한다. 그리고 수신측은 해쉬값을 재 계산함으로써 그 메시지를 인증하게 되는 과정으로 구성되어 있다[8,9]. 이러한 해쉬 함수는 여러 가지 방식이 제안되었으나 MD4의 안전성 문제를 해결하기 위해 미 상무성 표준국(NIST:National Institute of Standard and Technology)이 SHA-1을 개발하여 1993년 FIPS(Federal Information Processing Standard) PUB 180으로 공포하였는데, 이 SHA-1은 MD(Message Digest) 4를 기반으로 하였기 때문에 MD4와 유사하게 설계되었다[10].

SHA-1은 512비트 미만의 길이를 갖는 서명문을 입력으로 하여 160비트의 해쉬값을 출력시키는 함수로서, 512비트 단위로 동작되도록 구성되었다. 즉 MD5의 동작과정과 유사하게 SHA-1에서도 해쉬를 하기 전에 서명문을 512비트의 배수로 만드는 과정을 거치게 된다. 따라서 그림 3에서 보는바와 같이 패딩된 서명문은 L× 512비트로 원래의 서명문에 패딩 비트와 서명문 길이를 표시하는 64비트를 포함하게 된다.

SHA는 5개의 32비트 레지스터(A,B,C,D,E)로 구성된 160비트 버퍼값과 패딩된 서명문의 512비트 블록을 80단계로 구성된 모듈(Hsha)에 입력하여 160비트의 메시지 다이제스트를 생성한다. 그림 4는 SHA의 메시지 다이제스트 생성과정을 보여주고 있다.

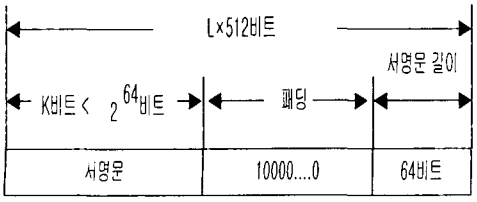

그림 3. SHA의 패딩된 서명문 형식
Fig. 3 A form of padded plain text in SHA

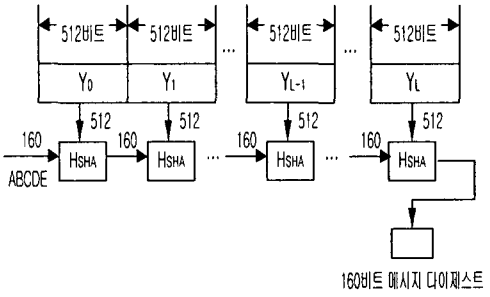

그림 4. SHA의 메시지 다이제스트 생성 과정
Fig. 4 Message digest generation procedure of SHA

3. 웹서버 보안취약점

웹은 여러 가지 웹 관련 서비스, 구성요소, 다른 서비스와의 연계 등을 통해 발생되는 복합적인 보안취약점을 가지고 있다. 특히 웹서버 구현상의 취약점, CGI 관련 취약점, 그리고 웹서버 구성상의 취약점으로 구분할 수 있다. 웹서버 구현상의 취약점은 공개용 웹서버 또는 상용 웹서버의 구현상의 문제로 인하여 보안취약점이 존재하는 경우다. CGI 관련 취약점은 외부의 사용자에게 호스트의 정보를 보여주는 취약점과 사용자 입력 양식(form)을 통해서 임의의 명령을 수행할 수 있는 취약점이 존재하는 경우가 있다. 그리고, 웹서버 구성상의 취약점은 웹서버 구성의 잘못으로 인한 파일 접근 권한 획득, 디렉토리 내용 리스팅, 심볼릭 링크등의 취약점을 유발할 수 있다[11,12]. 특히 웹브라우저와 웹서버 사이에 사용자 ID와 패스워드와 같은 보안 정보는 아무런 암호화 없이 전송되어 진다. 물론 100% 평문 그대로 전송되는 것은 아니며, 사용자 ID와 패스워드를 ASCII 에서 Base64 형식으로 인코딩하여 전송한다. 따라서 이런 보안적인 면을 최소화하기 위해서는 사용자 ID와 패스워드를 메세지 다이제스트 인증을 이용하여 파라미터를 덧붙여 전송하게 되면, 기본인증 방법보다는 안전하게 전송할 수가 있다.

