情 報 保 護 學 會 誌
第17卷 第6號, 2007. 12

환경독립적인 클라이언트 PKI 툴킷에 대한 기술 검토

김기영*, 권태경**

요 약

오픈웹(OpenWeb)으로 인하여 운영체제나 브라우저에 독립적인 클라이언트 PKI툴킷 지원에 대해 짧은 기간 동안 다양한 관계 기관에서 검토가 이루어졌다. 그러나 PKI 툴킷은 개인 정보 보호에 따른 보안과 밀접한 관계가 있으므로 단순히 소비자 권익만을 생각하거나 서비스 제공업체의 입장만을 고려하여 제공할 수 없다. 즉 보안 요구 사항 및 제한된 인적, 시간적, 재정적 환경을 고려를 할 경우 지원해야 할 대상 환경과 기술에도 제약이 따를 수 있다. 본고에서는 웹 브라우저의 확장 기능으로 제공되고 있는 PKI 툴킷에 대해 다양한 운영체제 및 브라우저 환경에서의 서비스 지원에 대해 살펴보고 제약 요소를 고려한 적절한 대응방안을 고려한다.

Ⅰ. 서 론

2006년 초 오픈웹(OpenWeb)은 특정 운영체제와 특정 웹브라우저에 국한되어 공인인증서 서비스 제공에 대해 정부 기관과 금융결제원 등의 개선을 요구하고 있다.[1] 이에 따라 운영체제나 브라우저에 상관없이 웹사이트에 누구나 접속할 수 있도록 본격적인 환경 독립적인 클라이언트 PKI 툴킷에 대한 기술 검토가 시작 되었다.

2000년 전후, 네스케이프(Netscape)를 지원하는 인터넷 뱅킹을 개발하여 서비스를 제공하였다. 그러나 네스케이프를 통한 실질적 국내 사용자 접속 수는 2001년에서 2003년 사이 1년에 1 ~ 2건으로 서비스는 거의 사용이 되지 않았다. 이는 다양한 인터넷 브라우저 환경을 지원할 필요가 없다는 것을 의미하는 것은 아니다. 국내의 웹 브라우저는 마이크로소프트사의 윈도우즈(Windows)의 인터넷 익스플로러(IE) 환경으로 변경되었으며 일부 웹페이지는 IE에 대해서만 지원을 하게 되어 사용자에게 다른 브라우저를 선택할 기회가 없음을 의미한다.

오픈웹에서는 정부의 정책 오류나 공공기관의 위법 행위에 원인이 있다고 판단을 하여 민원 및 소송을 제기하고 있으나 이는 시장의 선택이 가장 큰 영향을 미친 것으로 보인다. 정부는 공공부분에 대해서 오픈웹의 요구에 의해 개선을 고려하고 있으나 몇 가지 문제점이 존재한다. 특히 서비스 제공에 있어 가장 크게 고려해야 할 것은 개인 정보 보호이다. 최근에는 단순히 네트워크 상에서 이루어지는 공격에 대한 방어나 노출만이 아니라 PC상에서 이루어지는 키보드 보안, DRM, 인증서 사용 등 보안이 요구된다. 이 중 PC 보안 기술은 해당 PC의 운영체제에 의해 구현되어 있으므로 오픈웹의 요구사항 대로 단기간에 지원될 수 없는 경우도 있다.

본고에서는 인터넷 환경의 변화를 살펴보고 웹브라우저의 확장 기능으로 제공되고 있는 PKI 툴킷을 바탕으로 다양한 운영체제 및 브라우저 환경에서 서비스를 지원할 수 있는 방안을 제안하고자 한다.

Ⅱ. 인터넷 환경의 변화

2.1. 마이크로소프트사의 윈도우즈

1980년대 말 폐쇄적인 정책을 기본으로 한 애플사(Apple)의 맥(MAC)에 비해 IBM은 마이크로소프트사의 DOS 운영체제를 기반으로 한 PC를 보급함과 동시에 하드웨어적으로 호환 가능한 스펙을 공개하였다. 이

본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음.
[IITA-2007-S-601-01, 자기통제 강화형 전자ID지갑 시스템 개발]
* 소프트포럼 SW연구개발실 연구소장(kiyoung@softforum.com)
** 세종대학교 컴퓨터공학부 컴퓨터소프트웨어과 부교수(tkwon@sejong.ac.kr)

러한 IBM의 공개 정책은 유럽과 아시아를 비롯한 많은 나라에서 쉽게 선택하여 사용자층이 확대되었으며 저렴하게 하드웨어를 구입할 수 있게 되었다. 그러나 북미에서는 애플사의 맥이 사용자 인터페이스(GUI) 기반의 편리한 운영체제를 제공함으로써 비싼 가격에도 불구하고 상당한 시장 점유율을 기록했다.

