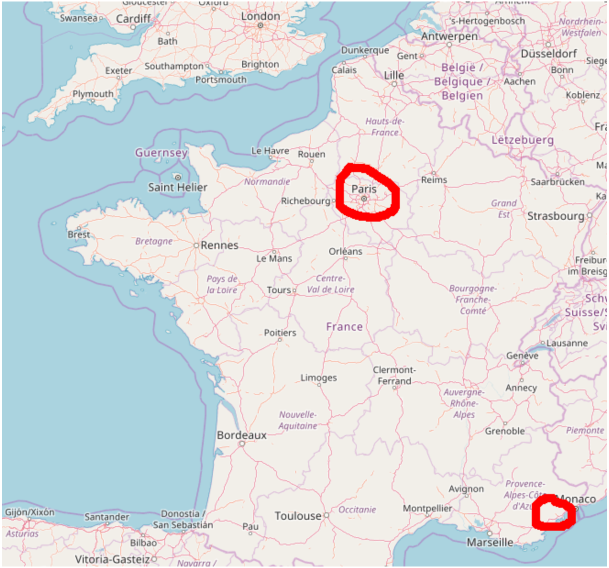
An Overview of Security requirements

Ludovic Apvrille
ludovic.apvrille@telecom-paris.fr

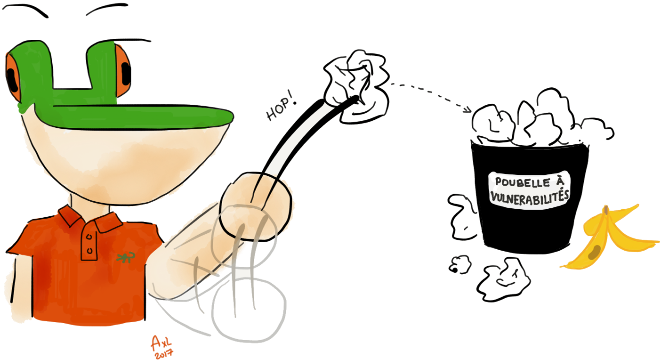
Journées GDR-GPL - groupe Ingéniérie des Exigences

Plan

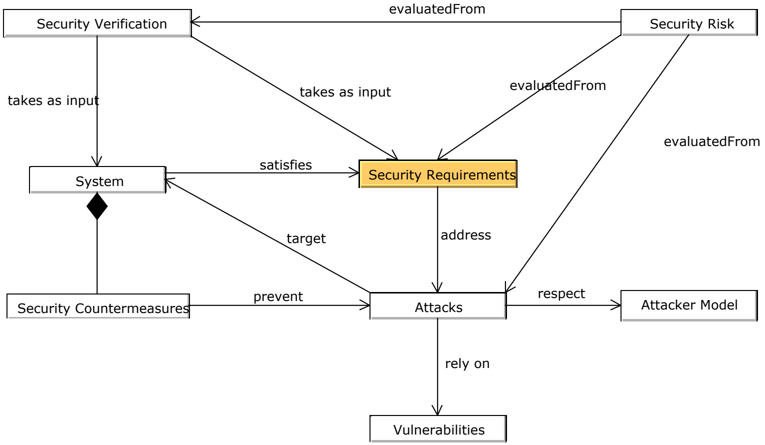
Telecom Paris - LabSoC



Security in Dev. Cycles



Security in Dev. Cycles (Cont.)



Security Requirements

Definitions

- Defined in the scope of FP7 EVITA
- Automotive embedded architecture
  - Include the onboard networks

Hacker vs. attacker

- **Hacker**: smart use of objects / systems
- **Attacker**: criminal / terrorist
  - Financial gain
  - Harm / injury

Confidentiality

*Confidentiality is satisfied when **authorized entities** are the only ones that can know a given **quantum of information***

Example of requirements

- The content of Messages sent from A to B shall be known only by A and B
- The state of a state machine shall be known only by its execution engine

Typical countermeasures

Message ciphering with secret symmetric keys

Privacy

*Privacy is guaranteed if the **relation** between the **entity** and the **set of information** is confidential*

Typical sub-categories: anonymity, unlinkability.

Example of requirements

- In a social network, for non administrator users, the user of a message shall not be linkable to that message but two messages sent by the same user shall be linkable to each other

Typical countermeasures

Data anonymisation, etc.

