Smart-M3 security model: research and design

Kirill Yudenok

Open Source & Linux Lab

http://osll.fruct.org

Agenda

- Motivation
- Tasks & Goals
- Top view to Smart-M3 platform
- Smart-M3 security view
- Discretionary model and it’s overview
- Proposed solution and scheme
- Security research and design
- Conclusion
- Next steps

Motivation

What we need

- control access mechanism for the smart space platform, for example Smart-M3;
- mechanism to protect information of the space;
- research information security within smart space area.

Tasks & Goals

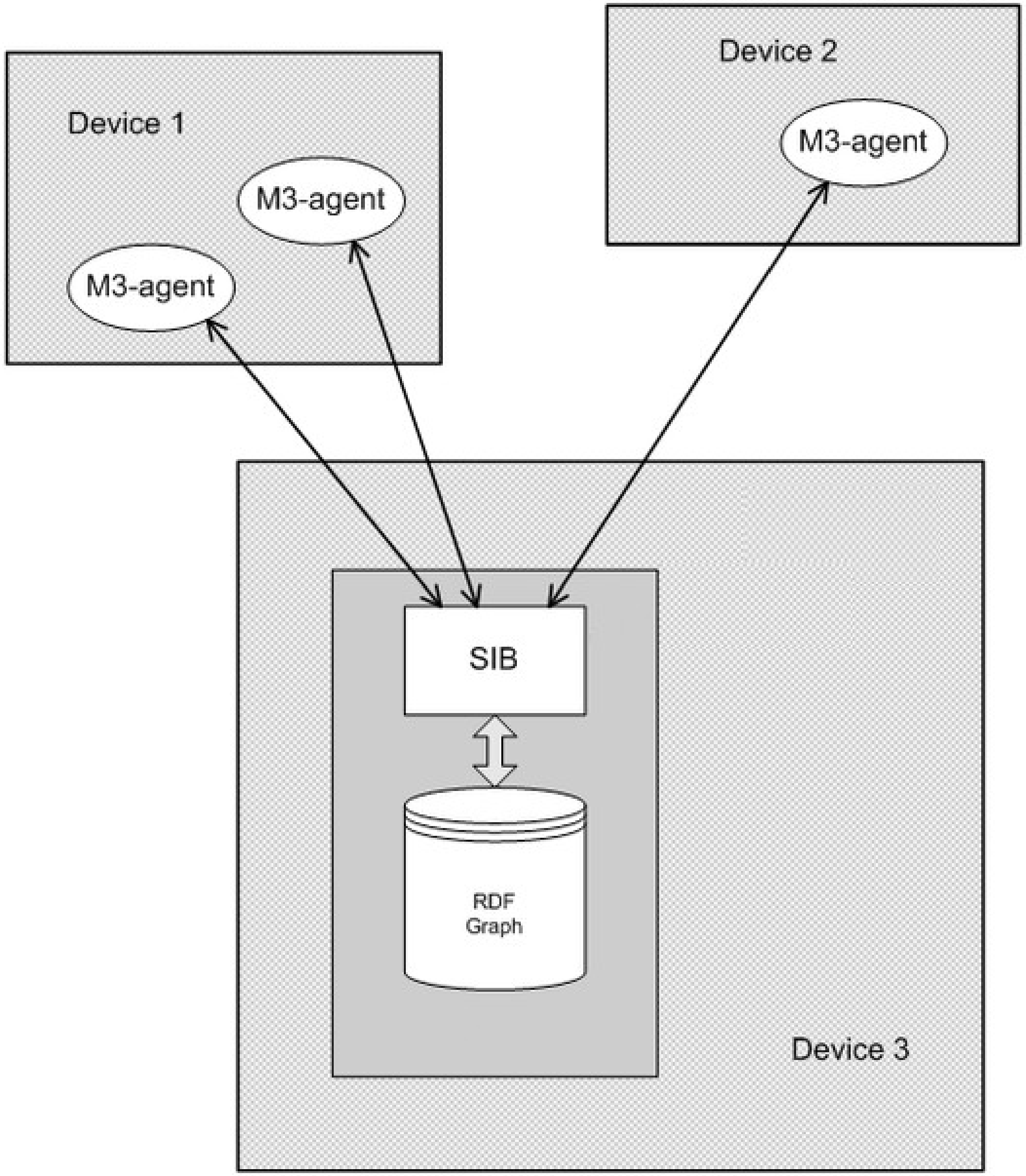
The main goals of project

- to develop a security model for smart spaces;
- design access and control algorithms for one of smart space platform, Smart-M3;
- test the components on the Smart-M3 platform.

