쿠키를 이용한 HTML 문서의 암호화 제안

한소희, 조동섭
이화여자대학교 컴퓨터과학
wourlde@gmail.com, dscho@ewha.ac.kr

Proposition of HTML Encryption Using Cookie

Sohee Han, Dongsub Cho
Dept. of Computer Science, University of Ewha Woman's

요 약

대부분의 인터넷 사용자들은 자신도 모르는 사이에 쿠키의 사용을 허용하고 있다. 또한 쿠키를 사용함으로써 인터넷 속도가 빨라진다는 사실도 아울러 간과하고 있다. 그리고 인터넷을 떠도는 모든 페이지는 HTML 문서로 표현된다. 따라서 누구나 웹 사이트의 소스에 쉽게 접근하여 트랙킹이나 해킹의 도구로써 사용할 수 있다. 본 논문은 필연적으로 사용하게 되는 쿠키를 키로 하여 HTML 문서의 암호화를 제안한다. 특히 웹 서버와 클라이언트 환경으로 한정하여 웹 서버가 클라이언트에게 암호화된 HTML 문서를 전송함으로써 제 3자로 하여금 클라이언트의 액티비티가 전혀 노출되지 않도록 한다.

1. 서 론

인터넷이 빠르게 보급된 이래로 쿠키는 tracking의 위험에도 불구하고 사용하지 않기 어려운 명사가 되었다. 먼저 쿠키는 웹 사이트에 의해서 클라이언트 PC에 쿠키 내용이 저장됨으로써 중복된 내용을 보낼 필요가 없어 HTML 문서의 전송속도를 높인다. 또한 전자상거래가 상업적 판매의 매우 큰 비중을 차지하고 있는 요즘 장바구니는 쿠키가 없다면 아마도 불가능했을 기능일 것이다. 따라서 쿠키의 차단 정도를 부분적으로 조절할 수 있는 기능이 추가되긴 했지만 모든 브라우저들은 임시 인터넷 파일을 통해 기본적으로 쿠키저장 기능을 제공하고 있다.

최근에는 전 세계적으로 블로그와 같은 개인 컨텐츠 중심의 사이트들이 크게 각광을 받았다. 사람들은 자신들이 가지고 있는 개인 컨텐츠에 대해, 또한 인터넷을 떠도는 수많은 지적 재산들에 대해 크게 의미를 부여하면서 함부로 자료를 가져가지 못하도록 자료의 소스보기 기능을 거부하길 요구했다. 이에 HTML 문서 암호화의 필요성이 대두되었다 .

* 이 논문은 한국학술진흥재단 이화여자대학교 BK21 사업의 지원으로 쓰여졌습니다.

도입에서 쿠키를 언급한 것은 HTML 암호화에 쿠키를 이용하고자 함이다. 본 논문은 고유한 쿠키값에 해쉬함수를 적용해 해쉬값을 얻은 후 그 해쉬값을 암호화의 키로 사용, HTML 문서를 암호화하는 것을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련연구를 언급, 3절에서는 HTML 문서의 암호화 방식을 제안하고, 4절에서는 결론과 향후 연구과제를 제시한다.

2. 관련연구

2.1 쿠키

쿠키는 클라이언트가 웹 서버에 접속할 때 웹 서버가 클라이언트 컴퓨터 내에 저장하는 클라이언트의 접속기록이다. 웹 브라우저는 임시 인터넷 파일을 통해 쿠키를 자동 생성해주기도 한다. 쿠키는 클라이언트가 접속한 기록뿐만 아니라 페이지의 컨텐츠 내용도 포함하고 있기 때문에 일종의 캐쉬와 같은 역할을 하게 된다. 또한 최근의 자동 로그인이나 인터넷 쇼핑몰의 장바구니와 같은 기능은 쿠키가 있어 가능한 기능이므로 쿠키를 완전히 사용하지 않은 것은 생각하기 어려운 일이 되어버렸다. 그러나 쿠키는

아이디나 패스워드 같은 개인정보도 포함하고 있어 신상정보 유출에 대한 문제가 발생하기도 한다. 대부분의 웹사이트들은 쿠키를 암호화해서 전송하지만 그렇지 않은 사이트들에 의한 쿠키는 심각한 정보유출의 가능성을 안고 있다.

