# 공개키 기반의 안전한 쿠키 설계

박 정 화*, 이 경 현**
*부경대학교 전산정보학과, **부경대학교 전자컴퓨터정보통신공학부

# The Design of Secure Cookie based on PKI

Jung-Hwa Park*, Kyung-Hyune Rhee**
*Dep. of Computer and Information Science, Pukyong Nat'l Univ.
**Div. of Electronic, Computer and Telecommunication Engineering, Pukyong Nat'l Univ.

요 약

웹 프로토콜인 HTTP는 효율적으로 동작하기 위하여 이전 상태 정보를 저장하지 못하는 Stateless 특성을 가지고 있으므로, 이러한 문제를 해결하여 상태 유지 및 사용자 편의성을 제공하기 위해 만들어진 것이 쿠키(cookie)이다. 그러나 쿠키는 평문 형태로 전송되고 사용자 컴퓨터에 일반 텍스트 형태로 저장되므로 쉽게 노출되어 쿠키 파일의 복사, 수정이 가능하여 안전성에 심각한 위험이 존재한다. 본 논문에서는 이러한 쿠키의 보안 문제를 해결하기 위해 안전한 쿠키 설계 및 구현 방안을 제시하였다.

## 1. 서론

컴퓨터와 통신의 비약적인 발달로 인터넷을 통한 개인간, 기업간, 국가간의 정보 교류가 빈번해지고 점점 복잡해지는 추세이다. 최근 들어 웹을 이용한 정보 교환에 많은 허점들이 대두되면서 접근 제어 및 보안 시스템을 필요로 하고 있다. 특히, 전자상거래와 같은 상업적인 목적의 웹사이트 경우 지불 서비스가 필수적이고, 그에 따른 신용카드 및 개인정보 노출을 방지할 수 있는 안전한 보안 시스템이 필요하게 되었다.

웹 서버는 HTTP 프로토콜을 이용한다. 이 프로토콜은 단순하기 때문에 클라이언트와의 이전 연결 상태를 유지하지 못한다. 즉, 웹 서버가 클라이언트의 요구에 응답을 완료하면 요구한 클라이언트와 관련된 모든 정보를 잃어버린다. 따라서 이러한 문제점을 보완하기 위해 만들어진 것이 쿠키 기술이다.

일반적으로 쿠키는 웹서버를 방문한 사용자의 ID, 패스워드 등과 같은 사용자 정보를 저장하여 다음 접속할 때 ID와 패스워드의 입력 없이 웹서버에 곧바로 접속할 수 있는 기능 등을 제공한다. 그러나 대부분의 쿠키는 사용자 정보 및 패스워드와 같은 비밀이 보장되어야 할 자료들임에도 불구하고, 네트워크상에 전송되거나 클라이언트에 저장될 때 평문 형태로 이루어지므로 안전성이 보장되지 않는 단점이 있다.

안전하지 못한 쿠키에 보안 기능을 추가하면 사용자 관련 자료(예, 사용자 ID, 패스워드, 신용카드 정보)를 데이터베이스에 저장하지 않고 클라이언트 시스템 쿠키에 저장함으로써, 사용자가 웹서버로 접근할 때마다 ID와 패스워드를 입력하는 불편을 줄일 수 있고 서버의 데이터베이스 관리 유지보수 비용을 줄일 수 있으며, 사용자의 속성 정보 기반의 제어가 가능한 장점이 있다[3].

본 논문은 기존의 안전하지 못한 쿠키를 공개키기반의 암호화 기법으로 암호화하고, 서버에서 암호화된 정보을 해석하고, 클라이언트 측에서는 플러그인을 통하여 암호화된 쿠키에 접근할 수 있도록 구성된 하나의 프레임웍을 제안한다. 논문 구성은 다음과 같다. 2장에서는 웹에서의 쿠키 개념 및 보안 취약점을 설명하고, 3장에서는 제안하는 안전한 쿠키에서 사용되는 공개키 기반 구조 및 인증서 기술에 대해서 간략히 약술하며, 4장에서는 제안하는 안전한 쿠키를 이용한 사용자 인증 시스템에 대해서 기술하며, 5장에서는 제안하는 시스템의 구현방안을 제시하며, 마지막으로 6장에서는 결론을 맺는다.

