A MODEL-BASED TESTING FRAMEWORK FOR TRUSTED PLATFORM MODULE

USAMA THARWAT FARAG ELHAGARI

A thesis submitted in fulfilment of the
requirements for the award of degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

NOVEMBER 2015

DEDICATION

To my beloved mother, father, mother in law, father in law, wife, children and family

ACKNOWLEDGEMENT

First and foremost, gratitude and praises goes to Almighty Allah, in whom I have put my faith and trust in. I would like to extend my deep gratitude to my father, and my lovely mother for their honorable and courteous support all the way through my life to date. Furthermore, I would like to prolong gratitude to my adorable father-in-law and mother-in-law; the memory will not forget the joy and the concrete support they made. I am indebted to my lovely wife Wafa’ for her worthy, hearted cooperation, inspiration, encouragement and funding reinforcement to me, and also my friendly adorable children Abdul-Rahman, Mohammad, and Rahmah, whose my heart always have the affection of their joy, may Almighty Allah increases his mercy and blessing on them, guide them and protect them forever. I placed in record my deep gratitude and special thanks to Assoc. Prof Dr. Zailani Mohamed Sidek and Dr Jamalul-lail Ab. Manan for their support, encouragement, guidance and positive criticism. I owe my concrete learning to them. I do not have enough words to express my thanks to them but I pray Allah (SWA) will continue to give them the strength, guidance, wisdom and reward them with the best of both worlds, Ameen. I will also like to thanks Dato' Professor Dr. Norbik Bashah Idris for his help and support in finishing my PhD successfully. In addition, my special deep gratitude to my supervisor Dr. Bharanidharan Shanmugam for his precious advice, timely response, valuable comments, guidance and fund support. I will also use this opportunity to express my gratitude to Prof. Dr. Mohamed Mahmoud El-Sayed Nasef and Dr. Abdurahman Kalema, their support and courage to me are unforgettable. Last but not the least I would like to thank sisters, brothers, sister - in- law, brothers-in-law, nephews, nieces and friends whom I owe a debt of attitude for their prayers, encouragement and moral support throughout the whole duration of my studies.

ABSTRACT

Trusted Computing Group (TCG) provides Trusted Platform Module (TPM) specifications, as the core of Trusted Computing (TC) technology, to industry in order to overcome the failure of protecting sensitive data using software-only security mechanisms. Currently, TPM is implemented as integrated circuit mounted in computing platforms. Over 200 million TPM, from different vendors, nowadays are already mounted in computing platforms, such as laptops and desktops. So, there is an urgent need to verify the correctness of these TPM implementations and testing their security functionality. However, research on TPM testing and evaluating TC products is still in the initial stage. As far as our knowledge goes, a TPM Testing Framework (TPM-TF) and Test Automation (TA) have not been well established yet. This research contributes in the TPM testing by designing and developing an enhanced TPM-TF that combines the TPM compliance testing, TPM security testing, and simulation of the TPM allowed behaviour. The proposed TPM-TF is proven to be scalable, where it could conduct three different on-line automatic tests namely function test, command test, and security test for any TPM implementation of certain TPM specifications version. These tests serve four testing quality dimensions which are functionality, reliability, robustness, and security. For these tests, TPM-TF has generated valid and random off-line and on-the-fly test cases using Input-Output Conformance testing theory and its algorithm, without suffering from the state space explosion problem. Additionally it has the capability of automatic and interactive simulating the TPM specifications based on Coloured Petri Nets (CPN) theory. This capability serves not only TPM experts but also users who have abstract background about TPM. The main contribution of this research is TPM-TF can provide TPM testing services to government, organisations and most importantly the Common Criteria facility in Malaysia.

ABSTRAK

Kumpulan Pengkomputeran Dipercayai (TCG) menyediakan spesifikasi Modul Platform Dipercayai (TPM), sebagai teras teknologi Pengkomputeran Dipercayai (TC), kepada industri untuk mengatasi kegagalan melindungi data sensitif menggunakan perisian sebagai mekanisme keselamatan. Pada masa ini, TPM dilaksanakan sebagai litar bersepadu yang dipasang di dalam platform perkomputeran. Lebih 200 juta TPM, daripada vendor yang berlainan, pada masa kini telah dipasang dalam platform perkomputeran, seperti komputer riba dan komputer meja. Oleh itu, terdapat keperluan segera untuk mengesahkan kebenaran daripada pelaksanaan TPM dan menguji fungsi keselamatannya. Walau bagaimanapun, penyelidikan pada ujian TPM dan penilaian produk TC masih dalam peringkat awal. Sepanjang pengetahuan kami, Pengujian Rangka Kerja TPM (TPM-TF) dan Ujian Automasi (TA) masih belum mantap lagi. Kajian ini menyumbang dalam pengujian TPM dengan mereka bentuk dan membangunkan TPM-TF yang diperbaiki yang menggabungkan ujian pematuhan TPM, ujian keselamatan TPM dan simulasi tingkah laku TPM yang dibenarkan. TPM-TF yang dicadangkan terbukti boleh diskalakan, di mana ia boleh menjalankan tiga perbezaan ujian automatik dalam talian iaitu ujian fungsi, ujian perintah dan ujian keselamatan bagi sebarang pelaksanaan bagi spesifikasi versi TPM. Ujian ini menjurus kepada empat ujian kualiti dimensi terdiri daripada kefungsian, kebolehpercayaan, keteguhan, dan keselamatan. Bagi ujian ini, TPM-TF telah menjana pengesahan dan ujian kes-kes rawak seperti luar talian dan *on-the-fly* menggunakan teori ujian Pergesahan Input-Output dan algoritmanya, tanpa mengalami masalah letupan ruang. Selain itu ia mempunyai keupayaan mensimulasi secara automatik dan interaktif spesifikasi TPM berdasarkan teori Petri Nets Berwarna (CPN). Keupayaan ini berguna bukan sahaja untuk pakar TPM tetapi juga pengguna yang mempunyai latar belakang TPM yang abstrak. Sumbangan utama penyelidikan ini adalah TPM-TF yang menyediakan perkhidmatan ujian TPM kepada agensi kerajaan, organisasi dan yang paling penting ialah kemudahan Kriteria Bersama di Malaysia.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Preamble

Over the past fifteen years, a highly developed information society has changed our everyday lives. Many types of data are converted into digital information and secret information is often sent through the internet, which gives us all the advantages of connectivity but also brings heightened security risks. Hence, information has become an organization's and individuals' precious asset which should be secured. Therefore, information security looks like a peripheral fence protecting our modern life from threats such as loss, manipulation, espionage, and information leakage.

Recently, software on computing platforms gets increasingly complex which leads to a large number of vulnerabilities. Consequently, protecting information technology systems through software-only mechanisms can not solve all the security problems alone (Berger, 2005; Lin, 2005). Therefore, the approach of using hardware-based embedded security solution was introduced to the information technology industry. Given the importance of using the hardware-based approach, the Trusted Computing Platform Alliance (TCPA), replaced by the Trusted Computing Group (TCG), has proposed the Trusted Computing (TC) concept. TC becomes a base of new computing platform architecture (hardware and software) that, practically, has a trusted hardware component built in hardware layer and a trusted software component installed in operating system level (Kallath, 2005). The trusted hardware component is called Trusted Platform Module (TPM) whose specifications have been issued by TCG group and they are implemented by industry as a tamper-resistant integrated circuit. So, any platform equipped and enforcing TC functionalities using TPM chip is called Trusted Platform (TP). TCG defines TPs as "the platforms that can be expected to always behave in certain manner for and intended purpose". TCG has issued two specifications, so far, for TPM: version 1.1b (TCPA, 2002b), version 1.2 (TCG, 2011d). In March 2013 TCG updated the TPM specifications by issuing the TPM 2.0 library specifications (TCG, 2013) which have been posted to for public review in May 2014.

This thesis mainly covers the last revision of TPM specifications version 1.2 which is revision 116. There are two main reasons for selecting this version. Firstly, to the best of our knowledge, all the TPM chips in the market nowadays are implantations of the TPM specifications version 1.2. Secondly, many security analysis research works have been done using formal methods to analyse the TPM specifications. These research works have mainly analysed the TPM specifications versions 1.1 and 1.2, such as in (Donglai *et al.*, 2013; Gürgens *et al.*, 2008; Seifi, 2014).

1.2 Background of the Problem

According to (Challener *et al.*, 2008), TCG had specified the design goals of the TPM. The design should be capable of doing the following:

i. Securely report the environment that has been booted
ii. Securely store data
iii. Securely identify the user and system (without encountering privacy concerns)
iv. Support standard security systems and protocols
v. Support multiple users on the same system while preserving security among them
vi. Be produced inexpensively

Figure 1.1 presents the main components of Trusted Platform (TP) model: a TPM chip, a firmware called Core Root of Trust for Measurement (CRTM) uploaded in the Platform's BIOS, and a trusted software component called TCG Software Stack, Trusted Software Stack, or Trusted platform Support Service (TSS) (Balacheff *et al.*, 2002).

TP's architecture contains modified BIOS, which includes CRTM as piece of code added to the conventional BIOS firmware. In conventional platforms, booting process starts by running the BIOS firmware, whereas TP runs, first, the CRTM code, which performs integrity check of every hardware and software component in TP, starting with integrity check of BIOS firmware. The TSS has many functions; for instance TSS works as an interface to TPM and manages its resources. In order to prove originality of TP, Certification Authorities (CAs) issue certificates to vouch that the TP is genuine.

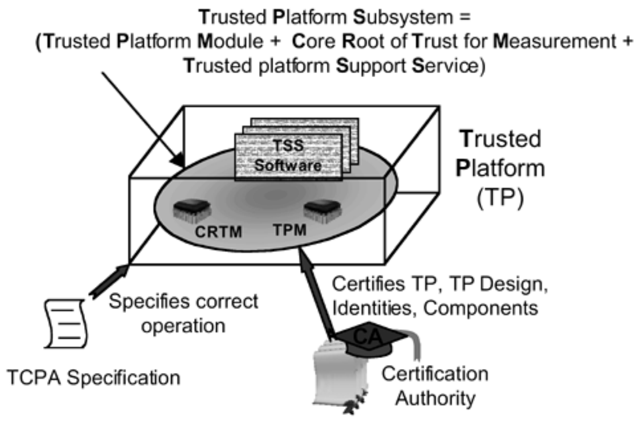

Trusted Platform Subsystem =
(Trusted Platform Module + Core Root of Trust for Measurement +
Trusted platform Support Service)
TSS
Software
CRTM
TPM
Trusted
Platform
(TP)
Specifies correct operation
TCPA Specification
Certifies TP, TP Design, Identities, Components
CA
Certification Authority

Figure 1.1 Trusted computing platform

Nowadays, hundreds of millions PC laptops and desktops have been equipped with TPM chips (Hardjono, 2008). In practice, there are different vendors producing TPM chips and, of course, with different implementations. However, there is an urgent need to have a testing methodology helping security application developers and end-users to verify the compliance of their TPM-enabled systems against TPM specifications (Sadeghi *et al.*, 2006a). Similarly, the compliance test of TSS with TCG specifications is a critical necessity to ensure its quality (He *et al.*, 2008a). The recent efforts show that many TPMs available on the market are noncompliant to the TPM specifications (Chen, 2009; He *et al.*, 2010; Sadeghi, 2006; Xu *et al.*, 2009a; Zhan *et al.*, 2008; Zhang *et al.*, 2008; Zhang *et al.*, 2010). At this point, it is worth mentioning that China has its own specifications and trusted hardware component called Trusted Cryptography Module (TCM). Although TCM chip has been specified and manufactured by China, there is a gap between the TCM implementations and the Chinese specifications (Li *et al.*, 2009b). Generally, this emphasises the need of testing of any implementation against its specification. In addition to the non-compliance of many TPMs, researchers showed also non-compliance of several related products of TSS (He *et al.*, 2008b; Li *et al.*, 2009b; Tóth *et al.*, 2008; Zhang *et al.*, 2008).

According to the literature the quality assurance in the field of TPM has two main directions namely: concrete TPM testing and security analysis on TPM specifications. Additionally, real attacks against TPM implementations have been conducted and revealed security flaws in some of current TPM implementations.

In the TPM testing direction, The TPM testing frameworks can be classified based on the formality of the specification that each framework relies on. However, the TPM specifications that issued by TCG are basically informal specifications. The first valuable contribution in the concrete TPM testing was done by the researchers in (Sadeghi *et al.*, 2006a; Sadeghi *et al.*, 2006b). However, their TPM testing framework is informal(Li *et al.*, 2009b) which means that the test cases generation was not automatic and relied on informal specifications. Thus the testing method needs to be improved so that it becomes automatic and systematic (Zhang et al., 2010).

The other existing TPM testing frameworks are mainly based on either the finite state machine (FSM) (Zhan *et al.*, 2008; Zhang *et al.*, 2008) or the extended finite state machine (EFSM) (LI *et al.*, 2009a; Li *et al.*, 2009b; Xiao-Feng, 2009).

The first formal TPM testing framework was based on FSM and introduced by (Zhan *et al.*, 2008; Zhang *et al.*, 2008). Its testing approach is model-based testing (MBT) based on FSM specification. Despite the simplicity of modelling specifications using FSM, modelling large systems by using FSM might not be practical (Petrenko *et al.*, 2004). This is because the number of states in FSM model increases dramatically, which leads to the state space explosion problem (Bourhfir *et al.*, 1997). This makes the FSM not capable of modelling real systems in concise way (Lee and Yannakakis, 1996).

The EFSM-based TCM testing framework in (LI *et al.*, 2009a; Li *et al.*, 2009b) has made some improvement to the FSM-based TPM testing framework introduced by (Zhan *et al.*, 2008; Zhang *et al.*, 2008) in modelling and generating test cases. However, the research of the EFSM-based TCM testing framework stated that test cases generation was semi-automatic. Thus, it is needed further development test cases generation and in parameter relation among the TCM commands.

Researchers in (Xiao-Feng, 2009) have proposed a TPM testing framework based on EFSM. This framework is similar to the EFSM-based TCM testing framework. However, starting point of the proposed EFSM-based TPM testing framework was modelling the TPM specifications using Z language. An EFSM model was extracted from the constructed Z model. The EFSM model was verified using a model checking tool, such as SPIN, and consequently test cases were generated. It is reported that the test cases generation was not completely automated.

It is worth noting that all of these TPM testing frameworks depend on batch mode testing which means that test cases are generated earlier than conducting the test. furthermore, although (Sadeghi *et al.*, 2006a) stated that the TPM testing framework should help security application developers and end-users, i.e. TPM stakeholders, to verify the compliance of their TPM-enabled systems against TPM specifications, all the existing TPM testing framework serve TPM users who have solid knowledge background about TPM.

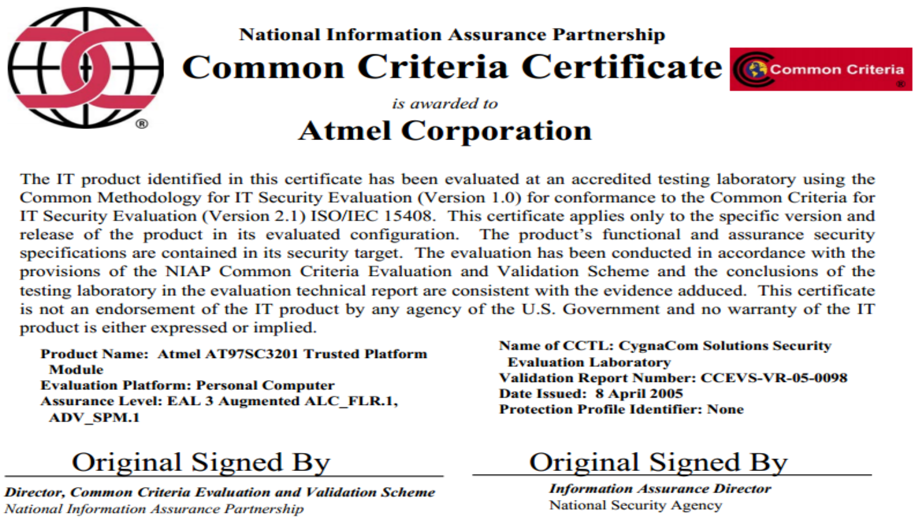
Considering the direction of security analysis on the TPM specifications, although the main function of TPM chips is establishing trust and is to provide security services to their host platforms, many attacks have been performed against either the TPM chip itself or its environment, such as communication interface with the other platform's components. These attacks are either practical attack, such as reset attack (Bernhard, 2007) and the physical attack which was performed by Christopher Tranovsky (Tarnovsky, 2010), or security flaws that have been revealed by security analysis research work on the TPM specifications, such as the Object-Independent Authorization Protocol (OIAP), which is a TPM security protocol mainly intended to prevent replay attack, has found having a problem in its design which makes it vulnerable to replay attack (Bruschi *et al.*, 2005) . Additionally, The authors of (Delaune *et al.*, 2011; Donglai *et al.*, 2013; Gürgens *et al.*, 2008) proved formally the integrity of the TPM_CertifyKey command can be violated due to a design problem in the Hash-Based Message Authentication Code (HMAC) calculation. Furthermore, both of BitLocker, an encryption feature provided by Microsoft Windows, and the Intel Trusted Execution Technology (TXT) have been successfully attacked by Fraunhofer Institute for Information (SIT) (Chen *et al.*, 2009) and Invisible Things Lab (ITL) (Wojtczuk and Rutkowska, 2009) respectively.

Although the results of the security analysis on the TPM specifications do play a crucial role in evaluating the quality of the TPM specifications and subsequently the security functionality provided by the TPM chips that implemented based on the specifications, to the best of our knowledge, none of the existing TPM testing frameworks, (LI *et al.*, 2009a; Sadeghi *et al.*, 2006a; Xiao-Feng, 2009; Zhan *et al.*, 2008; Zhang *et al.*, 2008), has ever used these testing results to evaluate the TPM under test.

