웹 하드의 개인프라이버시 강화를 위한 환경구축

황선명, 염희균, *김시창
대전대학교 컴퓨터공학과
{sunhwang, yeom}@dju.ac.kr
sichang0160@naver.com

A server Construction for Personal Privacy Enforcement of Web Hard Disk

Sun-Myung Hwang
Hee-Gyun Yeom
Si-Chang Kim*
* Deajoen University

요 약

기존 시중의 웹 하드는 개인이 웹 하드 업체로부터 가상 서버를 위탁 받는다. 그 와중에서 사용자는 자신의 프라이버시를 웹 하드업체에 위탁하여 프라이버시의 신용도가 떨어진다. 그것을 개선하기 위하여 개인 프라이버시 강화를 위한 자체 서버 WEB과 WAS서버 구축과 자체 보안을 위한 SEED 관용키 사용, 마우스 클릭을 활용한 키보드 보안으로 구축된 자체 서버를 사용함으로서 개인프라이버시와 내부보안의 강화, 그리고 시중의 웹 하드와의 속도차이가 없음을 보여주는 것이 본 논문의 목표이다.

1. 서론

IT가 스마트해진 시대의 스마트한 사용자들의 웹 하드를 스마트하게 활용하는 사용자 인터페이스와 그런 사용자들의 스마트한 웹 하드 성능과 보안, 가장중요한 개인 프라이버시를 위한 사용자 최적의 인터페이스를 갖는 것이 웹 하드이다. 그런 사용자 인터페이스 UI를 설계하고 구현하여 성능에서는 WAS서버와 WEB서버의 적절한 서버분배를 사용하고, 보안성은 업체 보안기기 등으로 구성된 성능을 갖는 웹 하드 시스템이 상용화 되고 있는 추세다. 그런 웹 하드에는 Naver의 N드라이브와 LG의 U+웹 하드 등이 있는데 특히 먼저 나온 N드라이브는 웹 하드를 웹에서 지원하는 웹 하드 UI를 넘어서 윈도우 자체에 가상 하드디스크 기능까지 제공하는 실정이다. 또한 기능적 측면에서 ActiveX인 어도비에어를 설치하지 않고도 2기가라는 대용량을 전송할 수 있다. ActiveX 설치시는 4G라는 대용량을 전송할 수 있다. LG U+웹 하드 같은 경우에는 대다수가 유료 이며 N드라이브보다 더 큰 대용량50GB에서부터 1TB 데이터를 제공한다. 하지만 이런 편리성을 제공하는 기업에서 운영하는 웹 하드에는 개인 프라이버시 문제가 있다. 기업은 업체 자체 서버를 사용하여 사용자들의 100%의 신뢰를 얻지 못한 면이 있으며, 이런 개인 프라이버시 문제를 해결하는 것이 본 논문의 목표이다.

2. 관용키 테스트 및 웹 하드 테스트

개인이 사용자가 많은 시중의 웹 하드 들을 보면, 수많은 사용자 편의성을 제공한다. 하지만 사용자들의 입장을 고려했을 경우 자체 외장하드나 내장하드, USB 등을 갖고 있는 것 보다는 사적인 내용이 유출되지는 안을까 하는 걱정이 되는 측면이 있다. 이런 측면을 해결 하기위해서는 다양한 방법이 있을 수 있다. 그중 본 논문에서는 seed암호화 방식을 적용시킨 개인 서버 클라이언트 웹 하드 구축방안을 소개하려 한다.

2.1 seed키를 적용시켜 보안성 측면

seed 256 관용키를 사용하여 서버 클라이언트간의 데이터 전송에 암호화를 하여 기업 웹 하드와의 동등한 보안성을 갖는다. 다음은 그림은 seed 256 관용키의 전체적인 구조이다[1].

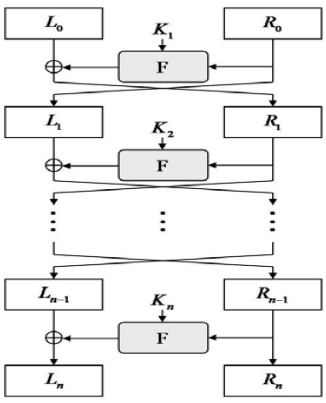

그림 1 SEED256 관용키 구조

C ′ G (G ((C⊕K i, 0)⊕(D ⊕K i, 1))⊞ C⊕K i, 0))

⊞G (G (G ((C⊕K i, 0)⊕(D ⊕K i, 1))⊞ C⊕K i, 0))⊞G ((C⊕K i, 0)⊕(D ⊕K i, 1)))

D ′ G (G (G ((C⊕K i, 0)⊕(D ⊕K i, 1))⊞ C⊕K i, 0))⊞G ((C⊕K i, 0)⊕(D ⊕K i, 1)))[2][3].

