Exploring Digital Timestamping using Smart Contract on the Solana Blockchain

Jianshu Wang*[a]
[a]School of Information Science and Technology, Qingdao University of Science and Technology, No.99 Songling Road, Qingdao, China 266061
* 1803060219@mails.qust.edu.cn

ABSTRACT

Timestamps establish a means to dictate the existence of a message at a particular moment in the past. Traditional digital timestamping service utilizes Public Key Infrastructure (PKI) and thus requires the presence of a Time Stamping Authority, whose job is to ensure the message's validity. However, with the rise of distributed computing and blockchain technology, it has become possible to obtain timestamps in a decentralized manner, eliminating the need for central authorities. Previous research involves sending transactions or utilizing a smart contract mechanism to store hashes on blockchain and validating the outcome using public ledgers. By prototyping a simple digital timestamping protocol on the Solana blockchain, this article explores the natural advantages of the Solana blockchain for timestamping, implements a smart contract of digital timestamping on the Solana blockchain, and measures its latency, costs, and performance. The initial estimation expects the accuracy of the timestamps produced by the prototype to reach the average sub-second slot time. However, due to network propagation delay, it is not always possible to hit this level. The cost incurred during the process is also analyzed and discussed. Finally, this article highlights the potential ability of the blockchain to provide modern decentralized services.

Keywords: trusted timestamping, blockchain, smart contract, solana, timestamp

1. INTRODUCTION

As manipulating and forging digital documents are significantly easier than modifying physical files due to the volatile nature of digital storage mediums, the ability to verify the integrity and the exact creation time of a given piece of data is frequently proved vital in distinct fields, such as forensics, financial audits, and patent issues. Formerly in legal practices, notaries are used to protect the integrity of documents, along with the time it is witnessed. Even though this approach is still widely adopted, its procedure is over-complex and costly, making it unsuitable for general use.

Fortunately, with the help of computer technology and cryptography, it is now possible to obtain proof of integrity and time by interacting with trusted online authorities. The principle of digital document timestamping was initially discussed by Haber, S. et al. in 1990[1]. Meanwhile, with the further development of Public Key Infrastructure, it establishes a platform to create cryptographic-valid timestamps, and the standards for digital timestamping on PKI are formalized by the Internet Engineering Task Force in RFC3161[2] and RFC3628[3], while the latter further defined a role named Time Stamp Authority(TSA). Later the International Organization for Standardization also published ISO/IEC 18014 in 2008[4], forming a general framework for any time-stamping technique. In a typical centralized digital timestamping scenario, a Time Stamping Authority is required to receive the file, register a timestamp from a trusted time source, and digitally sign it[5]. The credibility and accuracy of such timestamps mostly rely on the reputation of said authority.

As the result of the emergence of the distributed computing and blockchain technology since the first appearance of Bitcoin[6] in the 2010s, an alternative path is made possible, and it brought a fashion to eliminate central authorities and use consensus algorithms instead, making it self-governing. The data produced by such a system is also stored across the entire network, to minimize the risk of single-point failure. Blockchain technology enables us to build a distributed and append-only journal that provides the immutability of records, which is invaluable in a network with zero trust provided.

Numerous ways are founded to store redundant data on the blockchain. For example, it is possible to send transactions to a specific address, where the address is calculated by the hash of given data. Later the transaction can be revisited by

Second International Conference on Green Communication, Network, and Internet of Things (CNIoT 2022), edited by Xiaofang Yuan, Proc. of SPIE Vol. 12586, 125860T
 · 0277-786X · doi: 10.1117/12.2667788

browsing the public ledger. Bitcoin also provides a special opcode called OP_RETURN[7] which can be used to put arbitrary data on the blockchain. Ethereum network[8] allows smart contracts, which are tiny programs running automatically and independently by all nodes on the network, that manipulate the state of the blockchain in a determined way[9]. Christidis K. et al. discussed the possibility to utilize smart contracts to store data on the chain. There is also a different approach named InterPlanetary File System[10, 11], which is a hash-addressed distributed file storage.