마이크로소프트사는 맥과 유사한 사용자 인터페이스를 제공하는 윈도우즈 운영체제를 개발, 보급함으로써 윈도우즈를 사용하는 사용자가 급증하게 되었다. 최근에 Apple의 Mac이 가격이 낮아짐에 따라 상당한 판매 증가가 있었으나 사무용 OS를 비롯한 다양한 소프트웨어가 윈도우즈 운영체제 위주로 지원하게 됨으로써 마이크로소프트사 윈도우즈의 위치는 더욱 견고해 졌다. 최근 유럽에서도 맥의 시장 점유율 10%대로 떨어져 미국을 제외한 대부분의 나라에서 윈도우즈 운영체제를 이용하고 있으며 가격효율성(Cost-Effective)을 고려하여 저렴하게 보급 가능한 윈도우즈 기반의 PC를 주력으로 도입하고 사용하게 되었다.

2.2. 인터넷 익스플로러

인터넷이 널리 사용되기 시작한 1990년대 초반에는 네스케이프 브라우저를 이용하여 웹 서비스를 제공받거나 개인적인 브라우징, 업무 등을 활용하였다. 1990년대 말 마이크로소프트사는 인터넷 익스플로러의 기능을 지속적으로 갱신하면서 사용자들에게 제공하였으나 네스케이프는 상대적으로 업그레이드에 소홀히 하여 사용자들로부터 점점 외면을 받게 되었다. 또한 W3C에서는 지속적으로 웹 관련 표준을 제정하였으며 이에 마이크로소프트사는 적극적으로 수용하여 인터넷 익스플로러의 기본 기능 외의 ActiveX, BHO, Pluggable Protocol 등 다양한 확장 기능을 부여하였다.[5,6] 따라서 웹을 통하여 다양한 서비스를 제공하고자 하는 서비스 제공자들은 확장기능을 가지고 있는 인터넷 익스플로러를 가장 적합한 브라우저로 인식하게 되었으며 인터넷 익스플로러 위주로 서비스를 제공하기 시작하였다.

국내 인증 시스템 구축에 있어 미국의 보안 정책으로 인하여 Triple-DES와 같이 비도가 높은 암호 알고리즘을 적용할 수 없었으며 국내 환경에 적합한 암호 알고리즘을 개발, 제정하게 되었다. 따라서 웹 브라우저에 적용하여 사용할 수 없게 되었으며 이는 확장기술을 통한 구현이 필요하게 되었다. 마이크로소프트사의 인터넷 익스플로러는 다양한 확장 기능을 제공하고 있었으며 브라우저 시장에서 네스케이프를 압도적으로 앞서고 있었으므로 마이크로소프트사의 종속적인 기술들을 많이 사용하여 구현하였다. 따라서 익스플로러의 국내 점유율은 다른 브라우저에 비해 높은 비율을 보이게 되었다.

Ⅲ. 브라우저 기능 확장 기술

브라우저의 확장 기술은 1990년도 중반부터 도입되기 시작하였다. 그러나 W3C의 표준을 살펴보면 단순 열람 위주로 되어 있으며 보안상의 이유로 스크립트만으로는 하드디스크를 포함한 PC의 로컬시스템에 접근할 수 없어 이를 필요로 하는 서비스를 제공할 수 없다. 또한 상업적인 목적으로 사용되는 경우 자바스크립트만으로 구성되는 웹페이지는 스크립트 소스에 포함된 데이터를 보호할 수 없으며 이 스트립트 자체도 보호할 방법이 없다. 이와 같이 표준적인 기능만으로는 시각적 효과를 극대화하기 어렵고 기능에 제약이 있기 때문에 브라우저의 확장 기능의 사용이 필요하다. 그러나 브라우저마다 확장 기술에 차이가 있어 모든 브라우저 환경에서 확장기능이 동작하게 하기 위해서는 각 브라우저와 운영체제에서 동작하는 확장기능을 각각 만들어야만 한다. 이러한 다양성의 문제점을 해결하기 위해서 운영체제와 브라우저에 독립적인 자바 애플릿을 사용할 수도 있다. 본 장에서는 각 운영체제에서 지원하는 브라우저 확장 기술을 살펴보고 확장 기술의 사용시 발생하는 문제점에 대해 고려한다.

