Device Attestation using Java Card

Patrick Van Haver
Security Solutions Architect
Oracle - Java Platform Group

Nicolae Bors
Software Engineer
Oracle - Java Platform Group

Dec. 2020
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle’s products may change and remains at the sole discretion of Oracle Corporation.
Agenda

Purpose of Device Attestations

How Device Attestation works?

Device Attestations using Java Card – Demo

Conclusion
Agenda

Purpose of Device Attestations

How Device Attestation works?

Device Attestations using Java Card – Demo

Conclusion
Purpose of Device Attestations

• Get reliable evidence on the characteristics and state of a device
  – Device identity and manufacturer,
  – Security state and capabilities,
  – Software versions installed,
  – Location
  – ...

• Typically used
  – to detect rogue devices during on-boarding,
  – to perform remote monitoring and enforce security policies,
  – to manage device lifecycle, detect non-updated or tampered devices,
  – ...

[Diagram showing cloud service interacting with various devices: a car, a building, a network device, and possibly a device with a lock symbol, indicating security measures.]
Entity Attestation Tokens

Requirements
- Self-contained (no dependency on protocol)
- Extensible list of claims
- Simple and compact encoding
- Support for integrity, authenticity and confidentiality
- Supports for multiple signing and encryption schemes

Token structure based on existing standards, and extended with specific claims
- Either JSON Web Token (JWT – RFC7519),
- Or CBOR Web Token (CWT – RFC8392), CBOR Object Signing & Encryption (COSE – RFC8152)
Agenda

Purpose of Device Attestations

How Device Attestation works?

Device Attestations using Java Card – Demo

Conclusion
How Device Attestation works?

**Actors**

- **Relying party**: Service Provider who wants to get reliable information from a device (characteristics, state, ...)
- **Verification service**: Service to verify authenticity of attestation tokens

**Device**

- **Device application**: Secure Element
- **Attestation service**: Produces and digitally sign attestation tokens
How Device Attestation works?

Key Provisioning

Relying party

Verification service

Attestation keypair

Public key

Private key
How Device Attestation works?

*Key Provisioning – using HSM*
How Device Attestation works?

Key Provisioning – using on-board key-generation
How Device Attestation works?

*Attestation Request*

1. **EAT request**

```json
request {
    nonce: "eH04-fhtZ#",
    ueid: "",
    sec_rating: "",
    ...
}
```

List of characteristics to retrieve from device

- **Device application**
- **Secure Element**
- **Attestation service**
- **Relying party**
- **Verification service**
How Device Attestation works?

Attestation Token generated by Secure Element

Relying party

1. EAT request

Verification service

Device application

Device

Secure Element

Create attestation token and sign it with private key

[  
header { alg: "ECDSA P-256 SHA-256" } 
claims { 
nonce: "aH04-fhtZ#" 
ueid: "017F123A051012...", 
sec_rating: "15", 
... 
} 
signature { ... } 
]
How Device Attestation works?

Attestation Response

1. EAT request
2. EAT response

Relying party → […] → Device application → Secure Element

Device application

Attestation service

Verification service
How Device Attestation works?

**Attestation Verification**

1. **EAT request**
2. **EAT response**
3. Verification request
4. Verification result
How Device Attestation works?

More complex scenarios

• Nested Entity Attestation Tokens
  – To get information from multiple modules within the device
  – Each signed by the corresponding module, using its own key

• Privacy, Confidentiality
  – Each EAT can also have its claims encrypted to ensure confidentiality

```json
[  header { alg: ECDSA P-256 SHA-256 }  
  claims {  
    ...  
    submods:  
    [  
      header { ... }  
      claims { ... }  
      signature { ... }  
    ]  
  }  
  signature { ... } ]
```
Agenda

Purpose of Device Attestations

How Device Attestations work?

