Java Script를 이용한 홈페이지 보안

제출일자 : 2012년 6월 5일
과 목 명 : 캡스톤 디자인
팀　　명 : 선 남 선 녀
팀　　장 : 곽순광
팀　　원 : 90816554 곽순광
90816530 강민우
91017040 고은선
91017301 장윤아

지도교수 : 양환석 교수님

요 약 문

1. 연구제목

JavaScript를 이용한 jsp 홈페이지 보안

2. 연구 목적 및 필요성

오늘날 우리는 facebook, 싸이월드, 네이버 등 홈페이지를 통해서 정보를 공유하고 대화를 나누며 생활한다. 이러한 홈페이지의 서비스를 이용하기 위해서는 회원가입이 필요한데 회원가입을 할 시 중요한 개인 정보를 입력하게 된다. 그래서 회원가입 시 우리가 작성하는 아이디와 패스워드가 안전하게 보안이 되어야 개인정보가 유출되지 않게 된다. 하지만 여전히 많은 곳에서는 홈페이지의 보안이 취약하여 개인 정보가 유출이 되는 곤란한 상황이 발생되고 있다. 비용, 키 관리의 어려움 등 많은 이유로 홈페이지의 보안을 중요시 하지 않고 있는데 이는 정보화 사회에서 개인정보를 침해하는 심각한 문제로 대두되고 있다.
이러한 관리의 어려움, 비용 등을 최소화하고 이식성이 뛰어난 자바를 이용하여 어떠한 운영체제 에서도 사용이 가능한 암호화 프로그램을 개발하여 손쉽게 아이디 패스워드를 암호화 하여 개인정보의 유출을 방지하기 위한 프로그램을 구현하는 것을 목표로 한다.

3. 연구 내용

본 연구에서는 개인정보 유출을 방지하기 위한 프로그램으로 클라이언트 내에서 암호화를 하여 서버로 보내게 된다. 기존의 홈페이지의 서버에서 암호화하는 방식과 달리 클라이언트 자체 내에서 암호화를 하는 방식으로 기존의 홈페이지 암호화방식 보다 더 강력히 개인정보의 유출을 막을 수 있다. 클라이언트 상에서 암호화를 하여 서버로 아이디와 패스워드를 보내기 때문에 중간에 와이어샤크(wireshark) 를 이용하여 스니핑(sniffing) 공격 시 아이디와 패스워드가 암호화 되어 알아볼 수 없게 제작하여 개인정보 유출을 방지할 수 있다.

4.연구 결과

처음 설계 계획은 jsp를 이용하여 클라이언트와 서버를 구축하고 클라이언트에서 서버로 아이디와 패스워드의 정보를 전송 시 클라이언트 내에서 자바 웹 스타트(Java WebStart)를 이용하여 서버로 보내기 전에 자바 웹 스타트를 이용하여 암호화를 한 후 그 정보를 서버로 전송하여 로그인이 되는 시스템을 구현하려 하였으나 자바 웹 스타트에서의 변수전달이 문제가 되어 클라이언트 내에서 암호화를 하여 보내는 방식으로 바뀌게 되었다. 현재는 클라이언트 내에서 자바 암호 라이브러리를 사용하여 RSA(Rivest Shamir Adleman)를 사용하여 암호화 하는 방식을 채택 하였다. 추후에 자바 웹 스타트를 이용하여 암호화 하는 프로그램을 계속 해서 진행할 예정이고 또한 다양한 암호화 라이브러리를 사용하여 ARIA, AES 등의 암호로 암호화 하는 방법을 연구해 볼 것이다.

2. 서 론

2.1 연구의 배경 및 목적

학교 홈페이지나 사설 홈페이지, 지방자치단체 홈페이지 같은 경우 보안에 취약한 모습을 보이는 곳이 한 두 곳이 아니라는 것을 본적이 있다. 일반 대학 경우 아이디와 패스워드를 학번과 주민등록 번호로 지정하는 곳이 여러 곳에서 발견 되었다. 그 경우 와이어 샤크를 이용하여 스니핑(sniffing) 공격을 해보면 학번과 주민등록번호의 뒷자리를 모두 볼 수 있게 되어있다. 악의를 가진 공격자가 이런 일반 대학교에 스니핑 공격을 시도할 경우 학생들의 개인정보를 손쉽게 갈취할 수 있게 된다. 또한 학생들은 교수들의 아이디와 패스워드를 동일한 방법으로 갈취하여 성적 조작이나 레포트 점수들을 임의로 조작하여 다른 학생들에게 많은 피해를 줄 수 있게 된다. 이러한 스니핑 공격을 사전에 방지하게 위해 전송되는 패킷의 내용을 암호화 하여 중간에 스니핑 공격을 받더라도 암호화가 된 패킷의 정보가 보여지게 되므로 효과적으로 스니핑 공격을 방어할 수 있다.

2.2 연구 목적

앞에서 말한 연구 동기에서처럼 학교 홈페이지 나 사설 홈페이지는 스니핑에 취약하므로 보다 안전하고 큰 비용이 들지 않는 측면에서 개발을 할 목적으로 만들게 되었다.
또한 기존에 홈페이지에서 보안을 하는 것 보다 쉽게 활용될 수 있는 프로그램을 목적으로 연구를 하였다.

2.3 개발 환경

Eclipse IDE for Java EE Developers
apache-tomcat-6.0
JSP
MySQL5.5
JavaScript

3. 이론적 배경

3.1 Java

웹 브라우저인 넷스케이프에서 사용할 수 있는 객체 지향 프로그래밍 언어로서 보안성이 뛰어나며 컴파일한 코드는 다른 운영 체제에서 사용할 수 있도록 클래스(class)로 제공된다. 객체 지향 언어인 C++ 언어의 객체 지향적인 장점을 살리면서 분산 환경을 지원하며 더욱 효율적이다.
<자바의 특징>

① 자바는 객체 지향 언어이다 (object-oriented)
요즘의 컴퓨터 언어는 객체 지향 언어로 개발된다. 자바도 객체 지향 언어이다.
부모 객체로부터 자식 객체는 상속을 한다. 자식 객체가 부모 객체로부터 상속을 받으면 부모 객체의 데이터와 메소드를 사용할 수 있음을 의미한다. 따라서 소프트웨어를 개발할 때 재활용 측면에서 많은 장점을 가진다.
② 자바는 보안에 강하다 (secure)
자바는 원래부터 네트워크 분산 처리 환경에서 사용하기 위해 디자인된 언어이다. 네트워크 환경은 다른 환경보다 보안의 측면이 강조되는 환경인 만큼 자바는 보안에 중점을 두고 있다. 자바는 바이러스가 침투하지 못하는 구조를 가지고 메모리에서 데이터 접근을 제한할 수 있다. 접근을 허용하지 않으면, 애플리케이션의 데이터 구조 또는 데이터에 대한 접근은 불가능하다.
③ 자바 아키텍처는 중립적이다 (architecture neutral)
네트워크는 다양한 기종의 컴퓨터와 다양한 플랫폼(예를 들면, 윈도 NT, 솔라리스, 매킨토시 OS 등의 운영 체제)과 다양한 하드웨어로 이루어져 있다. 자바는 자바 코드 소스를 컴파일 하여 바이트 코드를 만들어내며 이 바이트 코드는 다양한 플랫폼에 설치된 자바 인터프리터에 의해 해석되기 때문에 어떠한 플랫폼에서도 실행 가능하다. 따라서 새로운 기계라도 자바 인터프리터만 설치되어 있으면 바이트 코드를 해석할 수 있다.
④ 자바는 이식성이 높다 (portable)
기존의 언어는 각각의 플랫폼마다 수치 연산 문제 등으로 인하여 약간씩 다른 코드를 사용한다. 그러나 자바는 이식성이 강하여 다른 운영 체제, 다른 CPU에서도 같은 코드를 사용할 수 있다. 이식성이 높을 때의 단점은 각각의 시스템의 특성을 고려하지 않기 때문에 최적의 성능을 얻어낼 수 없는데, 자바는 이러한 것을 극복한 언어이다.

