Pure Java Server Signature Modules

Modules for Creating and Verifying Digital Signatures

Peter Lipp
Karl Scheibelhofer
Contents

- „Citizen Card“ in Austria
- Server Modules for Digital Signatures
  - Architecture
  - Signature Creation/Verification
  - Configuration
  - Performance
  - Outlook
Concept „Citizen Card“

Any Device can be a Citizen Card

- Digital Signature according to Law
- Identify the Person
- Conforms to the Security-Layer Specification
Signature Requirements

Given by the Austrian Law for Electronic Signatures

- Algorithms (until End of 2005)
  - RSA ($\geq$1023), DSA (1024), ECDSA ($\geq$160)
  - RIPEMD-160, SHA-1

- Certified Signature Creation Device (Smart Card)
Identity of the Person

- Through a unique ID
- ID is bound to the Public Key with a Signature
- Used to Identify Persons in e-Government Processes
Security-Layer Specification

- XML based Request-Response Protocol
- TCP or HTTP as Transport (Port 3495)
- Specifies Commands for
  - Creation and Verification of XML Signatures
  - Creation and Verification of CMS Signatures
  - Access to Info-Boxes
  - Key Agreement (DH)
  - Query Properties
Security-Layer

Application

Security-Layer Specification

Citizen Card Environment

Card API

Trusted Viewer

CA Components

CA Components

Trusted Viewer

Card API

Citizen Card Environment

Security-Layer Specification

Application

IAIK
The Server Side

- Signature Cards enable Users to sign Documents, Identify themselves, ...
- The Systems of Official Bodies and Companies also need to process Signatures
- Signature Creation and Verification is often similar in different workflows
MOA SP – checking Signatures

Citizen uses:
- Web-Browser
- His citizen card + local security layer or equivalent

Application
[XML/HTML]
John Doe
[signieren]

Online Application

MOA SP
Signature Verification
Certificate Verification
Manifest Verification

Webservice/API
MOA SS – signing

Citizen uses:

- Web-Browser
- His citizen card + local security layer or equivalent
Server Modules

Java Library for Server Application
- XML Signature Creation/Verification
- CMS Signature Verification
- Designed to fit the Security-Layer Specification (but **not** restricted to it)
- Configurable Certificate Chain Building and Verification (PKIX and Chain Model)
- Support for Crypto Hardware (HSMs) via PKCS#11
Additional Requirements

- Stability and Robustness
- Run 24 by 7
  - Reconfiguration at Runtime
    - Change Keys, Modify Trust Settings
- At least 10 Signatures per Second Throughput
- Based on International Standards
  - XML Signature, CMS, PKCS, PKIX
Basic Architecture
Signature Creation

```
create(Data, Profile, …)
hash(Data)
sign(Data, KeyID)
prepareSignature
```

- Application
- XML Signature Creation
- Crypto Module
- Key Module
- HSM
Signature Verification

Application → XML Signature Verification

verify(Signature, Profile, …)

XML Signature Verification → Crypto Module

hash(Data)

verify(SignatureValue)

Crypto Module → PKI Module

validate(Certificate)

PKI Module
Key Management

- Multiple Keystores in Key Module
  - PKCS#12 (File)
  - PKCS#11 (HSM)
    - Access to any PKCS#11 compliant Hardware through JNI
- No direct Access to Keys
- Reference Keys by ID
Configuration

- **Key Module**
  - PKCS#12 Files, PKCS#11 Modules

- **Crypto Module**
  - Software only, Hardware Acceleration

- **PKI Configuration**
  - Certificate Stores, Revocation Checking, ...

- **Logging**
  - Default - Log4J
Profiles

Specified as Java Interfaces

- XML Signature Creation
  - Keys, Algorithms, Transformations, Structure,..

- PKI
  - Trusted Roots, Validation Model, Revocation, ...

- XML Signature Verification
  - PKI Profile, Manifest Check, Return Data, ...

- CMS Signature Verification
  - PKI Profile
XML Signature Features

- **Algorithms**
  - RSA, DSA, ECDSA
  - SHA-1, SHA-256, RIPEMD-160, RIPEMD-128, MD5, MD2

- **Transformations**
  - (Exclusive) Canonicalization, Base-64, Enveloped-Signature, XPath Filter, XPath Filter 2, XSLT

- **Manifests**

- **ETSI Signed Properties**
Crypto Performance

- Asymmetric Crypto is the Slowest
  - Software
    - 50-100 Signatures/Second (RSA 1024, P IV 2G)
    - RSA, DSA, ECDSA are available (simple to add)
  - Hardware
    - up to ~1000 Signatures/Second (RSA 1024)
    - decreases with increasing Security Level
    - often only RSA, sometimes DSA, hardly ECDSA
XML Performance

- XML often slower than Crypto
  - Transformations are Critical
  - XSLT Stylesheet Transformations
  - XPath Filter
- Use Transformations Carefully
PKI Performance

- **Dynamic In-Memory Cache**
  - Certificates, CRLs
  - Verifies Certificate Signature only once

- **Permanent Store**
  - Database (SQL) or File System
  - Stores Certificates
  - Stores CRLs (sometimes >700k)
Outlook

- Implement a Web-Service Layer
- Support for Long-Term Signatures (ETSI – XAdES)
- Complete OCSP support
- Complete Support for XKMS
- Add Support for Timestamping
- Release Server Modules as a Product ([http://jce.iaik.tugraz.at](http://jce.iaik.tugraz.at))
References

- IAIK, Graz University of Technology
- Austrian Citizen Card (Buergerkarte)
  - http://www.buergerkarte.at/ (German)
- Chief Information Office, Austria
  - http://www.cio.gv.at/ (German)
- XML Signature
  - http://www.w3.org/Signature/
Questions and Answers

Questions?