Assuring the Integrity of Videos from Wireless-based IoT Devices using Blockchain

Dominik Danko*, Suat Mercan†, Mumin Cebe†, and Kemal Akkaya†
*Dept. of Math and Computer Science, Clark University, Worcester, MA 01610
Email: ddanko@clarku.edu
†Dept. of Elec. and Comp. Engineering, Florida International University, Miami, FL 33174
Email: {smercan,mcebe,kakkaya}@fiu.edu

***Abstract*—Blockchain technology has drawn attention from various communities. The underlying consensus mechanism in Blockchain enables a myriad of applications for the integrity assurance of stored data. In this paper, we utilize Blockchain technology to verify the authenticity of a video captured by a streaming IoT device for forensic investigation purposes. The proposed approach computes the hash of video frames before they leave the IoT device and are transferred to a remote base station. To guarantee the transmission, we ensure that this hash is sent through a TCP-based connection. The hash is then stored on multiple nodes on a permissioned blockchain platform. In case the video is modified, the discrepancy will be detected by investigating the previously stored hash on the blockchain and comparing it with the hash of the existing frame in question. In this work, we present the prototype as proof-of-concept with experiment results. The system has been tested on a Raspberry Pi with different quality of videos to evaluate performance. The results show that the concept can be implemented with moderate video resolutions.**

***Index Terms*—Video Integrity; Blockchain; IoT device; hyperledger; digital forensics**

I. Introduction

Video scene is a very important material to interrogate a crime and resolve any dispute because it reveals so much detailed information about the case [1], [2]. The proliferation of IoT devices enables convenient video recording opportunities which can be quickly transferred through the availability of various wireless communication options. This may include drones which are used in many smart city applications as well as other wireless-based cameras deployed on streets/buildings, including those used by police officers in pursuing crimes [3] [4].

However, video forgery techniques are so sophisticated that a video is susceptible to much manipulation, such as being falsified with insertion or deletion of objects. Advanced video editing tools allow users to tamper with videos easily and in visually undetectable ways [5]. In particular, if the video is being transmitted through wireless channels, this may become a more prevelant issue as untrusted communication channels might also be the source for data tampering [6].

Differentiating a tampered video from an authentic one is now harder than tampering with it. In order to use a video as evidence in court, the authenticity must be proven. Therefore, it is important to equip the camera with integrity verification abilities. There exist various methods to achieve this goal in the literature. One way is using a digital watermarking technique, [7] in which an invisible signature is inserted into the video that can be used to check if it has been modified. Another approach is analyzing the content itself to detect distortions in the image [8]. Hashing has also been in use for long time, where a hash is calculated and distributed along with the video. While these approaches may address the issue, there are risks when the data is stored on a server. Data stored on servers may be subject to unauthorized alterations, especially if the process of digital evidence gathering and storage is not followed properly. In addition, if streaming is ongoing, the video frames are transmitted separately, which may require individual attention. Therefore, there is a need to ensure the integrity of video at all levels, from capturing to storage, so that it can be permissible as evidence in courts, even if the authenticity of the video is questioned as being fake or tampered.

Blockchain has emerged as an alternative method to transfer money between non-trusting participants without having a trusted third party such as a bank [9]. The idea of creating a distributed public ledger has led to many other applications and blockchain has exceeded its original purpose; some people consider it a game changer. The main idea is to store any data in a distributed and retroactively unchangeable manner which will ensure the authenticity of data.

In this paper, we apply this promising idea to verify the video recorded by wireless-based IoT devices, which can be used in many scenarios. For example, police officers need to carry a wearable camera on their shoulder or forehead which records incidences, and the video footage they capture can contain evidence that would be useful in court. Similarly, drone videos of crime or accident scenes can contain forensic evidence. Drones are also capable of transmitting real-time video to a base station.