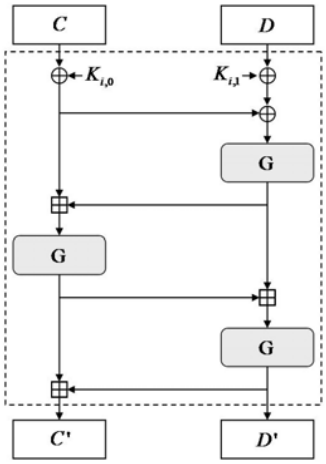

그림 2 SEED256 관용키 구조

2.2 Up Load를 위한 자바 패키지를 사용 테스트

가장 유명한 파일 업로드 패키지 Multipart Request와 MultipartParser를 사용방법 코드이다[6].

```
import="com.oreilly.servlet.MultipartRequest"
import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"
MultipartRequest multi = new MultipartRequest(request, savePath, sizeLimit, "UTF-8", new DefaultFileRenamePolicy());
// 일반 jsp는 euc-kr로 해야함
String user = multi.getParameter("user");
String fileName = multi.getFilesystemName("fileform");
String originFileName = multi.getOriginalFileName("fileform");
if(fileName==null){
        out.print("파일이 업로드 되지 않았습니다!!");
}
```

2.3 데이터 공개키 암호화 복호화 코드 테스트

```
// Round keys for encryption or decryption
int pdwRoundKey[] = new int[32];
// User secret key
byte pbUserKey[] = {(byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00,
(byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x00};
// input plaintext to be encrypted
byte pbData[] = {(byte)0x00, (byte)0x01, (byte)0x02, (byte)0x03, (byte)0x04, (byte)0x05, (byte)0x06, (byte)0x07,
(byte)0x08, (byte)0x09, (byte)0x0A, (byte)0x0B, (byte)0x0C, (byte)0x0D, (byte)0x0E, (byte)0x0F};
byte pbCipher[] = new byte[16];
byte pbPlain[] = new byte[16];

// Derive roundkeys from user secret key
SeedRoundKey(pdwRoundKey, pbUserKey);

// Encryption
SeedEncrypt(pbData, pdwRoundKey, pbCipher);
```

2.4 웹 하드 속도 및 SEED키 테스트 결과

테스트 내용은 속도 테스트와 안정성 테스트로 나눈다. 시중에 상업용으로 상용되는 N드라이브와 본 테스트프로그램과의 Up Load 속도 테스트이다.

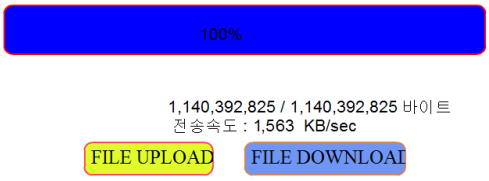

그림 3 테스트 프로그램 업로드 전송 속도

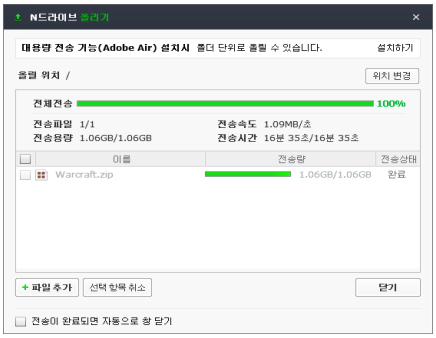

그림 4 N드라이브 업로드 전송 속도

위 그림 3과 그림 4에서 보는 것과 같이 전송속도에서는 시중의 웹 하드와 크게 성능 차이가 나지 않는다. 다음은 암호화된 데이터 파일의 데이터 흐름 표시 테스트이다.

```
[ Test Encrypt mode ]
Key        : 00000000000000000
Plaintext  : 01234567890abcdef
Ciphertext: 5eba c6e05 4e16 6819 aff1 cc6d 346c db

[ Test Decrypt mode ]
Key        : 00000000000000000
Ciphertext: 5eba c6e05 4e16 6819 aff1 cc6d 346c db
Plaintext  : 01234567890abcdef
```

그림 5 SEED 암호화 및 복호화 테스트

2.5 사용자 UI 접근 및 키보드 보안 테스트

사용자UI 접근은 기존 ID 패스워드 방식을 키보드가 아닌 web자체 혹은 스마트폰 앱 자체 환경에서 패스워드를 마우스를 통해 클릭하여 입력하는 방식이다. 본 방식은 한번 클릭 후에 자동 랜덤 수를 발생시켜 새로운 번호판을 발생시킴으로서 보안성을 강화한다. 더불어 키보드 보안과 사용자 신뢰도를 증폭시키는 효과가 있다.

