UNIVERSITY OF BIRMINGHAM

# Research on Cryptographic Backdoors

Degree: MSc. Cyber Security

Supervisor: Dr. David Galindo

Bancha Upanan

Student ID: 1604026

School of Computer Science, College of Engineering
& Physical Sciences

September 2016

# Contents

## 0.1 Abstract

should be a succinct and self-standing summary of the basis and achievements of the project. Minimally an abstract does three things: (1) it states the problem that you set out to solve, (2) it describes your solution and method, (3) it states a conclusion about the success of the solution. Be straightforward and factual and avoid vague statements, confusing details and ”hype”. Do not be tempted to use acronyms or jargon to keep within the half-page limit. Consider that search engines, librarians and non-computer scientists wishing to classify your Report rely on the abstract. You may if you wish provide a short list of keywords (2-6 is reasonable) at the end of the abstract.

## 0.2 Acknowledgements

## 0.3 Introduction

In this section, you should describe the problem that you set out to solve with the project. An introduction might, for example, begin by stating, ”The aim of the work described in this Report was to provide a software tool with which people can arrange meetings.” Avoid starting a Report with an irrelevant history of information technology. For example, the following would not be a good introductory sentence, ”Since Bill Gates launched Outlook people have been using technology to arrange meetings.”

Explain whatever background the reader will need in order to understand the problem. The background might refer to previous work in the academic literature that provides evidence that the problem is a real and significant problem worth solving. Include a clear and detailed statement of the project aims and provide an overview of the structure of the solution.

Conventionally, the last part of the introduction outlines the remainder of the Report, explaining what comes in each section.

## 0.4 Further Background Material

It is often appropriate to provide more information than was given in your Introduction. Try to limit yourself just to what the reader needs to know to understand the solution that you have developed in your project. Put your work in the context of related existing work, commercial products, and research papers (if relevant).

## 0.5 Analysis and Specification

How you analysed the problem, including user requirements. Give an appropriate specification of the solution. For projects involving software development you may include a formal Software Requirements Specification in an Appendix.

## 0.6 Design

If it is a software development project then give a high-level account of the structure of your software and how it works. What algorithms does it use? How do these compare with alternatives? What were the main design decisions you took, and their justifications?

## 0.7 Implementation and Testing

A detailed account of the implementation and testing of your software. Explain the conceptual structure of the algorithms. Also explain what data structures you used, and how the algorithms were implemented. What implementation decisions did you take, and why? There is no need to list every little function and procedure and explain its working in elaborate detail; use your judgement on what is appropriate to include. Explain your testing strategy and provide convincing examples of tests. In project involving software development it is usual to relate the tests to your Software Requirements Specification.

## 0.8 User Interface

If the interface forms a substantial part of your project (e.g. an HCI project), you will wish to include quite a lot of detail and explanation. In other projects the interface may be less of an important focus.

## 0.9 Project Management

How have you managed your project and the writing of a substantial piece of software? Discuss the appropriateness of your methods in the light of your experience on the project.

## 0.10 Results and Evaluation

Describe and present the results of your work in an appropriate form (proofs, outputs generated by software, screen shots, etc). It is most important that you evaluate the success of your software. How does it compare with the original specification? How reliable is it? Comment on its robustness (e.g., to unexpected data), performance (e.g., response times or processing times in different situations), etc. In most cases you will want to include evaluation of your product (software, manuals, etc) by potential users.

## 0.11 Discussion

Here you will summarise your achievements and also the deficiencies of your project. You can also say what you would or could have done, if you had had more time or if things had worked out differently. It is important to be completely honest about the deficiencies and inadequacies of your work, such as they are. Part of your aim is to demonstrate your ability to recognise problems that remain.

## 0.12 Conclusion

Give a brief statement of how the solution that you have provided addresses the problem stated in the introduction. Provide an evaluative statement based on the results. You should not introduce new material.

“I always thought something was fundamentally wrong with the universe” [**?** ]

# Bibliography

For your Final Year project it is required that you cite and reference work to which you owe an intellectual debt. It is required that you cite and reference work that provides supporting evidence. It is required that you cite and reference work so that the reader can find the sources that have been quoted.

## 0.13 Appendices

The Report must contain an appendix explaining file structure of the .zip file submitted electronically (see below). The appendix must also contain an information on how the code should be run. Other appendices may include documents such as: the project proposal; Software Requirements Specification; a selection of experimental data; schedules; testing strategy; risk management plans; glossary; manual; etc. Don't include the source code as an appendix (submit it as a part of a .zip file; see below). Don't include voluminous appendices (these should also be submitted electronically, as a part of a .zip file).