Identity-Based Cryptosystems for Enhanced Deployment of OSGi Bundles

Pierre Parrend, Samuel Galice, Stéphane Frénot, Stéphane Ubéda
Pierre.parrend@insa-lyon.fr
Lab. CITI, 21, Avenue J. Capelle
69621 Vileurbanne Cedex
Introduction

• The Use Case
  – Management of System Update of Home Gateways
    • ADSL Modems, TV Set Top Boxes
  – Study realized in the context of the MUSE European Project

• The Problem
  – Secure Deployment
    • Need to authenticate the Gateways
  – Potentially Hundred of Thousands of Gateways
    • Key Management

• Our Proposition
  – Used Identity-based Cryptography
    • Instead of the classical Public Key Approach
Summary

- **Context**
- Identity-based Cryptosystems
- The OSGi Security Architecture with CZK-IBS
- Validation
Context

• Home Gateways
  – Currently linux-based
  – Runtime Extensibility required
  – The OSGi Platform (JSR 291) is a good candidate

• The OSGi Platform
  – Lightweight Overlay to Java
  – Makes Systems extensible at Runtime through the installation of `Bundles`
  – Becomes popular in embedded systems, Application Servers
    • Early adopter: Eclipse IDE
    • Automotive Entertainment, Home Gateways
    • IBM Websphere 6.1, JBoss
Context

• OSGi Security
  – Secure Deployment: Bundle Signature
    • Based on Sun Jar Signature
    • With more constraints: bundles can not be modified after Signature
    • Digital Signature using Public Key Cryptography: DSA/SHA1
  – Secure Execution: JVM + Java Permissions
    • Java Security
      – Type Safety, Bytecode Validation, Garbage Collector, Secure Class Loader
    • OSGi Architecture
      – Namespace isolation through Class Loaders, Management
    • Permissions
      – Java Customized Sandboxing
      – OSGi specific Permissions for Platform Management
Summary

• Context
• **Identity-based Cryptosystems**
• The OSGi Security Architecture with CZK-IBS
• Validation
Identity-based Cryptosystems

- Classical PKI
Identity-based Cryptosystems

- Principles
  - Implementation: Boneh & Franklin (2001)
Identity-based Cryptosystems

• Benefits
  – Certification scheme greatly simplified
    • No Public Key publication necessary
  – Only the Private Key Owner need to connect to the PKG
    • Performant in case of asymmetrical system
      – Few Private Key Owners, many users
      – Less remote communications, less risk, less cost
  – Suitable for frequent Key update
    • Revocation is performed transparently
Identity-based Cryptosystems

- Known limitations
  - The PKG MUST be trustworthy
  - If corrupted, all the system is broken
  - Key Escrow drawback (PKG knows all the private keys)

- Solution: Chen-Zhang-Kim signature scheme
  - Impersonification can be detected
  - No known issues about its security
Summary

- Context
- Identity-based Cryptosystems
- The OSGi Security Architecture with CZK-IBS
- Validation
The OSGi Security Architecture with CZK-IBS

• Initialization
  – The Public Key Generator has a Master Key S
  – The Entities that want a Private Key register at the PKG
  – The Entities that want to derive Public Keys make a request to the PKG
The OSGi Security Architecture with CZK-IBS

- Bundle Deployment
  - Bundle Digital Signature
    - By the Signer Si, with its Private Key
  - Publication
    - By the Signer
    - On a (more or less) public repository
  - Discovery, Download
    - By the Client
  - Bundle Signature Verification
    - Public Key of Si is derived from the PKG Parameters
    - Is the Signature for Si done with a Private Key issued by the PKG
  - Installation
    - And execution
The OSGi Security Architecture with CZK-IBS

• Bundle Deployment
Summary

- Context
- Identity-based Cryptosystems
- The OSGi Security Architecture with CZK-IBS
- Validation
Validation

