CS 465 Computer Security BYU | Computer Science

# Fall 2021

Section 1: TTh 3:30pm - 4:45pm - JFSB B092 (changed from MARB 130)

# Project #2: Hash Attack

## Objectives

The objectives for this assignment are:

- Understand the distinction between collision attacks and pre-image attacks.
- Test whether theoretical costs are observed in practice
- Demonstrate ability to design and conduct an experiment
- Demonstrate ability to convey technical results clearly

## Background

A collision attack in hashing is when an attacker finds two separate source messages that both hash to the same value. The expected time for this attack is 2^(n/2) where n is the number of bits in the hash digest.

A pre-image attack in hashing is when an attacker is given a specific hash value (usually the hash value of an intercepted message) and is asked to find a source message that hashes to that value. The expected time for this attack is 2^n where n is the number of bits in the digest.

## Project Description

In this lab, you will gather experimental data regarding how difficult it is to perform a collision and preimage attack on a hash.

For this lab you will need to do the following:

1. Create a wrapper for SHA-1 which takes as input the string to hash as well as the number of bits (n) that the hash should be. The output will be the SHA-1 hash of the given string truncated to the provided number of bits.
2. Using this wrapper, conduct a series of collision and preimage attacks at different bit sizes.
   - You should test at least 4 different bit sizes between the range 8 to 32. Choose reasonable values for bit sizes (e.g., 8, 10, 16, 20, 24). You should have at most 2 trials at bit sizes between 8-15.
   - You should include data for non-multiple-of-8 bitsizes.
   - Don't attempt attacks against bit sizes that take you hours to complete. Be reasonable.
   - Keep track of the number of attempts (hashes) it took for the attack to be successful. Do not track wall-clock time.
   - You should gather at least 50 samples at each bit size.
3. Write a 2-4 page report, describing how your experimental results compare to the theoretical difficulty of attacking a hash of a given size.
   - Your report should be clear and concise.
   - Your report should include a graph or table summarize your experimental results. The graph should compare the bitsize with the average number of attempts it took for each attack type to be successful.
   - Your graph or table should report on the variance you saw during testing.
   - Recommendation: Use a logarithmic scale graph.
   - Recommendation: Plot theoretical cost vs experimental results
   - Target Audience: Technical people who do not know about this specific property of hashes and hash attacks
4. Have another person review your report and give you feedback on the clarity of the writing and results. Include the name of your reviewer in your final written report for full credit.

# Starting Point

SHA-1 typically produces a 160-bit digest. To create our toy SHA-1 implementation, we will use a valid SHA-1 implementation and truncate the results so that the digest is of a reduced size.

Although there are many implementations of SHA-1 available on the web, we suggest using one of the following:

### Java

- included SHA-1 algorithm (simple example of hashing a string)

### C/C++

- OpenSSL (Simple example)
- Crypto++
- hashlib++

### C#

- System.Security.Cryptography.SHA1CryptoServiceProvider (Usage example with tips)

### Python

- Look at the built-in hashlib. And for python3 understand the bytes type versus the bytearray type (bytearray is mutable). Even though there is a learning curve, I recommend python3.

When choosing your language/library, consider that more common libraries like OpenSSL or those included with Java or C# often have better documentation and are easier to get help with than a less commonly used implementation. They are also good experience to list on a resume.

If you find another library or source code that could be helpful to other students, please share them on Piazza.

## Grading

The specification for this assignment has been left open-ended intentionally. We expect that each student will create their own plan for performing the attacks and find an effective way of presenting their results.

## Grading Rubric:

- 20 points for report describing your experiment - think: reproducability
- 10 points will be awarded based on the quality of the writeup that explains your experimental design
- 10 points for the quality and clarity of the graph(s) presenting your data
- 5 points if results are consistent with the theory or include an explanation why they might (mildly) differ from the expected cost
- 5 points for soliciting one external review of your report

## Submission

Submit your written report as a PDF file along with a zip file or gzip file with your source code to Learning Suite.

## Tips and Hints

- Make sure that you use BIT sizes not byte sizes.
- If you find results that a drastically different from the theoretical values, take a look at your code and look for implementation errors.