Bitmed Algorithm

Alex Amellal

May 2021

1 Introduction

1.1 Order of difficulty and malicious deterrence

In consideration of any endeavor, the anticipated benefits must in some way match or exceed the estimated costs. Within the context of data security, the barrier to entry for intruders must increase in proportion to the value and quantity of data thereof. Otherwise, sensitive data is put at considerable risk of attracting interest from malicious actors.

This issue has become relevant to recent trends in medical record management, as patient data is increasingly being digitized and stored in centralized databases. The application of new digital record keeping models which offer non-linear increases in security relative to the value and quantity of data will be imperative to enable the fruition of digital medical record keeping to its full potential.

1.2 Patient data without identification is meaningless

Medical records are of little value to doctors, patients or hackers without personal identification. If records could still be stored and identified without the need to include personal identification, the potential for large collections of digital medical records to attract interest from malicious actors would be greatly reduced.

Additionally, patient data presents a major liability for physicians as they are legally bound to the storage and handling thereof. In considering the possibility of storing and retrieving records without direct association to personal patient identifiers, the solution may free physicians from the lifelong responsibility over patient records.

2 Hash functions

2.1 Basics

Hash functions are used to generate data identifiers known as hashes. The data may be of arbitrary length but the resulting hash is of a fixed size.

A simple example of a hash function could be to add all of the alphabetical indices of each letter in a message:

$$Helloworld$$

$$8 + 5 + 12 + 12 + 15 + 23 + 15 + 18 + 12 + 4 = 124$$

In this case, 124 is the resulting hash or identifier for $Helloworld$.

An issue which immediately comes to mind is the risk of two different messages resulting in the same hash, which is known as a collision.

Using our example algorithm above, $Takingflight$ would produce the same hash as $Helloworld$:

$$20 + 1 + 11 + 9 + 14 + 7 + 6 + 12 + 9 + 7 + 8 + 20 = 124$$

2.2 Cryptographic hash functions

Hash functions used within cryptography are known as cryptographic hash functions (CHF). They are expected to meet more stringent criteria, such as minimizing the frequency of collisions.

Another one of these criteria is the implementation of the avalanche effect. That is, the ability for small changes in input to result in dramatic changes in output.

This can be shown using the SHA224 algorithm:

> SHA224("The quick brown fox jumps over the lazy dog") =
> 0x 730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525
>
> SHA224("The quick brown fox jumps over the lazy dog.") =
> 0x 619cba8e8e05826e9b8c519c0a5c68f4fb653e8a3d8aa04bb2c8cd4c

As shown, the addition of a single period to the input makes a dramatic difference in the resulting hash. This property is significantly useful in cryptography.

2.3 Asymmetry

Hash functions are one-way operations. It is computationally inexpensive to generate a hash but incredibly difficult to reverse the operation. As such, hashes can be used to make untraceable or anonymous associations between data entries, such as login credentials on public webservers.

3 The patient-hash

The bulk of medical record retrieval is performed by hospital staff and physicians who rely on personal details to distinguish one patient from another (name, date of birth, address, etc). This approach needlessly places the patients' identities at risk as personal patient data (PPD) is necessarily retrievable in tandem with the respective patient's records.

Conversely, disguising personal patient information within a hash would minimize the risk of direct identity exposure without requiring too many adjustments to the existing medical record interfaces. This is because the same PPD that is normally input can be used to generate a hash on the fly.

Thus, patient data identification by means of a patient-hash would enable both unique and anonymous labeling of data entries.

3.1 Patient-hash generation

The patient-hash is to be generated using an agglomeration of the patient's personal information, such as a combination of their name, date of birth, etc. as input into a designated hash function. Once generated, it becomes possible to label and identify subsequent medical record entries using the respective patient-hash.

$$P = hash(\kappa)$$

where,

P is the patient-hash,

$hash()$ is a designated hash function,

κ is static personal patient information.

