Hash Functions

Success Academy High School of the Liberal Arts
Dr. Anthony Schultz

September 15 2016

Project folder available at:
https://github.com/Trismeg/Success

Figure 1: Breakfast hash

This is an introduction to one of the most fundamental tools in modern cryptography, hash functions. Hash functions are awesome because they secure global digital communications and make amazing things possible. Today we will do the following:

- learn what is a hash function and its properties
- learn how to compute a hash function
- imagine useful applications of hash functions

The word **hash** entered English in the mid 1600's, during the dawn of science. It meant *to chop into small pieces* and represented a brilliant culinary advancement. It came from the French ***hacher***, *chop up*. This deriving from Old French ***hache***, *ax*.

Expectations

During this lesson students are expected to do the following.

- direct attention toward the subject of hashing with laser-like precision (**no talking to neighbors**)
- use only the indicated computing utilities while using the computer (**no texting or snapchatting grandma**)
- muster deep intellectual enthusiasm for the tasks at hand (**participate fully**)

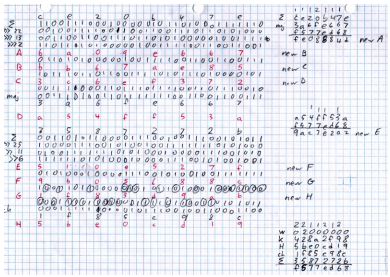

Figure 2: Hashing with SHA-256 by hand: www.youtube.com/watch?v=y3dqhixzGVo

Hash Functions

Hashing is something we do to data/numbers which adds randomness. A hash function, H, takes input data called a **message**, m, and outputs data called a **digest**, d.

Using function notation from mathematics we can describe a hash function H taking an input message m and output digest d.

$$H(m) \longrightarrow d$$

Features

A good hash function should be straightforward to use and have the three following features.

1. It is physically infeasible to determine an input message from its output hash value.
2. Any change to the input message should change the output has value so extensively as to make it appear uncorrelated to the old hash value.
3. It is physically infeasible to find two different input messages with the same output hash values.

Feature 1: We should not be able to get input m from output d. A hash function should have no inverse.

$$m \nleftarrow H^{-1}(d)$$

Feature 2: A hash function should be so random that changing one little thing in m completely changes d.

Feature 3: It should be impossible to find two different input m values which give the same output d values. This is called a collision and it is BAD!!!

$$H(m_1) \nleftrightarrow H(m_2)$$

How to Hash

The easiest way to get started hashing is to visit the following website.

www.fileformat.info/tool/hash.htm

The website will compute various hash digests in hexidecimal (base 16).

Hexadecimal or hex numbers are base 16. In our day to day lives we use base 10 numbers. The numerals for hexadecimal numbers are written:

```
01234567890abdef
```

Activity

Go to the website and hash the word "Hello".

Complete the following:

1. How many different kind of hash digests does the website compute?
2. Write down the Adler32 digest of the word "Hello".
3. Write down the Adler32 digest of the word "hello".
4. Explain which feature of hash functions requires the digest of "Hello" and "hello" to be totally different.
5. Is there another word (string) that has the same Adler32 digest of the word "Hello"?

   If so, how could we find it?

   How hard would it be to find it?

6. What would be the use of taking a hash digest of a 3D printer .STL file?