Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2016 Proceedings

Mediterranean Conference on Information Systems (MCIS)

2016

Securing Video Integrity Using Decentralized Trusted Timestamping on the Bitcoin Blockchain

Bela Gipp
University of Konstanz, bela.gipp@uni-konstanz.de

Jagrut Kosti
University of Konstanz, jagrut.kosti@uni-konstanz.de

Corinna Breitinger
University of Konstanz, corinna.breitinger@uni-konstanz.de

Follow this and additional works at: http://aisel.aisnet.org/mcis2016

Recommended Citation

Gipp, Bela; Kosti, Jagrut; and Breitinger, Corinna, "Securing Video Integrity Using Decentralized Trusted Timestamping on the Bitcoin Blockchain" (2016). *MCIS 2016 Proceedings*. 51.
http://aisel.aisnet.org/mcis2016/51

This material is brought to you by the Mediterranean Conference on Information Systems (MCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in MCIS 2016 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

SECURING VIDEO INTEGRITY USING DECENTRALIZED TRUSTED TIMESTAMPING ON THE BITCOIN BLOCKCHAIN

Gipp, Bela, University of Konstanz, Germany, bela.gipp@uni-konstanz.de[1]

Kosti, Jagrut, University of Konstanz, Germany, jagrut.kosti@uni-konstanz.de

Breitinger, Corinna, University of Konstanz, Germany, corinna.breitinger@uni-konstanz.de

Abstract

The ability to verify the integrity of video files is important for consumer and business applications alike. Especially if video files are to be used as evidence in court, the ability to prove that a file existed in a certain state at a specific time and was not altered since is crucial. This paper proposes the use of blockchain technology to secure and verify the integrity of video files.

To demonstrate a specific use case for this concept, we present an application that converts a video-camera enabled smartphone into a cost-effective tamperproof dashboard camera (dash cam). If the phone's built-in sensors detect a collision, the application automatically creates a hash of the relevant video recording. This video file's hash is immediately transmitted to the OriginStamp service, which includes the hash in a transaction made to the Bitcoin network. Once the Bitcoin network confirms the transaction, the video file's hash is permanently secured in the tamperproof decentralized public ledger that is the blockchain. Any subsequent attempt to manipulate the video is futile, because the hash of the manipulated footage will not match the hash that was secured in the blockchain. Using this approach, the integrity of video evidence cannot be contested. The footage of dashboard cameras could become a valid form of evidence in court.

In the future, the approach could be extended to automatically secure the integrity of digitally recorded data in other scenarios, including: surveillance systems, drone footage, body cameras of law enforcement, log data from industrial machines, measurements recorded by lab equipment, and the activities of weapon systems.

We have made the source code of the demonstrated application available under an MIT License and encourage anyone to contribute: www.gipp.com/dtt

Keywords: Blockchain Applications, Trusted Timestamping, Video Integrity, Mobile Application.

[1] corresponding author

1 Introduction

The blockchain is a collaboratively maintained and decentralized data structure that has enabled a variety of applications. Beyond the realm of cryptocurrencies, the protocol underlying blockchain has implications in any domain where a trustless, anonymous, and tamperproof means of recordkeeping is required.

The decentralized and collaboratively maintained ledger technology at the heart of the blockchains of cryptocurrencies, such as Bitcoin (Nakamoto, 2008), represents a seminal contribution to Information Technology. Some have described the blockchain as a 'disruptive technology' with the potential to 'revolutionize the interface between economic agents' (Pilkington, 2016). Others have compared blockchain to the paradigm shift that resulted from the introduction of the PC or the World Wide Web (Swan, 2015).

Indeed, the unique characteristics of the blockchain, namely its distributed and tamperproof characteristics, and its public yet anonymous recordkeeping, have enabled the rethinking of applications beyond the realm of finance. Some recently introduced applications include: the creation and transaction of digital assets[2], the design of smart contracts that automatically enforce according to predetermined rules (Kosba, Miller, Shi, *et al.*, 2015); and the blockchain application upon which this paper builds, namely the trusted timestamping of digital data (Gipp, Meuschke & Gernandt, 2015).

