OriginStamp: A blockchain-backed system for decentralized trusted timestamping

Article *in* it - Information Technology · November 2018

DOI: 10.1515/itit-2018-0020

4 authors, including:

Thomas Hepp
Universität Konstanz
11 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Alexander Schoenhals
University of Konstanz
7 PUBLICATIONS **35** CITATIONS
SEE PROFILE

Bela Gipp
Bergische Universität Wuppertal
138 PUBLICATIONS **2,266** CITATIONS
SEE PROFILE

Thomas Hepp*, Alexander Schoenhals, Christopher Gondek, and Bela Gipp

OriginStamp: A blockchain-backed system for decentralized trusted timestamping

https://doi.org/10.1515/itit-2018-0020
Received August 14, 2018; accepted November 12, 2018

Abstract: Currently, timestamps are certified by central timestamping authorities, which have disadvantages of centralization. The concept of the decentralized trusted timestamping (DTT) was developed by Gipp et al. to address these drawbacks. The paper provides insights into the architecture and implementation of a decentralized timestamp service taking the integration of multiple blockchain types into account. Furthermore, the components are introduced and the versatile application scenarios are presented. A future direction of research is the evaluation of blockchain technology and their suitability for timestamping.

Keywords: Decentralized Trusted Timestamping, Blockchain Technology, Blockchain Systems, Distributed Ledger Technology

ACM CCS: Security and privacy → Security services → Privacy-preserving protocols

1 Introduction

The proof of existence is more important than ever before. This is about the proof that content exists at a certain point in time. On the one hand, more and more processes are being digitized, accelerating our businesses and life. On the other hand, the progressive digitization increases the possibility of manipulation. The race between encryption and decryption technologies is gaining momentum [17]. More and more complex encryption methods are being developed, which suspects try to circumvent. To change the author of a decrypted document, just a few clicks are usually sufficient. In the event of copyright infringement, the existence of a document must be proven beyond doubt for the author to receive his or her attribution. Timestamping plays an important role in different business processes, e. g., copyright of intellectual property or in future patent litigation to prove who had an idea first. Images and videos are often not recognized in courts as evidence of the possibility of manipulation [9]. Therefore, a method is required to timestamp digital content. In the past, central services were used for this purpose. The users have to trust this central instance and are therefore directly dependent on their security to be protected against manipulation. The blockchain as part of the peer-to-peer electronic cash system by Nakamoto was introduced in 2008 [19] and is regarded as a tamper-proof, decentralized data structure. The characteristics of blockchain technology are perfectly suited to address this challenge. The following paper deals with the technical solution of this problem. How can the blockchain technology be integrated into an application to generate and verify timestamps of any digital content?

In Section 2 the current state of the art is examined, which methods are available for timestamping and what the strengths and weaknesses of these techniques are. The approach for timestamping on the Bitcoin blockchain is proposed in Section 3. Moreover, the components which are essential for the timestamping are explained in detail. Our technical solution is presented in Section 4. There we describe the individual modules and their interaction in more detail. At the end of this work in Section 5, we present the conclusions of our solution and give an outlook on future research.

2 Background

The idea of verifiable timestamps in the digital world is not new. Before the digital era, even post stamps were used as timestamps to prove the existence of documents. To protect a novel discovery, inventors in the 18th century sent letters to themselves and collected them in a safe.

Traditional methods, such as public notaries, are not applicable for the increasing digitization of the everyday life. The most important aspect is the rapidly increasing amount of digital content.

***Corresponding author: Thomas Hepp,** Universität Konstanz, Information Science Group, D-78464 Konstanz, Germany, e-mail: thomas.hepp@uni-konstanz.de, ORCID: http://orcid.org/0000-0002-7671-0602
Alexander Schoenhals, University of Konstanz, Information Science Group, D-78464 Konstanz, Germany, e-mail: alexander.schoenhals@uni-konstanz.de
Christopher Gondek, Bela Gipp, Universität Konstanz, Information Science Group, D-78464 Konstanz, Germany, e-mails: christopher.gondek@uni-konstanz.de, bela.gipp@uni-konstanz.de

2.1 Traditional timestamping

The authentication of a document is traditionally carried out by a central institution, e. g., public notaries. So it was obvious that the first electronic approaches followed this single party paradigm. As early as the 1980s, so-called e-notary concepts were presented and introduced [18]. They took advantage of cryptographic mechanisms and were thus able to provide secure mailing systems, two-way channels or even secure user files. Digital timestamps were managed in the 1990s by single party trust service providers (TSP) for the first time, e. g., Belsign. The implementations were mostly based on the timestamp scheme which Haber and Stornetta had published [12]. The principal approach was even then based on one-way hash functions, i. e., secure hash algorithms, which were used to determine fingerprints (checksum) of digital documents. To ensure the integrity of those signatures, the most important functional unit is the time stamping authority (TSA). TSAs ensure the existence of certain documents at a certain point in time. Even if multiple TSAs are used to increase reliability and reduce vulnerability — the trust service provider as a single party can't avoid the manipulation of information completely. Therefore, the use of these timestamps presupposes confidence in the issuing authority. However, there are some different timestamping techniques, most of them use checksums calculated by hash functions and store them in *central* servers one have to trust. In the past, efforts have also been made to research on decentralized solutions to reduce fraud.

Various papers of distributed timestamping systems have been published. Takura, Ono, and Naito describe a technique in which a digital signature and a secret key are created by a TSA. The issued encrypted timestamp is then stored partially on distributed servers [23]. By distributing the storage, manipulation is decreased [5]. Nonetheless, the required trust in the TSA as unique issuer is indispensable. A further approach to eliminate a fraudulent authority is the use of multiple issuers. Whereas the issuers are also requesters at the same time. It is mandatory that few of these have to be trustworthy regarding designated TSA specified in RFC 3161 [1]. Even Haber and Stornetta described this decentralized scheme. Anyway, this scheme has no widespread adoption achieved in practice so far.

2.2 Decentralized trusted timestamping

A novel approach which addresses these needs were introduced 2015 by Gipp, Meuschke, and Gernandt. They showed that cryptocurrencies, e. g., Bitcoin or Ethereum, can serve as a decentralized trusted timestamping (DTT) ledger [10]. The main idea is that a unique fingerprint of a document is inserted into the cryptocurrency underlying blockchain. The fingerprint is, in this case, an SHA-256 checksum based on the definition of RFC 6234 [14]. In general, blockchain technology allows a network of several computers (nodes) to agree at regular intervals on the true state of a distributed ledger, i. e., all nodes are connected in a decentralized way. Such ledgers can list any digital data, e. g., transaction records, credentials or even fragments of programming code. To briefly describe the functionality of the blockchain which is founding almost all cryptocurrencies, we will use an example of a single transaction. Transactions are broadcast to all nodes, and once a node receives a transaction message, it will work this transaction into a block. A block is a data structure storing all transactions received during its creation. Each block consists of four parts [2]: *Block size*, *Block header*, *Transaction counter* and *Transactions*. The block header contains an aggregate of the hashed transactions, the previous block's hash, and a Unix epoch timestamp. The work of a node is to validate a transaction. As soon as a miner calculates a proof, the collected transactions are integrated into the block, and the block is propagated to the entire network. The other nodes then check the block for correctness. The block that is validated first is accepted by the network and linked to the chain. The validation is based on a predefined consensus. In the case of Bitcoin blockchain, this is the "proof of work" consensus. The blockchain protocol grants incentives for successful calculation of proof tasks. The mechanism to enforce consensus is accepting the longest chain since it took the most resources to compute and was thus accepted by the majority of nodes.

As a result, the tamper-proof characteristics of a blockchain are predestined for tracking and versioning of digital data and its copyright — additionally, this also lead to a trustworthy network. Summarized, involving blockchain technology into DTT by Gipp et al. led to following benefits:

1. decentralized, cryptographic integrity validation of the timestamping process;
2. high incentives for computing nodes to contribute to the decentralized record-keeping process at the heart of the Bitcoin protocol;
3. minimal effort for users: no need to setup specialized hardware or software;
4. low cost of operation, which allows us to provide the service free of charge.

For a detailed description of the first model of a DTT, please refer to [10]. Several decentralized trusted timestamping

services are already provided by some platforms. They differ mostly in their implementation of building the transactions to broadcast. There is a web service called POEX.io[1] which is developed by Manuel Araoz. They build a custom transaction for each submitted document. A SHA-256 checksum of document as well as a marker is inserted into this transaction which is then broadcasted into Bitcoin blockchain. The marker is used to identify all of the transactions processed inside the blockchain. For this service, users must pay 0.005 BTC to cover the operator's expense of running the service and initiate the transaction. An additional service, OpenTimestamps[2] is fully available for download but also can used as an online web service. This service was initiated by Bitcoin core developer Peter Todd, with the goal to provide a general interface to execute timestamps on various blockchains — not only on Bitcoin blockchain. An essential difference to POEX.io service is its scalability. For example, suppose one want to submit 100 different files: at POEX.io this would require 100 Bitcoin transactions — inefficient and expensive. In summary, most of the time, only small startups deal with this topic. Large IT providers such as Dropbox, Box, Google, SalesForce and Microsoft are not yet represented with marketable products, but it is foreseeable, that they will adapt this service because the advantages are evident. Recently, a luminous signal was ignited by Apple. They filed a patent application, where they present a multi-check architecture for trusted timestamping, which is also based on the blockchain technology [15]. The multi-check combines traditional public key infrastructure (PKI) on the one hand and decentralized block's validation on the other. In addition to the many benefits of the decentralized timestamping approach, more and more critical voices are being raised to expose the problems and risks of the blockchain by using it as a timestamping issuing service. Mainly it is criticized that the blockchain is wasted by the enormous amount of transactions, so that the network does not comply with the validation process. The individual choice of transaction value and transaction fee arises this problem as well. It is up to a miner to include a pending transaction into the block he is currently working on. Because block size is mostly limited, transaction with low fee might be pending for a long time until one miner decides to include it into his block. This influences the timestamps accuracy negatively. However, even if the criticism is justified, these characteristics can be addressed by adjustments of the certain blockchain and depend on the implementation of the respective blockchain. Additionally, transaction fees could be dynamically chosen by the respective timestamp service provider, depending on cryptocurrency's price and requested timestamp amount.

