Master's Thesis Proposal

Model-Based Testing of Network Security Protocols in Java Card Applications

Author:
Richard Ssekibuule
Student Number: 0440752

Supervisors:

Dr. Vlad Rusu

Dr. Martijn Oostdijk

Dr. Jan Tretmans

Contents

1 Introduction

Development of a reliable software system has to be approached in a systematic way and requires use of appropriate tools and mechanisms to ensure a high level of quality for the system under development. Testing is an important phase in software development life cycle that has to be managed very well. A detailed explanation and usefulness of structured testing is outlined by Pol et al.[1] in a Test Management approach(TMap).

In this research, we investigate techniques for automatic testing of a security sensitive application, in particular a Java Card application implementing Kerberos network authentication protocol[2]. Our plan of work is to perform software tests using test cases derived from models that describe functional and security aspects of the systems. The research project will evaluate challenges of automatic model-based testing for a practical security application.

The frame work for model-based testing is appealing to our research, because it facilitates automation of the testing process as discussed in the papers[3, 4] and thus is seen as a fore-runner for testing techniques that can be easily integrated in the software development life cycle.

1.1 System Under Test

The system that is to be investigated is an Electronic Banking (E-Banking) application, which uses Kerberos network authentication protocol to provide a secure environment for communication between the Java card client, who is typically an individual with a mobile phone containing a smart card running a Java card application and the banking system hosted at the bank. The application uses smart card to enhance security for a Java 2 Micro Edition (J2ME)[5] application.

The application can be used on a cell phone to allow its users to access their bank accounts and perform any supported transactions. Essentially, if the transaction is not an inquiry one (For example, asking for bank balance) then, it would have to result into a debit or credit of the customer's account. The Java Card application on the Java Card would contain the authentication logic that ascertains whoever is trying to credit or debit an account.

The communication between the banking application (server) and the client on the customer's cell phone is secured by an implementation based on Kerberos[6, 7]. All information exchange is encrypted by a secret key installed on the customer's smart card. The secret key would comprise of the user's pin code which is chosen at the time of installation and an E-banking key which is known only to the bank.

The Java Card technology has to ensure that the E-bank's key is never exposed to any client application accessing the Java Card application.

1.2 Problem Definition/Research Goal

The research aims to explore and investigate challenges in the use of formal methods for automatic validation and conformance testing of security protocols in a Secure Java Card Application. Why should we perform validation and conformance testing? Almost every average user of information systems would agree with the fact that a large percentage of software is implemented with errors. Worse still, security applications cannot provide a guarantee to the confidentiality and integrity of information if the implementations are not correct. The research project explores techniques for validating an implementation of a security protocol and specifying it formally so that conformance testing can be performed. Model-based testing has many advantages, but the most important one for this research is automation of the testing process for software systems[4]. Model-based testing has been successful for many types of tests, but little has been said about model-based testing for security properties of an application. The research project aims to validate correctness of the protocol implementation and also perform conformance testing for the implemented Kerberos network security protocol in relation with the standard specification. The research project intends to achieve the objectives listed below:

i. Model implementation of Kerberos in the system using high level protocol specification language[8] and then perform validation using AVISPA tool[9]. Figure 1 below shows the sequence of steps to be taken. Methods implementing the network authentication protocol will be annotated with JML[10] conditions inspired (or translated) from a HLPSL model of Kerberos.

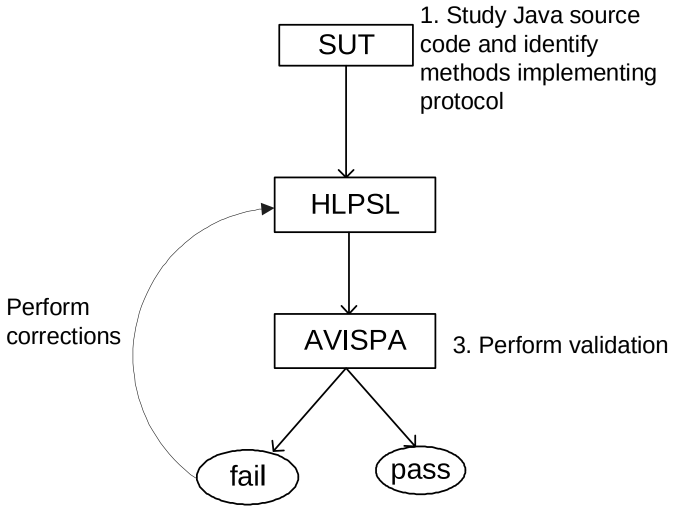

SUT
1. Study Java source code and identify methods implementing protocol
HLPSL
Perform corrections
AVISPA
3. Perform validation
fail
pass

Figure 1: *Protocol Validation*

ii. Develop a formal specifications using either LOTOS[11] or PROMELA[12] for the implementation and the standard Kerberos specification using LOTOS and test for conformance of the implementation to the specification using TORX[4]. Figure 2 below shows the sequence of steps to be taken.