2.2 HTML 문서의 암호화의 필요성

현재 웹 페이지를 구성하기 위해 쓰이는 HTML 문서는 일반적인 텍스트로, 웹 페이지의 모든 구성 내용을 전달하며 원본의 출처도 쉽게 읽을 수 있도록 되어있다. 따라서 누구에게나 모든 내용이 쉽게 공개되는데, 최근에는 HTML 문서의 일부 혹은 전체 내용을 암호화 함으로써 다른 사람에게 보이지 않길 원하는 경우가 많아졌다. 몇몇 HTML 인코더 툴들은 HTML 문서의 태그들을 인코딩하여 사람들이 내용을 읽을 수 없도록 하는 기능을 제공한다. 하지만 이러한 툴들은 단순히 문자들을 다른 문자로 매칭하는 방법으로 인코딩 하는 것이기 때문에 디코딩 또한 간단하다.

3. HTML 문서 암호화

3.1 웹 서버와 클라이언트 환경

많은 트랙커들은 웹 사이트에 접속하는 웹 유저들의 접속 기록이나 컨텐츠 내용을 추적하여 개인정보를 빼내갈 수 있다. 또한 HTML 문서상의 컨텐츠의 소스를 이용해 웹 상의 자료들을 무단 도용하고 이것이 디지털 저작권 침해 사례로 불거진 것은 비단 어제 오늘에만 발생하고 있는 문제가 아니다. 이러한 문제들은 클라이언트가 웹 서버로부터 받는 HTML 문서의 무방비 노출로부터 기인한다. 따라서 본 논문에서는 HTML 암호화의 제안을 웹 서버와 클라이언트 환경에 중점을 두고 설명하고자 한다. 다음은 연구에 사용하고자 하는 웹 서버와 클라이언트 프로그램, 해쉬와 암호화 알고리즘, 그리고 클라이언트가 웹 서버의 모든 HTML tag를 가져온 내용이다.

[표 1]

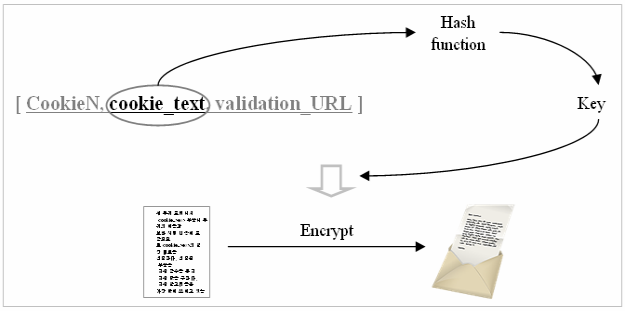
웹 서버 프로그램	MICROSOFT FOUNDATION CLASS LIBRARY : httpsvr
클라이언트 프로그램	MICROSOFT FOUNDATION CLASS LIBRARY : Spider
해쉬 알고리즘	SHA-1
암호화 알고리즘	DES