Trusted timestamping is a procedure for verifying the existence of unaltered digital data at a specific point in time (Haber & Stornetta, 1990). In a previous paper, we demonstrated how the blockchain of a cryptocurrency, such as Bitcoin, can serve as a decentralized trusted timestamping service if the hash value of a digital file is embedded into the transaction record of the cryptocurrency (Gipp, Meuschke & Gernandt, 2015). In that paper, we introduced OriginStamp (http://www.originstamp.org/), a web service implementing decentralized trusted timestamping (DTT) on the Bitcoin blockchain. Of course, another blockchain could also be used.

The service allows the public to create trusted timestamps for digital files free of charge. To keep fees to a minimum, OriginStamp aggregates the hashes it receives and then performs a single Bitcoin transaction using the smallest transferrable amount (1 Satoshi = 0.00000001 BTC) every 24 hours. With this approach, the operating costs of OriginStamp are covered by its developers, given that they amount to only around 10 USD per year (Gipp, Meuschke & Gernandt, 2015).

Even if Bitcoins would become very valuable in the future, the service would not become prohibitively expensive. Currently, the service maintenance cost is attributable mainly to the transaction fees levied by the Bitcoin network, and not the transacted amount of Bitcoin.

In this paper, we present the scenario in which DTT is used to automatically secure video file integrity for dashboard cameras. The proposed solution enables the cost-effective recording of what we term Tamperproof Video Evidence in the Blockchain (TVEB).

1.1 Scenario

The use of dashboard cameras to record video evidence is increasing. Today, not only police and highway patrol vehicles are routinely equipped with video recording devices, but even consumers are purchasing such systems. Dash cams are more widely adopted in countries where reckless driving is a problem, where law enforcement is corrupt, or the police is not trusted (Paul and Kitson, 2014). However, also in the UK, the number of consumers protecting themselves against insurance fraud by purchasing dashboard cameras is on the rise (Gammell, 2015). If a fraudulent individual purposefully initiates an accident, a dash cam user can present the video recording to their insurance in an attempt to prove that they are not to blame. As Mark Godfrey, director of RAC Insurance, states "*Dashcams give*

[2] http://coloredcoins.org/

drivers an added level of protection to guard against unexpected malicious events which they might otherwise struggle to prove." (Gammell, 2015).

Additionally, the existence of a video recording can prevent disputes, thus saving time and legal fees. Only recently, a German appellate court ruled that the use of video material from dashboard cameras is permissible as evidence to prosecute traffic violations (Oberlandesgericht Stuttgart, 2016). Given this development, it is more crucial than ever to have secure means for *proving that the video footage has not been manipulated after the accident*.

Currently, there is no simple, cost-effective and automated method available to consumers to prove that video footage was not tampered with after a specific point in time. If the authenticity of a video file is contested, the status quo requires testimony of witnesses, or the hiring of experts to verify that the digital file has retained its integrity. These approaches are costly and time consuming. For example, it took two years for an international investigation into Malaysia Airlines Flight 17, which was shot down over the Ukraine in 2014, to confirm that several satellite images released by the Russian military had been digitally manipulated (Kramer, 2016). Additionally, there is no guarantee that a carefully and meticulously performed fraudulent modification of metadata or video frames will be discovered.

A secure and cost-effective method is needed to allow anyone to verify that a video file has not been tampered with after a claimed date. To address this scenario, we introduce a mobile application that acts as a dashboard camera, which automatically creates a tamperproof distributed trusted timestamp on the Bitcoin blockchain for video files recorded during a collision.

2 Background: Trusted Timestamping, and Characteristics of the Blockchain

The concept of attaching a timestamp to physical documents or evidence to certify their integrity at a specific time is not new. Historically, public notaries, governments, or other trusted third parties have been tasked with verifying the authenticity of files. However, this approach has the shortcoming that timestamps are vulnerable to a central vouching party.

As digitally stored data began playing an increasingly important role in business in the early 1990s, 'Trusted Timestamping' schemes for digital data were proposed (Fischer, 1991; Haber & Stornetta Jr, 1992). In 2001, the Time-stamp Protocol (TSP), or RFC 3161 standard, was introduced, which relies on public key infrastructure to generate trusted timestamps (Adams & Pinkas, 2001). This protocol is still being used by today's Time Stamping Authorities (TSA).

