WWW에서 프라이버시 보호와 익명성 제공

박영호, 이경현
부경대학교 전자계산학과

Privacy Protection and Anonymity Services for the WWW

Young-Ho Park, Kyung-Hyune Rhee
Dept. of Computer Science, PuKyong Nat'l University

요 약

WWW(World Wide Web)이 인터넷 사용영역의 대부분을 차지하면서 웹을 이용한 전자상거래가 인터넷의 새로운 활동영역으로 등장하였다. 이와 함께 마케팅 차원에서 웹 사용자들에 대한 신상정보와 개인성향에 대한 정보를 요구하는 사이트가 증가하고 있고, 이로 인해 점차 웹 사용에 대한 개인 프라이버시 보호와 익명성 제공에 대한 관심이 증가하고 있다. 하지만 현재의 HTTP 혹은 WWW의 구조는 부가적인 메커니즘이 없이 개인의 프라이버시 보호나 익명성 같은 서비스를 거의 제공해주지 못하고 있다. 본 논문에서는 현재 WWW에서 프라이버시 보호와 익명성을 제공하기 위한 기법들을 살펴본다.

1. 서론

최근 인터넷의 급속한 발전은 WWW, 즉 웹을 사용한 인터넷의 성장이라고 해도 과언이 아니다. 웹을 이용한 인터넷의 사용이 활발해지면서 웹이 정보수집을 위한 중요한 수단이 되었고, 또한 웹을 기반으로 하는 전자상거래가 성행하게 되었다. 하지만 웹의 기능들이 점점 편리해지는 만큼 사용자의 프라이버시 손실도 대등해지고 있다. 사용자들은 웹을 통해 무엇을 읽었고, 어디에서 쇼핑을 해서 무엇을 샀으며, 누구와 접촉했는지에 대한 정보를 남기게 되지만 이러한 사실을 인식하지 못하는 경향이 있다[9].

전자상거래를 위한 웹의 사용은 사용자의 쇼핑 습관이나 소비 패턴과 같은 개인의 행동성향을 노출시킬 수 있으며, 실제로 최근 조사자료에 의하면 고객관리와 마케팅 차원에서 제공되는 개인의 신상정보의 누출에 대한 우려가 가장 높게 나타났다[8].

이처럼 웹에서의 개인 프라이버시 보호와 익명성에 대한 요구가 증가하면서 웹브라우저나 웹서버 모두 자신의 IP 주소나 호스트 이름 등이 알려지지 않고 웹을 통해 정보를 제공해 주거나 제공받기를 원할 것이다. 하지만 현재의 WWW 구조는 이에 대한 기능을 제공해 주지 못하며, 이를 위한 부가적인 메커니즘이 제공되어야 한다.

브라우저의 익명성을 제공해주기 위한 기법으로는 웹서버와 브라우저 사이에 HTTP 프락시(proxy) 서버를 사용하는 방법이 있고, 웹서버의 익명성을 제공해 주기 위한 기법으로는 서버의 URL을 암호화해서 제공하는 방법이 있다. 또한 프락시 서버의 캐시(cache)기능은 서버로의 요청에 대한 트래픽 분석(traffic analysis)을 어렵게 할 수도 있다.

본 논문에서는 WWW에서 익명성을 제공해 줄 수 있는 메커니즘들을 웹 브라우저와 웹 서버의 측면에서 각각 살펴보고 논의하도록 한다.

2. 웹 클라이언트의 익명성을 위한 기법

이번 장에서는 웹 사용자의 프라이버시를 보호하고 클라이언트의 익명성을 제공하기 위한 기법들에 대해 살펴본다.