3.1. 액티브 엑스(ActiveX)

마이크로소프트사는 Java, C++와 같은 객체 지향 프로그램(Object oriented program)이 SW지적재산권 등의 문제로 인하여 현실적으로 효과나 활용성이 미미할 것이라는 것에 판단하여 코드의 재사용이 아닌 컴포넌트 재사용에 초점을 맞추었다. 이를 바탕으로 마이크로소프트사는 액티브 엑스를 개발하여 COM, OLE 등의 기술을 만들어 널리 활용할 수 있게 제공하였다. 따라서 마이크로소프트사가 개발한 많은 제품들은 이 기술들을 사용하여 구현되었으며 제3자의 컴포넌트들도 이용할 수 있게 함으로써 일반 개발자들도 널리 사용하게 되었다. 인터넷 익스플로러에서도 액티브 엑스를 지원하여 일반 업무용으로 만들었던 컴포넌트들을 그대로 브라우

저상에서 사용 가능하도록 제공하였으며 PC용 프로그램들도 약간의 수정만으로도 액티브 엑스로 변형할 수 있게 됨에 따라 액티브 엑스는 더욱 활성화 되게 되었다.

3.2. 자바 애플릿(Java Applet)

자바 애플릿은 1995년 자바 언어(Java)에 반영된 기술로 다른 프로그램 내에서 동작하는 컴포넌트 프로그램이다. 기본적으로 자바의 특성을 그대로 따르게 되어 자바의 장점과 단점을 모두 가지고 있다. 즉 자바 애플릿은 운영체제에 종속되지 않고 자바 가상 머신(Java Virtual Machine)이 설치된 곳에서 어디서나 재 컴파일 없이 실행이 가능하다. 또한 메모리 관리를 직접 수행하므로 사용자의 오류로 인한 메모리 오류는 거의 발생하지 않는다.

그러나 자바 코드를 실행하기 위해서 필요한 자바 가상 머신이 사전에 설치되어 있어야 한다. 만약 설치되어 있지 않는 경우 자바 가상 머신의 다운로드가 필요한데 이 부분에서 적지 않는 부담이 발생하게 된다. 또한 자바 가상 머신의 버전이 다양하여 개인 PC에 설치된 자바 가상 머신에 따라 지원해야 하므로 이는 서비스 제공자에 큰 부담이 되고 있으며 가동 시간이 길어져 사용자의 불편을 초래할 수 있다.

일반적인 애플릿의 경우 로컬 PC의 자원에 접근이 제한되어 있으므로 로컬 PC의 자원에 접근하기 위해 서명된 애플릿(Signed Applet)을 사용하여야 한다. 현재 일부 은행에서는 자바로 개발된 뱅킹 클라이언트를 배포해서 사용자 서비스를 제공하고 있으나 인터넷 익스플로러의 액티브 엑스에 비해 사용자의 불만이 많이 제기된다고 한다.

자바 애플릿의 사용에 있어 또 다른 문제점으로 사용자 인터페이스 제공을 들 수 있다. 자바는 운영체제에 독립적으로 이용할 수 있으나 해당 운영체제에서 적절한 사용자 인터페이스를 제공하지 않는다. 따라서 다양한 사용자 인터페이스를 중요시 하는 사용자에게는 외면을 받고 있는 실정이다. 사용자 인터페이스의 기능 외의 사용자가 요구하는 기능이 필요하나 이 부분 역시 지원이 어렵다는 문제점을 가지고 있다.

이러한 문제점 중에서 가장 고려해야 할 부분은 자바 애플릿이 코드 디컴파일이 가능하다는 점이다. 자바는 운영체제에 독립적이기 위해서 바이트 코드를 사용을 하기 때문에 코드를 디컴파일 하여 모든 소스 및 데이터를 볼 수 있다. 유사한 개념을 사용하는 Adobe의 Flash 역시 동일한 문제점을 내포하고 있다.[11] 물론 소스 코드를 공개하는 것은 암호 알고리즘의 안전성에 영향을 미치지는 않으나 암호화를 위한 키 관리나 인증서 패스워드 처리, 인증 값 전달 등의 보안상의 문제가 발생할 수 있다. 즉 코드 디컴파일 후 약간의 수정을 통해 다시 컴파일 하여 기존의 모듈과 교체를 할 수 있으므로 이는 해킹툴로 사용할 수 있다는 것이다.

오픈웹에서는 모듈 교체에 대한 부분은 이미 보안이 무너진 상태이므로 논외라는 설명을 하나 현재 금융권에 적용되고 있는 보안의 대부분은 위의 같은 상황을 전제로 보안 제품을 도입하고 있다. 즉 키보드 보안(Anti Key Logger)이나 안티스팸(Anti spam) 제품 모두 이미 PC안에 악성 코드나 해킹 도구가 존재한다는 가정하에 제품을 개발하고 있으므로 디컴파일이 된다는 특성은 보안적인 취약점을 내포하는 것을 의미한다.

따라서 자바는 서블릿(Servlet)과 같은 형태의 서버용 프로그램은 널리 활용되고 있으나 PC용 프로그램이나 애플릿은 상업적인 용도로 사용한 예를 찾아보기 어렵다.

만약 무료로 만들어서 배포하는 프로그램이라면 자바 애플릿의 도입을 고려해볼 수도 있으나 서비스의 안정성이 중요시 되고, PKI 툴킷과 같이 서비스 내용이 문제가 발생할 경우 사회적 문제를 초래할 수 있는 경우에는 지속적인 소프트웨어 갱신이 필요하며 문제발생 시 즉각적으로 조치가 요구된다. 따라서 PKI 툴킷과 같은 서비스를 제공하는 프로그램은 인력과 비용이 소요되게 되며 무료로 제공하기 어려운 점을 가지고 있다.

3.3. 네스케이프 플러그인(Netscape Plugin)

플러그인은 애플, 네스케이프, 어도비(Adobe) 등에서 특정 프로그램에 확장 기능을 제공하기 위하여 도입된 개념이다. 네스케이프 버전 3.0은 마이크로 소프트 사의 인터넷 익스플로러와 경쟁하기 위해 WYSIWYG 기능과 다양한 플러그인을 도입하였다. NSAPI를 제공하여 플러그인을 개발할 수 있도록 하였으며 이는 JNI와 유사한 LiveConnect라는 기술로 자바 스크립트를 이용하여 플러그인을 조작할 수 있도록 하였다.[9] 초기에 제공된 플러그인은 주로 HTML 상에서 지원되지 않는 그래픽과 사운드 등의 부가 기능을 제공하는데 사용되었다. C나 C++로 개발할 수 있도록 되었기 때문에 브라우저가 실행되는 운영체제의 사용자 인터페이스를 그대

로 사용할 수가 있다. 또한 DLL 또는 Shared Object /Library를 로드라는 형태이므로 동작 속도도 빠르다. 이 기술은 네스케이프 5.x까지 사용되었다.

그러나 네스케이프의 개발 정책에 따라 대폭적인 네스케이프의 업그레이드가 원활하게 이루어지지 않아 마이크로소프트사의 인터넷 익스플로러가 대부분의 웹 브라우저 시장을 점유하게 되었다. 네스케이프 6.x에서 플러그인 기술을 사용하다가 현재 네스케이프 7.x부터 모질라 파이어폭스(Mozilla Firefox)의 기술을 채용하여 갱신하였다.

3.4. 모질라 / 파이어폭스 플러그인(Mozilla/Firefox Plugin)

네스케이프 네비게이터 브라우저의 코드네임으로 시작한 모질라(Mozilla)는 네스케이프 네비게이터를 전담하여 개발하는 재단이 되어 업그레이드를 하게 된다. 이 때 개발한 브라우저는 모질라 브라우저로, 초기에는 모질라의 단순화된 버전인 파이어폭스(Firefox)와 같이 출시되었다가 현재는 파이어폭스라는 이름으로 출시되고 있다.