The credibility of on-chain timestamps is guarded by the strength of the cryptographic hash algorithm and the append-only nature of the blockchain. The blockchain guarantees the order of transactions described in different blocks. Nowadays there exist several digital timestamping platforms on popular networks, notably OriginStamp[12] on the Bitcoin blockchain, and several others on the Ethereum chain implemented using smart contracts. OriginStamp takes a more complicated path, by combining digests of multiple parts of data into a key pair, and depositing dust to the respective address. Due to the current network status of the Bitcoin blockchain, the average time used for a transaction to obtain 6 confirmations takes approximately one hour[13]. To minimize the impact on the blockchain network and increase efficiency, OriginStamp concatenates all requests received within a 24-hour window and submits them at once. This approach, however, negatively impacts the accuracy of the timestamp produced. The option of creating an immediate timestamp exists but comes with a more significant cost than that of combined signatures. Additionally, according to Loop[14], the timestamp on the blockchain is vulnerable to the so-called time warp attack, which significantly defeats its purpose. Other similar solutions exist with different features offered, for example, Catena[15] and CommitCoin[16]. Also, some proposals suggest enhanc- ing the classic timestamping protocol with blockchain technology, through the use of TSA[17]. A significant improvement is proposed by G. Estevam et al. by integrating time oracles into the smart contract, rather than solely relying on block timestamps. It claims an average accuracy of 121.10ms[18]. Other related works are classified in Table 1.

Table 1. How the related work compares to this work. [12, 15–27]

Related Works by Categories	Accuracy			Properties	
Improved RFC3161 Timestamps	Seconds	Minutes	Hours	TSA	Platform
Stavrou and Voas		✓	✓	✓	Bitcoin
Blockchain Timestamps	Seconds	Minutes	Hours	TSA	Platform
Hepp et al.		✓	✓		Bitcoin
Gipp, Meuschke, and Gernandt		✓	✓		Bitcoin
Gipp, Breitinger, Meuschke, and Beel		✓	✓		Bitcoin
Tierion and Inc.		✓	✓		Bitcoin
Factom		✓			Bitcoin
Tomescu and Devadas		✓	✓		Bitcoin
Ali, Nelson, Shea, and Freedman		✓	✓		Bitcoin
Clark and Essex		✓	✓		Bitcoin
Zhang, Xu, Li et al.	✓	✓			Ethereum
Zhang, Xu, Cheng et al.	✓	✓			Ethereum
This work	✓				Solana
Blockchain Improved Timestamps	Seconds	Minutes	Hours	TSA	Platform
Landerreche et al.	—	—	—		--
Szalachowski		✓		✓	Bitcoin
G. Estevam et al.	✓				Ethereum

This work attempts to exploit the Proof of History nature of Solana chain[28], to explore the possibility of building a digital timestamping platform that is both responsive and cost-free.

2. METHODOLOGY

In this section, a protocol is proposed to implement digital timestamping on the Solana blockchain. In Section 2.1, an overview of the involved parties and the protocol is presented. Section 2.2 describes the full procedure of the protocol.

Section 2.3 briefly introduces the prototype implementation used to demonstrate the process. Finally, in Section 2.4, various possible attack vectors are discussed.

2.1 Overview

The ultimate goal of this protocol is to generate verifiable proofs of existence given a piece of data. The users will directly interact with the smart contract on the blockchain, without the involvement of any third parties.

The design divides its users into two distinct roles. Individuals who request timestamps to be included in the blockchain are referred to as committers. The second participants who wish to verify the integrity of given data are referred to as challengers. The smart contract itself maintains an append-only ledger.

The whole process starts with the smart contract creating accounts for committers to hold their signed digests. The account is a mechanic in the Solana blockchain to hold persistent data for smart contracts. Following this, the committers can call the smart contract to include the digest into its ledger. Finally, the challenger can then find the transaction or look up in the committers' account, and check the correspondence and integrity of the document, the signature, and the signer's identity. Figure 1 briefly illustrates the process.

