Embedded Java & Secure Element for high security in IoT systems

JavaOne - September 2014

Anne-Laure SIXOU - ST
Thierry BOUSQUET - ST
Frédéric VAUTE - Oracle
Anne-Laure SIXOU
Smartgrid Product Line Manager, ST
Anne-laure.sixou@st.com

Thierry BOUSQUET
Smartgrid Application Leader, ST
Thierry.bousquet@st.com

Frédéric Vaute
Master Principal Sales Consultant, Oracle
Frederic.vaute@oracle.com
Session objectives

What is security in IoT systems?

How to combine Embedded Java and a Secure Element to secure an IoT system?
Real-world Consumer IoT security today ...

Welcome to the “Internet of Things,” where even lights aren’t hacker safe
Malware attacks on Internet-connected Philips Hue lights cause blackouts.
by Dan Goodin - Aug 14 2013, 12:23am EDT

Meet the men who spy on women through their webcams
The Remote Administration Tool is the revolver of the Internet’s Wild West.
by Nate Anderson - Mar 11 2013, 1:02am EDT

Control panel backdoor found in D-Link home routers
D-secret is D-logon string allowing access to everything
by Richard Chirgwin, 12th October 2013

Your Home Appliances May be Spying on You

Smart Meters
Smart meters are a "time bomb" for utilities, warns insurance expert
Jul 23, 2014

Quick Take: As an industry, we’ve done a lot of thinking about the smart meter cost/benefit equation. But I wonder if we’ve adequately considered what would happen if smart meters make insurance rates go up? Two recent articles in the Insurance Journal suggest that the insurance industry is waking up to this new concern. – Jesse Brust

Cyber attacks on infrastructure have become a major worry for utilities, warns a recent article in the Insurance Journal. Traditionally, energy utilities have kept the grid safe by keeping it separate from the open Internet. But that is rapidly changing as smart meters connect customers to their utilities through the web.

Utilities claim customers have little to fear since those meters will use the same security measures as online banking. But “the risk is being underestimated outside of the industry,” said Eberhard Oehler, managing director of German utility Stadtwerke Ettlingen. A recent simulated attack came close to shutting down power to Ettlingen’s 40,000 residents. The experiment revealed that “sensitive, critical infrastructure is not sufficiently protected.”

Shows the importance of proper security analysis & practical pitfalls …
KERKEY & Embedded Java SE for SmartGrid
a “pre-industrial” tool for players

**KERKEY**
- Highly secure solution certified CC EAL4+
- Flexible solution Java OS and JavaCard application
- Turnkey solution with Industrialization services
  Compliant with European & BSI smart metering requirements

**Host Embedded Java SE**
- Portability on any operating system running on standard desktop system
- High performance system
- Reliable development platform highly deployed

[Video: Secure Smartgrid solution]
General security concepts
Security is the degree of resistance to, or protection from, harm. It applies to all vulnerable and valuable assets such as:

- Person
- Home
- Nation
- Community

There are two reasons why security should be an important item for everyone:

**Personal Protection of Information**

**Social Responsibility**

To protect the group you join when you connect your machine to the network.
Individuals or companies expect that their personal information contained in IoT products or systems

- Remains private
- Not to be subjected to unauthorized modification
- Be available to them
Security concepts and relationships

Owner

value

wish to minimize

evaluate

Threats

that exploit

Vulnerabilities / Attacks

Countermeasures

Risks

To

Assets

to protect

wish to abuse

Assets

Owner
Designing secure systems

- Infrastructure and set of rules

- Components
  - Secure devices (e.g. Microcontrollers)
    - performing crypto with
    - ... cryptographic keys
    - ... protected logically and physically
  - Software on other platforms
    - offering only limited protection to data and code
  - Central computers: hosts
  - Telecommunication infrastructure

- Set of participants, each with a specific role
  Every party has a set of rules he/she should follow
How to combine Embedded Java and a Secure Element to secure an IoT system?
From threats in Smart Metering …

- **Fake server**
  - Commercial & cyber crime usage
  - Disruption of administration system

- **Data Collection & Corruption**
  - Fraud for invoices

- **Concentrator / Gateway**
  - Misuse of private customer data

- **Data storage collection & corruption**
  - Identified empty houses
  - Fraud for invoices
  - Manipulated meter readings
  - Misuse of private customer data

