Chaining up Time

Bitcoin and Long-Term Validity of PKI Timestamp Tokens

Emanuele Cisbani
ecisbani@intesigroup.com
August 25, 2020

Abstract. Timestamping is a technique used to prove the existence of certain digital data prior to a specific point in time. With the digitalization process of the economy, timestamping has become an important technique to ensure the integrity of digital data for a long time period.

The most popular timestamping scheme and service, based on Public Key Infrastructure (PKI) technology, still has some shortcomings in terms of long-term validation. Since 2009 Bitcoin has been creating a thermodynamically immutable register. That register, aka blockchain, can be used as well to prove the existence of certain digital data prior to a specific point in time, and without any predefined limitation for long-term validation.

This paper describes the aspects that make timestamping on the Bitcoin blockchain complementary to commonly widespread Time Stamping Authority services, which are standardized and legally recognized; and proposes some technically viable solutions to integrate the two timestamping schemes.

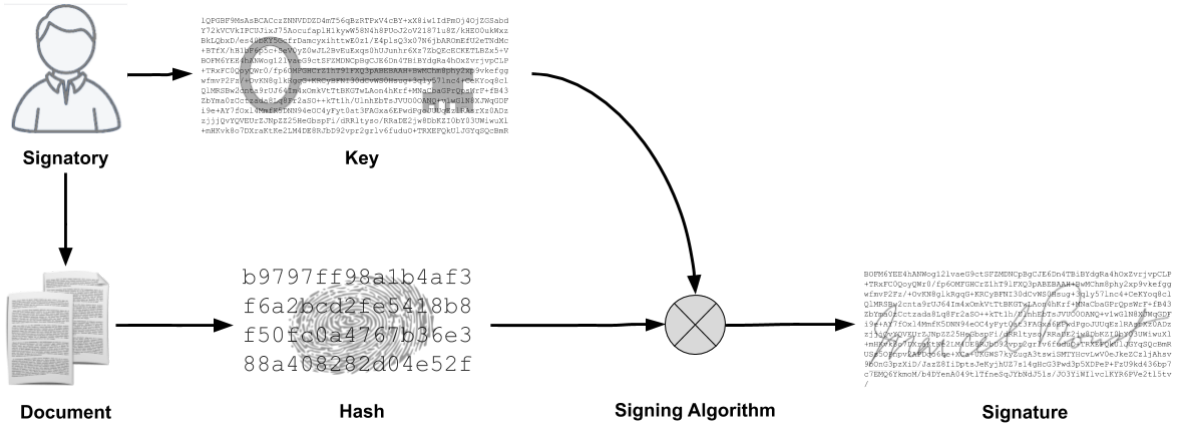
1. Time and Signature

Nowadays, documents and transactions are increasingly digitally processed and recorded. However, it is not easy to detect alteration in digital documents, files or messages. In order to use digital documents as securely as paper based documents, it is necessary to guarantee their integrity and authenticity over time.

The electronic signature fulfilling this purpose has reached wide circulation in recent decades. Without going into the technical aspects of electronic signature, it is possible to illustratively conceive cryptographic keys as passwords - sequences of bytes whose length makes it essentially impossible to guess them by trial and error. The hash too is a sequence of bytes obtained by a calculation resulting from the data we want to sign (a file, a document, a message). The characteristic of a hash is that it identifies the original document in a univocal way from a practical point of view: creating a second document with the same hash would in fact require an unfeasible calculation time.

The electronic signature is the result of a cryptographic calculation combining the private key of the signatory and the hash of the document. Only the person who owns the private key is able to generate that specific sequence of bytes we call the electronic signature of the

document. For this reason, particular attention is paid to the custody of cryptographic keys using specially designed electronic devices according to the best security standards.

