Intellectual Property Protection using Decentralized Trusted Timestamping Based on the Blockchain

Yuefei Gao[1] and Hajime Nobuhara[2]
[1, 2] Department of Intelligent Interaction Technologies, University of Tsukuba
1-1-1 tennodai, Tsukuba, Ibaraki 305-8573, Japan
E-mail: [1] kou@cmu.iit.tsukuba.ac.jp, [2] nobuhara@iit.tsukuba.ac.jp

Abstract: Present decentralized trusted timestamping based on the blockchain only provide 40-byte storage for digital files. This does not permit sufficient storage to encode other information. The proposed method expands the storage space to a maximum of N*40 bytes, thereby enabling the storage of additional information (e.g., file name, creator name, and keywords). Experimental results indicate that the proposed method can timestamp a file in an average of 20 min at a possible cost of 0.24 USD. We consider that the proposed method can prove the existence and integrity of a digital file.

Keywords—**Blockchain, Timestamp, Decentralize, Intellectual property**

1. Introduction

Considerable intellectual property is created and shared everyday on the Internet. For example, approximately 300 hours of videos are uploaded on Youtube every minute [8], and on average, 1.83 million photos are uploaded publicly on Flickr every day [5]. Such digital intellectual property can be tampered with or forged relatively easily. One solution to this problem is a technique called "trusted timestamping", issued by a central Time-Stamping Authority (TSA). This process can track the creation and modification time of digital data to ensure the existence and integrity of the data. Trusted timestamping is issued by a central Time-Stamping Authority (TSA). Users send a digital file to the TSA, where it is signed with the current time digitally [3]. However, this process has security problems. For example, if the TSA's timestamp server is hacked, the timestamp will be unreliable. Recently, decentralized trusted timestamping has been implemented to address this problem [4]. Currently, there are several decentralized timestamping services based on Bitcoin's peer-to-peer digital currency infrastructure, called "blockchain".

Essentially, the blockchain is a distributed database that does not rely on central servers, i.e., it is decentralized. The blockchain stores all confirmed Bitcoin transactions [1]. The existence and integrity of transactions are protected, i.e., the transactions cannot be tampered with or forged. Based on such features, web services, such as BTProof [2] and Proof of Existence (PoE) [7], have implemented decentralized timestamping. However, existing web-based trusted timestamp services only embed a maximum of 40-byte hash in a transaction in the blockchain. This study proposes a methodology for storing a maximum of N*40 bytes of data. A comparison of the TSA, the current trusted timestamping process, and the proposed method is shown in Figure 1.

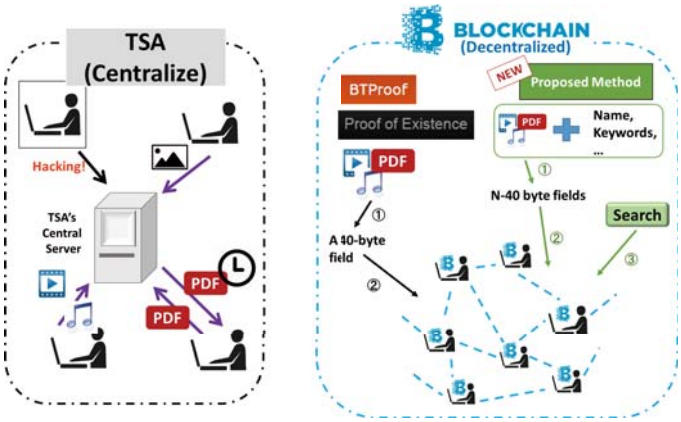

Figure 1. Concept of trusted timestamping based on TSA and blockchain

Existing web-based trusted timestamp services embed a maximum of 40-byte hash in the blockchain. The hash data cannot be reversed, which makes it difficult to determine the creators and other related information. To store such information, we propose a methodology for storing a maximum of N*40 bytes in the blockchain. For creators who prefer not to timestamp their data anonymously, the proposed method can embed related information (e.g., file name, creators' name, and comments) and the hash of the digital data.

We conducted three experiments to evaluate the proposed method. We set N = 3 and conducted experiments to evaluate the proposed method. If the transaction created by proposed method is stored in the blockchain successfully, it means that the proposed method is effective. In addition, we evaluated the cost and computation time (broadcast time) of the proposed method. The experimental results show that the proposed method can prove the existence of intellectual property at a particular time and that the intellectual property has not been modified since that time (integrity). We also find that the proposed method implements decentralized trusted timestamping in an average time of 20 min at a possible cost of 0.24 USD.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of Bitcoin and the blockchain, and describe current decentralized trusted timestamping services. In Section 3 the proposed method is described in detail. Experimental results are presented and discussed in Section 4, and conclusions are presented in Section 5.

2. Related work

2. 1 Overview of Bitcoin and the Blockchain

Bitcoin is a peer-to-peer crypto digital currency system proposed in 2009 by an unknown person or entity using the pseudonym Satoshi Nakamoto [6]. Bitcoin is decentralized and does not rely on governments or other legal entities. This means that users can make transactions directly and securely anywhere via the Internet.

The blockchain is the public ledger of Bitcoin and it has three important features:

1) It is equivalent to a large record book that contains a massive number of entries. All confirmed Bitcoin transactions are recorded in the blockchain. Although nearly all entries describe Bitcoin transactions, it is possible to record other information in the blockchain.

2) No individual or entity controls the blockchain. It is shared among all Bitcoin users; thus, whenever something is written on the blockchain, agreement among many records is required.

3) It is extremely difficult to tamper with or forge records stored in the blockchain. Multiple copies of the blockchain are owned by numerous people around the world; therefore, forging or tampering requires modification of all blockchain records in various locations, which is exceedingly difficult.

The existence and integrity of data in the blockchain are guaranteed by the above features. Existence means that the data have actually existed since a certain time. Integrity means that the data have not been altered after a certain time. Transactions are stored in chronological order in the blockchain.

2. 2 Current trusted timestamping services

There are several web applications implemented using different methods, e.g., BTProof and PoE, for decentralized timestamps based on the blockchain. They help prove that a digital file existed at a point in time (existence) and that it has not being altered since the given time (integrity). Note that BTProof and PoE require a digital file as input

In BTProof's trusted timestamping, the input file is first hashed and converted into a Bitcoin address at first. An address is a string of characters and numbers. Then, by making a small payment to the address, the transaction and thus the hash are stored in the blockchain. Transactions in the blockchain are extremely difficult to tamper with or forge; therefore, the digital file is stored securely [2].

PoE implements Bitcoin's script language to embed digital files in the blockchain. Bitcoin transactions are validated by executing a script written in a Forth-like scripting language. Currently, most Bitcoin transactions have the form "A pays B". Such transactions are based on scripts referred to as Pay-to-Public-Key-Hash (P2PKH) scripts. However, Bitcoin transactions are not limited to the "A pays B" form. There are five standard types of transaction scripts: P2PKH, public-key, multi-signature (limited to 15 keys), pay-to-script-hash (P2SH), and data output (OP_RETURN). PoE applies two types of these, i.e., P2PKH and OP_RETURN [1]. Figure 2 shows these two types of script.

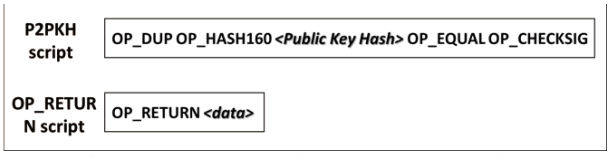

Figure 2. P2PKH and OP_RETURN scripts

In PoE, a document is also hashed; however, it is hashed to a 32-byte string rather than a Bitcoin address. Then the 32-byte string is embedded in the scriptPubKey field of the transaction. Thus, a special transaction is constructed. After broadcasting this special transaction, the 32-byte hash is stored in the blockchain with the special transaction, thereby making it difficult to tamper with or forged PoE also use "DOCPROOF" as a marker for their transactions by placing it at the beginning of the 32-byte string, which makes it easier to search for transactions [7]. The format of PoE's script can be represented as: *OP_RETURN <DOCPROOF + 32 bytes hash>*.

2. 3 Problems of current trusted timestamping services and the proposed solutions

BTProof and PoE provide decentralized trusted timestamping based on the blockchain. However, the hashed digital content cannot be reversed, which means that it is impossible to determine if any special data have been stored in the transaction. Current method is limited to 40 bytes of data in a transaction which makes it impossible to include more related information. To address these problems, we propose a method to store a maximum of N*40 bytes of data in transactions with N data fields.

3. Proposed method

We propose a method for storing a maximum of N*40 bytes in the blockchain. We choose the BTProof method to embed digital data in a Bitcoin transaction. We did not choose the PoE method, which embeds digital data using the OP_RETURN script, because multiple outputs with OP_RETURN scripts tend to be rejected by the blockchain. Transactions with multiple OP_RETURN outputs are recognized as "strange transactions", which have a high probability of not being recorded in the blockchainExcept storing the hash of the digital file, we also store some related information such as title, creator name and keywords. Instead of using OP_RETURN script, we convert the inputs to N Bitcoin addresses. Figure 3 shows the steps of proposed method.

