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, 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 menulis menjadi 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

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td></td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td></td>
<td>iv</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>ABSTRAK</td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>xii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>xv</td>
</tr>
<tr>
<td>LIST OF APPENDICES</td>
<td></td>
<td>xxi</td>
</tr>
</tbody>
</table>

1 INTRODUCTION

1.1 Preamble                   | 1                                           |
1.2 Background of the Problem  | 2                                           |
1.3 Statement of the Problem   | 8                                           |
1.4 Aim of the Study           | 11                                          |
1.5 Objectives of the Study    | 12                                          |
1.6 Scope of the Study         | 12                                          |
1.7 Significance of the Study  | 13                                          |
1.8 Organization of the Thesis | 14                                          |

2 LITERATURE REVIEW

2.1 Introduction               | 16                                          |
2.2 Trusted Computing           | 17                                          |
2.3 TPM Fundamentals           | 19                                          |
     2.3.1 Fundamental Features of Trusted Platform | 20 |
     2.3.2 TPM Components         | 21                                          |
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3</td>
<td>Basic TPM Security Services</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>2.3.3.1 Roots of Trust</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>2.3.3.2 Boot Process</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>2.3.3.3 Secure Storage</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>2.3.3.4 Attestation</td>
<td>30</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Types of TPM Keys</td>
<td>30</td>
</tr>
<tr>
<td>2.3.5</td>
<td>TPM Ownership</td>
<td>33</td>
</tr>
<tr>
<td>2.3.6</td>
<td>TPM Operational Modes</td>
<td>34</td>
</tr>
<tr>
<td>2.3.7</td>
<td>TPM Commands</td>
<td>36</td>
</tr>
<tr>
<td>2.3.8</td>
<td>Object Independent Authorisation Protocol (OIAP)</td>
<td>39</td>
</tr>
<tr>
<td>2.3.9</td>
<td>Object Specific Authorisation Protocol (OSAP)</td>
<td>39</td>
</tr>
<tr>
<td>2.4</td>
<td>Testing</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>2.4.1 Formal Testing</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>2.4.2 Model-Based Testing</td>
<td>46</td>
</tr>
<tr>
<td>2.5</td>
<td>Testing the Trusted Platform Module</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>2.5.1 Informal TPM Testing</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>2.5.2 FSM-Based TPM Testing</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>2.5.3 EFSM-Based TPM Testing</td>
<td>62</td>
</tr>
<tr>
<td>2.6</td>
<td>Formal Testing Based on Labelled Transition Systems (LTS)</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>2.6.1 Input-Output Conformance (IOCO) Theory</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>2.6.1.1 Formal Preliminaries for IOCO</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>2.6.2 IOCO-Based Tools</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>2.6.3 LTS-Based TPM Testing</td>
<td>74</td>
</tr>
<tr>
<td>2.7</td>
<td>Attacks and Security Analysis on the Trusted Platform Module</td>
<td>78</td>
</tr>
<tr>
<td>2.8</td>
<td>Common Criteria for Information Technology Security Evaluation</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>2.8.1 Security Evaluation of TPM in CC and Academia</td>
<td>82</td>
</tr>
<tr>
<td>2.9</td>
<td>Simulation</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>2.9.1 Coloured Petri Nets (CPN)</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>2.9.1.1 CPN Model Components</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>2.9.1.2 Formal Preliminaries for CPN</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>2.9.1.3 CPN Model Verification</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>2.9.2 CPN in Trusted Computing</td>
<td>94</td>
</tr>
</tbody>
</table>
3 RESEARCH METHODOLOGY
3.1 Introduction 98
3.2 Research Development Phases 98
   3.2.1 TPM Testing Framework 101
   3.2.2 Modelling Preparation 106
   3.2.3 Simulation Level 107
   3.2.4 Concrete-TPM Level 107
3.3 Research Environment 110
3.4 Research Validation and Evaluation 111
3.5 Summary 112

4 THE PROPOSED TPM TESTING FRAMEWORK
4.1 Introduction 113
4.2 TPM Testing Framework (TPM-TF) 113
4.3 The Main Items of the TPM-TF Framework 117
4.4 Case Study 125
4.5 Comparison between the TPM-TF Framework and the other Frameworks 127
4.6 Constructing Command Matrices for the TPM Specifications 131
4.7 Summary 134