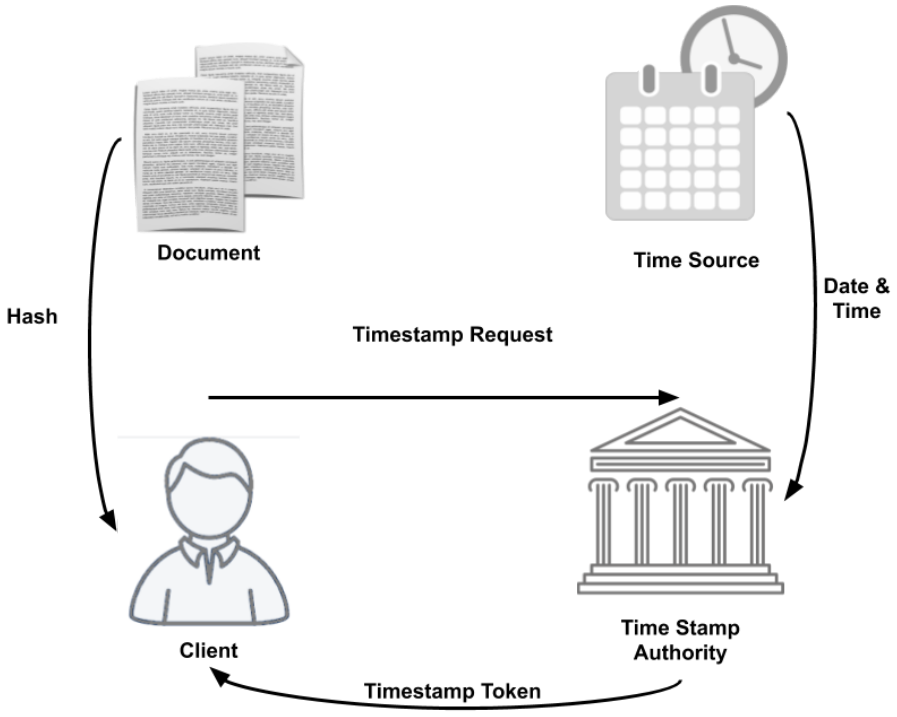


There are two quite different kinds of electronic signature. On the one hand, there is the Public Key Infrastructure (PKI) scheme regulated by the Internet Engineering Task Force (IETF) [RFC5280, RFC5652] and based on the establishment of the Certification Authorities (CA). These are trust entities that offer the registration service of personal identification data associated with public keys by issuing certificates signed by the staff in charge of the CA. On the other hand, there is the Pretty Good Privacy (PGP) scheme rooted in libertarian and Peer-to-Peer (P2P) culture. It is based on the Web of Trust model adopted by online communities, whereby anyone can certify - with an electronic signature - the personal data associated with the public keys of someone they trust or know personally.

Over the years the PKI electronic signature has had a great diffusion in the commercial field, thanks to the introduction of regulatory standards, both at a technical and legislative level. PKI technology has adequately started to address the problem of time validation by introducing trust entities - the Time Stamp Authorities (TSAs) - whose task is to issue authenticated time certificates, uniquely associated with any data in digital form [RFC3162].

These certificates are essential to sort the expiration or revocation of the key used to perform the electronic signature. As a matter of fact, at the occurence of one of the two events, the exclusive control of the key is no longer guaranteed - it becomes thus essential to know whether the signature had been affixed during the validity period. And this is precisely the task performed with the timestamp tokens issued by a TSA.

For simplicity's sake, a time certificate can be represented as a message electronically signed by the TSA, containing the date and time detected by the TSA at the time of signature and the hash of the data whose existence you want to prove in that specific moment. In the following diagram the client sends to the TSA a Time-Stamp Request (TSR) containing the hash of the document. The TSA instantaneously takes date and time from a trusted time source and signs it along with the received hash. The result is returned to the sender in the form of a Time-Stamp Token (TST).



In an in-depth appraisal of Time Stamping Schemes conducted in 2001 by Masashi Une [IMES], PKI TSA was evaluated as one of the most desirable schemes in terms of security against alteration of a time stamp. However, the PKI timestamp has limited effectiveness when it comes to expiration and revocation. The problem is shifted from the signatory's key which usually expires after a few years to the TSA key which usually expires after ten years - but can be revoked before the deadline. This situation brought about some solutions (DSS long term validity) [ETSI.TS.102.778-4] aimed at least at mitigating this inconvenience by extending the validity of TSA timestamps.