However, videos can be tampered with during transmission or while stored on a server or other media. Thus, it is necessary to verify if a video has been changed. This fact has motivated us to apply blockchain to address the issue. Specifically, the video data gathered from an IoT device needs to be stored on Blockchain immediately as it leaves the IoT device. This will mitigate risk of tampering by eliminating intermediate stages. Since Blockchains consensus mechanism distributes the data to all stakeholders, everyone will have the original copy, and

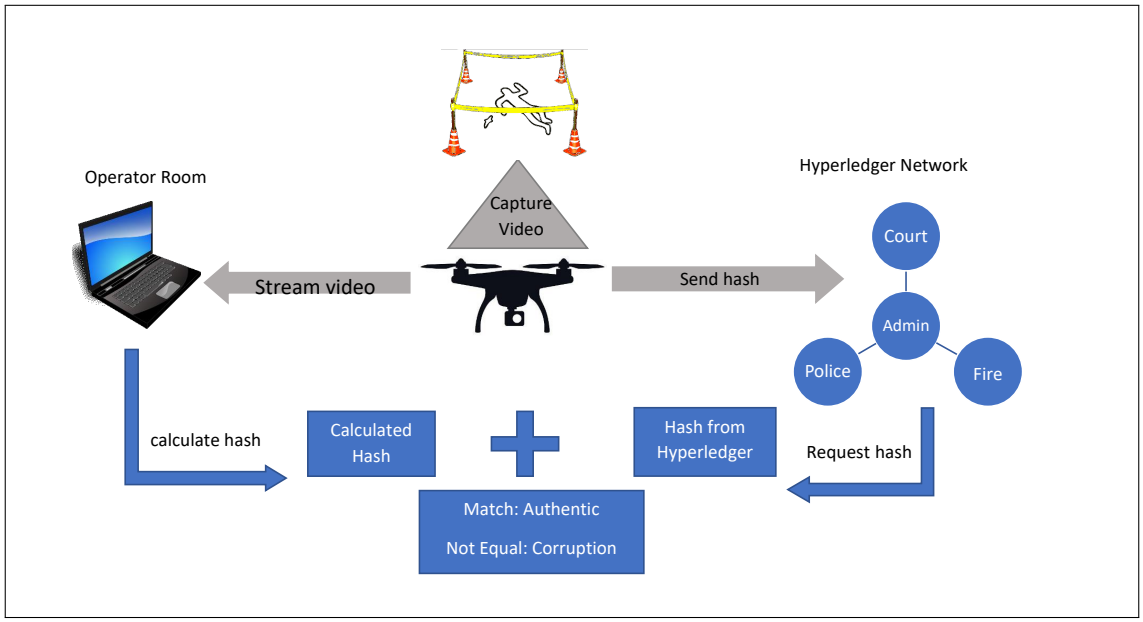


Fig. 1: System Design

thus no one can modify the data by themselves.

Nevertheless, there are challenges in this approach. Storing the whole video on a public blockchain is not free due to transaction fees and blocksize limits. Thus, it is necessary to calculate the hash of each video frame and only write this hash to save space. We propose using a reliable communication protocol to ensure that the hash values will not be lost and thus advocate the use of TCP protocol while the original video data can be sent using an unreliable protocol such as UDP.

This is still not a viable option as there are thousands of frames in videos. Therefore, we opt for a private (i.e., permissioned) Blockchain approach where the distributed ledger is maintained by a group of stakeholders who are permitted to become members of the private group. In such a case, the proof of work for transaction verification will be different, and the costs will be eliminated. Therefore, we propose using IBM's Hyperledger for our purposes that will act as a distributed storage for our frame hash values.

We implemented the proposed mechanism on a Raspberry Pi that has a camera using WiFi connection to a server. The results indicate that the mechanism is feasible and does not interfere with the performance of the real-time streaming.

This paper is organized as follows: In the next section, we summarize the related work in the literature. Section III provides some background on the used concepts while Section IV presents the system model along with our approach. In Section V we assess the performance of the proposed mechanism. Section VI concludes the paper.

II. Related Work

Some recent works in the literature started to investigate the validation of the integrity of video data for different purposes using blockchain. The main work in this context is presented in [13] which tries to protect not only video content but also camera settings such as angle of camera in surveillance systems. The authors try to prevent hackers from changing camera orientation which might either violate the privacy of neighbors or prevent the recording of some criminal scenes. They distinguish the background and foreground images. Background is used to deduce the camera settings by using some features that do not change over time such as corners and edges, while the foreground is used to identify events occurring in the scene. The hash of the video and metadata is then stored on blockchain. This work's main concern is the parameter settings and more importantly it assumes the availability of the whole video to get the hash. Our goal in this work is different as we deal with video streaming where integrity depends on the reliability of each frame as it leaves the IoT device. Hashing needs to be done in real-time, which puts burden on the IoT device.