5 DEVELOPING THE PROPOSED TPM TESTING FRAMEWORK
5.1 Introduction 135
5.2 Modelling the TPM Specifications Using Labelled Transition System (LTS) for IOCO Testing Theory 136
5.3 TPM Model-Based Testing based on IOCO 139
   5.3.1 Test Setup 142
   5.3.2 Adapter and TPM-Behaviour Emulator Algorithms 143
   5.3.3 Test Run Execution and Reporting 148
5.3.4 Verification of the LTS Model

5.4 Modelling the TPM Specifications for the
TPM-Function Test
  5.4.1 TPM-FT-LTS Modelling Preparation
  5.4.2 TPM-FT-LTS Model
  5.4.3 Implementing the TPM-Behaviour Emulator and the
      Adapter for the TPM-Function Test
  5.4.4 Validation and Verification of the
      TPM-FT-LTS Model
  5.4.5 Test Run Execution and Results for the
      TPM-Function Test

5.5 Modelling the TPM Specifications for the
TPM-Command Test
  5.5.1 TPM-CT-LTS Modelling Preparation
  5.5.2 TPM-CT-LTS Model
  5.5.3 Implementing the TPM-Behaviour Emulator and the
      Adapter for the TPM-Command Test
  5.5.4 Validation and Verification of the
      TPM-CT-LTS Model
  5.5.5 Test Run Execution and Results for the
      TPM-Command Test

5.6 Modelling the TPM Specifications for the
TPM-Security Test
  5.6.1 TPM-SecT-LTS Modelling Preparation
  5.6.2 TPM-SecT-LTS Model
  5.6.3 Implementing the TPM-Behaviour Emulator and the
      Adapter for the TPM-Security Test
  5.6.4 Validation and Verification of the
      TPM-SecT-LTS Model
  5.6.5 Test Run Execution and Results for the
      TPM-Security Test

5.7 Modelling the TPM Specifications Using CPN

5.8 Modelling the Eight Operational Modes (EOM) of TPM
    Using CPN
5.8.1 Modelling Preparation 234
5.8.2 Colour Set Definition 248
5.8.3 TPM-EOM CPN Model 250

5.9 Validation and Verification of the TPM-EOM CPN Model 255
5.9.1 Simulation Analysis 255
5.9.2 Basic State Space Analysis 258
5.9.3 Sweep-Line Analysis 269
  5.9.3.1 Deadlock Analysis based on the Sweep-Line Method 272
  5.9.3.2 Verifying the Predicted Behaviour based on the Sweep-Line Method 274

