A Blockchain Implementation for the Cataloguing of CCTV Video Evidence

Michael Kerr
Australian Criminal Intelligence Commission
Melbourne, Australia
michael.kerr@acic.gov.au

Fengling Han
School of Science (Computer Science)
RMIT University
Melbourne, Australia
fengling.han@rmit.edu.au

Ron van Schyndel
School of Science (Computer Science)
RMIT University
Melbourne, Australia
ron.vanschyndel@rmit.edu.au

Abstract

Presented here is a functional implementation of Distributed Ledger Technology applied to the task of cataloguing CCTV video evidence. We describe and demonstrate a prototype camera that participates in blockchain creation in real time, and the system designed to manage and coordinate its distribution and use. This application is of specific interest to law enforcement agencies charged with the management of high volumes of CCTV evidence. We discuss applicability and scalability with reference to simulation results and real-world testing. The combination of blockchain technology with a novel digital watermarking application is demonstrated here providing immediate benefit against an existing real-world problem of trustworthy evidence protection in distributed network environments.

1. Introduction

Current technology trends are rushing to embrace Distributed Ledger Technology (DLT), otherwise known as Blockchain. Blockchain's central concept of taking some unique data from one record and immutably linking it to the following record is tremendously elegant in its simplicity. It is not a new concept, such linking was described as early as 1991 in research concerning sequential signing of digital documents [1]. It was later, in Satoshi Nakamoto's Bitcoin whitepaper [2] that the idea of "chaining blocks" was used to describe the creation of linked records, and this language later morphed into the term Blockchain. In recent years Blockchain has been integrated into many different applications; some of these applications are compelling, others however possess weak use case arguments and some others are fuelled entirely by entrepreneurial fervour. With this in mind we show here that DLT can provide real enhancement to specific applications, and we provide a prototype example in our law enforcement scenario using a combination of real-world testing and simulations.

The results of these tests provide performance and scalability indicators regarding our system's block creation algorithm and the distribution of data. It is designed to facilitate management of Closed-Circuit Television (CCTV) evidence in specific scenarios, our testing reflects those scenarios and is focused on working within a typical networked environment as would be used for collection of potential evidence.

CCTV's migration to IP based transmission, combined with rapid increases in analysis capability have opened up enormous opportunities for law enforcement agencies (LEA). However, the ubiquitous nature of CCTV installations has for some time also been the subject of concern for privacy advocates across the globe. This risk has been well examined from both a technical and ethical perspective[3].

Individual businesses or law enforcement organizations now control vast collections of CCTV assets, their sharing of collected data can lead to the application of various intrusive analysis techniques in ways that may be unintended. Context is key, the intended purpose of specific camera installations form a primary justification for that systems approval. The propagation of that digital video into other networked systems can violate any or all privacy caveats originally imposed on such sources with no visibility at its origin. This is a well understood problem of interest to citizen advocacy groups and those agencies charged with protecting society. Potential scenarios put forward only a few years ago[3] are now technically mature and being routinely explored by government agencies in places such as Queensland, Australia[4], and more broadly though the Council of Australian governments (COAG)[5].

These growing concerns can be addressed by robust compliance requirements imposed on the creation and collection of such data. It has been previously argued that a technological solution to this concern would be the implementation of inherent auditing capability within the data itself. We have previously recommended solutions that include audit systems modelled on the Chain of Custody concept, a mechanism well suited to the safeguarding of CCTV data against unauthorised distribution or repurposing[3]. This concept provides a strong initial principle that can be further enhanced using the blockchain concept.

Law enforcement generally has a requirement to demonstrate the integrity and legal authenticity of any video data produced as evidence in legal proceedings. Traditional methods employed to assist in the legal authentication of evidence include manual record keeping using documents

like maintenance logs or shift registers; modern methods can incorporate technologies such as digital watermarking and file hashing. None of these methods provide a complete solution, manual methods being particularly susceptible to human fallibility, whilst technical mechanisms provide only partial solutions; they have their own gaps in protection that may expose attack vectors to malicious actors or dispute in the legal arena. The benefit of applying Blockchain technology to the management of digital evidence has already been identified within law enforcement. Public releases from the UK Ministry of Justice (MoJ)[6] have theorised on how a blockchain system would protect catalogues of evidence during its Chain of Custody life cycle. The MoJ authors do not offer specifics on how such a system would be implemented. The system demonstrated herein closely follows the same application and offers a comprehensive solution to the law enforcement scenario.