Ⅲ. SHA-1 방식을 이용한 패스워드 인증 시스템

1. 전체 시스템 구성

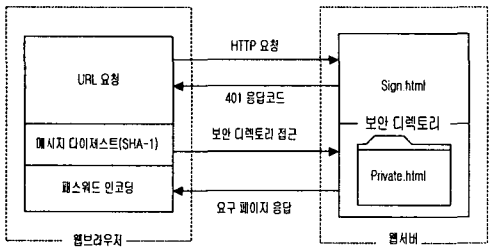

그림 5. 전체 시스템 구성
Fig. 5 System Configuration

앞서 Ⅱ장 전반에 걸쳐 설명했던 웹서버 자체상의 문제점과, 웹서버를 확장한 CGI의 문제점을 극복하기 위해 본 논문에서 제안한 전체 시스템 구성도는 그림 5와 같다.

본 논문에서는 보안정보의 안전한 전송을 위해 MD5 보다는 다소 느리지만, 160-bit 의 메시지 요약들이 폭력적 충돌 및 도치 공격을 받을 때, 좀더 안전하게 지켜주는 장점을 가지고 있는 SHA-1 알고리즘을 사용한다. 그림에서와 같이 접근 제한된 웹페이지를 나타내는 Private.html 은 SHA-1 해쉬 알고리즘에 의해 생성된 디렉토리에 저장되어 있다. 각 사용자별 디렉토리의 이름은 '패스워드 _인코딩' 구조를 가진다. 입력한 ID와 패스워드를 이용하여 디렉토리 이름을 얻어 그 이름과 일치하는 디렉토리가 웹서버 아래에 있는지 비교하여 액세스 제어를 하고, 이는 클라이언트측 자바스크립트 코드를 통하여 수행된다. 여기에서 보안디렉토리는 웹 영역과 동일하다. 만약 사용자가 패스워드를 안다면 올바른 URL이 구성되어지고 보안 디렉토리 내부의 접근 제한된 웹 페이지를 검색하는데 이용될 수 있다. 반면에 요구된 URL이 잘못되고, 웹서버가 요구된 웹페이지를 발견하지 못하는 경우를 제외하면, 패스워드 정보는 접근 제한된 웹페이지의 URL에 담겨지게 된다. 구체적으로, 자바스크립트 코드는 보안 디렉토리 외부의 웹 영역의 입구로서 동작하는 sign-in 웹 페이지에 삽입된다. 사용자는 이 페이지에 패스워드를 입력할 수 있다. 사용자가 입력을 마친후, 자바스크립트 코드는 웹 영역 내부의 접근 제한된 웹페이지의 URL을 구성하도록 사용자가 입력한 패스워드의 해쉬값을 이용할 것이다. 이 해쉬값은 SHA-1 알고리즘을 통하여 수행되며, 보안디렉토리를 생성하는데 이용된다. 이것은 패스워드를 웹서버 관리자에게 노출시키지 않게 함이다. 해쉬 기능은 자바스크립트 코드에 의해 클라이언트 영역에서 수행되므로 접근 제한된 웹 페이지들은 웹서버의 디렉토리 구조가 유지되는 한 어떤 변화없이 다른 웹사이트에 옮겨질 수 있다.

2. 시스템 설계 및 구현

2.1 그룹 인증

이 방법은 공유된 패스워드를 웹 영역에 접근하는데 이용하는 것이다. 즉 모든 유효한 그룹의 사용자는 같은 패스워드를 얻는다. 이 경우에 웹 페이지 운영자는 단지 하나의 패스워드를 설정하는 것이다. 이 패스워드는 먼저 사용자들에 의해 알려질 필요는 없다. 운영자는 단순히 sign-in 웹 페이지에 그룹의 사용자들이 공통적으로 알고 있는 힌트 질문을 넣는다. 답은 유효한 사용자들에 의해 알려지고 패스워드로 사용된다.

입력된 패스워드는 SHA-1 알고리즘을 이용하여 메시지 다이제스트를 수행하고, 바이트 배열 폼으로 된 메시지 다이제스트 값을 base64 부호화 형식으로 변환한다. 이 부호화된 패스워드를 통해 디렉토리 이름을 얻어 접근 제한된 웹 페이지가 있는 디렉토리에 접근이 가능하다. 이 구현에서, 함수 sha1(password)를 통하여 얻은 패스워드의 해쉬값은 보안디렉토리의 이름으로 이용된다. 일반적으로 해쉬값은 바이너리 형식이고, 텍스트 변환은 해쉬값에 적용되며, base64 인코딩 방법을 사용하여 변환된 결과는 보안디렉토리의 이름으로 사용된다. 표 1은 그룹 인증일 경우 인증 웹 페이지의 헤드 부분에 삽입된 코드를 나타내며, 표 2는 바디 부분에 삽입된 코드를 나타낸다.

표 1. 그룹 인증 헤드 코드
Table 1. Group authentication head code

```
<SCRIPT language = "javascript">
// URL 주소에 대한 전역변수
var url = null;
function sha1(password)
{
// sha1 함수
        hashvalue = ......;
        return hashvalue
}
</SCRIPT>
```

표 2. 그룹 인증 바디 코드
Table 2. Group authentication body code

```
<FORM
action
="javascript:window.location=url+'/p
rivate.html'; method="get">
비밀번호:<INPUT name="Password"
value="" type="Password">
<INPUT value="전송" type="submit"
onClick="url=sha1(Password.value));
Password.value=''">
</FORM>
```

2.2 개인 인증

사용자 개인에 대한 변환 방식은 그룹 인증 방식과 유사하다. 하지만 패스워드는 그룹 사용자들 사이의 공유된 정보이기 때문에 구성되기 쉽고 그룹의 변화가 있을 때마다 변화될 수 있다는 결점을 가지고 있다. 이 결점은 사용자 이름과 패스워드를 담고 있는 보안 디렉토리에 관계된 접근 제한 리스트를 이용하여 해결할 수 있다.

따라서 이 구현에서는 사용자 ID와 패스워드의 해쉬값을 보안 디렉토리의 이름으로 이용한다. 이는 함수 sha1(password)를 통하여 얻은 패스워드의 해쉬값과 사용자 ID를 함께 적용하여 사용자ID.sha1(password)를 보안 디렉토리의 이름으로 사용하는 것이다. 여기서 제안된 방식은 사용자들이 각각 다른 패스워드를 가지며, 사용자는 접근 제한된 웹 페이지의 URL이 자신의 패스워드 정보를 가지고 있고, 절대 다른 사용자에 의해 전달될 수 없다라는 것을 확신할 수 있다.

사용자 인증일 경우 인증 웹페이지의 헤드 부분에 삽입된 코드는 그룹 인증과 동일하며, 표 3은 바디 부분에 삽인된 코드를 나타낸다.

표 3. 사용자 인증 바디 코드
Table 3. User authentication body code

```
<FORM
action
="javascript:window.location=url+'/p
rivate.html'; method="get">
사용자 ID:<INPUT name="Username"
value="" type="text">
비밀번호<INPUT name="Password"
value="" type="Password">
<INPUT value="전송" type="submit"
onClick="url=Username.value + "."
 + sha1(Password.value));
Username.value=''; Password.value=''">
</FORM>
```

Ⅳ. 실험 결과 및 고찰

웹 서비스를 이용하기 전에 회원 인증의 절차를 거치는데, 회원 인증은 컴퓨터의 웹 브라우저를 통해 실행한다. 그룹 인증의 경우 사용자가 입력한 패스워드를 이용하여 디렉토리가 생성되며, 개인 인증의 경우 입력한 아이디와 패스워드를 이용하여 디렉토리가 생성된다.

그림 6과 같이 그룹 인증의 경우, 그룹의 사용자가 웹 브라우저에서 URL을 요청하게 되면 사용자 인증을 나타내는 sign.html 파일이 나타나게 되며, 이 사용자 인증 화면에서 그룹의 구성원들은 모두 알고 있는 패스워드를 입력한다. 위의 예에서 패스워드를 "ischo"라고 입력하면, SHA-1 알고리즘으로 메시지 다이제스트를 하고 base64 부호화한 값은 "oyOTqniDPXFaSquu1O+FaQ==" 이며, base64 인코딩 결과로 디렉토리 이름인 "ioyOT-qniDPXFaSquu1O+FaQ=="를 얻을 수 있다. 그룹 인증과 마찬가지로 사용자 인증일 경우, 그림 7에서와 같이 아이디를 "ischo", 패스워드는 "ischo"라고 입력하였다. 패스워드는 SHA-1 알고리즘으로 메시지 다이제스트를 하고 base64 부호화한 값은 oyOTqniDPXFaSquu1O+FaQ==" 이며, '아이디. 패스워드_인코딩'으로 조합하여 디렉토리 이름인 "ischo.oyOTqniDPXFaSquu1O+FaQ=='를 얻을 수 있다. 그림 8,9는 그룹 사용자일 경우와, 개인 사용자일 경우, 이 디렉토리의 웹 페이지에 인증된 결과를 보여주고 있다.