Due to the security evaluation of TPM being highly required and essential for giving the consumers the necessary confidence of using any TC products, both vendors of TPM chips and researches are interested in evaluating the security of TPM, but of course they have different goals. Vendors will always seek for attracting more consumers and open new markets, but researchers always want to assure that the claims of the TPM vendors can be proven or verified to be true (Sadeghi *et al.*, 2006a; Sadeghi *et al.*, 2006b; Zhang *et al.*, 2008; Zhang *et al.*, 2010).

The security evaluations of TPM took different ways for vendors and researcher. On the one hand, TPM vendors wanted to prove to the TPM consumers the quality of the security services provided by their TPMs, and hence they often requested Common Criteria (CC) laboratories to evaluate their TPM chips. On the other hand, past researchers wanted to evaluate and verify the security of TPM by performing testing, i.e. TPM compliance testing, and security analysis on the TPM specifications. Many research works have been done to test and verify the validity of the vendors claims that the TPM chips they produced comply with TPM Specifications. Consequently, performing compliance testing to TPM and its supporting software, TSS, became more prominent in many research works such as in (He *et al.*, 2008b; Zhan *et al.*, 2008). The authors of (Xu *et al.*, 2009a) stated that "trusted platform must be under security evaluation, otherwise, neither the quality of TPM products, nor the security of information systems can be guaranteed". Furthermore, (Sadeghi, 2008; Sadeghi *et al.*, 2006a) stated that the complexity of the TPM specifications may lead to implementing TPM chips that not exactly as specified.

Common Criteria (CC, 2005b) is a standard containing common set of requirements for evaluating the security of IT products. Zhang, H., et al. (Zhang *et al.*, 2010) stated that while CC is a complex process, as far as their knowledge goes, there has been no framework for solid security analysis of TPM as well as trusted platform. Furthermore, CC depends strongly, during the evaluation process, on Security Target (ST) document and test cases for performing tests of the TPM, which ironically, are both produced by TPM vendors themselves. CC evaluates only the TPM, regardless its operational environment (i.e. the TP) which from CC point of view, is the vendor’s responsibility. It can be safely concluded that CC evaluation process has somehow been mainly vendor-dependent and not an independent CC evaluation. This conclusion explains why the Reset Attack and Passive Attack happened in the past. Due to this also CC evaluation results may not satisfy the community users' requirements and CC may not have the tools to do thorough examination of all the TPM vendors' claims. As an example of such unsatisfactory CC evaluation case is the following. In April, 2005 the Atmel AT97SC3201 TPM chip passed successfully the CC security evaluation, reported in (CC, 2005a) ; the CC certificate is shown in Figure 1.2 (NIAP, 2005). However, in May 2006, the authors of (Sadeghi *et al.*, 2006a; Sadeghi *et al.*, 2006b) found the Atmel AT97SC3201 TPM chip did not comply with the TPM specifications. This unsatisfactory case also emphasises the need of developer-independent test cases for testing the TPM implementations.



National Information Assurance Partnership

Common Criteria Certificate

Common Criteria

is awarded to

Atmel Corporation

The IT product identified in this certificate has been evaluated at an accredited testing laboratory using the Common Methodology for IT Security Evaluation (Version 1.0) for conformance to the Common Criteria for IT Security Evaluation (Version 2.1) ISO/IEC 15408. This certificate applies only to the specific version and release of the product in its evaluated configuration. The product's functional and assurance security specifications are contained in its security target. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced. This certificate is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

Product Name: Atmel AT97SC3201 Trusted Platform Module
Evaluation Platform: Personal Computer
Assurance Level: EAL 3 Augmented ALC_FLR.1, ADV_SPM.1

Name of CCTL: CygnaCom Solutions Security Evaluation Laboratory
Validation Report Number: CCEVS-VR-05-0098
Date Issued: 8 April 2005
Protection Profile Identifier: None

Original Signed By

Director, Common Criteria Evaluation and Validation Scheme
National Information Assurance Partnership

Original Signed By

Information Assurance Director
National Security Agency

Figure 1.2 CC certificate for Atmel AT97SC3201 TPM chip

1.3 Statement of the Problem

In light of the above, the motivation of doing this study can be briefed as follows:

i. Almost all laptops and majority of desktops have been equipped with TPM chips.

ii. There is gap (incompliance) between TPM implementations and TPM specifications, and also between TCM and the Chinese specifications.

iii. Different attacks, either by physical attack or by performing security analysis, against TPM have been announced (Reset attack, dictionary attack, replay attack, physical attack). However, to the best of our knowledge, none of the existing TPM testing framework has used any of the results of these research works of the security analysis on the TPM specifications.

iv. The existing TPM testing frameworks mainly serve the TPM users who have solid knowledge background about the TPM.

v. The existing TPM testing frameworks mainly generate the test cases prior to executing them in the TPM. This method is called batch mode or off-line testing and the resulting test suite is fixed and being used for testing any TPM implementation. However, many off-line test case generation methods suffer from the state space explosion problem. The on-line testing approach helps in eliminating this problem (Jüri *et al.*, 2011; Kull *et al.*, 2009). Generating test cases in the on-line testing is made on the fly, i.e. the test cases can be executed while they are derived from specification model. Furthermore, the on-line testing is normally based on so-called random walk state exploration strategy (Mihail and Papadimitriou, 1994; West, 1989). Based on this strategy, the tracing of the specification model is made randomly to generate different test suite in every test cases generation. This randomness helps in detecting unexpected and sophisticated bugs, in implementation under test (IUT), that hard to be detected by off-line testing (Jüri *et al.*, 2011). Additionally, this strategy is an effective strategy in testing complex and large systems where it leads to generate stressful, complicated, and long test cases.

vi. Results of CC security evaluation of TPM may not satisfy the community users' requirements and does not examine the TPM vendors' claims. It is crucially needed to verify vendors’ claims.

This study addresses the problem of accurate and verified TPM testing in computer systems. It is generally known that vulnerabilities in systems provide attractive motivations for attackers to perform security attacks; these attacks are done by insiders or outsiders. More so, providing software-only security mechanisms in most cases fail to protect sensitive data. The TPM is TCG's answer of how to provide a security mechanism which is based at hardware-component level. In practice, the current implementation of the TPM specifications is hardware Integrated Chip (IC), mounted in every TC product; Different TPM vendors have different TPM implementations, claiming that their TPMs are compliant to TPM specifications. However, TC products consumers have no concrete method to assess the compliance level of the TC products due to the strict non-disclosure policies of TPM vendors, leading to blindly trusting these products. The authors of (Sadeghi *et al.*, 2006a; Sadeghi *et al.*, 2006b; Zhang *et al.*, 2010) have made an urgent call to develop a vendor-independent TPM testing framework to verify the correctness TPM implementations and testing their security functionality as well as verifying the claims of the TPM vendors.

According to the literature, there are four existing frameworks that have been developed for testing TPM implementations (LI *et al.*, 2009a; Sadeghi *et al.*, 2006a; Xiao-Feng, 2009; Zhan *et al.*, 2008; Zhang *et al.*, 2008). Although the good achievements of these TPM testing frameworks, they suffer from the following weaknesses:

(1) Past research works on testing frameworks which were designed based on FSM or EFSM generated their test cases prior to conducting the test (aka batch-mode testing), and suffered from the issue of a high possibility of space state explosion problem.
(2) The existing TPM testing frameworks, so far, serve TPM testers or security application developers who have solid background about TPM only but normal users do not have the required skill to use them with confidence.
(3) To the best of our knowledge, none of the existing TPM testing frameworks has ever used the results of the security analysis on the TPM specifications to evaluate the TPM under test.

This study is a response to the call made by (LI *et al.*, 2009a; Sadeghi *et al.*, 2006a; Xiao-Feng, 2009; Zhan *et al.*, 2008; Zhang *et al.*, 2008). Additionally, it tackles the mentioned weaknesses of the existing TPM testing frameworks. Furthermore, the rationale of this study is to assist both TC products' users (consumers or governments) not only to judge whether the TC products they purchased are appropriate but also to use these products with confidence. The problem that this study tackles is:

Enhancing the testing framework for Trusted Platform Module with emphasis on using simulation and on-line automated testing

The study seeks to propose and develop enhanced testing framework that is capable of simulating the TPM specifications and conducting on-line automated testing. Furthermore, the proposed testing framework provides, not only test the TPM compliance, but also the TPM security. To guide us on the intended full fledge approach, the research questions to be answered by the study are:

i. What are the needed components that can be used to propose an enhanced TPM testing framework that can tackle the weaknesses of the existing TPM testing frameworks?
ii. How to design the proposed TPM testing framework?
iii. How to model the TPM specifications for on-line automated testing and simulating the allowed behaviour of the TPM according the TPM specifications?
iv. How to test and evaluate the proposed TPM testing framework?

1.4 Aim of the Study

The aim of this study is to propose and develop an enhanced TPM testing framework that combines automatic TPM compliance testing, automatic security testing, and (automatic and interactive) simulation/ demonstration of TPM allowed behaviour based on the specifications. The proposed framework is based in IOCO testing theory to generate random on-line test cases from the TPM specifications which is modelled using Labelled Transition System (LTS). The simulation/demonstration part in the proposed framework is based on the Coloured Petri Nets (CPN). Finally, the security testing that will be conducted using our framework will be based on the results of the published security analysis (past works) on the TPM specifications. Applicable security tests and scenarios will be extracted from these results and then modelled by LTS to generate random on-line test cases based on IOCO.

