情報保護學會論文誌
第11券 第4號, 2001. 8

쿠키를 이용한 웹 보안시스템 설계 및 구현*

송 기 평**, 박 기 식**, 한 승 희***, 조 인 준****

The design and Implementation of Web Security System using the Cookies

Gi-pyeung Song**, Ki-Sik Park**, Seung-heui Han***, In-june Jo****

요 약

웹 서버는 HTTP(Hyper Text Transfer Protocol) 통신프로토콜을 사용한다. HTTP 프로토콜은 서버가 다음 통신전에 서 필요한 클라이언트의 상태정보를 유지하지 않는 특성을 지니고 있다. 따라서, 웹 서버는 클라이언트의 요구에 대응한 응답메시지 전송과 동시에 클라이언트에 관련된 모든 정보를 제거한다. 이러한 HTTP 프로토콜의 특성은 클라이언트 사용자에게 반복된 정보입력 부담을 요구케 한다. 이러한 불편 해결책으로 쿠키(Cookie)기술이 구현되어 활용되고 있다. 하지만, 쿠키는 평문형태로 전송되고 저장되기 때문에 정보가 쉽게 노출될 수 있다. 따라서, 쿠키정보가 유출, 복사, 수정이 가능하여 안전하지 않다.

본 논문에서는 이러한 웹 환경에서의 쿠키 특성에 착안하여 안전한 쿠키를 제시하고, 이를 이용하여 웹 보안시스템을 설계 및 구현하였다. 구현된 시스템은 어떤 웹 환경에서나 활용이 가능하고, 사용자 개인정보의 기밀성 보장과 더불어 인증, 무결성 등의 보안서비스를 제공한다.

ABSTRACT

A Web server makes use of the HTTP(Hyper Text Transfer Protocol) to communicate with a client. The HTTP is a stateless protocol; the server does not maintain any state information for ongoing interactions with the client. Therefore, the HTTP inevitably requires additional overhead as repeating data key-in to user for continuing communications. This overhead in Web environment can be resolved by the cookie technologies. However, the cookie is usually unsecured due to the clear-text to transfer on the network and to store in the file. That is, information in the cookie is easy to exposure, copy, and even change. In this paper, we propose a secure cookie mechanism appropriate to Web environment, and then present a design and implement of secure Web system based on the scheme. The Web system can be used to any web environment. It also provides some security services, such as confidentiality, authentication, integrity.

keyword : *Cookie, User Authentication, Web Security*

1. 서 론

인터넷을 통해서 자유롭게 정보를 유통시키고자 설계된 웹이 최근 들어 사용자 접근에어를 비롯한 여러 보안 기능들을 필요로 하고 있다. 특히 공간적, 시간적으로 제약을 받지 않는 전자상거래와 같이 상업적인 목적에 웹을 사용할 경우에는 더욱 그러하다. 웹 상에서 지불서비스를 전제로 전자상거래가 활성화

* 본 연구는 한국전자통신연구원 연구과제 지원으로 수행하였습니다.
** 한국전자통신연구원 표준연구센터(gpsong@pec.etri.re.kr, kipark@pec.etri.re.kr)
*** (주)아로통신기술(han@aaron.co.kr)
**** 배재대학교 컴퓨터공학과 네트워크 & 보안 연구실(injune@mail.paichai.ac.kr)

됨수록 더 절실하게 사용자 개인 정보를 요구하게 된다. 따라서, 개인 정보 노출을 방지할 수 있는 안전한 웹 시스템 필요성이 더욱 커지게 되었다.

웹 서버는 HTTP 통신프로토콜을 사용하여 클라이언트와 통신한다. 이 프로토콜은 클라이언트와의 연결상태를 유지하지 않는 특성을 지니고 있다. 따라서 이의 보완을 위해 웹 프로그램에서 쿠키(Cookie)[1] 기술을 추가하여 사용하기도 한다. 일반적으로 쿠키는 사용자 식별 및 인증정보 저장용으로 사용된다. 저장된 정보는 클라이언트가 웹 서버에게 요구메시지를 보낼 때마다 피기백(Piggyback)되어 서버에게 전달된다. 이를 전달받은 웹 서버는 이 정보를 이용하여 사용자 식별 및 인증 행위를 행한다. 따라서, 쿠키에는 사용자 아이디(ID), 패스워드(Password) 등과 같은 기밀정보가 다루어진다. 하지만, 쿠키에 저장된 사용자정보가 클라이언트에 저장되거나 네트워크에 전송될 때 평문상태이기 때문에 안전하지 않다. 그러나 쿠키에 보안기능을 추가하여 안전성을 보장하면, 웹 서버는 이를 이용하여 사용자의 접근제어를 비롯한 여러 가지 보안서비스를 제공할 수 있다. 이에 대한 연구[2]가 이루어지고 있지만 그 가능성만 제시할 뿐 구체적인 설계 및 구현은 다루지 않고 있다.

본 논문에서는 이와 같은 연구내용을 기반으로 상용으로 사용이 가능한 웹 보안시스템을 구체적으로 설계 및 구현하였다. 논문의 구성은 다음과 같다. 2장에서는 현재 웹 상에서 이루어지는 사용자 인증 방법과 쿠키기술을 검토하고, 3장에서는 쿠키의 안전성 보장을 위한 보안서비스를 기술하고, 4장에서는 안전한 쿠키를 이용한 웹 보안시스템을 설계하고, 5장에서 이를 구현하고 고찰하였다. 마지막으로 6장에 결론을 맺었다.

Ⅱ. 웹 사용자 인증 및 쿠키의 취약성

웹 상에서 이루어지는 사용자 인증은 주로 아이디와 패스워드를 사용자로부터 입력받거나 혹은 접속하고자 하는 웹서버가 생성한 쿠키를 클라이언트가 소유하고 있을 경우에는 쿠키로부터 이를 획득하여 서버에 저장된 사용자 등록정보와 비교하여 인증 여부를 판단하는 방법을 사용한다. 하지만 사용자 입력 정보 또는 쿠키정보가 네트워크 상에 평문상태로 전송되기 때문에 보안에 취약하다.