Improvements to the RFC 3161 standard were made in the 2005 ANSI ASC x9.95 standard for trusted timestamps, which defined additional data-level security requirements (American National Standards Institute, 2005). However, for both protocols, the specified timestamping process relies on a *centralized* trusted third party (TTP). The TTP acts as the TSA and is tasked with both issuing timestamps and verifying timestamp validity. Because the TSA is the central record-keeper, the integrity of timestamps is vulnerable to the trustworthiness and the security standards enforced by the TSA. If, for example, the private key used by the TSA for the public key encryption scheme is compromised, the validity of timestamps is at risk.

With the introduction of cryptocurrencies, the blockchain for the first time allowed for tamperproof, trustless, anonymous, and automatically programmable 'Decentralized Trusted Timestamping' (DTT) (Gipp, Meuschke & Gernandt, 2015). DTT is the concept of submitting the hash of a digital file to be included in a transaction made in a cryptocurrency network, which results in the hash being permanently stored in the decentralized and collaboratively maintained blockchain data structure.

Figure 1 shows the procedure for DTT as implemented in the OriginStamp service, which is described in detail in Gipp et al. (2015). As can be seen in the figure, the procedure involves an intermediate step in which the individual file hashes are collected and hashed again. Only this ‘aggregated hash’ is used to initiate the Bitcoin address calculation to which the minimum transactional amount of Bitcoin is sent. By aggregating the hashes, OriginStamp is able to offer the DTT service free of charge, because the transaction fees that are levied by the Bitcoin network for each transaction (currently around 0.0001 BTC) are kept to a minimum. The downside of this approach is that timestamps are created with a delay, of up to 24 hours, depending on when the hash is received. However, for the majority of consumer use-cases this granularity is sufficient. Additionally, users have the option to pay a small fee on the OriginStamp platform to immediately send their hash for DTT in the Bitcoin network.