The other closely related work is reported in [14] where the authors try to provide data assurance for the collected data through IoT sensors. They calculate the hash of the data and store it on the blockchain network instead of the whole data. This work differs from ours as theirs focuses on light text data instead of video data, and they do not deal with wireless communication.

III. Background

Blockchain: Blockchain is a list of records called blocks, first proposed by Satoshi for Bitcoin [9]. These blocks are linked together by containing the hash of the previous block, while containing the data of the current block. The list of blocks continues to grow with the addition of new ones as it is not possible to delete existing blocks. A critical component of blockchain is that all of these blocks, and the data they contain, are distributed among many different nodes. These nodes have to agree on the state of the blockchain, making it nearly impossible to modify any data that has been written to a blockchain. This working scheme of blockchain carries

unique properties such as relieving central authority trust, immutability, and timestamping. These powerful properties are why blockchain is useful and appropriate to use in verifying the authenticity of data and video in our case.

Consensus Mechanism: The process of adding a new block to the chain is carried out via a protocol, which establishes consensus among participants to confirm the new block. In other words, it validates the transactions within the block and provides an agreement on the last state of blockchain. There are two types of blockchain structure, public and permissioned, according to the used consensus mechanism [17]. The most widely-known blockchains, such as Bitcoin and Ethereum fall into public blockchain category where consensus is established via a mechanism called Proof-of-Work (PoW). The PoW consensus is typically a form of hash puzzle which requires finding a predefined hash value. This consensus protocol brings a significant level of security on the chain (withstand up to 50% of nodes are being malicious), but at the cost of computational power and time. For instance, Bitcoin's maximum throughput is 7 transactions per second and the consensus finality can take an hour. On the other hand, permissioned blockchains utilize some kind of Byzantine fault tolerant voting based algorithm as consensus mechanism, such as Practical Byzantine Fault Tolerance (PBFT) [16] or Stellar Consensus Protocol (SCP) [15], which do not require computationally expensive hash puzzles. As a result, reaching a consensus is faster which means higher transaction throughput. However, permissioned blockchains generally require more than two-thirds of nodes to be trustworthy rather than 51%.

Voting based consensus protocol: Voting based consensus protocol first emerged in distributed computing [16] to provide reliability of data or computation, even if arbitrary nodes conduct malicious actions or fail. Permissioned blockchain mechanisms adapt the same idea to establish consensus where some of some nodes may act maliciously. In this setting, where there are n nodes, a consensus can be achieved if at least $(2n-1)/3$ number of nodes act honestly. Honesty means providing correct information to the other participants. In a permissioned blockchain, there are two different types of nodes called *Leader* and *Validator*. First, a randomly selected Leader builds a block from transactions. This block is then distributed by the leader to Valitador nodes for verification. Validators check the transactions within the block, sign it, and distribute it again to the other Validators, as shown in Fig. 2. Each node, again, distributes the block captured from the other node. This continues until each Validator node collects individually signed versions of the block from the other ones. After $n-1$ version of blocks are gathered, the Validators check differences between blocks. If $(2n-1)/3$ of these blocks are valid, the Validator nodes inform the Leader about confirmation and add the block to its local chain.

Hyperledger: Hyperledger [11] is an opensource platform, distrubuted ledger founded by Linux Foundation and supported by over 50 companies including IBM, Intel. It is implementing permissioned blockchain technology by utilizing a voting based consensus protocol called Practical Byzantine Fault Tolerance (PBFT). It is a permissioned blockchain platform where access is restricted to stakeholders unlike the public blockchain where anyone can access the produced blocks. A permissioned blockchain would make more sense for this use case, where only certain entities can access and modify the data stored on the blockchain.

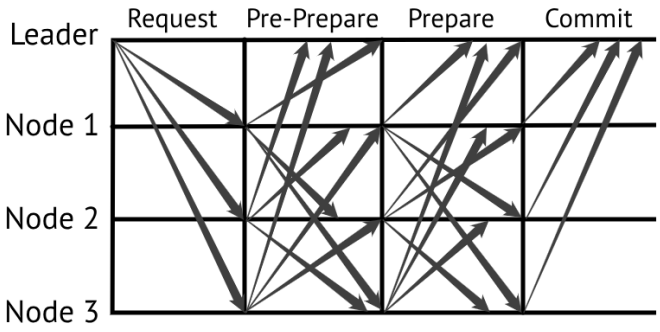


Fig. 2: An illustration of how BA protocol works with replicated nodes.