2 Methodology

The project has been divided into phases that are described below.

2.1 Preparation and Planning

This stage will mainly require reading the research papers relevant to the project, planning on how the information will be used in the project and preparing the environment in which the project will take place. The out come of this phase is the project plan in this document and development environment for the E-banking application.

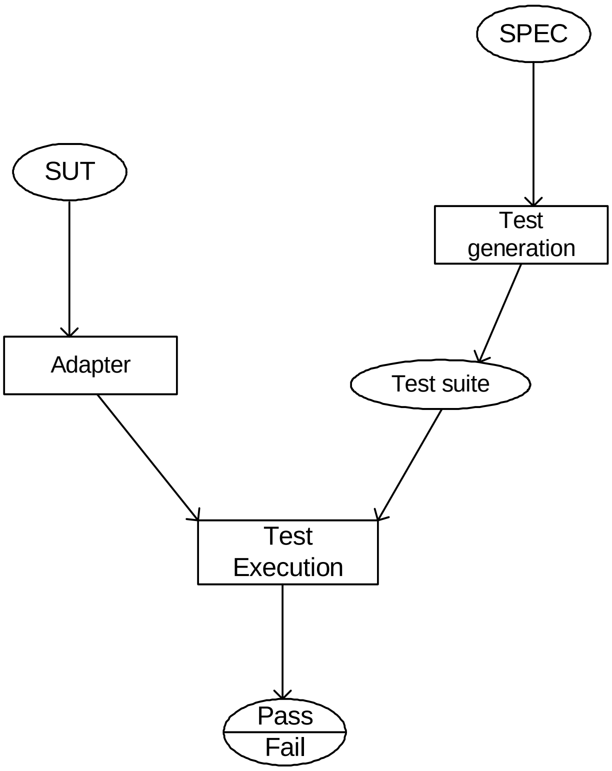


Figure 2: *Conformance Testing*

2.2 Research Phase

This phase will provide answers to challenges presented in the proposal and any questions whose answers may not be known. Research will be continuous process until the development phase is finished. The results of this phase will appear in the final thesis report.

2.3 Development Phase

The development phase will be carried out in three phases. The first phase will involve modeling the network authentication protocol in the E-Banking Java Card application in HLPSL in order to perform validation of security protocol using the Avispa tool. The second phase will involve developing a formal specification of the system under test. The formal specification will be developed in either LOTOS or PROMELA. The third phase will constitute development of the ADAPTER, which is the application specification component to provide connection with the system under test(SUT).

2.4 System Analysis

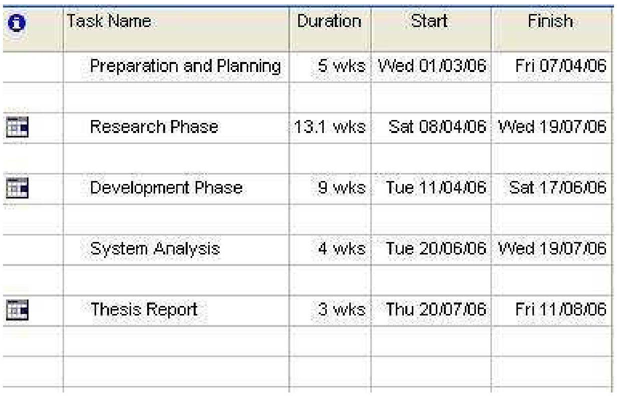
The first part of the analysis phase will go hand in hand with the development phase to refine models of the system. After the development phase is completed, the analysis phase will be carried on to perform higher level analysis of the system for conformance and correctness. The output of this phase will appear in the progress reports and final report.

2.5 Finalizing Thesis Report

The final part of the project will be concluded with a report, part of which would have been written in the earlier stages.

2.6 Schedule

The figure 3 below shows an out line of the schedule that will be followed and the corresponding Gantt chart is presented in the appendix.