The PGP scheme, at the same time, has had great diffusion in the technical field, especially in the context of Free and Open Source Software (FOSS), and represents a de facto standard for signing messages in the mailing lists of developers, and for signing the commits on project repositories and software packages. However, until 2016 the P2P world based on the network-of-trust did not give rise to a solution for time stamping similar to the TSA of the PKI scheme.

2. (Block)chaining up Time

"The solution we propose begins with a timestamp server."
Satoshi Nakamoto, 2008

Working in a company that specializes in the development of a cryptographic engine and offers PKI solutions, I could not help but look with great interest at the increasing consolidation of Bitcoin, not only as a digital asset, but - most importantly - as a technology largely based on cryptography. Moreover, the Bitcoin Distributed Ledger[1] (aka blockchain) is based on a distributed trust scheme which significantly increases security, as already noted by Haber and Stornetta in 1991 [HaberStornetta]. The blockchain can be seen as an untrusted logger - serving a number of clients who wish to store their events in the log - kept honest by a number of auditors who will challenge the logger to prove its correct behaviour [CrosbyWallach].

But there is another aspect that caught my attention. The production of a reliable distributed ledger such as the Bitcoin blockchain is based on the construction of an ordered data sequence which in fact constitutes a reference time axis for transactions. But such time reference can also become useful in solving the problem of file dating. Basically a time attestation on the Bitcoin blockchain can benefit from a resilient and decentralized system, without a single point of failure, which aims to survive perpetually.

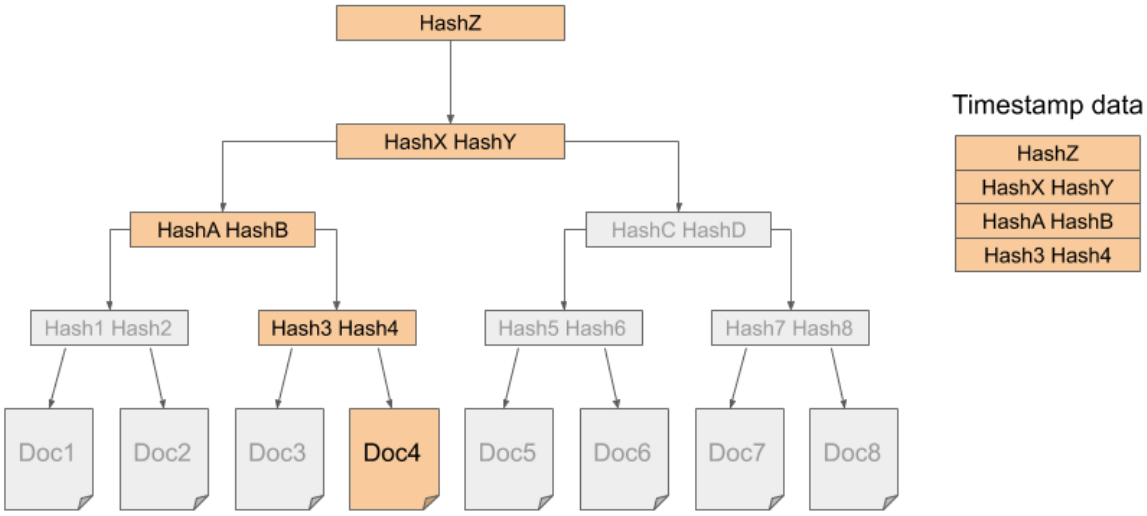
One of the forerunners who had this intuition was Peter Todd, one of the developers of Bitcoin at that time, who in 2016 published OpenTimestamps (OTS), a software[2] that provides the time attestation service on the Bitcoin blockchain. In an article, shortly after the publication of the alpha version of the software, Todd highlighted how the service can be particularly useful, in the world of PGP electronic signature, to fill the gap due to the absence of a time attestation, which is necessary for the validity of the signature in case of revocation or expiration of keys [Todd].