5.10 Summary 276

6 CONCLUSION 278
6.1 Introduction 278
6.2 Research Contributions 278
6.3 Future Work 284
6.4 Conclusion 287

REFERENCES 290
Appendices A - C 308 - 336
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>PCR Standard Usage</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>TPM Operational Modes</td>
<td>35</td>
</tr>
<tr>
<td>2.3</td>
<td>Mapping the FSM models used in the FSM-based TPM testing framework with the TPM operational modes</td>
<td>59</td>
</tr>
<tr>
<td>2.4</td>
<td>Attacks and security flaws on TPM</td>
<td>78</td>
</tr>
<tr>
<td>2.5</td>
<td>Research gap in the existing TPM testing frameworks</td>
<td>95</td>
</tr>
<tr>
<td>2.6</td>
<td>Main features of a proposed enhanced testing framework for TPM</td>
<td>96</td>
</tr>
<tr>
<td>4.1</td>
<td>Case study of applying TPM-TF testing framework on TPM specifications ver. 1.2 rev. 116</td>
<td>126</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison between the TPM-TF framework and the other TPM testing frameworks</td>
<td>129</td>
</tr>
<tr>
<td>4.3</td>
<td>Determination of the operational modes</td>
<td>132</td>
</tr>
<tr>
<td>4.4</td>
<td>Matrix of operational modes and return codes for TPM_SetOwnerInstall command</td>
<td>133</td>
</tr>
<tr>
<td>4.5</td>
<td>Matrix of the input-message operands and the return codes for TPM_SetOwnerInstall command</td>
<td>133</td>
</tr>
<tr>
<td>4.6</td>
<td>Other parameters matrix for the TPM_SetOwnerInstall command</td>
<td>134</td>
</tr>
<tr>
<td>5.1</td>
<td>List of labels for the TPM commands which have different configurations</td>
<td>140</td>
</tr>
<tr>
<td>5.2</td>
<td>List of the mutation operators used in the mutation analysis</td>
<td>153</td>
</tr>
<tr>
<td>5.3</td>
<td>Transition table of the TPM-FT-LTS model</td>
<td>162</td>
</tr>
<tr>
<td>5.4</td>
<td>Mutating the TPM-behaviour emulator of the TPM-function test</td>
<td>172</td>
</tr>
</tbody>
</table>
5.5 Results of the mutation analysis applied on the TPM-FT-LTS model 174
5.6 Transition table of the TPM-CT-LTS model 182
5.7 Results of mutation analysis applied on the TPM-CT-LTS model 209
5.8 Transition table of the TPM-SecT-LTS model 217
5.9 Results of mutation analysis applied on the TPM-SecT-LTS model 229
5.10 List of the modified command's names 235
5.11 Main TPM return codes for TPM_TakeOwnership command 236
5.12 TPM return codes in S5 and S7 for TPM_TakeOwnership command 236
5.13 Main TPM return codes for TPM_OwnerClear command 236
5.14 TPM return codes in S1 for TPM_OwnerClear command 237
5.15 Main TPM return codes for TPM_ForceClear command 237
5.16 TPM return codes in S1 and S5 for TPM_ForceClear command 237
5.17 Main TPM return codes for TPM_SetTempDeactivated command with operator AuthData 238
5.18 TPM return codes in S1,S2,S5 and S6 for TPM_SetTempDeactivated command with operator AuthData 238
5.19 Updated values of flags and data fields 239
5.20 Main transition table for the eight operational modes of TPM 240
5.21 Requirements for successful execution 247
5.22 Description of the CPN model places 253
5.23 List of the developed functions in the CPN model 254
5.24 Size of state spaces for different configurations 263
5.25 Description of a terminal (successful termination) marking 267
5.26 Reported statistics for the model analysis using the sweep-line method 273
5.27 Implementation of the property 6.4.3 275
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>List of contributions of this study per section number</td>
<td>279</td>
</tr>
<tr>
<td>6.2</td>
<td>Transition table of chain of trust</td>
<td>286</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Trusted computing platform</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>CC certificate for Atmel AT97SC3201 TPM chip</td>
<td>8</td>
</tr>
<tr>
<td>1.3</td>
<td>Thesis skeleton</td>
<td>14</td>
</tr>
<tr>
<td>2.1</td>