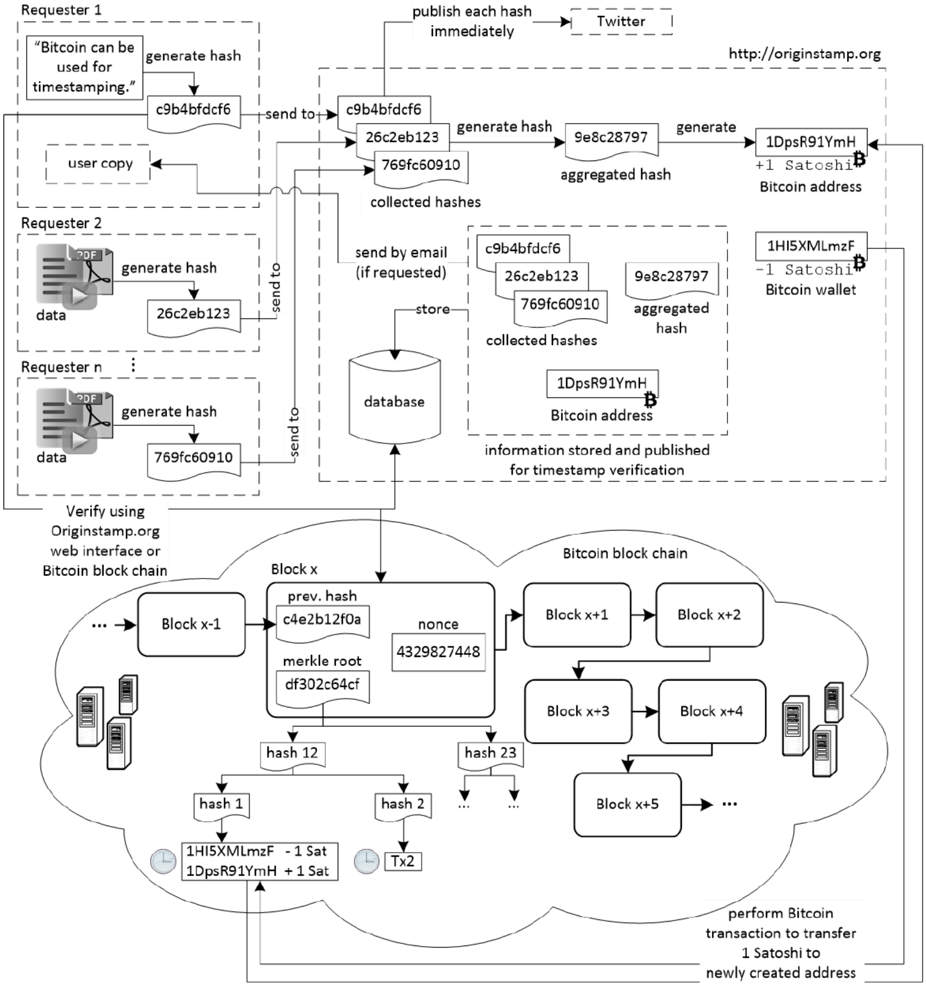


Figure 1: Procedure for Decentralized Trusted Timestamping on the Blockchain, source: (Gipp, Meuschke & Gernandt, 2015)

In the following section, we introduce an automated solution that can run on a user’s smartphone to secure file integrity and automatically attach a trusted timestamp[3] to video files recorded during a vehicular accident.

[3] For the sake of brevity, the terms ‘timestamping’ or ‘trusted timestamping’ are used interchangeably to refer to the ‘decentralized trusted timestamping procedure relying on the blockchain’ as described in Figure 1.

3 System for Trusted Timestamping of DashCam Video

The Android native application presented in this paper uses the phone's built-in camera to continuously record video in the background while the vehicle is moving. If the phone's built-in sensors register a sudden impact, the relevant video files are extracted and a SHA256 hash is calculated to secure the content of the file and prevent future tampering. The mobile application immediately transmits the hash to the OriginStamp trusted timestamping service. OriginStamp in turn submits the hash to the Bitcoin blockchain for persistent and tamperproof storage by performing a transaction with the minimum transactional amount to a Bitcoin address generated from the hash. Storing of the hash in the blockchain makes it impossible to tamper with the video file in retrospect without the change being apparent.

The components of the system can be divided into the two types of functionalities:

- Core functionalities – These functionalities are implemented in an Android *Service Class* that is capable of running in the background even when the user switches between applications.
- Non-core functionalities – These functionalities must not always be running. The only non-core functionalities are *Hash Submission Notification*, and the implementation of *Viewing and Verifying Collision Videos*.

Since collisions cannot be anticipated, all core modules of the system must continuously run in the background. The application's core functionalities are:

1. continuously record video
2. detect a collision incident using the phone's accelerometer
3. identify and merge video files relevant for the "collision incident"
4. generate a hash (SHA256) of the prepared video files
5. transmit the hash for trusted timestamping in the blockchain
6. notify the user-specified contacts

Figure 2 gives an overview of the system's actions in the case of a collision.

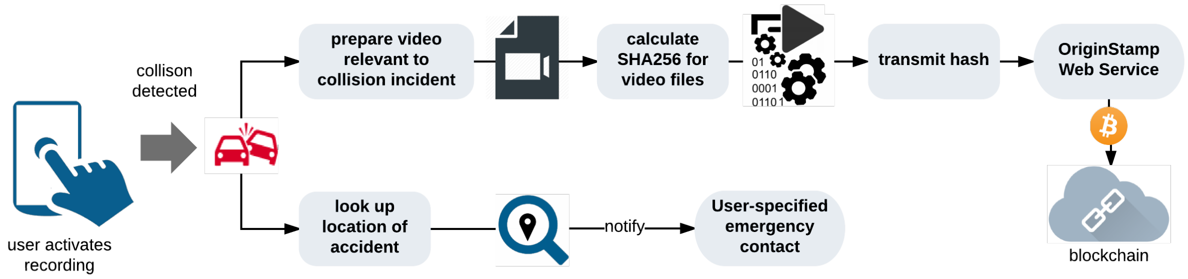

collison detected
user activates recording
prepare video relevant to collision incident
calculate SHA256 for video files
01 0110 0001 0110 1
transmit hash
OriginStamp Web Service
blockchain
look up location of accident
notify
User-specified emergency contact

Figure 2. Sequence of application actions upon collision detection

To record the critical time of a collision without quickly using up device storage, the application continuously records sections of video that are temporarily stored and then overwritten. If an impact is registered, the application combines the recording at the time of impact with the temporarily stored older recording, and with the recording from immediately after the collision. Since only the video footage from around the time of the collision must be stored, the default video quality is set to maximum.