이러한 쿠키에 대한 공격유형은 네트워크 공격, 종단시스템 공격, 쿠키획득 공격으로 분류된다[2]. 네트워크 공격은 네트워크 상에 전송되는 평문형태의 쿠키가 노출되어 수정되는 것을 막한다. 이러한 공격은 서버와 브라우저에 설치된 SSL(Secure Socket Layer)[2] 프로토콜 사용으로 저지할 수 있다. 하지만, 이는 쿠키가 네트워크 상에 전송되는 동안에만 보호될 수 있다는 단점이 있다. 그리고 종단 시스템 공격은 브라우저가 설치된 종단시스템에 쿠키가 전송되어 평문형태로 하드디스크나 메모리에 존재하기 때문에 가능해진다. 이러한 쿠키는 사용자에 의해 쉽게 수정될 수 있고 다른 컴퓨터로 복사될 수 있다. 쿠키를 수정할 수 있는 능력은 악의적인 사용자에게 쿠키 내의 인증정보를 위조하고, 정당한 사용자로 위장할 수 있게 한다. 또한 쿠키를 복사할 수 있는 능력은 위조와 위장 모두를 용이하게 만든다. 마지막으로 쿠키 획득공격은 공격자가 정당하게 쿠키를 사용하는 사이트로 위장하여 쿠키를 모으고, 그 쿠키를 정당한 사용자의 것인 것처럼 사용하는 공격유형이다.

이러한 공격들은 쿠키가 평문상태로 다루어질 뿐만 아니라, 이의 공격에 고도의 해킹기술이 요구되지 않는데 연유된다. 하지만, 쿠키 자체는 사용자의 하드디스크 내용들을 인지하거나 웹 상에 사용자 기밀 데이터를 노출시키는 기능을 갖지 않는다. 그러므로, 쿠키내의 사용자 인증정보를 안전하게 보호할 수 있다면 쿠키를 통해서 안전한 웹 보안서비스를 제공할 수 있다. 다음 장에서 안전한 쿠키 실현을 위해 필요한 보안서비스를 살펴본다.

Ⅲ. 안전한 쿠키 실현을 위한 보안서비스

안전한 쿠키 실현을 위해서는 3가지 보안서비스가 쿠키에 제공되어야 한다. 즉, 쿠키에 대해 기밀성, 인증, 그리고 무결성 보안서비스 등이 제공되어야 함을 의미한다. 이에 대해 참고문헌 [2]에서 제시한 방안과 본 논문에서 설계 구현한 방안을 설명하면 다음과 같다.

3.1 쿠키 기밀성(Confidentiality)

2장에서 살펴본 바와 같이 평문상태의 쿠키 노출은 심각한 보안상의 취약성을 내포하고 있다. 이의 해결은 쿠키에 보안기술을 적용하여 기밀성을 유지하는 것이다[2].

이를 위해 본 논문에서는 쿠키의 생성 및 활용의

주체인 서버만이 쿠키를 암·복호화 할 수 있도록 쿠키 기밀성 유지방안을 제시하였다. 즉, 쿠키 암호화에 공개키 암호기술을 이용하였다. 이때, 웹 서버는 공개키 쌍을 소유한다. 웹 서버가 쿠키 생성 시에 쿠키 값을 공개키로 암호화하고 쿠키 검증 시에 암호화된 쿠키 값을 개인키로 복호화한다. 공개키 암호기술은 공개키로 암호화한 값이 공개키와 대응되는 개인키로만 복호화가 가능하다. 따라서 공개키로 암호화되어 저장된 쿠키는 개인키를 소유한 해당 웹 서버 외에 다른 사용자가 원본을 읽을 수 없으므로 가로채기에 의한 노출공격에 대응하여 쿠키 값의 기밀성을 보장할 수 있다.

3.2 쿠키 인증(Authentication)

공격자가 다른 사용자의 쿠키를 획득한 후, 이를 사용하여 쿠키를 생성한 서버에게 쿠키의 실제 소유자임을 흉내낼 수 있다. 이 문제에 대한 대응책으로 참고문헌 [2]에서 4가지 쿠키 인증 방안을 제안하고 있다.

첫째, 주소기반 인증은 쿠키에 사용자 시스템의 IP(Internet Protocol) 주소를 저장한다. 이를 *IP_Cookie*라 명명한다. 웹에서 제공되는 IP 주소는 사용자 환경변수의 하나이기 때문에 웹 서버가 이를 얻어서 *IP_Cookie*에 저장이 가능하다. 웹 서버는 사용자가 접속할 때마다 사용자의 현재 IP 주소와 *IP_Cookie*에 저장된 주소가 같은지 비교 검사한다. 그 값이 같으면, 서버는 자신이 현 사용자에 대해 생성한 쿠키라고 판단하여 쿠키를 인증한다. 즉, 현 사용자가 쿠키의 실제 소유자임을 믿는다. IP 주소의 사용은 서버와 클라이언트 사이에 통신하는 동안 인증정보 입력이 필요 없기 때문에 매우 편리하다. 하지만 동적 주소를 사용하거나 프록시 서버를 사용하는 도메인에서는 적합하지 않으며 IP 스푸핑(spoofing)을 피할 수 없다는 단점을 지닌다.

둘째, 패스워드기반 인증은 암호화된 사용자 패스워드 값이 저장된 쿠키를 이용한다. 이를 *Pwd_Cookie*라 명명한다. 이는 사용자가 입력한 패스워드의 값과 *Pwd_Cookie*로부터 복호화된 패스워드 값을 계산하여 이들의 일치 유무로 쿠키를 인증한다. 일치하면 쿠키의 실제 소유자라고 믿는다. 따라서 실제 쿠키 소유자만이 쿠키 사용이 가능하다.

