

CHEM1047 - Week 15 Lecture 1 – Data integrity control

□ G.J. Simmons, “Contemporary Cryptology: the Science of Information Integrity”, IEEE Press, 1994.

Modern analytical instruments (NMR, HPLC, X-Ray, ICP-MS, etc.) produce datasets that are kept in some form of digital storage. An important problem is making sure that a dataset is not altered, either spontaneously (bit flips in magnetic media, flash memory deterioration, alpha particle hits on RAM chips) or deliberately. In high-stakes situations, such as forensic analysis and litigation, the storage system must be able to survive attacks by highly skilled individuals (definitions of *good* and *bad* can differ), with the data provably intact.

1. Boolean functions

Digital data is stored and manipulated in the binary format – as strings of ones and zeroes. Elementary functions that operate on such data are called *Boolean functions*. The most common Boolean function of one argument is called “NOT”; it swaps ones for zeros and zeros for ones:

$$\text{NOT}([1 \ 0 \ 1 \ 1]) = [0 \ 1 \ 0 \ 0]$$

Common Boolean functions of two arguments are:

A	B	AND(A,B)	OR(A,B)	XOR(A,B)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

The following notation is often used:

$$\begin{aligned} \text{XOR}(A,B) &= A \oplus B, & \text{AND}(A,B) &= A \wedge B \\ \text{OR}(A,B) &= A \vee B, & \text{NOT}(A) &= \neg A \end{aligned}$$

The behaviour of logical circuits in modern computers is formally equivalent to Boolean algebra.

2. Hash functions

One way to detect data modifications is to create a “watchdog” function – something that is guaranteed to change if anything is changed anywhere in the data. The value of this function can be computed when the dataset is acquired, and then stored independently. Every time the dataset is used, the guardian function can be re-calculated and compared with the stored value – a match would guarantee that the data has not been tampered with. Such functions are called hash functions.

Formally, a cryptographic *hash function* is a function that accepts an input (called *message*) of any length and produces a fixed length output (called *digest* or *hash*) with the following properties:

1. The complexity of computing the digest is linear with the size of the message.
2. A modification in the message is almost certain to change its digest.
3. Two randomly selected messages are almost certain to have different digests.
4. It is impractical to find a message that has a specific digest.

The definition of *almost certain* in this context is that the probability of two randomly chosen messages having identical N -bit hashes (this is called *hash collision*) is $2^{-N/2}$. Digests of modern hash functions are at least 128 bits long. The probability of a collision is therefore $2^{-64} \approx 5 \cdot 10^{-20}$ – it is a vanishingly rare event. In practice, any probability smaller than the risk of committing suicide (approximately 10^{-4} per year in the UK, according to ONS) may be considered negligible.

Table 1. Examples of hash strings of randomly chosen quotations, obtained using the popular MD5 hashing algorithm. Changing, adding, or removing even one symbol in the message text changes the hash string completely.

Message	MD5 hash (converted into ASCII text)
In the beginning the Universe was created. This has made a lot of people very angry and been widely regarded as a bad move. – Douglas Adams	5705e6805572ea5217f3bfd2da2ec4b3
Those who beat their swords into ploughshares will till the soil for those who did not. – Benjamin Franklin	5a69709df2ffa9818171b70ed4dfc59b
For centuries, the battle of morality was fought between those who claimed that your life belongs to God and those who claimed that it belongs to your neighbours – between those who preached that the good is self-sacrifice for the sake of ghosts in heaven and those who preached that the good is self-sacrifice for the sake of incompetents on earth. And no one came to say that your life belongs to you and the good is to live it. – Ayn Rand	3e0133cec9e5a0f364232bf12f330353
Socialism in general has a record of failure so blatant that only an intellectual could ignore or evade it. – Thomas Sowell	fec766a206192474b3e1075800346e88

A schematic diagram of one of the simplest hashing algorithms is shown in Figure 1. MD5 is not currently recommended for mission-critical systems because viable collision attacks have been discovered, but it is still used a lot, for example in scientific computing.

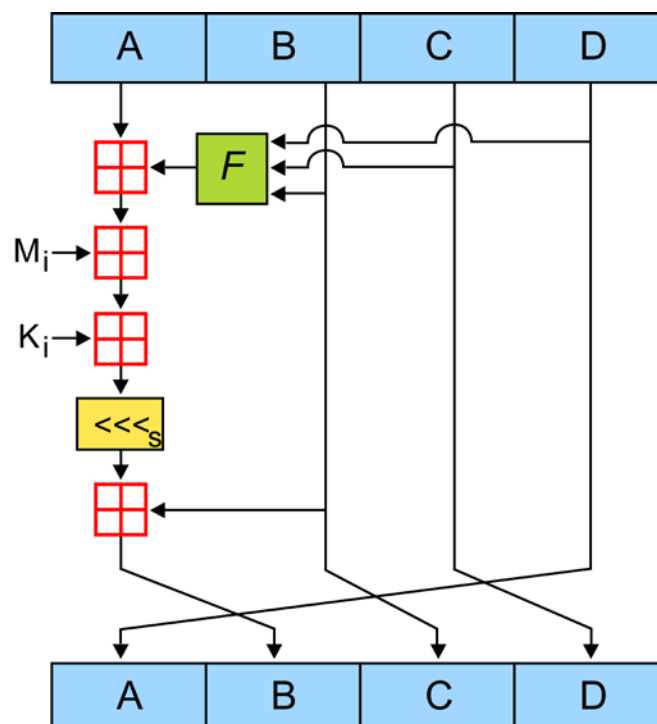


Figure 1. A schematic diagram of a single step of the MD5 hash function. The blocks A, B, C, D are all 32 bits long and start from fixed constants. The $[+]$ symbol denotes addition modulo 2^{32} ; $\lll_s$ symbol denotes bit rotation by s places (the value of s changes at each step). F is a complicated combination of Boolean functions that changes at each step. K_i are 32-bit constants that change at each step. The message is fed in 32-bit blocks into M_i variable. Source: Wikipedia.