Basically, trusted timestamps are inevitable in today's world to protect authors and their intellectual property which is mostly shared in digital manner. After reviewing the weaknesses and strengths between traditional trusted timestamping and blockchain based timestamping we see the main point in the undeniable better security at the DTT. Traditional schemes are not applicable due to the massive amount of digital content and also allow gaps in the manipulation of information, because they usually require a trustworthy authority. The problem that the speculation price of cryptocurrencies affects transaction fees is a major problem, as it causes a transaction delay which decrease the solidity of timestamp service. Therefore, a solution must be developed to determine the most matching fees to keep the running costs as low as possible. A timestamping service which is based on the idea of integrating digital fingerprints into the cryptocurrency blockchain has to consider and overcome these challenges.

3 Originstamp trusted timestamping

In this section we introduce our concept for generation of timestamps using the Bitcoin blockchain. We give a detailed explanation on how to write the hash of any digital content efficiently into the blockchain without spamming the network. Furthermore, we will demonstrate how the verification of the timestamp of a document, independently on our system.

3.1 Timestamp creation

To create a tamper-proof timestamp of digital content such as a file, the cryptographic hash of that file is calculated and submitted to the Bitcoin blockchain in multiple steps. These steps are documented in detail in Figure 1:

First, the SHA-256-hash [14] of the file that is submitted for timestamping is calculated. OriginStamp does only submit this cryptographic hash of a file for mainly two reasons: As a file that is submitted to OriginStamp can possibly contain confidential information, it is mandatory that the confidentiality of the submitted file is not compromised. The SHA-256-hash function is beneficial regard-