<td>Organisation of the literature review</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Trusted platform</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Mounting TPM chip into platform</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>TPM architecture</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>Chain of trust</td>
<td>25</td>
</tr>
<tr>
<td>2.6</td>
<td>Roots of trust</td>
<td>26</td>
</tr>
<tr>
<td>2.7</td>
<td>Integrity measurement</td>
<td>26</td>
</tr>
<tr>
<td>2.8</td>
<td>Boot process</td>
<td>28</td>
</tr>
<tr>
<td>2.9</td>
<td>Data sealing</td>
<td>29</td>
</tr>
<tr>
<td>2.10</td>
<td>Platform attestation</td>
<td>30</td>
</tr>
<tr>
<td>2.11</td>
<td>Key tree</td>
<td>32</td>
</tr>
<tr>
<td>2.12</td>
<td>TPM operational modes</td>
<td>35</td>
</tr>
<tr>
<td>2.13</td>
<td>Transition of the TPM operational modes</td>
<td>36</td>
</tr>
<tr>
<td>2.14</td>
<td>The structure of the input message of TPM commands</td>
<td>37</td>
</tr>
<tr>
<td>2.15</td>
<td>The structure of the output message of TPM commands</td>
<td>37</td>
</tr>
<tr>
<td>2.16</td>
<td>Establishing OIAP session</td>
<td>39</td>
</tr>
<tr>
<td>2.17</td>
<td>Establishing OSAP session</td>
<td>40</td>
</tr>
<tr>
<td>2.18</td>
<td>General testing process</td>
<td>41</td>
</tr>
<tr>
<td>2.19</td>
<td>Testing types</td>
<td>43</td>
</tr>
<tr>
<td>2.20</td>
<td>Test cases generation in MBT</td>
<td>46</td>
</tr>
<tr>
<td>2.21</td>
<td>On-line test execution</td>
<td>48</td>
</tr>
<tr>
<td>2.22</td>
<td>The process of model-based testing</td>
<td>49</td>
</tr>
<tr>
<td>2.23</td>
<td>TPM black-box testing</td>
<td>53</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.24</td>
<td>The operational levels of the informal TPM testing framework</td>
<td>53</td>
</tr>
<tr>
<td>2.25</td>
<td>Conceptual block diagram for the informal TPM testing framework</td>
<td>54</td>
</tr>
<tr>
<td>2.26</td>
<td>TPM dependency graph</td>
<td>55</td>
</tr>
<tr>
<td>2.27</td>
<td>Activity graph for TPM commands</td>
<td>56</td>
</tr>
<tr>
<td>2.28</td>
<td>TPM return codes for all possible errors related to TPM_Seal command</td>
<td>56</td>
</tr>
<tr>
<td>2.29</td>
<td>Informal TPM testing framework</td>
<td>57</td>
</tr>
<tr>
<td>2.30</td>
<td>FSM model of the TPM operational modes in the FSM-based TPM testing framework</td>
<td>60</td>
</tr>
<tr>
<td>2.31</td>
<td>spec¹ FSM model in the FSM-based TPM testing framework</td>
<td>60</td>
</tr>
<tr>
<td>2.32</td>
<td>The FSM-based TPM testing framework</td>
<td>61</td>
</tr>
<tr>
<td>2.33</td>
<td>Conceptual block diagram for the EFSM-based TCM testing framework</td>
<td>63</td>
</tr>
<tr>
<td>2.34</td>
<td>Dependency graph for the TCM commands in owner management and key management sub-modules</td>
<td>64</td>
</tr>
<tr>
<td>2.35</td>
<td>EFSM model for the TCM commands in owner management and key management sub-modules</td>
<td>65</td>
</tr>
<tr>
<td>2.36</td>
<td>The EFSM-based TCM testing framework</td>
<td>66</td>
</tr>
<tr>
<td>2.37</td>
<td>JTorX tool architecture</td>
<td>73</td>
</tr>
<tr>
<td>2.38</td>
<td>FSM-to-LTS transformation tactic</td>
<td>75</td>
</tr>
<tr>
<td>2.39</td>
<td>Common Criteria evaluation process of TPM</td>
<td>82</td>
</tr>
<tr>
<td>2.40</td>
<td>Threats countermeasures from Common Criteria perspective</td>
<td>84</td>
</tr>
<tr>
<td>2.41</td>
<td>Petri Net example</td>
<td>86</td>
</tr>
<tr>
<td>2.42</td>
<td>CPN model of the simple protocol</td>
<td>89</td>
</tr>
<tr>
<td>2.43</td>
<td>Simulating the simple protocol CPN model</td>
<td>91</td>
</tr>
<tr>
<td>2.44</td>
<td>Partial state space of the simple protocol CPN model</td>
<td>93</td>
</tr>
<tr>
<td>3.1</td>
<td>Research design framework – part 1</td>
<td>104</td>
</tr>
<tr>
<td>3.2</td>
<td>Research design framework – part 2</td>
<td>105</td>
</tr>
<tr>
<td>4.1</td>
<td>TPM testing framework based on IOCO testing theory and CPN theory</td>