3.2 JSP (Java Server Page)

Java Server Page로서 ASP, PHP와 같은 서버 스크립트이다. 자바를 서버환경에서 사용하는 스크립트 방식의 언어로 단일 스레드로 클라이언트의 요청에 서비스한다.

요청이 있을 때 마다 프로세스를 생성하는 기존의 CGI와는 달리 하나의 메모리를 공유하면서 서비스되는 원리는 서버 측에 부하를 줄여주며, JSP 내부에는 보여주는 코드만 작성하고 작업하는 부분은 자바 빈으로 구성하여 분리 할 수 있다. 이것은 서로 영향을 주지 않고 수정할 수 있는 장점을 가지고 있으며, JAVA의 장점인 재사용성을 높일 수 있다.
서버에서 실행되기 때문에 HTML소스 에는 결과 값만을 보여주고 웹 브라우저 에서 소스보기 등을 해도 결과 값 외에 소스 코드를 확인할 수 없어 보안에 유용하다.

3.3 RSA (Rivest Shamir Adleman)

1977년 론 리베스트(Ron Rivest)와 아디 셰미르(Adi Shamir), 레오나르드 아델만(Leonard Adleman) 등 3명의 수학자에 의해 개발된 알고리즘을 사용하는 인터넷 암호화 및 인증시스템이다. 3명의 이름 가운데 첫 글자를 모아 붙인 용어이다. 마이크로소프트 윈도, 넷스케이프 브라우저를 비롯해 로터스 등 수백 개의 소프트웨어와 연동이 가능하며, 국제표준화기구(ISO)를 비롯하여, ITU·ANSI·IEEE 등 여러 국제기구에 암호표준으로 제안되

어 있다.
이 알고리즘은 두 개의 큰 소수(보통 140자리 이상의 수)를 이용한다. 이 수들의 곱과 추가 연산을 통해 하나는 공개키를 구성하고 다른 하나는 개인키를 구성하는데, 사용되는 두 세트의 수 체계를 유도하는 작업이 수반된다. 이렇게 구성된 공개키와 개인키로 인터넷에서 사용하는 정보(특히 전자우편)를 암호화하고 복호화 할 수 있다
개인키의 암호를 해독하려면 슈퍼컴퓨터로도 1만년 이상이 소요되므로 공개키 암호방식의 대명사로서 거의 모든 분야에 응용되고 있다. 그러나 계산량이 많은 것이 단점으로 꼽힌다. 비트 수에 따라 다르나 펜티엄급 컴퓨터에서 공개키와 개인키를 만들려면 짧게는 20여 초, 길게는 몇 분까지 기다려야 한다. 복호화에도 많은 계산량이 요구되고 있어 휴대용 단말기에서는 사용하기 어렵다.

3.4 JWS (Java Web Start)

자바 웹 스타트는 2001년 초에 소개된 기술로 한번 클릭으로 자바 애플리케이션을 쉽게 수행 시킬수 있는 환경으로 자바가 추가하던 사상 즉 "언제 어디서나 같은 코드로서 쉽게 접근하고 강력한 기능을 발휘하는 언어" 라는 강점을 유감없이 발휘할수 있는 기술이다.
자바 웹 스타트는 기존의 웹 환경의 한계점을 뛰어넘어 네트웍을 통해 자유롭게 풍부하고 강력한 응용프로그램을 실행시킬수 있으며, 버젼 및 배포 관리를 자동으로 수행하는 JNLP(Java Network Lanunching Protocol)을 기반으로 하여 관리함으로써 기존의 자바 애플릿등의 보안상의 단점을 해결하는 새로운 개념의 자바 배포 기술이다.

3.5 JDBC (Java DataBase Connectivity)

자바 프로그램 내에서 데이터베이스 질의문, SQL을 실행하기 위한 자바 API(application programming interface)이다.
JDBC는 데이터베이스 및 애플리케이션 개발자들을 위한 표준 API를 제공하고 순수 자바 API만으로도 데이터베이스 응용업무를 만들게 해준다. JDBC를 사용하면, 어떠한 관계 데이터베이스(relational database)에서도 SQL문을 사용하기 쉽다.
응용업무를 자바로 작성한다면 보유한 기종이나 소프트웨어에 따라 다르게 작성하지 않아도 되기 때문에 자바와 JDBC의 결합은 하나의 프로그램이 어디에서나 동작할 수 있게 해준다. 자바는 사용하기에 견고하고 안전하고 쉽고 이해하기 쉬우며 데이터베이스 응용업무를 만드는데 있어서 최적의 언어라 할 수 있다. 필요한 것은 다양한 데이터베이스에 연결하는 방법일 것이다. JDBC는 이러한 것을 위한 도구이다.

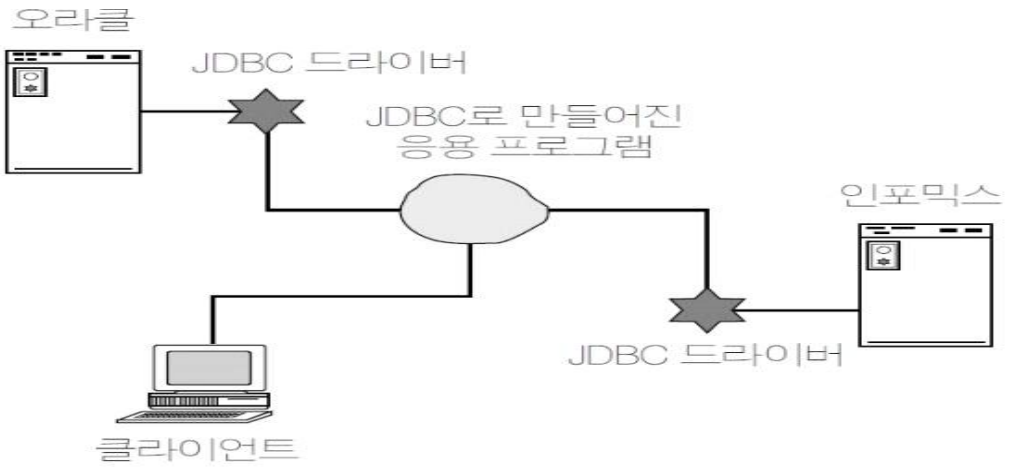


그림1 JDBC를 이용한 DB 접근

3.6 스니핑 (sniffing)

네트워크 상에서 자신이 아닌 다른 상대방들의 패킷 교환을 엿듣는 스니핑 공격은 웹호스팅, 인터넷 데이터센터(IDC) 등과 같이 여러 업체가 같은 네트워크를 공유하는 환경에서는 매우 위협적인 공격이 될 수 있다. 하나의 시스템이 공격당하게 되면 그 시스템을 이용하여 네트워크를 도청하게 되고, 다른 시스템의 사용자 ID와 비밀번호를 파악하는 것이 가능하다.

가장 많이 사용되는 해킹(hacking) 수법으로, 이더넷(Ethernet) 상에서 전달되는 모든 패킷(packet)을 분석하여 사용자의 계정과 암호를 알아내는 것이다.

