Proposal for a SEVECOM SW Architecture

Frank Kargl
frank.kargl@uni-ulm.de

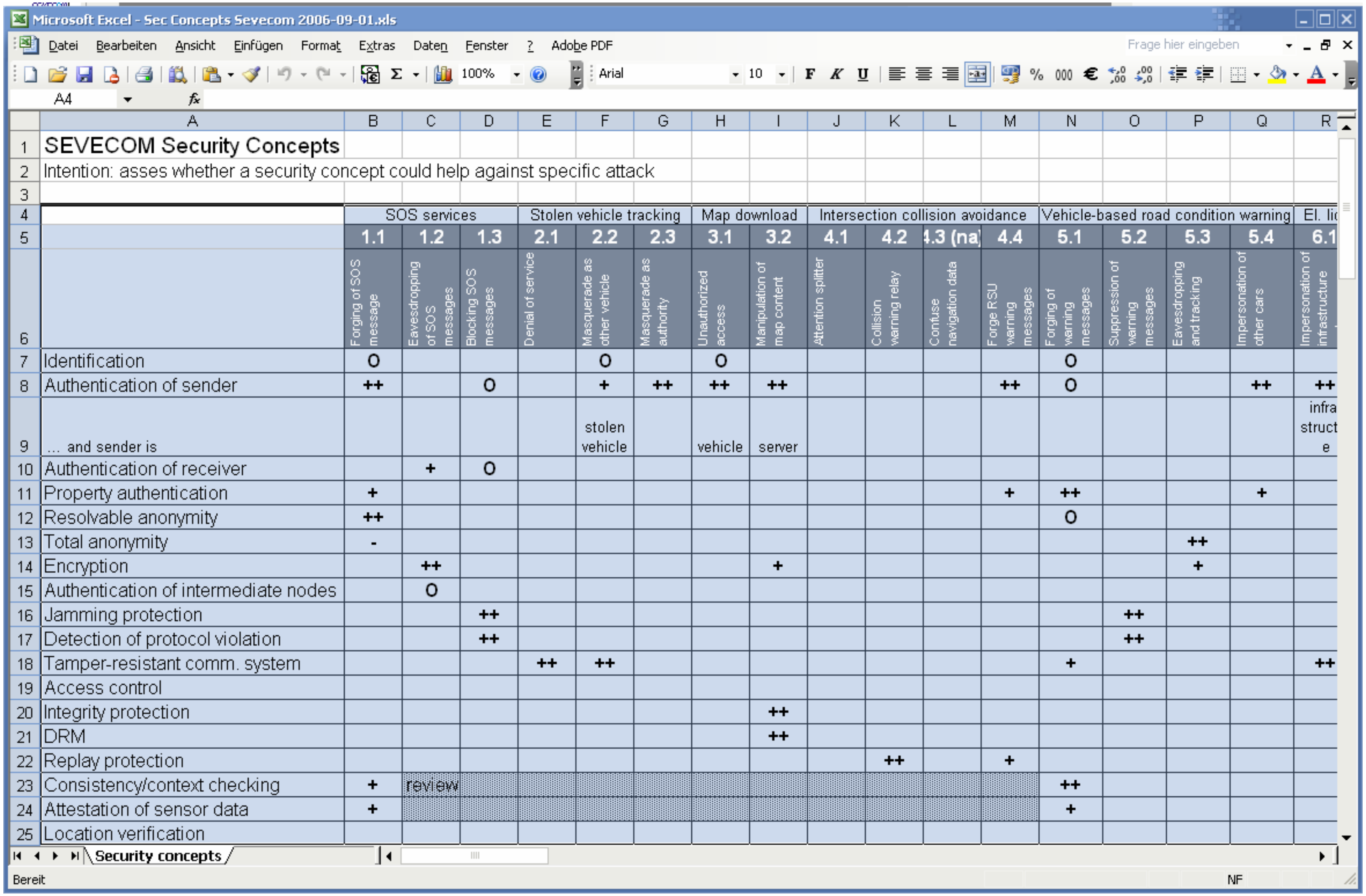
Media Informatics Dep.
Ulm University

Budapest Meeting, 4./5. Sept. 2006

The Problem

- SEVECOM Requirements Engineering discovered that (at least) the following security modules are needed:
  - Identification
  - Authentication of sender
  - Authentication of receiver
  - Authentication of intermediate nodes
  - Property authentication
  - Resolvable anonymity
  - Total anonymity
  - Encryption
  - Jamming protection
  - Tamper-resistant comm. system
  - Access control
  - Integrity protection
  - DRM
  - Replay protection
  - Detection of protocol violation
  - Consistency/context checking
  - Attestation of sensor data
  - Location verification
  - …

Not all modules are active all the time



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	SEVECOM Security Concepts																	
2	Intention: asses whether a security concept could help against specific attack																	
3																		
4		SOS services			Stolen vehicle tracking			Map download		Intersection collision avoidance				Vehicle-based road condition warning				El. li
5		1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	4.1	4.2	4.3 (na	4.4	5.1	5.2	5.3	5.4	6.1
6		Forging of SOS message	Eavesdropping of SOS messages	Blocking SOS messages	Denial of service	Masquerade as other vehicle	Masquerade as authority	Unauthorized access	Manipulation of map content	Attention splitter	Collision warning relay	Confuse navigation data	Forge RSU warning messages	Forging of warning messages	Suppression of warning messages	Eavesdropping and tracking	Impersonation of other cars	Impersonation of infrastructure
7	Identification	O				O		O						O				
8	Authentication of sender	++		O		+	++	++	++				++	O			++	++
9	... and sender is					stolen vehicle		vehicle	server									infra struct e
10	Authentication of receiver		+	O														
11	Property authentication	+											+	++			+	
12	Resolvable anonymity	++												O				
13	Total anonymity	-														++		
14	Encryption		++						+							+		
15	Authentication of intermediate nodes		O															
16	Jamming protection			++											++			
17	Detection of protocol violation			++											++			
18	Tamper-resistant comm. system				++	++								+				++
19	Access control																	
20	Integrity protection								++									
21	DRM								++									
22	Replay protection										++		+					
23	Consistency/context checking	+	review											++				
24	Attestation of sensor data	+												+				
25	Location verification																	