셋째, 상호인증을 위해 쿠키에 Kerberos[3]를 사용할 수 있다. Kerberos는 네트워크에서 비밀키 기반 인증서비스이다. 인증을 위해 Kerberos를 쿠키에 사용하려면, 사용자는 쿠키 값을 대체할 추가적인 브라우저 소프트웨어가 필요하다. 이는 사용자에게 프로그램 설치, 사용의 부담을 줄 수 있다.

넷째, DSA[4] 와 RSA[5] 같은 전자서명 기술을 사용하여 쿠키를 상호인증 할 수 있다. 즉, 전자서명 기술을 사용하여 클라이언트는 이 쿠키가 정당한 서버가 생성하였음을 인증하고, 서버는 자신이 생성해 준 클라이언트로부터 쿠키가 전송되었음을 인증한다. 하지만, 이는 클라이언트의 브라우저를 수정해야하는 문제를 지니고 있다.

본 논문에서는 상기에 제시된 방안 중에서 네 번째 방안을 사용하여 쿠키를 인증하였다. 하지만 참고문헌 [2]의 문제점인 웹 브라우저를 수정하지 않고 실행될 수 있는 시스템을 구현한 점이 특징이다. 이에 대한 사항은 4장에서 자세히 설명한다.

3.3 쿠키 무결성(Integrity)

공격자가 쿠키내의 사용자 아이디, 패스워드 등의 변경이 가능하다. 따라서 서버는 이들의 변경 여부를 확인할 수 있는 방안이 필요하다. 이러한 문제해결을 위해 웹 서버가 MD5[6]와 SHA-1[7]를 사용하여 쿠키 값의 다이제스트를 생성한다. 그리고 이를 웹 서버 개인키로 전자서명하여 쿠키에 저장한다. 이를 *Seal_Cookie*라고 명명한다.

사용자가 웹 서버에 재 접속할 때, 클라이언트 브라우저로부터 쿠키 집합을 얻는다. 웹 서버는 공개키를 사용하여 *Seal_Cookie*에 저장된 서명을 검증한다. 쿠키가 변형되지 않아 유효하다면 웹 서버는 쿠키 값이 변경되지 않았음을 확인한다.

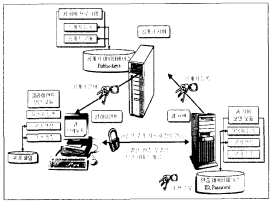
본 논문에서는 MD5를 사용하여 쿠키 값들의 다이제스트를 계산하고 이에 RSA를 사용하여 전자서명하는 방법으로 *Seal_Cookie*를 생성하였다.

Ⅳ. 쿠키를 이용한 웹 보안시스템 설계

이 장에서는 안전한 쿠키를 이용한 웹 보안시스템의 설계를 설명한다. 이를 위해 시스템 구성요소를 정의하고 안전한 쿠키 실현을 위한 보안서비스를 설계하여 이를 기초로 전체적 시스템을 설계하였다.

4.1 시스템 구성 요소

안전한 쿠키를 이용한 웹 보안시스템의 구성요소는



[그림 1] 시스템 구성 요소

[그림 1]과 같이 클라이언트, 웹 서버, 공개키 서버로 이루어진다.

4.1.1 클라이언트

이 구성요소는 웹 서버에 접근을 요구하고 이에 대응한 응답을 웹 서버로부터 수신하는 기본적인 기능을 한다. 본 설계에서 추가된 기능은 클라이언트가 인증이 필요한 웹 서버의 URI(Uniform Resource Identifier) 문서를 요구하면, 웹 서버가 본 시스템에서 제작한 '보안모듈(JAVA Applet)'이 포함된 프로그램을 전송하게 되는데 이를 수행하는 기능을 한다. 클라이언트에서 수행되는 '보안모듈'의 기능은 첫째, 웹 서버의 공개키를 분배받고, 둘째, 웹 서버의 공개키로 사용자 요구메시지를 암호화하고, 셋째, 이를 웹 서버로 전송하는 기능을 한다. 또한, 웹 서버로부터 수신된 안전한 쿠키를 자신의 쿠키 파일에 저장한다.

4.1.2 웹 서버

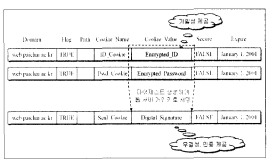
이 구성요소는 클라이언트로부터 접근요구를 수신하면 이에 대응한 응답을 만족하는 기본적인 서버기능을 한다. 본 설계에서 추가된 기능은 첫째, 클라이언트로부터 인증이 필요한 URI 요구를 수신하면, 위에서 언급한 '보안모듈'을 해당 클라이언트에게 전송하는 기능을 한다. 둘째, 클라이언트로부터 암호화된 메시지를 수신하면, 본 시스템에서 제작된 '보안 모듈'이 기밀성, 인증, 무결성 검사를 수행한다. 셋째, 클라이언트에 대응한 안전한 쿠키를 생성하여 클라이언트로 전송한다.

4.1.3 공개키 서버

이 구성요소의 기능은 클라이언트 사용자 인증 메시지의 암호화를 위해 웹 서버 공개키를 클라이언트에게 분배 기능을 한다.

4.2 안전한 쿠키 설계

기존의 클라이언트와 웹 서버 사이에서 사용되던 쿠키기술에 3장에서 제시한 방안을 추가하여 [그림 2]와 같은 안전한 쿠키를 설계하였다. 안전한 쿠키는 기밀성, 인증, 무결성 보안서비스를 제공하기 때문에 공격자의 공격으로부터 안전하게 보호된다.



[그림 2] 안전한 쿠키 설계

[그림 2]에서 *ID_Cookie*, *Pwd_Cookie*는 사용자 식별 및 인증을 위한 쿠키로 여기에 저장되는 값(ID, Password)은 웹 서버 공개키로 암호화되어 기밀성이 제공된다. *Seal_Cookie*는 해쉬 알고리즘(MD5)으로 쿠키 값들(*Encrypted_ID, Encrypted_Password*)의 다이제스트를 생성하고, 웹 서버 개인키로 서명함으로서 무결성과 인증을 제공한다. 각각의 쿠키가 가지는 값은 [표 1]과 같다.