1 https://poex.io

2 https://opentimestamps.org/

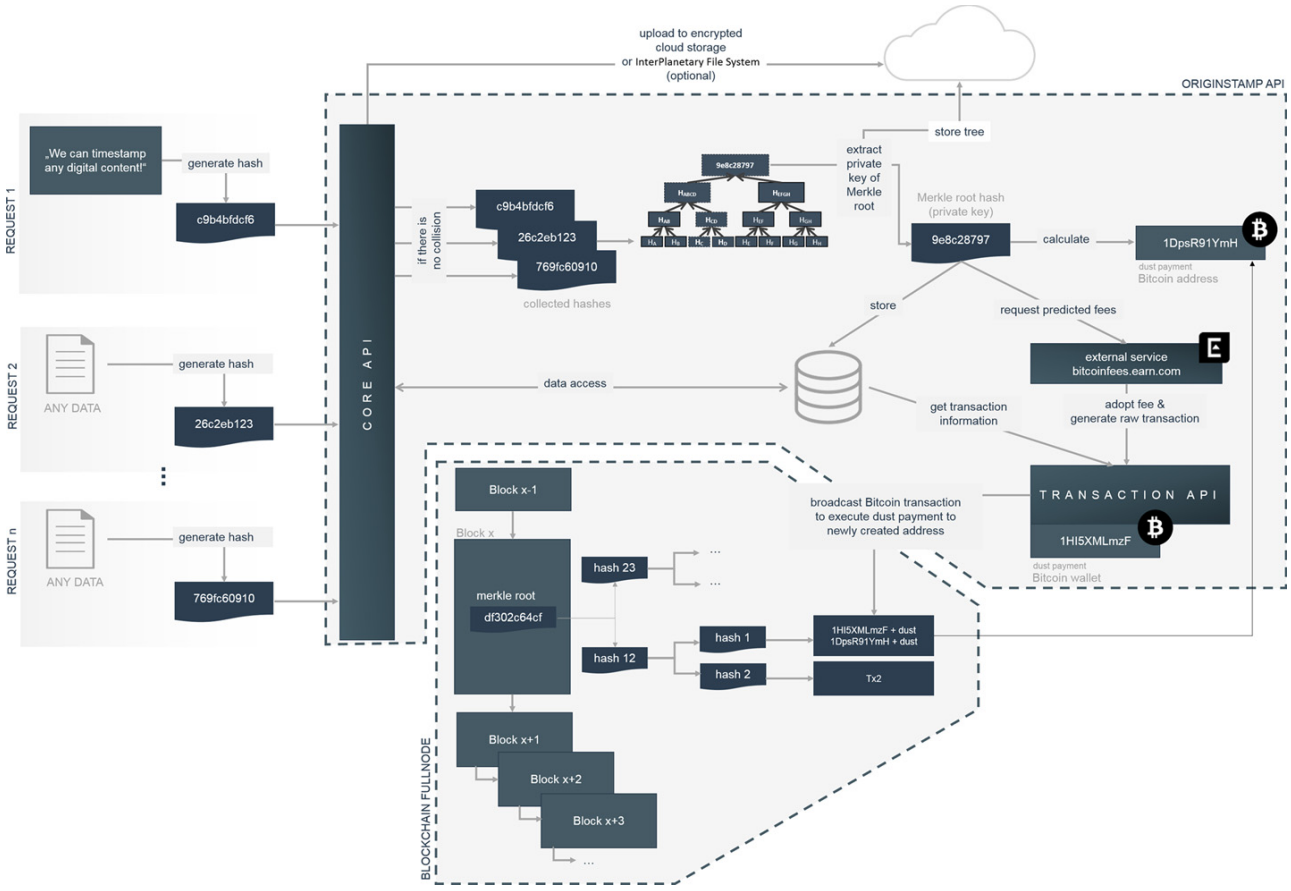

Figure 1: Decentralized Trusted Timestamping (DTT) activity flow as implemented in *originstamp.org*.

ing this requirement, since it is a mathematical one-way function [20]. From a given hash, however, it is due to high computational complexity infeasible to determine a matching input file. The other beneficial property of the SHA-256-hash function which is utilized is that any input file is mapped to 32 bytes. As a result, the fingerprint of a file does not grow with the size of that file. This allows to generate efficient timestamps even for very large files. On the other hand, it implies that, since the number of possible input files is larger than the number of possible output hashes, two files will map onto the same hash. A hash collision, however, is extremely unlikely and can safely be ignored for the practical purposes of OriginStamp. After the hash of a file as its fingerprint has been created, it must be embedded in the Bitcoin blockchain. And although each input file results in a hash of only 32 bytes, each hash that should be embedded in the blockchain would naively require a single transaction. Thus, the total hash submission cost would rise linearly with the number of timestamps as explained below. Furthermore, it would artificially increase the load on the Bitcoin network and potentially block other transactions. To avoid most of this load and to reduce the total cost to be linear with time, the OriginStamp system combines all hashes that are submitted within each 24-hour period to form a single submission: Triggered by a scheduler, all hashes that were submitted in such a period and additionally are not yet timestamped are collected and sorted in lexicographical ascending order. These hashes are then concatenated in this order to form a single character string, where white spaces are inserted as delimiters between the hashes. These hashes are used to build a balanced Merkle tree. This has the advantage over seed files that only a small section is required for verification. The root of the tree (formerly seed hash) is embedded in the Bitcoin blockchain using a transaction. Importantly, any changes to the proof also alters the root hash of the merkle tree. Therefore, a multi-seed must not be changed, once it is successfully timestamped. Due to the greatly reduced costs, the multi-seed submission is currently the preferred method for most use cases: Currently, the OriginStamp system holds 1868679 hashes, which were submitted in 1246 transactions. One disadvantage of using multi-

seeds is that files which have been submitted at the beginning of a 24-hour period will be timestamped only with a considerable delay. Thus, also the time which is associated with the timestamp will be significantly delayed from the actual submission time to the OriginStamp system. To serve also use cases that require an immediate timestamp with a time that is as close as possible to the time at which the file was submitted to OriginStamp, the system also allows to directly submit the hash of a file to the blockchain if it is explicitly requested by a user. Such a hash is then denoted as a *single seed*. To perform the actual submission, the hash of the multi-seed or the single is directly and without any further modifications used as a private key for the blockchain. This private key can then be cryptographically used to determine an uncompressed Bitcoin address to which a payment with the lowest amount is sent (*dust payment*). Any slight alteration of the seed thus yields a completely different address. Due to the very large number of possible private keys and addresses that are valid on the blockchain, it is highly unlikely that this address has already received a payment. Hence, the time of the first payment that is sent to this address from the OriginStamp system is a valid proof of the existence of the corresponding private key at that time.

3.2 Verification of a timestamp

One advantage of Merkle trees over seed files is that not all hashes are required for the verification. Since the Merkle tree proof does not have to include all hashes, but only the meshes to the root, this approach scales unlike linear seed files. With 1 million hashes, the size of the proof is smaller than 10 KB, which saves costs for OriginStamp and the users. Figure 2 demonstrates how this data structure looks like and how the mashes are determined.

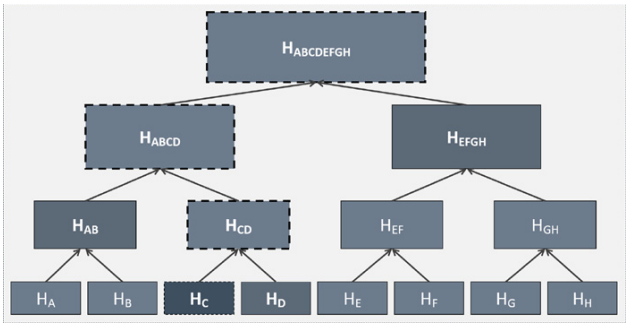

Figure 2: Example of a Merkle tree representing verification of Hash C (H_C).