ℹ	Task Name	Duration	Start	Finish
	Preparation and Planning	5 wks	Wed 01/03/06	Fri 07/04/06
▦	Research Phase	13.1 wks	Sat 08/04/06	Wed 19/07/06
▦	Development Phase	9 wks	Tue 11/04/06	Sat 17/06/06
	System Analysis	4 wks	Tue 20/06/06	Wed 19/07/06
▦	Thesis Report	3 wks	Thu 20/07/06	Fri 11/08/06

Figure 3: *Schedule*

3 Discussion

In this section we discuss the potential impact of the research.
Firstly, this research will help us learn details about development of smart card applications using Java Card Technology and the use of Kerberos network authentication protocol in a Java Card application.
The research work will provide an opportunity of learning how to incorporate formal methods in the software development life cycle to facilitate automation of the testing process for a security sensitive application. Furthermore, techniques for analyzing security protocol in a practical application will be studied and presented.
After the research project, we expect to have gained an experience in the process of verifying correct implementations of network protocols in Smart Card applications. Knowledge of model-based testing for conformance testing is also expected to be achieved. A verdict on the practicalities of automatic testing techniques in Java Card applications will be presented.

References

[1] *Software Testing: A Guide to the TMap Approach.* Addison Wesley, London, ISBN 0201745712, 2001.

[2] BC Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.

[3] R. Heckel and M. Lohmann. Towards Model-Driven Testing. *Electronic Notes in Theoretical Computer Science*, 82(6), 2003.

[4] J. Tretmans and E. Brinksma. Torx: Automated model based testing. In *A. Hartman, K. Dussa-Zieger, First Eur. Conf. on Model-Driven Software Engineering. Imbuss, Moehrendorf, Germany, 2003.*

[5] Sun Developer Network. Java platform, micro edition (java me). http://java.sun.com/javame/.

[6] Bella Giampaolo and Paulson Lawrence. Kerberos version iv: Inductive analysis of the secrecy goals. *Lecture Notes in Computer Science*, 1998.

[7] Kerberos: An Authentication Service for Open Network Systems. *Proc. Winter USENIX Conference*, 1988.

[8] A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols. *Proc. SAPS*, 4.

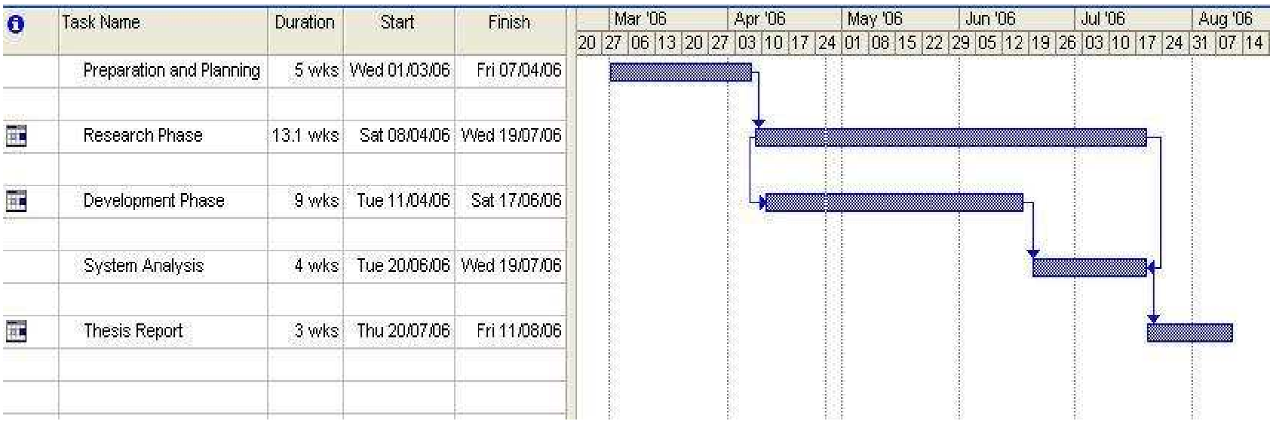
[9] The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. *17th International Conference on Computer Aided Verification, CAV*, pages 281–285, 2005.

[10] Gary T. Leavens, Albert L. Baker, and Clyde Ruby. Preliminary design of JML: A behavioral interface specification language for Java. Technical Report 98-06i, 2000.

[11] T. Bolognesi and E. Brinksma. Introduction to the iso specification language lotos. *COMP. NETWORKS ISDN SYST.*, 14(1):25–59, 1987.

[12] The PROMELA Language. *http://www.dai-arc.polito.it/dai-arc/manual/tools/jcat/main/node168.html*.

Appendix: 1



Task Name	Duration	Start	Finish
Preparation and Planning	5 wks	Wed 01/03/06	Fri 07/04/06
Research Phase	13.1 wks	Sat 08/04/06	Wed 19/07/06
Development Phase	9 wks	Tue 11/04/06	Sat 17/06/06
System Analysis	4 wks	Tue 20/06/06	Wed 19/07/06
Thesis Report	3 wks	Thu 20/07/06	Fri 11/08/06

Figure 4: *Gantt chart*