Computer Systems and Networks Department

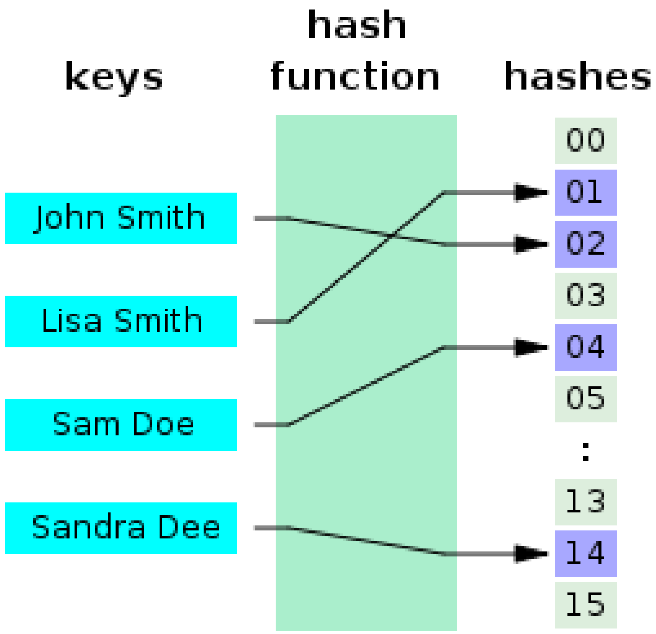
Hash functions.

Contents

- Definition. General Description
- Properties
- Types of hash-functions
- Collisions
- Areas of applications
- Conclusions

Definition. General Description

A **hash function** is any function that can be used to map digital data of arbitrary size to digital data of fixed size(with slight differences in input data producing very big differences in output data).



Definition. General Description

Hash functions widely use for

- **Hash tables**, to quickly locate a data record

- **Caches,** for large data sets stored in slow media

- **Bloom filters,** a space-efficient probabilistic data structure that is used to test whether an element is a member of a set

- **Finding duplicate records,** in a large unsorted file, we may use a hash function to map each record to an index into a table *T*, any two duplicate records will end up in the same bucket

- **Protecting data,** a hash value can be used to uniquely identify secret information

Definition. General Description

-**Finding similar records**, hash functions can also be used to locate table records whose key is similar, but not identical, to a given key

-**Geometric hashing**, This principle is widely used in computer graphics, computational geometry and many other disciplines, to solve many proximity problems in the plane or in three-dimensional space, such as finding closest pairs in a set of points, similar shapes in a list of shapes, similar images in an image database, and so on

- **Cryptography.**

Properties

Good hash functions, in the original sense of the term, are usually required to satisfy certain properties. The exact requirements are dependent on the application, hash function well suited to indexing data will probably be a poor choice for a cryptographic hash function.

Determinism - for a given input value it must always generate the same hash value

Uniformity - A good hash function should map the expected inputs as evenly as possible over its output range

Properties

- Defined range - It is often desirable that the output of a hash function have fixed size

-Variable range - In many applications, the range of hash values may be different for each run of the program, or may change along the same run

-Variable range with minimal movement (dynamic hash function). When the hash function is used to store values in a hash table that outlives the run of the program, and the hash table needs to be expanded or shrunk, the hash table is referred to as a dynamic hash table

Properties

-**Data normalization**, the input data may contain features that are irrelevant for comparison purposes, so it should be normalized

-Continuity. A hash function that is used to search for similar data must be as continuous as possible.