<td>114</td>
</tr>
</tbody>
</table>
4.2 Test architecture and SUT for a concrete TPM 123
4.3 Test architecture and SUT for a software TPM emulator 124
4.4 Test architecture and SUT for the TPM-behaviour emulator 124
4.5 Conceptual block diagram of the TPM-TF Framework 127
4.6 TPM command execution model 131
5.1 Transforming from FSM to LTS; (a) FSM; (b) LTS 138
5.2 Message Sequence Diagram of the TPM Model-Based Testing 142
5.3 Common JTorX configuration used in the concrete-TPM operational level 144
5.4 TPM-Behaviour Emulator algorithm 145
5.5 Adapter-Main algorithm 146
5.6 Algorithm of the Adapter-Server (int PortNo) function 146
5.7 Algorithm of the HandleJTorX(TCPClient client) function 147
5.8 Initiation of a test run 148
5.9 Test run example in the TPM-TF framework 149
5.10 Visualising the Test Run Example as a dynamically updated FSM 150
5.11 Message sequence chart of the test run example 150
5.12 Stored traces of the test run example 151
5.13 Mutation analysis algorithm 154
5.14 TPM-FT-LTS model 165
5.15 LTS model of the TPM operational modes 166
5.16 LTS model of the TPM storage functions 167
5.17 LTS model of the TPM capability commands 168
5.18 LTS model of the changing AuthData function 169
5.19 Detecting a mutated sentence in the TPM-behaviour emulator of the TPM-function test 173
5.20 Partial view of the test run showing a detection of a mutant 174
5.21 Successful execution of the test run for the TPM-function test 175
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.22</td>
<td>Partial view of the message sequence diagram of the TPM-function test</td>
</tr>
<tr>
<td>5.23</td>
<td>Visualised test run of the TPM-function test, partial view</td>
</tr>
<tr>
<td>5.24</td>
<td>Partial views of the off-line test cases for two TPM-function test run executions</td>
</tr>
<tr>
<td>5.25</td>
<td>TPM-CT-LTS model parts for the TPM_OwnerSetDisable command</td>
</tr>
<tr>
<td>5.26</td>
<td>TPM-CT-LTS model parts for the TPM_PhysicalDisable command</td>
</tr>
<tr>
<td>5.27</td>
<td>TPM-CT-LTS model parts for the TPM_PhysicalSetDeactivated command</td>
</tr>
<tr>
<td>5.28</td>
<td>TPM-CT-LTS model parts for the TPM_SetOperatorAuth command</td>
</tr>
<tr>
<td>5.29</td>
<td>TPM-CT-LTS model parts for the TPM_SetTempDeactivated command</td>
</tr>
<tr>
<td>5.30</td>
<td>TPM-CT-LTS model part for the TPM_OwnerClear command</td>
</tr>
<tr>
<td>5.31</td>
<td>TPM-CT-LTS model part for the TPM_ForceClear command</td>
</tr>
<tr>
<td>5.32</td>
<td>TPM-CT-LTS model part for the TPM_DisableOwnerClear command</td>
</tr>
<tr>
<td>5.33</td>
<td>TPM-CT-LTS model parts for the TPM_PhysicalEnable command</td>
</tr>
<tr>
<td>5.34</td>
<td>TPM-CT-LTS model part for the TPM_SetOwnerInstall command</td>
</tr>
<tr>
<td>5.35</td>
<td>TPM-CT-LTS model parts for the TPM_TakeOwnership command</td>
</tr>
<tr>
<td>5.36</td>
<td>TPM-CT-LTS model part for the TPM_CreateWrapKey command</td>
</tr>
<tr>
<td>5.37</td>
<td>TPM-CT-LTS model part for the TPM_LoadKey2 command</td>
</tr>
<tr>
<td>5.38</td>
<td>TPM-CT-LTS model part for the TPM_Seal command</td>
</tr>
<tr>
<td>5.39</td>
<td>TPM-CT-LTS model part for the TPM_Sealx command</td>
</tr>
<tr>
<td>5.40</td>
<td>TPM-CT-LTS model part for the TPM_Unseal command</td>
</tr>
</tbody>
</table>
5.41 TPM-CT-LTS model part for the TPM_GetPubKey command
5.42 TPM-CT-LTS model part for the TPM_UnBind command
5.43 Successful execution of the test run for the TPM-command test
5.44 Partial view of the message sequence diagram of the TPM-command test
5.45 Visualised test run of the TPM-command test, partial view
5.46 Partial views of the off-line test cases for two TPM-command test run executions
5.47 Messages of the TPM_CertifyKey command (a) input message (b) output message
5.48 Sequence of commands to certify a public portion of key1 key using the TPM_CertifyKey and key2 key as a certifier
