Mytitle.com

Jan Záveský, Vít Krajíček, Eliška Szopová

August 3, 2023

Abstract

This document describes methods used by Mytitle.com to provide users with notarization and timestamping services and products. It is aimed at non-technical users yet it should be readable by technical and expert level readers without significant pain.

Mytitle.com is a timestamping and notarization service based on properties of blockchain technology and cryptography in general – properties of major blockchain networks allow Mytitle to issue a 'certificate of holding of digital file' which serves user as a solid indirect proof of authorship in case of intellectual property lawsuit. This certificate serves as a standalone printable document which does not necessarily need any successive action and/or assistance of Mytitle.com service to prove the holding of specific file at specific time by the user as all necessary information to prove it are contained. To prove the validity of the claim only the certificate, the original file and an access to the blockchain are needed – any third party e.g. the court can inspect the validity of the certificate. Though Mytitle.com provides a public validation interface as a convenience. Mytitle.com also provides a disaster recovery backup of the files in encrypted form as it is critical to the functionality that the file stays completely unchanged from the moment of timestamping.

The innovation of the Mytitle.com service lies in combination of known technologies with some added new approaches in user centered product usable routinely by non-technical users which is truly unusual in the field.

1 Introduction

There are many use case scenarios when one ore more participants need to prove that they actually held the digital file of specific content at specific date and time – solid and easy to exhibit proof of their claim is invaluable for them at the moment. Following are some of them:

- Alice is first grade student making the homework. Eve is a bully who steals the homework and claim the authorship. Bob the teacher pragmatically sides to Eve – both say that it's their work and Eve says it much louder than Alice.
- Alice is a screenwriter working with a large studio, she submit her screenplay to Eve the producer. Eve decides to decline the screenplay. In one year the studio come up with successful movie which is surprisingly similar to Alice's work, but Eve decides to ignore Alice's claims because she knows that Alice can't prove her authorship easily and she would need to go to legal war which given the lack of evidence and power of the studio she would lose.
- Alice is executive of PepsiCo and she is responsible for negotiations of a merger of Coca Cola company. It is a really long and complex process with high stakes. She needs to rely on law firms. The shape of documents she needs to sign change quickly and unpredictably and at some point she signs something she really shouldn't – a document which draft she checked ten times yesterday with her lawyers, but today there is unexpected change she doesn't have a chance to discover in these hundreds of pages.

The first case even if looking childishly states the main problem: how support an authorship claim and provide a solid evidence:

- routinely: at a low cost and instantly
- easy to obtain (first-grader should be able to do so)
- easy to exhibit and vindicate

The other two even if looking almost the same as the first one are stressing the need for solidity of the evidence. The claim should not be backed by Mytitle.com – a startup for profit company. The evidence should stand on it's own supported only by major public blockchain and mild use of cryptographic primitives.

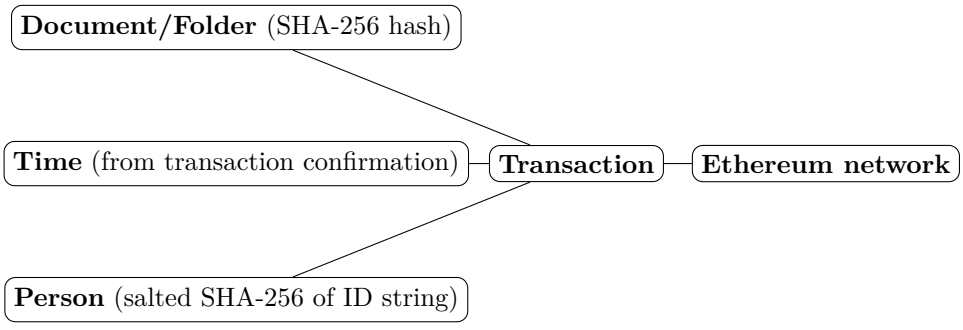
2 Making relation between the holder, the document and the time point

The core principle is based on the trivial assumption that what's needed to provide user with solid evidence of holding a file at a specific time and date is immutable **record** of properties of the situation – in another words a record which would **relate** the **document**, the **person** and **time point**, a record which no one can feasibly tamper with. At the same time it is critical to the user not to leak the document contents nor their own personal ID data. As well there is the need to place all these information or at least something which points to them publicly so the record is **inspectable** by everyone without the need to rely on Mytitle.com.

Our solution to this problem is simply to embed the following payload into the blockchain transaction:

- the document's UUID
- the document's SHA-256 hash
- the salted SHA-256 hash of 'id string' (which itself is a data structure containing users identification details, optimized for portability and human readability)

Then immediately commit the transaction so when the network confirms it there is no feasible way to remove it or alter it. It is there, with the timestamp to the end of the lifetime of the blockchain itself. So it can be said, that the evidence then is only as solid and trustworthy as one of the top major blockchains which drive multi-billion economy worldwide. Mytitle.com is compatible with both Bitcoin-based and EVM-based blockchains. In the case of Bitcoin-based blockchain (we use Litecoin specifically), we use the OP_RETURN opcode to embed the data in the transaction. In the case of EVM blockchain, the data is stored as transaction input data.



3 The Document

As it's been said: to obscure the **contents** of the document and at the same time to establish **public and solid relation** to it, we derive a cryptographic hash from its contents using SHA-256 hash function from SHA-2 family[1]. Cryptographic hash functions are really handy for this purpose as one of their core properties is their one way nature: it is relatively fast and easy to compute cryptographic hash from given data, but it is not feasible to revert the data from the hash. This is important to stress because hashing and encrypting are two completely different concepts often misunderstood and interchanged. There is no feasible way to extract the contents of the original document from the hash (as opposed to encryption). The resulting hash can be understood as a fingerprint of the person: you can't tell the name or a nature of the person from their fingerprint, yet you can identify the person if you compare their fingerprint with the one given. The length of the hash is fixed – it is same length for few kilobytes document as for several gigabytes it doesn't provide any hints about the nature of the document.