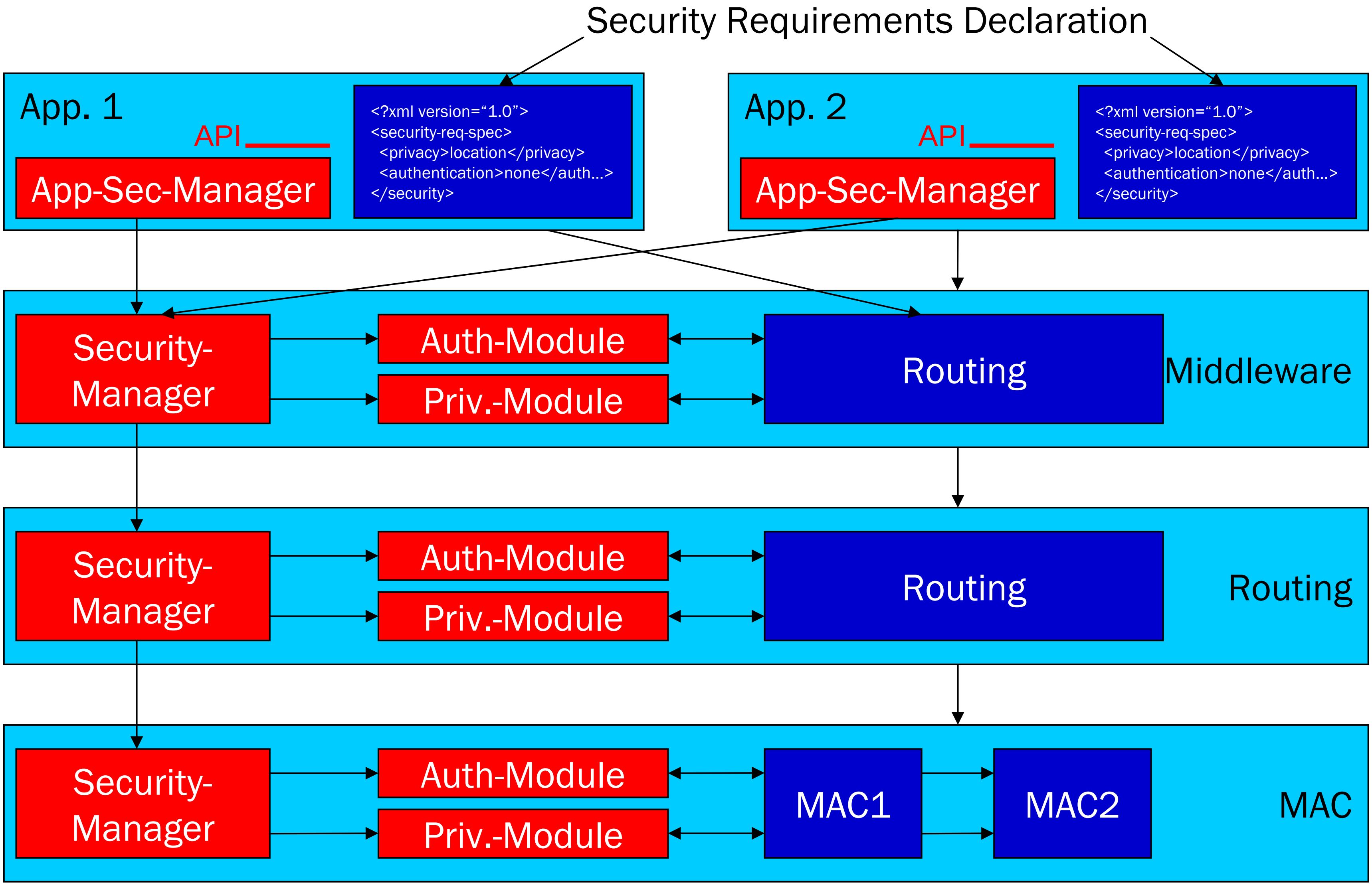
Problems

- Some modules influence each other
  - E.g. Authentication vs. Anonymity
- Modules are located on different layers
  - E.g. Anonymity requires changed IDs on MAC-, IP-, application-layer
- Will the security system needs to be changed, when new applications are installed?

➔ Solution: Security architecture which is

- Modular
- Flexible
- Dynamically configurable at runtime

SW Architecture Proposal



```
<?xml version="1.0">
<security-req-spec>
  <privacy>location</privacy>
  <authentication>none</auth...>
</security>
```

```
<?xml version="1.0">
<security-req-spec>
  <privacy>location</privacy>
  <authentication>none</auth...>
</security>
```

Security Requirements Specification

- Syntax could be
  - XML-based
  - Resource Description Framework / RDF
    - Similar e.g. to CC/PP
- Example

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="..." xmlns:sv="http://www.sevecom.org/schema#">
  <rdf:Description rdf:about="http://www.c2c-cc.org/vehicle-based_road_cond_warning">
   <rdf:type rdf:resource="esafetyApplication"/>
    <sv:requires>
     <sv:SecurityRequirement module="PropertyAuthentication">
      <sv:nodeType>Vehicle</sv:nodeType>
     </sv:SecurityRequirement>
    </requires>
    <requires>
     <sv:SecurityRequirement module="Privacy">
      <sv:idPrivacy changeInterval="5s"/>
     </sv:SecurityRequirement>
    </sv:requires>
  </rdf:Description>
</rdf:RDF>
```

Priorities

- If two applications have contradicting requirements?
  - Ruleset determines which requirement takes priority

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="..." xmlns:sv="http://www.sevecom.org/schema#">
  <rdf:Description rdf:about="http://www.c2c-cc.org/defaultPriorities">
   <rdf:type rdf:resource="PriorityRules"/>
    <sv:priority rdf:resource="eSafetyApplication" priority="10" />
    <sv:priority rdf:resource="maintenanceApplication" priority="4" />
    <sv:priority rdf:resource="entertainementApplication" priority="1" />
</rdf:RDF>
```

  - Applications can be informed via callbacks, if their security requirements are not met and then decide to proceed or stop operation
- Rulesets may also adapt security mechanisms to national regulations, personal preferences, etc.

Application Callbacks

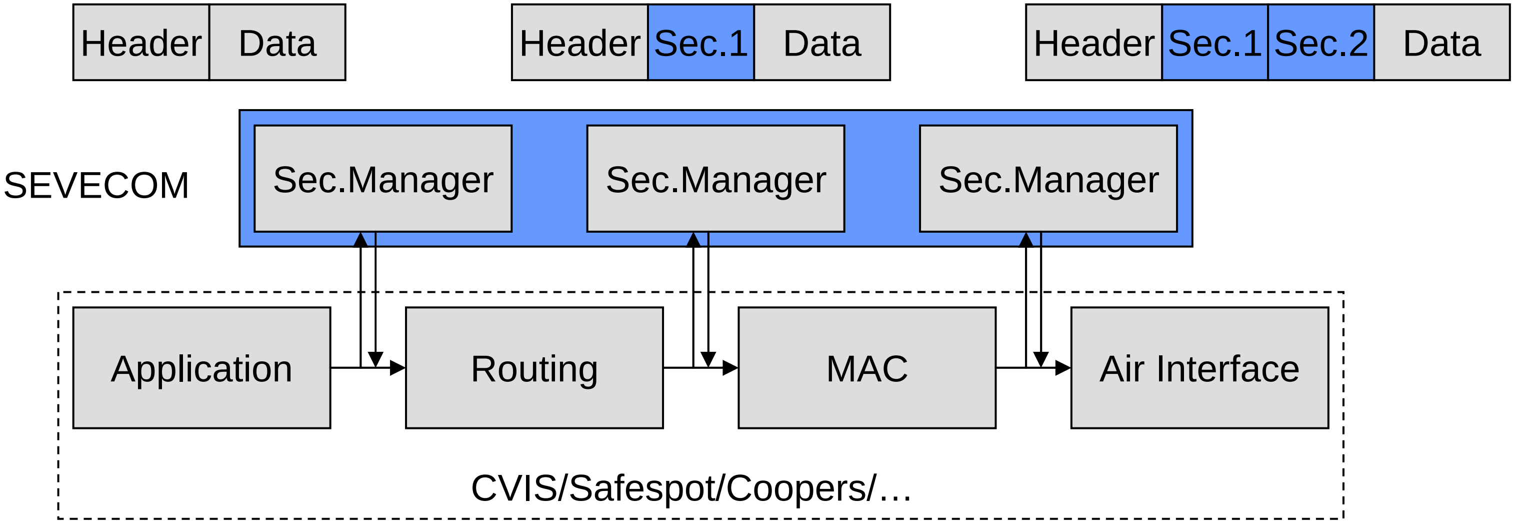
- Security modules can inform applications
  - about results of security operations
    - e.g. transmit user ID after authentication
  - about problems with security operations
    - e.g. when privacy requirements can not be met, because of contradicting requirements in other applications

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="..." xmlns:sv="http://www.sevecom.org/schema#">
  <rdf:Description rdf:about="http://www.c2c-cc.org/vehicle-based_road_cond_warning">
   <rdf:type rdf:resource="esafetyApplication"/>
    <sv:requires>
     <sv:SecurityRequirement module="IdentityAuthentication">
      <sv:InformApplication method="org.sevecom.VehBasRoadCondWarning.authenticated"/>
      …
     </sv:SecurityRequirement>
</sv:requires>
  </rdf:Description>
</rdf:RDF>

package org.sevecom;
public class VehBasRoadCondWarning {
    public void authenticated(Credentials identity) { … }
}
```

Packet Capture and Modification

- How to combine security modules and other functionality?
  - Communication infrastructure allows registration of callbacks at specified hooks, security modules can analyze, modify, and even drop packets at defined hooks
  - Security headers can be attached
  - Similar to Linux netfilter architecture



Open Questions

- Can such a mechanism be integrated into the selected architecture?
- Properties of this architecture?
  - Implemented in Java, C, … ?
  - Access to communication infrastructure
  - Reflection mechanism
  - …