Some characteristics that substantially differentiate OTS from TSA timestamps are more easily understood in the light of the former's operating mechanisms. An OTS server is

[1] Various definitions of blockchain and distributed ledger technology exist, and some of these stress different technical features. Given the nature and scope of this document and the lack of definitional consensus I chose to use the term as defined by UK Government Chief Scientific Adviser [UK-GCSA] "A distributed ledger is essentially an asset database that can be shared across a network of multiple sites, geographies or institutions. All participants within a network can have their own identical copy of the ledger. Any changes to the ledger are reflected in all copies in minutes, or in some cases, seconds. The assets can be financial, legal, physical or electronic. The security and accuracy of the assets stored in the ledger are maintained cryptographically through the use of 'keys' and signatures to control who can do what within the shared ledger. Entries can also be updated by one, some or all of the participants, according to rules agreed by the network".

[2] https://opentimestamps.org

composed of two subsystems: the calendar and the stamper. The calendar deals with the collection of requests from clients and the preservation of proofs produced by the stamper. These requests are aggregated through the construction of a Merkle Tree [Merkle] in which the received hashes are associated two by two, calculating the resulting hash. Iterating this process eventually leads to a single hash called the Merkle Root Tree.



With the Merkle Tree data structure the calendar accomplishes the aggregation process in a very efficient way by aggregating many requests and submitting periodically to the stamper only the root hash of the tree. The second subsystem is called stamper and - periodically - is responsible for signing and sending a particular transaction containing the Merkle Root Tree to a Bitcoin node. When this transaction is recorded in a block of the Bitcoin blockchain, the time attestation of all the hashes collected by the calendar server is created. It should be noted that in this way, at the cost of a single transaction (a fee of a few euro cents), thousands of files can be timestamped - be it documents or messages of any size.

Since the consensus on the status of the Bitcoin blockchain is statistical, a Bitcoin transaction takes usually an hour on average (the mean time needed to record six blocks) before it can be considered definitively confirmed. At that point in most cases the thermodynamic cost to rewrite those six blocks already exceeds its usefulness. This means waiting for a significant period of time between the issuance of the time attestation in the form of a promise and its upgraded version in the form of proof by the OTS server.

3. An Irreducible Complementarity

If we want to make a comparison between the two schemes, something I would define as an irreducible complementarity emerges quite clearly. This complementarity is due to the profound difference existing between two underlying philosophies - the centralization of the institutional world versus the decentralization of the peer-to-peer world.

PKI-TSA	**Bitcoin-OTS**
A standard with legal value that has been around for a long time	Not yet a standard
A service that depends on a central trustee	A service based on a permissionless, resilient and decentralized system, without a single point of failure
Verifying a timestamp requires the involvement of the original issuing TSA	Anyone can verify the timestamp autonomously running a Bitcoin full node or connecting to any trusted block explorer
Usually a TSA undertakes to guarantee the validity of a timestamp for no more than twenty years	There is no predefined limit to the validity of an OTS timestamp, the system aims to survive perpetually
The service of qualified TSAs usually has a specific cost per single attestation	The service is free of charge for clients and the cost for the provider is very low (a negligible fee for a small Bitcoin transaction, approximately every hour, no matter how many requests are aggregated each time)
The timestamp issue is immediate	The time attestation in the form of a promise is immediate, its upgrade takes about an hour
Timestamps can reach fractional second precision	The time attestation proves data existence only in an interval of hours[3]

Note that even the technical differences derive more or less directly from a different underlying philosophy. But since both schemes seem to be unable to assimilate each other, a solution combining the benefits of both PKI-TSA and Bitcoin-OTS is certainly worth exploring.

[3] The timestamp of Bitcoin nodes can be potentially manipulated in hours [Boverman].