3.7 MySQL

표준 데이터베이스 질의 언어인 SQL(Structured Query Language)을 사용하는 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템(RDBMS). 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 쓰레드를 지원하고, C, C++, Eiffel, 자바, 펄, PHP, Python 스크립트 등을 위한 응용 프로그램 인터페이스(API)를 제공한다. 유닉스나 리눅스, 윈도 운영 체제 등에서 사용할 수 있다.

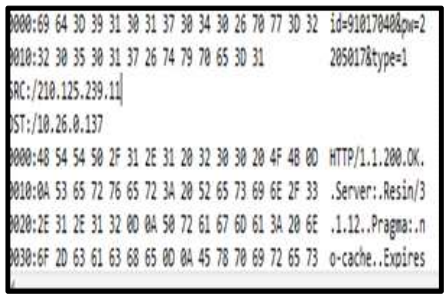
리눅스 운영 체제와 Apache 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 오픈 소스로 개발되는 무료 프로그램이어서 홈 페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다.

4. 과제목표 및 추진체계

4.1 구성원

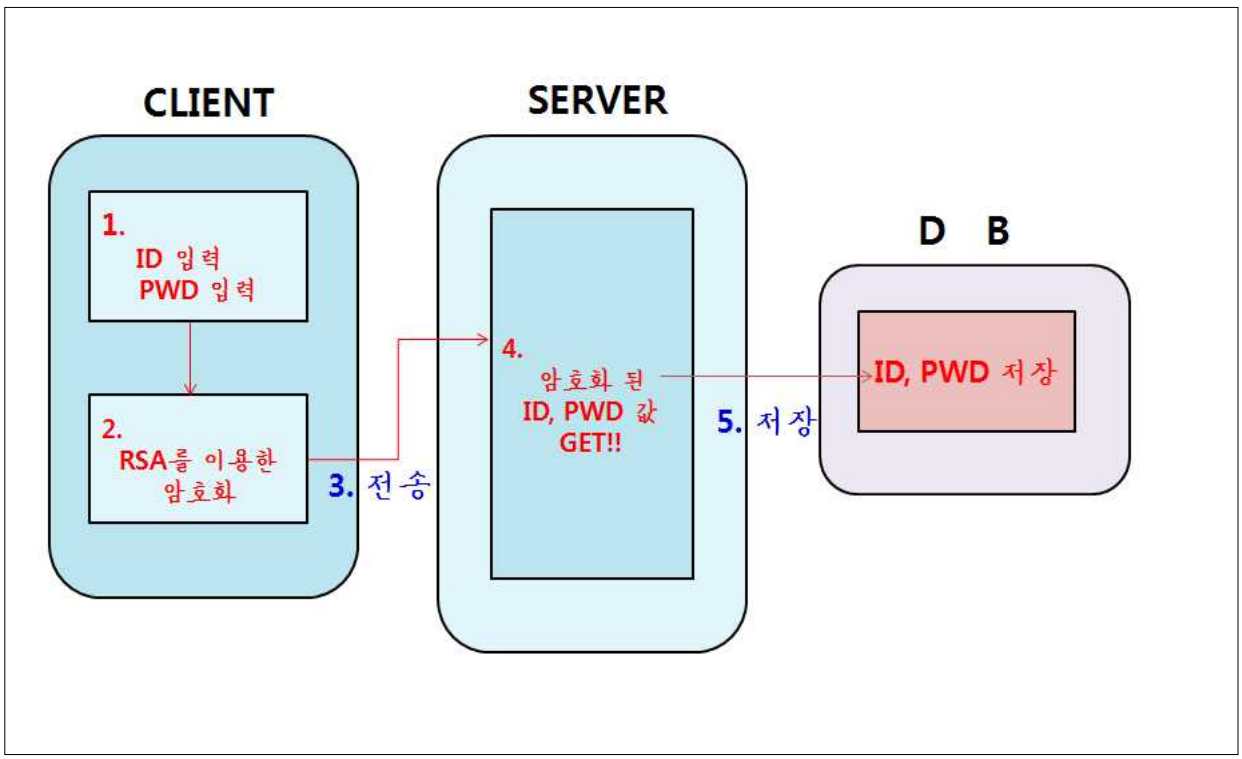
구성원	역할분담
곽순광	총괄책임 역할분담
강민우	프로그램제작 및 자료조사
고은선	암호 프로그램제작/자료조사
장윤아	자료조사 및 프로그램 제작

4.2 주간활동 보고서

<table>
<tr><td colspan="2">내용
주기</td><td colspan="2">주간활동사항</td></tr>
<tr><td colspan="2">1주차</td><td colspan="2">• 졸업 작품 주제 설정 및 회의</td></tr>
<tr><td colspan="2">2주차</td><td colspan="2">• 주제 확정 및 차후 계획 일정 수립</td></tr>
<tr><td colspan="2">3주차</td><td colspan="2">• JSP 홈페이지 토의 및 자료 조사 시작
• 암호화 방식에 대하여 토의</td></tr>
<tr><td rowspan="4">4주차</td><td>곽순광</td><td rowspan="2">• RSA암호화 공부</td><td rowspan="3">• RSA로 암호화 결정.
• Client와 Server 간단하게 구축</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td><td>• Server 구축 작업</td></tr>
<tr><td>고은선</td><td>• Client 구축 작업</td></tr>
<tr><td rowspan="4">5주차</td><td>곽순광</td><td colspan="2" rowspan="4">• 홈페이지 DB연동 시작
• Server와 Client 연동 시작</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
<tr><td rowspan="5">6주차</td><td>곽순광</td><td>• 자바 패킷캡쳐 프로그램 제작</td><td> 
</td></tr>
<tr><td>강민우</td><td colspan="2">• Server와 Client 연동</td></tr>
<tr><td>장윤아</td><td colspan="2" rowspan="2">• Java Web Start 공부 시작</td></tr>
<tr><td>고은선</td></tr>
<tr><td rowspan="4">7주차</td><td>곽순광</td><td colspan="2" rowspan="4">• 졸업작품 계획서 발표</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
</table>

<table>
<tr><td rowspan="4">8주차</td><td>곽순광</td><td rowspan="4">• Java Web Start를 활용하여 암호화 하는 프로그램 제작 돌입</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
<tr><td rowspan="4">9주차</td><td rowspan="2">곽순광</td><td rowspan="2">• Java Web Start를 활용하기 위해서 공부 시작</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>강민우</td><td>• Server 오류 점검</td></tr>
<tr><td>고은선</td><td>• Client 오류 점검</td></tr>
<tr><td rowspan="4">10주차</td><td>곽순광</td><td rowspan="4">중간점검</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
<tr><td rowspan="4">11주차</td><td>곽순광</td><td rowspan="4">• 중간정검 후 필요한 부분 수정 및 추가</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
<tr><td rowspan="4">12주차</td><td>곽순광</td><td rowspan="4">• 홈페이지 디자인 수정</td><td rowspan="4"> 
</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
<tr><td rowspan="4">13주차</td><td>곽순광</td><td rowspan="4">• 최종 수정 및 프로그램 오류점검 실시</td></tr>
<tr><td>강민우</td></tr>
<tr><td>장윤아</td></tr>
<tr><td>고은선</td></tr>
<tr><td colspan="2">14주차</td><td>• 발표 준비 및 선 테스트</td></tr>
<tr><td colspan="2">15주차</td><td>• · 제 작 발 표</td></tr>
</table>

4.3 프로그램 구성도



4.4 프로그램 설명

Client에서 ID와 Password를 입력할 시 Client내에서 암호화하여 서버로 전송한다.
암호화 된 ID와 Password는 서버를 거쳐 DB에 저장된다.
따라서 기존의 Server에서 암호화 하는 방식보다 스니핑에 안전하다.
대칭방식은 암호화할 때의 비밀키와 복호화 할때의 비밀키가 일치하므로 클라이언트에서 암호화 할때 비밀키의 노출이 불가피하기 때문에 비대칭 방식으로 클라이언트에서 공개키로 암호화를 하여 서버에서 비밀키로 복호화하는 안전한 암호화 방식인 RSA암호를 택 하였다.
키관리는 java.security.KeyPairGenerator를 통해 공개키와 개인키 생성, java.secrity.key와 java.security.KeyPair를 통해 암호화키를 캡슐화하고 공개키,개인키 캡슐화한다.

javax.crypto.spec.* 를 사용하여 키를 바이트배열의 형태로 키를 저장하고, ID와 패스워드를 입력받은 후 에 세션을 이용하여 공개키와 비밀키를 찾고 복호화 한다.

5. 결론

서버에서 암호화하여 저장하는 기존의 홈페이지 방식과는 달리 클라이언트 내에서 암호화한 후 서버로 전송하여 스니핑 등 해킹 기술에 보다 안전하게 대응할 수 있다.
또한 자바스크립트를 이용하여 클라이언트 내에서 암호화 하는 것이기 때문에 기존에 가장 곤란했던 비용 문제를 해결 할 수 있다.

5.1 향후 계획

좀 더 다양한 암호화 기술을 연구하고, 좀 더 보안이 강력하도록 웹 스타트를 사용하여 암호화가 가능하게 구현 할 것이다.

6. 부록

6.1 소스코드
로그인 소스 코드

```
<%@ page contentType="text/html; charset=euc-kr" %>
<%@ page
import="java.security.*,java.security.spec.*,sun.misc.*,javax.crypto.*"  %>
<%
KeyPairGenerator generator = KeyPairGenerator.getInstance("RSA");
//키쌍을 생성하기 위하여 초기화.
generator.initialize(1024);    //1024로 셋팅

KeyPair keyPair = generator.generateKeyPair();  //키 쌍을 생성함.
KeyFactory kf = KeyFactory.getInstance("RSA");

PublicKey publicKey = keyPair.getPublic();  //공개키
PrivateKey privateKey = keyPair.getPrivate();    //개인키

RSAPublicKeySpec publicSpec = (RSAPublicKeySpec)
kf.getKeySpec(publicKey, RSAPublicKeySpec.class);
//공개키의 모듈과 제곱지수를 얻기위하여 RSAPublicKeySpec으로 변경

session.setAttribute(publicSpec.getModulus().toString(16),privateKey);
//16진수 hex코드로 변경.
//공개키 값에 비밀키 클래스를 넣어줌.
%>
<html>
<title>로그인</title>
<script language="JavaScript" type="text/javascript" src="jsbn.js"></script>
<script language="JavaScript" type="text/javascript"
src="prng4.js"></script>
<script language="JavaScript" type="text/javascript" src="rng.js"></script>
<script language="JavaScript" type="text/javascript" src="rsa.js"></script>
<script language="JavaScript" type="text/javascript"
src="base64.js"></script>
<script>
```

```
function send1()
{
        if(document.main.id.value == '')
        {
                alert('ID를 입력해 주십시오.');
                return
        }
        if(document.main.password.value == '')
        {
                alert('PASSWORD를 입력해 주십시오.');
                return
        }
        do_encrypt();
}
```

```
function do_encrypt() {

  var rsa = new RSAKey();

rsa.setPublic(document.main.keyModulus.value,document.main.keyExponent.
value);
  //키와 공통주소
  var res = rsa.encrypt(document.main.password.value);
  //password 암호화
  if(res) {
     document.main.password.value = linebrk(hex2b64(res), 64);
//바이트 배열을 아스키 문자로 표현해서 전달하여야 하므로 암호문을 base64로
받음.
document.main.submit();
  }
}
</script>
<body>

<form method="post" action="login2.jsp" name="main">
<input type="hidden" name="keyModulus"
value="<%=publicSpec.getModulus().toString(16)%>"> <!-- 16진수 변환
필수 -->
<input type="hidden" name="keyExponent"
value="<%=publicSpec.getPublicExponent().toString(16)%>">
<table>
<tr><td>
ID
</td><td>
<input type="text" name="id" size="10" style="width:100height:22">
</td></tr>
<tr><td>
PASSWORD
</td><td>
<input type="password" name="password" size="10"
style="width:100height:22">
```

```
</td></tr>
<tr><td>
PASSWORD
</td><td>
<input type="password" name="password" size="10"
style="width:100height:22">

</td></tr>
<tr><td colspan="3" align="center">
<input type="button" value="확인" onClick="javascript:send1();">
</td><td>
</table>

</form>

</body>
</html>
```

로그인 처리 소스

```
<%@ page contentType="text/html; charset=euc-kr" %>
<%@ page
import="java.security.*,java.security.spec.*,javax.crypto.*,sun.misc.*"  %>
<%
String keyModulus = request.getParameter("keyModulus");
String id = request.getParameter("id");
String password = request.getParameter("password");

PrivateKey privateKey = (PrivateKey)session.getAttribute(keyModulus);
//keyModulus공캐키(hex)와 일치하는 개인키를 세션에서 받아온다.
Cipher cipher = Cipher.getInstance("RSA");
//RSA복호화를 위하여 인스턴스를 받아온다.
BASE64Decoder decoder = new BASE64Decoder()
byte[] encrypted = decoder.decodeBuffer(password)
//받아온 base64문을 바이트 배열로 다시 변경.
cipher.init(Cipher.DECRYPT_MODE, privateKey);
//개인키로 복호화 셋팅.
byte[] pw = cipher.doFinal(encrypted);  //복호화함.
%>
<html>
</head><title>암호화 로그인</title><head>
<body>
ID : <%=id%> <br>
PASSWORD : <%=new String(pw)%>
</body>
</html>
```

RSA 암호화 소스

```
function parseBigInt(str,r) {
  return new BigInteger(str,r);
}

function linebrk(s,n) {
  var ret = "";
  var i = 0;
  while(i + n < s.length) {
    ret += s.substring(i,i+n) + "₩n";
    i += n;
  }
  return ret + s.substring(i,s.length);
}

function byte2Hex(b) {
  if(b < 0x10)
    return "0" + b.toString(16);
  else
    return b.toString(16);
}
function pkcs1pad2(s,n) {
  if(n < s.length + 11) {
    alert("Message too long for RSA");
    return null;
  }
  var ba = new Array();
  var i = s.length - 1;
  while(i >= 0 && n > 0) ba[--n] = s.charCodeAt(i--);
  ba[--n] = 0;
  var rng = new SecureRandom();
  var x = new Array();
while(n > 2) {
    x[0] = 0;
    while(x[0] == 0) rng.nextBytes(x);
ba[--n] = x[0];
  }
  ba[--n] = 2;
  ba[--n] = 0;
```

```
return new BigInteger(ba);
}

function RSAKey() {
  this.n = null;
  this.e = 0;
this.d = null;
  this.p = null;
  this.q = null;
  this.dmp1 = null;
  this.dmq1 = null;
  this.coeff = null;
}
function RSASetPublic(N,E) {
  if(N != null && E != null && N.length > 0 && E.length > 0) {
    this.n = parseBigInt(N,16);
    this.e = parseInt(E,16);
  }
  else
    alert("Invalid RSA public key");
}
function RSADoPublic(x) {
  return x.modPowInt(this.e, this.n);
}

function RSAEncrypt(text) {
  var m = pkcs1pad2(text,(this.n.bitLength()+ 7)>>3);
  if(m == null) return null;
  var c = this.doPublic(m);
  if(c == null) return null;
  var h = c.toString(16);

 if((h.length & 1) == 0) return h; else return "0" + h;
}
RSAKey.prototype.doPublic = RSADoPublic;
RSAKey.prototype.setPublic = RSASetPublic;
RSAKey.prototype.encrypt = RSAEncrypt;
```