Even if the user navigates to another application on the phone, the dash cam application continues to record video in the background. The video being recorded is shown to users as an unobtrusive 50x50 pixel box in the top right corner of the screen, refer to Figure 5, b). This allows users to make regular use of their smartphone, while reminding them that video is being recorded in the background. To achieve this feature, the application creates a new *Surface View* to which an instance of Android's

MediaRecorder Class is attached. This Surface View is created in the *Service Class*, which runs in the background and thus ensures that the view is not destroyed when the user switches between applications.

3.1 Collision detection

The built-in accelerometer of the user's smartphone is used to detect the impact of a collision. Acceleration values are queried every 100 milliseconds and checked for all three axes: x, y, and z. A sudden change, as determined by the magnitude of the difference in the recorded values is used to signal the occurrence of a collision. The application uses the *SensorEventListener* to access the accelerometer data. Upon detecting a significant change in the sensor values, the *Listener* calls the *onSensorChanged* method. Additionally, a check for GPS movement is performed. If significant movement continues after the collision, then no collision is declared.

3.2 Extracting and securing video

To prevent the application from using up unnecessary memory on the user's phone, video recording occurs in continuous chunks of 10-second intervals. Figure 3 shows how this approach is used to free up device memory and allow for continuous recording in the background.

If no collision is detected within the "current" time window, then the "previous" recording is discarded, the "current" time frame is saved as the new "previous" timeframe, and a new "current" recording window is started. Using this approach, no more than two video chunks are stored in memory at any given time. This process continues until the user exits the application or a collision is detected.

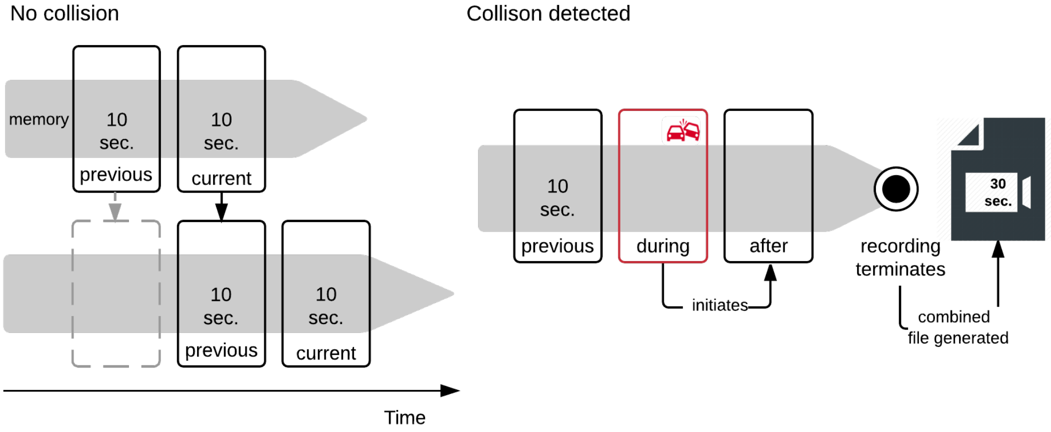


Figure 3. Video recording procedure

If a collision is detected in the current time window, the application records an additional chunk of video. Given this approach, in the case of a collision, the video recording consists of three 10 second chunks, resulting in a video clip of 30 seconds. This relatively short recording length was implemented to free up static memory on the device and allow users to make regular use of their phone while recording video in the background. Of course, the recording length can be customized in the future.

In addition to automatic collision detection, the user can press a button within the application to prompt the immediate saving and timestamping of video at any time. This allows capturing events related to crime, traffic, wildlife, extreme weather, etc.

Once the video files have been recorded and ordered, a hash is computed of the resulting video file. The variable *finalByte* contains the data of the ordered video files in byte form. The byte form of the video data is hashed using SHA-256. This hashing algorithm was chosen because it is considered secure against collision attacks (Gilbert & Handschuh, 2003). The resulting hash is also stored to the user's smartphone for future reference and file verification.