IV. Proposed Approach

We propose a method that leverages the blockchain concept to improve video integrity and to detect tampered video captured by drones. The idea can be accomplished by storing the video frames on a blockchain, which would make it immutable: no one can change it, thus it can be used as evidence for forensic purposes. However, video frames are too big to put on a blockchain, so we take hashes of video frames and put them on a blockchain. These hashes are immutable and can later be used to verify the video. They are sent from the IoT device directly to the blockchain to prevent any alteration during data transmission. Fig. 1 illustrates the approach that we propose in detail using a drone video communication application. The drone equipped with a camera captures a video frame by frame.

A. Hash Computation

The hash calculations occur on the IoT device itself so that any intervention by adversaries with the video during transmission will be detected. The hash is calculated either for each frame or for a sequence of frames before it is transferred. However, hashing each frame and sending it to blockchain might be costly in terms of computation and time. Thus, we try various optimization methods to reduce the total time and to handle higher quality videos.

To get the hash of a frame, the frame is first converted to a string value and then a hash is calculated on this string. Therefore, conversion from frame to string and hashing the string can take a lot of time, especially if this is done for each and every frame. As this will limit the quality of the videos that can be processed, we propose to perform frame selection by adjusting the number of frames we select for hashing. In this respect, one potential approach is to focus on I-frames which are the frames that cannot be encoded using

popular encoding techniques such as MPEG or H.264. As the number of such frames would be lower, this will reduce the computation time and eliminate the necessity to store each frames' hash. When a video is encoded, it is converted to GOPs (Group of Pictures) which consists of I, B and P frames. Only I-frames are complete images, B and P frames reflect the changes from surrounding I-frames.

B. Communication of the Hashes

Since the hash of a frame is a crucial element to be stored in blockchain, it needs to be transmitted in a reliable manner. Therefore, we propose that the IoT device will open a TCP connection to hyperledger so that any loss value could be re-transmitted through the wireless channel. The video frames are typically sent via UDP as it is better suited for video streaming, although this may cause some of the frames to be lost. Therefore, each frame ID will also be appended to the hash in order to be able to compare it with the hash computed for the frame at the server. Any missing IDs will be discarded on the blockchain.

C. Writing to Blockchain

In a permissioned Blockchain, there need to be members. For this work, we assumed that there may be three peer organizations in the hyperledger network: 1) Related court unit; 2) Police Department; and 3) Local fire departments. Note that the number of participants can be increased depending on the nature of the application if needed, but to enable any proof of work algorithm, it should not be less than three. These members will execute the Hyperledger transaction verification process through voting. If there are fewer than three participants, then it would not make sense to use a permissioned blockchain, a public blockchain should be used instead; however, the transaction costs associated with a public blockchain must be considered.

D. Integrity Verification

Later when it comes to use this video as evidence in court, the same hashing process is repeated for the stored video frames. Using the frame number as an index, each frame can be queried in the hyperledger which will return the original hash computed and stored in hyperledger. If the hashes match, the frame is authentic; if they do not match, it can be inferred that the video is a fake or altered video. The stored hash is secure because it is distributed on all stakeholders and they agree on its correctness. If any stakeholder is compromised, the other nodes will still provide the correct information.

V. Performance Evaluation

A. Experiment Setup

In order to evaluate the performance of the proposed approach, we set up a testbed and performed various tests. We used a Raspberry Pi3 to simulate a drone, or similar IoT device. It ran the code that would be installed on a drone equipped with a camera. This setup is illustrated in Fig 3. To ensure consistency in our tests, we stored prerecorded videos of various resolutions on the Raspberry. We also set up a hyperledger network with three participants. The hyperledger network runs on a laptop and communication is achieved through Wi-Fi. The performance greatly depends on the hardware. Raspberry Pi3 B+ specs shown in Table I are used to run the experiments. Therefore, the values in the results are specific to this configuration. We used OpenCV [18] to process the frames. And we employed MD5 hash function for hashing purposes.