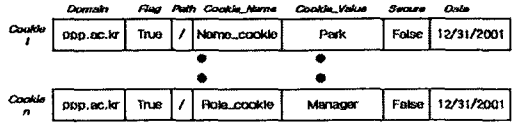
## 2. 쿠키의 구조 및 취약점

### 2.1 쿠키의 구조

현재 웹에서 사용되고 있는 일반적인 쿠키의 구조는 <그림 1>과 같다[2].

- Domain : 쿠키를 사용할 수 있는 호스트 혹은 도메인을 나타낸다.

- Flag : 쿠키 정보를 access할 수 있는 도메인의 모든 컴퓨터들이 access할 수 있는 지의 여부를 나타낸다.
- Path : 도메인 내 Path에 지정된 페이지만이 쿠키를 access할 수 있다.
- Cookie_Name : 쿠키의 이름을 나타낸다.
- Cookie_Value : 쿠키에 저장될 값을 나타낸다.
- Secure : True가 설정되어 있으면 SSL(Secure Sockets Layer)과 같은 안전한 통신 채널을 통해서만 전송이 가능하다.
- Date : 쿠키의 유효기간을 나타낸다.




<그림 1> 웹 상에서의 일반적인 쿠키 구조

## 2.2 쿠키의 취약점

쿠키에 대한 알려진 보안 위협은 아래와 같다[1][4].

- 네트워크 위협 : 쿠키들은 네트워크 상에서 평문으로 전송되므로, 재생공격이나 변경에 취약하다.
- 종단 시스템 위협 : 쿠키는 하드디스크나 메모리에 평문으로 저장되어 있으므로, 변경, 복사가 가능하다. 따라서, 가장 공격에 취약하다.
- 쿠키 획득 위협 : 공격자가 적법한 웹사이트로 가장하여 쿠키들을 수집하고, 그 쿠키들을 이용하여 다른 사이트의 접근이 가능하다.

# 3. 공개키 기반 구조 및 인증서 검증 기술

## 3.1 공개키 기반 구조

개방형 네트워크 환경에서의 보안 요구사항을 만족시키기 위해 공개키 암호화 인증서의 사용을 가능하게 해주는 기반구조를 공개키 기반구조(PKI)라고 한다. 네트워크를 이용해 전송되는 데이터에의 메시지 도청, 변조, 위조, 송수신 부인 등의 위협요소가 존재하기 때문에, 이러한 위협을 방지하기 위해서 PKI는 기밀성, 무결성, 부인방지, 접근제어, 인증과 같은 보안 서비스를 제공한다.

공개키 기반 구조를 구성하는 구성 요소와 그의 특징은 아래와 같다.

- 인증기관(CA:Certificate Authority) : 인증서의 등록, 발급, 조회시 인증서의 정당성에 대한 관리를 총괄하는 시스템이다.
- 등록기관(RA:Registration Authority) : 인증기관과 물리적으로 멀리 떨어져 있는 사용자들을 위해 인증기관과 인증서 요청, 객체 사이에 등록기관의 역할을 수행한다.
- 디렉토리(Directory) : 인증서와 사용자 관련 정보, 상호 인증서 쌍 및 인증서 취소목록 등을 저장, 검색하는 장소로 응용에 따라 이를 위한 서버를 설치하거나 인증기관에서 관리한다.
- 사용자(User) : 공개키 기반 구조내의 사용자뿐만 아니라 사용자가 사용하는 모든 시스템을 의미한다.

## 3.2 인증서 취소 검증 기술

공개키 기반 구조에서 인증서를 통한 상호 인증의 수행시, 인증서의 적법성을 검증하기 위한 기술로 다음의 방법들이 사용 및 제시되고 있다.