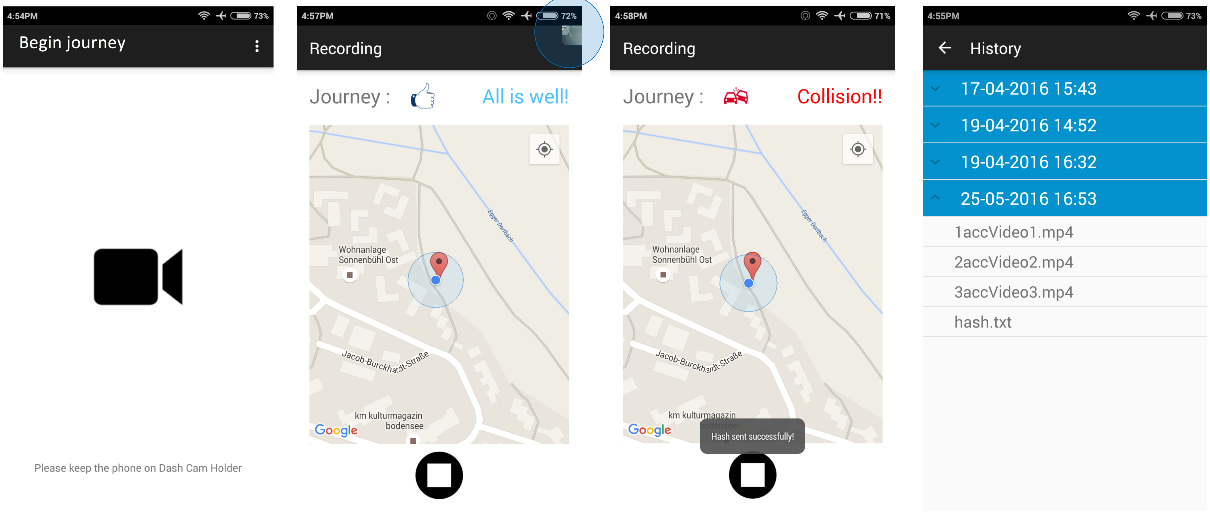
3.3 Transmitting hash for trusted timestamping

For the hash of the video file to be associated with a tamperproof decentralized trusted timestamp in the Bitcoin blockchain, the hash is transmitted to the OriginStamp service. A POST request to OriginStamp's REST API is made in the *sendHashServer* method. The API of OriginStamp can be used free of charge with no rate limit[4].

Once the hash has been transmitted, the user is shown a confirmation on their mobile device. Additionally, the application notifies the user's emergency contacts, which the user can specify within the application. A simple SMS sending mechanism is used to notify the contacts of the collision location. In the future, the system could be expanded to automatically notify the region's emergency medical services.

3.4 System frontend

Figure 4 shows several screens of the user interface of the application. Image a) shows the start-up screen where the user is prompted to begin video camera recording. Once the recording is initiated, the application displays the user's location on a Google map and an unobstructive square of the real-time recording in the upper right corner of the phone at all times, image b). Upon the detection of sudden impact, the application registers a collision incident, prepares the required video files to be persistently stored and the screen flashes a confirmation of hash submission to the user, image c). In image d), the user can browse video recordings of accident incidents, and can view the hash digest stored in a .txt file.



a) b) c) d)

Figure 4. Application interface

3.5 Proving video file integrity

To prove that the video file has not been modified since the hash was embedded in the blockchain, the user can now provide the original video files as stored on their smartphone together with the file hash to the relevant authorities. The lawyer, claims agent, or anyone wanting to verify the file's authenticity can compare the presented video file's hash with the hash stored in the Bitcoin blockchain. Verifying the existence of the file hash (i.e. the Bitcoin address generated from the hash) can be performed using the OriginStamp website, or using any Bitcoin blockchain inspection tool, such as blockexplorer.com, or blockchain.info.

[4] http://www.originstamp.org/developer

Since the mobile application automatically performs DTT in the background with no user involvement, the user has no time to tamper with the video files. If the user were to tamper with the video files in retrospect, the resulting hash will be different than the hash submitted to the Bitcoin blockchain, which will disavow their claim to file integrity and time of file existence.