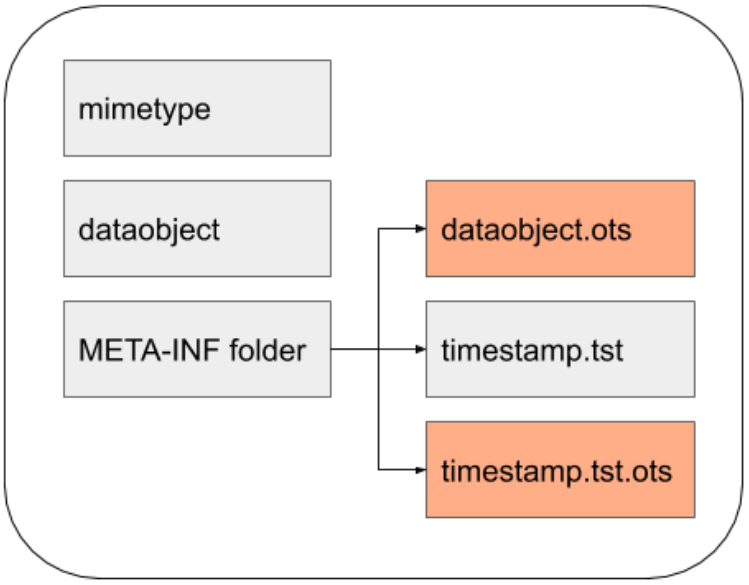
4. Proposals

A possible solution came to me by serendipity about two years ago while I was looking through the specifications of the ASiC (Associated Signature Containers) file format [ETSI.TS.102.918]. It is a simple archive compressed with the common zip algorithm and containing a META-INF directory, similar to the jar format commonly used in java programming. In the technical specifications it is clearly stated that the META-INF directory, in addition to the timestamp token issued by the TSA (a file named timestamp.tst), may also contain other files for different application purposes.

> "Other application specific information may be added in further metadata objects contained within the META-INF folder."
>
> [ETSI.TS.102.918 - Cap. 5.2.2 "Contents of the container"]

Last winter I decided to make TimeBags[4], a software that can be seen as a simple proof-of-concept integration of the two timestamping schemes. TimeBags uses the ASiC format to store files together with the traditional token timestamp issued by a TSA and the Bitcoin time stamp issued with OTS.

The ASiC compliant structure of a file generated with TimeBags is represented in the following diagram.



The most interesting file is *timestamp.tst.ots* which contains an OTS timestamp token certificate provided by the TSA. It is particularly useful when the signature of the timestamp token has expired by exceeding time limits, or due to a weakness or obsolescence of the algorithms, or when the key has been revoked. In this case, the OTS time attestation can demonstrate the existence of the timestamp token before these disabling events - thus demonstrating its full validity.

[4] https://timebags.org

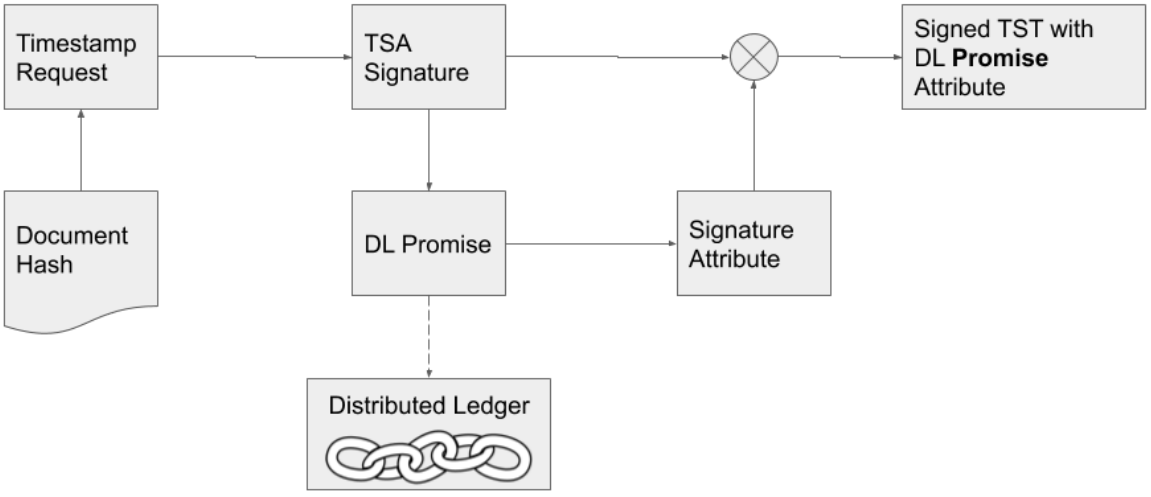
The *dataobject.ots* file contains the OTS time attestation for the original document or archive whose existence you want to certify on the current date. This attestation is useful only in extreme cases such as when the TSA has ceased to exist, or if for some other reason it is no longer possible to validate the timestamp token. In this case this second attestation would still allow you to demonstrate the existence of the document on the date of the Bitcoin transaction.