The verification of H_C begins with the following step:

H_C = 2cf24dba5fb0a30e26e26e83b2ac5b9e29e1b1
61e5c1fa7425e73043362938b9824

H_D = 486ea46224d1bb4fb680f34f7c9ad96a8f24ec
88be73ea8e5a6c65260e9cb8a7

H_{CD} = SHA256(H_C+H_D) = 15e178b71fae8849ee562c
9cc0d7ea322fba6cd495411329d47234479167cc8b

These steps must be repeated until you reach the root of the tree. The root contains the hash we have stored in the blockchain (= private key). If the manually calculated values match the Merkle Tree, this step is completed and verification can continue. From the private key, we use *ECDSA* by Johnson, Menezes, and Vanstone [16] to compute the key pair based on the *secp256k1* curve [21]. The public key must be uncompressed. To calculate a Bitcoin address, the SHA-256 of the public key is calculated, then *RIPEMD-160* [7] is applied. The Bitcoin address is then determined by adding 0x00 in front, and 4 byte checksum behind and finally convert the result into *Base58 encoding*. By having a look at the transactions for this address, the block-time of the earliest represents the tamper-proof timestamp.

If the OriginStamp service does not exist in the future for any reason, the timestamps can still be retraced. Through this transparent workflow, combined with the features of the Blockchain, OriginStamp provides a sustainable and reliable theoretical background for trusted timestamping of digital content.

4 System concept

Blockchain-based applications require a special architecture since it is not efficient to read data directly from the blockchain on request. It is further possible that the Bitcoin protocol changes (see hard fork in August 2017) and the application needs to be adapted to the new consensus. Apart from that, no push notifications are sent from the full node when any transaction is mined into a block. According to Antonopoulos, a full node is a client in the Bitcoin network, storing the entire history of transactions. “A full node handles all aspects of the protocol and can independently validate the entire blockchain and any transaction.” [2] In addition to the core application, which is used to manage the hashes, other micro-services can be accessed via RESTful interfaces [8] as illustrated in Figure 3. The idea is to have individual components that can be easily replaced in case of a Bitcoin protocol change without having to migrate complex updates. Another advantage is the easy extensibility of the application, which is considered as future work in Section 5. The input is the trans-

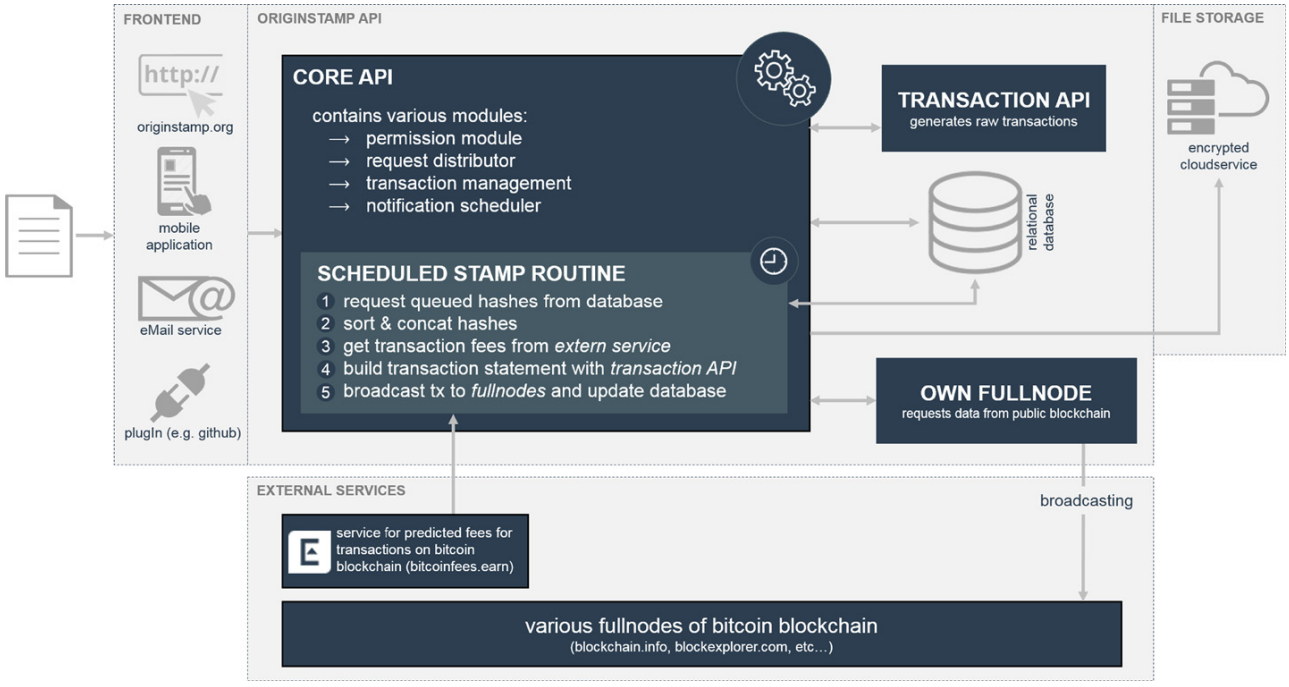

Figure 3: Overview of software components for Decentralized Trusted Timestamping.