There exists recent work on utilising the Bitcoin blockchain for the decentralisation of evidentiary metadata such as timestamps in video[7]. There are no doubt applications for this approach, however, government agencies are unlikely to willingly cede control of their evidential audit capability to an international third party such as the Bitcoin network. There will always be a requirement for an internally managed, but transparently auditable, ledger system.

By exploiting Blockchain's integrity and assurance capabilities and applying them to our audit and authentication requirement we demonstrate a system that is novel, practical and demonstrably useful. A further goal of our system is the end to end protection of video evidence, from collection to presentation. This has been achieved through a novel approach to on camera blockchain creation combined with the embedding of digital watermarks prior to transmission from the cameras. This system can provide assurances on the integrity and context of video data in such a way that can satisfy LEA requirements whilst also providing visibility to civil interest groups wishing for insight into the distribution and approval frameworks surrounding CCTV surveillance.

2. A blockchain system for CCTV evidence protection

Camera sources may include public spaces, covert installations, third party agencies or private entities that assist investigations. A viable blockchain system will link to the video stream as primary data whilst recording vital metadata pertaining to that video's creation, relevance and integrity. It is a feature of our system that the blockchain creation and verification process can be performed entirely independently of the Video Management System (VMS).

Architectural design reflects the functional components of the system. Individual CCTV cameras participate in a mutually assured network contributing to blockchain creation that is distributed amongst multiple ledger locations. Figure 1 shows the system and its components working with existing VMS systems, and interacting directly with cameras.

Master Nodes communicate directly with cameras, coordinating the distribution of block sequence numbers and managing the integrity of the blockchain sequence. The Primary Ledger resides on the Master Node, this is the complete log of transaction data produced by the camera network.

The deployment of Slave Nodes, particularly to external locations, can ensure the ledger cannot be corrupted without broad collusion of Master and Slave node databases. There can be any number of Slave Nodes, they listen for published updates to the chain that are broadcast from Master Nodes. They do not communicate directly with cameras and therefore can be hosted on entirely separate networks and secured appropriately for evidential or archival storage. This provides record keeping compliance that is a requirement for any evidential recordkeeping system. Slave Nodes can also participate in the verification requests.

Individual cameras contribute transaction data specific to their environment and create the blocks onboard the device. Cameras communicate with Master Nodes, requesting the last block hash file, they build and return a completed block which is accepted into the chain.

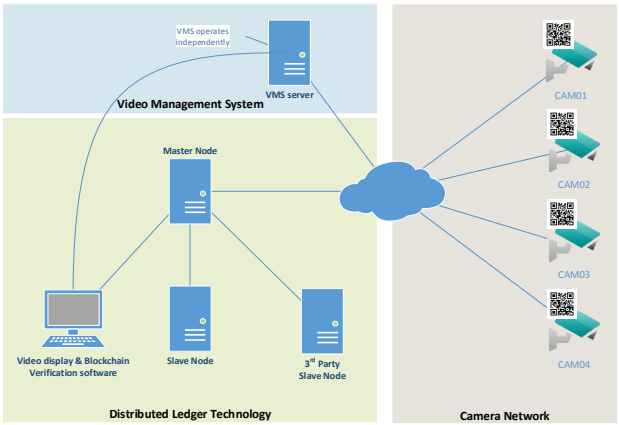

Figure 1 DLT system architecture

2.1. Network environment & communications

In a typical agency deployment, the network could be expected to consist of a distributed camera network spanning several locations, security zoned networks containing the VMS, and further remote locations for client access. To work effectively in such environments, our software system is highly distributed; Master Node hosts are positioned with direct communication access to the cameras as well as broadcast capability to Slave Nodes. Slave Nodes can be located in highly secure network zones,

or even at external locations, acting as a secure real time copy of the ledger. Verification software communicates with both Master and Slave Nodes but has no requirement to access the VMS or the camera network.
Communications between devices and nodes is via a reliable messaging queue system (Rabbit MQ)[8] offering routing capability as well as security and persistence. The message queue service can be located on any centrally accessible host; the ability to decouple the system from the communications infrastructure makes for a scalable and adaptive solution.

2.2. Camera metadata

Regarding video surveillance data. legal and administrative requirements differ internationally and even regionally in the case of Australia or the US. In different locations what organisations can and cannot collect under warrant, share between agencies or present in court will differ. For our reference system we use a generic set of metadata properties shown in Table 1, they would be relevant for almost all camera deployments, and can be easily extended and their values modified in real time.

Property	Description
cameraID	The unique identifier for the physical camera
time	The time of the block creation (Unix)
location	A method to determine the cameras physical location e.g. GPS position or address.
warrant_number	The reference number to the formal authorisation of the camera
targetID	A method to determine the intended target of the installation.

Table 1, Minimum common properties

2.3. Block transactions and creation

Block creation occurs on the cameras themselves, video segments are linked to blocks via the embedding of a watermark into the video stream. Any video segment can be cross referenced against its relevant metadata. Conversely, each block implies the existence of a corresponding video segment, with the archive watch (section 2.5) service deployed, neither the video or the block record can be deleted without detection. Block requests and updates occur between all cameras and the Master Node before broadcasting to Slave Nodes.
Differing from some blockchain systems, there is no block election process, blocks are managed and indexed centrally by the Master Node. The camera requests the next block ID and the hash of the previous record. The Master Node responds with this information and locks the chain from any updates. During this lock any subsequent requests for the next block are deferred for a millisecond value to allow the first camera time to finish its block and submit it back to the Master Node. Integrity of the block sequence is thus managed centrally.

To produce a new block the camera receives two values from the Master Node; the current blockID (bID_i) and the output of the previous block's hash function (H_{i-1}). With its own current transaction data being time (T_i), cameraID (C_i), warrant_number (W_i), Location (L_i) and targetID (tID_i) we define a 256-bit SHA hash function, H_i, as

$$H_i = sha256(bID_i + T_i + C_i + W_i + L_i + tID_i + H_{i-1}) \quad (1)$$

The camera has enough information to produce a new block (B_i) as

$$B_i = bID_i + T_i + C_i + W_i + L_i + tID_i + H_i \quad (2)$$

H_{i-1} links the current block to the previous one, with the output of the H_i function being also added to the next block, B_{i+1}.
In contrast to cryptocurrency focused blockchain implementations, there is no use for the Hashcash type "Proof of Work" requirement introduced in Bitcoin [2]. Such a process introduces difficulty to the block creation process in order to limit the number of blocks. This has the obvious monetary benefit of appreciating the value of the token by ensuring its rarity, but on an embedded device introducing complexity for its own sake is undesirable.
Other security mechanisms are implemented off the chain to eliminate malicious or spurious blocks. Protection from unauthorised block creation or Denial of Service (DOS) style attacks are implemented through a layered security model. Access control is implemented both in the message queue infrastructure and in the systems own custom message protocol. After meeting the message queue's access control requirements, messages are checked for correct format before a participation key is assessed by the Master Node. If the format is unexpected or the key is rejected the message is dropped; This occurs prior to locking the blockchain. Block creation does not commence until the message's authorization is verified and its format is confirmed as valid.

2.4. On camera digital watermarking

Each blockchain record is referenced in the video stream via a digital watermark. The watermark payload contains the block ID, which links the video sequence to associated data within the Blockchain. There are multiple alternative watermarking strategies that are of interest, however, there exist several limiting requirements.

- The scheme must be computationally inexpensive. In an effort to provide the best quality video most CCTV cameras are pushed to their hardware limits. This leaves limited resources for complex additional processing.
- Image quality is paramount for CCTV camera vendors, this is a key market differentiator in the industry. Vendors

will not adopt a system that reduces current image quality expectations.

• Portability is important. Vendors use a range of open and proprietary encoding methods including H.264 and H.265, MP4 and MJPEG. Although watermarking in the transform domain generally offers better performance it is unlikely to be compatible across a range of vendor specific codecs.