The TimeBags application is very easy to use. It aims to be a free tool for exploring and disseminating - at individual level - these two time attestation schemes. In the professional field, however, the ASiC format is not very widespread. Tools commonly used to affix a time stamp usually integrate TSA services in the PKI electronic signature process.

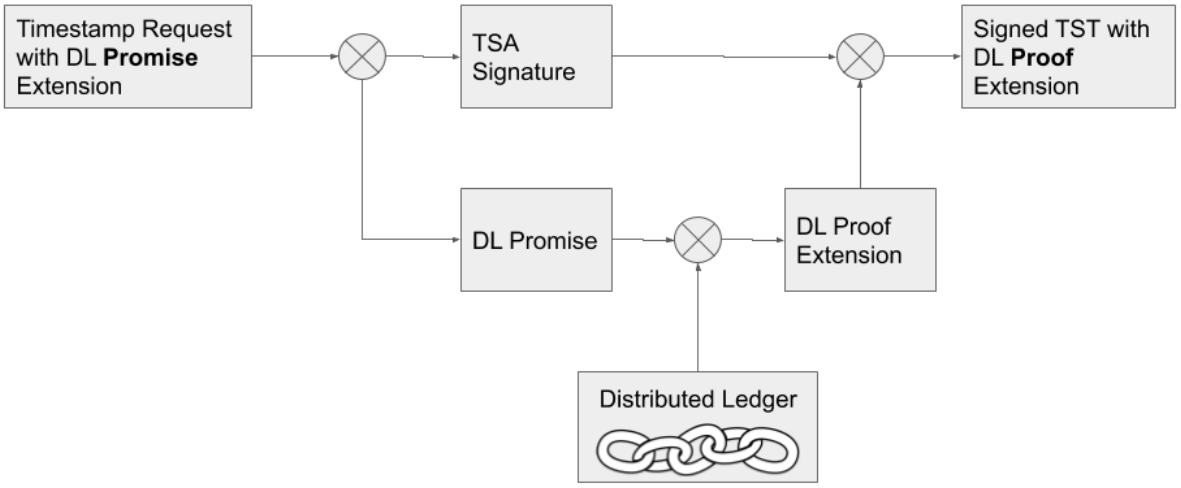
For this reason our Intesi Group team (consisting of Giuseppe Damiano, Daniele Ribaudo and myself) drafted in the first half of this year a proposal integrating those timestamps generated by a Distributed Ledger (DL) scheme within the timestamp tokens provided by a traditional TSA.

Our decision to refer to a generic DL is motivated by the need for a neutral protocol - open to any up-and-coming technology in the blockchain world. Among these, OpenTimestamps on Bitcoin is certainly one of the most interesting technologies, at least for the resilience demonstrated in more than 11 years by its blockchain. However, we cannot exclude *a priori* the emergence of other solutions with more functional features for time validation.

Our proposal [DLTS] was submitted as Internet-Draft for the evaluation of the Internet Architecture Board (IAB). We would like to add the temporal attestation on DL as an (unsigned) attribute of the signature used by the TSA to issue the PKI temporal attestation.



In the case of permissionless DLs such as the Bitcoin blockchain, we proposed to use a non-critical extension of the timestamp request in order to convert the promise into a proof.



The possibility to add attributes and extensions to the PKI timestamp already complies to regulatory standards. We are thus empowering the traditional PKI timestamp with an indefinite validity offered by Distributed Ledgers in a single solution.