[표 1] 쿠키 구성

쿠키 이름	설 명
ID_Cookie	사용자의 아이디(ID)
Pwd_Cookie	패스워드(Password)
Seal_Cookie	쿠키의 서명

4.3 보안서비스 설계

네트워크로 전송되는 사용자정보는 평문으로 전송되어 안전하지 않다. 따라서, 전송메시지에도 보안기능의 추가가 필요하다. 이 절에서는 쿠키(4.3.1절)와 전송메시지(4.3.2절)에 보안서비스가 제공되는 과정을 설계하였다. 이들은 공히 기밀성과 인증 보안서비스는 공개키 암호기술과 공개키 서명기술을 사용

[표 2] 기호 설명

기 호	설 명
M	메시지(ID, Password, IP)
I	사용자 등록 정보
E	암호화(Encryption)
D	복호화(Decryption)
S	전자서명(Digital Signature)
V	전자서명 검증(Verification)
H	해쉬 알고리즘(Hash algorithm)
C	클라이언트(Client)
W	웹 서버(Web Server)
KU_w	웹 서버 공개키(Public Key)
KR_w	웹 서버 개인키(Private Key)
+	연결(Concatenate)
-	할당(Assignment)
→	전송 방향

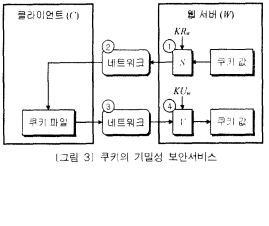
하여 제공되며, 투명성 보안서비스는 해쉬기술을 사용하여 제공된다. 설계에 사용된 기호는 [표 2]와 같다.

4.3.1 쿠키의 보안서비스

첫째, 쿠키의 기밀성 보안서비스는 [그림 3]과 같이 쿠키에 저장할 값을 웹 서버 공개키(KU_w)로 암호화하여 제공한다. 암호화된 쿠키는 쿠키를 생성한 웹 서버만이 개인키(KR_w)를 소유하고 복호화 할 수 있기 때문에 기밀성을 보장할 수 있다.

[그림 3]의 기밀성 보안서비스 과정을 [표 2]에서 정의한 표기법으로 기술하면 다음과 같다.

① W : $ID\_Cookie\{Encrypted\_ID\} = E_{KU_w}[ID]$
$Pwd\_Cookie\{Encrypted\_Password\}$
$= E_{KU_w}[Password]$
② $W \rightarrow C$: ID_Cookie, Pwd_Cookie



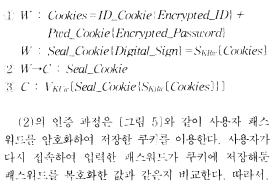
[그림 3] 쿠키의 기밀성 보안서비스

③ $C \rightarrow W$: ID_Cookie, Pwd_Cookie
④ W : $ID = D_{KR_w}[ID\_Cookie\{E_{KU_w}[ID]\}]$
$Password = D_{KR_w}[Pwd\_Cookie\{E_{KU_w}[Password]\}]$

동째, 쿠키 인증 보안서비스는 다음과 같이 세 가지 인증을 통해서 제공된다.

(1) 클라이언트가 정당한 서버로부터 쿠키가 생성되었는지를 인증한다.
(2) 서버가 정당한 클라이언트로부터 수신된 쿠키인지를 인증한다.
(3) 서버가 자신이 생성한 정당한 쿠키인지를 인증한다.

(1)의 인증 과정은 [그림 4]와 같이 쿠키 값을 웹 서버 개인키(KR_w)로 전자서명하고, 서명된 쿠키는 클라이언트에서 웹 서버 공개키(KU_w)로 서명을 검증한다. 따라서, 클라이언트는 쿠키가 정당한 서버로부터 생성되었는지를 인증할 수 있다.

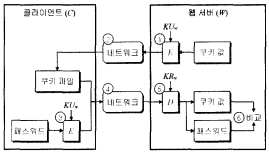


[그림 4] 쿠키의 인증 보안서비스 (1)

[그림 4]의 인증 보안서비스 과정을 [표 2]에서 정의한 표기법으로 기술하면 다음과 같다.

① W : $Cookies = ID\_Cookie\{Encrypted\_ID\} +$
$Pwd\_Cookie\{Encrypted\_Password\}$
W : $Seal\_Cookie\{Digital\_Sign\} = S_{KR_w}[Cookies]$
② $W \rightarrow C$: $Seal\_Cookie$
③ C : $V_{KU_w}[Seal\_Cookie\{S_{KR_w}[Cookies]\}]$

(2)의 인증 과정은 [그림 5]와 같이 사용자 패스워드를 암호화하여 저장한 쿠키를 이용한다. 사용자가 다시 접속하여 입력한 패스워드가 쿠키에 저장해둔 패스워드를 복호화한 값과 같은지 비교한다. 따라서, 서버는 쿠키가 정당한 클라이언트로부터 수신되었는지를 인증할 수 있다.



〔그림 5〕 쿠키의 인증 보안서비스 (2)

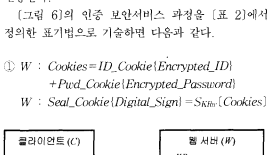
〔그림 5〕의 인증 보안서비스 과정을 〔표 2〕에서 정의한 표기법으로 기술하면 다음과 같다.

① W : $Pwd\_Cookie\{Encrypted\_Password\}$
$= E_{KUw}[Password]$
② $W \rightarrow C$: Pwd_Cookie
③ C : $E_{KUw}[Password]$
④ $C \rightarrow W$: $Pwd\_Cookie,\ E_{KUw}[Password]$
⑤ W : $Password$
$= D_{KRw}[Pwd\_Cookie\{E_{KUw}[Password]\}]$
$Pwd = D_{KRw}[E_{KUw}[Password]]$
⑥ W : $Password = Pwd$?