With consideration to these limitations a visible spatial domain watermark was chosen. The watermark payload is updated in real time and is inserted in each frame prior to any compression. The Quick Response (QR) [9] standard provided an open and portable method of linking video sequences to the ledger that can be verified both programmatically and visually.

The prototype system's CCTV cameras request a new block sequence number at a timed interval. This triggers the creation of a new block and an updating of the embedded QR code. This process is managed in its own threaded context to minimise the impact on normal camera operation. A range of triggers could initiate this process, for example: the opening or closing of a physical input connected to the camera, a locally processed movement analysis event, or the remote triggering via some Internet of Things (IoT) management platform. These triggers could form part of the transaction data of the block payload. Our time-based trigger is useful for testing as it enables a constant and controllable frequency of block creation, but it has only limited real world application.

2.5. Archive monitoring

An Archive monitoring process completes the cycle of evidence acquisition to archiving. This process runs on the VMS's file storage and detects file save and delete events. By evaluating a file's embedded watermark, the service produces a create or delete event linked to the original block ID and publishes it to the Master Node. Combined with the cameras block creation functionality the system delivers a unique capability by cataloguing video evidence from the point of creation through to its storage and finally its deletion.

2.6. Video authentication

The following are some of the uses of the verification tool.

• Verify the block ID of a specific video clip and examine its specific blockchain metadata. After identifying video segments of interest, the Verification Tool can retrieve the linked block and display any associated metadata for that clip. In our implementation this details the warrant, the camera location and the target details. A recursive verification performs a rebuild of adjacent blocks to compare the recomputed hash values with the original ledger.

• Perform an audit of all produced video linked to the blockchain. To ensure video product is stored and retained as per any legal requirements, blocks could contain a minimum retention period that when used in conjunction with the timestamp can audit the retention of data in the VMS. Even if video is maliciously removed from the system its associated block cannot be deleted or modified. The absence of video that links to a specific block and violates the retention metadata directives would be a reportable event.

• Link target identification to warrant. Information on specific camera approvals may be found in warrant documentation, analysing specific footage provides the block ID within the chain. Further exploration of the blockchain can enable the collection of all associated cameras under that same warrant ID. The collection could be reported on together along with their own verification outcomes. This could be a valuable audit function for prosecution, or defence, wishing to locate new related evidence on an investigation.

3. Implementation & testing results

3.1. VMS compatibility

The system is designed to work in tandem with existing VMS platforms by inserting the watermark message on the camera, prior to the video being ingested into that system.

Watermark retrieval and blockchain verification is performed outside of the viewing platform on exported video. This process fits in well with the preparation of video for submission as evidence, the blockchain forming a much longer Chain of Custody than simple hashing of video files at rest on the VMS file system. A key requirement is the robustness of any watermarking feature, it must survive both the camera's compression and any transcoding performed by the VMS itself. We have verified compatibility with two market leading VMS vendors, Genetec[10] and Milestone[11].

3.2. Testing in a real-world environment

Two prototype hardware cameras were installed into a production CCTV environment to test interoperability and reliability. A further fifty-three virtual cameras were run against the same Master Node to test scalability. Over one million blocks were created on the system during this period.

Two block creation rates were tested, both using a consistent camera frame rate of 15 frames per second; firstly 1 per 900 frames, yielding 1 block per minute, secondly 1 per 15 frames, being 1 block per second. From a law enforcement perspective there are likely no practical scenarios that would require block creation at this higher frequency, other than very large numbers of cameras. Our underlying message queue system is capable of transaction speeds far in excess of our requirements [12], and the server hardware (hosting both Nodes and the virtual cameras) was provisioned for high performance on a local cabled

network, we therefore were able to isolate the blockchain creation algorithm for performance testing.

We defined the camera and node creation process P, as

1. Request bID_i and H_{i-1} from Master Node
2. Calculate B_i as per (2)
3. Submit B_i to Master Node

A sample of 20 transactions found P to take a mean time of .019 milliseconds, during which no other camera can request bID_i. This presents an obvious limitation on the maximum number of blocks per second of 52. However, achieving this number would require optimal sequencing of requests into contiguous time slots, which is not the case at all. With no mechanism to prevent it, one or more cameras create contention by requesting bID_i whilst the chain is locked.