Integrity

Integrity is satisfied when a ***quantum of information*** *has not been modified between* ***two observations***

(Integrity is also called ”weak authenticity”)

Example of requirements

- The system shall ensure the integrity of messages sent from A to B
- The integrity of the instructions executed on the system processor shall be ensured

Typical countermeasures

Message Authentication Code with secret keys

Data origin authenticity

*Data origin authenticity is satisfied when the **data** (quantum of information) truly originates from the **author***

(authenticity on origin is also called ”strong authenticity”)

Example of requirements

- All information received from sensors by the main controller shall be authentic in terms of origin.

Typical countermeasures

Asymetric cryptography (public / private keys) with certificates provided by trusted authorities

Non-Repudiation

*The non-repudiation of an **action** is guaranteed if it is impossible for the **entity** that performed the action to claim that it did not perform this action*

Example of requirements

- The payment system shall guarantee that neither the payer nor the billing system can deny a transaction once it has been performed

Typical countermeasure

Digital signature (Identity, certificate, MAC, . . .)

Controlled Access (a.k.a. Authorization)

*Controlled access is guaranteed if specified **entities** are the only entities that can perform the **actions** or **access the information***

Example of requirements

- Only explicitly authorized users shall be able to execute processes on the computer
- Controlled access to read data from a hard disk must be ensured

Typical countermeasures

User management (hashes passwords), firewalls.

Freshness

*Freshness is satisfied if a **quantum of information** received by an **entity** at the **given time** is not a copy of the same information received by the same or another entity in the past*

(Usually related to replay attacks)

Example of requirements

- Freshness of all messages sent from A to B must be ensured.
- Execution of instruction in processor P must apply only to fresh instructions

Typical countermeasure

Timestamps / data id with integrity and authencity

Availability

*Availability is satisfied when a **service** or a **physical device** is operational*

(Usually related to Denial of Service Attacks - DoS)

Example of requirements

- The webserver must always respond in less than 1 second to requests
- The availability of the flight management system must be ensured

Typical countermeasure

Firewalls (traffic shaping, bans), redundancy, etc.

And many others . . .

. . . That are often domain-dependent.

E-voting system[1]

- *Eligibility, Uniqueness*: Only eligible voters must be able to vote, and exactly once
- *Individual verifiability*: a voter can check that her/his own vote is correctly counted
- *Universal verifiability*: everyone can verify that all valid votes, and no others, were counted
- . . .

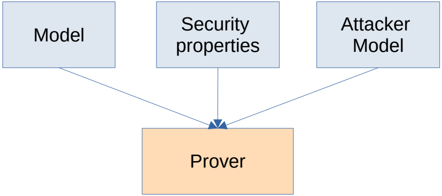
[1] `http://www.cs.cornell.edu/courses/cs513/2002SP/proj.00.StuSolns/mbt91.html`

TELECOM Paris

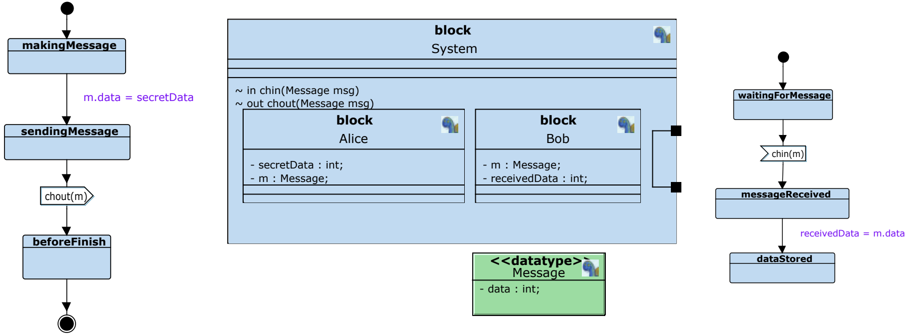
IP PARIS

Method

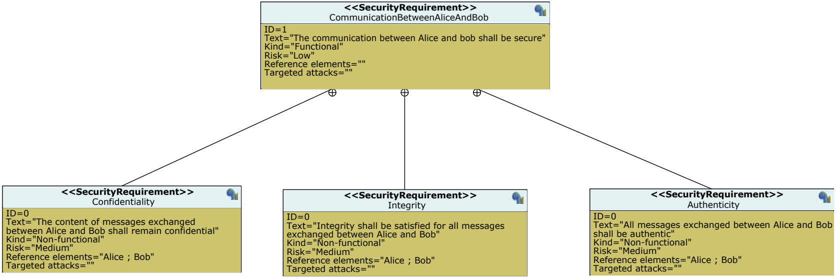
1. Perform a SysML design (from functional requirements)
2. Define Security requirements
3. Relate the security requirements to SysML elements
4. Define an attacker model and attacks
5. Describe security requirements as security properties
6. Perform security verifications
7. Iterate . . .



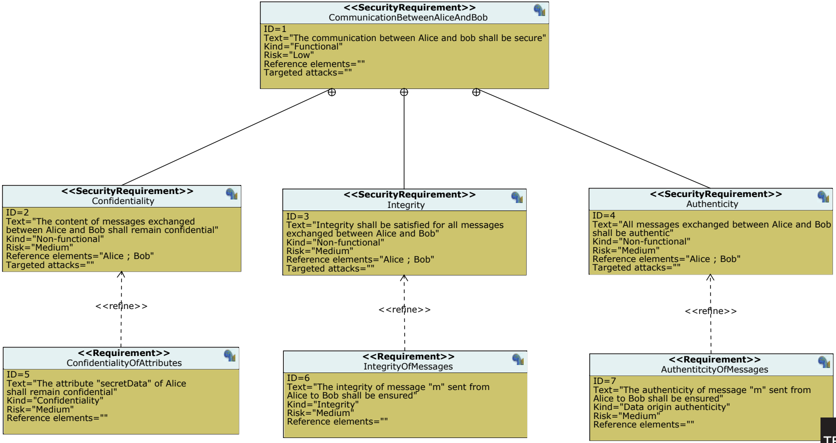
Functional Model



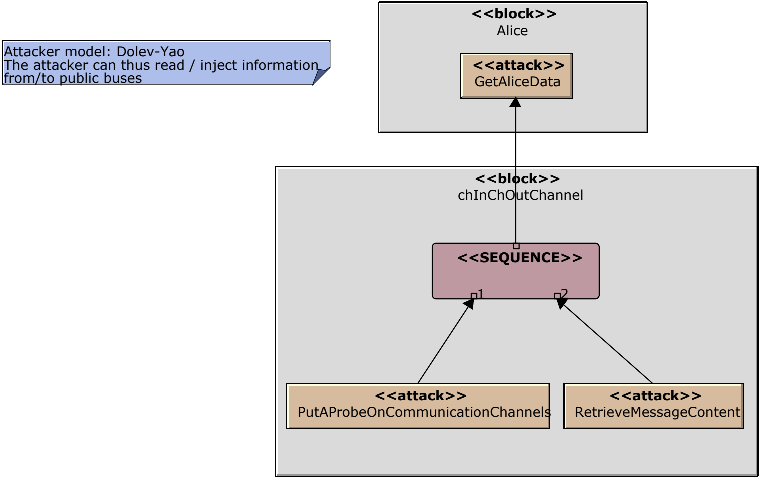
Security Requirements in SysML



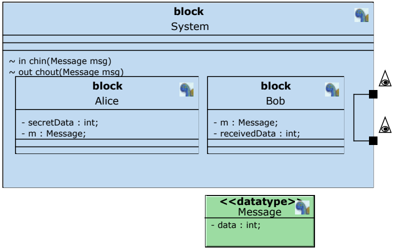
Security Requirements Related to Design

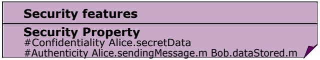


Attacker Model and Attacks

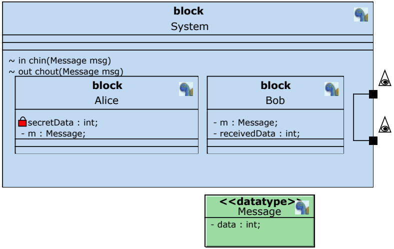


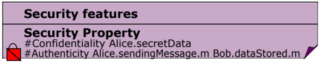
Adding Attacker Model and Security Properties



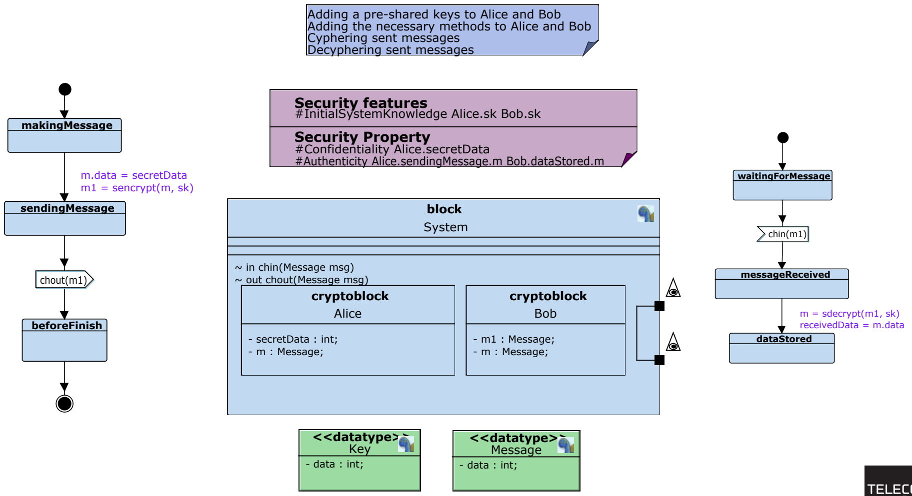


Security Verification (1)

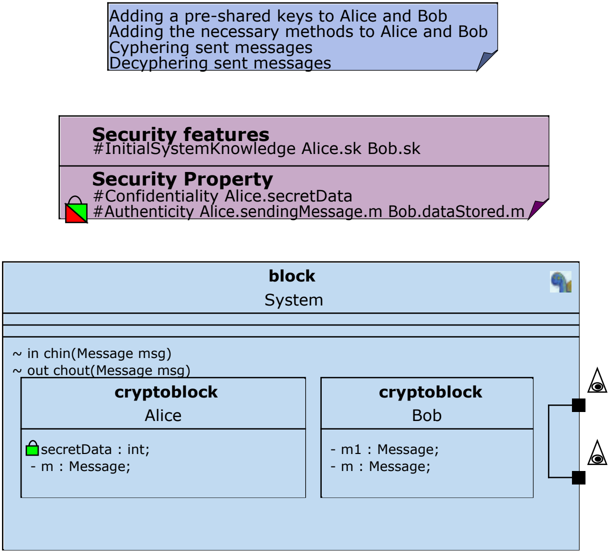




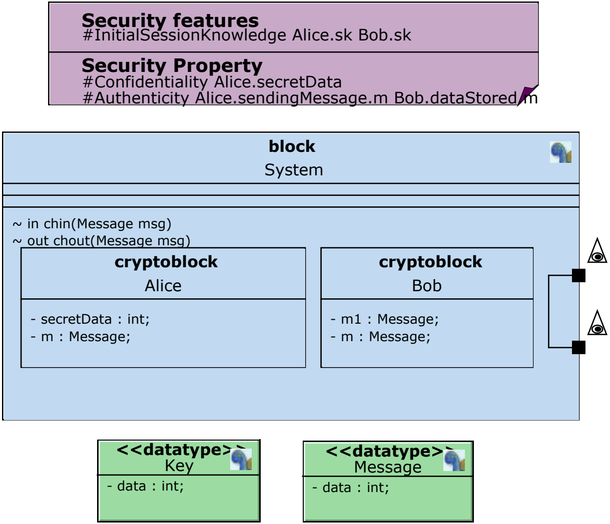
Taking into account Confidentiality



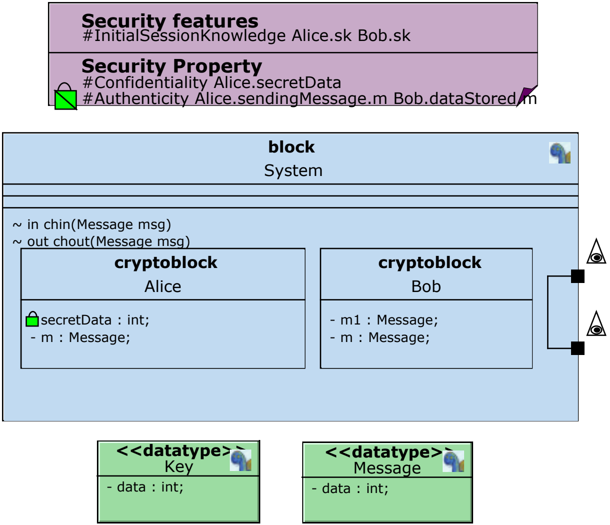
Security Verification (1)



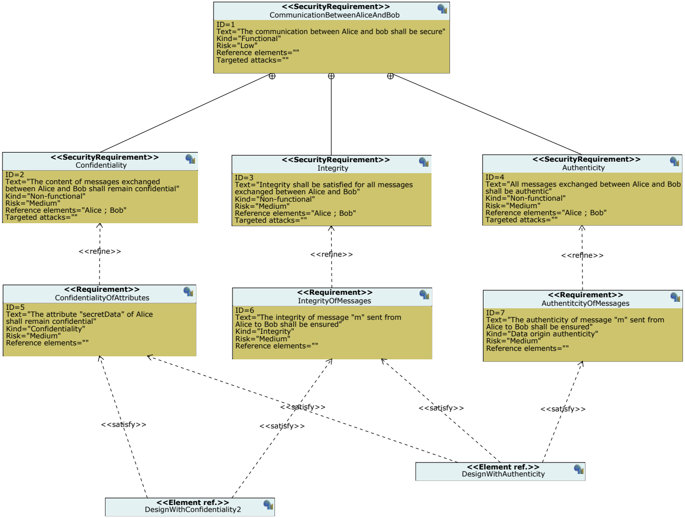
Handling Authenticity



Finally (1/2) . . .


Security features
#InitialSessionKnowledge Alice.sk Bob.sk
Security Property
#Confidentiality Alice.secretData
#Authenticity Alice.sendingMessage.m Bob.dataStored.m
block
System
~ in chin(Message msg)
~ out chout(Message msg)
cryptoblock
Alice
secretData : int;
- m : Message;
cryptoblock
Bob
- m1 : Message;
- m : Message;
<<datatype>>
Key
- data : int;
<<datatype>>
Message
- data : int;

Finally (2/2)



<<SecurityRequirement>>
CommunicationBetweenAliceAndBob
ID=1
Text="The communication between Alice and bob shall be secure"
Kind="Functional"
Risk="Low"
Reference elements=""
Targeted attacks=""
<<SecurityRequirement>>
Confidentiality
ID=2
Text="The content of messages exchanged between Alice and Bob shall remain confidential"
Kind="Non-functional"
Risk="Medium"
Reference elements="Alice ; Bob"
Targeted attacks=""
<<SecurityRequirement>>
Integrity
ID=3
Text="Integrity shall be satisfied for all messages exchanged between Alice and Bob"
Kind="Non-functional"
Risk="Medium"
Reference elements="Alice ; Bob"
Targeted attacks=""
<<SecurityRequirement>>
Authenticity
ID=4
Text="All messages exchanged between Alice and Bob shall be authentic"
Kind="Non-functional"
Risk="Medium"
Reference elements="Alice ; Bob"
Targeted attacks=""
<<refine>>
<<refine>>
<<refine>>
<<Requirement>>
ConfidentialityOfAttributes
ID=5
Text="The attribute "secretData" of Alice shall remain confidential"
Kind="Confidentiality"
Risk="Medium"
Reference elements=""
<<Requirement>>
IntegrityOfMessages
ID=6
Text="The integrity of message "m" sent from Alice to Bob shall be ensured"
Kind="Integrity"
Risk="Medium"
Reference elements=""
<<Requirement>>
AuthentitcityOfMessages
ID=7
Text="The authenticity of message "m" sent from Alice to Bob shall be ensured"
Kind="Data origin authenticity"
Risk="Medium"
Reference elements=""
<<satisfy>>
<<satisfy>>
<<satisfy>>
<<satisfy>>
<<satisfy>>
<<Element ref.>>
DesignWithAuthenticity
<<Element ref.>>
DesignWithConfidentiality2

To Go Further . . .

- **TTool**. http://ttool.telecom-paris.fr
- **SysML-Sec**: http://sysml-sec.telecom-paris.fr

*L. Apvrille, L. W. Li, "**Harmonizing Safety, Security and Performance Requirements in Embedded Systems**", Proceedings of the Design Automation and Test in Europe conference (DATE), March 25-29, Firenze, Italy, 2019*