base64

```
var
b64map="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz012345
6789+/";
var b64pad="=";

function hex2b64(h) {
  var i;
  var c;
  var ret = "";
  for(i = 0; i+3 <= h.length; i+=3) {
    c = parseInt(h.substring(i,i+3),16);
    ret += b64map.charAt(c >> 6) + b64map.charAt(c & 63);
  }
  if(i+1 == h.length) {
    c = parseInt(h.substring(i,i+1),16);
    ret += b64map.charAt(c << 2);
  }
  else if(i+2 == h.length) {
    c = parseInt(h.substring(i,i+2),16);
    ret += b64map.charAt(c >> 2) + b64map.charAt((c & 3) << 4);
  }
  while((ret.length & 3) > 0) ret += b64pad;
  return ret;
}
function b64tohex(s) {
var ret = ""
  var i;
  var k = 0; // b64 state, 0-3
  var slop;
  for(i = 0; i < s.length; ++i) {
    if(s.charAt(i) == b64pad) break;
    v = b64map.indexOf(s.charAt(i));
    if(v < 0) continue;
    if(k == 0) {
      ret += int2char(v >> 2);
      slop = v & 3;
      k = 1;
}
```

```
else if(k == 1) {
      ret += int2char((slop << 2) | (v >> 4));
      slop = v & 0xf;
      k = 2;
    }
    else if(k == 2) {
      ret += int2char(slop);
      ret += int2char(v >> 2);
      slop = v & 3;
      k = 3;
    }
    else {
      ret += int2char((slop << 2) | (v >> 4));
      ret += int2char(v & 0xf);
      k = 0;
    }
  }
  if(k == 1)
    ret += int2char(slop << 2);
  return ret;
}

function b64toBA(s) {
  //piggyback on b64tohex for now, optimize later
  var h = b64tohex(s);
  var i;
  var a = new Array();
  for(i = 0; 2*i < h.length; ++i) {
    a[i] = parseInt(h.substring(2*i,2*i+2),16);
  }
  return a;
}
```