Our virtual camera simulation demonstrated a much lower practical limitation than this. When a camera requests bID_i and the chain is already locked, the Master Node issues a "die" response, D, and the camera abandons that request, waiting a short time until requesting again on subsequent frames, until the node can service the request. A key indicator of camera contention on the system during a given time period is the number of D events, as well as the total number of blocks created. The time period used was 1 minute for both values.

Simulation One added 1 camera per minute until a total of 53 cameras operated concurrently at a block creation rate of 1 block per minute, per camera. The mean of D was 4.51 per minute, with block creation rates at close to optimal for every increment. These results as shown in Figure 2 represent the system operating effectively in a typical deployment scenario of a single agency or small to medium scale regional system.

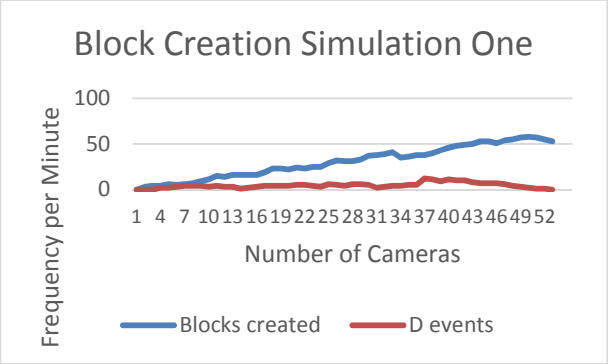

Figure 2 Block creation simulation one trend results showing normal operation of system

Simulation Two ran cameras requesting bID_i every 1 second and added an additional camera every minute until 53 cameras were operating concurrently.

Figure 3 represents the data of Simulation Two, and shows D increasing rapidly as the number of active cameras increases. The number of blocks created in a minute did not reach 300, with peak block creation occurring at 14 cameras before dropping slightly and stabilizing. The algorithm is tolerant to contention, but without a timing mechanism in place allocation of bID_i was very inefficient. Although there may be no real-world requirement for higher block creation rates, it is possible that such rates could be required to support a high number of cameras in large installations. This limit represents a significant scalability problem for systems hosting large number of cameras. Even with a mechanism to coordinate timing, enabling the best-case support for 52 blocks per second, extrapolated to 3,120 cameras each attempting to produce 1 block per minute it is clear the system cannot meet probable scalability requirements for very large networks, even when using a more practical creation rate of one block per minute per camera. This design iteration is therefore unsuitable for very large numbers of cameras utilising a time-based high frequency of block creation.

It is important to note that the block creation limitation does not impact normal operation of the camera or the VMS. Block creation and watermark payload generation occurs independently to frame capture, compression and transmission. Its visible symptom is the introduction of delay when updating the dynamic watermark in real time.

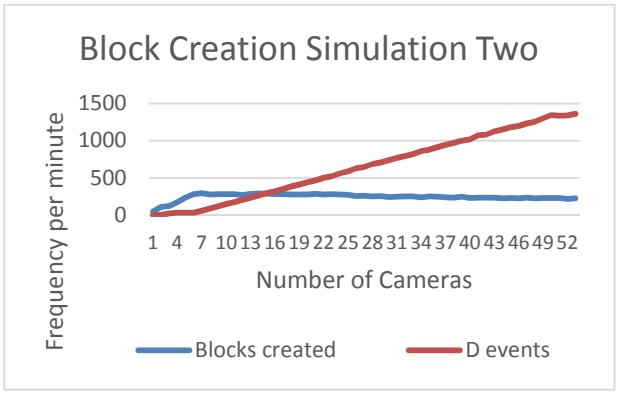

Figure 3. Block creation simulation two trend results showing rapidly increasing contention and severe limitation of block creation rate

3.3. Further work

The system currently meets the functional requirements of the initial use case; There is little evidential value in producing a high volume of time-based blocks on single cameras, therefore the demonstrated scalability limitations will only affect larger installations with many cameras. Systems utilising event triggers that create fewer blocks would likely be unaffected by this limitation. None the less, there exists potential for contention if the system generates blocks rapidly and a clear limitation on the number of blocks that can be generated per minute.