Given the recent ruling of a German appellate court to accept video evidence for prosecuting traffic violations (Oberlandesgericht Stuttgart, 2016), it is only a matter of time before courts must also consider secure methods for proving the validity of video files captured by dashboard cameras. As observed by D. Cawrey, "courts around the world will eventually need [to] become familiar with the use of cryptographic hashes as a form of verification" (Cawrey, 2014). Currently, precedence has yet to be established regarding the obligation of courts to recognize trusted timestamping using the blockchain of a cryptocurrency as evidence in court proceedings.

3.6 Limitations

The current application submits the hash of the video file to Bitcoin's blockchain with an involuntary delay due to the approach of the OriginStamp service aggregating all hashes received over a 24-hour period and only submitting a master hash to the blockchain to save transaction fees and offer the service free of charge. In the future, the granularity of timestamps can be improved by giving users the option to submit each hash individually without delay. Currently, OriginStamp additionally publishes each received hash immediately to Twitter to mitigate this limitation (https://twitter.com/originstamp).

Future implementations to automatically secure video integrity using blockchain technology could explore the use of non-Bitcoin coupled blockchains. A benefit of using, for example, Ethereum[5], or a data layer built on top of Bitcoin's blockchain, such as Factom[6], is that the application remains uninfluenced by potential future Bitcoin transaction fee increases. Currently, the maintenance costs of the service are almost exclusively attributable to the transaction fees levied on transaction in the Bitcoin network, since these fees are required as an incentive for Bitcoin miners to integrate transactions into new blocks. A second benefit would be that Bitcoin's blockchain is not unnecessarily bloated with non-Bitcoin related transactions. However, a major drawback of using a non-Bitcoin blockchain is that the service no longer benefits from the trustworthiness that comes with its widespread use and a market capitalization of approximately 10 billion USD. Currently, one can assume that as long as Bitcoins hold any value, the Bitcoin blockchain cannot be manipulated.

4 Future Work

4.1 Web Portal for Emergency Services

In the future, the proposed system's mobile frontend will be extended with a web portal component to allow app users to better manage and download their video files, or enable friends or emergency services to check up on the users of the app.

Such a web portal would be supplementary to the presented mobile application and could allow users to enter medical data to be used by emergency services. Emergency medical services could have an access point to this portal that allows them to see the location of a collision, the emergency contacts, and the user-specified medical data, such as blood type, allergies to medication, etc. The web portal will update in real-time with the data and video files recorded during collision incidents in the mobile application.

[5] https://www.ethereum.org/

[6] http://factom.org/

4.2 Outlook

In the future, the concept of automatically securing video footage with decentralized trusted timestamps in the Blockchain could be applied to other domains. Beyond the specific example of a dashboard camera application – as presented in this paper – other areas of use that could benefit from securing Tamperproof Video Evidence in the Blockchain (TVEB) include:

- public and private video surveillance systems for security, or theft prevention.
- video conferencing systems, where the automatic timestamping of important meetings, or contract agreements could protect individuals or firms, for example, when discussing ideas that may be patented.
- the movie and creative industry, where the trusted timestamping of video footage at the time of production could be used to prove priority or copyright claims.
- aerial footage from civil and military aircraft or drones. For example, the video footage of a drone documenting the current state of buildings and infrastructure, or of a nature reserve, could be automatically associated with a decentralized trusted timestamp to prevent governments or special interest groups from disputing the observations.
- body cameras worn by police and law enforcement officers that implement automatic DTT of video files to prove video file integrity.

Beyond securing video integrity, we propose that the concept described in this paper is expanded to any use case where the integrity and creation time of digitally recorded data must be immediately and automatically secured.

Given this scenario, other trusted timestamping use cases involve: cameras, scanners, environmental sensors, self-driving cars, and any data-recording instrument used by industry or in research labs. The log files of machines, including their usage statistics, or error messages, could also be automatically affixed with a trusted timestamp using blockchain technology. The decentralized trusted timestamping of log data could verify the integrity of the log record and thus help companies receive insurance claims, or prove that security mechanisms were in place in the case of workplace accidents.

A military application for automatic decentralized trusted timestamping is the secure documentation of weapon activities. The approach could be used to, for example, verify the time of ammunition discharge, or to confirm that the required maintenance of a weapon systems has taken place.

