Applied Crypto

Project: Cryptographic Hash

This project aims to deepen understanding of cryptographic hashes through practical experience with hash collisions and Merkle trees (traditional binary Merkle tree). It consists of literature review, discussion questions, and hands-on coding exercises. The goal is for students to not only grasp the theoretical aspects but also gain insights into the practical challenges and applications of cryptographic hashing.

Part I: Literature Review

A literature review refers to a comprehensive survey of existing research, studies, and scholarly articles related to specific topics. In this project, students are asked to familiarize themselves with the topics on Merkle trees, hash collisions, and hash puzzles. Students are expected to identify and analyze sources that discuss these concepts and their recent developments.

Requirements:

- Compile a reference list of at least three sources that provide insight into the concepts of Merkle trees, hash collisions, hash puzzles, and their recent development.
- For each reference, summarize the main ideas in a brief paragraph.
- Sources can include book chapters, peer-reviewed journal articles, conference papers, or credible internet sources.
- The following references are provided as starting points but must not be used in your list:
    - B. Weber and X. Zhang. Parallel hash collision search by rho method with distinguished points. Proc. of the 14th IEEE LISAT 2018, Farmingdale, NY, May 4, 2018, pp. 1-7.
    - Mark Stamp's book (as listed in the course syllabus), specifically sections 5.2 (Birthday attack, Nostradamus attack) and 5.3 (MD4).
    - J. Kelsey and T. Kohno, Herding hash functions and the Nostradamus attack, eprint.iacr.org/2005/28l.pdf.

Part II: Essay Questions

1. Explain why hash collisions are a mathematical inevitability.
2. Considering a room with N people, including Trudy, what's the probability that at least one other person shares Trudy's birthday? At what minimum N does this probability exceed 50%?
3. In a room of N people ($N \leq 365$), what's the probability of any two sharing a birthday, and what's the minimum N for this probability to be over 50%?
4. Describe the principle of the birthday attack on hashing and how it offers efficiency over brute-force attacks.
5. Discuss the main issues associated with hash functions created using the Merkle-Damgård Construction process.

Part III: Code Project

Use jupyter notebook to do the coding problems.

1. Merkle Tree Implementation

- Implement Merkle trees using SHA256 or some other hashing algorithms. Students can utilize existing crypto packages for hashing functions.
- Each leaf node references a plaintext file.
- Conduct a test with four leaf nodes, displaying the tree structure and hashes.
- Conduct a second test with six leaf nodes, displaying the tree structure and hashes.

2. Root Hash Observation
   - Modify the content of one text file in the four-leaf-node scenario and compare the root hashes. Discuss your observations.

3. Hash Collision
   - Define a hash function using SHA256 but take only 4 bits as hash output.
   - Use the implementation in step 1(Merkle Tree Implementation) with this hash function.
   - Attempt to generate multiple text files with identical meanings but different hashes by altering file contents (e.g., adding spaces).
   - Find a hash collision among the text files. Discuss how many such files need to be generated.
   - Discuss strategies for finding collisions with hashes ranging from 4-bit to 160-bit in length.

4. Hash Puzzle
   - Using the chosen hashing algorithm (4 bits output), solve hash puzzles by finding hashes with a leading 1 zero bit, and then 2 zero bits.
   - Briefly discuss the workload involved in solving a puzzle requiring a 20-bit zero prefix for the SHA256.

Submission

Submit to Brightspace as we did in the last several projects.

Grading Rubrics (total 12points)

	Max Points	Expectations
Part I	2.0	• Three references (any resources including books) are listed in APA standard; each reference has a well written synopsis. • Concepts of the Merkle tree, hash collision and hash puzzle are covered by the references. • References must be relevant and accurately cited. Synopses should demonstrate an understanding of the concepts and convey the main ideas clearly.
Part II		
1	0.5	• A clear explanation with logical reasoning is required. • mathematical formula is preferrable. •
2	0.5	• Correct calculation of N
3	1.0	• Correct calculation of N

4	0.5	• Clearly explain the concepts. Reasoning is essential; mere statements without support will result in a loss of points. Avoid excessive length and ensure originality in responses.
5	0.5	• Clearly explain the concepts. Reasoning is essential; mere statements without support will result in a loss of points. Avoid excessive length and ensure originality in responses.
Part III		
1	2.0	• Correct implementation and demonstration of a Merkle tree with both four and six leaf nodes. The tree structure and hashes must be accurately displayed.
2	1.0	• Insightful observation and explanation of the root hash changes upon altering a leaf node's content.
3	2.0	• Successful demonstration of finding a hash collision with a 4-bit hash function. • Students should outline a clear strategy for approaching hash collisions with hash functions ranging from 4-bit to160-bit.
4	2.0	• Completion of hash puzzles with 1 and 2 zero prefixes. • Students must discuss and estimate the workload accurately for solving puzzles with varying complexities.