- Benefits for Security Management
  - In comparison to classical PKI, with DSA Public Keys
    - No Public Key Publication
      - Less Communications, less risks
      - Only Signers need to communicate with the PKG
      - Make frequent Key update easier
    - No Complex Key Revocation scheme
    - Elliptic Curve Algorithms
      - Smaller key
      - Signature Verification speed is Higher
    - Make Bundle Signature Possible for highly asymmetric systems
      - Bundle providers: ~ 10
      - Client Gateways: ~ 10^6
Validation

• Benefits for Security Management

<table>
<thead>
<tr>
<th></th>
<th>PKI</th>
<th>IB-PKI</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Key Management</td>
<td>Public Key Dissemination</td>
<td>Public Key is Identifier</td>
<td></td>
</tr>
<tr>
<td>Key Revocation</td>
<td>Heavyweight</td>
<td>Transparent, through regular Key Update</td>
<td></td>
</tr>
<tr>
<td>Cryptographic Operations</td>
<td>Signature Speed</td>
<td>Key Size, Signature Verification Speed</td>
<td></td>
</tr>
<tr>
<td>CA Trust Level</td>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Coms with the CA</td>
<td>N+M</td>
<td>N</td>
<td>1/M</td>
</tr>
</tbody>
</table>
Validation

- Implementation
  - Java implementation using a multi-precision number library
  - Using original elliptic curves from Boneh-Franklin paper
  - Using bilinear pairings and map-to-point functions
  - All stuff embedded in SFelix framework
    - [http://sfelix.gforge.inria.fr/](http://sfelix.gforge.inria.fr/)
Conclusions

• Proposition
  – Modify the Algorithms used for Signing OSGi Bundles
    • Take advantage of a recent cryptographic scheme
    • Applied on a successful execution platform

• Validation
  – Prototype Implementation
  – Radical Management Gain
    • In the considered Use Case
    • Likely to be appliable in other contexts

• Limitation
  – PKG as Single Point of Failure
Perspectives

- Further development is needed
  - Field validation, to demonstrate asserted properties on real projects
  - Other Use Cases
    - Automotive Entertainment Service
    - Software distribution (cf. Eclipse)
Questions?
OSGi Platforms and Threats

- Overview of the OSGi Platform
OSGi Platforms and Threats
Secure OSGI Tool Suite

- Overview
- OSGi Signed Bundle

![Diagram of OSGi Signed Bundle](image-url)
Signning OSGi Bundles

1. Private and Public Key Distribution

2. Manifest.mf file generation
   new SignedJarManifest
     (JarFile, HashAlgorithm)

3. 'Signature File' generation
   new SignatureFile
     (signedManifest, signer)

4. 'Signature BlockFile' generation
   new SignatureBlock
     (SignatureFile, Cert, privateKey)

5. Signer Jar Archive Generation
   SignedJarFile.buildSignedJarFile
     (JarFile, SignedJarManifest, SignatureFile, SignatureBlock)
• OSGi Bundles Signature Verification

1. Signer Authentication
   KeystoreManager.isValid
   (Certificate, password)

2. Validation of Resource Order in Archive
   SignedJarFile.ResourceOrderValid()

3. 'Signature Block File' Validation
   CryptographicLibrary.
   checkCMSDataValidity(signatureFile, block)

4. 'Signature File' Validation
   CryptographicLibrary.
   checkHashValue(manifestHash, pretendedHash, algo)

5. Manifest Validation
   For each resource:
   CryptographicLibrary.
   checkHashValue(FileHash, pretendedHash, algo)
The OSGi Security Architecture with CZK-IBS

• Bundle Signature
  – Similar to OSGi R4 Bundle Signature
  – Only the used algorithms are changed
  – Principles
    ● Hash values for all resources are stored in the MANIFEST.MF File of the bundle
    ● Hash values of the Entries of the Manifest are stored in SIGNER.SF
    ● Digital Signature of SIGNER.SF is stored in SIGNER.ALG
      – ALG the Algorithm name
      – Modified CMS (Cryptographic Message Syntax = PKCS14) format
The OSGi Security Architecture with CZK-IBS

- Bundle Signature