- CRLs(Certificate Revocation Lists) : CA는 취소되어 더 이상 사용되지 않는 모든 X.509 기반의 인증서 리스트를 주기적으로 발행하고 전자적으로 서명한다. 특정 인증서의 취소여부를 확인하고자 하는 사용자들은 적절한 CA에게 CRLs을 질의하여, 수신한 인증서의 취소 여부를 판단한다.
- OCSP(Online Certificate Status Protocol) : 최종 개체가 특정의 인증서에 대한 유효성과 취소 정보를 효율적이고 신속하게 온라인 상에서 알 수 있게 하는 방법이다[6].
- Short-lived Certificate : 24-48 시간의 짧은 유효기간을 갖는 인증서를 사용하는 기술로서, 인증서를 취소할 경우 CA는 CRLs을 발행하는 것이 아니라 더 이상 인증서를 발행하지 않음으로써 인증서 취소 사실을 알리는 방법이다[5].
- Certificate-URL : 인증을 받고자 하는 통신 개체는 실제 인증서를 전송하는 것이 아니라, 인증서의 위치를 나타내는 *Certificate-URL*을 전송하고, 이를 수신한 상대방 통신 개체는 Certificate-URL에 따라, 인증서 저장소로부터 검출하여 인증한다[5].

# 4.안전한 쿠키를 이용한 사용자 인증 시스템

본 장에서는 사용자가 웹서버로 접근 시에, 사용자 인증을 위해서 쿠키를 사용하는 환경을 고려하여 기존의 안전하지 못한 쿠키를 대신해서, 암호 기술과 공개키 기반 구조를 적용한 쿠키 기반의 새로운 사용자 인증 시스템을 제안한다. 제안 시스템에서는 2.2절에서 언급한 쿠키의 취약점을 극복하기 위해서, 각 쿠키들의 쿠키값을 비밀키(secret key)로 암호화함으로써 기밀성을 제공하고, 쿠키를 해쉬 및 서명하여 무결성을 검증하고, 인증서를 통한 사용자와 서버의 상호 인증(mutual authentication)을 제공한다.

## 4.1 용어정리

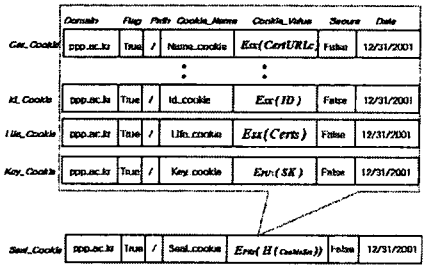
본 논문에서 사용되는 암호 프로토콜의 기술을 위해서 아래와 같은 용어를 사용한다.

- $C$ : 클라이언트를 나타내는 식별자.
- $S$ : 서버를 나타내는 식별자.
- $U$ : 사용자 ID를 나타내는 식별자.
- $PR_X$ : 공개키 암호 시스템을 위한, 통신 개체 $X$의 개인키(private key).
- $PU_X$ : 공개키 암호 시스템을 위한, 통신 개체 $X$의 공개키(public key).
- $SK$ : 대칭키 암호 시스템을 위한 공유키(shared

key)로서, 쿠키들의 쿠키값을 암호화하기 위해서 사용. 이 공유키는 일반적인 대칭키 암호 시스템인 DES, IDEA, RC5와 같은 알고리즘의 키로서 사용될 수 있다.

- $SCert_X$ : short-Lived Certificate.
- $CertURL_X$ : 통신 개체 $X$의 certificate-URL.
- $m$ : 전송되는 메시지를 나타내는 식별자
- $H(m)$ : $m$ 을 해쉬 처리한 것. 적용하는 해쉬 함수로, MD5, SHA-1등 사용될 수 있다.
- $E_A(m)$ : $m$을 암호화 키( $A$)로 암호화.
- $D_A(m)$ : $m$을 복호화 키( $A$)로 복호화
- $SIG_A(m)$ : $m$을 개인키( $A$)로 전자 서명
- $VER_A(m)$: $m$을 공개키( $A$)로 전자 서명 검증

본 논문에서 사용되는 안전한 쿠키(Secure Cookies)들의 전체적인 구조는 <그림 2>와 같다.