Based on the properties listed above, hash functions have the following applications:

1. Data integrity control: a hash string match guarantees that the data has not deteriorated and has not been tampered with. The hash string also serves as a “fingerprint” – existence of another dataset with the same hash is very unlikely.
2. Secure storage of passwords: hash functions are virtually impossible to invert. Storing a hash would therefore allow the password to be checked, but would not expose the password if the server is compromised. This is how passwords are stored.
3. Fast database searching: it is much faster to find the hash string of a large file than to look up the content of the file. Hash table search is implemented in all modern databases.
4. Cryptocurrency mining: it is impossible to predict or engineer the value of the digest – a hash matching some pre-defined pattern must therefore be laboriously obtained by checking a large number of messages. This is the basic principle behind cryptocurrency mining.

Modern hash functions (SHA-2, Whirlpool, RIPEMD) are so secure that US intelligence agencies use them for fingerprinting documents classified Top Secret and above.

4. Digital signatures

If an attacker modifies a file, and also updates the hash to match the new version, hash-based integrity control would not detect the modification. A way of identifying the creator of the file is also often required. The tool that provides this functionality, and also has the useful properties of hash functions listed above, is called *digital signature*.

In a simplified form, a digital signature scheme is a combination of the following:

1. A *key generation algorithm* that makes a *private key* (typically two large prime numbers) and a *public key* (their product). It is not realistic to find large primes from their product.
2. A *signing algorithm* that takes the message and uses the private key to generate a digest that is sent with the message. Signing is impossible with the public key.
3. A *verification algorithm* that uses the public key to confirm that the digest and the message match. Anyone with the public key can check the signature.

The following are some of the applications:

1. Digital rights management – modern consoles would refuse to play games that have not been signed by the companies that have created the consoles.
2. Hardware boot security – mobile phones and computers would refuse to start operating systems unless the boot loader has been signed by the maker of the phone or computer.
3. Author identification – because a digital signature can be checked by anyone, but only created by the holder of the private key, it is a guarantee that multiple messages signed by the same key have originated from the same entity.
4. Cryptocurrency transactions – a transaction record in a blockchain is a message signed using the private key associated with a particular account. A cryptocurrency “account” is just a public-private key pair.

5. Authenticity verification – a valid digital signature is a guarantee (in the same probabilistic sense as hash) that the message has not been tampered with since being signed.

In your practical work on modern analytical instruments, you will notice that they either sign the datasets using private keys stored in hardware, or provide an option for the user to sign datasets with their own private key. This is done to maintain traceable and verifiable chain of custody for the data that could potentially be the subject of, for example, a patent dispute.

5. Cryptocurrencies

A basic cryptocurrency is a combination of a digital signature scheme, a *proof-of-work scheme*, and a distributed transaction ledger called *blockchain*. The following operations are essential:

1. Mining – an entity submits a proof of work signed by their private key. One example is a message with a hash that matches a certain pattern (for example, first twenty bits are zero). Because hash functions are resistant to pre-image attacks, such a message can only be found by computing hashes of a large number of random messages.
2. Transaction block creation – the entity that has found the message with the required hash collects outstanding transaction records from the buffer, collates and signs them, and appends the record as the new “block” to the distributed ledger. A reward is typically associated with creating a block, which is credited to that entity’s key pair.
3. Transaction – anyone with credit in their account can create a message, signed with their private key, assigning some of their balance to another account. The message typically carries a fee that is credited to the creator of the block that would include that message into the ledger. The message sits in the buffer until it is picked up and included into a block.

The blockchain is maintained by consensus – the version that is held by more than 50% of the nodes on the network is the current version. Note the lack of central authority in the system.

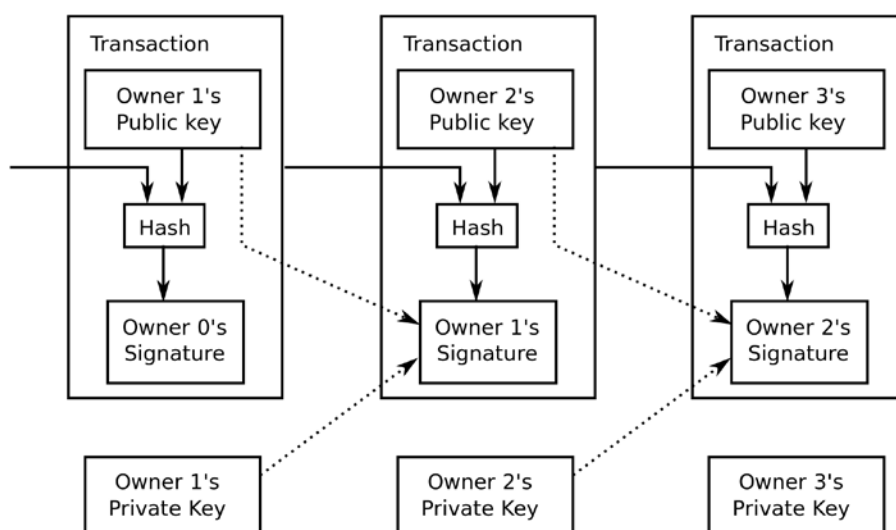


Figure 2. Bitcoin transaction flowchart showing how a combination of digital signatures and hashes makes it possible to create a chain of verifiable transaction records. Source: Wikipedia.