파이어폭스는 근본적으로 확장을 기본으로 해서 만들어진 브라우저로서 XPCOM이라는 기술을 사용하여 이전의 플러그인을 대체하고 있다.[7] 외부적 동작 면에서는 크게 다르지 않으나 운영체제에서 제공하는 사용자 인터페이스 기능 뿐만 아니라 운영체제 독립적인 XUL을 이용하여 사용자 인터페이스를 구현할 수도 있다는 점이다. 현재 네스케이프 네비게이터 7.x 부터는 바로 이 파이어폭스를 기반으로 개발되고 있어 플러그인의 호환이 가능하다. 따라서 파이어폭스용으로 만든 플러그인은 모질라와 네스케이프 네비게이터를 모두 지원하여 서비스를 제공할 수 있다.[8]

전반적인 외형적 특성은 이전과 동일하나 현재의 플러그인이라고 한다면 이전의 네스케이프 플러그인이 아닌 XPCOM 기반의 플러그인을 의미한다.

3.5. 사파리 플러그인(Safari Plugin)

사파리(Safari)는 맥 OSX 출현과 함께 등장한 브라우저로서 맥 OSX의 기본 브라우저이다. 기존의 맥에서는 인터넷 익스플로러가 지원이 됐으나 현재 맥에서는 파이어폭스나 사파리 브라우저가 주로 사용된다. 3장에서와 같이 맥은 플러그인 기술을 지속적으로 고려하였는데 이는 단순히 브라우저의 확장기능이 아닌 운영체제 자체에서 확장 기능을 위한 플러그인 체제를 지원하였다. 사파리는 이러한 플러그인 중에서 사파리에 적합한 플러그인을 사용하며 파이어폭스의 XPCOM이나 네스케이프의 LiveConnect와는 호환되지 않지만 동작이나 특성에 있어서는 큰 차이를 보이지 않는다.

Ⅳ. 클라이언트 PKI 툴킷

범용적인 운영체제와 브라우저에서는 기본적으로 인증서 사용을 위한 환경이 제공된다. 예를 들면 마이크로소프트사의 윈도우즈는 운영체제 내에서는 CSP (Cryptographic Service Provider)를 통해 인증서 사용을 위한 환경을 제공하고 있다. 또한 인터넷 익스플로러에는 개인정보를 위한 데이터 보소, 디지털 서명 데이터 등의 암호화 API를 사용할 수 있는 컴포넌트인 CAPICOM을 제공하여 인증서를 사용할 수 있게 해준다. 파이어폭스(Firefox)를 비롯한 네스케이프 계열에서는 운영체제에서 제공하는 CSP를 사용하지 않으나 PKCS#11을 지원하여 인증서를 사용할 수 있는 환경을 제공한다.

그러나 국내에서 사용하는 PKI는 운영체제나 브라우저에서 사용하는 표준과 호환성 문제나 상충하는 부분이 존재하여 브라우저에서 사용하는 PKI 관련 기능 대신 별도의 PKI 툴킷을 사용한다. 가장 큰 차이점으로 인증서 저장에 따른 부분을 들 수 있다. 현재 정보통신부와 한국정보보호진흥원의 집계에 따르면 2006년 말을 기점으로 공인인증서 사용자가 1,500만 명을 넘은 것으로 나타났다. 그러나 사용자는 편의성을 위해 HDD나 USB 메모리 등 안전하게 보관할 수 없는 곳에 인증서를 저장하여 사용하고 있다. 국외의 사례를 보면 발급받은 인증서는 개인키가 PC메모리를 비롯한 외부로 유출되지 않는 하드웨어적인 보안 장치인 HSM (Hardware Security Module)에 보관을 한다. HSM에 저장된 인증서를 사용하는 경우에는 CSP나 PKCS#11 의 인터페이스를 이용한다. 따라서 운영체제나 브라우저에서 손쉽게 인증서를 사용할 수 있게 된다.[10]

발급된 인증서는 CSP나 PKCS#11에 인증서를 옮겨 담아 사용할 수도 있다. 특히 네스케이프 계열의 PKCS#11에서는 인증서에 대한 처리가 엄격하며 RFC 2459나 RFC 3280에 따라 알 수 없는 인증서 정책

(Certificate Policy)을 가진 인증서에 대해서는 주입(Import)을 할 수 없다.[2,3,4] 국내 공인인증서에서는 인증서 정책을 인증서 용도를 구분하기 위해 사용하고 있어 네스케이프 계열에서는 국내의 공인인증서의 인증서 정책에 대한 정보를 알 수 없다고 분류하여 결국 사용자는 인증서를 사용할 수 없게 된다. 만약 이 필드의 속성을 처리 필수(Critical) 속성을 무시해도 좋은 속성(non-Critical)으로 바꾼다면 네스케이프 계열의 PKCS#11에서도 사용할 수 있을 것이다.