1.5 Objectives of the Study

Based on the research questions, the specific objectives of this study are:

i. To identify the components of the TPM testing framework.
ii. To design the TPM testing framework.
iii. To develop the TPM testing framework.
iv. To test and evaluate the TPM testing framework.

1.6 Scope of the Study

In this thesis, our scope of research will mainly focus on TPM testing in TC products, especially laptops and desktops PCs equipped by TPM chips. The target beneficiaries are TC consumers and security evaluators. It has already determined that currently there is no testing framework that combines the TPM compliance testing with security testing and serves TPM users with different background knowledge about TPM as well as the state space explosion is a main problem in the existing TPM testing frameworks. Furthermore, due to the TPM industry non-disclosure polices, the public is restricted from getting TPM testing tools, or to know the results of the TPM compliance and TPM security tests that have been performed by the TPM developers. Therefore, recognizing these facts and to solve these problems, the scope of our study is as follows:

i. This study will provide a comprehensive analysis on the testing of TPM implementations, excluding TSS implementations or TPM chain of trust, then propose an enhanced TPM testing framework, which will be based on sources such as extensive review of literature, TPM specifications, security analysis of research works on TPM specifications, trusted computing tutorials and summer school.
ii. Due to the lack of information from industry, our study will depend very much, on the currently available technology (for example; open source software, available tools for modelling, verification, and testing).

iii. Because of time constraints of the developing process in the proposed framework, a portion of the TPM specifications version 1.2 (revision116) will be covered only. Additionally, some of the results of the security analysis from past research works will be used in the framework. Additionally, a TPM-behaviour emulator, which emulates the correct behaviour of TPM according to the TPM specifications, will be used rather than a concrete (actual) TPM implementation.

The outcome of this study offers additional insight into the TPM Testing area.

1.7 Significance of the Study

Currently, the TPM is a hardware component built into the motherboards of laptops and desktops PCs. Therefore, these platforms can be considered as trusted platforms, so long as the TPM functionalities are enforced. Because of the variety of TPM vendors and, of course, their different TPM implementations, there is an urgent need to test these TPMs, to enable users to use these platforms with confidence.

Our study proposes an enhanced testing framework for the TPM. This proposed framework combines automated compliance testing, automated security testing, and simulating the allowed TPM behaviour. It is capable of generating random on-line test cases. Therefore, it can be used for the Common Criteria (CC) laboratories to generate developer-independent test cases for evaluating the security of TPM implementations. It is believed that this framework will contribute in increasing the quality of evaluating the TPM using CC. Furthermore, the simulation capability in the proposed framework will serve TPM stakeholders with different knowledge background about the TPM. Therefore, this framework not only helps government to manage the TC products but also assists users/consumers in selecting their TC product requirement.

Above all, the developing world is not a producer of the TC technology, but only a consumer. So it is crucial for the sub-region to at least have their own TC testing tools rather than trusting blindly on current TC technology.

According to the literature, there are two developed TPM testing frameworks that have been developed, namely by Ruhr University, Germany, and Wuhan University, China. Our proposed enhanced TPM testing framework makes Malaysia the third country to have produced the TPM testing framework. Furthermore, the CC facility in Malaysia can benefit from our framework to generate developer-independent test cases to increase the quality of evaluating the TPM implementations within Malaysia.

1.8 Organization of the Thesis

In this chapter the research study has been introduced. Firstly, general introduction about the study was presented. After that, the background and motivation highlighting the problems that motivate for doing this study were described. Also, the statement of the problem, aim, objectives, scope, and significance of the study were described respectively.

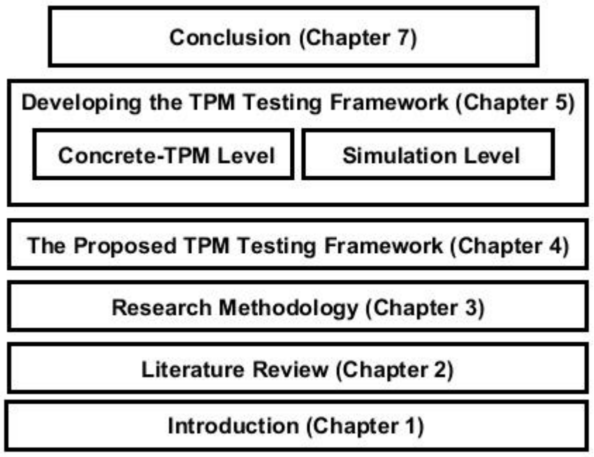

Conclusion (Chapter 7)
Developing the TPM Testing Framework (Chapter 5)
Concrete-TPM Level
Simulation Level
The Proposed TPM Testing Framework (Chapter 4)
Research Methodology (Chapter 3)
Literature Review (Chapter 2)
Introduction (Chapter 1)

Figure 1.3 Thesis skeleton

Figure 1.3 shows the thesis skeleton. Chapter 2 presents a detailed review on the body of knowledge related to the trusted computing technology, Trusted Platform Module, and TPM testing frameworks. Chapter 3 describes the research methodology of the study. Chapter 4 explains the proposed TPM testing framework, aka TPM-TF which contains two operational levels, namely, simulation level and concrete-TPM level. Additionally, Chapter 4 conducts a comparison between the existing TPM testing framework and the proposed framework. Chapter 5 is dedicated for the development of the proposed TPM testing framework. Thus, Chapter 5 explains the modelling, verification, and development processes of the two operational levels: concrete-TPM level and simulation level. Additionally, the results and discussions of developing the proposed framework are demonstrated in Chapter 5. The conclusion and recommendations are presented in Chapter 6.

REFERENCES

Aaraj, N., Raghunathan, A., and Jha, N. K. (2008). Analysis and Design of a Hardware/Software Trusted Platform Module for Embedded Systems. *Transactions on Embedded Computing Systems, 8*(1), 1-31.

Aarhus-University. (2007a). ASCoVeCo Project Downloads. Retrieved December, 2014, from http://www.cs.au.dk/~ascoveco/download.html

Aarhus-University. (2007b). ASCoVeCo Project Homepage. Retrieved December, 2014, from http://www.cs.au.dk/~ascoveco/asap.html

Ables, K. (2009). *An Alleged Attack on Key Delegation in the Trusted Platform Module.* MSc Advanced Computer Science First semester mini-project. University of Birmingham, United Kingdom.

Abran, A., Moore, J. W., Bourque, P., Dupuis, R., and Tripp, L. (2004). Guide to the Software Engineering Body of Knowledge, 2004 version. *1*.

Aichernig, B. K. (2013). Model-Based Mutation Testing of Reactive Systems. In *Theories of Programming and Formal Methods* (pp. 23-36): Springer.

Al-Azzoni, I., Down, D. G., and Khedri, R. (2005). Modeling and Verification of Cryptographic Protocols Using Coloured Petri Nets and Design/CPN. *Nordic Journal of Computing, 12*(3), 201.

Appel, A. W., and MacQueen, D. B. (1991). *Standard ML of New Jersey.* Proceedings of the Programming Language Implementation and Logic Programming, 1-13.

Baier, C., and Katoen, J.-P. (2008). *Principles of Model Checking* (Vol. 26202649): MIT press Cambridge.

Balacheff, B., Chen, L., Pearson, S., Plaquin, D., and Proudler, G. (2002). *Computing Platforms: TCPA Technology in Context.* Englewood Cliffs, NJ.: Prentice Hall PTR.

Banks, J. (2000). *Introduction to Simulation.* Proceedings of the Winter Simulation Conference 2000. vol.11, 9-16.

Belinfante, A. (2010). JTorX: A Tool for On-Line Model-Driven Test Derivation and Execution. In *Tools and Algorithms for the Construction and Analysis of Systems* (pp. 266-270): Springer.

Belinfante, A., Feenstra, J., de Vries, R. G., Tretmans, J., Goga, N., Feijs, L., et al. (1999). Formal Test Automation: A simple experiment. In *Testing of Communicating Systems* (pp. 179-196): Springer.

Belinfante, A., Feenstra, J., Heerink, L., and de Vries, R. G. (2001). Specification Based Formal Testing: The Easylink Case Study. Proceedings of the 2nd Workshop on Embedded Systems, 73-82.

Belinfante, A. F. E. (2014). *JTorX: Exploring Model-Based Testing*. Ph.D. Thesis. University of Twente.

Berger, B. (2005). *Trusted Computing Group History*. Information Security Technical Report, 10(2), 59-62.

Bernhard, K. (2007). *OSLO: Improving the Security of Trusted Computing*. Proceedings of the 16th USENIX Security Symposium, Vol. 7, 1-9.

Bernot, G., Gaudel, M. C., and Marre, B. (1991). Software Testing Based on Formal Specifications: a theory and a tool. *Software Engineering Journal, 6*(6), 387-405.

Blanchet, B. (2001). *An Efficient Cryptographic Protocol Verifier Bbased on Prolog Rules*. Proceedings of the Computer Security Foundations Workshop, IEEE, 82-96.