jsbn 소스

```
var dbits;

var canary = 0xdeadbeefcafe;
var j_lm = ((canary&0xffffff)==0xefcafe);

function BigInteger(a,b,c) {
  if(a != null)
    if("number" == typeof a) this.fromNumber(a,b,c);
    else if(b == null && "string" != typeof a) this.fromString(a,256);
    else this.fromString(a,b);
}

function nbi() { return new BigInteger(null); }
function am1(i,x,w,j,c,n) {
  while(--n >= 0) {
    var v = x*this[i++]+w[j]+c;
    c = Math.floor(v/0x4000000);
    w[j++] = v&0x3ffffff;
  }
  return c;
}
function am2(i,x,w,j,c,n) {
  var xl = x&0x7fff, xh = x>>15;
  while(--n >= 0) {
    var l = this[i]&0x7fff;
    var h = this[i++]>>15;
    var m = xh*l+h*xl;
    l = xl*l+((m&0x7fff)<<15)+w[j]+(c&0x3fffffff);
    c = (l>>>30)+(m>>>15)+xh*h+(c>>>30);
    w[j++] = l&0x3fffffff;
  }
  return c;

}
```

```
function am3(i,x,w,j,c,n) {
  var xl = x&0x3fff, xh = x>>14;
  while(--n >= 0) {
    var l = this[i]&0x3fff;
    var h = this[i++]>>14;
    var m = xh*l+h*xl;
    l = xl*l+((m&0x3fff)<<14)+w[j]+c;
    c = (l>>28)+(m>>14)+xh*h;
    w[j++] = l&0xffffff;
  }
  return c;
}
if(j_lm && (navigator.appName == "Microsoft Internet Explorer")) {
  BigInteger.prototype.am = am2;
  dbits = 30;
}
else if(j_lm && (navigator.appName != "Netscape")) {
  BigInteger.prototype.am = am1;
  dbits = 26;
}
else { // Mozilla/Netscape seems to prefer am3
  BigInteger.prototype.am = am3;
  dbits = 28;
}

BigInteger.prototype.DB = dbits;
BigInteger.prototype.DM = ((1<<dbits)-1);
BigInteger.prototype.DV = (1<<dbits);

var BI_FP = 52;

BigInteger.prototype.FV = Math.pow(2,BI_FP);
BigInteger.prototype.F1 = BI_FP-dbits;
BigInteger.prototype.F2 = 2*dbits-BI_FP;


var BI_RM = "0123456789abcdefghijklmnopqrstuvwxyz";
```

```
var BI_RC = new Array();
var rr,vv;
rr = "0".charCodeAt(0);
for(vv = 0; vv <= 9; ++vv) BI_RC[rr++] = vv;
rr = "a".charCodeAt(0);
for(vv = 10; vv < 36; ++vv) BI_RC[rr++] = vv;
rr = "A".charCodeAt(0);
for(vv = 10; vv < 36; ++vv) BI_RC[rr++] = vv;

function int2char(n) { return BI_RM.charAt(n); }
function intAt(s,i) {
  var c = BI_RC[s.charCodeAt(i)];
  return (c==null)?-1:c;
}



function bnpCopyTo(r) {
  for(var i = this.t-1; i >= 0; --i) r[i] = this[i];
  r.t = this.t;
  r.s = this.s;
}

function bnpFromInt(x) {
  this.t = 1;
  this.s = (x<0)?-1:0;
  if(x > 0) this[0] = x;
  else if(x < -1) this[0] = x+DV;
  else this.t = 0;
}
function nbv(i) { var r = nbi(); r.fromInt(i); return r; }
function bnpFromString(s,b) {
  var k;
  if(b == 16) k = 4;
  else if(b == 8) k = 3;
  else if(b == 256) k = 8; // byte array
  else if(b == 2) k = 1;
  else if(b == 32) k = 5;
  else if(b == 4) k = 2;
  else { this.fromRadix(s,b); return; }
  this.t = 0;
```

```
 this.s = 0;
  var i = s.length, mi = false, sh = 0;
  while(--i >= 0) {
    var x = (k==8)?s[i]&0xff:intAt(s,i);
    if(x < 0) {
      if(s.charAt(i) == "-") mi = true;
      continue;
    }
    mi = false;
    if(sh == 0)
      this[this.t++] = x;
    else if(sh+k > this.DB) {
      this[this.t-1] |= (x&((1<<(this.DB-sh))-1))<<sh;
      this[this.t++] = (x>>(this.DB-sh));
    }
    else
      this[this.t-1] |= x<<sh;
    sh += k;
    if(sh >= this.DB) sh -= this.DB;
  }
  if(k == 8 && (s[0]&0x80) != 0) {
    this.s = -1;
    if(sh > 0) this[this.t-1] |= ((1<<(this.DB-sh))-1)<<sh;
  }
  this.clamp();
  if(mi) BigInteger.ZERO.subTo(this,this);
}

function bnpClamp() {
  var c = this.s&this.DM;

while(this.t > 0 && this[this.t-1] == c) --this.t;
}

function bnToString(b) {
  if(this.s < 0) return "-"+this.negate().toString(b);
  var k;
  if(b == 16) k = 4;
  else if(b == 8) k = 3;
  else if(b == 2) k = 1;
```

```
else if(b == 32) k = 5;
  else if(b == 4) k = 2;
  else return this.toRadix(b);
  var km = (1<<k)-1, d, m = false, r = "", i = this.t;
  var p = this.DB-(i*this.DB)%k;
  if(i-- > 0) {
    if(p < this.DB && (d = this[i]>>p) > 0) { m = true; r = int2char(d); }
    while(i >= 0) {
      if(p < k) {
        d = (this[i]&((1<<p)-1))<<(k-p);
        d |= this[--i]>>(p+=this.DB-k);
      }
      else {
        d = (this[i]>>(p-=k))&km;
        if(p <= 0) { p += this.DB; --i; }
      }
      if(d > 0) m = true;
      if(m) r += int2char(d);
    }
  }
  return m?r:"0";
}
function bnNegate() { var r = nbi(); BigInteger.ZERO.subTo(this,r); return r; }


function bnAbs() { return (this.s<0)?this.negate():this; }


function bnCompareTo(a) {
  var r = this.s-a.s;
  if(r != 0) return r;
  var i = this.t;
  r = i-a.t;
  if(r != 0) return r;
  while(--i >= 0) if((r=this[i]-a[i]) != 0) return r;
  return 0;
}


function nbits(x) {
```

```
 var r = 1, t;
  if((t=x>>>16) != 0) { x = t; r += 16; }
  if((t=x>>8) != 0) { x = t; r += 8; }
  if((t=x>>4) != 0) { x = t; r += 4; }
  if((t=x>>2) != 0) { x = t; r += 2; }
  if((t=x>>1) != 0) { x = t; r += 1; }
  return r;
}


function bnBitLength() {
  if(this.t <= 0) return 0;
  return this.DB*(this.t-1)+nbits(this[this.t-1]^(this.s&this.DM));
}

function bnpDLShiftTo(n,r) {
  var i;
  for(i = this.t-1; i >= 0; --i) r[i+n] = this[i];
  for(i = n-1; i >= 0; --i) r[i] = 0;
  r.t = this.t+n;
  r.s = this.s;
}

function bnpDRShiftTo(n,r) {
  for(var i = n; i < this.t; ++i) r[i-n] = this[i];
  r.t = Math.max(this.t-n,0);
  r.s = this.s;
}


function bnpLShiftTo(n,r) {
  var bs = n%this.DB;
  var cbs = this.DB-bs;
  var bm = (1<<cbs)-1;
  var ds = Math.floor(n/this.DB), c = (this.s<<bs)&this.DM, i;
  for(i = this.t-1; i >= 0; --i) {
    r[i+ds+1] = (this[i]>>cbs)|c;
    c = (this[i]&bm)<<bs;
  }
```

```
for(i = ds-1; i >= 0; --i) r[i] = 0;
  r[ds] = c;
  r.t = this.t+ds+1;
  r.s = this.s;
  r.clamp();
}
function bnpRShiftTo(n,r) {
  r.s = this.s;
  var ds = Math.floor(n/this.DB);
  if(ds >= this.t) { r.t = 0; return; }
  var bs = n%this.DB;
  var cbs = this.DB-bs;
  var bm = (1<<bs)-1;
  r[0] = this[ds]>>bs;
  for(var i = ds+1; i < this.t; ++i) {
    r[i-ds-1] |= (this[i]&bm)<<cbs;
    r[i-ds] = this[i]>>bs;
  }
  if(bs > 0) r[this.t-ds-1] |= (this.s&bm)<<cbs;
  r.t = this.t-ds;
  r.clamp();
}
function bnpSubTo(a,r) {
  var i = 0, c = 0, m = Math.min(a.t,this.t);
  while(i < m) {
    c += this[i]-a[i];
    r[i++] = c&this.DM;
    c >>= this.DB;
  }
  if(a.t < this.t) {
    c -= a.s;
    while(i < this.t) {
      c += this[i];
      r[i++] = c&this.DM;
      c >>= this.DB;
    }
    c += this.s;
  }
```

```
else {
    c += this.s;
    while(i < a.t) {
      c -= a[i];
      r[i++] = c&this.DM;
      c >>= this.DB;
    }
    c -= a.s;
  }
  r.s = (c<0)?-1:0;
  if(c < -1) r[i++] = this.DV+c;
  else if(c > 0) r[i++] = c;
  r.t = i;
  r.clamp();
}
function bnpMultiplyTo(a,r) {
  var x = this.abs(), y = a.abs();
  var i = x.t;
  r.t = i+y.t;
  while(--i >= 0) r[i] = 0;
  for(i = 0; i < y.t; ++i) r[i+x.t] = x.am(0,y[i],r,i,0,x.t);
  r.s = 0;
  r.clamp();
  if(this.s != a.s) BigInteger.ZERO.subTo(r,r);
}
function bnpSquareTo(r) {
  var x = this.abs();
  var i = r.t = 2*x.t;
  while(--i >= 0) r[i] = 0;
  for(i = 0; i < x.t-1; ++i) {
    var c = x.am(i,x[i],r,2*i,0,1);
    if((r[i+x.t]+=x.am(i+1,2*x[i],r,2*i+1,c,x.t-i-1)) >= x.DV) {
      r[i+x.t] -= x.DV;
      r[i+x.t+1] = 1;
    }
  }
  if(r.t > 0) r[r.t-1] += x.am(i,x[i],r,2*i,0,1);
  r.s = 0;
  r.clamp();
}
```

```
function bnpDivRemTo(m,q,r) {
  var pm = m.abs();
  if(pm.t <= 0) return;
  var pt = this.abs();
  if(pt.t < pm.t) {
    if(q != null) q.fromInt(0);
    if(r != null) this.copyTo(r);
    return;
  }
  if(r == null) r = nbi();
  var y = nbi(), ts = this.s, ms = m.s;
  var nsh = this.DB-nbits(pm[pm.t-1]); // normalize modulus
  if(nsh > 0) { pm.lShiftTo(nsh,y); pt.lShiftTo(nsh,r); }
  else { pm.copyTo(y); pt.copyTo(r); }
  var ys = y.t;
  var y0 = y[ys-1];
  if(y0 == 0) return;
  var yt = y0*(1<<this.F1)+((ys>1)?y[ys-2]>>this.F2:0);
  var d1 = this.FV/yt, d2 = (1<<this.F1)/yt, e = 1<<this.F2;
  var i = r.t, j = i-ys, t = (q==null)?nbi():q;
  y.dlShiftTo(j,t);
  if(r.compareTo(t) >= 0) {
    r[r.t++] = 1;
    r.subTo(t,r);
  }
  BigInteger.ONE.dlShiftTo(ys,t);
  t.subTo(y,y);  // "negative" y so we can replace sub with am later
  while(y.t < ys) y[y.t++] = 0;
  while(--j >= 0) {
var qd = (r[--i]==y0)?this.DM:Math.floor(r[i]*d1+(r[i-1]+e)*d2);
    if((r[i]+=y.am(0,qd,r,j,0,ys)) < qd) {// Try it out
      y.dlShiftTo(j,t);
 r.subTo(t,r);
      while(r[i] < --qd) r.subTo(t,r);
    }
  }
```

```
if(q != null) {
     r.drShiftTo(ys,q);
     if(ts != ms) BigInteger.ZERO.subTo(q,q);
   }
   r.t = ys;
   r.clamp();
   if(nsh > 0) r.rShiftTo(nsh,r);  // Denormalize remainder
   if(ts < 0) BigInteger.ZERO.subTo(r,r);
}

function bnMod(a) {
   var r = nbi();
   this.abs().divRemTo(a,null,r);
   if(this.s < 0 && r.compareTo(BigInteger.ZERO) > 0) a.subTo(r,r);
   return r;
}



function Classic(m) { this.m = m; }
function cConvert(x) {
   if(x.s < 0 || x.compareTo(this.m) >= 0) return x.mod(this.m);
   else return x;
}
function cRevert(x) { return x; }
function cReduce(x) { x.divRemTo(this.m,null,x); }
function cMulTo(x,y,r) { x.multiplyTo(y,r); this.reduce(r); }
function cSqrTo(x,r) { x.squareTo(r); this.reduce(r); }

Classic.prototype.convert = cConvert;
Classic.prototype.revert = cRevert;
Classic.prototype.reduce = cReduce;
Classic.prototype.mulTo = cMulTo;
Classic.prototype.sqrTo = cSqrTo;

function bnpInvDigit() {
   if(this.t < 1) return 0;
   var x = this[0];
   if((x&1) == 0) return 0;
```

```
 var y = x&3;
  y = (y*(2-(x&0xf)*y))&0xf;    // y == 1/x mod 2^4
  y = (y*(2-(x&0xff)*y))&0xff;  // y == 1/x mod 2^8
  y = (y*(2-(((x&0xffff)*y)&0xffff)))&0xffff;

  y = (y*(2-x*y%this.DV))%this.DV;

  return (y>0)?this.DV-y:-y;
}

function Montgomery(m) {
  this.m = m;
  this.mp = m.invDigit();
  this.mpl = this.mp&0x7fff;
  this.mph = this.mp>>15;
  this.um = (1<<(m.DB-15))-1;
  this.mt2 = 2*m.t;
}
function montConvert(x) {
  var r = nbi();
  x.abs().dlShiftTo(this.m.t,r);
  r.divRemTo(this.m,null,r);
  if(x.s < 0 && r.compareTo(BigInteger.ZERO) > 0) this.m.subTo(r,r);
  return r;
}

function montRevert(x) {
  var r = nbi();
  x.copyTo(r);
  this.reduce(r);
  return r;
}

function montReduce(x) {
  while(x.t <= this.mt2) // pad x so am has enough room later
    x[x.t++] = 0;
  for(var i = 0; i < this.m.t; ++i) {
```

```
var j = x[i]&0x7fff;
    var                                    u0                                    =
(j*this.mpl+(((j*this.mph+(x[i]>>15)*this.mpl)&this.um)<<15))&x.DM;

    j = i+this.m.t;
    x[j] += this.m.am(0,u0,x,i,0,this.m.t);

    while(x[j] >= x.DV) { x[j] -= x.DV; x[++j]++; }
  }
  x.clamp();
  x.drShiftTo(this.m.t,x);
  if(x.compareTo(this.m) >= 0) x.subTo(this.m,x);
}

function montSqrTo(x,r) { x.squareTo(r); this.reduce(r); }
function montMulTo(x,y,r) { x.multiplyTo(y,r); this.reduce(r); }

Montgomery.prototype.convert = montConvert;
Montgomery.prototype.revert = montRevert;
Montgomery.prototype.reduce = montReduce;
Montgomery.prototype.mulTo = montMulTo;
Montgomery.prototype.sqrTo = montSqrTo;

function bnpIsEven() { return ((this.t>0)?(this[0]&1):this.s) == 0; }
function bnpExp(e,z) {
  if(e > 0xffffffff || e < 1) return BigInteger.ONE;
  var r = nbi(), r2 = nbi(), g = z.convert(this), i = nbits(e)-1;
  g.copyTo(r);
  while(--i >= 0) {
    z.sqrTo(r,r2);
    if((e&(1<<i)) > 0) z.mulTo(r2,g,r);
    else { var t = r; r = r2; r2 = t; }
  }
  return z.revert(r);
}
function bnModPowInt(e,m) {
  var z;
  if(e < 256 || m.isEven()) z = new Classic(m); else z = new Montgomery(m);
return this.exp(e,z);
}
```

```
BigInteger.prototype.copyTo = bnpCopyTo;
BigInteger.prototype.fromInt = bnpFromInt;
BigInteger.prototype.fromString = bnpFromString;
BigInteger.prototype.clamp = bnpClamp;
BigInteger.prototype.dlShiftTo = bnpDLShiftTo;
BigInteger.prototype.drShiftTo = bnpDRShiftTo;
BigInteger.prototype.lShiftTo = bnpLShiftTo;
BigInteger.prototype.rShiftTo = bnpRShiftTo;
BigInteger.prototype.subTo = bnpSubTo;
BigInteger.prototype.multiplyTo = bnpMultiplyTo;
BigInteger.prototype.squareTo = bnpSquareTo;
BigInteger.prototype.divRemTo = bnpDivRemTo;
BigInteger.prototype.invDigit = bnpInvDigit;
BigInteger.prototype.isEven = bnpIsEven;
BigInteger.prototype.exp = bnpExp;

BigInteger.prototype.toString = bnToString;
BigInteger.prototype.negate = bnNegate;
BigInteger.prototype.abs = bnAbs;
BigInteger.prototype.compareTo = bnCompareTo;
BigInteger.prototype.bitLength = bnBitLength;
BigInteger.prototype.mod = bnMod;
BigInteger.prototype.modPowInt = bnModPowInt;

BigInteger.ZERO = nbv(0);
BigInteger.ONE = nbv(1);
```

