A Simple and Secure E-Ticketing System for Intelligent Public Transportation based on NFC

Ivan Gudymenko  
Felipe Sousa  
Stefan Köpsell
Outline

Introduction

System Description

Validation
Outline

Introduction

System Description

Validation
Intelligent Public Transportation and Urban Environment of the Future

- Intelligent Transportation Systems (ITS) are going to shape the urban environment of the future

- Public transportation is an integral part of ITS
E-ticketing in Public Transportation

[Courtesy of Münstersche Zeitung.de]
The Notion of an E-ticket

- A digitalized version of a travel permission (e.g., a token)
- Stored on a user device:
  - Smart Card
  - NFC-enabled smart phone
Non-interactive vs. Interaction-based E-ticketing Systems

- Non-interactive

- Interaction-based
Non-interactive vs. Interaction-based E-ticketing Systems

- Non-interactive

- Interaction-based
Outline

Introduction

System Description

Validation
Our Goal

- Providing a digital alternative to a conventional paper-based ticketing
- Based on open source components
- NFC-enabled smart phone as a user device
Many Cards – One Single App

Glasgow

Karlstad

Amsterdam

Porto

eTicket app

Interoperable solution
System Main Actors

- **User Device**
  - eTicket
  - app

- **Vending Machine**

- **Stamping Machine**
  - Stamp

- **Checking Machine**

---

A Simple and Secure E-Ticketing System

I. Gudymenko, F. Sousa, S. Köpsell

TU Dresden, UFCG
Core Processes Considered

(1) **E-ticket acquisition:**
   a) *(online)* via a Web interface
   b) *(offline)* via NFC with a vending machine

(2) **E-ticket stamping:**
   a) *(offline)* via NFC with a stamping machine

(3) **E-ticket validation**
   a) *(offline)* via NFC with a checking machine

→ The processes are implemented through corresponding protocols (see further)
Main Requirements

1. Open source components
2. Offline stamping and checking (as opposed to vending)
3. Ticket unforgeability
4. Protection from replay attacks
5. Ticket unclonability
6. Double spending prevention
7. Timing (especially for stamping and checking)
An E-ticket

- Essentially is a digital token
- Describes the acquired travel permission
- E-ticket is bound to the user’s public key $PK_u$
- Different ticket types are supported through attributes
  - time-bounded (e.g., hourly tickets)
  - single ride
  - and many more...

<table>
<thead>
<tr>
<th>Attributes</th>
<th>VM Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PK_u$</td>
<td>$sig_{vm}(h(attr))$</td>
</tr>
<tr>
<td>$ID_{vm}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PK_u$</td>
</tr>
<tr>
<td>$ID_{vm}$</td>
</tr>
<tr>
<td>Serial number</td>
</tr>
<tr>
<td>Time limit</td>
</tr>
<tr>
<td>Zone</td>
</tr>
<tr>
<td>Zone</td>
</tr>
<tr>
<td>Price</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Protocols: Vending

A Simple and Secure E-Ticketing System
I. Gudymenko, F. Sousa, S. Köpsell
TU Dresden, UFCG
Protocols: Vending

- **Actors:** E-ticket app (EA) and a Vending machine (VM)
- Either online or offline
- EA requests a certain ticket type and sends user public key $PK_u$
- After payment, VM issues an e-ticket binding it to $PK_u$
- EA verifies the e-ticket
Protocols: Stamping

A Simple and Secure E-Ticketing System
I. Gudymenko, F. Sousa, S. Köpsell
TU Dresden, UFCG
Protocols: Stamping

- Actors: **E-ticket app (EA)** a **Stamping machine (SM)**
- Offline
- Stamping essentially activates the e-ticket for a ride
- As a result, a stamp is obtained