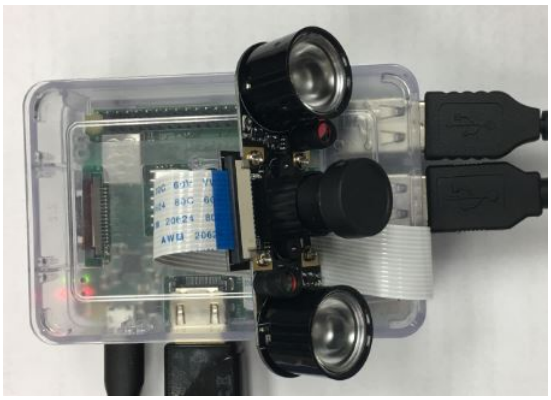


Fig. 3: System Implemenentation

TABLE I: Raspberry Pi 3 specifications

CPU(SoC)	BCM2837B0 quad-core A53 1.4GHz
RAM	1GB LPDDR2 SDRAM
Networking	2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi

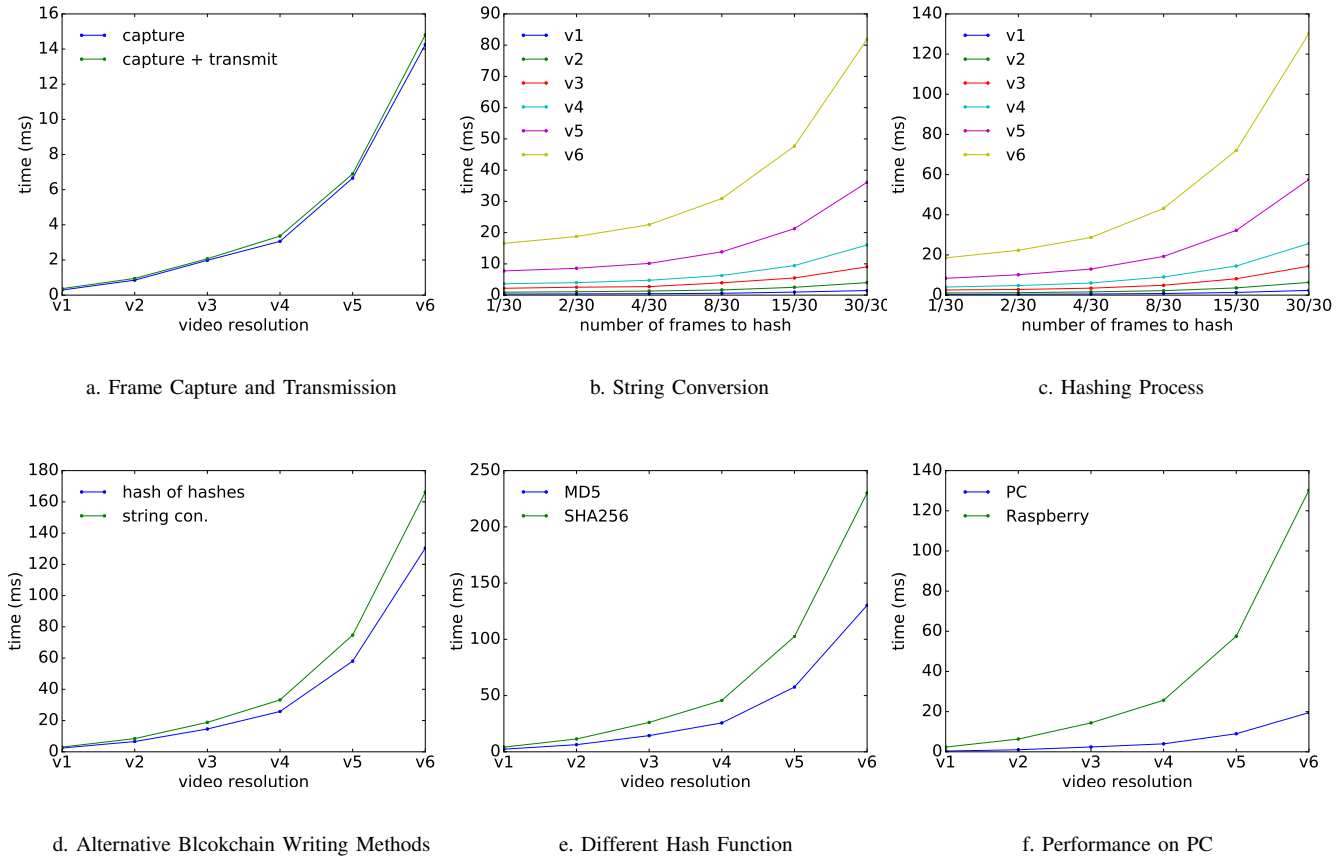
We used 6 different resolutions of the same prerecorded video to be precise in measurements. It is a 10 second video in mp4 format which has 303 frames in total. MPEG is used as the encoding function. The size and resolution of each version is shown in Table II. Each video has different resolutions; thus, they vary in size, which will affect processing and transmission time. Since the Raspberry has limited computational and memory capacity, the frame size will impact the performance significantly.