prng4. 소스

```
function Arcfour() {
  this.i = 0;
  this.j = 0;
  this.S = new Array();
}

function ARC4init(key) {
  var i, j, t;
  for(i = 0; i < 256; ++i)
    this.S[i] = i;
  j = 0;
  for(i = 0; i < 256; ++i) {
    j = (j + this.S[i] + key[i % key.length]) & 255;
    t = this.S[i];
    this.S[i] = this.S[j];
    this.S[j] = t;
  }
  this.i = 0;
  this.j = 0;
}

function ARC4next() {
  var t;
  this.i = (this.i + 1) & 255;
  this.j = (this.j + this.S[this.i]) & 255;
  t = this.S[this.i];
  this.S[this.i] = this.S[this.j];
  this.S[this.j] = t;
  return this.S[(t + this.S[this.i]) & 255];
}
Arcfour.prototype.init = ARC4init;
Arcfour.prototype.next = ARC4next;

function prng_newstate() {
  return new Arcfour();
}

var rng_psize = 256;
```

rng 소스

```
var rng_state;
var rng_pool;
var rng_pptr;

function rng_seed_int(x) {
  rng_pool[rng_pptr++] ^= x & 255;
  rng_pool[rng_pptr++] ^= (x >> 8) & 255;
  rng_pool[rng_pptr++] ^= (x >> 16) & 255;
  rng_pool[rng_pptr++] ^= (x >> 24) & 255;
  if(rng_pptr >= rng_psize) rng_pptr -= rng_psize;
}

function rng_seed_time() {
  rng_seed_int(new Date().getTime());
}
if(rng_pool == null) {
  rng_pool = new Array();
  rng_pptr = 0;
  var t;
  if(navigator.appName == "Netscape" && navigator.appVersion < "5" &&
window.crypto) {

var z = window.crypto.random(32);
    for(t = 0; t < z.length; ++t)
      rng_pool[rng_pptr++] = z.charCodeAt(t) & 255;
  }
  while(rng_pptr < rng_psize) {  // extract some randomness from Math.random()
    t = Math.floor(65536 * Math.random());
    rng_pool[rng_pptr++] = t >>> 8;
    rng_pool[rng_pptr++] = t & 255;
  }
  rng_pptr = 0;
rng_seed_time();
}
```

```
function rng_get_byte() {
  if(rng_state == null) {
    rng_seed_time();
    rng_state = prng_newstate();
    rng_state.init(rng_pool);
    for(rng_pptr = 0; rng_pptr < rng_pool.length; ++rng_pptr)
      rng_pool[rng_pptr] = 0;
    rng_pptr = 0;

  }

  return rng_state.next();
}

function rng_get_bytes(ba) {
  var i;
  for(i = 0; i < ba.length; ++i) ba[i] = rng_get_byte();
}

function SecureRandom() {}

SecureRandom.prototype.nextBytes = rng_get_bytes;
```

