information

Article

Advancing Video Data Privacy Preservation in IoT Networks through Video Blockchain †

Kasun Moolikagedara *, Minh Nguyen, Weiqi Yan and Xuejun Li *

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; minh.nguyen@aut.ac.nz (M.N.); weiqi.yan@aut.ac.nz (W.Y.)

* Correspondence: kasun.moolikagedara@aut.ac.nz (K.M.); xuejun.li@aut.ac.nz (X.L.)

† This article is a revised and extended version of a paper entitled Which was presented at 5th International Congress on Blockchain and Applications, BLOCKCHAIN 2023, Guimarães, Portugal, 12–14 July 2023; pp, 42–51.

Abstract: In the digital age, where the Internet of Things (IoT) permeates every facet of our lives, the safeguarding of data privacy, especially video data, emerges as a paramount concern. The ubiquity of IoT devices, capable of capturing and disseminating vast quantities of video data, introduces unprecedented challenges in ensuring the privacy and security of such information. This article explores the crucial intersection of video data privacy and blockchain technology within IoT networks. It aims to uncover and articulate the unique challenges video data encounter in the IoT ecosystem, such as susceptibility to unauthorized access and the difficulty in ensuring data integrity and confidentiality. By conducting a thorough literature review, this study not only illuminates the intricate privacy challenges inherent in IoT environments but also showcases the immutable, decentralized nature of blockchain as a potent solution. We systematically explore how blockchain-based methods can be pragmatically implemented to fortify video data privacy, scrutinizing the efficacy of these approaches in the IoT context. Through critical assessment, the paper delineates the strengths and limitations of video blockchain solutions, underscoring the transformative potential of blockchain technology as a cornerstone for enhancing data privacy in IoT networks. Conclusively, this work advocates for blockchain as an indispensable tool in the advancement of data privacy measures for video content, thereby reinforcing trust and security in the increasingly connected fabric of our digital world. As IoT applications burgeon, the fusion of blockchain technology with IoT infrastructures promises a robust framework for protecting sensitive video data, heralding a future of enhanced trust and security in our interconnected ecosystem.

Keywords: video blockchain; IoT network; video data privacy

Citation: Moolikagedara, K.; Nguyen, M.; Yan, W.; Li, X. Advancing Video Data Privacy Preservation in IoT Networks through Video Blockchain. *Information* **2024**, *15*, 171. https://doi.org/10.3390/info15030171

Academic Editor: Nelly Leligou

Received: 9 February 2024
Revised: 6 March 2024
Accepted: 11 March 2024
Published: 21 March 2024

1. Introduction

In the contemporary era, devices connected through the Internet of Things (IoT) have become indispensable to our daily lives, simplifying tasks across a vast array of applications. Despite their convenience, the surge in IoT device deployment has ushered in a new era of challenges, particularly in data privacy. The intimate integration of our personal data with these interconnected technologies has heightened concerns about privacy breaches, especially through the unauthorized sharing of personal data with third parties [1]. Among the various types of data captured by IoT devices, video data are notably high-risk due to their visual nature, which presents unique challenges and vulnerabilities. These concerns are not only about privacy but extend to legal and ethical dimensions, underscoring the need for robust security measures and adherence to legal frameworks [2].

The risk of sensitive information exposure is a paramount concern, as video data often contain highly personal and identifiable information. This raises not only privacy issues but also legal and ethical considerations. Ensuring the security and privacy of video data

in IoT networks requires adherence to legal frameworks and ethical guidelines, which vary across regions and contexts.

Existing schemes often fall short of addressing these challenges effectively, particularly in the context of video data. Many traditional approaches lack the capability to handle the high volume and sensitivity of video data and do not fully leverage the decentralized and immutable nature of blockchain technology.

Our research introduces a novel approach to the integration of video blockchain technology with IoT networks, specifically tailored to address the unique challenges of video data privacy. Unlike conventional schemes, our method emphasizes the importance of selecting the best cryptographic function and data structure for video blockchain, ensuring optimal performance and security. This innovative integration not only enhances the security and privacy of video data but also leverages the inherent advantages of blockchain technology, such as decentralization and immutability, to provide a more robust and scalable solution for IoT networks.

The burgeoning field of video blockchain, originating from our preceding research endeavors, offers a promising avenue for enhancing the privacy of video data within IoT networks [3–6]. We embark on this exploration of data privacy in IoT networks. Our approach introduces a novel integration of video blockchain technology with IoT networks, specifically tailored to address the unique challenges of video data privacy. Unlike conventional schemes, our method emphasizes the importance of selecting the best cryptographic function and data structure for video blockchain, ensuring optimal performance and security [7].