mission of a hash from various sources like our website, mobile application, mails or plugins as illustrated in Figure 1. Besides hashes, files can be optionally archived on OriginStamp. All the data is finally transferred through a RESTful interface, which is free of charge and allows the development of applications with Trusted Timestamping. The core of the OriginStamp service is presented in Section 4.1 and addresses the central management of hashes, user management and the connection to external services. The interface for the calculation of efficient transaction fees is presented in Section 4.2, followed by our implementation of the Bitcoin protocol in Section 4.3. Moreover, OriginStamp offers an optional file archiving option, which is proposed in Section 4.4. Finally, Section 4.5 provides insights on how the service can be integrated into existing business processes and projects.

4.1 Core

The core of the OriginStamp api handles all requests and business-related logic. Before processing the submitted data, the API key is checked according to permissions. Some functions like the verification of a mail subscription require permissions to prevent abuse. Additionally, basic tests are performed: The format of a hash is checked by regular expressions. Mail addresses are validated for the notifications. A user can enter his or her e-mail address and will be notified as soon as the timestamp has been successfully included in the blockchain. If all preconditions are met, the hash is further processed by the system. First, the full node is used to check whether there is already a transaction for the transmitted hash. The associated file is uploaded to the encrypted cloud service if desired by the requester. All associated meta-data such as user account, e-mail notification are stored in the SQL database for later verification processes. To achieve higher performance, hashes with the corresponding transactions are synchronized between full node and database. As illustrated in Figure 3, the core also controls and triggers the schedulers. These schedulers are necessary for synchronization of the transaction status with our database. As a part of this scheduled timestamp submission, the optimal transaction fee must be determined before we broadcast the corresponding transaction.

4.2 Fee determination

A timestamp is created by embedding the SHA-256 fingerprint into the blockchain, which is achieved by broadcasting a transaction that is going to be mined into a block. Usually, a transaction is verified and mined every 10 minutes by the Bitcoin network. This time depends on various factors like the offered transaction fee, which is set by the sender of the transaction, and the mining complexity, which is set by the network. One major drawback

of the timestamping service by Gipp, Meuschke, and Gernandt [10] was the usage of a fixed fee. The speculations on cryptocurrencies, especially Bitcoin, result in a volatile exchange rate and finally in dynamic transaction fees. If there are many transactions in the memory pool, transaction fees increase due to the limited block size. A factor for miners to choose the transaction to be mined is the transaction fee. The higher, the more profitable for miners. This is why senders are competing with each other, and the fees are increasing. Having a static fee, either the transaction fee is too high, which increases the running costs or the fees are too low and transactions might stick in the mempool of the network for hours or days. Earn[3] makes a prediction that is based on the number of unconfirmed transactions that were included in the Bitcoin blockchain within the last 3 hours. According to this service, a prediction of a likely future mempool and miner behavior is predicted using Monte Carlo simulation [4].

Having analyzed 187 transactions (from 06/01/2016 until 12/05/2017), the average time, from broadcast until mined into a block, was around 55 min and 32 s. By analyzing the data, we identified three outliers, caused either by insufficient Bitcoins on our address or by bugs with microservices, which are not related to the fee calculation. If ignoring the outliers, the average time is 34 min and 13 s, which is accurate considering the target of 60 min with 90 % likelihood.

4.3 Transaction generation

Having determined a reasonable fee, the raw transaction is generated. The transaction API in Figure 3 introduces methods for the generation of raw Bitcoin transactions. Instead of using existing wallets, we propose a general adapter layer, which defines functions and methods required for timestamping:

- Generation of a transaction
- Broadcasting a transaction
- Reading the status of a transaction
- Check the balance of the wallet
- Find the transaction for a hash