-Non-invertible. In cryptographic applications, hash functions are typically expected to be non-invertible

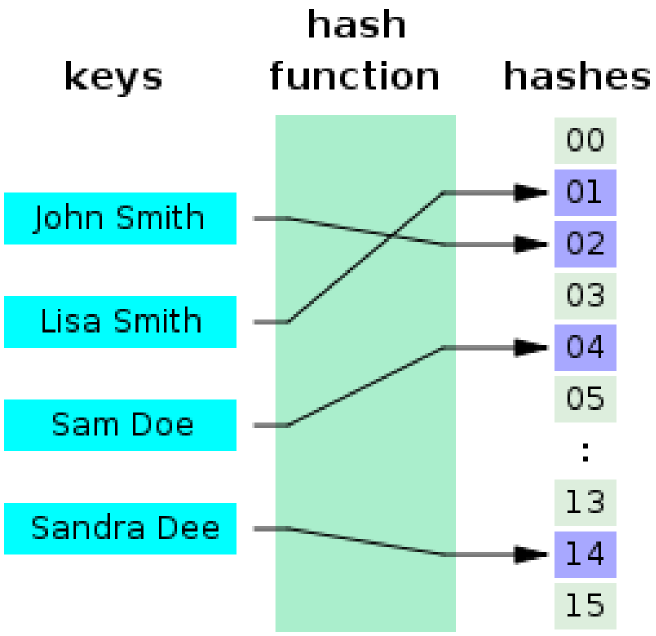
Types of hash-function

- Good hash function must satisfy at least two properties
  - quickly computed;
  - minimize collision

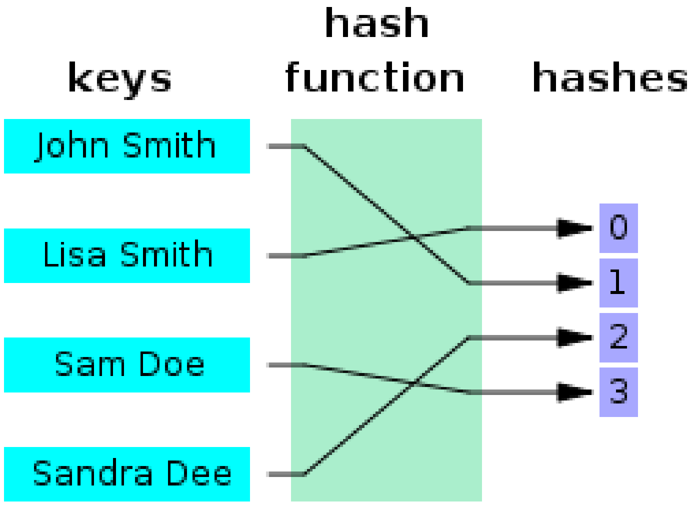
  In some case, for example for cryptographic hash-functions will be additional requirements.

  - Hash functions based on the division
  - Hash function based on the multiplication
  - Perfect hash function, hash function that is injective—that is, maps each valid input to a different hash value
  - Minimal perfect, perfect hash function for *n* keys is said to be **minimal** if its range consists of *n consecutive* integers, usually from 0 to *n*−1

Types of Hash-functions



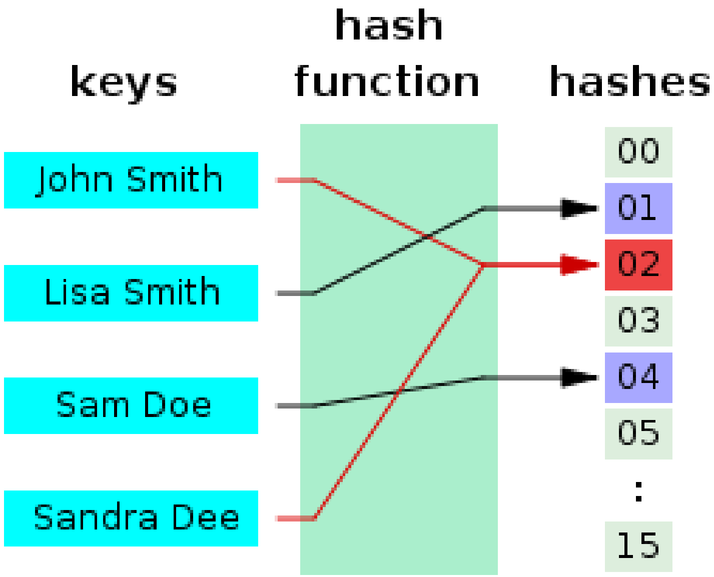
Perfect hashing



Minimal perfect hashing

Collisions

Collisions (sometimes conflict)of hash functions are called situation when two input data blocks produce the same hash codes.