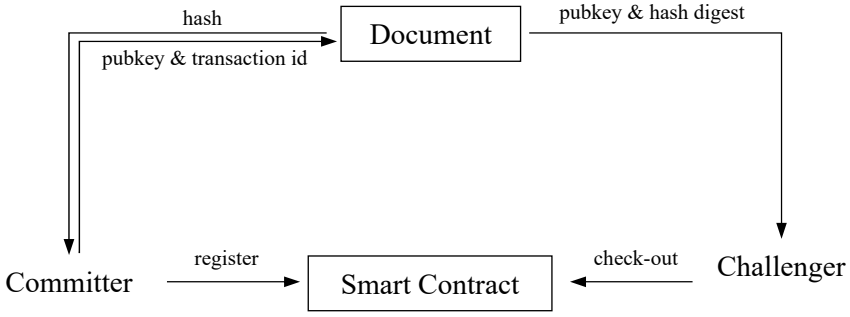

Figure 1. Relationship between the parties involved.

2.2 Procedure

The full procedure is specified below:

- The committer instructs the smart contract to create an account if there exists none. The account A is owned by the smart contract but held by the committer. The committer also needs to pay a space-and-time fee F1 for the account to be persisted as defined by the Solana network. Either a one-time payment to make the account rent-exempt or a limited payment is possible. However, currently the Solana blockchain requires all new accounts to be rent-exempt[29]. A fixed transaction fee $F2$ is also charged.
- The committer calculates a cryptographic hash H of the document and submits it to the smart contract. A fixed transaction fee F3 is charged.
- The smart contract receives the inquiry R(H), combines it with slot ID, and then puts the final result onto the account A keyed by H.
- The committer can choose to attach the transaction ID to the signed document.
- The challenger interacts with the smart contract to retrieve the information of the document from the committer's account by its hash H, along with the slot information the timestamp action is on.

A few assumptions can then be made:

- The ownership of the document belongs to the holder of the account used to request the timestamp, which is the key pair of the committer.
- The integrity of the document can be verified by comparing the hash of the document with the hash stored in the account.
- The creation time is calculated from the slot number and the start time of the corresponding epoch.

2.3 Prototype

A prototypical implementation is developed as part of the project. The prototype consists of two parts: the smart contract and the client script.

The smart contract is a Berkeley Packet Filter (BPF) program written in Rust, to run on the verifier nodes of the network. It provides an interface for both the committers and the challengers to act according to the protocol. The smart contract is mainly responsible for creating accounts for committers, appending data into them, and extracting information based on the request of the challengers. The instructions provided by the smart contract are listed in Table 2.

Table 2. Instructions provided by the smart contract.

Instruction	Description	On-Chain
createAccount	Create an account under the current user	✓
registerDocument	Register a document into a given account	✓
findDocumentByAccount	Look for a document hash in a given account	

The script, written in JavaScript, is mainly used to compose instructions for the smart contract. Both committers and challengers use the script to instruct the Solana client to create the correct transaction for the access to the blockchain.

2.4 Possible attacks

The issue of a full blockchain digital timestamping scheme mainly resides in the stability of the blockchain itself.

Historically, the Solana network experienced a DDoS attack which slowed down the entire network, making time-related applications unreliable. Additionally, the network had a history of clock degradation, leading to a half-hour difference between the wall clock and the chain time.

There is also a risk on data management. Different from Bitcoin and Ethereum, for which the space for smart contracts to store runtime data is infinite and free, in the Solana network a rent has to be paid upfront to keep the data from being purged. In this case, the timestamp records is created by the committers. The accounts used to hold timestamps should be kept rent-exempt if possible.

3. EXPERIMENTS AND RESULT

This section shows experiments conducted to examine the feasibility and cost of the proposed protocol.

3.1 Speed of propagation

This experiment aims to learn the average time used for a transaction to be announced on the blockchain. A tiny program is written to subscribe to the Solana Testnet RPC node and to compare the time a new transaction is sent to the time the validator node learns and broadcasts it.

3.1.1 Test environment

Two participants are used to carry out this experiment. A monitor program and a broadcast program are deployed at Falkenstein (Germany). The broadcast node routinely sends a new transaction via a public validator in Amsterdam (Netherlands) to a designated address and logs the time the transaction is sent. The monitor node maintains a long-lived connection to the Solana-provided node in Amsterdam and subscribes to the destination address for changes. Once the validator node verifies the transaction, a notification is sent to the monitor node; the program logs the moment it sees the message.

The monitor program is a Node.JS script with Solana Web3.js library solely used for listening and logging, hosted on a container created on an AX161 instance provided by Hetzner. The broadcast program uses solana-cli 1.10.31 for communication to the cluster.

3.1.2 Result

The lifecycle of a transaction on the Solana blockchain is illustrated in Figure 2. In order to study the timing characteristics of the transaction, the author instruments each stage and records timestamps in milliseconds during the process.

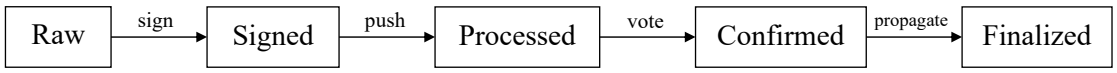

Figure 2. Stages of commitment during a transaction.

After a signed transaction is sent via the JSON RPC API, it is verified by the validator node directly connected to the client. The block is marked processed on the node if it independently verifies the given transactions. As soon as the block receives a supermajority of ledger votes, the status turns confirmed across the cluster. Generally, a status of confirmation is sufficient for user-facing applications to assume the transaction is completed. In stricter situations, a finalized status may be intended to keep the risk of reversion as low as possible. A block is reached its finality if at least one correct validator has rooted the block or a descendant of it.

Table 3 summarizes the statistics of time elapsed from transaction signature to the different stages of the final block across 200 transactions made in this experiment. According to the table, a transaction generally takes 3.97 seconds to be acknowledged first by the remote validation nodes and it takes 5.95 seconds on average to collect enough quantities of votes to be considered safe. An extra 42.18 seconds is required to ensure the cluster will not skip the block. It is worth noticing that the standard deviation also reaches more than 40% of the average, which corresponds to the discovery made by G. Estevam et al.[18] and indicates the transactions on the network are not propagated uniformly. The time used for a block to reach the 'processed' state (3.97 seconds with a deviation of more than 75%) is considered the lower bound to theaccuracy of such timestamping services.

Table 3. Time elapsed from transaction signature to given block stages.

Stages	Avg. (ms)	Std dev	Min	Max
processed	3965	3080	1009	10690
confirmed	5945	2709	1899	10567
finalized	48138	21596	21358	92396

3.2 Cost analysis

This experiment attempts to evaluate the total cost incurred during a timestamping procedure with the prototypical contract described in Section 2.3. The contract is deployed to the Mainnet Beta cluster of Solana blockchain to receive an accurateestimation of the transaction cost.

3.2.1 Test environment

The environment used in this experiment is similar to that described in Section 3.1.1. with the difference that the smart contract is involved, rather than sending only plain transactions. Also, the cluster switches from Testnet to Mainnet Beta.

3.2.2 Result

The total cost incurred by the protocol mainly resides in two parts: the rent and the transaction cost (gas).

Currently, as the Solana Design Proposal states, the rent cost is a fixed value and is subject to changes, and all new accounts are required to be rent-exempt[29], which requires that the full rent-exempt price must be paid upon the account creation. Assuming a record occupies 40 bytes of account space, Table 4 gives an estimation of the rent cost corresponding to account size.