As we embark on this exploration of data privacy in IoT networks, we introduce three central hypotheses that guide our research:

Hypothesis 1. *Measuring best cryptographic function for video Blockchain to enhance data privacy in IoT network.*

Hypothesis 2. *Selecting the best data structure for video blockchain.*

Hypothesis 3. *The integration of video blockchain technology into IoT networks.*

This article aims to explore and tackle the presented hypotheses through the application of suitable research methodologies. To contextualize our research, we will review the current state of the field, highlighting the challenges and controversies related to IoT data privacy. Additionally, we will provide an overview of key publications that have contributed to the discourse on data privacy within IoT networks. By examining the current landscape, we will lay the foundation for the hypotheses and experimental analysis presented in this paper.

In assumption, this article underscores the compelling importance of blockchain as a catalyst for enhancing data privacy, particularly in the context of video data, within the intricate web of IoT networks. As IoT applications continue to rapidly proliferate, the integration of blockchain technology offers a promising solution to address the critical issue of data privacy, ultimately fostering trust and security in our interconnected world. We provide insights that transcend disciplinary boundaries, making our findings comprehensible to scientists and researchers outside the specific field of IoT and data privacy. The structure of this paper is laid out in the following manner: Section 2 delves into the materials and methods utilized in this study. Section 3 presents the findings of the research. The paper is concluded in Section 4.

2. Materials and Methods

In this section, we outline our methods for testing hypotheses to obtain pertinent results. We aim to extract feasible solutions and conduct a comparative analysis of the methodologies we have employed. Our goal is to achieve the desired outcomes and implement a secure IoT network capable of transmitting video data securely.

2.1. Evaluating the Performance of Existing Hashing Functions

In the process of testing Hypothesis 1, it is crucial to identify the most suitable cryptographic functions to establish a secure and efficient video blockchain system. To achieve this, we must refer to a comprehensive literature review of hashing functions and select the most commonly used hashing functions for blockchain implementations. In Section 2, we present an in-depth background study of the current research in this field, focusing on identifying the most robust hashing functions [4].

The criteria used for evaluating the performance of hashing functions include their security properties, such as resistance to collision and preimage attacks, computational efficiency, and compatibility with blockchain technology. We assess the security properties by examining the cryptographic strength of the functions, ensuring they meet the required standards for data integrity and confidentiality. Computational efficiency is evaluated based on the time taken to perform hashing operations, which is crucial for the real-time processing of video data in IoT networks. Compatibility with blockchain technology is determined by the ability of the hashing functions to integrate seamlessly with existing blockchain infrastructures, facilitating interoperability and scalability.

Hence, as a result of background literature studies, we figure out the hashing functions, SHA-256 and Merkle Tree, as possible functions to support achieving the required outcome [8]. The hashing function has been used in specifically tailored applications for blockchain [9]. We prioritize efficiency as a crucial aspect, requiring the enhancement of the function's performance to be compatible with the distinct hardware and software framework of blockchain implementations. Additionally, it is vital to ensure compatibility to facilitate seamless integration and interoperability within the current blockchain infrastructure [10].

The performance of both hashing functions, SHA-256 and SHA-3, is assessed through benchmarking. A script is employed to determine the average time required to execute each hashing operation on a designated sample data string. To ensure precision, the script iterates through the hashing process 1000 times, providing a more accurate measurement of performance. The average time for each hashing function is calculated using Equation (1), where N is the total number of iterations, and $t_{end,i}$ and $t_{start,i}$ are the end and start times for the ith iteration, respectively. This process is outlined in Equation (1), where we detail the steps for measuring the performance of our selected hashing functions and compare their efficiency.

Given:

- N: the total number of interactions
- $t_{end,i}$ and $t_{start,i}$: Start and end times for the ith iteration, respectively.

The average time for the hashing function, $T_{average}$, can be expressed as follows:

$$T_{average} = \frac{1}{N} \sum_{i=1}^{N} (t_{end,i} - t_{start,i}) \quad (1)$$

Additionally, to visualize the distribution of execution times, the individual times are converted into a histogram. The process involves grouping the execution times into bins and counting the number of times that fall within each bin. First, the execution time for each iteration is calculated: $t_{exec,i} = t_{end,i} - t_{start,i}$. The execution times are grouped into bins: B_1, B_2, and B_k. Then, the frequency of the execution times are counted in each bin, $f_j = count\{t_{exec,i} \in B_j\}$ for $j = 1, 2, \ldots, k$. Finally, the histogram is represented by the set of frequencies $\{f_1, f_2, \ldots, f_k\}$ corresponding to each bin. The results of this analysis are presented in Figures 1 and 2, which display the performance of SHA-256 and SHA-3, respectively. The comparison between these hashing functions ensures that the selection made from the literature review is validated and supported by empirical evidence.