<그림 2> Secure Cookie 구조

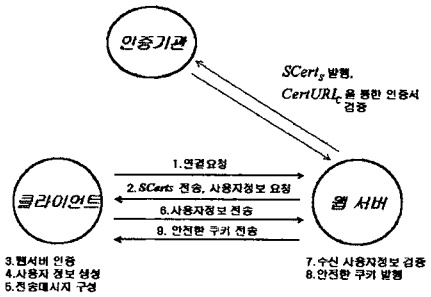
- Cer_Cookie : 서버가 클라이언트를 인증할 수 있는 $CertURL_C$로 Key_cookie의 $SK$를 사용하여, 암호화한 것을 가진다.
- Id_Cookie : 서버가 ID기반의 접근제어를 할 수 있는 서버내 쿠키 사용자의 ID( $U$ )값으로 $SK$를 사용하여 암호화한 것을 가진다.
- Life_Cookie : 쿠키 셋의 만기일을 $SK$를 사용하여 암호화한 것을 가진다.
- Key_Cookie : Cert_Cookie, Id_cookie, Life_Cookie 등의 쿠키값들을 암호화하기 위해 사용되는 $SK$를 서버의 공개키( $PU_S$)를 사용하여 암호화한 값을 가진다.
- Seal_Cookie : 위에서 생성된 쿠키들을 해쉬하여, 서버의 개인키( $PR_S$)로서 서명한 것을 가진다.

제안하는 안전한 쿠키는 중요한 사용자 정보를 $SK$로 암호화하고 Seal_cookie를 사용함으로써, 쿠키들이 네트워크로 전송되거나 클라이언트 시스템 내에서 보관 시에 공격자로부터의 정보 누출 및 변경 공격으로부터 안전할 수 있다.

## 4.2 안전한 쿠키의 발행

제안 방안에서의 쿠키 발행 과정을 <그림 3>에서 간략히 보이고 있다.

클라이언트가 웹서버로 접속을 시도하면, 웹서버는 클라이언트에게 쿠키를 요구한다. 이때, 클라이언트는 쿠키를 가지고 있지 않기 때문에, 웹서버는 자신의 short-lived certificate와 사용자 정보를 요청하는 메시지, $SCert_S \| Request\ user\ information$를 전송한다.




<그림 3> 안전한 쿠키 발행 절차

클라이언트는 웹서버의 short-lived certificate의 유효성을 검사한다. 만약 웹서버가 인증서를 전송하지 않았거나 유효 기간이 만료된 인증서일 경우 사용자 정보를 전송하지 않는다. 서버의 인증서가 적합하다고 판정되면, 웹서버에서 사용자 식별을 위해서 사용될 사용자 ID를 입력받고, 생성될 쿠키들의 쿠키값을 암호화하기 위해서 사용되는 $SK$를 생성하여, 다음과 같이 메시지를 구성하여 웹서버에게 전송한다.

$$E_{PU_S}( CertURL_C \| U \| SK \| SIG_{PR_C}( H( CertURL_C \| U \| SK ) ) )$$

위의 메시지를 수신한 웹서버는 수신한 자료를 자신의 개인키( $PR_S$)로서 복호화하고, 수신한 $CertURL_C$에 따라서 인증서 저장소로부터 클라이언트의 인증서를 획득해서 클라이언트를 인증하게 되며, 인증서내에 있는 클라이언트의 공개키( $PU_C$)를 사용하여, 수신한 메시지의 서명부분을 아래와 같이 검증한다.

$$VER_{PU_C}( SIG_{PR_C}( H( CertURL_C \| U \| SK ) ) )$$

그리고, 수신 메시지의 무결성을 검증하기 위해서, $CertURL_C \| U \| SK$ 값을 해쉬 처리하여, 서명내에 있는 해쉬값과 비교를 수행한다.