TABLE II: Video Properties

version	resolution	size
v1	256x134	156 KB
v2	426x224	350 KB
v3	640x338	576 KB
v4	854x450	1.31 MB
v5	1280x674	2.73 MB
v6	1920x1012	5.71 MB

B. Metrics and Baselines

The main metric we are concerned with is the time to process and stream the videos. We considered both the com-



a. Frame Capture and Transmission

b. String Conversion

c. Hashing Process

d. Alternative Blcokchain Writing Methods

e. Different Hash Function

f. Performance on PC

Fig. 4: Experiment Results

putation time on the IoT device and communication time. The other metric is the resolution of the videos used.

As a baseline, we considered different numbers of frames to be hashed. Specifically, frequency of frames ranging from 1/30 (one frame from each second) to 30/30(each frame) are considered. We also compared our approach on Raspberry PI to that on a more powerful machine such as a desktop.

C. Performance Results

First, we assessed the video capturing and transmission time of the Raspberry without having any additional processes running on it. Fig 4.a shows the time needed for a Raspberry to capture and transmit the frames to a server over WiFi. It took 14.25 seconds to capture a 10 second video in the highest quality which means that there will be pauses when we stream it. This indicates that it is not possible to record video in this resolution using a Raspberry even with no other processes slowing it down. A lower resolution video will help the hardware handle the process smoothly. Transmitting the frame from IoT device to a remote station is the next step. The transmission time depends on the quality of the communication channel. Current congestion level, protocol, distance, transmission environment are some factors that can effect the time. In our case, the IoT device connects to the server through a router using WiFi. We tried UDP and TCP connections and did not change other parameters, as that is not the purpose of this study. Both UDP and TCP gave very similar results as shown in Fig. 4.a and this is almost negligible compared to computation overhead for capturing the video. It should be noted that the video is in mp4 format. Another encoding technique might generate a different size for the same quality which will effect the transmission time.

The next thing we analyzed is string conversion and the hashing process. Fig 4.b shows the time for capturing and converting the frames to string for each video. It seems conversion to string is an intensive process which adds a lot to the total time, especially for larger frames. If all frames are processed, the total time for the highest resolution reaches around 80 seconds. Only the lowest 3 resolutions are under 10 seconds. If we consider processing only selected frames instead of each frame, then the task can be completed faster. If only two frames each second are converted, then all the resolutions except the highest one can be handled. The hashing process involves first converting the frame to a string, and then taking the hash of the string.

Table III lists approximate times for conversion and hashing of a single frame based on the frame size. It takes almost half a second to convert and hash a video frame in 1920x1012. The results indicate that hashing takes less time once string conversion is done. Fig 4.c shows the total time needed for all

TABLE III: Processing Times

resolution	conversion(ms)	hashing(ms)
256x134	3.8	2.9
426x224	10.3	7.7
640x338	23.3	17.6
854x450	42.9	31.7
1280x674	97.1	70.8
1920x1012	223.2	159.6

the tasks together. Low resolution videos (v1=256x134 and v2=426x224) are handled within the time limit even when all the frames are processed, but higher resolutions require frame selection which lets us go all the way up to v5 without exceeding the time limit. If all frames are needed to check integrity, we can achieve it only for for v1 and v2.

Recall that there was another concern to to take into consideration. Storing hash of every frame in hyperledger might be costly and unattractive even though we are able to do so. Hence, we tested two other options: 1) appending strings from each frame so that we will have only one string to be hashed for every 30 frames; 2) appending the hash of each frame to be hashed and stored in blockchain. That means, we connect to the blockchain once for every 30 frames. Basically here, we take all the frames into consideration in two different ways, but we only need to write to blockchain occasionally. We tested these two options to see the performance, as is shown in Fig 4.d. We can achieve both options for v1 and v2 under 10 seconds while the second option performs a little better because the string does not get too big.

We also considered using different hash functions. The experiments up to now have been performed using MD5. We wanted to test another one to see if it makes a difference. Thus, we employed SHA256 instead of MD5. The results Fig 4.e shows the total time using two different hashing functions while processing each frame. MD5 outperforms SHA256 in terms of speed. The difference indicates that developing lightweight specific hashing algorithms for video enables higher quality videos to be transmitted.