3. 환경 구축

3.1 WEB, WAS 서버 연동 구축 방법

Web Server 구축환경에서는 Apache2.2버전(http://httpd.apache.org)을 사용하였으며, Was Server는 Tomcat7.0버전(http://tomcat.apsche.org)을 사용하였다. 연동 방법에는 톰캣에서 지원하는 Http Service인 mod_jk.so 파일을 다운받아 Apach2.2\modules에 저장한다. 다음은 Apache2.2\conf\httpd.conf 파일을 열어 다음과 같은 권한 설정을 추가한다. 아래와 같은 설정으로 Web Server 아파치에서는 HTML, JPG, GIF, PHP등과 Was Server 톰캣에서는 JSP, XML, SWF등의 권한이 설정 및 데이터 연동이 완료되었다.

```
##### PHP TEST #####
AddType application/x-httpd-php .php .html .htm .inc
LoadModule jk_module modules/mod_jk.so
JkWorkerProperty worker.list=ajp13w
JkWorkerProperty worker.ajp13w.type=ajp13
JkWorkerProperty worker.ajp13w.host=localhost
JkWorkerProperty worker.ajp13w.port=8009
JkLogFile logs/jk.log
JkLogLevel info
JkLogStampFormat "[%a %b %d %H: %M: %S %Y]"
#JkMount /*.jsp ajp13w
JkMount /* ajp13w
JkUnMount /*.html ajp13w
JkUnMount /*.gif ajp13w
JkUnMount /*.jpg ajp13w
JkUnMount /*.php ajp13w
#JkMount /servlet/* ajp13w
# PHP 5.x 버전, 아파치 2.2.x 버전 설치환경일 경우.
LoadModule php5_module "C:/php5/php5apache2_2.dll"
# php.ini 파일이 위치하는 경로입니다.
PHPIniDir "C:/php5"
```

3.2 개발 환경 구축

본 테스트 프로그램은 ActionScript3.0과 Java, JSP를 사용하였다. 이와 같은 다양한 언어를 사용하기 위해서는 어도비(http://www.adobe.com/)에서 지원하는 개발 툴인 FlashBuilder_4_5_LS10를 사용하여 eclipce에 플러그인을 하여 통합 개발 툴에서 개발하면 된다.

3.3 테스트 웹 하드 개발

ActionScript3.0으로 디자인 UI설계 및 개발한다. FileReference를 사용하여 Down Load 개발과 서블릿의 cos.jar(http://www.servlets.com/cos/)를 사용한 Up Load를 개발한다. Java와 Jsp, Seed키를 사용한 암호화 복호화 코드를 개발한다.

3.4 테스트 프로그램 UI

아래 그림6은 테스트 프로그램 전체 UI이다. 자신이 웹하드에 올린 파일을 다운로드 받을 수 있으며, 웹 하드에 보관할 파일을 업로드 할 수 있다.

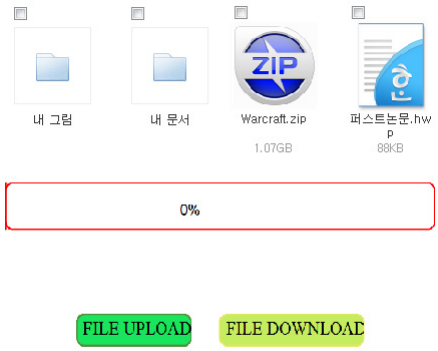

그림 6 테스트프로그램 전체 UI

4. 결론

보안 부분에서 본 테스트에서는 관용키만 사용하였지만, 관용키와 SSL등을 적용시킬 경우 시중에서 사용하는 웹하드와 외부적 보안이 동등하면 자체 서버를 사용하기 때문에 보다 내부적 보안은 위탁 서버를 사용하는 것 보다 좋아진다. 또한 테스트 결과 시중의 웹 하드 성능과 거의 유사한 것으로 나타났으며, 사용자 인터페이스의 UI측면에서 키보드를 사용하지 않고 마우스를 사용하는 방식으로 키보드 보안이 확실해졌다. 그로인해 웹 하드를 사용하는 사용자는 개인 프라이버시가 확실히 지켜진다.

참고문헌

[1] Korea Information Security Agency, *SEED Algorithm Specification*. Available at http://www.kisa.or.kr/kisa/ seed/down/SEED Specification english.pdf.

[2] Korea Information Security Agency, *SEED Algorithm Self Evaluation*. Available at http://www.kisa.or.kr/ kisa/seed/down/SEED Self Evaluation-English.pdf.

[3] D. Kwon, J. Kim, S. Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. Chee, D. Han and J. Hong, *New Block Cipher: ARIA*, ICISC'03, LNCS 2971, pp. 443 - 456, Springer-Verlag, 2003.

[4] NIST, FIPS 197: *Advanced Encryption Standard*, 2001.

[5] H. Yanami and T. Shimoyama, *Differential Cryptanalysis of a Reduced-Round SEED*, SCN'02, LNCS 2576, pp. 186 - 198, Springer-Verlag, 2003.

[6] http://servlets.com/cos/

[7] C. Couvreur, and J. Quisquater, "Fast Decipherment Al-gorithm for RSA Public-Key Cryptosystem," Electronics Letters, 18(21), pp. 905-907

[8] T. Narten, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6" RFC3041, Jan 2001