<!DOCTYPE html><html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"><title>Google</title><style type="text/css"><!--
#svc-tab .bgp-fr{background:transparent url(/ig/f/KoR1V7i-xBw/intl/ALL_kr/tab_sprite_all.gif) 0 no-repeat} #svc-toolbar .bgp-fr{background: transparent url(/ig/f/JB08UxZqEXw/intl/ALL_kr/svc_sprite_all.gif) 0 0 no-repeat} .ttv{background: transparent url(/ig/f/JB08UxZqEXw/intl/ALL_kr/svc_sprite_all.gif) 0 0 no-repeat}
--></style>
<script src="/ig/f/4GmrZsj9vDQ/ig.js"></script><link rel="stylesheet" href="/ig/f/-n3eOqyLOCA/intl/ALL_kr/homepage.css" type="text/css"></head><body onload="_KO.init()"> iGoogle|로그인 <img id="logo" alt="" src="http://img0.gmodules.com/ig/f/oKstlUEg20s/intl/ALL_kr/logo.gif"><form action="http://www.google.co.kr/search" name="f" ><script><!--
function qs(el) {if (window.RegExp && window.encodeURIComponent) {var ue=el.href;var qe=encodeURIComponent(document.f.q.value);if(ue.indexOf("q=")!=-1){el.href=ue.replace(new RegExp("q=[^&]*"),"q="+qe);}else{el.href=ue+"&q="+qe;}}return 1;}
// -->

[그림 1]

3.2 HTML 문서 암호화 과정

HTML 문서를 암호화 하기 위한 알고리즘으로는 가장 널리 알려진 DES 대칭키 알고리즘을 사용한다. 먼저 클라이언트가 웹 서버에 접속을 요청하면 웹 서버는 대칭키를 사용하여 요청된 HTML 문서의 내용을 암호화 하여 클라이언트에게 전송한다. 클라이언트는 전송 받은 HTML 문서를 역시 같은 대칭키를 이용하여 복호화한다. 이때 사용되는 키는 고유값인 쿠키를 사용한다. 이러한 전체 과정을 그림[2]로 표현하였다.



[그림 2]

3.3 대칭키

본 연구에서는 대칭키로써 쿠키를 사용한다. 쿠키는 클라이언트마다 unique하고 또한 쿠키의 길이는 최대 4byte 이내로 길지 않기 때문에 대칭키로 사용하기에 적당하다. 그러나 암호화된 쿠키 값 이라도 트랙킹의 위험이 있기 때문에 쿠키 값의 일부를 해쉬 함수를 사용하여 해쉬값으로 전환한 뒤 대칭키로 이용한다. 쿠키의 기본 포맷은 다음과 같다.

[CookieN(cookie_legnth, URL_length), cookie_text, validation_URL]

위 쿠키 포맷에서 cookie_text 부분에 쿠키의 이름과 보안 세팅 내용이 포함되므로 cookie_text의 일정 비트를 선택한다. 선택된 부분을 해쉬 함수를 통해 해쉬 값을 구한다. 해쉬 알고리즘은 가장 많이 쓰이고 있는 SHA-1 알고리즘을 사용한다.

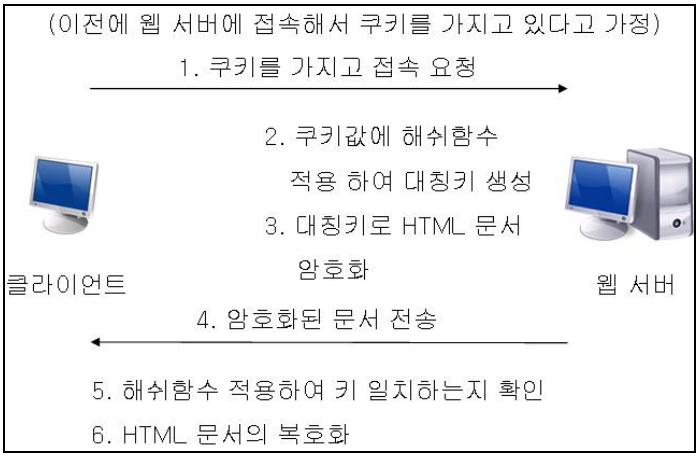
```
array<Byte>^data=gcnew array<Byte>( DATA_SIZE );
array<Byte>^ result;

SHA1^ sha = gcnew SHA1CryptoServiceProvider;
// This is one implementation of the abstract
    class SHA1.
result = sha->ComputeHash( data );
```