These functionalities are implemented in the microservice and a configuration parameter can be used to specify which currencies are used for timestamping. This makes it easy to integrate additional blockchains into the system or adapt existing ones without having to change the business logic of the core application. Since there is de-facto no standard for blockchains, the choice can be very flexible and other services can be connected depending on the application and needs. Our implementation of the Bitcoin protocol to be independent of third-party services and to use additional features such as *OP_RETURN* and collection of dust payments. The generation of transactions is triggered via the corresponding scheduler, for which private key, seed hash and the fee per byte are required. The Bitcoin blockchain provides two possibilities for embedding hashes: Firstly, a hash is converted to a Bitcoin address and a *dust payment*, lowest possible payment, is transferred. Secondly, a hash is directly stored in an *OP_RETURN* transaction. On the one hand, coins are not wasted and the blockchain is not spammed, on the other hand, the source address is necessary to find the timestamp. Besides, the transaction size of *OP_RETURN* is longer (256 Bytes vs. 226 Bytes), which means a higher fee in total. One major drawback of the first option is the generation of unspent outputs (UTXO), which are stored in memory and increase the difficulty of the mining process and these outputs remain unspent forever. Therefore, we provide a service for the collection of the UTXOs and keep the blockchain clean. Our micro-service enables both types of transactions and can be configured, based on an organizations preference.

4.4 Archive storage

A blockchain can be used to store documents. However, it is not advisable to store large amounts of data, because of high costs and restrictions from the network. Assuming a file size of 1MB, the file has to be split into several transactions, due to network restrictions like maximum block size and block time controlled by the mining complexity. Without these restrictions, only a few participants could afford to become part of the network, due to the fast increasing size of the blockchain. The more participants, the more secure the network is. Furthermore, this approach is not applicable due to the enormous transaction costs. The taxonomy of Xu et al. [24] gives a classification for the design of blockchain-based systems. The onchain storage of data is neither cost-effective nor of high performance and nor flexible. Their recommendation is to use off-chain storage, as it is generally cost-efficient and more flexible. They propose two basic ways for dealing with files: On a private or third-party storage service (e. g. Amazon Cloud or file system) or in a peer-to-peer network like IPFS[3]. We decided to use an encrypted cloud service for archiving files privacy-preserving because it is easier to maintain and thus more reliable for users.

3 https://bitcoinfees.earn.com/

4.5 Applications and plugins

OriginStamp is a web-based, trusted timestamping service that uses the decentralized Bitcoin blockchain to store anonymous, tamper-proof timestamps for any digital content. We host a ready-to-use instance of the service at https://originstamp.org. In addition to the web service, there is also a mobile application for OriginStamp[4]. This app can be used to timestamp different types of media, such as ideas in the form of voice recordings or evidence of accidents. Trusted Timestamping on the blockchain is also applied for a variety of use-cases [9, 11] Building up on OriginStamp, there are already further applications that have been described more specifically in the past. One example is the protection of intellectual property already during the development of particular innovations [22]. Even in the early phases of problem-solving processes. For this purpose, during the entire innovation cycle all inventors' contributions are digitally recorded. The digital outcome during a, e. g., ideation session, is timestamped with the help of OriginStamp service and should in the future enable the early inclusion of third parties or even competing companies for the protection of innovations long before patent protection – according to the Open Innovation paradigm. Another concept that usefully integrates the OriginStamp service deals with tracking individual goods within a supply chain. Hepp et al. [13] provides insights into how objects can be protected and monitored by a varnish with a unique crack pattern, as an example of a Physical Unclonable Function. The perceptual hash of the unique pattern is used to encrypt the associated data to ensure privacy. Instead of logging each event on the blockchain individually, which is not possible due to the limited transaction throughput, OriginStamp is used to preserve data integrity on the blockchain. In the future, the craquelure-based tracking approach could be extended to supply chain integration to secure the origin of products, including prevention of counterfeiting, securing the place of manufacture for trademark law or state surveillance of the agricultural economy.

5 Conclusion and future work

In this paper, we presented a new approach to timestamping and archiving of digital content using blockchain technology. The OriginStamp service has been developed from scratch to be as independent of third party services as possible.

As described by Croman et al., a large gap exists in the number of transactions processed per day between Bitcoin and VISA. The authors conclude that the Bitcoin protocol must be fundamentally redesigned to solve the scaling problem [6]. By changing the protocol, the dependent components must also be adjusted. To alleviate the impact of such changes, we implemented the core independently of the Bitcoin protocol.

The market capitalization of Bitcoin blockchain is currently the most promising, which does not mean that this technology will prevail. Nevertheless, our flexible architecture allows to conveniently utilize other promising blockchains in the future as well. OriginStamp can be used free of charge for time-stamping of digital content. Currently only timestamps are created within 24 hours.

Future research will focus on what an ideal blockchain to the timestamp would look like.

