



Entrust[®]
TECHNOLOGIES

We Bring Trust to e-Business™



We Bring Trust to e-Business™

Entrust/Toolkit Java™ Edition

March 2000





Entrust/Toolkit Java™ Edition Overview

- **Java vs CGI scripts**
- **Why use Java for security?**
- **Entrust/Toolkit Java™ Edition**
 - **Capabilities and Benefits**
 - **Architecture**
 - **Key Classes and Class Usage**
- **Web Application Server Security**
- **Web Application Server Limitations**
- **Entrust/PKI™ and Web Application Servers**



Java vs CGI Scripts

- **CGI scripts are known for being insecure**
- **ActiveX controls, ASP, VBScript, JavaScript possess known vulnerabilities**
- **Notorious for memory leaks and buffer overruns**
- **Hackers have already exploited these holes**
- **Web security breaches are related to memory vulnerabilities**



Why Use Java for Security?

- **Java uses a mechanism called “Type Safety”**
- **Bypasses memory vulnerabilities**
- **Compiler verifies code, ensures that no dangerous access to memory is performed**
- **Security manager verifies code before it is executed**
- **Hacker has no direct access to server-side Java code**
- **Hacker can only view client HTML source generated by the application server**



Entrust/Toolkit Java™ Edition Capabilities

- **Leverages Entrust/PKI for Java applications and applets**
- **Supports use of Entrust Profiles and Entrust PKIX-based protocols**
- **Full X.509 Version 3 certificate and chain validation**
- **Uses a high-level API to provide PKCS#7 enveloped messages**
- **SSLv3 connections, server side and mutual authentication**
- **Multi-ca support**
- **RMI support**
- **PKCS12 key repository access**



Entrust/Toolkit Java™ Edition Benefits

- **Entrust-Ready security provider for the Java Cryptography Extension (JCE)**
- **Provides many of the features found in Entrust's C / C++ toolkits**
- **Consistent and manageable interface**
- **Platform neutral Java-based security**
 - **Linux, Solaris, Win2000...**
- **Comprehensive cryptographic core**
- **Supports JDK 1.1 and 1.2 (Java 2)**
- **Client and server ID**



Entrust/Toolkit Java™ Edition Architecture

**Entrust
Profiles**

*X.509
Certificate
Validation*

PKCS#7

SSL

ASN.1

**JCE
RSA, DES, CAST...**





Entrust/Toolkit Java™ Edition Architecture

- **Entrust profile module supports**
 - **Creation and recovery**
 - **Key and certificate usage**
 - **Key and certificate management**
- **X.509 certificate validation module supports**
 - **Cross-certification**
 - **CRL checking**
 - **X.509 certificate extensions**
- **PKCS#7 module supports PKCS#7 for securing data**
 - **S/MIME uses PKCS#7**