One potential solution to contention could be the introduction of an election process to the Master Nodes, allowing the cameras to always create blocks on demand,

but possibly rejecting them once they are transmitted back if the block ID is no longer available, forcing the camera to retry. Although this would address scalability, it could also negate a key feature if the camera were no longer able to stamp video in real time with the appropriate watermark, thus reintroducing our original symptom.

Another approach could be to remove the requirement for the blockchain to link sequentially, allowing for the immediate allocation of bID_i without locking the chain. Each block could include the address of its own H and link backwards non-sequentially. Such an approach complicates the blockchain but should also significantly increase the number of blocks that can be allocated per minute.

The current method of linking video to the chain via a visible low complexity watermark can also be improved. The visible watermark does not protect itself or the video file from modification. Although video is linked to metadata and that metadata secured, we currently have limited assurances that the video itself is intact. Traditional forensic hashing methods are useless on the camera as the image is about to undergo compression. Practically speaking, video segment files can be hashed once they are at rest on the VMS system, however, this process is only a partial solution, our design goal being the end to end protection of collected data. It follows that a clear area for improvement would be the implementation of a watermark algorithm that could both link the video to the blockchain and protect the video file from modification whilst still being tolerant to compression and minor processing.

4. Conclusion

We have demonstrated here the inherent usefulness of the blockchain concept when applied to a specific real-world law enforcement problem. This implementation demonstrates one approach to the application of blockchain for evidence protection. In terms of functionality our prototype implementation performed well in a production environment and indicated promising audit and reliability properties. Solutions to the identified scaling limitations are compelling topics for further research.

We are confident that blockchain will form a fundamental part of evidential record keeping procedures for law enforcement once the complete lifecycle of the data is addressed. Standards must also be defined that enable the widespread implementation of on-camera block creation and watermark embedding across camera vendors in the industry.

5. References

[1] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, Jan. 1991.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[3] M. Kerr and R. van Schyndel, "Adapting Law Enforcement Frameworks to Address the Ethical Problems of CCTV Product Propagation," *IEEE Security & Privacy*, vol. 12, no. 4, pp. 14–21, 2014.

[4] J. Robertson, "Privacy concerns voiced over photo database link to real-time surveillance | Australia news | The Guardian," *The Guardian*, 2017. [Online]. Available: https://www.theguardian.com/australia-news/2017/oct/06/privacy-concerns-voiced-over-photo-database-link-to-real-time-surveillance. [Accessed: 24-Jun-2018].

[5] J. Evans and C. Sibthorpe, "Feature creep may impose facial recognition in all aspects of our lives, expert warns," 2017. [Online]. Available: http://www.abc.net.au/news/2017-10-05/facial-recognition-coag-privacy-concerns-about-the-capability/9017494. [Accessed: 25-Jun-2018].

[6] A. Davidson, "Increasing trust in criminal evidence with blockchains," *GOV.UK, MOJ Digital & Technology*, 2017. [Online]. Available: https://mojdigital.blog.gov.uk/2017/11/02/increasing-trust-in-criminal-evidence-with-blockchains/. [Accessed: 26-Jun-2018].

[7] B. Gipp, K. Jagrut, and C. Breitinger, "Securing Video Integrity Using Decentralized Trusted Timestamping on the Blockchain," *10th Mediterranean Conference on Information Systems (MCIS)*, vol. 26, no. 2, pp. 3–17, 2016.

[8] "RabbitMQ - Messaging that just works." [Online]. Available: http://www.rabbitmq.com/. [Accessed: 14-Jun-2018].

[9] ISO, "ISO/IEC 18004:2000 - Information technology - Automatic identification and data capture techniques - Bar code symbology - QR Code," *ISO Standards*, vol. 2000, p. 122, 2000.

[10] "Genetec." [Online]. Available: http://www.genetec.com/. [Accessed: 14-Jun-2018].

[11] "Milestone Systems." [Online]. Available: https://www.milestonesys.com/. [Accessed: 14-Jun-2018].

[12] Grzegorz Gogolowicz, "Google Cloud Platform Blog: One click to deploy a RabbitMQ cluster handling over 1 million msg/sec," *Google Cloud Platform Blog*, 2014. [Online]. Available: https://cloudplatform.googleblog.com/2014/06/rabbitmq-on-google-compute-engine.html. [Accessed: 11-Jun-2018].