사용자 디렉토리는 웹 서버 아래에 생성되고, 이 디렉토리 아래에 사용자가 요청한 웹 페이지인 private.html 파일이 나타나게 된다. 그림 10은 웹 서버의 사용자 개인 디렉토리가 생성된 그림을 나타내며, 각각의 사용자에 대한 디렉토리가 같은 위치에 디렉토리로써 생성되어 진다.

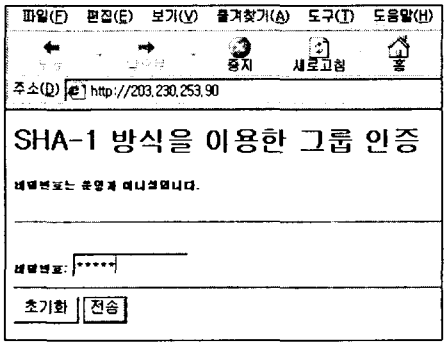

그림 6. SHA-1을 이용한 그룹 인증
Fig. 6 Group authentication Using the SHA-1 Algorithm

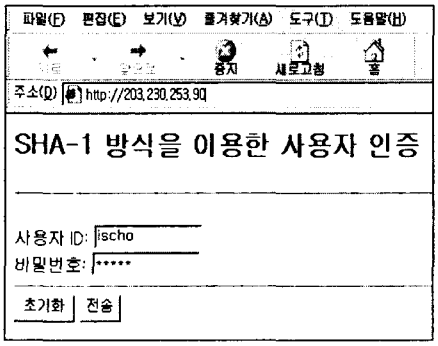

그림 7. SHA-1을 이용한 개인 인증
Fig. 7 User authentication Using the SHA-1 Algorithm

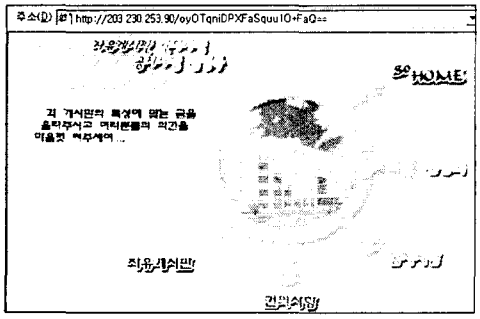

그림 8. 그룹 인증 결과
Fig. 8 Group authentication result

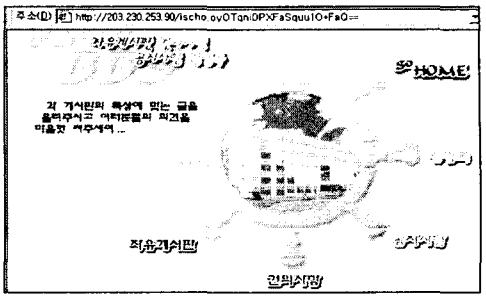

그림 9. 개인 인증 결과
Fig. 9 User authentication result

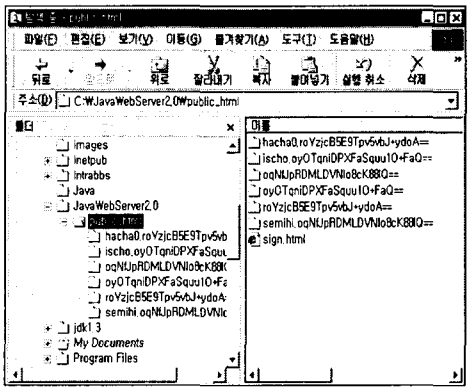

그림 10. 웹서버 사용자 디렉토리
Fig. 10 Web server user directory

V. 결론

본 논문에서는 웹 페이지 운영자가 웹 서버에 패스워드로 보호된 웹 영역을 배치시키는데 편리함을 제공하는 목적으로 새로운 패스워드 인증 방안을 제안하였다. 본 논문의 타당성 검토를 위해 그룹 인증과 개인 인증을 구현하여 실험하였다. 실험 결과에 따라 웹 영역은 접근 제한된 웹 페이지가 저장된 웹서버의 보안 디렉토리에 연결된다. 패스워드는 보안 디렉토리의 이름을 생성하는데 이용되며, 자바스크립트 코드는 보안 디렉토리 영역 외부의 인증 웹 페이지에 포함된다. 이 스크립트 코드는 사용자가 입력한 패스워드를 디렉토리 이름으로 변환시키고, 보안 디렉토리 내부의 접근 제한된 웹 페이지를 지시하는 완전한 URL을 형성한다. 따라서, 패스워드를 아는 사용

자만이 유효한 URL을 구성할수 있고, 접근 제한된 웹페이지를 검색할 수 있다. 이 방법에 따라, 웹 페이지 운영자는 서버 독립적인 방식으로 패스워드로 보호된 웹 영역을 배치할 수 있다는 것을 확인할 수 있었다.

본 논문에서 제안한 두가지 방법은 서버측 프로그래밍을 요구하지 않는다. 즉 공유된 웹서버에 접근 제한된 웹페이지를 구성하려는 웹 운영자들의 필요성을 잘 충족시키며, 클라이언트측 플러그인도 요구하지 않는다. 패스워드 변환은 현재 모든 웹브라우저에 의해 지원되는 클라이언트측 자바 스크립트코드에 의해 계산된다.

향후 연구로는 CGI 프로그램이 이 패스워드 인증 시스템과 공동 개발된다면 다른 보안과 나은 기능이 지원될수 있을 것이라고 사료된다.

참고문헌

[1] Mohammed J. Kabir, Apache Server: Administrators Handbook, IDG Books Worldwide, Mar. 1999.

[2] Matt Powell, Matthew Powell, and Leon Braginski, Running Microsoft Internet Information Server 4.0, Microsoft Press, Jul. 1998.

[3] Kaveh Gh. Bassiri, Programming Applications for Netscape Servers, Addison-Wesley Pub Co, Oct. 1998.

[4] T.Berners-Lee, et al, "Hypertext Transfer Protocol HTTP/1.0", IETF RFC1945, May 1996.

[5] R. Fielding, et al, "Hypertext Transfer Protocol HTTP/1.1", IETF FRC1945, May 1996.

[6] A. Freier, P. Karlton, and P. Kocher, "The Secure Socket Layer Protocol, Version 3.0", Netscape Communications, Inc., Now. 1996.

[7] T.Dierks and C.Allen, "The Transport Layer Security Protocol, Version 1.0", IETF RFC 2246, Jan. 1999.

[8] X. Lai and J.L Massey, "Hash function based on block cipher", Eurocrypt'92, pp.55-70, 1993.

[9] 윤 세 미, 조 익 성, 임 재 홍, "안전성을 위한 JOTP(Java One Time Password) 보안 알고리즘의 설계 및 구현," 한국정보처리학회 2001 춘계학술발표 논문집, 제 8 권, 제 1 호, pp.1049-1052, 2001년 4월 14일.

[10] National Institute of Standards and Technology, NIST FIPS PUB 180, "Secure Hash Standard", U. S, Department of Commerce, May 1993.

[11] L. Gong, "Increasing availability and security of an authenticated service", IEEE Journal on Selected Areas in Communications, vol. 11, no. 5, 1993.

[12] Ravi Kalakota, Andrew B. Whinston, "Readings in Electronic Commerce", Addison-Wesly publishing Company, 1997.

저 자 소 개

하 창 승

1984년 2월 한국해양대학교 항해학과 졸업(공학사)

1992년 2월 한국해양대학교 전자통신공학과(공학석사)

2001년 2월 한국해양대학교 전자통신공학과(공학박사 수료)

1996년 9월 ~ 현재 동명대학 정보통신계열 조교수

조 익 성

1997년 2월 한국해양대학교 전자통신공학과 졸업(공학사)

1999년 2월 한국해양대학교 전자통신공학과 졸업(공학석사)

2001년 한국해양대학교 전자통신공학과 졸업(박사수료)

2001년 3월 ~ 현재 동명대학 정보통신계열 전임강사