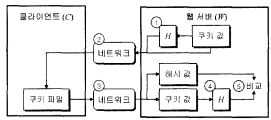
(3)의 인증 과정은 〔그림 6〕과 같이 쿠키 값을 웹 서버 개인키(KR_w)로 전자서명하고, 서명된 쿠키가 다시 웹 서버로 전송되면 공개키(KU_w)로 서명을 검증한다. 따라서, 자신이 생성한 정당한 쿠키인지를 인증한다.

〔그림 6〕의 인증 보안서비스 과정을 〔표 2〕에서 정의한 표기법으로 기술하면 다음과 같다.

① W : $Cookies = ID\_Cookie\{Encrypted\_ID\}$
$+ Pwd\_Cookie\{Encrypted\_Password\}$
W : $Seal\_Cookie\{Digital\_Sign\} = S_{KRw}[Cookies]$



〔그림 6〕 쿠키의 인증 보안서비스 (3)



〔그림 7〕 쿠키의 무결성 보안서비스

② $W \rightarrow C$: $Seal\_Cookie$
③ $C \rightarrow W$: $Seal\_Cookie$
④ W : $V_{KRw}[Seal\_Cookie\{S_{KRw}[Cookies]\}]$

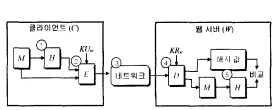
마지막으로, 쿠키의 무결성 보안서비스는 〔그림 7〕과 같이 쿠키 값을 해쉬 알고리즘(MD5)로 다이제스트 하여 제공한다. 쿠키에 저장해둔 다이제스트가 쿠키 값의 다이제스트와 같은지 비교하여 무결성을 검사한다.

〔그림 7〕의 무결성 보안서비스 과정을 〔표 2〕에서 정의한 표기법으로 기술하면 다음과 같다.

① W : $Seal\_Cookie = H(Cookies)$
② $W \rightarrow C$: $Seal\_Cookie,\ Cookies$
③ $C \rightarrow W$: $Seal\_Cookie,\ Cookies$
④ W : $H(Cookies)$
⑤ W : $Seal\_Cookie\{H(Cookies)\} = H(Cookies)$?

4.3.2 네트워크 전송메시지의 보안서비스

네트워크 전송메시지의 보안서비스는 〔그림 8〕과 같이 메시지의 다이제스트를 생성하고 웹 서버 공개키(KU_w)로 암호화하여 무결성, 기밀성 보안서비스를 동시에 제공하도록 설계하였다. 이를 수신한 웹 서버는 개인키(KR_w)로 복호화하여 기밀성을 제공하고, 복호화하여 얻은 다이제스트를 원본 메시지의 다이제스트와 비교하여 무결성을 검사한다. 또한 사용자 인증 DB로부터 아이디, 패스워드를 검사하여 사용자에 대한 인증을 한다.



〔그림 8〕 전송 메시지의 보안서비스

[그림 8]의 전송 메시지 보안서비스 과정을 [표 2]에서 정의한 표기법으로 기술하면 다음과 같다.

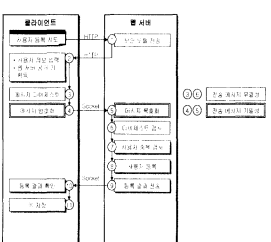
1. $C : H(M)$
2. $C : E_{KUw}[H(M), M]$
3. $C \rightarrow W : E_{KUw}[H(M), M]$
4. $W : D_{KRw}[E_{KUw}[H(M), M]]$
   $H(M), M$
5. $W : H(M) = H(M)$?

4.4 시스템 전체 동작 설계

앞서 설명한 안전한 쿠키를 이용하여 설계한 웹 보안시스템 전체동작은 다음과 같다.

4.4.1 사용자 등록 과정

웹 서버에 처음 접속한 사용자가 개인정보를 입력하여 새로운 사용자로 등록되는 과정을 [그림 9]에서 보여주고 있다. 전송메시지에 보안서비스를 제공하는 3)에서 6)까지의 과정을 설명한다.



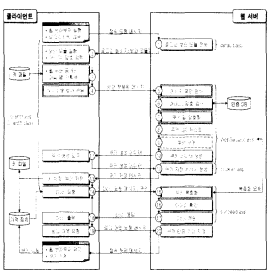
[그림 9] 사용자 등록 과정

3) 전송할 메시지의 다이제스트를 생성한다.
   $C : H(U)$
4) 다이제스트를 원본 메시지와 함께 웹 서버 공개키로 암호화하여 웹 서버로 전송한다.
   $C : E_{KUw}[H(U), U]$
5) 암호화된 메시지를 개인키로 복호화한다.
   $W : D_{KRw}[E_{KUw}[H(U), U]]$
6) 복호화하여 얻은 다이제스트를 원본 메시지의 다이제스트와 비교하여 무결성을 검사한다.
   $W : H(U), U$
   $H(U) = H(U)$?

4.4.2 쿠키 생성 및 검증 과정

웹 서버가 사용자 입력정보를 이용해 쿠키 데이터를 생성하여 클라이언트로 전송하고 클라이언트의 쿠키를 검증하는 과정을 [그림 10]에서 보여주고 있다.

전송되는 메시지는 아이디와 패스워드이며 보안서비스를 제공과정을 사용자 등록 과정과 같다.



[그림 10] 쿠키 생성 및 검증 과정

[웹 서버]

1) 클라이언트로부터 접속요청을 받으면 보안모듈이 포함된 로그인 웹 페이지를 클라이언트로 전송한다.

[클라이언트]

2) 웹 브라우저에서 클라이언트 보안프로그램이 실행되어 사용자로부터 아이디, 패스워드를 입력받는다.
3) 웹 서버 공개키(KU_w) 파일을 열어 공개키를 가진다. 파일을 열지 못하면 공개키 서버로 요청하여 제공받는다.
4) 사용자 아이디, 패스워드에 보안을 적용하여 웹 서버로 전송한다.
   $C : H(ID, Password)$
   $C : E_{KUw}[H(ID, Password), (ID, Password)]$
   $C \rightarrow W : E_{KUw}[H(ID, Password),$
   $(ID, Password)]$

[웹 서버]

5 수신된 메시지의 보안검사를 수행한다.

