Contract Based Late Security Binding

Ioannis Sakarelis, Thomas Strang, Thaddäus Dorsch, Patrick Robertson

DLR Oberpfaffenhofen
Multimedia Car Platform

- Open platform: Open for applications of any provider
- Security model by certification authority
- Vendor (car or terminal) implements own Look&Feel
- Uses any available network
- Terminal is open for any input/output technology (MMI; e.g. voice controlled input, video output etc.)
- Definition of APIs for integration of network, navigation, car data - bus, phone
  → controlled access to any resource by any application
In-Car Services

- **Classic Services**
  - radio, phone, (TV, WWW)

- **Car-specific Services**
  - Navigation, Location Based Services
  - Device Control (Heater, Properties, Bordcomputer)
  - Emergency Services

- **New and Combined Services**
  - combination of navigation and information
    dynamic routing, traffic control,...
  - local services and local information by short range
    networks (10m – 1km; Bluetooth, WLAN, DVB-T short
    range)
  - automatic drive-through-payment (toll, parking,...)
  - remote diagnostics, remote garage service
  - car as sender/relay station as part of the network
Modular Car Terminal

External Services
- UMTS, DVB, DAB
- GPS, Galileo
- WLAN, Bluetooth

Network Interface
- "electronic wallet"
- personal data (addresses, schedule, keys)
- personal applications
- CPU
- no or minimum UI

CAR TERMINAL

In-Car Services
- Database
- Smart Caching
- Navigation
- Car Data

User Interface
Car Terminal, PDA, Laptop…
Security Requirements

High security requirements on a car multimedia terminal:

- **Data protection**: personal data, car data, context (e.g. current location)
- **Authentification** of application developer and trust center
- **Usability**: *Driver must not be distracted!*
  - terminal is responsible for (automatic) application control: priorities, access rights
  - legal restrictions (country dependent) must be considered
  - robust runtime behaviour
  - car manufacturer must be able to give selective access to car resources

→ MCP designed security architecture for spatio-temporal distributed authentication and authorisation
Certification Procedure

Trust Center issues keys and certificates after negotiation of a legal contract with application provider

- **Access rights** of applications on the terminal; priorities (*authorisation*)
- **Automatic identification** of applications by the terminal (*authentification*)
- **Responsibility** for application can be traced back to the application developer based on a legal contract
- **Applications** may be, but are not required to be tested or analysed on conformance to the legal contract
- **Several levels of security** are possible
Contract Negotiation

- Trust Centre
- Application Body
- Security Body
- MCP Terminal
- Application Developer

Legal Contract

Application Keys

- List of accessible API’s
- Allowed resources (eg. files, network connections)
- Service groups or single services
- Context rules
- Priority rules
Passive Security Body

Application Body

Application

signature: application key

Security Body

Access Rights
Permission Request File; resources, APIs, Jini services,...

public application key

signature: trust center
Active Security Body

Application Body

Application

signature: application key

Security Body

Access Rights
(MHP Permission Request File; resources, APIs, Jini services,...)

Security Manager Plug-in

application key - public

signature: trust center

“Policeman“
Contract Based Late Security Binding, Eurescom Summit 2002, Heidelberg, Germany

Classic Java Signing

1. deliver application
2. check application
3. sign & integrate application into the file system
4. verify file system
5. initiate application in sandbox
6. start application

User's VM

Trust Center TC

Applic. Developer AD

Application & PRF check

Transmit all

Classic Java Signing
Our Approach 1: Lax Certification Procedure

1. User’s VM requests certification
2. Contract (liability, application access rights, etc)
3. Generate application key pair
4. Public application key
5. Security body
6. Generate application and application body
7. Application body, security body
8. Check signature of trust center
9. Check application key signature
10. Start application
Our Approach 2: More Restrictive Certification Procedure

1. Trust Center requests certification
2. Contract (liability, application access rights, etc)
3. Generate application key pair
4. Create application
5. Application
6. Generate application body and security body
7. Application body and security body
8. User’s VM
9. Check signature of trust center
10. Check application key signature
11. Start application
Lax Procedure

• One *reusable* basic security body per application provider (can be reused for each updated version of the application)
• Security depends more on the legal binding to avoid misuse
• Private application key is owned by the application developer

Restrictive Procedure

• Every application must be passed to Trust Center
• A new security body for each application version
• Private application key is owned by the trust centre
Experiences from MCP
- New Services
- Context Awareness
- Car terminal has high security requirements
  → spatio-temporal distributed authentication and authorisation

Architecture enables **contract based security**

**Late Security Binding** by independent distribution of application body and security body
Thank you!

Thomas Strang <thomas.strang@dlr.de>
Demand Oriented App. Classification

- Former application classification **device oriented** (radio, video, navigation,...)

- MCP: **Demand oriented classification**: Feeling Comfortable, Having Fun, Being Informed, Getting Somewhere,....