

# Contents

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>1. About This Thesis</b>                                            | <b>1</b>  |
| 1.1. Motivation                                                        | 1         |
| 1.2. Methods                                                           | 1         |
| 1.3. Scope                                                             | 2         |
| 1.4. Outline                                                           | 3         |
| <b>2. Physical Attacks and Countermeasures</b>                         | <b>5</b>  |
| 2.1. Introduction                                                      | 5         |
| 2.2. Side-Channel Attacks                                              | 6         |
| 2.2.1. Data Acquisition                                                | 6         |
| 2.2.2. Pre-processing                                                  | 7         |
| 2.2.3. Leakage Hypotheses                                              | 7         |
| 2.2.4. Data Analysis                                                   | 10        |
| 2.3. Side-Channel Countermeasures                                      | 12        |
| 2.3.1. Hiding Countermeasures                                          | 13        |
| 2.3.2. Masking Countermeasures                                         | 15        |
| 2.3.3. Separation of Masking Shares                                    | 17        |
| 2.4. Tool Support for Side-Channel Mitigations                         | 17        |
| <b>3. Models of Computation and Leakage</b>                            | <b>21</b> |
| 3.1. Introduction                                                      | 21        |
| 3.2. Logic Circuits in CPUs                                            | 22        |
| 3.2.1. Combinatorial Circuits and Boolean Algebra                      | 22        |
| 3.2.2. Synchronous Sequential Circuits                                 | 23        |
| 3.3. Finite State Machines                                             | 24        |
| 3.4. Leakage Characteristics of CMOS Circuits                          | 26        |
| 3.4.1. Power Consumption of CMOS Inverters                             | 26        |
| 3.4.2. State Machine Based Leakage Model for Digital Circuits          | 28        |
| 3.5. Conclusion                                                        | 30        |
| <b>4. Side-Channel Emanations of Reduced Instruction Set Computers</b> | <b>31</b> |
| 4.1. The RISC Architecture                                             | 31        |
| 4.1.1. Datapath Implementation                                         | 31        |
| 4.2. Hypothetical Side-Channels of RISC CPUs                           | 32        |
| 4.2.1. Memories: Load and Store Instructions                           | 34        |
| 4.2.2. Reading and Writing the Register File                           | 34        |
| 4.2.3. Additional Leakage Sources                                      | 35        |

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| 4.3. The ARM Architecture . . . . .                                 | 37        |
| 4.3.1. Introduction . . . . .                                       | 37        |
| 4.3.2. ARM Cortex-M0 . . . . .                                      | 37        |
| <b>5. Determination of Side-Channel Characteristics . . . . .</b>   | <b>45</b> |
| 5.1. Measurement Framework . . . . .                                | 45        |
| 5.2. Methods for Side-Channel Characterization . . . . .            | 47        |
| 5.2.1. Leakage Threshold for the Correlation Coefficient . . . . .  | 49        |
| 5.2.2. A Note on the Use of nop-Instructions . . . . .              | 50        |
| 5.3. Side-Channel Characteristics of an ARM Cortex-M0 CPU . . . . . | 51        |
| 5.3.1. Introduction . . . . .                                       | 51        |
| 5.3.2. Memory Access . . . . .                                      | 51        |
| 5.3.3. Register File Access . . . . .                               | 64        |
| 5.4. Modeling Leakage Characteristics . . . . .                     | 72        |
| <b>6. Compilers . . . . .</b>                                       | <b>77</b> |
| 6.1. Introduction . . . . .                                         | 77        |
| 6.2. Compiler Frontend . . . . .                                    | 79        |
| 6.3. Intermediate Representations . . . . .                         | 79        |
| 6.3.1. Static Single Assignment Form . . . . .                      | 80        |
| 6.4. Machine Independent Optimizations . . . . .                    | 81        |
| 6.5. Compiler Backend . . . . .                                     | 82        |
| 6.5.1. Instruction Scheduling . . . . .                             | 82        |
| 6.5.2. Register Allocation . . . . .                                | 83        |
| 6.5.3. Security Concerns . . . . .                                  | 83        |
| 6.6. Generate Binary Machine Code . . . . .                         | 83        |
| <b>7. Leakage Aware Code Generation . . . . .</b>                   | <b>85</b> |
| 7.1. Introduction . . . . .                                         | 85        |
| 7.2. Identifying Critical Pairs of Variables . . . . .              | 85        |
| 7.2.1. Correlation Between Gate Inputs and Outputs . . . . .        | 86        |
| 7.2.2. Simulation Approach . . . . .                                | 88        |
| 7.3. Secure Scheduling . . . . .                                    | 89        |
| 7.4. Assembler Code Generator . . . . .                             | 90        |
| 7.4.1. Static Code Generator . . . . .                              | 91        |
| 7.4.2. Dynamic Code Generator . . . . .                             | 91        |
| 7.5. Secure Probabilistic Choice Operator . . . . .                 | 94        |
| <b>8. Use Case: Masked S-Box . . . . .</b>                          | <b>95</b> |
| 8.1. The PRESENT Block Cipher . . . . .                             | 95        |
| 8.2. Masked Bit-sliced PRESENT S-Box . . . . .                      | 95        |
| 8.2.1. Bit-sliced and Circuit Representations . . . . .             | 95        |
| 8.2.2. Boolean Masking . . . . .                                    | 96        |
| 8.2.3. Conflict Graph . . . . .                                     | 99        |

|                                                                    |            |
|--------------------------------------------------------------------|------------|
| 8.3. Analysis Results . . . . .                                    | 101        |
| 8.3.1. Non-Secure Scheduling with Static Code Generation . . . . . | 102        |
| 8.3.2. Secure Scheduling with Static Code Generation . . . . .     | 102        |
| 8.3.3. Secure Scheduling with Dynamic Code Generation . . . . .    | 104        |
| <b>9. Discussion and Outlook</b>                                   | <b>107</b> |
| <b>Bibliography</b>                                                | <b>113</b> |
| <b>Appendix A. Assembler S-Box Implementation</b>                  | <b>125</b> |