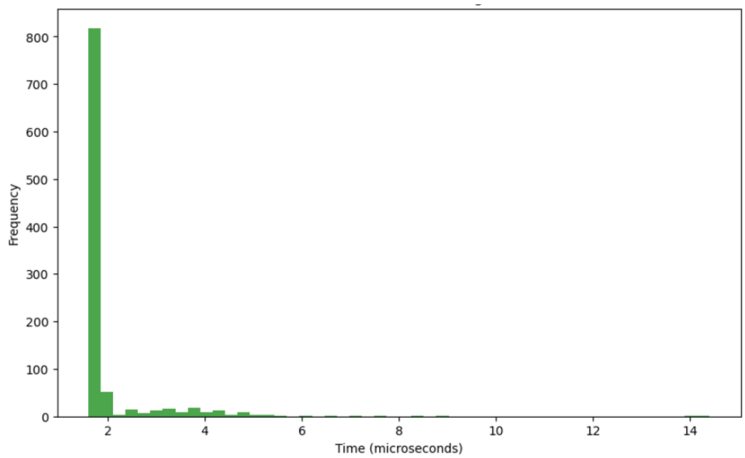

Figure 1. SHA-3 analysis.

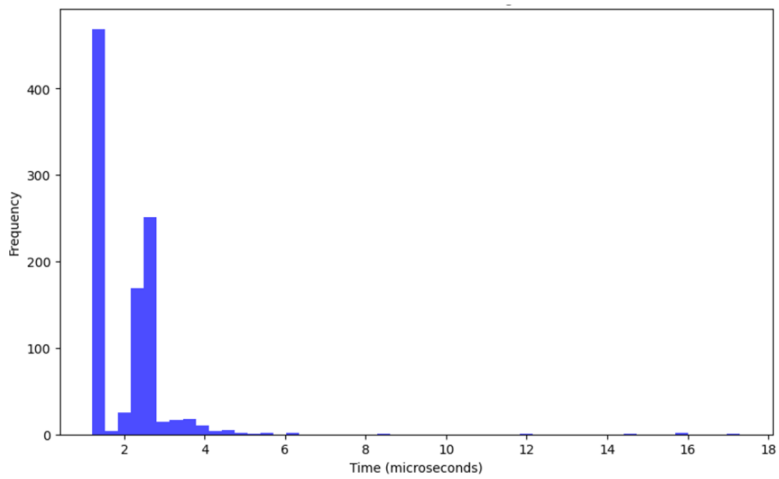

Figure 2. SHA-256 analysis.

After analysis of the performance, one of the best hashing functions needs to be chosen by comparing similar hashing functions. To conduct this hash function assessment, the Multi-Criteria Decision Making (*MCDM*) [11] is used as shown:

$$MCDM = \sum_{i=1}^{m} \frac{R_i}{M} \tag{2}$$

where matric R_i represents the rank on the i-th matric, and then M is the number of evaluation criteria. The main reason for applying this *MCDM* to select the best hashing function and effectiveness is based on the final results. In Figure 3, the *MCDM* result is shown for SHA256 and SHA3. The *MCDM* process applies different methods for different approaches. In our process, we use the Analytic Hierarchy Process (AHP) to compare SHA256 and SHA3. In this process, we create the pairwise comparison by using Equation (2), and the weights of the matrix were derived from our results. Finally, we calculate the consistency by using the Consistency Index (CI) and the Consistency Ratio (CR) [12].

The corresponding results of using Equation (1) are shown in Figures 1 and 2. These results employ the pairwise comparison matrix to show the scales in each comparison by measuring the average time.

Table 1 presents a pairwise comparison matrix that quantifies the relative performance of the hashing functions SHA-256 and SHA-3. The values in the matrix represent the ratio of the average times taken by the respective hashing functions, providing a clear and concise comparison of their performance.