Blanchet, B. (2005). *ProVerif Automatic Cryptographic Protocol Verifier User Manual*. CNRS, Departement dInformatique, Ecole Normale Superieure, Paris, from http://www.di.ens.fr/ blanchet/crypto-eng.html.

Bochmann, G. V., and Gecsei, J. (1977). *A Unified Method for the Specification and Verification of Protocols**. Proceedings of IFIP Congress 77, 229-234.

Bohnenkamp, H., and Belinfante, A. (2005). Timed testing with TorX. In *FM 2005: Formal Methods* (pp. 173-188): Springer.

Boss, E., and Poll, E. (2012). *Evaluating Implementations of SSH by Means of Model-Based Testing*. Bachelor Thesis. Radboud University.

Bourhfir, C., Dssouli, R., Aboulhamid, E., and Rico, N. (1997). Automatic Executable Test Case Generation for Extended Finite State Machine protocols. In *Testing of Communicating Systems* (pp. 75-90): Springer.

Bozga, M., David, A., Hartmanns, A., Hermanns, H., Larsen, K. G., Legay, A., et al. (2012). *State-of-the-art Tools and Techniques for Quantitative Modeling and Analysis of Embedded Systems.* Proceedings of the Conference on Design, Automation and Test in Europe, 370-375.

Brandan Briones, L. (2007). *Theories for Model-Based Testing: Real-time and Coverage*. Ph.D. Thesis. University of Twente.

Brandl, H. (2004). *Trusted Computing: The TCG Trusted Platform Module Specification*. Infineon Technologies AG, from http://www.wintecindustries. com/orderdesk/TPM/Documents/TPM1. 2_-_Basics. pdf

Brinksma, E. (1987). *On the Eexistence of Canonical Testers*. Memorandum INF-87-5 Twente University of Technology.

Brinksma, E. (1988). A Theory for the Derivation of Tests. In S. Aggarwal and K. Sabnani (Eds.), *Protocol Specification, Testing and Verification VIII*. (pp. 63-74). North-Holland: Elsevier.

Brinksma, E., and Tretmans, J. (2001). Testing Transition Systems: An Annotated Bibliography. In *Modeling and verification of parallel processes* (pp. 187-195): Springer.

Broy, M. , Jonsson, B., Katoen, J. P., Leucker, M., and Pretschner, A. (Eds.). (2005). *Model-Based Testing of Reactive Systems: Advanced Lectures* (Vol. 3472): Springer.

Bruschi, D., Cavallaro, L., Lanzi, A., and Monga, M. (2005). *Replay Attack in TCG Specification and Solution*, Tucson, AZ, United states, 127-137.

Burrows, M., Abadi, M., and Needham, R. (1988). *Authentication: A Practical Study in Belief and Action.* Paper presented at the Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge, 325-342.

Burrows, M., Abadi, M., and Needham, R. M. (1989). *A Logic of Authentication*. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 426(1871), 233-271.

Cai, L., Zhang, J., and Liu, Z. (2010). *Generating Test Cases Using Colored Petri Net.* Proceedings of the 2010 2nd International Symposium on Information Engineering and Electronic Commerce (IEEC), 1-5.

CC. (2005a). Common Criteria Evaluation and Validation Scheme, Validation Report, Atmel Corporation, Atmel AT97SC3201 Trusted Computing Module (TPM). from https://www.commoncriteriaportal.org/files/epfiles/st_vid3005-vr.pdf

CC. (2005b). Common Criteria for Information Technology Security Evaluation. from https://www.commoncriteriaportal.org/cc/

CC. (2009a). Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1.

CC. (2009b). Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements, Version 3.1.

CC. (2009c). Common Criteria, Trusted Computing Certified Products, from http://www.commoncriteriaportal.org/products_ALL.html#512010

Challener, D., Yoder, K., Catherman, R., Safford, D., and Doorn, L. V. (2008). *A practical Guide to Trusted Computing*: IBM Press.

Chen, L., Mitchell, C., Martin, A., Türpe, S., Poller, A., Steffan, J., et al. (2009). Attacking the BitLocker Boot Process. In *Trusted Computing* (Vol. 5471, pp. 183-196): Springer Berlin / Heidelberg.

Chen, L., and Ryan, M. (2009). Offline Dictionary Attack on TCG TPM Weak Authorisation Data, and solution. In *Future of Trust in Computing* (pp. 193-196).

Chen, L., and Ryan, M. (2010). Attack, Solution and Verification for Shared Authorisation Data in TCG TPM. In *Formal Aspects in Security and Trust* (pp. 201-216): Springer.

Chen, X. F. (2009). Formal Analysis and Testing of Trusted Platform Module. *Jisuanji Xuebao/Chinese Journal of Computers, 32*(4), 646-653.

Chevalley, P. (2001). *Applying Mutation Analysis for Object-Oriented Programs Using a Reflective Approach.* Proceedings of the Software Engineering Conference, 2001. APSEC 2001. Eighth Asia-Pacific, 267-270.

Chow, T. S. (1978). Testing software design modeled by finite-state machines. *IEEE Trans. Software Eng., 4*(3), 178-187.

Dahl, O.-J., Dijkstra, E. W., and Hoare, C. A. R. (1972). *Structured programming*: Academic Press Ltd.

Dalal, S. R., Jain, A., Karunanithi, N., Leaton, J., Lott, C. M., Patton, G. C., et al. (1999). *Model-Based Testing in Practice.* Proceedings of the 21st international conference on software engineering, 285-294.

Dasso, A., and Funes, A. (2006). *Verification, Validation and Testing in Software Engineering*. Hershey, United States of America: Idea Group Publishing (an imprint of Idea Group Inc.).

De Nicola, R. (1987). Extensional Equivalences for Transition Systems. *Acta Informatica, 24*(2), 211-237.

De Nicola, R., and Hennessy, M. C. B. (1984). Testing Equivalences for Processes. *Theoretical Computer Science, 34*(1–2), 83-133.

de Vries, R. G., Belinfante, A., and Feenstra, J. (2002). Automated Testing in Practice: The Highway Tolling System. In *Testing of Communicating Systems XIV* (pp. 219-234): Springer.

Delaune, S., Kremer, S., Ryan, M. D., and Steel, G. (2011). A formal Analysis of Authentication in the TPM. In *Formal Aspects of Security and Trust* (pp. 111-125): Springer.

DeMillo, R. A., Lipton, R. J., and Sayward, F. G. (1978). Hints on Test Data Selection: Help for the Practicing Programmer. *Computer, 11*(4), 34-41.

Derezinska, A., and Szustek, A. (2008). *Tool-Supported Advanced Mutation Approach for Verification of C# Programs.* Dependability of Computer Systems, 2008. DepCos-RELCOMEX'08. Third International Conference on, 261-268.

Dijkstra, E. W. (1969). Notes on Structured Programming. In O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare (Eds.), *Structured Programming*. Academic Press, 1-82.

Dolgikh, A. (2013). *Behavior Based Approach for Intrusion Detection Systems*: State University of New York at Binghamton.

Donglai, F., Xinguang, P., and Yuli, Y. (2013). Authentication of the Command TPM_CertifyKey in the Trusted Platform Module. *TELKOMNIKA Indonesian Journal of Electrical Engineering, 11*(2), 855-863.

Dorofeeva, R., El-Fakih, K., Maag, S., Cavalli, A. R., and Yevtushenko, N. (2010). FSM-Based Conformance Testing Methods: A Survey Annotated with Experimental Evaluation. *Information and Software Technology, 52*(12), 1286-1297.

Douglas, K., and Douglas, S. (2003). *PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgresSQL Databases*. SAMS publishing.

Doyle, E. M. (1997). *Automated Security Analysis of Cryptographic Protocols Using Coloured Petri Net Specifications*. Master's Thesis. Queen’s University, Canada.

Du Bousquet, L., Ramangalahy, S., Simon, S., Viho, C., Belinfante, A., and de Vries, R. G. (2000). Formal Test Automation: The Conference Protocol with TGV/Torx. In *Testing of Communicating Systems* (pp. 221-228): Springer.

Eclipse. (2008). GEF (Graphical Editing Framework). Retrieved January, 2015, from https://eclipse.org/gef/

Everse, W. M. (2009). *Modelling and Verification of a Shortest Path Tree Protocol for Wireless Sensor Networks: Towards a Platform for Formal Verification Experiments*. Master’s thesis, University of Twente.

Fan, Z., Guoqing, W., Min, J., and Xiaoli, L. (2008). *Concerning about Trust of platform Hardware,* Wuhan, China, 852-856.

FU, L., WANG, D., and KUANG, J. (2011). Research on Conformance Testing for Trust Chain of Trusted Computing Platform Based on BUIO Sequences. *Journal of Computational Information Systems, 7*(11), 4153-4160.

Fujiwara, S., Khendek, F., Amalou, M., and Ghedamsi, A. (1991). Test Selection Based on Finite State Models. *IEEE Transactions on Software Engineering, 17*(6), 591-603.

Gaudel, M. C. (1995). Testing Can Be Formal, Too. In *TAPSOFT'95: Theory and Practice of Software Development* (pp. 82-96): Springer.