W : $D_{KMw}[E_{KUw}[H(ID,\ Password),\ (ID,\ Password)]]$

$H(ID,\ Password),\ (ID,\ Password)$

W : $H(ID,\ Password) = H(ID,\ Password)$?

6 사용자 인증 DB로부터 아이디, 패스워드를 검사한다.

7 쿠키를 생성할 값들을 웹 서버 공개키(KU_w)로 암호화하여 클라이언트가 소유하게될 암호화된 쿠키값을 생성한다.

W : $ID\_Cookie\{Encrypted\_ID\} = E_{KUw}[ID]$

$Pwd\_Cookie\{Encrypted\_Password\}$

$= E_{KUw}[Password]$

8 암호화된 쿠키 값을 다이제스트 한다.

W : $Cookies = ID\_Cookie\{Encrypted\_ID\}$

$+Pwd\_Cookie\{Encrypted\_Password\}$

W : $H(Cookies)$

9 다이제스트를 웹 서버 개인키(KR_w)로 전자서명하여 $Seal_Cookie$를 생성한다.

W : $Seal\_Cookie\{Digital\_Sign\}$

$= S_{KRw}[H(Cookies)]$

10 클라이언트에 대응한 암호화된 쿠키 값과 쿠키 서명 결과를 클라이언트로 전송한다.

$W \rightarrow C$: $ID\_Cookie\{Encrypted\_ID\} = E_{KUw}[ID]$

$Pwd\_Cookie\{Encrypted\_Password\}$

$= E_{KUw}[Password]$

$Seal\_Cookie\{Digital\_Sign\}$

$= S_{KRw}[H(Cookies)]$

[클라이언트]

11 쿠키 생성 요구를 위해 쿠키 생성 데이터를 웹 서버로 전송한다.

$C \rightarrow W$: $ID\_Cookie\{Encrypted\_ID\} = E_{KUw}[ID]$

$Pwd\_Cookie\{Encrypted\_Password\}$

$= E_{KUw}[Password]$

$Seal\_Cookie\{Digital\_Sign\}$

$= S_{KRw}[H(Cookies)]$

[웹 서버]

12 쿠키 저장 메시지(HTTP 헤더)를 만들어 클라이언트로 전송한다.

[클라이언트]

13 웹 서버 공개키(KU_w)를 파일 저장하고, 웹 브라우저는 쿠키를 소유하게 된다.

14 도큐먼트를 요청한다.

[웹 서버]

15 클라이언트의 쿠키를 복호화하여 사용자 아이디를 얻어낸다.

16 사용자 아이디를 확인하여 도큐먼트의 제공 여부를 결정한다.

17 도큐먼트를 전송한다.

[클라이언트]

18 도큐먼트를 출력하고 다른 도큐먼트를 요청하거나 로그 아웃 요청을 한다.

클라이언트에서 로그 아웃 요청을 하면 웹 서버 측에서 쿠키의 만료기간을 지정하여 클라이언트가 소유한 쿠키를 강제로 삭제한다. 로그 아웃 요청 없이 사용자가 웹 브라우저를 닫으면 쿠키는 기억장치에서 자동으로 삭제된다.

V. 구현 및 고찰

5.1 구현 환경

본 논문의 구현은 시스템에 독립적인 특성을 지닌 자바 언어기반으로 한다. 또한 쿠키기술 역시 시스템에 독립적이며, 쿠키 생성이 가능한 환경이면 구현이 가능하기 때문에 확장성이 있다.

하지만, ASP(Active Server Page) 웹 서버인

〔표 3〕 구현 환경

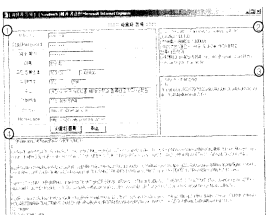
PC 시스템	
CPU	Intel Pentium III 333
RAM	256MB
웹 서버	
운영체제	MS Win 2000 Advanced Server
웹 서버	MS IIS 5.0
데이터베이스	MS SQL Server 7.0
클라이언트	
운영체제	MS Win 98, 2000 professional
웹 브라우저	MS Internet Explorer 5
프로그래밍	
프로그래밍 언어	JAVA(JDK 1.2.2), ASP, HTML
프로그래밍 툴	KAWA 3.5, Visual Cafe 3
암호 알고리즘	RSA, MD5

경우는 쿠키기술이 ASP 세션변수 기술과 비슷한 기술이이다. 후자의 기술사용은 ASP라는 특정 서스템에 의존적이다. 따라서, 제안시스템에서 이의 경우에도 ASP의 쿠키 생성기능을 사용하도록 구현하였다.

본 논문에서 설계한 안전한 쿠키를 이용한 웹 보안 시스템은 [표 3]과 같은 환경에서 구현하였다. 자바 프로그램 개발 키트(JDK) [9] 버전 1.2.2로 프로그래밍하였으며, GNU에서 공개한 자바 암호라이브러리를 사용하였다.

5.2 사용자 등록

새로운 사용자 등록 시에 클라이언트의 웹 브라우저에서 사용자정보가 처리되는 과정을 [그림 11]에서 보여준다. 신상정보를 입력하고 '사용자 등록' 버튼을 누르면 본 시스템에서 제작한 프로그램의 보안모듈이 이를 처리하여 웹 서버로 전송한다.

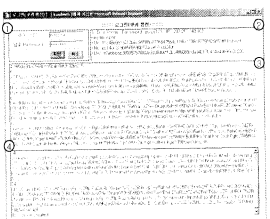


[그림 11] 사용자 등록 과정 보기 화면

1. 사용자로부터 등록정보를 입력받는다.
2. 평문상태의 사용자 입력정보를 얻는다.
3. 웹 서버 공개키를 요청하여 얻는다.
4. 사용자 입력 정보를 다이제스트, 암호화하여 웹 서버로 전송하고 사용자 등록결과를 확인한다. 웹 서버로부터 응답이 오면 웹 서버 공개키를 파일에 저장한다.

