Bern University of Applied Sciences

Project 1

Module: BTI 7301

Planning: "IoT Hardware Security Module Proof of Concept"

Students

Noli Manzoni, Sandro Tiago Carlao

Tutor

Dr. **Simon Kramer**

Contents

1 Purpose of the document

This document describes the planning of the project "IoT Hardware Security Module Proof of Concept". This document is written with LaTeX[1].

2 Planning

2.1 Work Packages

In this section are described all the planned tasks. Our work load per week is about 16 to 32 hours (8-16 hours pro person).

Scope

Determine project scope: Understand the project description (see attachments) and determine the project objectives. The project scope will be further clarified and finalized in the meeting with Dr. Simon Kramer.

State of the art: Analyze and search for new information to be sure about the state of the art.

IoT Devices: Search and compare information about the different IoT devices available to decide which device is more suitable.

IoT Security: Search and collect information about the security of IoT devices to improve the final product.

Connection: After that the IoT device is chosen an appropriate cable must be selected.

Analysis/Software Requirements

Project objectives: Document the various objectives discovered in the "Determine project scope" task.

System boundaries: Analyze the system to find the system boundary, the context boundary and the system context.

Requirements: Decide, after that the "IoT devices" research is done, which functionality can be completed in the given time with the chosen IoT device.

Design Design the software by following the project requirements. Choose the programming language to implement the connection between the devices.

Development Develop the software by following the design using iterative and incremental development.

Testing Test all implemented functionalities.

Documentation Document all the tasks, the problems and the choices that have been made.

2.2 Milestones

Meeting with Simon Kramer: Project scope clarified and start for the research. *Duration: two hours - Location: Biel/Bienne Rolex building*

Meeting with Peter Affolter: Discussion about the different IoT devices and proof of found information ("IoT Devices" task). *Duration: two hours - Location: Vauffelin Dynamic Test Center*

Device choice: Choice of the most suitable IoT device for the project.

Meeting with Gehard Hassenstein: Discussion about the IoT devices security and proof of found information ("IoT Security" task). *Duration: two hours - Location: Biel/Bienne Rolex building*

Connection choice: Choose the more suitable connection to transfer data between the two devices.

Requirement revision: Revision of the requirement. Start of the design phase.

Connection choice: Choice of the most suitable connection based on the requirements. Start of the development phase.

Code review: Code review with the expert to evaluate the code. *Duration: two hours - Location: Biel/Bienne Rolex building*

Project delivery: Delivery of the project (code and documentation) to be evaluated.

2.3 Development

2.3.1 Product backlog

ID	Work package	Estimation
1	Project structure initialization	8 hours
2	Keyczar infrastructure initialization	8 hours
3	Keyczar AddKey command	4 hours
4	Keyczar Decrypt command	4 hours
5	Keyczar Encrypt command	4 hours
6	Keyczar Sign command	4 hours
7	Keyczar Verify command	4 hours
8	Keyczar Create command	4 hours
9	Keyczar Revoke command	4 hours
10	Keyczar Demote command	4 hours
11	Keyczar Promote command	4 hours
12	Keyczar PubKey command	4 hours
13	Keyczar Delete Key Set command	4 hours
14	Serial Connection integration	16 hours
15	Login implementation	8 hours
16	Serial Connection AddKey command	4 hours
17	Serial Connection Encrypt command	4 hours
18	Serial Connection Decrypt command	4 hours
19	Serial Connection Sign command	4 hours
20	Serial Connection Verify command	4 hours
21	Serial Connection Create command	4 hours
22	Serial Connection Revoke command	4 hours
23	Serial Connection Demote command	4 hours
24	Serial Connection Promote command	4 hours
25	Serial Connection PubKey command	4 hours
26	Serial Connection Delete Key Set command	4 hours
27	Graphical interface	16 hours
28	Secure the OS	8 hours
29	Command line application	8 hours
30	Installer	8 hours
31	Documentation	30 hours

2.3.2 Sprint 1 03.04.2017 - 23.04.2017

ID	Workload	Remark
1 - Project structure initialization	8 hours	Problems with the OS configuration
2 - Keyczar infrastructure initialization	8 hours	Problems with the library installation
3 - Keyczar AddKey command	4 hours	
4 - Keyczar Decrypt command	4 hours	
5 - Keyczar Encrypt command	4 hours	
6 - Keyczar Sign command	4 hours	
7 - Keyczar Verify command	4 hours	
8 - Keyczar Create command	4 hours	
9 - Keyczar Revoke command	4 hours	
10 - Keyczar Demote command	4 hours	
11 - Keyczar Promote command	4 hours	
12 - Keyczar PubKey command	4 hours	
31 - Documentation	10 hours	Diagrams included
Total:	**66 hours**	