Table 4. Estimated rent cost for a single account. (1 SOL = $41.69 as of Jul. 2022)

Size	Capacity (record)	Rent/epoch (SOL)	Rent-exempt min. (SOL)	Est. Cost
40 bytes	1	0.000003201	0.00116928	$0.05
320 bytes	8	0.000008537	0.00311808	$0.13
4000 bytes	40	0.000078662	0.02873088	$1.20
10 MiB (max)	262144	0.199817492	72.98178048	$3042.61

Moreover, a transaction fee is required to trigger the execution of the on-chain program on validation nodes across the cluster. At present, the cost is a fixed value calculated by the total number of signatures contained in a transaction. Each signature costs an additional 5,000 lamports, which approximately values $0.0021. Because of the fixed transaction fee, the on-chain programs are under the limit of a maximum compute cost[30], which limits the number of total BPF instructions executed.

3.3 Discussion

The experiment in Section 3.1 observes the latency in general transactions on the Solana blockchain. With a closer look at this, the latency can be broken into several parts: network latency (ping), processing time, and inter-cluster communication time. While the latter two are unobservable outside the cluster, the network latency can be measured using ICMP Ping packets and the calculated round trip time (RTT). The average network latency from our server to the cluster is 10.67ms, which means that more than 99.7% of the total delay is spent on processing and propagation. Additionally, it is worth noting that during the experiment, several cluster downtimes were experienced, which indicates that the network stability still requires further improvements.

Section 3.2 presents the fee scheme of the blockchain. The method used by Solana is noticeably different compared to Ethereum. In the Ethereum blockchain, a dynamic gas fee is charged each time. After a transaction is sent, no additional cost is required to store data inside the smart contract. However, in Solana, the transaction fee is fixed, and the rent is involved for the storage.

In conclusion, there are both advantages and disadvantages for the Solana blockchain to be used as a digital timestamping platform.

4. CONCLUSION

As blockchain technology progresses, new chains with the different features are coming up constantly. Compared with classic blockchains like Bitcoin and Ethereum, Solana provides a higher throughput and a more flexible on-chain platform based on WebAssembly and Berkeley Packet Filter (BPF). Recognizing these features, this paper demonstrates a simple digital timestamping protocol deployed on the Solana blockchain and conducts the various experiments to test its perfor- mance. However, there still exists room for improvement. This paper discussed several possible scenarios when timestamps derived from the block time detach from the wall clock, which will cause severe issues for such time-sensitive applications. In order to circumvent this problem, a distributed TSA system can be adopted into the framework by cooperating with the external time oracles to construct a hybrid time source. Additionally, the high cost incurred for storing data on blockchain also poses a significant hindrance compared with Ethereum solutions, which is worth noticing in the future of Solana's development.

REFERENCES

[1] Haber, S. and Stornetta, W. S., "How to time-stamp a digital document," in [Advances in Cryptology-CRYPTO' 90], Menezes, A. J. and Vanstone, S. A., eds., 437–455, Springer Berlin Heidelberg, Berlin, Heidelberg (1991).

[2] Adams, C., Cain, P., Pinkas, D., and Zuccherato, R., "Internet x.509 public key infrastructure time-stamp protocol (tsp)," RFC 3161, RFC Editor (Aug. 2001).

[3] Pinkas, D., Pope, N., and Ross, J., "Policy requirements for time-stamping authorities (tsas)," RFC 3628, RFC Editor (Nov. 2003).