5.3 쿠키 생성

클라이언트 사용자가 아이디, 패스워드를 입력하여 웹 서버로부터 쿠키를 생성 받는 과정을 [그림 12]에서 보여준다.



[그림 12] 쿠키 생성 과정 보기 화면

1. 사용자로부터 아이디, 패스워드를 입력받는다.
2. 평문상태의 사용자 입력정보를 얻는다. 웹 서버 공개키 과정을 얻다.
3. 사용자 입력정보를 다이제스트, 암호화하여 웹 서버로 전송한다.
4. 웹 서버가 생성하여 응답한 쿠키 생성데이터를 웹 서버로 전송하고 쿠키 생성결과를 확인한다.

클라이언트의 웹 브라우저가 웹 서버로부터 쿠키를 생성받으면 쿠키를 기억장치(RAM)에 저장하고 있다가 웹 서버가 요청할 때마다 제공하게 된다.

웹 서버가 생성하고 클라이언트로 전송되어 저장된 쿠키 값을 [그림 13]에서 보여준다.

이는 실제 웹 브라우저가 기억장치에 저장하고 있는 쿠키 값이다. 이들 중에 *ID_Cookie*, *Pwd_Cookie*는 웹 서버 공개키로 암호화되어 저장되기 때문에 웹 서버만이 개인키로 복호화하여 사용할 수 있다. 이는 클라이언트에 저장되거나 네트워크에 전송 중에

[그림 13] 클라이언트 쿠키

도 안전하게 해준다. *Seal_Cookie*는 쿠키 값을 다이제스트하고 웹 서버 개인키로 전자서명하여 저장되기 때문에 무결성을 제공하면서 웹 서버가 생성한 쿠키를 인증할 수 있다.

5.4 구현 시스템 성능 고찰

본 논문에서 제안한 시스템으로 인해 성능에 미치는 요소는 다음과 같다. 클라이언트 측에서는 (1) 자바 애플릿('보안모듈') 내려 받기 지연시간, (2) 이 애플릿을 실행하여 네트워크 전송메시지에 다이제스트 계산시간, (3) 그리고 이의 암호화에 소요된 시간이다. 이러한 요소들은 (1)을 제외하고는 현재 상용화되어 활용 중인 보안제품(SSL 등)과 비교하여 대동소이하다. 여기에서 (1)에 의해 발생하는 지연시간은 애플릿 프로그램 크기에 좌우된다. 제안시스템에서 애플릿 크기가 15 Kbyte에 불과하기 때문에 현대의 고품질 네트워크 성능과 비교하여 성능에 미치는 영향은 미비하다. 다음으로 서버 측에서는 (1) 쿠키의 암호화에 소요된 시간, (2) 서명 및 다이제스트 계산에 소요된 시간이다. 이들 역시 상용화되어 활용 중인 제품과 비교하여 새롭게 추가된 성능 저해 요소가 아니기 때문에 성능측면에서 기존의 타 보안 시스템과 대동소이하다고 판단된다.

5.5 타 웹 보안 시스템과 비교 분석

현재 대표적으로 사용 중이거나 제안된 웹 보안용 소프트웨어 제품은 BAA(Basic Access Authentication)[13], DAA(Digest Access Authentication)[13], SSL 프로토콜이다. 이들과 본 시스템을 요약 비교한 결과가 [표 4]와 같다.

[표 4]에서 본바와 같이 BAA 방안은 사용자 패스워드를 비롯한 모든 정보가 평문상태로 다루어지기 때문에 안전하지 않다. 따라서 BAA방안에서 인증은 안전한 인증서버스라기 보다는 사용자 식별서비스에 가깝다. DAA 방안에서는 MAC을 사용하여 사용자 패스워드를 노출시키지 않고 인증서비스가 이루어지기 때문에 BAA 방안보다 안전한 인증방법이다. 하지만, 이는 단지 BAA를 대체한 것일 뿐, MAC 형태의 암호시스템에서 발생하는 모든 문제를 그대로 내포하고 있어 개인키 방안을 사용하는 인증방법 만큼 안전하지 않다. SSL 프로토콜은 기밀성, 인증, 재현공격 방지 등의 보안서비스를 제공한다. 하지만, 호스트와 호스트간 보안서비스만을 제공하기 때문에 클라이언트에 저장된 쿠키를 안전하게 보호하지 못할 뿐 아니라 클라이언트 사용자에게 인증서 관리부담이 수반된다.

[표 4] 보안서비스 비교

보안서비스		BAA	DAA	SSL	본 시스템
기밀성	전송 메시지	X	○	○	○
	쿠키	X	X	△	○
인증	클라이언트	X	○	○	○
	웹 서버	X	X	○	○
	쿠키	X	X	X	○
무결성	전송 메시지	X	○	○	○
	쿠키	X	X	X	○

○ : 제공, X : 제공하지 않음, △ : 일부 제공

본 시스템은 전송메시지의 보안서비스뿐만 아니라 로컬 호스트상의 쿠키정보도 안전하게 보호하기 때문에 다른 어떠의 방안들보다 더욱 안전하다고 볼 수 있다. 쿠키는 쿠키 값을 암호화하여 기밀성을 제공하고, 다이제스트 한 후, 전자서명하여 무결성과 인증 서비스를 제공한다. 쿠키를 이용한 인증은 클라이언트와 웹 서버 간에 상호인증이 가능하게 한다. 클라이언트는 정당한 웹 서버가 쿠키를 생성했는지 인증하고, 웹 서버는 정당한 클라이언트로부터 쿠키가 수신되었는지를 인증한다. 또한 웹 서버는 쿠키가 자신이 생성한 정당한 것인지 인증한다.

Ⅵ. 결 론

본 논문은 안전한 쿠키를 설계하고 이를 활용하여 보안서비스를 제공할 수 있는 안전한 웹 시스템을 설계 및 구현하였다. 즉, 안전한 쿠키 실현을 위해 전자서명과 암호화에 RSA 공개키 알고리즘, 메시지 다이제스트에 MD5 해쉬 알고리즘을 적용하여 기밀성, 상호인증, 무결성 등의 보안서비스를 제공하였고, 안전한 네트워크 전송메시지 실현을 위해서도 동일한 방법으로 이들 보안서비스가 제공될 수 있도록 하였다. 이로써 기존 쿠키와 전송 메시지에서의 보안 취약점들을 보완할 수 있었다.