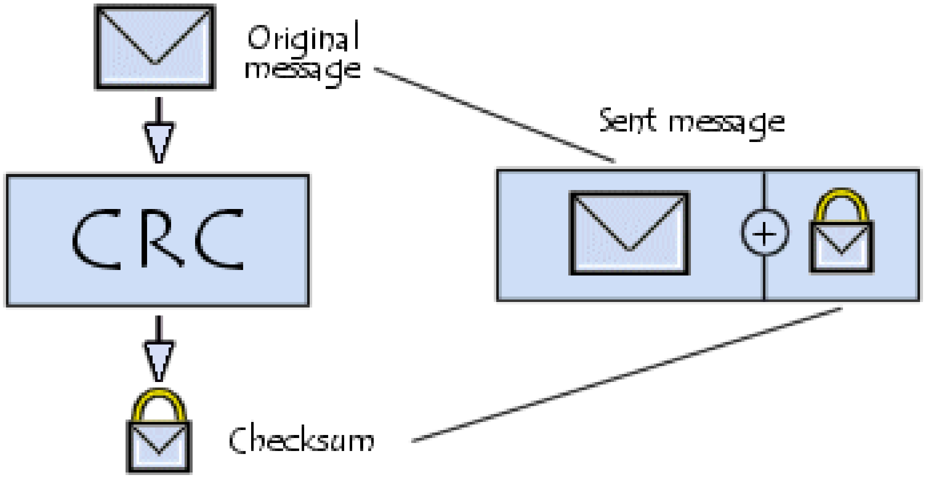


Methods of collision control

- In Hash-tables
  - Chain method (Method of direct coupling)
  - Methods of open addressing
- In Cryptography
  - Salt - adding cryptographic salt (random data string) to the input data. This method, for example, is used to store passwords in UNIX-like operating systems.

Areas of application

- **Control sum** - Simple, fast and very easy to implement hardware algorithms used to protect against unintended distortions, including hardware errors. Tens and hundreds of times faster than the cryptographic hash function, and is much simpler in hardware implementation.

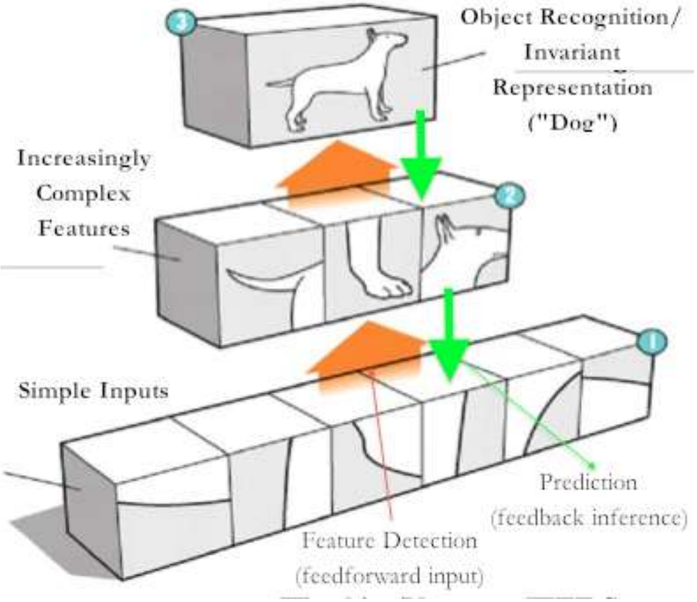


Areas of application

The price paid for such a high speed is the lack of secrecy - easy opportunity to fit a message to a known sum. Also commonly capacity of checksum (the typical number of 32-bit) lower than cryptographic hashes (typical number: 128, 160, and 256 bits), which means the possibility of unintentional collisions.

