Towards a Flexible Intra-Trustcenter Management Protocol

J. Buchmann, V. Karatsiolis, M. Lippert, A. Wiesmaier, A. Pitaev, M. Ruppert

TU Darmstadt
Problem

How can arbitrary trustcenter components communicate to each other?

Idea

Define a new protocol written in XML
FlexiTrust

- Flexi Trust trustcenter software
Communication

- Secured messages must be exchanged between these components
- PKCS#7 offers a container of data which can be secured
- Serializable Java objects but....
  - If interfaces change then libraries must be updated
  - What about C implementations?
  - New demands require programming
CMP

- Certificate Management Protocol (CMP)
  - RFC 2510
- CMP is very strong but complicated
- Written in ASN.1 (special libraries)
- Does not allow special features like dual control etc.
- Does not consider arbitrary trustcenter components
  - e.g. a Key Authority, a Backup Component, OCSP
Related Work

**Formats**
- PKCS#10
- XKMS
- CMS
- CRMF
- CMC

**Related Work**
- PKCS#7
- CMP
Protocol Criteria

How these criteria were set

- Based on other protocols (like CMP and XKMS)
- Experience while setting up trustcenters

1. Generality
   - All kind of data exchanged inside a trustcenter must be handled by the protocol.
   - No restriction in number and type of components
   - Therefore security can be applied to all messages
Protocol Criteria

2. **Extensibility**
   - Extended by new messages, new data, new components
   - Future requirements will be met

3. **Independency**
   - From TC structure and workflow
     - e.g. transport media, offline components, security mechanisms
   - All possible connections are supported as well as policies can be met

4. **Automation**
   - Process may be automated and the protocol must not prohibit this
Protocol Criteria

5. Scalability
   - Scalable with the security level and installation
   - Meet requirements for different TCs

6. Traceability
   - Messages, applications must be traceable
   - Human readable where possible
   - Support of auditing, special logging, easy debugging an error search

7. Security
   - All messages have to be secured (authentic, secret, unaltered)
   - Authorisation techniques (dual control), different algorithms
   - Data exchanged must be secured
XML

- Ideal for describing structured data
  - e.g. `<X509Certificate>`

- Portable format
  - XML supported from most languages

- Human and machine readable
  - Correctness of data
    - e.g. `<subjectDN>CN=Alice, o=TUD, c=DE</subjectDN>`

- Avoid ASN.1 structures
  - Special libraries, binary data
XML Signature

- XML data can also be signed! (and encrypted)
- XML Signature framework (W3C and IETF)
- Different algorithms
  - RSA, DSA, ECDSA in progress
- What you see is what you sign
- Certain portions can be signed
- Multiple signatures are allowed
- Signatures contain hints for the verification
An ITP message

<message version="1.0" id="123456789">
  <sender>Registration</sender>
  <recipient>Certification</recipient>
  <application id="12345678">
    <profile id="ProfileId">
      <element1Name>value</element1Name>
      <element2Name>value</element2Name>
      <!-- ... more elements -->
    </profile>
    <ds:Signature> <!-- signature elements... --> </ds:Signature>
    <!-- ... more signatures -->
  </application>
  <!-- ...... more applications -->
  <ds:Signature> <!-- signature elements... --> </ds:Signature>
  <!-- ...... more signatures -->
</message>
Tags

- **<message>**
  - An ITP message, version="1.0", id - The message ID

- **<sender>**
  - The component of TC sending the application
    - Symbolic Name, IP etc.

- **<recipient>**
  - The component of TC sending the application
  - Peer-to-Peer communication, flexible naming
Tags

- `<application>`
  - An application from one component to another
  - `id` - unique number identifying the application

- `<profile>`
  - `id` - identifies the kind of application
  - Encapsulates all necessary data for this type of application

- `<ds:Signature>`
  - The signature(s) over the application and/or message
Example

- Alice requests three certificates
- Virtual Hosting
- Offline CA
- Two operators must import the request to the CA
- Revocation password
- Send a mail to Alice with the certificates
- Public available certificates
RA to CA

<message version="1.0" id="20040202164445">
  <sender>Registration</sender>
  <recipient>Certification</recipient>
  <application id="20040202164832">
    <profile id="MultiCert">
      <clientName>Host A</clientName>
      <subjectDN>CN=Alice,OU=OrgUnitName,O=OrgName,C=DE</subjectDN>
      <revocationPassword>7c4a8 ... 8941c</revocationPassword>
      <email>alice@orgunitname.orgname.de</email>
      <publiclyAvailable>true</publiclyAvailable>
    </profile>
    <ds:Signature> other signature elements .... </ds:Signature>
    <ds:Signature> signature of first operator </ds:Signature>
    <ds:Signature> signature of second operator </ds:Signature>
  </application>
</message>
CA to Directory Services

<message version="1.0" id="20040202170134">
  <sender>Certification</sender>
  <recipient>Directory Services</recipient>
  <application id="20040202164832">
    <profile id="MultiCert">
      <clientName>Host A</clientName>
      <encCertificate>Base64 encoded certificate</encCertificate>
      <signCertificate>Base64 encoded certificate</signCertificate>
      <nonRepCertificate>Base64 encoded certificate</nonRepCertificate>
      <revocationPassword>7c4a8 ... 8941</revocationPassword>
      <email>alice@orgunitname.orgname.de</email>
      <publiclyAvailable>true</publiclyAvailable>
    </profile>
    <ds:Signature> other signature elements .... </ds:Signature>
  </application>
</message>
First ITP application

- A Web-Front end collects revocations
  - User provides the revocation password
- ITP messages are send to a Revocation Authority
- Messages are digitally signed from the server
- RevA issues indirect CRLs
Deployment

- TC which seek a way to arrange the information flow independent of
  - Number of components
  - Nature of components
  - Services
  - Kind of data exchanged between them and securely!

- ITP is ready for use in trustcenter specific applications
Future Work

- Define ITP v.2
- Specific set of messages
- Exception handling
- Critical flags
- Fine granular sign policy
- Standardise the protocol (W3C, IETF)
Contact

Thank you for your attention

Contact:

{buchmann,karatsio,mal,wiesmaie,epchtana,mruppert}
@cdc.informatik.tu-darmstadt.de