5.49 LTS Model of the security experiment no.1, as a part of the TPM-SecT-LTS model
5.50 LTS model of the security experiment no. 2, as a part of the TPM-SecT-LTS model
5.51 Message sequence chart visualising the steps of the security scenario
5.52 LTS model of the security scenario, as a part of the TPM-SecT-LTS model
5.53 Successful execution of the test run for the TPM-security test
5.54 Partial view of the message sequence diagram of the TPM-security test
5.55 Visualised test run of the TPM-security test, partial view
5.56 Partial views of the off-line test cases for two TPM-security test run executions
5.57 List of the model variables
5.58 User-TPM sequence of messages
5.59 TPM-EOM CPN model
5.60 TPM-EOM CPN model simulation
5.61 Partial simulation report for the TPM-EOM CPN model
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.62</td>
<td>State space report: statistics and integer bounds</td>
</tr>
<tr>
<td>5.63</td>
<td>State space report: Home, liveness, and fairness properties</td>
</tr>
<tr>
<td>5.64</td>
<td>Modified TPM-EOM CPN model with Limit place</td>
</tr>
<tr>
<td>5.65</td>
<td>Number of nodes versus Limit’s values</td>
</tr>
<tr>
<td>5.66</td>
<td>Full state space for Limit=1 configuration</td>
</tr>
<tr>
<td>5.67</td>
<td>Investigating dead markings in the configurations Limit=1 to 10</td>
</tr>
<tr>
<td>5.68</td>
<td>The SML code for the function IsValidTerminal</td>
</tr>
<tr>
<td>5.69</td>
<td>Verifying that all dead markings of the TPM-EOM CPN model (Limit=10) are desirable terminal markings</td>
</tr>
<tr>
<td>5.70</td>
<td>Partial state space of the modified TPM-EOM CPN model, arranged by progress value</td>
</tr>
<tr>
<td>5.71</td>
<td>Progress measure query for the TPM-EOM CPN model</td>
</tr>
<tr>
<td>5.72</td>
<td>JoSEL model checking job for checking deadlocks in TPM-EOM CPN model based on sweep-line method</td>
</tr>
<tr>
<td>5.73</td>
<td>Limit versus time for the basic state space and sweep-line methods in analysing the TPM-EOM CPN model</td>
</tr>
<tr>
<td>5.74</td>
<td>Counter example for the configuration Limit=230</td>
</tr>
<tr>
<td>5.75</td>
<td>JoSEL model checking job for safety properties in the TPM-EOM CPN model based on sweep-line method</td>
</tr>
<tr>
<td>5.76</td>
<td>Implementation of the property “There is always one command only in the execution process”</td>
</tr>
<tr>
<td>6.1</td>
<td>TPM-TF with test-purpose model</td>
</tr>
<tr>
<td>6.2</td>
<td>LTS model for chain of trust of trusted computing platform</td>
</tr>
</tbody>
</table>
# LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Command Matrices for the TPM Specifications</td>
<td>308</td>
</tr>
<tr>
<td>B</td>
<td>Source Codes of TPM-Behaviour Emulators and Adapters</td>
<td>319</td>
</tr>
<tr>
<td>C</td>
<td>SML Source Codes for the Functions of TPM-EOM CPN Model</td>
<td>328</td>
</tr>
</tbody>
</table>
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.

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 non-
compliant 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 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.

![CC Certificate for Atmel AT97SC3201 TPM chip](image)

**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 flodge 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 policies, 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.

![Thesis skeleton](image)

**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


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


Chris/BlackHat-DC-2010-Tarnovsky-DASP-slides.pdf

http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14/


http://www.trustedcomputinggroup.org/resources/tpm_main_specification

https://www.trustedcomputinggroup.org/resources/tpg_timeline


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


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