본 시스템의 특징은 어떤 웹 환경에서나 동작이 가능하다는 점이다. 즉, 기존의 웹 보안시스템(DAA, SSL 등)이 웹 서버와 클라이언트 소프트웨어 수정을 요구하여 이의 확산에 장애 요인이었다. 하지만 본 시스템은 자바 애플릿기술을 활용하여 이들 소프트웨어 수정 없이 어떤 웹 환경에서나 설치 및 동작

이 가능하여 이식성이 탁월하다는 점이다.

현재 구현된 시스템은 단문을 중심으로 한 웹 보안시스템을 설계하였다. 향후에는 장문의 문서에 대해 보안서비스를 제공할 수 있는 대칭 키 기반 보안서비스 설계가 요구된다.

참 고 문 헌

[1] D. Kristol, "HTTP State Management Mechanism", February 1997, RFC 2109. http://www.ietf.org/rfc/rfc2109.txt

[2] Joon S. Park, Ravi Sandhu, and SreeLatha Ghanta, "RBAC on the Web by secure cookie", In proceedings of the IFIP WG11.3 Workshop on Database Security, Chapman & Hall, July 1999.

[3] William Stallings, "Cryptography and Network Security: Principles and Practice, Second Edition", Prentice-Hall Inc., 1999.

[4] Federal Information Processing Standards Publication. Digital Signature Standard (DSS), 1994. FIPS PUB 186.

[5] R.L.Rivest, A.Shamir, and L.Adleman, "A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM", 21(2):120-126, 1978.

[6] R. Rivest, "The MD5 Message Digest Algori- thm", April 1992. RFC 1321. http://www.ietf.org/rfc/rfc1321.txt

[7] Federal Information Processing Standard (FIPS). Secure Hash Standard, 1995. FIPS 180-1.

[8] Java 2 SDK, Standard Edition Documentation Version 1.2.2-001, Sun Microsystems Inc., 1999.

[9] Java 2 SDK, Standard Edition 1.2, Sun Microsystems Inc., 2000. http://java.sun.com/products/jdk/1.2/

[10] Java Cryptography Extension(JCE) 1.2.1, Sun Microsystems Inc., 2000. http://java.sun.com/products/jce/

[11] KAWA. Tek-Tools. Inc, 2000. http://www.tek-tools.com/kawa/

[12] Bruce Schneier, "Applied Cryptography, Second Edition", John Wiley & Sons, Inc., 1996.

[13] 송기평, 손홍, 김선주, 조인준, "웹 환경에서 자바 기술을 이용한 안전한 사용자 식별 및 인증 모델 설계", *한국해양정보통신학회 2000 추계 종합학술대회지 제4권 제2호*, October 2000.

[14] 구경철, 조인준, 한승희, "쿠키를 이용한 웹 보안시스템 설계", *한국통신정보보호학회 종합학술발표회 논문집*, November 2000.

〈著者紹介〉

송 기 평 (Gi-pyeung Song)
1994년 2월 : 대전산업대학교 전자계산학과 졸업
1980년 7월~현재 : 한국전자통신연구원 선임기술원
1999년 : ITU-T TSAG 국가대표
1996년~현재 : 한국정보통신기술협회 EDH 위원
1996년~현재 : 한국정보통신기술협회 기획 및 전략특별위원회 위원
〈관심분야〉 인터넷 보안, 객체지향형 데이터베이스, 멀티미디어 웹, 워크플로우 설계

박 기 식 (Ki-sik Park)
1982년 2월 : 서울대학교 졸업(문학사)
1984년 2월 : 서울대학교 행정대학원 정책학 석사
1995년 8월 : 충남대학교 정책학 박사(정보통신기술정책 분야)
2000년 4월 : 대한민국 산업포장 수상(제3843호)
1985년 1월~현재 : 한국전자통신연구원 표준연구센터 센터장/책임연구원
한국정보통신기술협회 기획전략특별위원회 의장
표준화 운영위원회 위원
국가정보화 예산 심의 위원
정보통신진흥원 평가 및 과제 심의 전문위원
한남대학교 대학원 겸임 부교수
1996년~현재 : ITU-T TSAG Vice Chairman, ITU-T TSAG WP3 Chairman
(국제전기통신연합 표준화자문위원회 부의장 겸 전자적 문서처리분과 위원장)
1988년 2월~현재 : APT/ASTAP Advisory Board Member
(아·태 표준화 협의체 자문위원)
〈관심분야〉 정보통신표준화 관련 법·제도, 기술 전략기획, 정보통신정책, MIS

한 승 희 (Seung-heui Han) 학생회원
1999년 2월 : 배재대학교 전자계산학과 졸업
2001년 2월 : 배재대학교 컴퓨터공학과 석사
2001년 2월~현재 : (주)아론통신기술
〈관심분야〉 정보보호, 인터넷 보안

조 인 준 (In-june Jo) 정회원
1982년 2월 : 전남대학교 계산통계학과 졸업
1985년 2월 : 전남대학교 전자계산학과 석사
1999년 8월 : 아주대학교 컴퓨터공학과 박사
1983년 9월~1994년 2월 : 한국전자통신연구원 선임연구원
1991년 12월 : 전산조직응용기술사
1994년~현재 : 배재대학교 컴퓨터공학과 교수
1995년~현재 : 한국통신정보보호학회 종신 회원
1995년~현재 : 한국통신정보보호학회 정보보호응용 연구회 전문위원
1997년~현재 : 한국정보처리학회 시스템 통합 연구회 학술위원
1992년~현재 : 대한기술사회 회원
1996년~현재 : 한국정보통신 기술사 협회 회원
〈관심분야〉 정보보호, 컴퓨터네트워크, 전산조직응용