Device Attestations using Java Card – Demo

Conclusion
Device Attestation using Java Card

- Communication with remote server using device communication stack
- EAP API used to delegate process to the Attestation Service running in the Secure Element
- Attestation Service is a Java Card Applet
- EAT/CBOR library used to encode, decode and sign attestation tokens
Demo

Architecture

Verification service

Relying party

Device

Secure Element

EAP API

Network stack

EAT/CBOR API

Device application

TPS client API

Java Card platform

Attestation service

Relying party

EAT Request

EAT Response

Client

Claim

Value

Response

Client

Claim

Value

Signature

Service Provider

EAP Identity

Verification service

Relying party

Java Card platform

Musca-B1 test Chip Board

arm

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.
Device

SoC

SSE-200 subsystem

Network stack

EAP API

TPS client API

CI driver

S-RAM Cortex M33 Cortex M33

Flash Peripherals

Device application

EAT/CBOR API

Attestation service

Java Card Runtime Environment

Java Card API

Java Card Virtual Machine

Cortex M0+

Crypto Peripherals drv Alarms

Power ctrl Assets mgt Debug ctrl

Messaging Lifecycle ctrl Secure boot

mailbox RNG CryptoCell Timers peripherals OTP Secure RAM

Example of claims used for demo

Based on current IETF draft for EAT: https://www.ietf.org/archive/id/draft-ietf-rats-eat-04.txt

Nonce: Arbitrary number generated by the relying party

Universal Entity ID: UEID's identify individual manufactured entities / devices [...] UEID's must be universally and globally unique across manufacturers and countries.

Security level: Describes security environment and countermeasures available on the end-entity / client device where the attestation key reside and the claims originate. {Unrestricted:1, restricted:2, secure-restricted:3, hardware:4}

Based on GP Entity Attestation Protocol draft

Security rating Provides information about how secure the Entity is. {unknown: 0, basic: 5, substantial: 10, high: 15}

Card configuration The configuration the Secure Element complies to. “GP Compact IoT Configuration 1.0 with asymmetric crypto”: 2A 8648 86FC6B 05 0A 01 00 00 01

Secure Channel Protocol The Secure Channel Protocol used by Issuer Security Domain. “GP Secure Channel Protocol 03 option i=70”: 2A 8648 86FC6B 04 03 70

Java Card Version 3.1.0
Agenda

Purpose of Device Attestations

How Device Attestations work?

Device Attestations using Java Card – Demo

Conclusion
Device Attestation using Java Card

**Benefits**

**Secure Runtime**
- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

**Portable**
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

**Adaptable & Extensible**
- To support multiple attestation schemes
- To extend attestation service and include application specific claims

**Manageable**
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme
Device Attestation using Java Card

**Benefits**

**Secure Runtime**
- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

**Portable**
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

**Adaptable & Extensible**
- To support multiple attestation schemes
- To extend attestation service and include application specific claims

**Manageable**
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme
Device Attestation using Java Card

**Benefits**

**Secure Runtime**
- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

**Portable**
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

**Adaptable & Extensible**
- To support multiple attestation schemes
- To extend attestation service and include application specific claims
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme

**Manageable**
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme
Device Attestation using Java Card

**Benefits**

**Secure Runtime**
- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

**Portable**
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

**Adaptable & Extensible**
- To support multiple attestation schemes
- To extend attestation service and include application specific claims
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme

**Manageable**
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.
Device Attestation using Java Card

**Conclusion**

**Secure Runtime**
- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

**Portable**
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

**Adaptable & Extensible**
- To support multiple attestation schemes
- To extend attestation service and include application specific claims

**Manageable**
- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme
More Information

https://www.oracle.com/java/technologies/java-card-tech.html

**Java Card Platform Specification 3.1**
Latest release of the Java Card specification and the reference for Java Card products.

**Java Card Development Kit Tools**
The Java Card Development Kit Tools are used to convert and verify Java Card applications. The Tools can be used with products based on version 3.1, 3.0.5 and 3.0.4 of the Java Card Specifications.

**Java Card Development Kit Simulator**
The Java Card Development Kit Simulator includes a simulation component and Eclipse plug-in. Combined with the Java Card Development Kit Tools, it provides a complete, stand-alone development environment.

**Java Card IoT and Security blog**
This Blog covers the latest Java technology for small devices and security in the IoT, mobile, ID and Payment

- Webcast – Secure Business Runs Java Card
- Webcast – How to secure IoT Edge with Java Card
- Webcast: Oracle Java Card 3.1 Boosts Security for IoT Devices at the Edge

contacts: patrick.van.haver at oracle.com, nicolae.bors at oracle.com