Geometric hashing - widely used in computer graphics and computational geometry method for solving tasks in the plane or in three-dimensional space, for example to find the closest pairs in the set of points or to find similar images

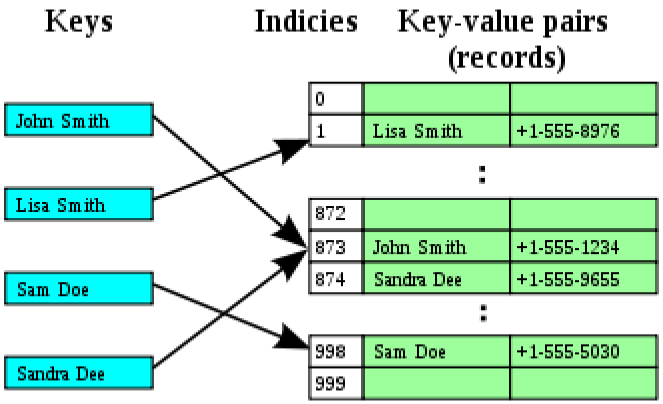
Areas of application



Hash function in this method usually takes as input any metric space and divides it by creating a grid of cells. The table in this case is an array with two or more indexes and called Grid file. Geometric hashing is also used in telecommunications when dealing with multi-dimensional signals

Areas of application

Acceleration data search. Hash table is called a data structure that can store pairs of the form (key, hash code) and supporting operations of searching, insertion and removal of the item. The object of the hash table is accelerate the search, for example, when recording the text fields in the database their hash code can be calculated and data may be placed in the section corresponding to the hash code



Keys

John Smith

Lisa Smith

Sam Doe

Sandra Dee

Indicies	Key-value pairs (records)	
0		
1	Lisa Smith	+1-555-8976
:		
872		
873	John Smith	+1-555-1234
874	Sandra Dee	+1-555-9655
:		
998	Sam Doe	+1-555-5030
999		