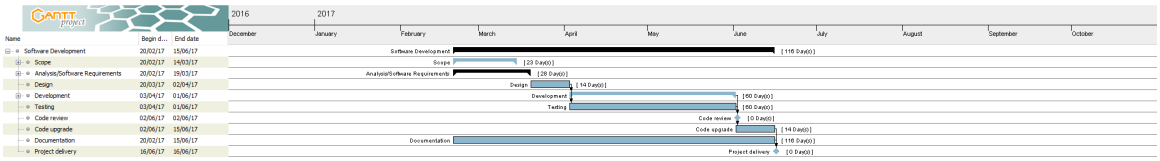
2.3.3 Sprint 2 24.04.2017 - 14.05.2017

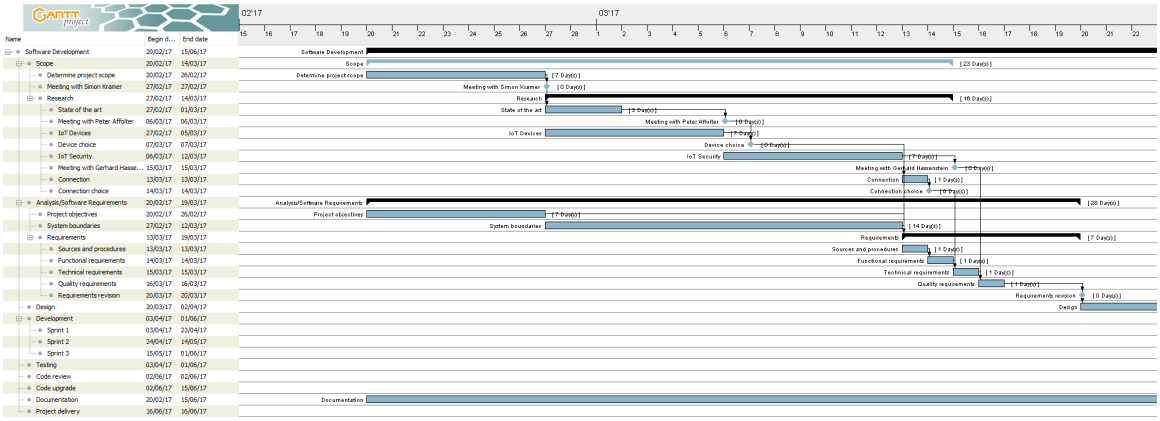
ID	Workload	Remark
13 - Keyczar Delete Key Set command	4 hours	
14 - Serial Connection integration	16 hours	Problems with the serial connection
16 - Serial Connection AddKey command	4 hours	Problems with the synchronization
17 - Serial Connection Encrypt command	4 hours	Problems with the file transfer
18 - Serial Connection Decrypt command	4 hours	
19 - Serial Connection Sign command	4 hours	
20 - Serial Connection Verify command	4 hours	
21 - Serial Connection Create command	4 hours	
22 - Serial Connection Revoke command	4 hours	
23 - Serial Connection Demote command	4 hours	
24 - Serial Connection Promote command	4 hours	
31 - Documentation	10 hours	Diagrams included
Total:	**66 hours**	

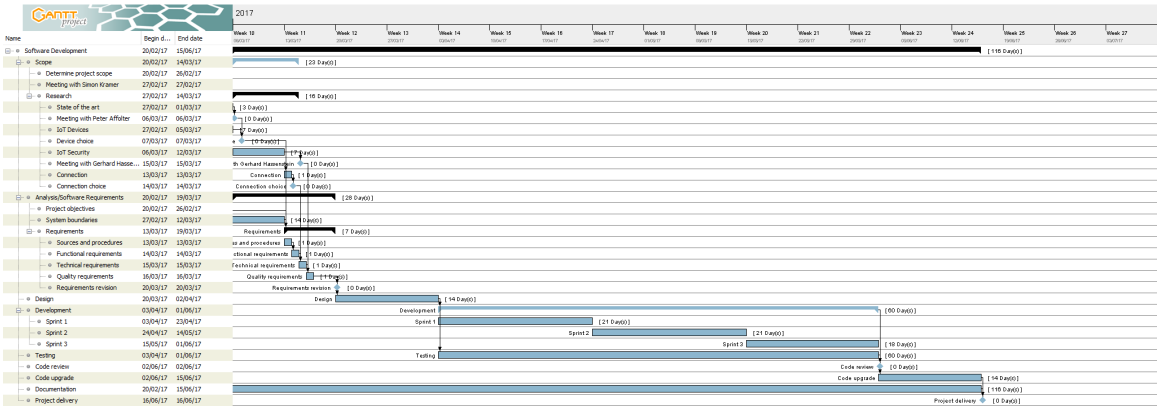
2.3.4 Sprint 3 15.05.2017 - 01.06.2017

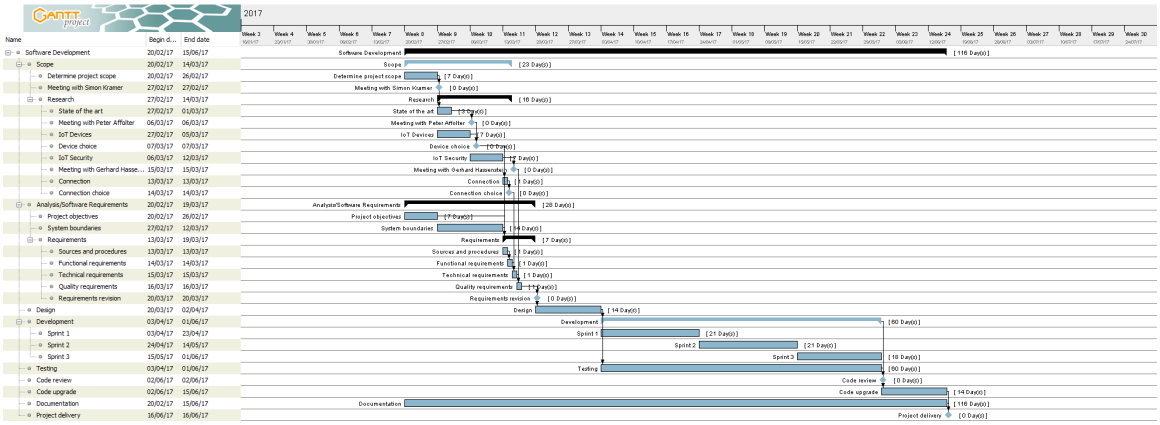
ID	Workload	Remark
14 - Log in implementation	8 hours	Problems with the log in library
25 - Serial Connection PubKey command	4 hours	
26 - Serial Connection Delete Key Set command	4 hours	
27 - Graphical interface	16 hours	
28 - Secure the OS	8 hours	
29 - Command line application	8 hours	
30 - Installer	8 hours	
31 - Documentation	10 hours	Diagrams included
Total:	**66 hours**	

3 Gantt chart









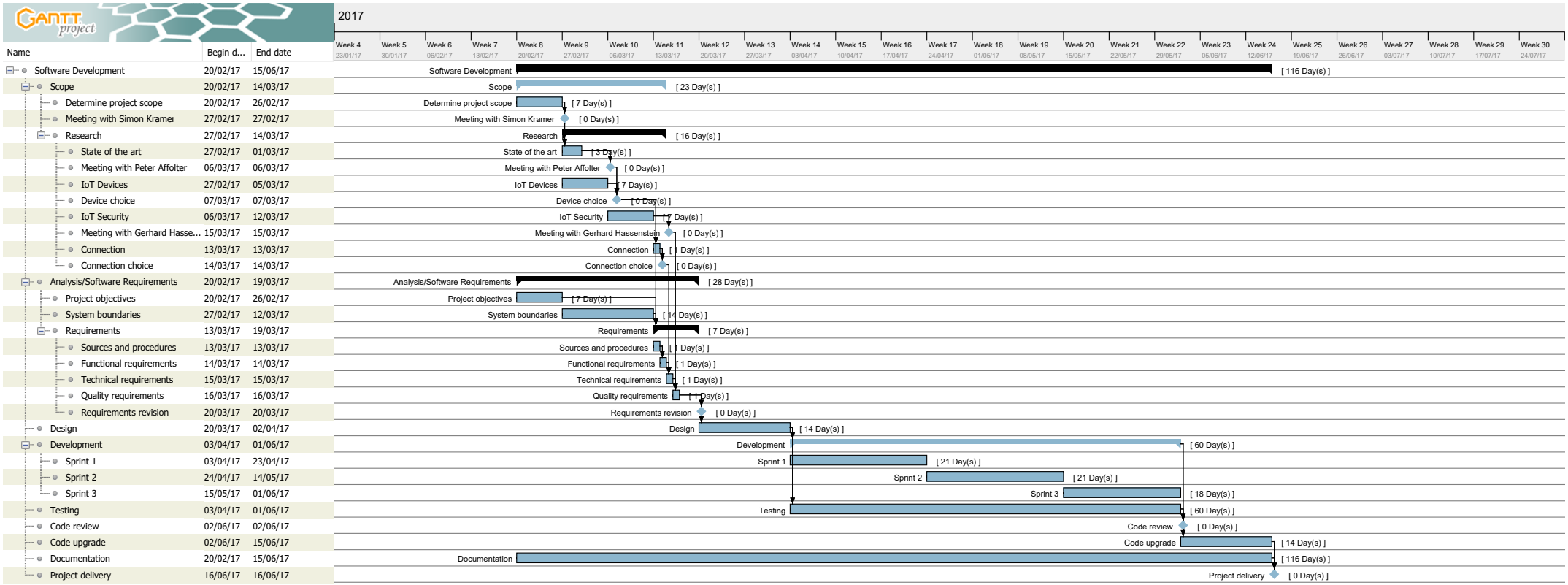
4 Attachments

- "IoT Hardware Security Module Proof of Concept" project description
- Complete Gantt chart

References

[1] *LaTeXis a typesetting system designed for the production of technical and scientific documentation*
`https://www.latex-project.org/`

Gantt Chart



Name	Begin d...	End date
Software Development	20/02/17	15/06/17
Scope	20/02/17	14/03/17
Determine project scope	20/02/17	26/02/17
Meeting with Simon Kramer	27/02/17	27/02/17
Research	27/02/17	14/03/17
State of the art	27/02/17	01/03/17
Meeting with Peter Affolter	06/03/17	06/03/17
IoT Devices	27/02/17	05/03/17
Device choice	07/03/17	07/03/17
IoT Security	06/03/17	12/03/17
Meeting with Gerhard Hasse...	15/03/17	15/03/17
Connection	13/03/17	13/03/17
Connection choice	14/03/17	14/03/17
Analysis/Software Requirements	20/02/17	19/03/17
Project objectives	20/02/17	26/02/17
System boundaries	27/02/17	12/03/17
Requirements	13/03/17	19/03/17
Sources and procedures	13/03/17	13/03/17
Functional requirements	14/03/17	14/03/17
Technical requirements	15/03/17	15/03/17
Quality requirements	16/03/17	16/03/17
Requirements revision	20/03/17	20/03/17
Design	20/03/17	02/04/17
Development	03/04/17	01/06/17
Sprint 1	03/04/17	23/04/17
Sprint 2	24/04/17	14/05/17
Sprint 3	15/05/17	01/06/17
Testing	03/04/17	01/06/17
Code review	02/06/17	02/06/17
Code upgrade	02/06/17	15/06/17
Documentation	20/02/17	15/06/17
Project delivery	16/06/17	16/06/17

15. IoT Hardware Security Module Proof of Concept

Titel	
Beschreibung	The goal of this project is to **find out to which extent off-the-shelf Internet of Things (IoT) modules (like Arduino, Raspberry) can be programmed to function as Hardware Security Modules (HSM).** A HSM is a **cryptographic device** for **secure** - **private and symmetric key generation and storage (protection from key compromise)** - **algorithm execution (protection from side-channel attacks).** You will build on such **existing proofs of concept,** improving and, if necessary, migrating them to **Java,** ideally obtaining your own functional HSM for your own personal use as a result of your Project-1 work.
Gruppengrösse	3-6 (6: split into 2 groups)
Anzuwendende Technologie	Arduino or Raspberry, or both, IoT, Java
Links	Arduino HSM proofs of concept: https://github.com/st3fan/arduino-aws-hsm https://randomoracle.wordpress.com/2013/01/15/arduino-tpms-and-smart-cards-redefining-hardware-security-module/ Raspberry HSM proof of concept: https://cryptosense.com/building-a-raspberry-pi-hsm-for-rsa-2014/
Betreuer	KMS4 (Stakeholder: Peter Affolter, Gerhard Hassenstein)