Research Paper

“Hash Functions in Discrete Mathematics”

Aftab Alam Masjidi

MAT-253: Discrete Mathematics I

Barry University

Professor Singh

April 1, 2025

Research Proposal

Title: *Hash Functions in Discrete Mathematics*

Student: Aftab Alam Masjidi

Introduction and Motivation

I propose investigating the topic of ***Hash Functions in Discrete Mathematics***. Hash functions are mathematically formulated functions that accept input data of arbitrary size and map it to fixed-size values. Hash functions are ubiquitous in cryptography, computer science, and data structures — all fields that heavily utilize discrete structures and number theory.

I chose this topic because hash functions demonstrate so beautifully how abstract mathematical concepts such as functions, modular arithmetic, the pigeonhole principle, and Boolean logic find powerful everyday applications. My background and interests in computer science and mathematics make the topic intellectually challenging and practically relevant.

Research Objectives

- Define and discuss hash functions and their key mathematical properties (e.g., collision resistance, determinism, pre-image resistance).
- Learn fundamental theorems such as the Pigeonhole Principle and Universal Hashing.
- Learn how discrete structures are used in SHA algorithms, digital signatures, hash tables, and blockchains.
- Explore future directions such as quantum-resistant hash functions and zero-knowledge proofs.

Mathematical Focus Areas

- Functions and mappings ($H: D \rightarrow R$)
- Modular arithmetic and combinatorics
- Complexity analysis (e.g., brute-force search space)
- Probability in Universal Hashing.

Expected Outcomes

- A deep understanding of the mathematical foundations of hash functions
- Real-world application examples with mathematical explanations
- Relationship to future-proof cryptographic techniques
- A complete 6–8 page research paper with references.

References

- Carter & Wegman, Universal Classes of Hash Functions, 1977
- Menezes et al., Handbook of Applied Cryptography, 1996
- Rivest, MD5 Digest Algorithm, 1992
- Rogaway & Shrimpton, Hash Function Basics, 2004

"Hash Functions in Discrete Mathematics"

Abstract

Hash functions are mathematical algorithms that map data of any size to a fixed-size value, a hash. Hash functions have many uses in data structures, cryptography, and computer science due to their efficiency and determinism. In this paper, we examine the mathematical nature of hash functions, including their properties like collision resistance, pre-image resistance, and avalanche effect. We examine the use of hash functions in algorithms and systems such as hash tables, digital signatures, and blockchain technologies. This examination also touches on the evolving security challenges and potential future research directions.

Keywords: Cryptography, Hash Functions, Discrete Mathematics, Collision Resistance, SHA, Avalanche Effect, Digital Signatures

MSC 2010 Subject Classification: 68P25 (Data structures), 94A60 (Cryptography), 68Q87 (Cryptographic aspects of algorithms), 11T71 (Algebraic coding theory).

Introduction

Hash functions are a fascinating intersection of discrete mathematics, number theory, and computer science. Essentially, they are deterministic algorithms that accept input data of any length and generate a fixed-length output, typically a bit string. The same input always produces the same hash, but even a minor change in the input generates an incredibly different output — a property known as the avalanche effect.

Their use in modern computing is ubiquitous. Hash tables facilitate lookups in constant time within data structures, and cryptographic hash functions like SHA-2 and SHA-3 secure internet communications. They also power blockchain mechanisms and digital signatures that verify data integrity without revealing the original data.

I wished to learn about hash functions because they represent the perfect illustration of abstract mathematical principles being applied to solve real problems. As someone who is interested in both mathematics and computer science, knowing how they work, their constraints, and their applications is both intellectually stimulating and helpful.

This paper provides an in-depth introduction to hash functions. We will begin with definitions and basic properties, then move on to seminal examples and theorems, and then explore current applications, limitations, and research frontiers.

Background

A ***hash function H*** is a mapping function from a domain D to a range R (or also known as the codomain), often written as:

$$H: D \rightarrow R$$

Typically, D is very big (perhaps even infinite), but R is fixed in size and finite — say, 256-bit binary strings. So, applying the pigeonhole principle, there are going to be unavoidable collisions (i.e., when two distinct inputs yield the same output).

Some key definitions and properties:

- **Collision**: A pair of different inputs $x \neq y$ such that $H(x) = H(y)$.
- **Collision resistance**: It is computationally infeasible to find such a pair.
- **Pre-image resistance**: Given a hash value h, it should be hard to find any input x such that $H(x) = h$.
- **Second pre-image resistance**: Given an input x_1, it is difficult to find a different input x_2 such that $H(x_1) = H(x_2)$.
- **Avalanche effect**: A small change in input results in a large, unpredictable change in output.
- **Determinism**: The same input always produces the same output.