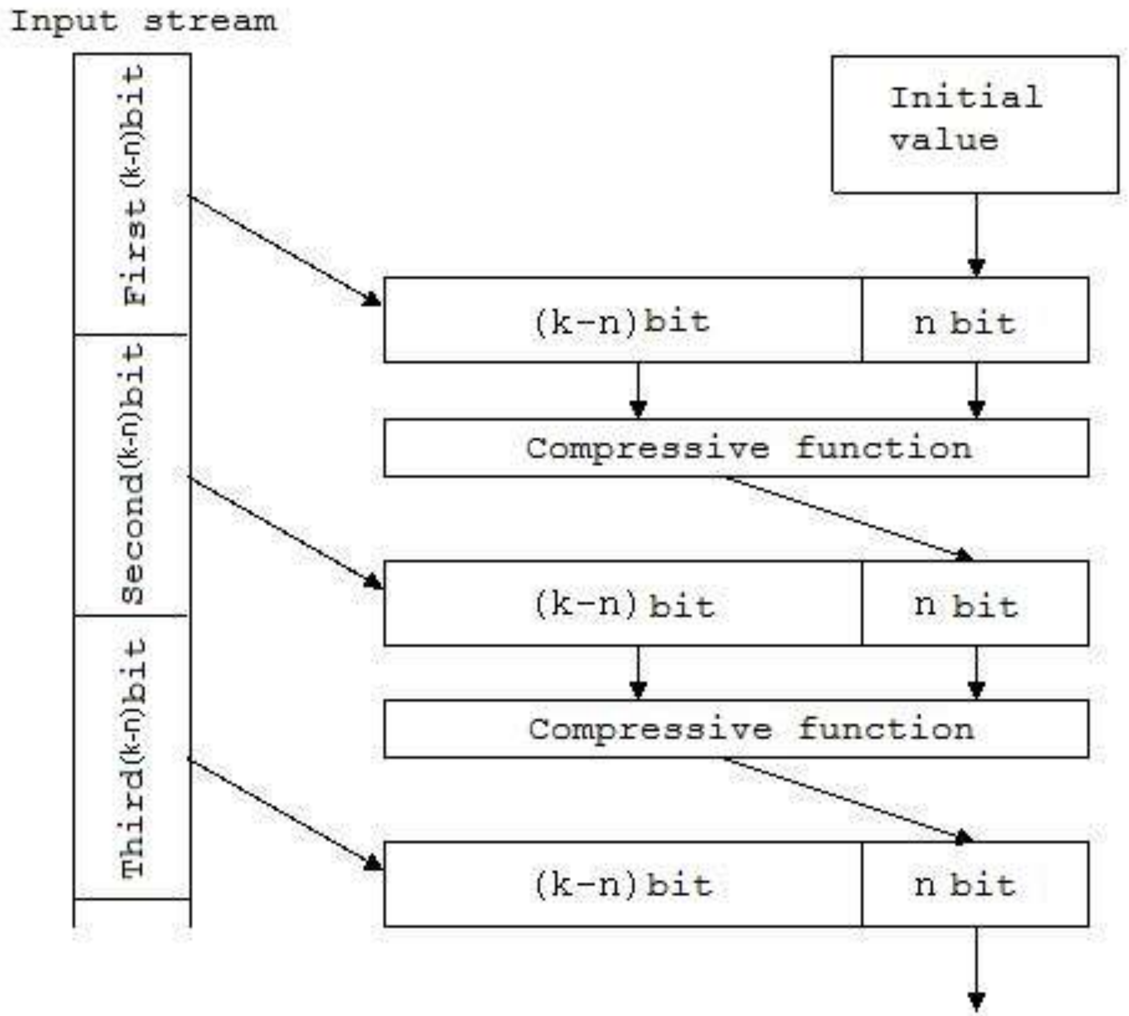
Areas of application

Cryptography. **cryptographic hash function** is a hash function which is considered practically impossible to invert, that is, to recreate the input data from its hash value alone.

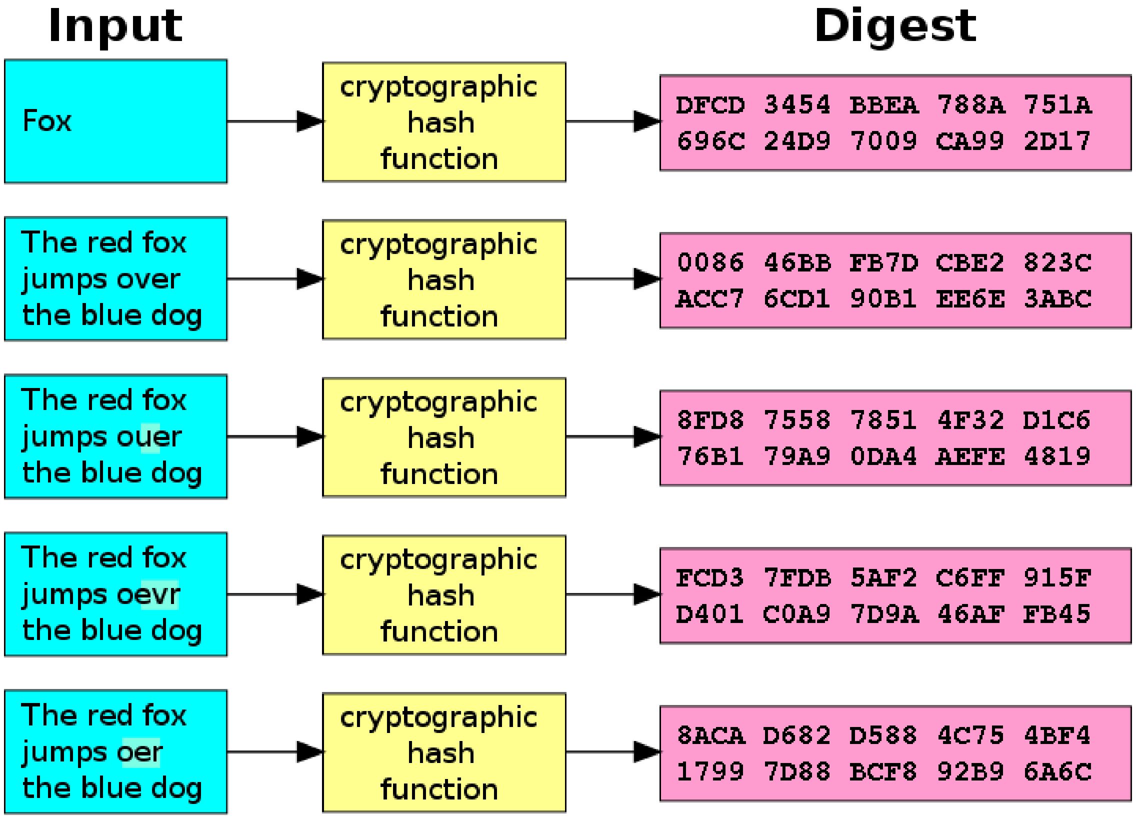
The ideal cryptographic hash function has four main properties:

- it is easy to compute the hash value for any given message
- it is infeasible to generate a message that has a given hash
- it is infeasible to modify a message without changing the hash
- it is infeasible to find two different messages with the same hash.

Areas of application



Areas of application



Input

Digest

Fox → cryptographic hash function → DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17

The red fox jumps over the blue dog → cryptographic hash function → 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC

The red fox jumps ouer the blue dog → cryptographic hash function → 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819

The red fox jumps oevr the blue dog → cryptographic hash function → FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45

The red fox jumps oer the blue dog → cryptographic hash function → 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

A cryptographic hash function (specifically, SHA-1) at work. Note that even small changes in the source input (here in the word "over") drastically change the resulting output, by the so-called avalanche effect.

Areas of application

- **Verifying the integrity of files or messages**.

  An important application of secure hashes is verification of message integrity.

  Determining whether any changes have been made to a message or not.

  In simple variant CRC32 can be used but sometimes MD5, SHA1, or SHA2 hashes are posted along with files on websites or forums to allow verification of integrity.

Areas of application

- **Password verification**

  A related application is password verification (first invented by Roger Needham).

  Storing all user passwords as cleartext can result in a massive security breach if the password file is compromised.

  One way to reduce this danger is to only store the hash digest of each password. To authenticate a user, the password presented by the user is hashed and compared with the stored hash.

  The password is often concatenated with a random, non-secret salt value before the hash function is applied/

Areas of application

- **File or data identifier**

  A message digest can also serve as a means of reliably identifying a file; several source code management systems, including Git, Mercurial and etc, use the sha1sum of various types of content (file content, directory trees and etc.) to uniquely identify them.

  Hashes are used to identify files on peer-to-peer filesharing networks. For example, in an ed2k link, an MD4-variant hash is combined with the file size, providing sufficient information for locating file sources, downloading the file and verifying its contents.

  Magnet links are another example.

Areas of application

- **Pseudorandom generation and key derivation**

  Hash functions can also be used in the generation of pseudorandom bits, or to derive new keys or passwords from a single, secure key or password.

  There are several methods to use a block cipher to build a cryptographic hash function, specifically a one-way compression function.