The last thing we wanted to investigate is how the whole process would run on a regular computer, which has more powerful resources to improve performance. We ran the program on a machine with i5 quadcore CPU and 6 GB RAM. Fig 4.f shows the comparison for the highest resolution, v6, with every frame processed. The performance on PC, especially for higher resolutions, is much better than on Raspberry. If the idea is applied on higher capacity machines for any surveillance purposes, it can allow for higher quality recording.

VI. Conclusion

In this paper, we proposed and implemented a system to verify the integrity of a video captured by wireless IoT devices by incorporating blockchain. The idea was for a video that is being streamed, to generate the hash values for individual frames and write those hashes in a distributed ledge, before the frames are transmitted. We implemented the system in a real setup with Hyperledger as the blockchain technology. The experiment results indicate that the idea is promising and usable as long as the right video resolution is picked. Further investigation is needed to handle higher resolution videos.

Acknowledgment

Dominik Danko in this work is supported by US National Science Foundation under the grant number CNS-REU-1757761. This work is also supported in part by #NPRP9-257-1-056 grant from the Qatar National Research Fund.

References

[1] M. Kim and M. Rick, Expert: Digital evidence just as important as DNA in solving crimes, 2008.
[2] R. Poisel and S. Tjoa, Forensics Investigations of Multimedia Data: A Review of the State-of-the-Art, in Proc. of 6th IEEE International Conference on IT Security Incident Management and IT Forensics (IMF), pp.48-61, May 10-12, 2011.
[3] Vattapparamban, Edwin and Güvenç, İsmail and Yurekli, Ali İ and Akkaya, Kemal and Uluağaç, Selçuk, "Drones for smart cities: Issues in cybersecurity, privacy, and public safety", 2016 International Wireless Communications and Mobile Computing Conference (IWCMC).
[4] Menouar, Hamid, Ismail Guvenc, Kemal Akkaya, A. Selcuk Uluagac, Abdullah Kadri, and Adem Tuncer. "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges." IEEE Communications Magazine 55, no. 3 (2017): 22-28.
[5] O. I. Al-Sanjary and G. Sulong. "Detection of video forgery: A review of literature." Journal of Theoretical Applied Information Technology 74, no. 2 (2015).
[6] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability, in International Symposium on Cluster, Cloud and Grid Computing. IEEE/ACM, 2017.
[7] I. Echizen, S. Singh, T. Yamada, K. Tanimoto, S. Tezuka, and B. Huet. "Integrity verification system for video content by using digital watermarking." International Conference on Service Systems and Service Management, 2006.
[8] J. Wang, G. Liu, Z. Zhang, Z. Wang and Y. Dai, Detection of forgery in digital video based on pattern noise, Journal of Southeast University (Natural Science Edition), no. S2, 2008.
[9] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system." (2008).
[10] A. Baliga "Understanding blockchain consensus models." In Persistent. 2017.
[11] C. Cachin, Architecture of the hyperledger blockchain fabric, in Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.
[12] B. Gipp, J. Kosti, and C. Breitinger. "Securing Video Integrity Using Decentralized Trusted Timestamping on the Bitcoin Blockchain." In MCIS, p. 51. 2016.
[13] P. Gallo, S. Pongnumkul, and U. Q. Nguyen. "BlockSee: Blockchain for IoT video surveillance in smart cities." In 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/ICPS Europe), pp. 1-6. IEEE, 2018.
[14] X. Liang, J. Zhao, S. Shetty, and D. Li. "Towards data assurance and resilience in iot using blockchain." In MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM), pp. 261-266. IEEE, 2017.
[15] Mazieres, David. "The stellar consensus protocol: A federated model for internet-level consensus." Stellar Development Foundation (2015): 32.
[16] Castro, Miguel, and Barbara Liskov. "Practical Byzantine fault tolerance and proactive recovery." ACM Transactions on Computer Systems (TOCS) 20.4 (2002): 398-461.
[17] Okada, Hitoshi, Shigeichiro Yamasaki, and Vanessa Bracamonte. "Proposed classification of blockchains based on authority and incentive dimensions." 2017 19th international conference on advanced communication technology (icact). IEEE, 2017.
[18] Bradski, G. and Kaehler, A., 2008. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.".