위의 모든 암호적인 검증 절차가 끝난 후에, 웹서버는 클라이언트를 인증하고 사용자( $U$ )를 위한 안전한 쿠키(secure cookies)들을 <그림 2>와 같이 발행한다. 안전한 쿠키의 발행 시에, Life_cookie의 쿠키값은 웹서버에 의해서 임의적으로 결정되며, 이 값은 전체 쿠키들의 만기일과 동일하게 설정이 된다. 그리고, 전송되는 쿠키들의 무결성을 검증하기 위해서 seal_cookie가 사용되며, 쿠키값을 아래와 같이 가진다.

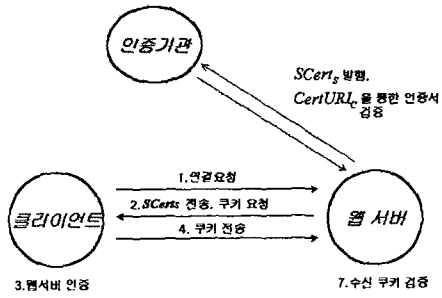
$$SIG_{PR_S}( H( Cer\_Cookie \| \cdots \| Key\_cookie ) )$$

## 4.3 안전한 쿠키를 사용한 사용자 인증

제안 방안에서의 안전한 쿠키를 사용한 사용자 인증 과정은 <그림 4>에서 간략히 보이고 있다.

클라이언트가 웹서버로 접속을 시도하면, 웹서버는 클라이언트에게 쿠키를 요구한다. 이때, 클라이언트는 수신한 메시지, $SCert_S \| Request\ user\ information$에 포함된 웹서버의 short-lived certificate를 통해서 서버

가 적법하다고 판단되면, 소유하고 있는 안전한 쿠키를 웹서버에게 전송한다.




<그림 4> 안전한 쿠키를 통한 사용자 인증

안전한 쿠키를 수신한 웹서버는 자신의 개인키($PR_S$)로서 Key_cookies의 쿠키값을 복호화하여, $SK$를 구한다. 구해진 $SK$를 통해서, 웹서버는 수신한 쿠키들의 쿠키값을 아래와 같이 복호화하여,

$$D_{SK}(E_{SK}(CertURL\_C)),\ D_{SK}(E_{SK}(U)),\ D_{SK}(E_{SK}(12/31/2001))$$

$CertURL_C$, $U$, $Life$(만기일)을 알게된다. 이 때, 웹서버는 $CertURL_C$에 따라서 인증서 저장소로부터 클라이언트의 인증서를 획득해서 클라이언트를 인증하게 된다. 그리고, 수신한 쿠키들의 무결성을 검증하기 위해서, 웹서버 자신의 공개키로 seal_cookie의 쿠키값을 아래와 같이 검증한다.

$$VER_{PU_S}(SIG_{PR_S}(H(Cer\_Cookie \| \cdots \| Key\_cookie)))$$

그리고, 수신한 쿠키들을 해쉬 처리하여, 서명내의 해쉬값과 비교하여, 두 값이 일치할 경우 웹서버는 수신한 쿠키값들이 변경되지 않았음을 알 수 있게 된다.

## 5. 구현방안

본 논문에서 제안한 안전한 쿠키를 통한 사용자 인증시스템의 구현 방법으로는 아래와 같이 클라이언트와 서버에서의 두 부분으로 구분할 수 있다.

- 클라이언트

제안 방안을 클라이언트에서 원활히 수행하기 위해서는 클라이언트의 브라우저에서 수행되는 부가적인 어플리케이션이 요구된다. 이 어플리케이션은 4.2절에서의 사용자정보를 생성하기 위하여, 사용자 ID를 입력, $SK$를 생성하는 기능, 사용자의 로컬 저장소에 위치한 certificate_URL을 읽는 기능, 사용자 정보 생성을 위한 암호 알고리즘 수행기능이 있어야 한다. 또한, 4.3절에서의 안전한 쿠키 전송을 위해서, 사용자의 로컬 저장소의 안전한 쿠키들을 읽을 수 있는 기능 또한 포함되어야 한다. 그리고, 안전한 쿠키는 클라이언트의 로컬 저장소에 저장되므로 실 사용자가 아닌 타인에 의해 쿠키가 이용되어 질 수 있다. 이를 방지하기 위해 안전한 쿠키의 자동적인 사용 권한을 부여하는 옵션(간단한 사용자 패스워드, 사용자 인증 문자열 이용)을 어플리케이션의 부가적인 기능으로 추가하는 것도 고려되어야 한다. 위의 기능을 수행하는 어플리케이션의 구현 환경으로는 현재 인터넷에서 많이 사용되는 Java Signed Applet과 브라우저의 Plug-in Program을 적용할 수 있다. 전자의 Signed Applet은 제한적인 Applet의 수행을 확장하여 수행할 Java Applet에 대한 서명을 통하여 인증성 및 무결성을 확보하는 방안으로, 정당한 서명이 아니라면 Applet 자체를 실행하지 않을 수 있고 정당하다면 기존 Applet과는 달리 로컬 저장소를 접근하는 권한을 허가하여 본 방안을 구현할 수 있다. 후자의 Plug-in Program은 본 방안을 수행하도록 브라우저의 기능을 확장하여 사용할 수 있는데, 제안 방안 이외의 여러 가지 새로운 보안문제가 발생할 수 있으므로 실제 구현시 Plug-in Program 자체에 대한 인증성 및 무결성, 안전한 배포 등이 고려되어야 한다.

- 웹서버

웹서버는 기존의 공개키 기반 구조를 기본적으로 지원하는 것과 별도로, 본 논문에서 사용되는 certificate-URL과 short-lived certificate 방안을 지원해야만 한다. 또한, 기존의 일반적인 쿠키 발행 절차는 제안 방안의 안전한 쿠키들을 생성하기 위하여 요구되는 암호 알고리즘의 추가와 같은 변경이 불가피하다. 실제 구현을 위해서는 기존의 CGI, Server-side Script 등이 적용가능하고, 안전한 수행을 위한 웹서버 보안 방안이 고려되어야 한다.

## 6. 결론

본 논문에서는 공개키 기반의 안전한 쿠키를 이용하여 기밀성, 인증, 무결성등 보안 서비스를 제공하는 안전한 쿠키 설계를 제시하였다. 본 논문에서 제시된 시스템은 인증서를 기반으로 하므로 클라이언트와 서버의 상호 인증을 제공하고, 서버가 쿠키 정보를 요청할 때 일반 쿠키처럼 사용이 가능하므로 사용자에게 투명성을 제공한다. 따라서, 전자상거래와 같은 사이트에서의 고객 정보를 안전한 쿠키에 저장함으로써 사용자의 편의성과 안전성은 물론 서버의 부담과 유지보수 등을 줄일 수 있다.

## 참고문헌

[1] http://www.certcc.or.kr/advisory/ka2000/ka2000-041.html

[2] http://www.netscape.com/newsref/std/cookie_spec.html

[3] Joon S. Park and Ravi Sandhu, "Secure Cookies on the Web" Park, J.S., Sandhu, R. IEEE Internet Computing, Volume : 4 Issue : 4, July-Aug. 2000, P.36-44

[4] Kyoungcheol Koo, Injune Jo, Seungheui Han, "The Design of Web Security System using the Cookie" CISC 2000, P.452-461

[5] WAP Forum, "Wireless Application Protocol Public Key Infrastructure Definition", 2000

[6] 이만영, 김지홍, 류재철 외 3명, "전자상거래 본안기술", 생능출판사, 1999