[Data의 SAH1 해시를 계산하여 이를 result에 저장]

3.4 웹 서버와 클라이언트 환경 HTML 암호화 시나리오

본 논문에서는 웹 서버와 클라이언트 환경에 초점을 두고 클라이언트가 어떤 페이지나 컨텐츠를 봤는지 노출되지 않기 위해 웹 서버가 클라이언트에게 HTML 문서를 보낼 때 그 문서를 암호화 하고자 한다. 이 전체 과정은 다음과 같다. 먼저 클라이언트가 웹 서버에 접속한 기록이 있어 쿠키를 가지고 있다고 가정한다. 그러면 클라이언트는 쿠키를 가지고 웹 서버에 접속을 요청한다. 웹 서버는 쿠키의 일부분(cookie_text)을 가지고 해쉬함수를 통해 해쉬값을 얻는다. 그리고 이 해쉬값을 대칭키로 사용하여 요청된 HTML 문서를 암호화 한다. 그리고 암호화된 HTML 문서를 클라이언트에게 전송한다. 클라이언트는 역시 해쉬함수로 대칭키를 얻어낸 다음, 웹 서버로부터 전송된 대칭키와 비교하여 일치하면 HTML 문서를 복호화한다. 이 과정을 [그림3]으로 표현하였다.



[그림 3]

4. 결론

본 논문에서는 쿠키를 대칭키로 사용한 HTML 문서의 암호화를 제안하였다. 대부분의 웹 서버는 클라이언트에게 쿠키를 남기고, 웹 사용자들은 그러한 쿠키를 고유하게 저장하고 있기 때문에 키로써의 역할로 적합하다. 그러나 쿠키는 쉽게 노출될 수 있는 위험을 안고 있기 때문엔 해쉬함수를 통해 해쉬값을 생성, 대칭키로 사용한다. 최근들어 HTML 문서의 암호화의 필요성이 대두되고 있다. 정보의 소스를 숨기기 위해서다. 따라서 웹 서버는 쿠키를 통해 얻은 대칭키를 사용해 클라이언트에게 보낼 HTML 문서를 암호화하고 클라이언트는 같은 대칭키로 HTML 문서를 복호화한다. 반드시 키를 알아야 복호화를 할 수 있으므로 클라이언트에게 보내지는 HTML 문서의

내용은 전혀 노출되지 않는다.

향후 연구로는 현재 연구 수행중인 웹 서버와 클라이언트 환경에서 제안한 해쉬함수를 통해 대칭키를 만드는 것과 암호화 알고리즘을 적용해 암호화된 HTML 문서의 전송 과정을 구현하는 것이다.

5. 참고문헌

[1] 최향창, 최은복, 노봉남, “쿠키 보호 시스템 설계”, 정보과학회, 2002년.

[2] 김정재, 박재표, 전문석, “동영상 데이터 보호를 위한 공유 키 풀 기반의 DRM 시스템”, 정보처리학회논문지 C 제12-C권 제 2호, 2005년.

[3] 김기성, 김광, 허신, “이기종 시스템에서 안저한 데이터 전송을 보장하는 웹 보안 모듈의 설계 및 구현”, 정보과학회 논문지 제32권 제 12호, 2005년.

[4] 최성욱, 김기태, “안전하고 신뢰성 있는 전자상거래를 위한 키보드 입력 보안 시스템의 설계 및 구현”, 정보처리학회논문지C 제13-권 제 1호, 2006년.

[5] David M. Kristol, " HTTP Cookies: Standards, privacy and politics", ACM Transactions on Interner Technology(TOIT), Volulme1, Issue2, 2001년.

[6] Daniel Lin, Michael C. Loui, "Taking the byte out of cookies: privacy, consent, and the Web", ACM SIGCAS Computers and Society, Volume 28 Issue 2, 1998.

[7] MSDN Library http://msdn2.microsoft.com/ko-kr/library/system.security.cryptography.sha1.aspx