만약 국내의 공인증서를 CSP나 PKCS#11을 통해 사용자 인증서를 안전하게 저장되었다 하더라도 소유자 확인을 위한 식별번호 검증과 같은 서비스는 운영체제나 브라우저의 기본 확장기능으로 지원되지 않으므로 별도의 브라우저 확장 기능 없이 국내 PKI 시스템을 지원하기 힘들다. 또한 정부의 공인인증시스템(GPKI)의 경우 RSA 알고리즘이 아닌 국내 서명 알고리즘인 KCDSA를 기본적으로 사용하기 때문에 별도의 PKI 툴킷 없이는 지원할 수 없다.[4]

그 외에 파일서명이나 타 보안제품과의 연동 등 서비스 제공자의 요구사항으로 인한 추가된 기능들은 운영체제와 브라우저에서 기본적으로 지원해 주지 않기 때문에 반드시 별도의 확장 기능을 필요로 한다.

V. 대응 방안

한 가지 정책만을 고려하여 다양한 운영 체제를 지원하는 PKI 툴킷을 만들 수는 없을 것이다. 본 장에서는 용도나 목적에 따라 적절한 대응 방법을 검토한다.

일반적인 상용 목적의 PKI 툴킷을 고려한다면 개발의 어려움이 아닌 빠른 반응 속도, 사용자에게 친숙한 사용자 인터페이스, 쉬운 배포 방법 등이 매우 중요한 요소로서 작용을 한다. 따라서 이러한 경우라면 각 운영체제와 각 브라우저에서 가장 효율적으로 지원하는 방식이 필요하다. 즉 마이크로소프트사의 윈도우즈 환경에서는 인터넷 익스플로러의 액티브 엑스를, 파이어 폭스 및 네스케이프 등에서는 플러그인을 사용하면 사용자가 요구하는 PKI 툴킷을 구현할 수 있다. 물론 리눅스(Linux)에서는 파이어폭스(Firefox)의 플러그인으로 지원이 가능하며 맥 OSX에서는 사파리나 파이어폭스의 플러그인을 통해 사용자의 요구사항을 충족시킬 수 있다.

그러나 주요 환경이 아닌 다양한 운영 체제 환경에서 동작하는 PKI 툴킷을 만들고자 할 경우 운영체제와 브라우저에 거의 영향을 받지 않는 자바 애플릿을 고려할 수 있다. 이 경우에는 Ⅲ장 2절에서 볼 수 있듯이 상업적인 부분이나 보안적인 부분의 지원은 어려울 것이다.

물론 위의 두 가지 정책을 모두 사용하는 방법을 고려할 수도 있다. 즉 주요 운영체제와 브라우저에는 첫 번째 방법인 운영체제 따른 PKI 툴킷 지원을 사용하고 소수(Minority) 운영체제와 소수 브라우저에 대해서는 자바 애플릿과 같은 방법을 사용할 수도 있다. 이 방법은 상업적인 용도와 공공적인 업무 모두에 적합할 수 있어야 하며 다만 모든 경우를 지원하기 위해서는 그만큼의 비용이 더 소요된다는 점도 고려를 해야 한다.

장기적이고 근본적인 대책으로서는 운영제제와 브라우저에서 기본적으로 지원하는 방식으로 인증체제의 변경을 고려할 수 있다. 최근에 메모리 해킹 등으로 기존의 인증서 저장장치들이 안전하지 않다는 것이 증명되었으므로 마이크로소프트사의 CSP와 PKCS#11을 지원하는 보안토큰(HSM)을 도입하여 사용을 하는 것이다. 즉 별도의 PKI 툴킷을 사용하지 않고도 공공업무나 금융업무 등을 사용을 할 수 있을 것이다. 그러나 현재 보안토큰 도입도 미미하고 어려운 상황으로 추진하기에는 업계의 상당한 반대가 따를 것으로 여겨진다. 또한 서비스 제공자의 요구에 따라 추가된 기능들을 기본적으로 제공되는 기능들로 구현하기 위해서는 상당한 기술적 어려움이 따를 것으로 예상된다.