- **Fake services**
  - Supply shut down – Disruption of service
  - Spread of wrong information (e.g., invoices)
  - Distributed denial-of-service
  - Malicious code

- **Fake server / device**
  - Supply shut down

- **Data Center**
  - Supply shut down – Disruption of service
  - Spread of wrong information (e.g., invoices)
  - Distributed denial-of-service
  - Malicious code

- **Meter**
  - Fraud for invoices
  - Supply shut down
… to Countermeasures

- Fake server
  - Authentication

- Data Collection Corruption
  - Authenticated & encrypted communication

- Concentrator / Gateway
- Meter

- Fake services
- Secure boot / code integrity

- Fake server / device
- Authentication

- Data storage collection & corruption
  - Data integrity
  - Data confidentiality
  - Tamper-resistance
  - Secure life cycle
## The solution with Java and ST products

<table>
<thead>
<tr>
<th>Threats</th>
<th>Solution</th>
<th>Implementation &amp; services requested</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake devices</td>
<td>Authenticated devices</td>
<td>Mutual authentication</td>
</tr>
<tr>
<td><strong>Data collection &amp; corruption</strong></td>
<td>Authenticated &amp; encrypted communications (secure channel)</td>
<td>Expertise, SW Crypto libraries, HW Crypto accelerators, Robust implementations, Network security protocols, Evaluated / Certified, Authenticated software stacks, Least privilege, Sand-boxing &amp; Isolation of assets, Detection &amp; Monitoring</td>
</tr>
<tr>
<td>Fake services</td>
<td>Robust Smart-Devices (secure boot &amp; code integrity)</td>
<td></td>
</tr>
<tr>
<td><strong>Data storage collection &amp; corruption</strong></td>
<td>Protected crypto keys &amp; private data (data integrity, data confidentiality and tamper-resistance)</td>
<td>From PCB attacks, From SW attacks, From sophisticated HW attacks, Provisioning of secrets in ST chips, Support for sophisticated multi-stakeholders scenarios &amp; field management</td>
</tr>
</tbody>
</table>
IoT requires smarter and more secure devices

Local intelligence and decision-making
Flexible networking
Performance and scalability
Security
Remote management
Functions become services
Java platforms on ST chipsets

- Cortex®-Mx
- Cortex®-A9
- ST23
- ST40

Footprint:
- 50KB-1MB
- 1MB-10MB
- 10MB-100MB

Java Card
Java ME Embedded
Java SE Embedded
OEP for Oracle Java

SECURITY
SMALL
MEDIUM
LARGE

Cortex®-SC
Cortex®-Mx
Cortex®-Mx
ST40

Cortex®-A9
Java SE Security Overview
Secure and controlled code execution

• Runtime security: “Sandbox” Concept
  • Controlled code loading
  • No file access on host, limited network access, no native code execution

• Security Manager / AccessController
  • Limits access to resources and data by means of runtime security

• Security Policy
  • Configurable definition of the limits of the Security Manager (permissions)

• Domains
  • Act as instances of Security Policy
  • Define access for different areas of code through source of the request
Java SE Cryptography Architecture (JCA)

- JCA (Java Crypto Architecture) Provides an extensible, full featured API for building secure applications
- Algorithm and implementation independent
- Provider-based architecture
  - Allows extension of Java Security to hardware based security with Secure Element

Software Crypto Provider

From software to hardware based security
Secure Element Growth Drivers

Personal Security

- Contactless platform: ST31
- e-Flash flexibility

Embedded Security

- NFC secure element: ST33
- NFC combo: SE + CLF

Consumer & Industrial

- End-to-end turn key solutions
- Hardware, Software, Perso

SE = Secure element
CLF = Contact-Less Frontend
KERKEY
Secure element for smartgrid system

- Highly secure solution certified CC EAL4+ (Hardware – firmware – personalization)
  - Java platform with modular Java Card application
  - Industrialization & Personalization services
  - QFN32 suitable package for Smart metering & Industrial design
Leading edge methodology for Security

Material / IP theft
- Secure manufacturing and development environments and flows
- Product life cycle management
- Internal & external audits, regular maintenance