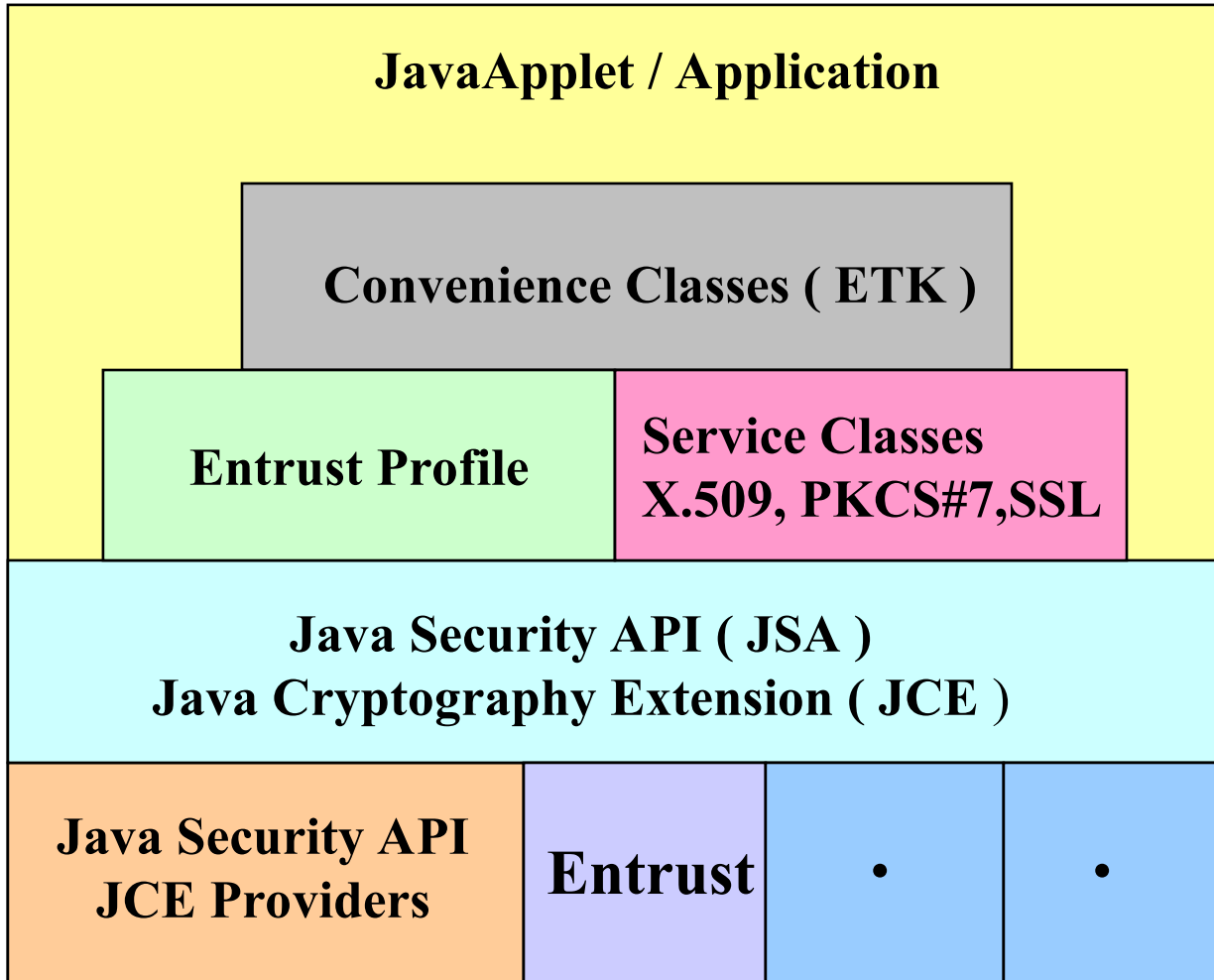


Entrust/Toolkit Java™ Edition Architecture

- **SSL module provides support for SSLv3**
 - **Server and mutual authentication**
- **ASN.1 module simplifies the process of encoding and decoding required structures, such as:**
 - **X.509 Certificates**
 - **X.509 Standard Extensions**
 - **Certificate Revocation Lists**
- **JCE module provides a modular interface to standard cryptographic algorithms**
 - **RSA, DSA, DH**
 - **SHA1, MD5, MD2**
 - **DES, Triple-DES, CAST, IDEA, RC2, RC4**



Entrust/Toolkit Java™ Edition Architecture





Entrust/Toolkit Java™ Edition Architecture

- **The Java Cryptography Architecture (JCA) provided by the JDK consists of:**
 - **Java Security API (JSA)**
 - **Java Cryptography Extension (JCE)**
- **The JCA allows you to specify third party cryptographic classes**
 - **Entrust classes are grouped as a security provider**
- **The Entrust/Toolkit contains a provider**
 - **The Entrust Provider acts as a bridge between the core cryptographic classes and elements of the Entrust/PKI architecture, such as Entrust Profiles**



Entrust/Toolkit Java™ Edition

Key Classes

- **EntrustProfile**
 - Provides a consistent interface for working with Entrust Profiles
- **ETKPKCS7**
 - Provides a consistent interface for performing PKCS#7 operations
- **SSLSocket, and SSLServerSocket**
 - Provide consistent interfaces for SSLv3 communications
- **ETKCertificateVerifier**
 - Provides a consistent interface for verifying certificates



Entrust/Toolkit Java™ Edition Class Usage

- **EntrustProfile Class**

- Instantiate an **EntrustProfile** object representing the entrust profile of your application's user
- Call the **EntrustProfile.Logon()** method so that the **EntrustProfile** object can be used perform security operations

// Log on with a profile

```
EntrustProfile profile = new EntrustProfile();  
profile.logon( profileStream, password );
```



Entrust/Toolkit Java™ Edition

Class Usage

- **ETPKCS7 class**
 - Construct **ETKPKCS7** object specifying the **EntrustProfile** object as the parameter in the **ETKPKCS7()** constructor
 - Call the **ETKPKCS7** object's methods for processing PKCS #7 data structures

// Prepare a PKCS#7 helper

```
ETKPKCS7 pkcs7 = new ETKPKCS7( profile );
```

// Sign your plain text and encrypt the signed copy with 128 bit CAST 5 CBC mode

```
Data data = pkcs7.encodeData( PlainText );
```

```
SignedData signedData = pkcs7.encodeSignedData( data,  
pkcs7.SIGNED_CONTENT );
```

```
EnvelopedData envelopedData = pkcs7.encodeEnvelopedData( signedData, Cert  
AlgorithmID.cast5_CBC, 128 );
```

```
ContentInfo contentInfo = pkcs7.encodeContentInfo( envelopedData );
```



Entrust/Toolkit Java™ Edition Class Usage

- **SSLSocket and SSLServerSocket class**
 - Construct a **SSLServerSocket** object
 - For a SSL **client**, construct a **SSLSocket** object
 - **SSLServerSocket** and **SSLSocket** classes extend the JDK's core **ServerSocket** and **Socket** classes

/ **Create a context**

```
SSLClientContext context = new SSLClientContext();
```

/ **Create an SSLv3 connection**

```
SSLSocket socket = new SSLSocket( server, port, clientContext );  
SSLout = socket.getOutputStream();
```





Entrust/Toolkit Java™ Edition Class Usage

- **ETKCertificateVerifier class**
 - To simplify obtaining and verifying certificates, the **ETKCertificateVerifier** class
 - For SSL connections in which one of the parties is not an Entrust user, the Toolkit provides the **TrustDecider** interface
 - Detailed information about the **ETKCertificateVerifier** and **TrustDecider** classes can be found in the Programmer's Guide

Design a TrustDecider

```
TrustDecider trustDecider = new ETKCertificateVerifier( directory, profile );  
context.setTrustDecider( trustDecider );
```





Web Application Server Security

- **LDAP for authentication**
- **X.509 certificates for encryption/signing**
- **Compatibility with firewalls**
- **Controlled access to pages, data sources**
- **EJB security focuses on access control**





Web Application Server Limitations

- Exposed to man-in-the-middle attack
- No private keystore mechanism
- No persistence for digital signature or encryption
 - End-to-end verification
 - Non-repudiation
 - Data privacy
- SSL crypto lowered to lowest common level
- No key management support
- No automatic key update, CRL...





Entrust/PKI and Web Application Servers

- **We secure transactions in a web-based architecture**
- **Applications written for PKI and web application servers offer flexible and reliable security**
- **We provide secure, efficient exchange of data across Enterprise and B2B environments**



Entrust/PKI and Web Application Servers

- **Private key storage**
- **Complete key management lifecycle**
- **End to end prevents man in the middle attacks**
- **Secure auditing for large value transactions**
- **Strong authentication to realms**
- **Non-repudiation for digitally signed transactions**
- **Standard compliance**

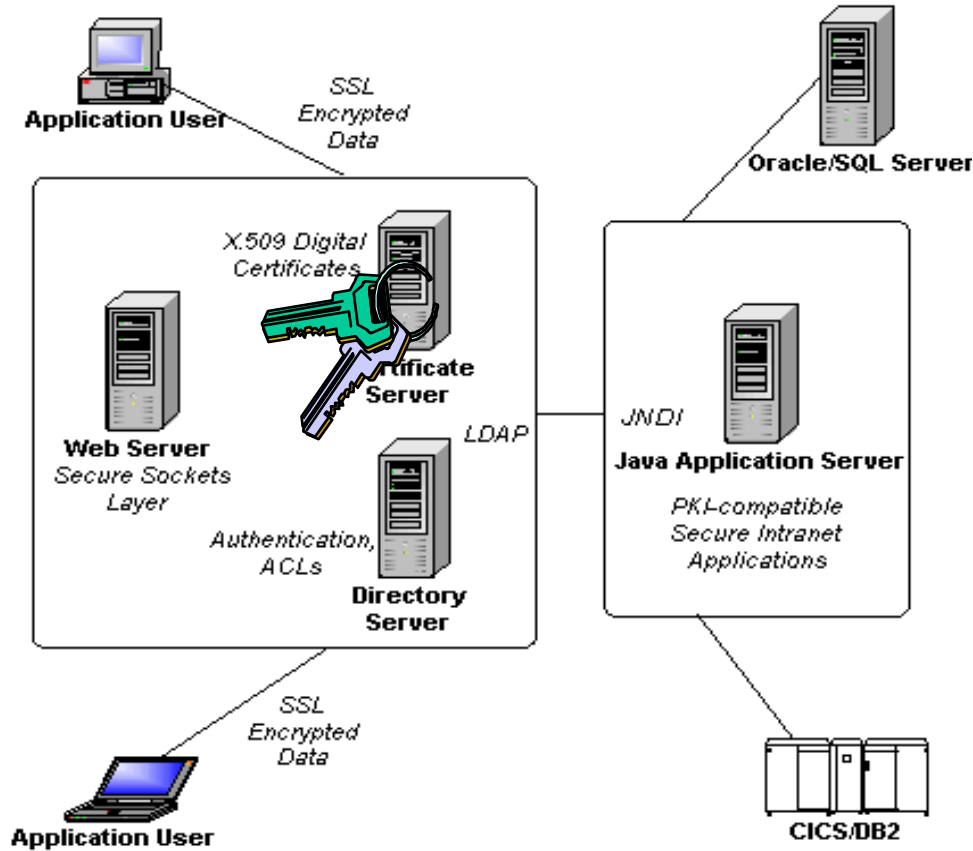




Entrust/PKI and Web Application Servers

**E
N
T
R
U
S
T

P
K
I**



Copyright 1999, App-Serv Corporation
All Rights Reserved



We Bring Trust to e-Business™

Thank You

**For more information, please visit
our web site at:**

<http://developer.entrust.com>