This solution has several advantages. First of all, an indefinite extended validity that saves reiteration costs for retrieving, processing and timestamping. Secondly, investments are safeguarded by using PKI standard formats and interfaces - without extra charges for adapting existing processes. Since we are using a standard format, the timestamp token is parsable by any software currently available on the market. This is a brand new enhancement, and for this reason it will be momentarily ignored by the current software in the short term. However, in ten years time, when the PKI timestamp expires, it will become crucial. Thirdly, this scheme is open and very secure, it follows IETF standards & EU regulations, and can thus be easily adopted by any Qualified Trusted Service Provider [eIDAS] in all EU member States and beyond. Finally, it is lock-in safe thanks to the autonomous timestamp verification.

5. Conclusions

I have tried to show how the currently available time validation schemes are complementary and provide several advantages. However, combining their respective features in a single scheme seems to be unlikely. For this reason, we are proposing two solutions to integrate those timestamps issued by consolidated trusted timestamping providers with timestamps issued by a service like OpenTimestamps on Bitcoin - or any other timestamping service on Distributed Ledger.

The first integration proposal has the advantage of using a standard format such as ASiC, which is not widespread and can be more suitable for occasional use. The second proposal was submitted to the IAB to enhance the current standard timestamp token. It can extend its validity without having to change any software currently in use. Both solutions extend the validity of current token timestamps without a fixed time limit, adding to the PKI scheme a *distributed security layer* coming from the P2P world.

Acknowledgements

I am thankful to the university lectures given by Ferdinando Ametrano not only for the discovery of OpenTimestamps, but also for a deeper and clearer understanding of the magical combinations existing between game theory, monetary theory and cryptography, which helped Nakamoto in the creation of *digital gold*. I would like to thank Marco Zoppas for the revision of the English text, a language that I have been practicing for many years with uncertain results. Last but not least, I must thank my wife who pushed and encouraged me to write this article, winning my "healthy animal laziness".

References

[Boverman]
Boverman A., "Timejacking & bitcoin", Culubas blog, 2011 <http://culubas.blogspot.com/2011/05/timejacking-bitcoin_802.html>

[CrosbyWallach]
Crosby, S. and D. Wallach, "Efficient Data Structures for Tamper-Evident Logging", Proceedings of the 18th USENIX Security Symposium, Montreal, August 2009, <http://static.usenix.org/event/sec09/tech/full_papers/crosby.pdf>.

[DLTS]
Cisbani, E., Damiano, G., Ribaudo, D., July 2020, "Distributed Ledger Time-Stamp", IETF Internet-Draft, <https://tools.ietf.org/html/draft-intesigroup-dlts-00>

[eIDAS]
The European Parliament And The Council Of The European Union, "Regulation (EU) No 910/2014", July 2014, <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32014R0910=EN>.

[ETSI.TS.102.778-4]
European Telecommunications Standards Institute, "Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term - PAdES-LTV Profile", December 2009, <https://www.etsi.org/deliver/etsi_ts/102700_102799/10277804/01.01.02_60/ts_10277804v010102p.pdf>.

[ETSI.TS.102.918]
European Telecommunications Standards Institute, "Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC)", June 2013, <https://www.etsi.org/deliver/etsi_ts/102900_102999/102918/01.03.01_60/ts_102918v010301p.pdf>

[HaberStornetta]
Haber, S. and W. S. Stornetta, "How to Time-Stamp a Digital Document", 1991, <https://www.anf.es/pdf/Haber_Stornetta.pdf>.

[IMES]

Une, M., "The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies (2001)", 2001, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.7486>.

[Merkle]
Merkle, R. C., "Secrecy, authentication, and public-key systems - Technical Report No. 1979-1", June 1979, <http://www.merkle.com/papers/Thesis1979.pdf>.

[Nakamoto]
Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash System", October 2008, <https://bitcoin.org/bitcoin.pdf>.

[RFC3161]
Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, DOI 10.17487/RFC3161, August 2001, <https://www.rfc-editor.org/info/rfc3161>.

[RFC5280]
Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5652]
Housley, R., "Cryptographic Message Syntax (CMS)", STD 70,RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[Todd]
Todd, P., "Solving the PGP Revocation Problem with OpenTimestamps for Git Commits", September 2016, <https://petertodd.org/2016/opentimestamps-git-integration>

[UK-GCSA]
UK Government Chief Scientific Adviser, "Distributed Ledger Technology: beyond blockchain", January 2016, <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf>