Cover Page:

Title Page:

Final Report

COMP520-18Y (HAM)

Automated Searching for Differential Characteristics in SHA-2

Mitchell Grout

Supervisor(s):

Ryan Ko

Aleksey Ladur

Cameron Brown

Abstract:

Acknowledgements

I would like to give thanks to my supervisors, Dr. Ryan Ko, Aleksey Ladur, and Cameron Brown, for the time and effort they have put into this project. Their input has been invaluable in helping fix the scope of this project, giving ideas for what I should focus on, and how to approach the problem. I would also like to thank Daniel Roodt of CROW for all of his assistance with learning about hash functions and their cryptanalysis, and giving advice on how to best implement particular features. Finally I would like to thank all other CROW members for their advice along the way; without them, this project would not have been as successful.

Table of contents: Todo with Word

Introduction

- Motivation; aims of the project; outline of the report

The SHA-2 family of functions is used in a variety of critical areas, such as file integrity, proof of work, and password hashing. All of these applications rely on several assumed properties, which are necessary to prevent abuse by malicious actors.

Background

- Related and relevant work, critiqued against the aims of your project

We begin by giving the formal definitions for the key properties specified above.

- Preimage resistance: Given a hash value h, it is computationally infeasible to find an input x such that H(x) = h.

- Second preimage resistance: Given an input x, and its hash value H(x) = h, it is computationally infeasible to find a distinct input x’ such that H(x’) = H(x) = h

- Collision resistance: It is computationally infeasible to find two distinct inputs, x and x’, such that H(x’) = H(x)  
 - Avalanche effect: Given an input x, if we flip one bit in x, then each bit in H(x) should flip with a probability of 0.5

By “computationally infeasible”, we mean that there exists no better method than brute-force. A special case however is collision resistance; in **any** hash function, it is always possible to find collisions in less than brute-force time using the Birthday attack. This is a probibalistic attack which runs in sqrt(2^n) = 2^(0.5n) time, where n is the number of bits in the input space. As such, when referring to collision resistance, computationally infeasible means no better than the Birthday attack.

We consider a hash function to be broken when we can demonstrate an attack which either runs faster than a brute-force attack, or in the case of collision resistance, give a collision.

An example of a broken cryptographic hash function is MD5. In 2004, Wang et al. both demonstrated a technique for producing collisions in less than 2^64 time, faster than the birthday attack, and additionally produced a collision. The current best cryptanalysis of MD5 allows the creation of arbitrary collisions in 2^18 time, which typically runs in less than one second on modern computers.

To effectively attack a hash function, we require some variety of framework. Commonly used frameworks in recent years are bicliques for preimage attacks, and differential/linear cryptanalyis for collision attacks. In this project, we have selected differential cryptanalysis.

Design

- Approach to the problem – likely to refer to Background

- Overall Solution/Approach Design

Implementation

- Technologies, libraries, schematics, data flow

Evaluation

- Does it work, does it meet expectations/requirements

Conclusion

- Drawing together outcomes and original plan. Future work.

References: Todo with Endnote

Appendices: Todo