2.1 HTTP 프락시 서버를 이용한 익명성 제공

가장 일반적으로 사용할 수 있는 기법으로, HTTP 프락시 서버를 이용해서 클라이언트의 익명성을 제공해 줄 수 있다[1]. 클라이언트가 직접 웹 서버로 자원에 대한 요청을 전송하는 대신에 http://proxy.com:80

/http://origin.web.server과 같은 중첩된 URL 형태로 지정된 프락시 서버의 포트로 연결하면, 프락시 서버가 해당 웹 서버의 URL로 요청하게 된다. 이때 웹 서버로 전송되는 요청에는 클라이언트의 정보가 프락시의 정보로 대체되므로 웹 서버는 실제 클라이언트는 알지 못하고 단지 프락시만 인식하게 된다. 그러므로 웹 서버는 요청에 대한 응답을 프락시로 전송하고, 프락시는 수신한 응답을 다시 해당 클라이언트로 전송한다.

실제로 Anonymizer 서비스가 사용자의 익명성을 위해 프락시 서버를 사용해서 서비스를 제공한다. 예를 들어, Anonymizer를 이용해서 www.yahoo.com에 접속하고자 한다면 브라우저의 URL로 http://www.anonymizer.com:8080/www.yahoo.com을 입력하면 Anonymizer의 프락시 서버가 야후의 페이지를 받아서 넘겨준다.

프락시 서버를 사용함으로써 웹 서버에 대해 사용자의 IP 주소나 도메인 이름 정보를 숨김으로써 익명성을 제공해주지만 적어도 두 가지 고려해야 할 사항이 있다. 첫 번째로, 프락시 서버에는 사용자들의 모든 요청에 대한 기록(log)이 남으므로, 사용자는 프락시 서비스 제공자가 자신의 정보를 누출하지 않도록 신뢰되어야 한다. 또한 프락시는 웹 서버에 대해 사용자의 실체를 숨기기에는 적합하지만, 사용자에게 웹 서버의 실체를 감추기에는 적합하지 않다.

2.2 익명의 연결을 위한 onion 라우팅

익명의 연결(Anonymous connections)은 도청(eavesdropping)이나 트래픽 분석(traffic analysis)으로부터 보호하기 위한 방법이다. US Naval Research Laboratory(NRL)은 익명적인 연결을 구현하기 위해 Chaum mix network의 개념을 적용한 'Oinion routing'이라는 모형을 제안했다[6]. 여기에서 'onion'은 계층적으로 암호화된 메시지(layering encrypted message)를 의미한다.

Onion 라우팅은 onion 라우터들과 어플리케이션 프락시로 구성된다. 브라우저(또는 Initiator)는 HTTP 요구를 해당 웹사이트(responder)로 직접 전송하지 않고 onion 라우팅 프락시 서버로 전송한다. 프락시는 계층화된 데이터 타입인 onion을 구성하면서 onion 라우터들로 임의의 경로를 설정하고, 구성된 onion을 전송한다. Onion의 각 계층은 해당 라우터의 키로 암호화되고, 경로상의 다음 홉에 대한 정보와 키를 포함한다. 각 onion 라우터는 인접한 라우터만 확인할 수 있으며, 암호화를 사용하기 때문에 익명의 연결을 통해 전달되는 데이터는 각 라우터상에서 서로 다르게 표현된다. 그림 1에서 보듯이, 각 라우터를 거치면서 암호화된 계층이 하나씩 제거되며, 최종적으로는 웹사이트의 onion 라우팅 프락시 서버가 수신된 요구를 평문형태로 웹사이트에 전달한다. 웹사이트에서 브라우저로 전송되는 데이터에 대해서는 역순으로 데이터의 계층화가 발생한다.

그림 1. Onion 라우터에서 데이터 전달 형태

Onion 구성을 위한 암호화된 메시지의 계층적 형태는 링크 계층에서의 암호화에 비해 암호학적 오버헤드는 동일하지만 데이터 보호에 있어서는 이점을 제공한다. 링크 계층의 암호화에서는 어느 한 라우터가 손상을 입으면 데이터에 대한 모든 정보가 누출 될 수 있지만, onion 라우팅에서는 각 라우터에서 표현되는 데이터가 각각 다르므로 이러한 문제가 발생하지 않는다. 그리고 onion 라우팅은 응용 계층 아래에서 구현되므로 telnet이나 ftp 같은 서비스에도 적용될 수 있다. 하지만 onion 라우터를 구성함에 있어서, 라우터들간의 링크가 동시에 도청될 수 없도록 고려되어야 한다.

2.3 익명을 제공하는 웹 에이전트 시스템 - Lucent Personalized Web Assistant

사용자들이 웹사이트를 접근하기 위해 자신의 계정을 해당 사이트에 등록해야 하는 경우가 있는데, 웹 사용자가 계정등록을 위해 제공하는 부가적인 정보들은 HTTP나 쿠키의 특성으로 인해 자신도 모르게 정보가 누출되는 현상이 발생할 수도 있다[5, 9].

Lucent Technologies는 간편하고, 안전하면서 익명의 서비스를 제공하기 위한 LPWA(Lucent Personalized Web Assistant) 시스템을 개발하였다. LPWA는 웹 사용자를 대신해서 웹 서버와 상호 동작하는 일종의 에이전트(agent)로서, 사용자의 이름, 패스워드, e-mail 주소를 각 웹사이트에 대해 익명(pseudonym) 혹은 가명(alias)으로 생성해서 해당 사이트에 제공하고 이를 관리한다.

사용자는 브라우저의 프락시로 lpwa.com:8000을 설정하고 계정을 등록하기를 원하는 웹사이트로 접속하면, lpwa가 사용자와 웹사이트 중간에서 실제 사용자의 이름과 패스워드를 익명으로 대체하여 웹사이트로 전송하므로, 실제 웹 서버의 로그에는 lpwa에 의해

제공된 정보만이 기록된다. 그러므로 LPWA는 사용자들이 각 사이트에 대한 자신의 계정을 기억해야 하는 부담을 덜어주고, 가명을 사용함으로 인해 사용자의 실제 신원이 노출되지 않도록 함으로써 개인화된 웹 브라우징과 익명성을 동시에 제공해 준다.

익명성을 제공하기 위한 위와 같은 서비스는 가명 생성을 위한 이름 변환(naming translation)이 신중히 고려되어야 한다. 가명은 익명성(anonymity), 비밀성(secrecy), 일관성(consistency), 그리고 유일성(uniqueness)을 만족해야 한다. 이러한 요구를 만족하기 위해 Janus 함수라는 충돌회피 일방향 해쉬 함수를 사용하고 있다. 실제로 LPWA는 사용자 이름과 패스워드 그리고 웹사이트 이름을 Janus 함수의 입력값으로 사용한다.

2.4 사용자 그룹을 이용한 익명성 - Crowds

Crowds는 WWW에서 사용자의 익명성을 보호하기 위해 AT&T Research에 의해 개발된 시스템이다[7]. Crowds "개인의 존재를 대중들 속에 감춘다(blending into crowds)"라는 개념을 적용한 시스템으로, 멤버들 간에 암호화된 링크가 존재하는 프락시 서버들이 연결된 형태이며. 사용자들의 그룹을 crowd라 한다.

각 사용자는 jondo(John Doe에서 유래)라는 HTTP 프락시를 통해 표현된다. 사용자는 먼저 crowd에 참여(join)해야 하고, 브라우저의 요구는 jondo를 통해 전달된다. 사용자의 jondo가 브라우저의 요구를 수신하면 crowd 내의 임의의 다음 jondo를 선택하고 사용자의 요구를 전송한다. 이를 수신한 jondo는 해당 요구를 다음 jondo로 전달할지 또는 웹서버로 바로 전달할지를 결정한다. 따라서 브라우저의 요구는 임의의 jondo들로 구성된 경로를 통해 웹서버로 전달되고 이후의 브라우저의 요청들은 동일한 경로를 따라 웹서버로 전송되며, 서버의 응답도 동일한 경로를 따라 되돌려 진다.

두 jondo들간의 모든 통신은 이 두 jondo들 만이 알고 있는 키를 사용해 암호화되며, 암호화 키들은 jondo가 crowd에 참여할 때 정해진다. 그러므로 그룹 멤버의 참여와 관련해서 그룹 멤버십 프로시저가 제공되어야 하며, blender라는 그룹 관리 시스템이 이를 담당한다. 사용자가 ID와 패스워드를 이용해서 계정을 설정함으로써 crowd에 참여하면, blender는 공유키를 생성해서 등록한 사용자들의 패스워드로 암호화해서 각 jondo들에게 전달하고 새 멤버의 참여를 알리게 된다. 여기서 고려되어야 할 사항으로 blender는 키 분배와 멤버 관리를 위해 신뢰되는 시스템이어야 한다는 것이다. Crowds에 대한 보안성과 성능에 대해서는 [7]에서 자세히 분석되었다.

3. 웹 서버의 익명성을 위한 기법들

현재의 WWW 구조는 웹 자원을 찾기 위해 URL에 서버의 정보가 포함되어야 하므로 웹 서버의 익명성을 제공하지 못한다. 이번 장에서는 익명의 웹 서버를 운영하기 위한 기법으로 서버의 URL을 암호화하는 JANUS와 rewebber 시스템에 대해 살펴본다.

3.1 JANUS 시스템

JANUS는 독일의 Forschungsinstitut Tele kommunikation(FTK) of Dortmund, Hagen과 Wuppertal이 공동으로 참여한 연구 프로젝트로서, web 서버와 브라우저에 대한 익명성을 동시에 제공해 준다.

JANUS는 브라우저의 익명성을 제공해주기 위해 Anonymizer와 유사한 proxy 서버로 동작하고 웹 서버의 익명성을 제공하기 위해 URL의 웹 서버에 대한 참조 부분을 공개키 암호방식을 사용해서 암호화하고 복호화 한다. 웹서버의 URL은 암호화된 형태로 공개되며, 웹 서버에 접근하고자 하는 사용자는 해당 웹 서버의 암호화된 URL을 이용해서 다음과 같이 JANUS 시스템에 요청을 한다.

http://janus.fernuni-hagen.de/janus_encrypted/encrypted-web-URL

JANUS 시스템은 암호화된 부분을 복호화 하여 해당 서버로 브라우저의 요청을 전송하고, 해당 웹 서버의 응답에서 웹서버의 주소부분을 다시 암호화하여 요청한 브라우저에게 전송된다.

JANUS 시스템을 사용함으로써 서버의 IP 주소나 호스트 이름을 숨길 수 있는 이점을 제공하지만 단지 URL만이 암호화되므로 실제 데이터 스트림에 대한 보안성은 제공하지 않는다.

3.2 Rewebber network과 TAZ 서버

Rewebber는 JANUS 시스템의 기능에 데이터 트래픽으 암호화 기능을 추가한 프락시 서버로, Ian Goldberg와 David Wagner에 의해 연구되었다[4]. 실제로 요구되는 URL은 http://proxy.com/!RFkK4J... 형태로 나타나며, 암호화된 부분은 '!'표시로 시작된다. Rewebber에서는 이 암호화된 부분을 실제 URL에 대한 locator라 한다. 이러한 기법은 하나이상의

rewebber들을 연결(chaining)해서 반복적으로 수행될 수 있으며, 예를 들어, A, B, C 세 개의 rewebber가 연결된 경우 그림 2와 같이 중첩된 URL 형식으로 표현된다.

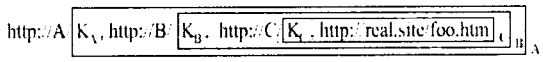

그림 2. 길이가 3인 rewebber 연결의 URL 형식

이러한 방법으로 전송 데이터에서는 실제 웹 서버의 위치를 숨길 수 있지만, 일단 브라우저가 rewebber를 통해 자원을 받았다면 WWW 검색 엔진을 사용해서 실제 사이트를 찾을 수 있는 문제점이 있으므로, [4]에서는 DESX 키로 서버에서 자원을 암호화해서 저장하는 방법으로 해결하고 있다. 서버는 각 rewebber의 DESX 키를 생성하여 자원을 다중으로 암호화하고, 그림 2에서 보듯이 locator에 DESX 키를 포함하여 rewebber가 문서를 복호화 할 수 있게 한다.

Rewebber들의 연결을 사용함으로써 완전히 복호화된 데이터는 브라우저로 연결되는 rewebber만이 알 수 있고, 실제 서버의 위치는 해당 서버로 연결되는 rewebber만이 알 수 있다.

그러나 rewebber network에는 중요한 단점이 있는데, 실제로 브라우저에서 rewebber를 통해 요청하는 locator는 아래와 같은 형태가 될 것이고, 이러한 형태로 서버의 이름을 사용하기에는 부적절하다.

http://rewebber.com/!RjVi0rawjGRT50ECKoBa7Qv3FJlRbyejh10gvpPA . . . . . rOIndpz5xgwZKc=/

이를 해결하기 위한 방법으로 [4]에서 .taz (Temporary Autonomous Zone) 라는 가상 이름공간(Virtual name space)을 제안했고, 이 이름공간을 위한 TAZ 서버를 개발했다. TAZ 서버는 .taz로 끝나는 가상 호스트 이름을 rewebber locator로 연관시키기 위한 공용 DB로 구성되며, TAZ 서버 관리자가 DB에 저장된 locator를 복호화 할 수 없어야 한다.

4. 결론

WWW을 사용한 인터넷 활성화로 웹이 정보수집 및 전자상거래의 주요 수단이 되었고, 이로 인해 개인 프라이버시에 대한 문제가 온라인 비즈니스의 신뢰성과 맞물리면서 더욱 중요한 문제로 부각되었다. 본 논문에서는 WWW의 사용과 관련해서 개인의 프라이버시 보호와 익명성을 제공하기 위한 메커니즘들을 클라이언트의 웹 사용과 서버의 웹 운영측면에서 살펴보았다.

클라이언트의 웹 사용과 관련해서는 HTTP 프락시 서버나 anonymous connections을 사용할 수 있고, 서버의 웹 운영과 관련해서는 서버의 URL을 암호화해서 공개하는 방법을 사용할 수 있다.

현재 사용자의 프라이버시와 익명성을 제공하기 위해 W3C는 P3P(Platform for Privacy Preference)[3]를 연구중이다. 그리고 HTTP 상태 제어 메커니즘을 위해 사용되는 쿠키(cookie)가 사용자의 정보 누출의 가능성으로 인해 안전하지 못한 것으로 간주되므로, 암호학적 기법을 사용한 안전한 HTTP 상태 제어 메커니즘에 대한 연구도 수행되고 있다[5].

[참 고 문 헌]

[1] A. D Rubin, D. Geer, M. J Ranum, Web Securtiy, source book, John Wiley & Sons Inc.

[2] E. Gabber, P. Gibbson, Y. Matias, A. Mayer, How to make personalized web browsing simple, secure, and anonymous, Proc. Financial Cryptography, Feb. 1997

[3] http://www.w3c.org/p3p

[4] I. Goldberg, D. Wagner, TAZ Servers and the Rewebber Network: Enableing Anonymous Publishing on the World Wide Web, CS 268 Final Report, UC Berkeley, 1997

[5] J.S. Park, R. Sandhu, Secure Cookies on the Web, IEEE Internet Computing, Volume: 4 Issue: 4, July-Aug. 2000

[6] M. G. Reed, P. F. Syverson, D. M. Gold Schlag, Anonymous Connections and Onion Routing, IEEE Journals on Selected Areas in Communications, Vol.16, No.4, May, 1988

[7] M. K. Reiter and A. D. Rubin, Crowds: Anonymous Web Transactions, ACM Trans. Information Systems Security, Apr, 1998

[8] National Consumers League, http://www.natlconsumersleague.org/pressessentials.htm

[9] Rubin, A.D. Geer, D.E., Jr. A Survey of Web Security, IEEE Computer , Volume: 31 Issue: 9, Sept. 1998