4 https://goo.gl/nQkx5A

References

1. C. Adams et al. RFC 3161: Internet X. 509 public key infrastructure timestamp protocol (TSP). In: (2001).
2. A. M. Antonopoulos. *Mastering Bitcoin – Programming the Open Blockchain*. OReilly Media (2017).
3. J. Benet and J. Ai. IPFS – Content Addressed, Versioned, P2P File System (DRAFT 3). In: (2014).
4. K. Binder et al. Monte Carlo Simulation in Statistical Physics. In: (1993), p. 156. DOI: 10.1063/1.4823159.
5. A. Bonnecaze et al. Secure time-stamping schemes: a distributed point of view. In: (2006), pp. 662–681.
6. K. Croman et al. On scaling decentralized blockchains (A position paper). In: (2016), pp. 106–125. DOI: 10.1007/978-3-662-53357-4_8.
7. H. Dobbertin, A. Bosselaers, and B. Preneel. RIPEMD-160: A strengthened version of RIPEMD. In: (1996), pp. 71–82. DOI: 10.1007/3-540-60865-6_44.
8. R. T. Fielding. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis. University of California, Irvine, 2000.
9. B. Gipp, K. Jagrut, and C. Breitinger. Securing Video Integrity Using Decentralized Trusted Times-tamping on the Blockchain. In: (2016), pp. 1–10. DOI: 10.1007/s11257-016-9174-x.
10. B. Gipp, N. Meuschke, and A. Gernandt. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. In: (2015), pp. 1–6.
11. B. Gipp et al. CryptSubmit: Introducing Securely Timestamped Manuscript Submission and Peer Review Feedback Using the Blockchain. In: (2017), pp. 1–4. DOI: 10.1109/JCDL.2017.7991588.
12. S. Haber and W. S. Stornetta. How to Time-Stamp a Digital Document. In: (1991), pp. 437–455.

13. T. Hepp et al. Securing Physical Assets on the Blockchain. In: (2018).
14. IETF. *RFC 6234 – US Secure Hash Algorithms b (SHA and SHA-based HMAC and HKDF)*. 2011.
15. A. Inc. *Obtaining and Using Time Information on a Secure Element*. 2017.
16. D. Johnson, A. Menezes, and S. Vanstone. The Elliptic Curve Digital Signature Algorithm (ECDSA). In: (2001), pp. 36–63. DOI: 10.1007/s102070100002.
17. G. S. Lunney et al. The death of copyright: Digital technology, private copying, and the digital millennium copyright act. In: (2001), pp. 813–920.
18. Christian Mueller-Schloer and Neal R. Wagner. *The implementation of a cryptography-based secure office system*. 1982.
19. S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. In: (2008), p. 9. DOI: 10.1007/s10838-008-9062-0.
20. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In: (1989), pp. 33–43. DOI: 10.1145/73007.73011.
21. C. Research. Standards for efficient cryprography – SEC 2: Recommended Elliptic Curve Domain Parameters. In: (2000).
22. A. Schoenhals, T. Hepp, and B. Gipp. Design Thinking using the Blockchain: Enable Traceability of Intellectual Property in Problem-Solving Processes for Open Innovation. In: ACM. 2018, pp. 105–110.
23. A. Takura, S. Ono, and S. Naito. A secure and trusted time stamping authority. In: 1999, pp. 88–93.
24. X. Xu et al. A Taxonomy of Blockchain-Based Systems for Architecture Design. In: (2017), pp. 243–252. DOI: 10.1109/ICSA.2017.33.

Bionotes

Thomas Hepp
Universität Konstanz, Information Science Group, D-78464 Konstanz, Germany
thomas.hepp@uni-konstanz.de

Thomas Hepp has been a PhD student in the Information Science group at the University of Constance since 2016. His research focuses on blockchain technology and how it can be used to increase transparency and reproducibility in supply chains. In addition to theoretical knowledge, Thomas is passionate about transferring these research results into an innovative product, which is why he is co-founder and CTO of OriginStamp.

Alexander Schoenhals
University of Konstanz, Information Science Group, D-78464 Konstanz, Germany
alexander.schoenhals@uni-konstanz.de

Alexander Schoenhals is a PhD candidate at Daimler AG, supervised at the University of Konstanz. In the past, he has implemented several interactive systems in VR/AR with the main emphasis on haptic feedback. His current research focuses on interactive methods to recognize, track and protect intellectual property in the very first stage of the innovation cycle with novel technologies. This plan requires an interdisciplinary exchange, therefore he maintains a lively exchange with business representatives, legal experts and also representatives of his area of expertise - computer science.

Christopher Gondek
Universität Konstanz, Information Science Group, D-78464 Konstanz, Germany
christopher.gondek@uni-konstanz.de

Christopher Gondek is a computer science master student at the Computer and Information Science Department of the University of Konstanz, Germany. Currently, he is also serving as a backend developer for OriginStamp. His work is mainly focusing on blockchain technology and data science. Christopher is also passionate about competitive programming, where algorithms and data structures are put to use.

Prof. Dr. Bela Gipp
Universität Konstanz, Information Science Group, D-78464 Konstanz, Germany
bela.gipp@uni-konstanz.de

Prof. Dr. Bela Gipp leads the Information Science Group at the University of Konstanz, Germany. His research lies at the intersection of information science and data science, where he focuses on the retrieval, analysis, and visualization of large volumes of data. The implications of blockchain technology – for the benefit of both industry and society – is another research domain Bela is passionate about. Currently, he serves as a juror and the university partner for the worlds largest Blockchain Competition.