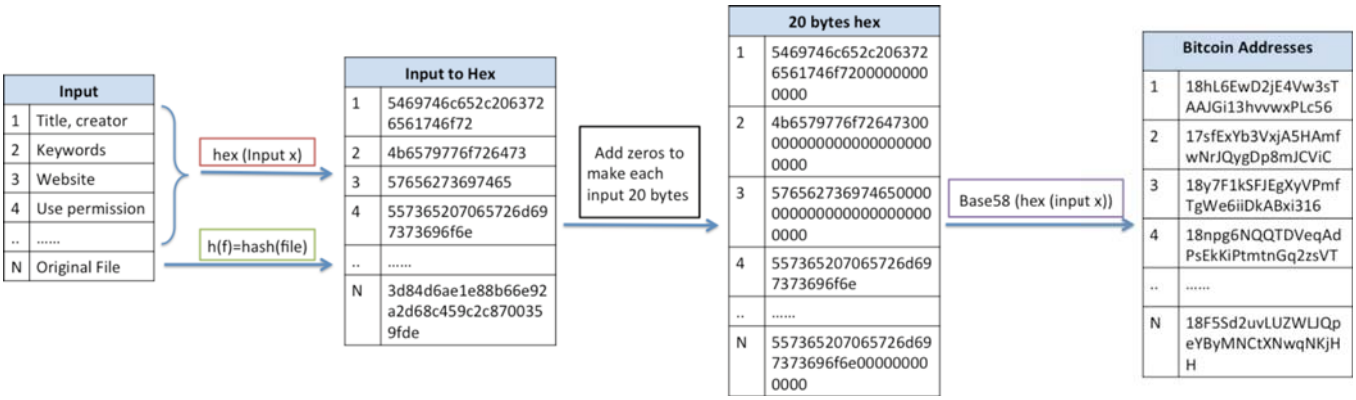

Figure 3 Processes of proposed method

Step 1: User inputs N field. From the 1st to the (N-1)th input are related information of a digital file, the Nth input requires the digital file.

Step 2: The 1st to (N-1)th inputs are converted to hexadecimal strings.

Step 3: The Nth input is hashed locally (which means the original digital file will not be uploaded).

Step 4: Add zeros after the hexadecimal strings to make the strings all 20 bytes.

Step 5: Turn these hexadecimal strings to N Bitcoin addresses by using Base58 encoding scheme.

Step 6: Create a transaction to make small payments to these N addresses.

An example of the application of proposed method is shown in Figure 4. User A writes an essay and wants to get copyright protection. User A inputs the related information of the essay, such as the title, writer's name, keywords and then choose the original file of the essay. Then the proposed method creates a special transaction with the related information of User A's essay included. This special transaction is broadcasted to the network and recorded in all of the blockchains around the world.

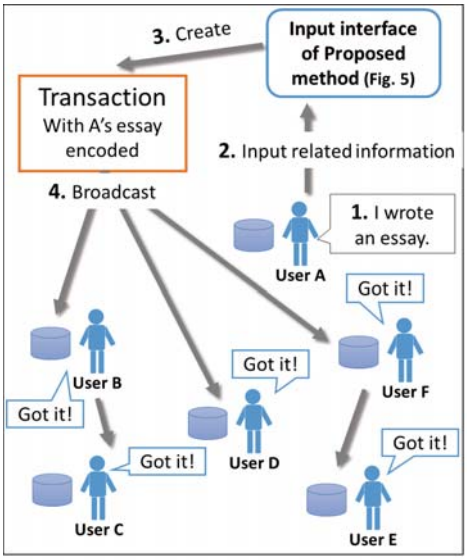

Figure 4. An application example of proposed method

Verification of the existence and integrity of a digital file is based on a hash function. We employ the RIPEMD-160 function. Even a small change in the original file creates a completely different hash value than that of the original file. It is impossible for other people to claim that they created this digital file in an earlier time and produced the hash. Using this feature, a digital file's existence and integrity can be proved. In order to verify a digital file, creator provides the original file to be hashed and converted to a Bitcoin address. Then search it from a blockchain explorer to see the special transaction containing the related information of this file. If the original file has been modified even only one character, a completely different hash will be produced, which will not be found in the blockchain.

4. Evaluation experiments

Here, we set N=3 and used a PDF file. Figure 5 is the homepage of the proposed method. The three required inputs are 1) file and creator names, 2) keywords, 3) and the digital file. The three inputs are converted into three addresses. Then, a transaction is created to pay small amounts to the three addresses. The transaction is broadcasted to Bitcoin's network and is stored in the blockchain.