When the patient's personal information is known, it is computationally inexpensive to generate and compare the patient-hash to existing records. Otherwise, brute-force or identify theft must be included in a successful attack if conventional security barriers are traversed.

3.2 Patient-hash issues

Creating more than one medical record entry for the same patient is nearly inevitable. And as it is unlikely for the patient's personal information to have changed between times of entry, the same patient-hash will identify both records. This configuration would expose all of the patient's entries if only one patient-hash is compromised.

4 Hash chains

A hash chain is obtained by successively applying a hash function onto an input hash. By using the resulting hash as input for the following hash, a sequence of unique and immutably linked hashes is obtained.

$$H_n = hash(H_{n-1})$$

where,

H_n is the n'th hash in the hash chain,

$hash()$ is a designated hash function.

4.1 Properties of hash chains

Hash chains are immutable in nature, extremely unlikely to repeat and opaque to their inputs. This is due to the nature of their underlying hash functions, taking into account the low probability of collisions, even distribution of outputs and asymmetry respectively.

As such, hash chains are an ideal basis for the generation of hash labels for data records.

5 Enter blockchain

Blockchains are an application of hash chains within data storage or ledger keeping contexts. Although typically decentralized and transactional in nature, some fundamental properties of blockchain enable incorporating hash chains with the patient-hash for anonymous record keeping.

5.1 The block hash

Let 'block' define the smallest unit of data entry, which in this case may represent an appointment or a procedure for a patient. The block itself is labeled with a hash known as the block hash.

The block hash is separate from the patient-hash; the former labels a unique data entry whereas the latter associates every respective entry to a single patient.

5.2 Generating the block hash

Given a sequence of blocks, it is possible to label them using hashes which were generated according to their order. The simplest implementation of this would be as follows:

$$B_n = hash(B_{n-1}) \tag{1}$$

where,

B_n represents the block hash of the n'th block,

$hash(x)$ represents the designated hash function.

Identically to hash chains, every block hash is unique and mathematically 'links' the blocks in an immutable order.

By combining the block hash with the patient-hash, a more sophisticated implementation of the block hash function would be as follows:

$$B_n = hash(B_{n-1} + P_n) \tag{2}$$

where,

P_n represents the patient-hash associated to the n'th block.

This way, if the block order and the patient-hash is known, it is possible to find blocks which belong to a specific patient without storing their patient-hash in plain sight.

5.3 Block identification

By labeling records with the block hash label instead of the patient-hash, blocks may not be identified by direct comparison with the patient's identity. Instead, the block's association to the patient-hash must be inferred.

Let P' be the patient-hash whose associated blocks are desired to be identified. Using equation (2):

$$B'_n = hash(B_{n-1} + P') \tag{3}$$

where,

B'_n represents the expected block hash in the n'th position.

Every matching instance of B_n and B'_n would signal an association to P' such that $P_n = P'$, thus identifying block hashes which were generated using the target patient-hash.

5.4 Measuring the difference

Certain assumptions about an intruder make it possible to quantify the relative improvement in security a new system can offer. If the assumptions are chosen to neutralize the previous system, the net difference is revealed.

Although the patient's identity is concealed within the patient-hash, the information used to generate it is by no means concealed in the real world. It would be unwise to treat the patient-hash alone as a significant barrier to entry, so it represents the 'zero-point' for this comparison.

Assumptions:

- The intruder has made it past conventional cybersecurity measures;
- The blockchain and its contents are exposed;
- The block order is known;

Given these assumptions, the computational difficulty of identifying every block associated to a target patient-hash is:

$$T(n) = O(n) \tag{4}$$

where,

$T(n)$: represents the time complexity of the operation.

That is, for every block in the blockchain, one hash operation is required to verify its association to the target patient-hash.

Although this does represent an improvement over the designated 'zero-point', the potential value of an attack grows proportionately to the computational difficulty. This is not sufficient to deter an attempt to uncover the underlying patient-block associations.

6 The Oracle

The practical implementation of a theoretical model always introduces limitations.

6.1 The block file name