[4] "Information technology – security techniques – time-stamping services – part 1: Framework," standard, International Organization for Standardization, Geneva, CH (Sept. 2008).
[5] Buchmann, J., Karatsiolis, E., Wiesmaier, A., and Karatsiolis, E., [Introduction to public key infrastructures], vol. 36, Springer Berlin Heidelberg, Berlin, Heidelberg (2013).
[6] Nakamoto, S., "A peer-to-peer electronic cash system," tech. rep. (2008).
[7] "Op_return." Bitcoin Wiki (June 2020). <https://en.bitcoin.it/wiki/OP_RETURN> (22 July 2022).
[8] Wood, G. et al., "Ethereum: A secure decentralised generalised transaction ledger," tech. rep. (2014).
[9] Christidis, K. and Devetsikiotis, M., "Blockchains and smart contracts for the internet of things," IEEE Access 4, 2292–2303 (2016).
[10] B., J., "Ipfs - content addressed, versioned, p2p file system," tech. rep. (2015).
[11] Baumgart, I. and Mies, S., "S/kademlia: A practicable approach towards secure key-based routing," in [2007 Inter- national Conference on Parallel and Distributed Systems], 1–8 (2007).
[12] Hepp, T., Schoenhals, A., Gondek, C., and Gipp, B., "Originstamp: A blockchain-backed system for decentralized trusted timestamping," it - Information Technology 60(5-6), 273–281 (2018).
[13] "Average confirmation time." Blockchain.com (2022). <https://www.blockchain.com/charts/avg-confirmation-time> (22 July 2022).
[14] Loop, J., "Bitcoin timestamp security," (2019). <https://blog.lopp.net/bitcoin-timestamp-security/>.
[15] Tomescu, A. and Devadas, S., "Catena: Efficient non-equivocation via bitcoin," in [2017 IEEE Symposium on Security and Privacy (SP)], 393–409, IEEE Computer Society, Los Alamitos, CA, USA (May 2017).
[16] Clark, J. and Essex, A., "Commitcoin: Carbon dating commitments with bitcoin," in [Financial Cryptography and Data Security], Keromytis, A. D., ed., 390–398, Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
[17] Szalachowski, P., "(short paper) towards more reliable bitcoin timestamps," in [2018 Crypto Valley Conference on Blockchain Technology (CVCBT)], 101–104 (2018).
[18] Estevam, G., Palma, L. M., Silva, L. R., Martina, J. E., and Vigil, M., "Accurate and decentralized timestamping using smart contracts on the ethereum blockchain," Information Processing & Management 58(3), 102471 (2021).
[19] Stavrou, A. and Voas, J., "Verified time," Computer 50(3), 78–82 (2017).
[20] Gipp, B., Meuschke, N., and Gernandt, A., "Decentralized trusted timestamping using the crypto currency bitcoin," (2015).
[21] Gipp, B., Breitinger, C., Meuschke, N., and Beel, J., "Cryptsubmit: Introducing securely timestamped manuscript submission and peer review feedback using the blockchain," in [2017 ACM/IEEE Joint Conference on Digital Li- braries (JCDL)], 1–4 (2017).
[22] Tierion and Inc., "Tierion - blockchain proof engine," (2020). <https://tierion.com/> (22 July 2022).
[23] Factom, "Factom - a blockchain innovations company," (2019). <https://www.factom.com/> (22 July 2022).
[24] Ali, M., Nelson, J., Shea, R., and Freedman, M. J., "Blockstack: A global naming and storage system secured by blockchains," in [2016 USENIX Annual Technical Conference (USENIX ATC 16)], 181–194, USENIX Association, Denver, CO (June 2016).
[25] Zhang, Y., Xu, C., Li, H., Yang, H., and Shen, X., "Chronos: Secure and accurate time-stamping scheme for digital files via blockchain," in [ICC 2019 - 2019 IEEE International Conference on Communications (ICC)], 1–6 (2019).
[26] Zhang, Y., Xu, C., Cheng, N., Li, H., Yang, H., and Shen, X., "Chronos+: An accurate blockchain-based time-stamping scheme for cloud storage," IEEE Transactions on Services Computing 13(2), 216–229 (2020).
[27] Li, J., Wu, J., Jiang, G., and Srikanthan, T., "Blockchain-based public auditing for big data in cloud storage," Information Processing & Management 57(6), 102382 (2020).
[28] Yakovenko, A., "Solana: A new architecture for a high performance blockchain v0. 8.13," tech. rep. (2018).
[29] Solana, "Accounts - solana docs." Solana Foundation (2022). <https://docs.solana.com/developing/programming-model/accounts> (27 July 2022).
[30] Solana, "Runtime - solana docs." Solana Foundation (2022). <https://docs.solana.com/developing/programming-model/runtime> (27 July 2022).