23/11/09

# ERC Project: R&D team’s roadmap for 2009-2010

## Introduction:

The ERC project started on the 1st of October 2009 and will last 5 years.

From the R&D’s point of view the aim of the project is to design and implement a wide and thorough platform of cryptographic tools. The purpose of this platform is twofold. On the one hand it should allow cryptography researchers to quickly implement, run and analyze performance of existing protocols. On the other hand it should provide an easy way to quickly devise and test possibly new protocols.

## R&D team’s general work plan:

The project will be divided into two main parts. The first part will concentrate its efforts in designing and implementing cryptographic tools or SDK. The second part will be geared towards developing high level protocols. These will be both known protocols as well as new ones developed by the research team.

A general timeline is suggested here:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Sub-task** | **Starting time** | **Finishing time** | **Overall duration** |
| Overall ERC project | Overall ERC project | 01/11/09 | 30/09/14 | 5 years |
| First part | Analysis | 01/11/09 | 29/04/10 | 6 months |
|  | Design of SDK | 02/05/10 | 01/11/10 | 5 months |
|  | Implementation of SDK | 02/11/10 | 04/10/11 | 11 months |
|  | Testing of SDK | 05/09/10 | 15/09/11 | Ongoing throughout the implementation stage |
| Second part | Design of isolated, known protocols | TBD | TBD |  |
|  | Implementation of isolated, known protocols | TBD | TBD |  |
|  | Testing of isolated, known protocols | TBD | TBD | Ongoing throughout the implementation stage |
|  | Design of application protocols, new protocols by our research team | TBD | TBD |  |
|  | Implementation of application protocols, new protocols by our research team | TBD | TBD |  |
|  | Testing of application protocols, new protocols by our research team | TBD | TBD |  |

#### High level description of the first part of the project

The first part of the project will last approximately two years.

Analysis stage: The first two quarters will be dedicated by the R&D team to learn as widely as possible cryptographic concepts as well as network communication issues. The cryptographic concepts will include crypto primitives, encryption schemes and protocols. The communication concepts will focus on understanding of TCP/IP protocols, point-to-point authenticated and secure channels, broadcasting channels.

Design stage: During the second two quarters the R&D team will write a thorough design of the SDK. This design will take into account current known components as well as it should allow the addition of unknown (at the time of design) components. (Robustness). The design will contain a detailed specification of the requirements of the SDK as well as a detailed description of all the needed classes, interfaces and their inter-relations. UML will be used to write the design.

Implementation and testing stage: The second year will be dedicated to implement the SDK as it was designed during the design stage. Throughout the implementation there will be regular testing “stops”. The testing will be performed at unit levels as well as at higher levels. For example, once a few basic primitives are implemented we will want to see how easy and simple it is to implement a higher level protocol that uses these primitives is. This will allow us to see and control that the design actually “works”, that is that it allows the usage of the primitives by higher level protocols that not necessarily were thought off at the time of the design.

#### Detailed description of the first part of the project

Analysis stage:

The main aim of this stage is to achieve a high level of understanding of the crypto domain/world in order to correctly understand the requirements of the project and be able to write a very detailed design in the second stage.

A secondary aim of this stage is to master design techniques that will be applied in the next stage when writing the design of the SDK.

Next to each item we specify which member of the development team is assigned the task, Y for Yael, M for Meital.

##### Crypto knowledge will be achieved by:

1. Learning from book: Introduction to modern cryptography (By Katz-Lindell). (Y, M)
2. Learning from book: Network security, private communication in a public world (By Kaufman-Perlman-Speciner). (Y, M)
3. Learning communication protocols specified below. (M)

Communication’s knowledge:

Meital will be assigned the task of learning the following protocols and preparing a few lessons about them that will be presented to the whole project team. This part should take about four weeks of full time research of existing protocols as well as implementations of these protocols.

1. Secure broadcasts (Byzantine generals).
2. Weak broadcast.
3. Broadcast with abort (non-unanimous).
4. Authenticated broadcast.
5. Broadcast with expected-constant rounds.
   1. Regular
   2. Weak
   3. Authenticated
6. Simultaneous broadcast (n to n). (This protocol may not exist yet).
7. Synchrony:
   1. In fully asynchronous network
   2. In partially synchronous network
   3. With clocks.

A good starting point will be the book “Distributed computing, fundamentals, simulations and advanced topics” by Attiya-Welch.

Other sources are:

<http://www.cs.technion.ac.il/Labs/dsl/projects/fts/main.html>

<http://marknelson.us/2007/07/23/byzantine/>

<http://dsl.cs.technion.ac.il/projects/Ensemble/overview.html>

<http://www.cs.cornell.edu/Info/Projects/HORUS/Overview.html>

<https://tao.truststc.org/Members/hweather/advanced_storage/Public%20resources/replication_fault/replication/download>

<http://www.cs.cornell.edu/projects/quicksilver/>

##### Software engineering knowledge will be achieved by:

1. Learning how to use UML (Unified modeling language). UML will be used to write the design of the SDK. (Y,M)
2. Learning how to use RSM. RSM is an IBM’s platform that we will use to write the design as well as an IDE for developing with Java. (It is also possible to develop C++ code with it). (Y, M)
3. Learning from book: ”Design Patterns, Elements of reusable object-oriented software” (By Gamma-Helm-Johnson-Vlissides). Design patterns are a very useful tool when writing the design of any project since the essence of many objects and their relations can be better seen and achieved through this tool. (Y,M)
4. Decide which programming language will be used for the development of the SDK. (Y ) (Done)
5. We chose the Java language to develop the SDK; therefore, Meital should learn Java programming. (M)
6. Secure coding. (Y, M)

We will look into the following resources:

1. <http://java.sun.com/security/seccodeguide.html>
2. <https://www.securecoding.cert.org/confluence/display/java/The+CERT+Sun+Microsystems+Secure+Coding+Standard+for+Java>
3. Book: Secure Coding in C and C++ by *Robert C. Seacord,* even though we will program in JAVA there might be important concepts found in this book that still apply to JAVA programming.
4. Learn what crypto primitives are already available. Look in:
   1. Java Sun Security package
   2. Java Cryptography Architecture (JCA) and Java Cryptography Extension (JCE)
   3. Bouncy Castle
   4. OAEP
   5. PKCS V1.5
   6. Cramer Shoup?
   7. CACE

Design Stage:

Follows a Gantt chart: