

CS 5523 Lecture 19: Secure Sockets and Certificates in Java

- Discuss laboratory 2
- Discuss laboratory 3
- Secure Socket Layer
- SSL Classes and Examples in Java

Secure socket layer (SSL) certificates contain:

- **Issuer** – if a user trusts the CA that issues a certificate, and if the certificate is valid, the user can trust the certificate.
- **Period of validity** - an expiration date that should be checked when verifying the validity of a certificate.
- **Subject** - includes information about the entity that the certificate represents.
- **Subject's public key** – the primary piece of information that the certificate provides. All other fields provide validity of this key.
- **Signature** - signed by the CA that issued the certificate to ensure the validity of the certificate. Because only the certificate is signed, not the data sent in the SSL transaction, SSL does not provide for non-repudiation.

Secure socket layer (SSL)

- Developed by Netscape, 1994
- Standardized by IETF January 1999 as TLS 1.0 (SSL 3.0)
- Is supported by most browsers for electronic transactions
- Has negotiable encryption and authentication algorithms
- Bootstraps by establishing a secure channel based on public-key encryption
- Channel is fully configurable so not everything has to be encrypted

Certificate authorities (CA):

- Browser typically stores information for several root CAs
- CA information includes the CA's public key.
- Well-known CAs are VeriSign, Entrust, and GTE CyberTrust.

Secure socket layer (SSL) encryption

- uses public key cryptography (RSA) to provide authentication
- uses secret key cryptography to provide privacy:
 - Data Encryption Standard (DES)
 - Triple-strength DES (3DES)
 - Rivest Cipher 2 (RC2)
 - Rivest Cipher 4 (RC4).
- uses digital signatures to provide data integrity.

Secure socket layer (SSL) protocol

- SSL Record Protocol layer implements a secure channel that encrypts and authenticates message through any connection-oriented protocol
- SSL Handshake Layer – has three modules:
 - SSL handshake protocol
 - SSL change cipher specification
 - SSL alert protocol
- Implemented as application level libraries
- Widely used as a session-layer protocol
- In web servers the https URLs initiate a SSL connection
- Provides a practical hybrid security scheme
- Requires public-key certificates issued by a recognized authority

Figure 7.17
SSL protocol stack

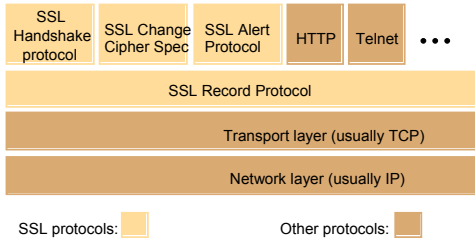


Figure 7.20
SSL record protocol

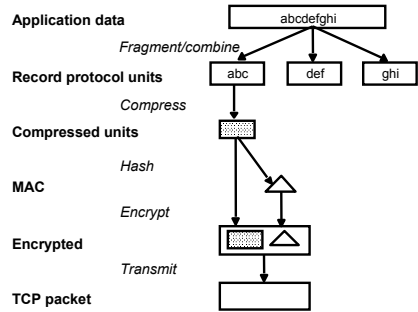
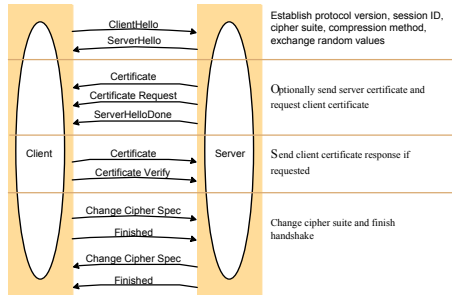


Figure 7.18
SSL handshake protocol



SSL in Java

- Part of Java 1.4
- The package `java.security.cert` contains the certificate classes
- The package `javax.net.ssl` contains the secure socket layer implementation
- Suitable for export from the US
- Well-known CAs include VeriSign, Entrust, and GTE CyberTrust.

Figure 7.19
SSL handshake configuration options

Component	Description	Example
Key exchange method	the method to be used for exchange of a session key	RSA with public-key certificates
Cipher for data transfer	the block or stream cipher to be used for data	IDEA
Message digest function	for creating message authentication codes (MACs)	SHA

Secure server using default

```
import java.io.*;
import javax.net.ssl.*;

...
int port = availablePortNumber;
SSLServerSocket s;
try {
    SSLServerSocketFactory sslSrvFact = (SSLServerSocketFactory)
        SSLServerSocketFactory.getDefault();
    s = (SSLServerSocket) sslSrvFact.createServerSocket(port);
    SSLSocket c = (SSLSocket)s.accept();
    OutputStream out = c.getOutputStream();
    InputStream in = c.getInputStream();
    // Send and receive messages
} catch (IOException e) { }
```

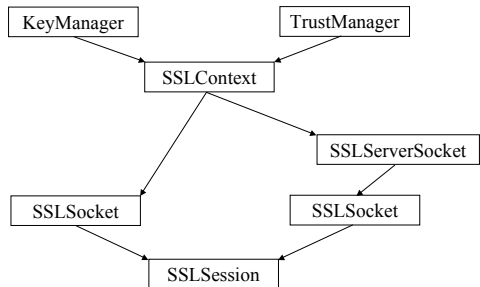
Secure client using default

```
import java.io.*;
import javax.net.ssl.*;
...
int port = availablePortNumber;
String host = "hostname";
try {
    SSLSocketFactory sslFact =
        (SSLSocketFactory)SSLSocketFactory.getDefault();
    SSLSocket s = (SSLSocket)sslFact.createSocket(host, port);
    OutputStream out = s.getOutputStream();
    InputStream in = s.getInputStream();
    // Send and receive messages
} catch (IOException e) { }
```

For next time:

- Finish reading Chapter 7
- Start reading Chapter 9 of Core Java Volume 2

Java security class relationships



SSLSession in Java

- Security context negotiated by the peers
- Contains the cipher suite used for communication
- Has management information such as creation time
- Contains a shared master secret for creating keys