5 Conclusion

Methods for verifying video file integrity are of crucial importance, especially if video files are to be presented as evidence in court. In this paper we proposed the idea and presented an implementation of creating trusted timestamps for the video footage of dashboard cameras to prove that the video footage was not tampered with after an accident. A fully functional application was developed and released under an MIT License. It can be downloaded free of charge here: www.gipp.com/dtt/

The presented mobile application continuously records video in the background. If a vehicular collision is detected by the phone's built-in sensors, the application automatically transmits a hash of the relevant video files to be stored in Bitcoin's decentralized and tamperproof transaction ledger, i.e. the blockchain.

Currently, courts do not routinely accept video footage as evidence, because it is impossible to prove that the files were not manipulated after the accident. By using the blockchain of a cryptocurrency to store a hash of the video file it can be proven that the footage was not manipulated. Any tampering with the file in retrospect would result in a file hash that no longer matches the hash that was embedded in the blockchain.

Beyond the scenario of timestamping video captured by dashboard cameras, the approach of automatically associating timestamps created on the blockchain with digital data has implications for various industrial and military use cases. Among others, use cases for surveillance systems, body cameras, drones, log data of machines, and weapon activities were presented.

A decentralized trusted timestamp of this paper has been stored in the Bitcoin blockchain: http://www.originstamp.org/u/mTNFhKzKaJ

References

Adams, C. and Pinkas, D. (2001). *Internet X. 509 public key infrastructure time-stamp protocol (TSP)*.

American National Standards Institute (ANSI) (2005). *X9 x9.95-2005 trusted time stamp management and security*.

Cawrey, D. (2014). *How Monegraph Uses the Block Chain to Verify Digital Assets*. 2014. CoinDesk. URL: http://www.coindesk.com/monegraph-uses-block-chain-verify-digital-assets/ [Accessed: May 20, 2016].

Fischer, A.M. (1991). *Public/key date-time notary facility*. U.S. Patent 5,001,752.

Gammell, K. (2015). *Is it time to invest in a dashcam?* The Guardian. February 2.

Gilbert, H. and Handschuh, H. (2003). "Security analysis of SHA-256 and sisters." In: *International Workshop on Selected Areas in Cryptography*. Ed. by M. Matsui, R. Zuccherato. Springer Berlin Heidelberg. pp. 175–193.

Gipp, B., Meuschke, N. and Gernandt, A. (2015). "Decentralized Trusted Timestamping using the Crypto Currency Bitcoin." In: *Proceedings of the iConference 2015*. March 2015, Newport Beach, California.

Haber, S. & Stornetta, W.S. (1990). "How to Time-Stamp a Digital Document." In: *Advances in Cryptology—CRYPTO '90 Proceedings*. Springer Berlin Heidelberg. pp. 437–455.

Haber, S.A. & Stornetta Jr, W.S. (1992). *Digital document time-stamping with catenate certificate*.

Kosba, A.E., Miller, A., Shi, E., Wen, Z., et al. (2015). "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts." *IACR Cryptology ePrint Archive*.

Kramer, A. (2016). *Russian Images of Malaysia Airlines Flight 17 Were Altered, Report Finds*. The New York Times. July 15. URL: http://www.nytimes.com/2016/07/16/world/europe/malaysia-airlines-flight-17-russia.html [Accessed: July 15, 2016].

Nakamoto, S. (2008). "Bitcoin: A peer-to-peer electronic cash system.". URL: https://bitcoin.org/bitcoin.pdf [Accessed: May 28, 2016].

Oberlandesgericht Stuttgart (2016). Court Report: „Dashcam"-Aufnahmen können zur Verfolgung schwerwiegender Verkehrs-ordnungswidrigkeiten grundsätzlich verwertet werden. *www.olg-stuttgart.de*.

Paul and Kitson, R.G. (2014). "Smile you're on dash-cam camera: More motorists are installing in-car recorders to capture instances of bad cycling and driving." *The Independent*. 2 February.

Pilkington, M. (2016). "Blockchain Technology: Principles and Applications." *Research Handbook on Digital Transformations*, edited by F. Xavier Olleros and Majlinda Zhegu. Edward Elgar.

Swan, M. (2015). *Blockchain: Blueprint for a New Economy*. O'Reilly Media, Inc.