Gehlot, V., and Nigro, C. (2010, 5-8 Dec. 2010). *An Introduction to Systems Modeling and Simulation with Colored Petri Nets.* Proceedings of the 2010 Winter Simulation Conference (WSC), 104-118.

Gollmann, D. (2006). Why Trust is Bad for Security. *Electronic Notes in Theoretical Computer Science, 157*(3), 3-9.

Goodenough, J. B., and Gerhart, S. L. (1975). Toward a Theory of Test Data Selection. *IEEE Transactions on Software Engineering,* (2), 156-173.

Gu, L., Guo, Y., Yang, Y., Bao, F., and Mei, H. (2011). Modeling TCG-based secure Systems with Colored Petri Nets, *Lecture Notes in Computer Science* *(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6802 LNCS, pp. 67-86). Beijing.

Gunupudi, V. (2008). *Exploring Trusted Platform Module Capabilities: A Theoretical and Experimental Study.* Ph.D. Dissertation. University of North Texas, USA.

Gürgens, S., Rudolph, C., Scheuermann, D., Atts, M., and Plaga, R. (2008). Security Evaluation of Scenarios Based on the TCG's TPM Specification. In *Computer Security – ESORICS 2007* (pp. 438-453).

Hamlet, R. G. (1977). Testing Programs with the Aid of a Compiler. *IEEE Transactions on Software Engineering, SE-3*(4), 279-290.

Han, Y. (2001). *Automated Security Analysis of Internet Protocols Using Coloured Petri Nets*. Master's Thesis, Queen's University, Canada.

Hardjono, T. (2008). Strengthening Enterprise Applications Using Trusted Platform Modules. *Network Security, 2008*(6), 15-18.

Hassine, J. (2013). Design and Classification of Mutation Operators for Abstract State Machines. *International Journal On Advances in Software, 6*(1 and 2), 80-91.

He, F., Len, J., Zhang, H., and Tan, Y. (2008a). *Evolutionary Testing of Trusted Computing Supporting Software Based on Genetic Algorithms*. Proceedings of the IEEE International Symposium on Knowledge Acquisition and Modeling KAM'08, 713-717.

He, F., Zhang, H., and Tang, M. (2008b). *A Test Method of Trusted Computing Supporting Software*. Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008, Zhang Jia Jie, Hunan, China, 2330-2334.

He, F., Zhang, H., Wang, H., Xu, M., and Yan, F. (2010). *Chain of Trust Testing Based on Model Checking.* Proceedings of the Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on IEEE, 273-276.

Helmer, G., Wong, J., Slagell, M., Honavar, V., Miller, L., Wang, Y., et al. (2007). Software Fault Tree and Coloured Petri Net–Based Specification, Design and Implementation of Agent-Based Intrusion Detection Systems. *International Journal of Information and Computer Security, 1*(1), 109-142.

Hierons, R. M., Bogdanov, K., Bowen, J. P., Cleaveland, R., Derrick, J., Dick, J., et al. (2009). Using Formal Specifications to Support Testing. *ACM Computing Surveys (CSUR), 41*(2), 9.

Hoeve, T. (2012). *Model Based Testing of A PLC Based Interlocking System*. Master’s thesis, University of Twente.

Holzmann, G. J. (1997). The Model Checker SPIN. *IEEE Transactions on software engineering, 23*(5), 279-295.

Huanguo, Z., Jie, L., Fei, Y., Mingdi, X., Fan, H., and Jing, Z. (2008, 14-17 Oct. 2008). *A Practical Solution to Trusted Computing Platform Testing.* Proceedings of the Trusted Infrastructure Technologies Conference, 2008. APTC '08. Third Asia-Pacific, 79-87.

ISO. (2009). ISO/IEC 11889-1:2009 Information technology -- Trusted Platform Module. from http://www.iso.org/iso/catalogue_detail.htm?csnumber=50970

Jackson, D. (2004). Alloy 3.0 Reference Manual. *Software Design Group*.

Jackson, W. (2010). Black Hat: Engineer Cracks 'Secure' TPM Chip, from http://redmondmag.com/articles/2010/02/03/black-hat-engineer-cracks-tpm-chip.aspx

Jansen, L., Zu Horste, M. M., and Schnieder, E. (1998). *Technical Issues in Modelling the European Train Control System (ETCS) Using Coloured Petri Nets and The Design/CPN Tools*. Proceedings of the 1st CPN Workshop, DAIMI PB 532,103–115.

Januzaj, V., and Kugele21, S. (2009). *Model Analysis Via A Translation Schema to Coloured Petri Nets.* Paper presented at the International Workshop on Petri Nets and Software Engineering, 273-292.

Jensen, K., Kristensen, L., and Wells, L. (2007). Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer, 9*(3-4), 213-254.

Jensen, K., and Kristensen, L. M. (2009a). *Coloured Petri Nets Modelling and Validation of Concurrent Systems*: Springer Berlin Heidelberg.

Jensen, K., and Kristensen, L. M. (2009b). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*: Springer.

Jensen, K., Kristensen, L. M., and Mailund, T. (2012). The Sweep-Line State Space Exploration Method. *Theoretical Computer Science, 429*(0), 169-179.

Jüri, V., Andres, K., Marko, K., Maili, M., and Kullo, R. (2011). Reactive Testing of Nondeterministic Systems by Test Purpose-Directed Tester. In *Model-Based Testing for Embedded Systems* (pp. 425-452): CRC Press.

Kalaji, A., Hierons, R. M., and Swift, S. (2009). *Generating Feasible Transition Paths for Testing From An Extended Finite State Machine (EFSM).* Proceedings of the International Conference on Software Testing Verification and Validation, 2009. ICST'09., 230-239.

Kallath, D. (2005). Trust in Trusted Computing - The End of Security As We Know it. *Computer Fraud and Security, 2005*(12), 4-7.

Kalman, J. A. (2001). *Automated Reasoning with Otter* (Vol. 3): Rinton Press Princeton NJ.

Keller, R. M. (1976). Formal Verification of Parallel Programs. *Communications of the ACM, 19*(7), 371-384.

Kinney, S. (2006). *Trusted Platform Module Basics Using TPM in Embedded Systems*: Newnes.

Kleinpenning, R. (2012). *Is Javacardsign Correct and Secure?.* Bachelor Thesis. Radboud University.

Kristensen, L. M., Mechlenborg, P., Zhang, L., Mitchell, B., and Gallasch, G. E. (2008). Model-based Development of a Course of Action Scheduling Tool. *International Journal on Software Tools for Technology Transfer, 10*(1), 5-14.

Kristensen, L. M., and Simonsen, K. I. F. (2013). Applications Of Coloured Petri Nets for Functional Validation of Protocol Designs. In *Transactions on Petri Nets and Other Models of Concurrency VII* (pp. 56-115): Springer.

Kristensen, L. M., and Westergaard, M. (2007). *The Ascoveco State Space Analysis Platform: Next Generation Tool Support for State Space Analysis.* Proceedings of the 8th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Vol. 584 of DAIMI-PB, 1–6.

KUANG, L. F. D. W. J. (2011). Conformance Testing for Trust Chain of Trusted Computing Platform Based on Finite State Machine. *Journal of Computational Information Systems, Vol. 7 (8)*, 2717- 2724.

Kull, A., Raiend, K., Vain, J., and Kääramees, M. (2009). *Case Study-Based Performance Evaluation of Reactive Planning Tester.* Proceedings of the Model-based Testing in Practice: 2nd Workshop on Model-based Testing in

Practice (MoTiP 2009), Enschede, The Netherlands. CTIT Workshop Proceedings Series WP09-08, 87-96.

Kursawe, K., Schellekens, D., and Preneel, B. (2005). *Analyzing Trusted Platform Communication*. Proceedings of the ECRYPT Workshop, CRASH – CRyptographic Advances in Secure Hardware.

Kurt, J. (1997). Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. *EATCS Monographs on Theoretical Computer Science. 2nd edition, Berlin: Springer-Verlag*.

Larsen, K. G., Mikucionis, M., and Nielsen, B. (2005). Online Testing of Real-Time Systems Using Uppaal. In: Grabowski, J., Nielsen, B. (eds.) FATES 2004. LNCS, vol. 3395, 79–94. Springer, Heidelberg.

Le, D., Alipour, M. A., Gopinath, R., and Groce, A. (2014). *Mutation Testing of Functional Programming Languages*. Technical report, Oregon State University.

Lee, D., and Yannakakis, M. (1996). *Principles and Methods of Testing Finite State Machines-A Survey*. Proceedings of the IEEE, Vol. 84, (8), 1090-1123.

LI, H., FENG, D., and CHEN, X. (2009a). Compliant Testing Method of Trusted Cryptography Module [J]. *Journal of Wuhan University (Natural Science Edition), 1,* 008.

Li, H., Hu, H., and Chen, X.-F. (2009b). Research on Compliant Testing Method of Trusted Cryptography Module. *Jisuanji Xuebao/Chinese Journal of Computers, 32*(4), 654-663.

Li, J.-h., Dai, G.-x., and Li, H.-h. (2009c). *Mutation Analysis for Testing Finite State Machines*. Proceedings of the 2nd International Symposium on Electronic Commerce and Security, ISECS'09. 620-624.

Lin, A. H. (2005). *Automated Analysis of Security APIs.* Master's thesis. Massachusetts Institute of Technology.

Lipton, R. (1971). *Fault Diagnosis of Computer Programs*. Student Report, Carnegie Mellon University.

Lochau, M., Peldszus, S., Kowal, M., and Schaefer, I. (2014). Model-Based Testing. In M. Bernardo, F. Damiani, R. Hähnle, E. Johnsen and I. Schaefer (Eds.), *Formal Methods for Executable Software Models* (Vol. 8483, pp. 310-342): Springer International Publishing.

Maghraby, A. O. (2013). *Bridging The Specification Protocol Gap in Argumentation*. Ph.D. Thesis. University of Edinburgh.

Malipatlolla, S. D. K. (2013). Sustainable Trusted Computing: A Novel Approach for a Flexible and Secure Update of Cryptographic Engines on a Trusted Platform Module. Ph.D. Thesis. TU Darmstadt University.

Mark Utting, B. L. (2006). *Practical Model-Based Testing: A Tools Approach*. San Francisco: Morgan Kaufmann Publishers Inc.

Mason, S. (2005). Trusting Your Computer to Be Trusted. *Computer Fraud and Security, 2005*(1), 7-11.

Matteo, L. (2011). *Performance Analysis of Airborne Networks Using Colored Petri Nets*. Master's thesis. Villanova University.

Mayes, K. E. (Ed.). (2008). *Smart Cards, Tokens, Security and Applications*. New York: Springer Science & Business Media, LLC.

McCune, J. (2009). *Reducing the Trusted Computing Base for Applications on Commodity Systems.* Ph.D. Thesis. Carnegie Mellon University.

Mihail, M., and Papadimitriou, C. H. (1994). *On The Random Walk Method for Protocol Testing.* Proceedings of the Computer aided verification, 132-141.

Milner, R. (1997). *The Definition of Standard ML (Revised)*. USA: MIT press.

Mitchell, C. (Ed.). (2005). *Trusted Computing* (First ed. Vol. 6). London, United Kingdom: Institution of Engineering and Technology

Mostowski, W., Poll, E., Schmaltz, J., Tretmans, J., and Schreur, R. W. (2009). Model-Based Testing of Electronic Passports. In *Formal Methods for Industrial Critical Systems* (pp. 207-209): Springer.

Müller, T. (Ed.). (2008). *Trusted Computing Systeme*. Berlin Heidelberg: Springer-Verlag.

Naik, K., and Tripathy, P. (Eds.). (2008). *Software Testing and Quality Assurance: Theory and Practice*. USA: John Wiley & Sons.

Neto, A. D., Subramanyan, R., Vieira, M., and Travassos, G. H. (2007). *Characterization of Model-Based Software Testing Approaches. Technical ReportES-713/07, PESC-COPPE/UFRJ. Available at http://www.cos.ufrj.br/uploadfiles/1188491168.pdf.*

NIAP. (2005). *Common Criteria Certificate for Atmel AT97SC3201 TPM Chip*. Retrieved January, 2012, from http://www.niap-ccevs.org/st/st_vid3005-ci.pdf

Nie, C. (2007). *Dynamic Root of Trust in Trusted Computing.* TKK T-110.5290 Seminar on Network Security, Helsinki University of Technology.

Nogueira, S., Sampaio, A., and Mota, A. (2014). Test Generation from State Based Use Case Models. *Formal Aspects of Computing, 26*(3), 441-490.

Ochsenschläger, P., Repp, J., Rieke, R., and Nitsche, U. (1998). The SH-Verification Tool—Abstraction-Based Verification of Co-operating Systems. *Formal Aspects of Computing, 10*(4), 381-404.

Pearson, S., Mont, M. C., and Crane, S. (2005). *Analysis of Trust Properties and Related Impact of Trusted Platforms*. Bristol, HP.

Petrenko, A., Boroday, S., and Groz, R. (2004). Confirming Configurations in EFSM Testing. *IEEE Transactions on Software Engineering*, *30*(1), 29-42.

Petrenko, A., and Yevtushenko, N. (2005). Testing from Partial Deterministic FSM Specifications. *IEEE Transactions on Computers, 54*(9), 1154-1165.

Petri, C. A. (1962). *Kommunikation mit Automaten*. PhD Thesis. University of Bonn.

Petri, C. A. (1966). Communication with Automata. Technical Report RADC-TR-65-377, Vol. 1, Griffis AFB, New York.

Pinto Ferraz Fabbri, S. C., Delamaro, M. E., Maldonado, J. C., and Masiero, P. C. (1994). *Mutation Analysis Testing for Finite State Machines.* Proceedings of the 5th International Symposium on Software Reliability Engineering. 220-229.

Pretschner, A., and Philipps, J. (2005). 10 Methodological Issues in Model-Based Testing. In *Model-Based Testing of Reactive Systems* (pp. 281-291): Springer.

Rusu, V. (2003). Combining formal verification and conformance testing for validating reactive systems. *Software Testing Verification and Reliability, 13*(3), 157-180.

Sadeghi, A. R. (2006). Challenges for Trusted Computing. *Lecture Notes in Computer Science*, 4249, 414.

Sadeghi, A. R. (2008). Trusted Computing -Special Aspects and Challenges. In SOFSEM 2008: *Theory and Practice of Computer Science* (pp. 98-117): Springer Berlin Heidelberg.

Sadeghi, A. R. (2009a). Lecture “Trusted Computing” Chapter 1: Introduction to Trusted Computing. Retrieved December, 2009, from http://www.ei.rub.de/studierende/lehrveranstaltungen/231/

Sadeghi, A. R. (2009b). Lecture “Trusted Computing” Chapter 4: Advanced Trusted Computing Topics. Retrieved December, 2009, from http://www.ei.rub.de/studierende/lehrveranstaltungen/231/

Sadeghi, A. R., Marcel, S., Christian, S., Christian, W., and Marcel, W. (2006a). *TCG Inside?: A Note on TPM Specification Compliance*. Proceedings of the first ACM workshop on Scalable trusted computing.

Sadeghi, A. R., Marcel, S., Christian, S., and Marcel, W. (2006b). TCG Inside? A Note on TPM Specification Compliance. from http://www.sirrix.com/download/TPMcompliance.pdf

Saifan, A., and Dingel, J. (2010). A Survey of Using Model-Based Testing to Improve Quality Attributes in Distributed Systems. In *Advanced Techniques in Computing Sciences and Software Engineering* (pp. 283-288): Springer.

Seifi, Y. (2014). *Formal Analysis of Security Properties in Trusted Computing Protocols*. Ph.D. Thesis. Queensland University of Technology.

Seifi, Y., Suriadi, S., Foo, E., and Boyd, C. (2011). *Analysis of Object-Specific Authorization Protocol (OSAP) Using Coloured Petri Nets - Version 1.0*. Unpublished Report. Queensland University of Technology.

Seifi, Y., Suriadi, S., Foo, E., and Boyd, C. (2012). *Analysis of Object-Specific Authorization Protocol (OSAP) Using Coloured Petri Nets*. Proceedings of the Tenth Australasian Information Security Conference - Volume 125.

Seifi, Y., Suriadi, S., Foo, E., and Boyd, C. (2014). Security Properties Analysis in A TPM–Based Protocol. *International Journal of Security and Networks*, *9*(2), 85-103.

Shafique, M., and Labiche, Y. (May 2010). *A Systematic Review of Model Based Testing Tool Support*. Technical Report SCE-10-04, Carleton University.

Shannon, R. E. (1975). *Systems Simulation: The Art and Science* (Vol. 1). Englewood Cliffs, NJ.: Prentice-Hall.

Sijtema, M., Belinfante, A., Stoelinga, M., and Marinelli, L. (2014). Experiences with Formal Engineering: Model-Based Specification, Implementation and Testing of a Software Bus at Neopost. *Science of Computer Programming*, *80*, 188-209.

Sloane, E., Way, T., Gehlot, V., Beck, R., Solderitch, J., and Dziembowski, E. (2007, 9-13 April 2007). *A Hybrid Approach to Modeling SOA Systems of Systems*

Using CPN and MESA/Extend. Proceedings of the Systems Conference, 2007 1st Annual IEEE, 1-7.

Song, W. J. (2010). *Protocol Validation for Mobility Management in Airborne Networks.* Master's Thesis. University of Colorado at Boulder, Ann Arbor.

Stavroulakis, P., Stamp, M., Lioy, A., and Ramunno, G. (2010). Trusted Computing. In *Handbook of Information and Communication Security* (pp. 697-717): Springer Berlin Heidelberg.

Tanenbaum, A. S. (2003). Computer Networks. (4th ed.). Englewood Cliffs, NJ.: Prentice Hall.

Tarnovsky, C. (2010). *Deconstructing a `Secure' Processor.* Proceedings of the Black Hat DC 2010. Retrieved January, 2011, from https://www.blackhat.com/presentations/bh-dc-10/Tarnovsky Chris/BlackHat-DC-2010-Tarnovsky-DASP-slides.pdf

TCG. (2007). Trusted Computing Group. TCG Specification Architecture Overview, Version 1.4. from http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14/

TCG. (2008). Trusted Computing Group. Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Version: 1.1; July 10, 2008.

TCG. (2009). Trusted Computing Group-about TCG. Retrieved September, 2009, from http://www.trustedcomputinggroup.org/about_tcg

TCG. (2011a). Trusted Computing Group. Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Revision 116 Version: 1.2; 18 May, 2011.

TCG. (2011b). Trusted Computing Group. TPM Main Part 1 Design Principles Specification, Level 2, Version 1.2, Revision 116

TCG. (2011c). Trusted Computing Group. TPM Main Part 2 TPM Structures Specification, Level 2, Version 1.2, Revision 116.

TCG. (2011d). Trusted Computing Group. TPM Main Specification, Version 1.2 Level 2 Revision 116. from http://www.trustedcomputinggroup.org/resources/tpm_main_specification

TCG. (2012). Trusted Computing Group (TCG) Timeline October 2012. from https://www.trustedcomputinggroup.org/resources/tcg_timeline

TCG. (2013). Trusted Computing Group. Trusted Platform Module Library Specification, Family "2.0". from http://www.trustedcomputinggroup.org/resources/tpm_library_specification

TCG. (2014). Trusted Computing Group. Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Revision 116 Version: 1.3; 14 July, 2014.

TCPA. (2002a). The Evolution of the Trusted PC. Retrieved 13th September, 2009, from http://www.silicon-trust.com/trends/comp_tcpa.html

TCPA. (2002b). Trusted Computing Platform Alliance, Main Specification, Version 1.1b. Retrieved 13th September, 2009 from http://www.trustedcomputinggroup.org/files/resource_files/64795356-1D09-3519-ADAB12F595B5FCDF/TCPA_Main_TCG_Architecture_v1_1b.pdf

TCPA. (2002c). Trusted Computing Platform Alliance,Trusted Platform Module Protection Profile, Version 1.9.7, July 1, 2002. Retrieved 13th September, 2009 from http://www.commoncriteriaportal.org/files/ppfiles/

Timmer, M., Brinksma, H., and Stoelinga, M. (2011). Model-based testing. In M. Broy, C. Leuxner, and C. A. R. Hoare (Eds.). *Software and Systems Safety: Specification and Verification*, volume 30 of *NATO Science for Peace and Security Series D: Information and Communication Security*, (pp. 1-32): IOS Press.

Tóth, G., Kőszegi, G., and Hornák, Z. (2008). *Case Study: Automated Security Testing on The Trusted Computing Platform*. Proceedings of the 1st European workshop on system security. Glasgow, United kingdom: ACM, 35-39.

Tretmans, J. (1996). Test Generation with Inputs, Outputs and Repetitive Quiescence. *Software---Concepts and Tools*. 17(3), 103–120.

Tretmans, J. (2002). *Testing Techniques*. Lecture notes, University of Twente, The Netherlands, 1, 1.

Tretmans, J. (2008). Model Based Testing with Labelled Transition Systems. In *Formal methods and testing* (pp. 1-38): Springer.

Tretmans, J. (2011). Model-Based Testing and Some Steps Towards Test-Based Modelling. In *Formal Methods for Eternal Networked Software Systems* (pp. 297-326): Springer.

Tretmans, J., and Belinfante, A. (1999). Automatic Testing with Formal Methods. Proceedings of the EuroSTAR'99: 7th European Int. Conference on Software Testing, Analysis & Review, Barcelona, Spain, November 8–12

Tretmans, J., and Brinksma, E. (2002). *Côte De Resyste: Automated Model Based Testing*. Proceedings of the 3rd Workshop on Embedded Systems. 246-255.

Tretmans, J., and Brinksma, E. (2003). *Torx: Automated Model-Based Testing*. Proceedings of the First European Conference on Model-Driven Software Engineering, Imbuss, Germany,31-43.

Tretmans, J., Prester, F., Helle, P., and Schamai, W. (2010). Model-Based Testing 2010: Short Abstracts. *Electronic Notes in Theoretical Computer Science, 264*(3), 85-99.

Twente. (2009). JTorX Tool. Retrieved March, 2014, from https://fmt.ewi.utwente.nl/redmine/projects/jtorx/files

Ullman, J. D. (1998). *Elements of ML Programming: ML97 Edition* (Vol. 2). Englewood Cliffs, NJ.: Prentice-Hall.

Ulrich, K., hn, Marcel, S., Christian, S., and ble. (2007). *Realizing Property-Based Attestation and Sealing with Commonly Available Hard- and Software*. Proceedings of the 2007 ACM workshop on Scalable trusted computing.

Utting, M. (2005). *Position paper: Model-Based Testing*. Proceedings of the Verified Software: Theories, Tools, Experiments (VSTTE) Conference, Vol. 2.

Utting, M., Pretschner, A., and Legeard, B. (2006). *A Taxonomy of Model-Based Testing*. Technical Report 04/2006, The University of Waikato (New Zealand)

Utting, M., Pretschner, A., and Legeard, B. (2012). A Taxonomy of Model-Based Testing Approaches. *Software Testing, Verification and Reliability, 22*(5), 297-312.

van der Bijl, H. (2011). *On Changing Models in Model-Based Testing*. PhD thesis, University of Twente, Enschede.

Vanit-Anunchai, S. (2010). Modelling Railway Interlocking Tables Using Coloured Petri Nets. In D. Clarke and G. Agha (Eds.), *Coordination Models and Languages* (Vol. 6116, pp. 137-151): Springer Berlin Heidelberg.

von Styp, S., Yu, L., and Quiros, G. (2011). *Automatic Test-Case Derivation and Execution in Industrial Control.* Proceedings of the Workshop on Industrial

Automation Tool Integration for Engineering Project Automation (iATPA 2011), 7-12.

Wessel. (2012). *Model Based Testing with F#, C# and JTorX*. Retrieved December, 2014, from http://blog.staalsoft.net/?p=81

West, C. H. (1989). Protocol Validation in Complex Systems. ACM. 19(4), 303-312.

Westergaard, M., and Verbeek, H. M. W. E. (2011a). *CPN Tools*. Retrieved December, 2014, from http://cpntools.org/

Westergaard, M., and Verbeek, H. M. W. E. (2011b). *CPN Tools Download*. Retrieved December, 2014, from http://cpntools.org/download

Wiens, J. (2008). A Tipping Point for TPM?. *InformationWeek* (1193), 39.

Wojtczuk, R., and Rutkowska, J. (2009). *Attacking Intel Trusted Execution Technology*. Proceedings of the Black Hat DC 2009, from http://www.invisiblethingslab.com/resources/bh09dc/Attacking Intel TXT-paper.pdf.

Xiao-Feng, C. (2009). The Formal Analysis and Testing of Trusted Platform Module. *Chinese Journal of Computers, 32*(4), 646-653.

Xu, M.-D., Zhang, H.-G., and Yan, F. (2009a). Testing on Trust Chain of Trusted Computing Platform Based on Labeled Transition System. *Jisuanji Xuebao/Chinese Journal of Computers, 32*(4), 635-645.

Xu, S., Zhang, H., Yan, F., Xu, M., and Li, Z. (2009b). *Security Analysis of OIAP Implementation Based on BAN Logic*. Proceedings of the 1st International Conference on Multimedia Information Networking and Security, MINES 2009, Hubei, 144-148.

Yue, J., and Harman, M. (2011). An Analysis and Survey of the Development of Mutation Testing. *Software Engineering, IEEE Transactions on, 37*(5), 649-678.

yWorks. (2011). *yEd Graph Editor*. Retrieved December, 2014, from http://www.yworks.com/en/downloads.html#yEd

Zhan, J., Zhang, H., Zou, B., and Li, X. (2008). *Research on Automated Testing of the Trusted Platform Model*. Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008, Zhang Jia Jie, Hunan, China: IEEE, 2335-2339.

Zhang, H., Luo, J., Yan, F., Xu, M., He, F., and Zhan, J. (2008). *A Practical Solution to Trusted Computing Platform Testing*. Proceedings of the Trusted

Infrastructure Technologies Conference, APTC'08, Wuhan, Hubei, China: IEEE 79-87.

Zhang, H., Yan, F., Fu, J., Xu, M., Yang, Y., He, F., et al. (2010). Research on Theory and Key Technology of Trusted Computing Platform Security Testing and Evaluation. *Science China Information Sciences*. *53*(3), 434-453.

Zhao, B., and Zhang, H. (2005). *Implementation of the Trusted Computing in Commercial Cryptogram Based on Embedded System*. Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China: IEEE (Vol. 2), 1296-1298

Zhao, X., Zhang, Y., Zheng, W., Tang, T., and Mu, R. (2010). *Modelling and Design of The Formal Approach for Generating Test Sequences of ETCS Level 2 Based on The CPN.* Proceedings of the WIT Transactions on the Built Environment, Beijing, 723-734.

Zhu, L., Tong, W., and Cheng, B. (2011). CPN Tools' Application in Verification of Parallel Programs. In *Information Computing and Applications* (pp. 137-143): Springer.

Zurowska, K., and Deters, R. (2007). *Overcoming Failures in Composite Web Services by Analysing Colored Petri Nets.* Proceedings of the Eighth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, 87.