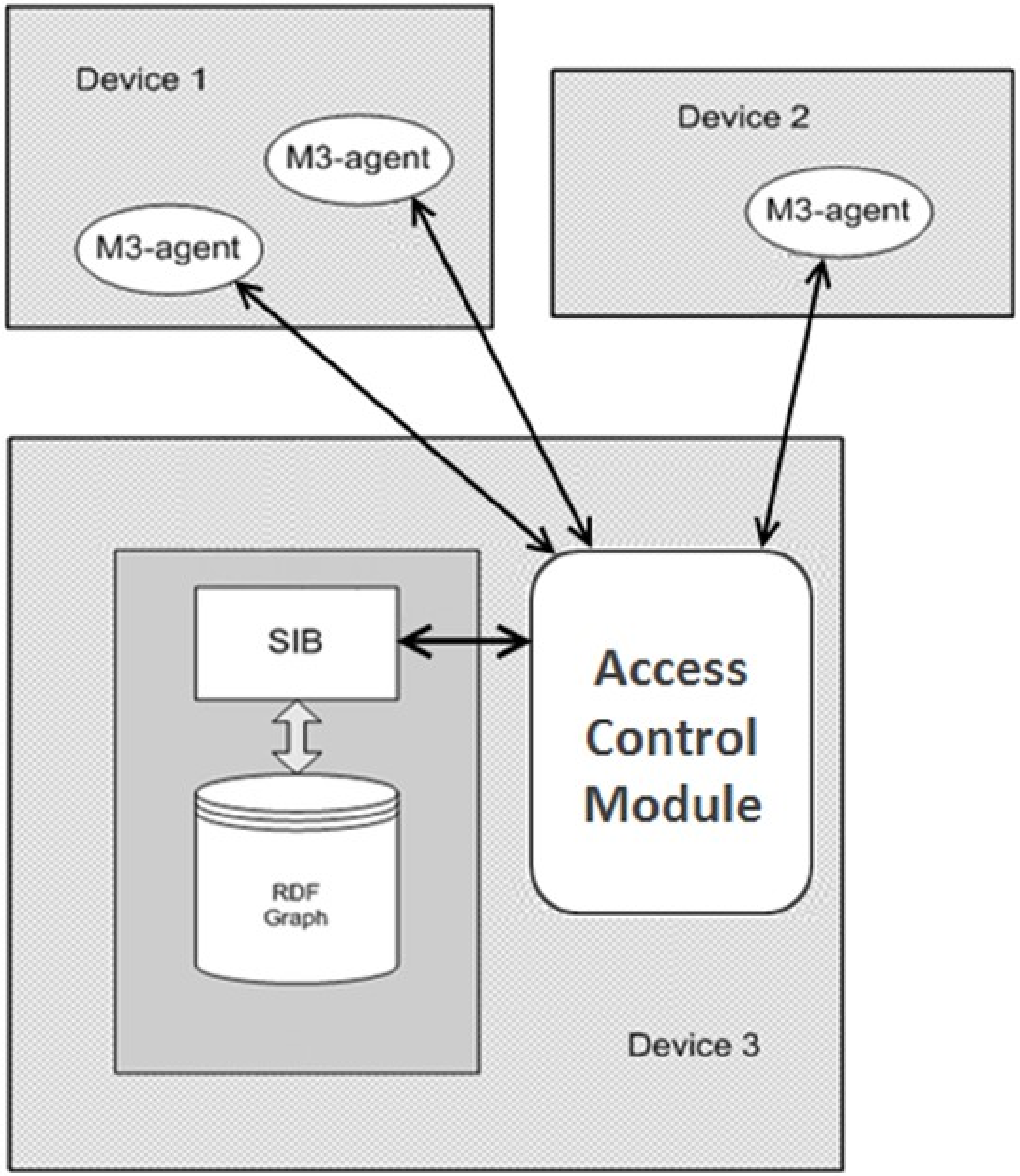
Short term tasks

- research a common security models;
- choose one of the model and describe it within the smart space area;
- provide the expected solution of the model;

Top view to Smart-M3 platform



Smart-M3 security view



Discretionary model

Why

- most widespread in practice;
- simple implementation;
- intuitive and flexible;
- easy of usage and setup.

But

- complexity of administration;
- low-level model;
- the problem of Trojan horses.

Discretionary model overview

The main element of this model is the **access matrix**.

State of protection system is described as a triple:

(S, O, M), where

S – subjects, O– objects and $M[S, O]$ – access rights of the subject (client) to object (space).

- The access rights regulate the management methods of the subject to access objects.
- The basis implementation of the access control is the analysis of the access matrix rows.

Access Matrix

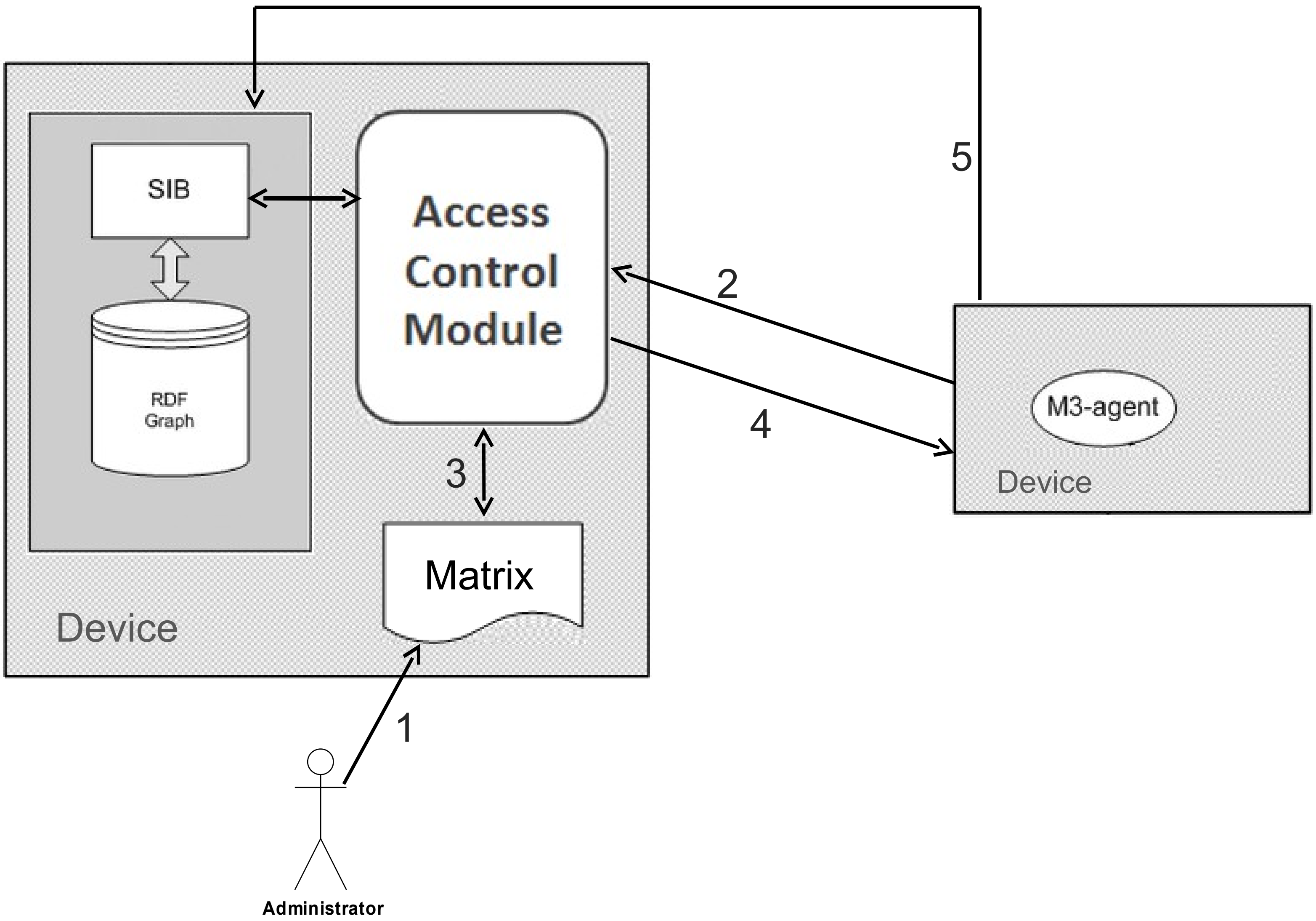
- view protection as a matrix (access matrix);
- rows represent subjects;
- columns represent objects;
- access(*i*, *j*) is the set of operations that a process executing in $Subject_i$ can invoke on $Object_j$

	Object1	**Object2**	**…**	**ObjectN**
Subject1	rw	--	…	rx
Subject2	--	rwx	…	---
…	…	…	…	…
SubjectM	rx	rw	…	rwx

Proposed solution

1. Access matrix configuration, the administrator sets access rights for all prospective clients of the smart space (SIB).
2. Knowledge Processor (KP) sends a connection request to the SIB.
3. The request is sent to a “special module”, that responsible for granting of access rights for KP.
4. Module analyzes the access matrix rows and returns a triplet, containing information with KP access rights to the SIB, if there are none, the connection request is rejected.
5. KP is connected to the SIB with issued rights.

Proposed solution scheme

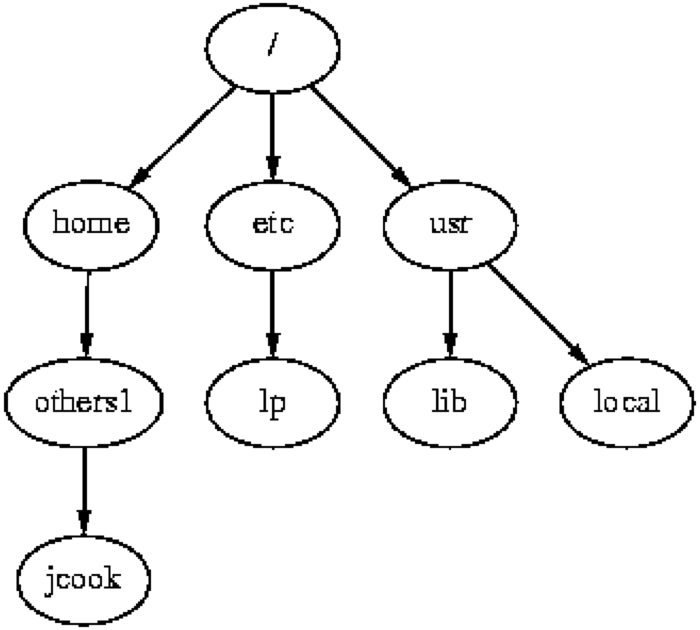


SIB-DB, as a file system!

Consider “SIB-DB” as a file system, that has follow access rights “R, W, X”.

List of control options rights:

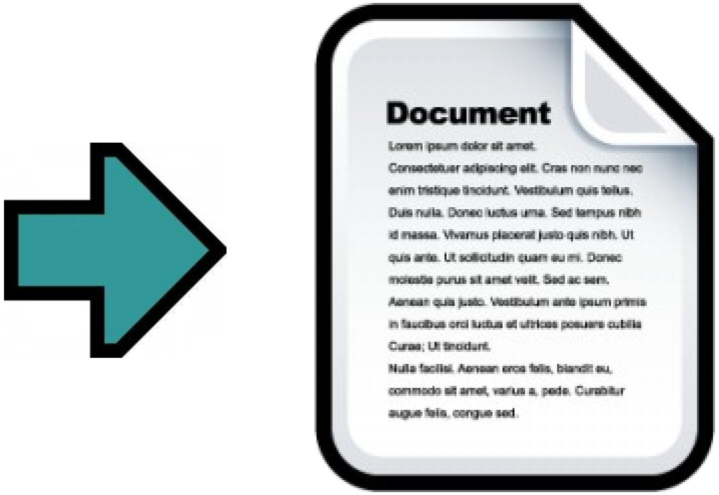
1. Get a list of rights on the connection.
2. Entity subscribes to operations on the connectior



Matrix location

- access matrix should be store near the data on the same device, in metadata form;

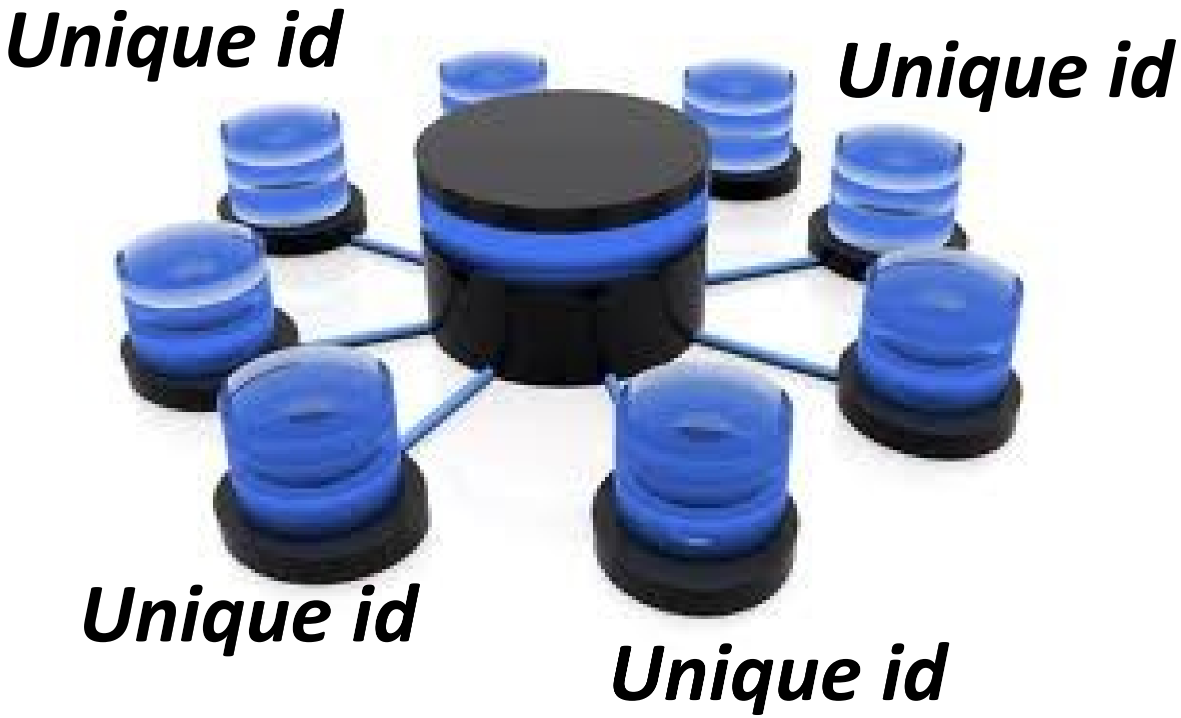
	Object1	…	ObjectN
Subject1	rw	…	rx
…	…	…	…
SubjectM	rx	…	rwx



- A copy of the matrix is stored on each client.

SIB identification

- for easy search SIBs, we can use SIB-registry, which allows you to specify all space SIBs by unique parameters;
- single “access service” for SIBs which controls of access rights to subjects;



Conclusion

Results

- investigated the major issues of model creation;
- described the proposed solution of model work;
- started the process of implementing the model within the Smart-M3 platform;

Next steps and future plans

Next steps

- to develop an access control mechanisms and algorithms for the Smart-M3 platform;
- test developed components on Smart-M3 platform.

Future plans

- design and implement role based model over the discretionary.

Q & A

Kirill Yudenok

kirill.yudenok@gmail.com

Open Source & Linux Lab,

http://osll.fruct.org, osll@fruct.org