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Serial #</strong></td>
<td>e-ticket number</td>
</tr>
<tr>
<td><strong>Timestamp</strong></td>
<td>stamping time</td>
</tr>
<tr>
<td><strong>ID_{st}</strong></td>
<td>station ID</td>
</tr>
<tr>
<td><strong>ID_{sm}</strong></td>
<td>SM ID.</td>
</tr>
</tbody>
</table>
Protocols: Stamping

1: sendNonceAndPKu()

1.1: sendCertSMAndSigChal()

2: checkCertSMAndSigChal()

3: resolveChal()

4: sendChalAnswerAndTicket()

4.1: checkChalAnswer()

4.2: checkTicket()

4.3: stampTicket()

4.4: sendTicketStamp()

5: checkTicketStamp()

6: ACK
Protocols: Checking

A Simple and Secure E-Ticketing System

I. Gudymenko, F. Sousa, S. Köpsell

TU Dresden, UFCG
Protocols: Checking

- Performed between an **E-ticket App** and a **Checking machine**
- Follows similar pattern as the stamping protocol
- Up to the point where the previously obtained stamp is checked:
  - the e-ticket must be in *stamped* and *valid* state
Outline

Introduction

System Description

Validation
Practical Evaluation

- Terminal part:
  - NFC front-end: PN532 RFID/NFC Breakout Board
  - Controller and logic: Raspberry Pi Model b

- Customer device:
  - Samsung Galaxy Nexus GT-I9250
  - Android 4.4 OS
Performance Figures and Demo

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Execution time</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RSA-1024</td>
<td>RSA-2048</td>
</tr>
<tr>
<td>Vending</td>
<td>0.09 s</td>
<td>0.12 s</td>
</tr>
<tr>
<td>Stamping</td>
<td>3.85 s</td>
<td>4.65 s</td>
</tr>
<tr>
<td>Checking</td>
<td>3.33 s</td>
<td>4.23 s</td>
</tr>
</tbody>
</table>

- And now a short demo is going to be presented
Conclusion

- The designed e-ticketing system has been presented
- It is based on open source components
- The first results of practical evaluation are feasible
Thank you very much for your attention! Do you have any questions/comments/suggestions?
Back up slides
What an E-Ticket is NOT

- A widely used “online ticket” (air transport, etc.)
- Pointing to the respective entry in the back-end DB
Adopted Attacker Model

(1) **External attackers**
   a) An observing attacker
   b) A modifying attacker (spoofing, masquerading)

(2) **Internal attackers**
   a) A user trying to forge/clone an e-ticket
   b) Vending machine issuing invalid e-tickets
   c) Stamping machine providing an incorrect stamp
   d) A conductor framing the user as having an invalid e-ticket
# Protocols: Stamping (Detailed)

<table>
<thead>
<tr>
<th><strong>E-ticket app (EA)</strong></th>
<th><strong>Stamping machine (SM)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>((PK_u, SK_u), \text{eticket})</td>
<td>(\text{cert}_{sm})</td>
</tr>
</tbody>
</table>

**Generate random** \(r_e\)

\[
\text{Generate random } r_e, PK_u \\
\text{Create challenge: } \text{chal} \leftarrow (r_e | r_{sm}) \\
\text{Encrypt } \text{chal} \text{ under } PK_u: ce \leftarrow E_{PK_u}(\text{chal}) \\
\text{Sign } ce \text{ using } \text{cert}_{sm}: \text{csign} \leftarrow \text{Sign}_{\text{cert}_{sm}}(ce) \\
\]

\[
\text{Check } \text{cert}_{sm}, \text{ check } \text{csign} \\
\text{Decrypt } ce, \text{ extract } r'_{sm} \\
\text{Compute the answer: } ans \leftarrow h(r'_{sm}) \\
\]

\[
\text{Check } ans, \text{ check if } \text{eticket} \text{ is bound to } PK_u \\
\text{Create a signed stamp for the } \text{eticket:} \\
\text{(stamp, ss} \leftarrow \text{Sign}_{\text{cert}_{sm}}(\text{stamp})) \\
\]

**Verify stamp and the signature ss**