6.2 완료 ppt


Java 를 이용한 jsp 홈페이지 보안
선남선녀
일시 : 5월 27일
장소 : c5-417
중부대학교 정보보호
발표자 : 장윤아


목 차
조원 소개 및 역할 분담
동기 및 주제
개발 환경
구상도
향후 계획
중부대학교 정보보호학과

동기, 주제 개발 환경 구상도 진행 경과

1 조원 소개 및 역할 분담

지도 교수님
이병천 교수님

08 곽순광 (조장)
-총괄 책임 및 역할 분담

10 장윤아
-프로그램 제작 및 자료조사

10 고은선
- 암호 프로그램 제작/자료조사

08 강민우
- 자료 조사 및 프로그램 제작

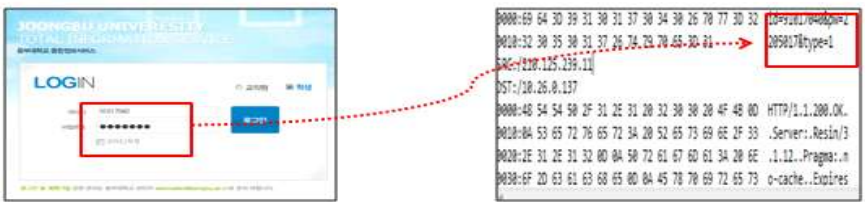
중부대학교 정보보호학과

조원 역할 분담 개발 환경 구상도 진행 경과

2 동기 및 주제

☞ 공공 기관 등의 홈페이지에서 id, pw 암호화하지 않고 전송 ➡ 보안 취약
- 지경부, 교과부 산하기관 중 39%가 홈페이지 비밀번호를 암호화 하지 않는 것으로 확인 (2013.3.28. 데일리 뉴스)

☞ 한국 앱손, 해킹으로 인해 35만명 고객정보 유출
- 회원들의 주민등록번호, 비밀번호를 암호화하지 않아 2차 피해 우려 (2011.8.20. MBN뉴스)



중부대학교 정보보호학과

조원 역할 분담 개발 환경 구 상 도 진행 경과

2 동기 및 주제

☞ 엑티브엑스(Active X)를 이용하여 패킷을 암호화 하여 스니핑(Sniffing)공격을 방어할 수 있지만 엑티브엑스 자체가 보안에 취약함을 보이고 있어, 다른 방법을 이용한 패킷 암호화를 하기로 하였다. 자바 웹 스타트(java web start)를 이용하여 암호화를 진행하던 중 변수 값을 전달하는 과정에서 기술적인 문제로 인하여 클라이언트 내에서 암호화 하여 서버로 전송하여 데이터베이스에 저장하는 방식을 사용 하였습니다.

조원 역할 분담 동기, 주제 구 상 도 진행 경과

3 개 발 환 경

Eclipse IDE for Java EE Developers

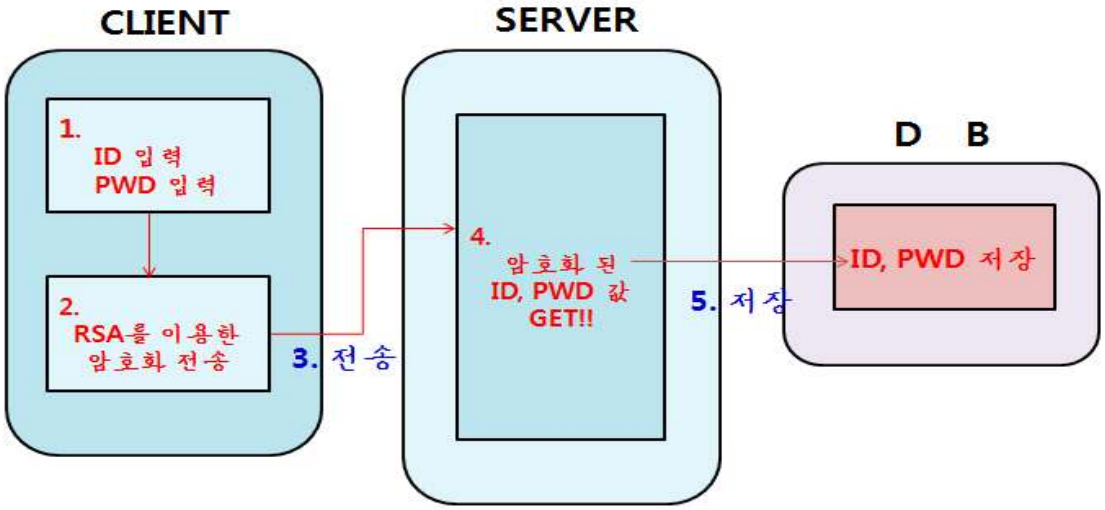
apache-tomcat-6.0

JSP

MySQL 5.5

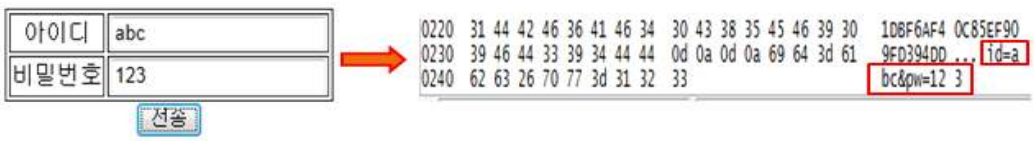
중부대학교 정보보호학과

4 구 상 도 (시스템 구성체계)

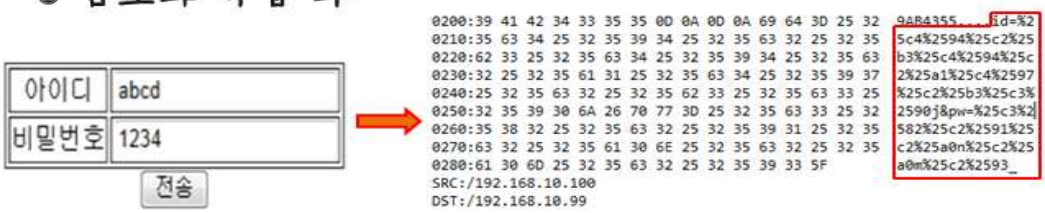


4 구 상 도 (보안 체계)

◎ 암호화 작업 전



◎ 암호화 작업 후



조원 역할 분담 동기, 주제 개발 환경 구 상 도

5 향후 계획

- 다양한 암호화 기술 연구
- 클라이언트내에 웹 스타트를 구동, 암호화

중부대학교 정보보호학과

중부대학교 정보보호학과

6.3 참고 문헌

[1] 폴. J.디텔 저
자바 프로그래밍 / 비팬북스
[2] 김은옥 저
JSP 2.0 웹 프로그래밍 /삼양미디어
[3] 최영관 저
소설같은 자바 4판 / 자 북
[4] 고경희 저
자바스크립트 무작정 따라하기
[5] 크리스 샌더즈 저
와이어샤크를 활용한 실전 패킷분석 / 에이콘