Hash functions can be:

- **Non-cryptographic** (used in data structures for efficiency), or

Cryptographic (designed to withstand malicious attacks and ensure data integrity).

The subject of hashing in discrete mathematics is related to functions, permutations, combinatorics, modular arithmetic, and complexity theory.

Mathematical Foundations and Structures

Let us start with a simple example: consider the modular hash function:

$$H(x) = x \bmod m$$

It's a simple hash function used in hash tables. For example, for m = 10 and

x = 23, H (23) = 3. It hashes values into m "buckets."

Theorem (Pigeonhole Principle)

If $|D| > |R|$, then H must map at least two distinct inputs to the same output.

This points to the inevitability of collisions in hash functions and forms the basis for the study of collision resistance.

Universal Hashing

A hash function family H is universal if for all $x \neq y$, the probability over a randomly chosen $h \in H$ that $h(x) = h(y)$ is $\leq 1/|R|$.

Universal hashing, as defined by Carter and Wegman (1977), provides theoretical foundations for minimizing collisions.

Cryptographic Hash Families

Cryptographic hash functions like MD5, SHA-1, SHA-2, and SHA-3 make use of bit-level manipulation, Boolean algebra, and modular arithmetic. SHA-256, for instance, utilizes:

- Padding the message to a 512-bit multiple.
- Initializing eight words of 32 bits each.
- Applying a series of logical operations (AND, XOR, NOT) to transform each block.
- Compressing the message block into a resulting 256-bit digest.

Algebraically, this is a sequence of bit operations and modular sums that yields a many-to-one mapping with good security properties.

Applications and Examples

1. **Hash Tables (Computer Science)**
   a. One of the most common uses of hash functions is in hash tables, which allow O (1) average-case lookups. A hash function sends each key to an array index.
      Example:
      Let the keys be strings (e.g., "Aftab", "Alam"). We convert characters to ASCII, sum them up, and mod with the table size.
      H("Aftab") = (65 + 102 + 116 + 97 + 98) mod 10 = 478 mod 10 = 8
      In case there are collisions (multiple keys mapping to a slot), we resolve them by chaining or open addressing.
2. **Digital Signatures (Cryptography)**
   In *digital signatures*, a hash of the message is signed instead of the message itself to save space and ensure integrity.
   i. Example process:
      1. Hash the message: h = H(m)
      2. Encrypt h with a private key: s = E(h)
      3. Send (m, s)
   ii. At the receiver's end:
      - Hash m again to get h′
      - Decrypt s to get h

- Compare h and h′: if equal, message is authentic.

This is used in SSL/TLS, e-mail authentication, and e-commerce.

3. **Blockchain Technology**

Blockchains like Bitcoin use SHA-256 to hash:

i. Single transactions
ii. Whole blocks
iii. Proof-of-work problems

It is the collision resistance and pre-image resistance of the hash functions that make the blockchain immutable. Each block's hash depends on the last one, forming an unalterable chain.

4. **Password Hashing**

Passwords are never saved as plain text. They are hashed using functions like bcrypt, Argon2, or PBKDF2.
Upon login, the input is hashed again and then matched with the stored hash. Salting (adding random bits) adds more security against rainbow table attacks.

Future Directions and Challenges

With the developments in quantum computing, classical hash functions will presumably be vulnerable to pre-image as well as brute-force attacks. Research is on to design post-quantum hash functions using lattice-based issues and code-based cryptography.

Another emerging area is zero-knowledge proofs (e.g., zk-SNARKs), which utilize hashes to prove knowledge without revealing secrets — a building block of privacy-safeguarding technologies.

There are also questions of research on perfect hash function construction (no collisions), minimizing optimal hash computation time, and determining lower bounds on collision resistance as a function of input size.

Conclusion

Hash functions are an excellent example of mathematical richness in combination with ease. From their theoretical underpinnings in discrete mathematics to their practical applications in internet security, they demonstrate how abstracted-out ideas in the mind transform into technological blocks. With the onset of quantum computing and complex cyber-attacks, the research of understanding and developing hash functions will remain a key area of focus.

Bibliography

Carter, J. L., & Wegman, M. N. (1977). Universal classes of hash functions. Journal of Computer and System Sciences, 18(2), 143–154.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.

Rivest, R. L. (1992). The MD5 Message-Digest Algorithm. Internet RFC 1321.

Rogaway, P., & Shrimpton, T. (2004). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second preimage resistance, and collision resistance. FSE 2004. Springer.