Cryptographic Hash Functions

Grant Haataja

February 19, 2019

Introduction

1. What is a Hash Function?
   - Definition of a Hash Function
   - Requirements of Hash Functions
2. How do Hash Functions Work?
3. Uses of Cryptographic Hash Functions
4. Types of Cryptographic Hash Functions
5. Hashing in Cybersecurity
   - MD5 (Message Digest Algorithm 5)
   - SHA-256 (Secure Hashing Algorithm)
6. Hashing Examples
7. Pros and Cons of Hashing
8. Hash Function Implementation

Definition of a Hash Function

A hash function is a mathematical algorithm that takes input of arbitrary length and maps it through a series of transformations to a string of fixed length.

Requirements of Hash Functions

- Computationally efficient
- Pre-Image Resistance
- Deterministic
- Collision Resistance
- Outputs should have the appearance of complete randomness

How do Hash Functions Work?

- Hash functions take an input of any size and go through a series of steps to change it into something theoretically unrecognizable to the original message
- The output of a hash function is commonly referred to as a *digest*

A hash function can be as simple as

$$f(x) = x \mod 7.$$

Then any input could be converted into a decimal number and sent through the function. For example, consider the input: "Message". If we apply the standard conversion from letters to numbers, (A → 1,B → 2,...,Z → 26), "Message" becomes 13 5 18 18 1 7 5. Then, we take it as the number 1,351,818,175 and send it through the function, giving

$$f(1351818175) = 1351818175 \mod 7 = 1$$

.

Uses of Cryptographic Hash Functions

- Verification of files or messages
- Password protection
- file or object identifier over file-sharing networks
- Bitcoin mining
- Bitcoin creation of addresses

Types of Cryptographic Hash Functions

- MD4 (Message Digest Algorithm)
- MD5
- SHA-1 (Secure Hashing Algorithm)
- SHA-2
- SHA-3
- RIPEMD-160
- CryptoNight
- PBKDF2
- bcrypt
- Argon2

MD5 (Message Digest Algorithm 5)

- 128-bit hash
- Created in 1991 by Ronald Rivest as a replacement for MD4
- Collision resistance broken in 2004 after roughly 2^{21} hashes, reportedly taking only an hour to complete
- Still used widely but no longer secure for sensitive data
- Common passwords can be found by typing their hashes into Google search

SHA-256 (Secure Hashing Algorithm)

- 256-bit hash
- Designed by the NSA as part of the SHA-2 set in 2001
- Collision resistance hasn't been fully broken yet
- Most commonly used and trusted hashing algorithm currently, although industry is in the process of switching to SHA-3

Hashing Examples

- We have the message ”Hashing is a secure way to store passwords, change my mind”.
- MD5: b30d5d9bc11f392daa4950be13d35106
- SHA-256: a1b196e3e571170f5111c9f0058d20cc4761546468fbe6bdc16f2663f6096e7
- Now we will remove the comma from the message and hash ”Hashing is a secure way to store passwords change my mind”.
- MD5: 91a6526d84520f2aad7f25116880d4e9
- SHA-256 7809dace6f2004d72611a4a59f2d6fa5d2ab0307c2b7de741b86779275681a

Pros and Cons of Hashing

- PROS
  - Portable and easy for storing
  - Simple and fairly secure way to store sensitive data to be used for verification, as long as the hashes are salted
  - Does not require any key for verification
- CONS
  - Cannot be used to send messages, since there is no way to reverse secure hashes
  - Unsalted hashes can become security risks

MD5 Implementation

Sources

- http://practicalcryptography.com/hashes/md5-hash/
- https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm
- https://komodoplatform.com/cryptographic-hash-function/
- http://alexander.holbreich.org/cryptographic-hash-functions/
- https://blockgeeks.com/guides/cryptographic-hash-functions/