Physical attacks
- Shields
- Intrusion detectors
- Internal assessment: best in class labs and methodology
- Silicon technology advanced analysis tools
- External Labs assessment - R&D projects, Technology watch

Fault injection
- Dedicated architecture and design
- Detectors
- HW and SW countermeasures

Side channel analysis
- Crypto and product architecture and design
- Solid crypto SW

Evaluation and Certification by public authorities, Common Criteria, EMVCO, FIPS …
Smartgrid solution architecture

**Energy Provider data Center**

**Concentrators (TLS server)**
- STIH416 processor: Cortex®-A9
- Linux + Java SE Embedded + ST Kerkey
- Java Card + Global Platform v2

**Secure Smart meters (TLS client)**
- ST STM32 + ST Kerkey
- Java Card + Global platform v2
# High level Security can be reached if Kerkey is added to Java solution

<table>
<thead>
<tr>
<th>Threats</th>
<th>Solutions</th>
<th>Java</th>
<th>Kerkey + Java</th>
<th>Kerkey Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fake Devices</strong></td>
<td>Authorized Devices</td>
<td>Mutual authentication</td>
<td>STD</td>
<td>HIIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expertise</td>
<td>STD</td>
<td>HIIGH</td>
</tr>
<tr>
<td>Data collection &amp; corruption</td>
<td>Authenticated &amp; Encrypted Communications (secure channel)</td>
<td>SW Crypto libraries</td>
<td>STD</td>
<td>HIIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW Crypto accelerators</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Robust implementations</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network security protocols</td>
<td>STD</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Evaluated / Certified</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td>Fake Services</td>
<td>Robust Smart-Devices (secure boot &amp; code integrity)</td>
<td>Authenticated Software Stacks</td>
<td>STD</td>
<td>HIIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Least privilege, Sand-boxing &amp; Isolation of assets</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Detection &amp; Monitoring</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td>Protected Crypto Keys &amp; private data</td>
<td>From PCB attacks</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>From SW attacks</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>From sophisticated HW attacks</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td>Security Provisioning &amp; Life Cycle Mgt</td>
<td>Secrets provisioning in ST chips</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Support for sophisticated multi-stakeholders scenarios &amp; field management</td>
<td>NA</td>
<td>HIGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exemple of high level security Java solution with Kerkey

<table>
<thead>
<tr>
<th>Threats</th>
<th>Solutions</th>
<th>Java</th>
<th>Kerkey + Java</th>
<th>Kerkey Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake Devices</td>
<td>Mutual authentication</td>
<td>STD</td>
<td>HIGH</td>
<td>Java key store is protected inside Kerkey</td>
</tr>
<tr>
<td>Data collection &amp; corruption</td>
<td>SW Crypto libraries</td>
<td>STD</td>
<td>HIGH</td>
<td>New security provider can be added to JCE/JCA to extend cryptographic features With Kerkey</td>
</tr>
</tbody>
</table>
Typical Software architecture

Meter MCU or Concentrator MPU

Operating system

Java SE or ME

ST PKCS11 Middleware

SUN PKCS#11 Provider

JCE/JCA

Java application

Secure element Kerkey

Java Card application : smart metering

Java Card 2.2

GP 2.1.1

Operating system

Serial link

ISO7816 or I2C
Demo for developers
Demo usage of Kerkey secure element with Java SE & Java card

• Demo 1 : Open a secure session from Java
  Read CPLC data's are often used to identify the chip in the field

• Demo 2 : Generation of certificate signature request using Kerkey
  • A certificate is an electronic document used to prove ownership of a public key
  • It allows to authenticate documents, open secure channel SSL, etc ...
  • Certificate signature request is one part of the creation of the certificate
  • It allows newly generated signature to be signed by Certificate Authorities.
Conclusion
From security in Smart metering

and the future…

to security in IoT or IT systems

For more information of how to address Smart Home system, visit “Universal Development Kit for Creating and Deploying Smart Home/Building Applications [CON2405] session”
Contacts

• Join ST people
  • USA : Serge.fruhauf@st.com
  • APAC : Bruno.batut@st.com
  • EMEA : Fabrice.gendreau@st.com
  • Japan / Korea : Michel.faure@st.com

• www.st.com / kerkey