Areas of application

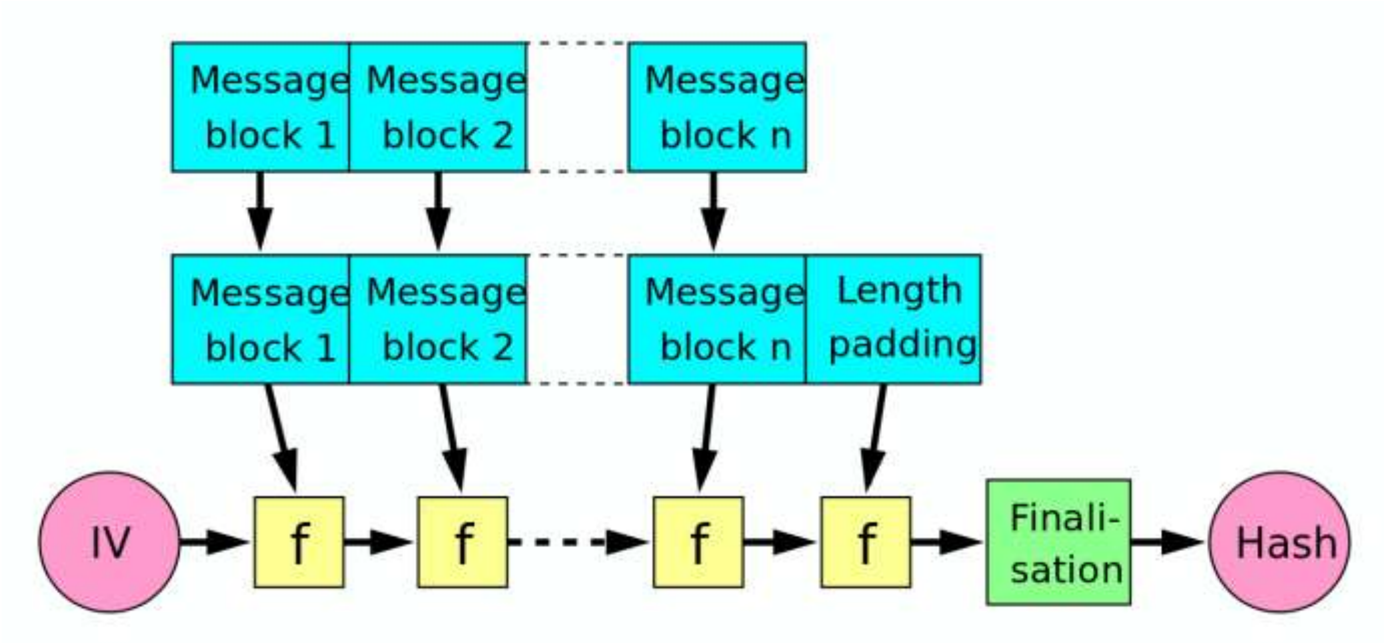
The methods resemble the block cipher modes of operation usually used for encryption. All well-known hash functions, including MD4, MD5, SHA-1 and SHA-2 are built from block-cipher-like components designed for the purpose, with feedback to ensure that the resulting function is not invertible.

SHA-3 finalists included functions with block-cipher-like components (e.g., Skein, BLAKE) though the function finally selected, Keccak, was built on a cryptographic sponge instead.

Areas of application

Merkle–Damgård construction

Hash function must be able to process an arbitrary-length message into a fixed-length output.



Areas of application

A hash function built with the Merkle–Damgård construction is as resistant to collisions as is its compression function; any collision for the full hash function can be traced back to a collision in the compression function.

The last block processed should also be unambiguously length padded; this is crucial to the security of this construction.. Most widely used hash functions, including SHA-1 and MD5, take this form.

Conclusions

A hash function can be used in a wide variety of applications. Particular:

-for finding duplicate entries in the tables

- for finding similar images

- for a wide range of cryptographic tasks

Properties:

- easy to compute the hash value
- infeasible to generate a message that has a given hash
- infeasible to modify a message without changing the hash
- infeasible to find two different messages with the same hash.