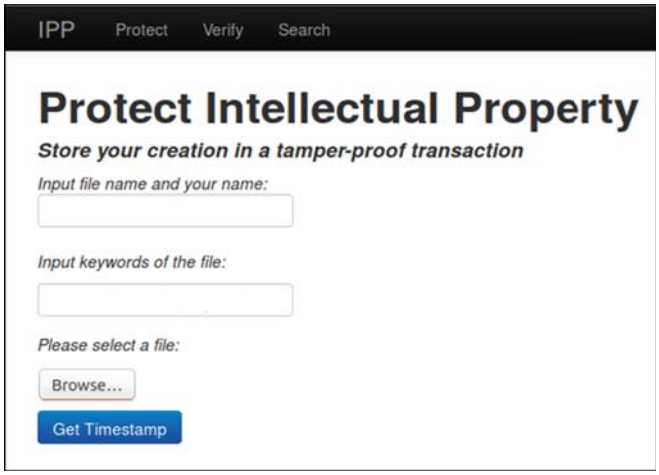

Figure 5 Input interface of proposed application

This experiment was performed to determine the relationship between cost and time. A PDF file was used in this experiment. We used 10 groups, each containing five proposed transactions with five different costs. Thus, this experiment was conducted 50 times. The five transactions in each group were broadcasted to Bitcoin's P2P network approximately simultaneously. This time point is also when the transactions were received by the blockchain (received time). Then we collected another time point for the transactions when they were recorded in the blockchain (included time). We then calculated the period between these two times. The results of the 50 experiments are shown in Table 1.

Table 1 Relationship between time and cost

	Time (min)									
Cost (BTC)	Group1	Group2	Group3	Group4	Group5	Group6	Group7	Group8	Group9	Group10
0.001	15	80	3	32	58	8	30	1	4	11
0.0005	15	80	3	32	58	8	30	4	4	11
0.0001	1407	1342	2671	1002	860	796	1000	1242	1247	1015
0.00005	1407	1342	2672	1002	861	796	1000	1242	255	1015
0.00001	1407	1342	3657	1002	2363	796	1000	1246	1247	1015

As can be seen from Table 1, transactions with higher costs tend to be recorded in the blockchain in a shorter time. When the costs are set to 0.00103 BTC (0.46 USD) or 0.00053 BTC (0.24 USD), it takes an average of 24 min to obtain a trusted timestamp using the proposed method. When the cost is less (e.g., 0.00013 BTC, 0.00008 BTC, and 0.00004 BTC), the average time increases to greater than 1,000 min. The longest time from a transaction being received to being recorded was 3657 min. The results of this experiment show how cost influences time. Although cost is not the only factor that contributes to time, it is one of the factors that can be set and controlled in the proposed method. By performing experiments relative to time and cost, we attempted to determine an appropriate cost by which transactions can be recorded in an acceptable average time. The proposed method timestamps a digital file in an average time of 20 min with the possible cost of 0.00053 BTC (0.24 USD).

5. Conclusion

Present decentralized trusted timestamping based on the blockchain only provide 40-byte storage for digital files. This does not permit sufficient storage to encode other information. The proposed method expands the storage space to a maximum of N*40 bytes, thereby enabling the storage of additional information (e.g., file name, creator name, and keywords). Experimental results indicate that the proposed method can timestamp a file in an average of 20 min at a possible cost of 0.24 USD. We consider that the proposed method can prove the existence and integrity of a digital file.

References

[1] Antonopoulos, A. *Mastering Bitcoin*. O'Reilly Media, Inc., 2014.

[2] *BTProof*. Retrieved from https://www.btproof.com/

[3] e-timestamp. *Protecting Your Intellectual Property*. Retrieved from https://www.digistamp.com/about-us/protect-your-intellectual-property

[4] Gipp, B., Meuschke, N., Gernandt. A. *Decentralized Trusted Timestamping using the Crypto Currency Bitcoin (preprint)*. In Proceedings of the iConference 2015.

[5] Michel, F. (2015). *How many public photos are uploaded to Flickr every day, month, year?* Retrieved from https://www.flickr.com/photos/franckmichel/6855169886/

[6] Nakamoto, S. (2008). *Bitcoin: A Peer-to-PeerElectronic Cash System*. Retrieved from https://bitcoin.org/bitcoin.pdf.

[7] *Proof of Existence*. Retrieved from https://www.proofofexistence.com/

[8] Robertson, M. R. (2015, November 13). *500 Hours of Video Uploaded To YouTube Every Minute [Forecast]*. Retrieved from http://www.reelseo.com/hours-minute-uploaded-youtube/#ixzz3th4DMvS0