The Art of Conventional Hash Function

Yi Wang1,2, ??, Diego Barrios Romero3, Daniel Lemire4, Li Jin1,2\*

1. Ministry of Education Key Laboratory of Contemporary Anthropology, Collaborative Innovation Center for Genetics and Development, School of Life Sciences, Shanghai, China.
2. Human Phenome Institute, Fudan University, Shanghai, China.
3. Fraunhofer Institute for Digital Medicine MEVIS, Bremen, Germany.
4. Université du Québec (TÉLUQ) Montreal, Canada

Abstract

Introduction

A hash function is a function to convert arbitary data to an integer of fixed length (figure1). The input data was called “keys” and the output integer was called “hashes”. It is the old working horse for computer science and have numerous applications: fast algorithms, hash table, file checksum, duplication/collision detection, password storage, unique ID generation, proof-of-work, etc. The keyword search of “hash” on GitHub results 61,762 projects and 266M codes, which is on-par with that of “algorithm” (393,000 projects and 138M codes) and “deep learning” (78,263 projects and 3M codes).

As the design space of hash function is open, there are hundreds of hash functions invented in last decades. With the abundance of hash function candidates, we can select few hash functions that satisfy our ideal criterions. These criteriones for ideal hash function are (from import to less import):

* Uniform distributed
* Fast to compute
* Secure\*
* Portable
* Minimal size
* Aesthetics

\*Note that security is a soft requirement for conventional (non-cryptographic) hash function but is a hard requirement for for cryptographic hash function.