그 외의 대안으로 자바 애플릿이나 어도비의 플래쉬(Flash)처럼 다양한 환경에서 구동 가능한 방식을 직접 구현하는 방법이 있다. 물론 애플릿이나 플래쉬를 고려한다면 코드 디컴파일 부분을 고려하여 구현하는 것이 필요하며 모든 운영체제와 브라우저에서 지원할 수 있도록 가상머신(VM)을 제작 배포해야 한다. 따라서 상당히 기간과 비용이 소요될 것으로 추정되므로 국내의 소프트웨어 산업 규모에서는 추진하기가 매우 어렵다고 할 수 있다.

Ⅵ. 결 론

국내 SW산업이 미국, 유럽, 일본 등과 비교하여 상당히 열악한 상황이므로 사용자가 거의 없는 소수 운영체제와 브라우저를 모두 지원하기는 어렵다. 실제로 어도비의 플래쉬의 경우를 살펴보면 소수 운영체제와 브라우저까지 지원하는 것은 아니다.

뿐만 아니라 국내의 보안 요구 사항은 미국이나 일본보다도 월등히 많고 어렵기 때문에 국내의 보안 환경의 고려 없이 이러한 나라들과 직접 비교하는 것 자체가 무리가 있다고 판단이 된다. 제공되는 서비스 또한 다양하기 때문에 단순 열람위주로 구성되어 있는 국외 서비스들과 단순한 비교는 어렵다.

그러나 최대한 많은 사람들이 평등하게 인터넷을 사용하게 하기 위해 마이크로소프트사의 윈도우즈뿐만 아니라 최소한 범용적인 리눅스나 맥 운영체제 정도는 지원이 되어야 한다. 이는 경제적인 관점과 공공적인 관점을 모두 고려했을 때 도달할 수 있는 합의점이라고 판단할 수 있다. 이러한 합의점을 전제로 할 경우 우리는 앞에서 언급한 각 운영체제와 브라우저에 맞는 확장 기술을 사용하여 접근하는 방식을 고려해 볼 수 있을 것이다.

참고문헌

[1] http://openweb.or.kr/.

[2] "전자서명 인증서 프로파일 기술규격", 한국정보보호진흥원, 2007.

[3] "식별번호를 이용한 본인확인 기술규격", 한국정보보호진흥원, 2002.

[4] "전자서명 알고리즘 규격", 한국정보보호진흥흥원, 2007.

[5] 이기영, "Internet Explorer 7 소개", 한국마이크로소프트, 2006.

[6] 이동석, "Windows Vista IE7의 새로운 보안 하에서 ActiveX 컨트롤 개발", 한국마이크로소프트, 2006.

[7] Alee Flett, "Introduction to XPCOM", http://www.mozilla.org/projects/xpcom/xpcom-intro/xpcom-intro.htm.

[8] "Plugins", mozilla.org, http://www.mozilla.org/projects/plugins.

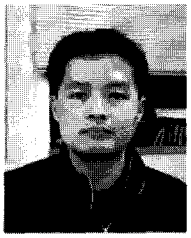
[9] "The LiveConnect/Plug-in Developer's Guide", Netscape, http://wp.netscape.com/eng/mozilla/3.0/handbook/plugins/.

[10] "PKCS #11 v2.20: Cryptographic Token Interface Standard", RSA Laboratories, 2004.

[11] "Adobe Flash", Wikipedia, http://en. wikipedia.org/wiki/Adobe_FlashCorporation, 2006

〈著者紹介〉

김 기 영 (Kim, Ki Young)
1997년 2월 : 한양대학교 전자공학과 졸업
1997년 3월 : 포스코그룹 입사
1998년 3월~2000년 9월 : 한국후지쯔 연구개발부 입사
2000년 10월~현재 : 소프트포럼 SW연구개발실 연구소장
<관심분야> 정보보호, 유비쿼터스



권 태 경 (Taekyoung Kwon)
1992년 2월 : 연세대학교 컴퓨터과학과 졸업
1995년 : 연세대학교 컴퓨터과학과 석사
1999년 : 연세대학교 컴퓨터과학과 박사
1999년 ~ 2000년 : U.C. Berkeley Post-Doc.
2001년 ~ 현재 : 세종대학교 컴퓨터공학부 컴퓨터소프트웨어과 부교수, 정보보호학회 편집위원, TTA 암호분과 특별위원
<관심분야> 정보보호, 암호프로토콜 등