4 The Folder

A folder is an entity that contains documents or more folders. Hash of a folder is computed by hashing the hashes of the items inside, in sorted order. The result is a form of a Merkle tree. This method is efficient since the contents of the documents don't need to be hashed again. It also takes into account the structure of the folder, since the resulting hash depends on in which folder the documents are.

5 The Person

As an identification of the holder we use so called 'ID string' a string of characters which is short enough to fit in QR code yet describing the holder sufficiently. This is information unique to the user – and in a readable, unobscured form it appears **only** on the certificate – similarly to the document we need to obscure the persons identity to disable any unwanted analysis by adversary. Since 'ID string' is much shorter than regular document and parts of it could be guessed, to completely mitigate any possibility of so called rainbow table attack[2] we need to employ a technique called salting – we append so called cryptographic salt to the 'ID string' before hashing. The salt is random string of characters unique to the document making the hashed string 'random enough'. The salt value is also provided on the certificate so given the 'ID string' and 'salt' one can compute the resulting ID hash which is present in the Ethereum transaction as a 'relation to the holder'.

[1] https://en.wikipedia.org/wiki/SHA-2
[2] https://en.wikipedia.org/wiki/Rainbow_table

Example of a plaintext ID string:

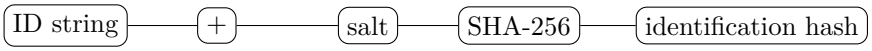
```
ID Address: 221b Baker St,
London NW1 6XE, UK; ID card:
passport, 983745238750,
2025-04-28; Name: Sherlock
Holmes; Email:
admin@therealsherlockholmes.me
```

Example of a cryptographic salt:

```
6b5d402d3665167fe7e14fc17039ebd0
```

Example of a resulting hash

```
1a61d1ba223ba7e2712e6ce32bfa1a14f2ac363c83fbd129f4c67891d639d41e
```



A solution to ambiguity of character representation in PDF and printed form of certificate

Everything above would be perfectly fine unless we need **exact** 'ID string' to be readable from PDF and from the possible printed out page of it to be able to verify the identification hash really belongs to given 'ID string'. Unfortunately there are several problems to it, above all:

- various non-printing characters
- composed characters
- exotic UTF-8 entities from galaxy far, far away

As an elegant and unambiguous solution we use QR code to make the 'ID string' readable – including non-printables, linebreaks, whitespaces of all kinds, emojis and/or exotic characters. So the resulting identification really matches, without any exceptions. Because a change to a single character would result in completely different hash – which would break the evidence we've built.

6 The Time

User receives the certificate **after** transaction is confirmed by the network. The time of the confirmation can be checked independently with the blockchain explorer like etherscan.io (which is third party to Mytitle.com)

7 The Result – The Certificate

The certificate comprises of two pages, the first one (displayed) aggregates all information necessary for a) validating b) brief explanation. The second page contains 'General Terms and Conditions of Mytitle'.

Certificate of Holding of Digital File

Mytitle[MY]

File name:

voynich_draft.doc[MY]

1,233,154 Bytes

Author:

Sherlock Holmes

Address:
221b Baker St
London NW1 6XE, UK

Personal ID:
Passport
Number:
983745238750
Valid until:
2025-04-28

Submission date:
July 23, 2019
09:40:02 UTC

Verify file authenticity at:

https://mytitle.com/verification/ef16c3c9-af63-40aa-9865-e9a05bc21f04

Blockchain Transaction Data:

Blockchain Transaction Hash:
0x60e59cdaac7618dfe4bded81b053d3e171b0aaca0d8e1145d314f5fe2da25aff

File Cryptographic Hash (SHA-512):
d54dc738136035acb64e3b182683c317eba8cc82926965a2daf2c2652429021d8608dccc02d96ad0bfe09fa12d9f0828e62d235cffcc2500dcf5b9eaf6da318f

Blockchain:
Ethereum Mainnet

Blockchain ID String:
ID Address: 221b Baker St, London NW1 6XE, GBR; ID card: passport, 9832475023987438, 2025-04-28; Name: Bioscop Pleograph; Email: admin@therealsherlockholmes.me

Cryptographic Salt:
6b5d402d3665167fe7e14fc17039ebd0

Resulting ID Hash:
5dc667810c760b87809b1e2e32068c40953174ebd7cdd547ac9602d05457fffd

* The resulting ID hash was written into a blockchain transaction and has been derived using the following formula:

SHA-256([ID] + [Salt])

where:
[ID] represents Blockchain ID string
[Salt] represents Cryptographic salt
SHA-256 is the SHA-256 hashing function

Use this QR code to read the Blockchain ID string data in case of ambiguity.

This is to certify that the person identified to Mytitle.com with the email address admin@therealsherlockholmes.me has uploaded a file to Mytitle.com with the digital fingerprint (listed above) with the intent to verify their possession of the file at the noted time and date above.

Mytitle.com has derived a cryptographic hash (SHA-512) which serves as a digital fingerprint of the file and had submitted this digital fingerprint into the Ethereum Mainnet blockchain with transaction:
0x60e59cdaac7618dfe4bded81b053d3e171b0aaca0d8e1145d314f5fe2da25aff

This certificate validates that the person identified to Mytitle.com had this digital file in their possession at the time the cryptographic hash was issued.