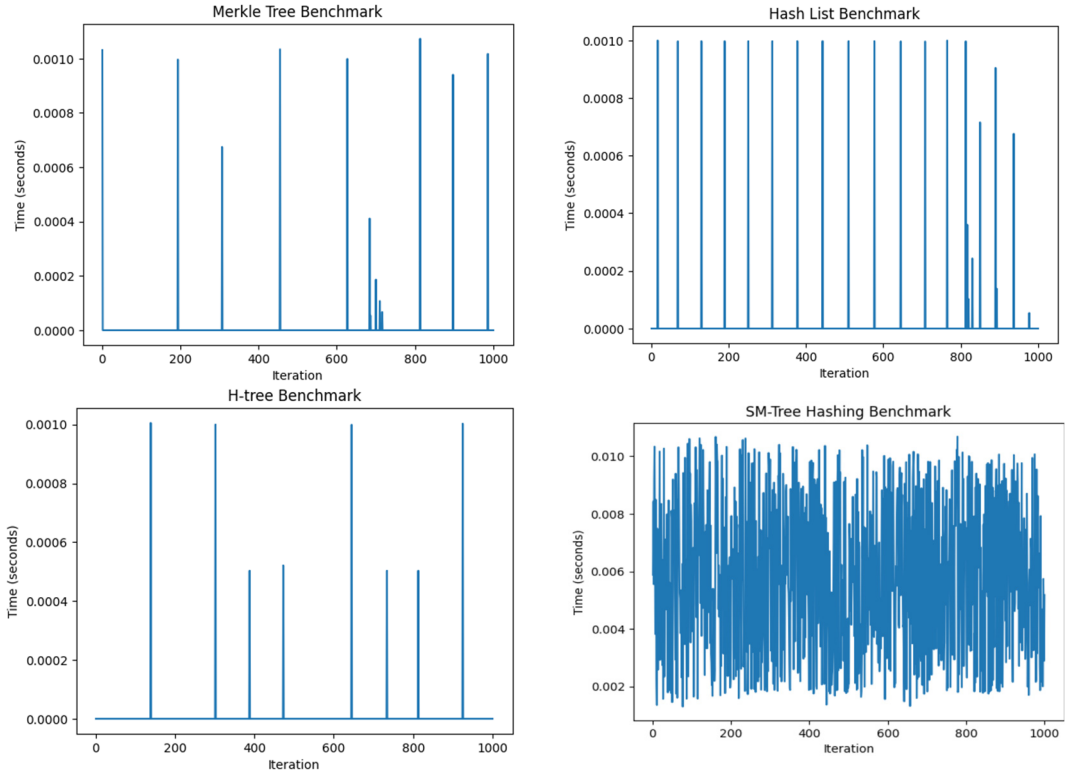

Figure 3. Performance analysis of cryptographic Merkel tree, Hash list H-tree, and SM-Tree (Sparse Merkle tree) data structures.

Table 1. Pairwise comparison matrix.

	SHA256	**SHA3**
SHA256	1	3
SHA3	1/3	1

In the implementation of the Analytic Hierarchy Process (AHP) method, a scale ranging from '1' to '9' was utilized, where '1' denotes equality and '9' signifies an extreme preference for one alternative over another. The second step in the AHP method involves calculating the relative priorities, or weights, of each hash function. This requires Table 1 to normalize the matrix.

Normalization begins with dividing each element by the sum of its column. For our data, the sum of the SHA256 column is 1.33, while the sum of the SHA3 column is 4. The normalized matrix values are then obtained by dividing each element by its column total. For column SHA256, this results in (0.75, 0.75), and for column SHA3, the values are (0.25, 0.25).

Subsequently, the average of each row is calculated to determine the weights. The weight for row SHA256 is (0.75 + 0.75)/2 = 0.75, and for Row SHA3, it is (0.25 + 0.25)/2 = 0.25. Therefore, the weights are 0.75 for SHA256 and 0.25 for SHA3, indicating that SHA256 is considered three times as important as SHA3 according to this specific criterion.

The final step in the AHP method is to check for consistency, ensuring that the judgments are not arbitrary. To compute the Consistency Index (*CI*), one must calculate λ_{max} which involves obtaining the average of row SHA3 and row SHA256 weights. These values should be close to the number of items being compared (n). Therefore, a λ_{max} value of 2.05

suggests a small degree of inconsistency, indicating a generally consistent judgment within the matrix. The Consistency Index (*CI*) is calculated as follows:

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{3}$$

After obtaining *CI*, it can be used to obtain the Random Index (*RI*). This is a value derived from randomly generated reciprocal matrices. The RI depends on the number of elements (n): *RI* is 0.00 for $n = 2$. It is used to obtain the Consistency Ratio (*CR*) as follows:

$$CR = \frac{CI}{RI} \tag{4}$$

If $CI = 0.05$, then $CR = 0.00$ as the final outcome. However, since RI is zero for $n = 2$, the CR cannot be meaningfully calculated. In practice, for a 2×2 matrix, if the CI is very small, the matrix is generally considered consistent. In cases with more criteria ($n > 2$), CR becomes a more meaningful measure because it obtained the result of 0.05 by ensuring consistency.

After following the steps of the AHP method to compare SHA256 and SHA3, the weights are calculated as 0.75 and 0.25, respectively. These weights were derived from the pairwise comparison matrix and subsequent normalization and averaging steps in the AHP method. The weight of 0.75 for SHA256 is significantly higher than the weight of 0.25 for SHA3. This difference indicates that SHA 256 is three times more suitable for our research work base. Furthermore, the constituency was verified by CI which turned out to be 0.05; since the RI for a matrix of size 2×2 is 0.00 and the CI is small, the pairwise comparisons are deemed consistent, even though the Consistency Ratio (CR) could not be meaningfully calculated for a matrix of this size.

To determine the relative effectiveness of SHA256 and SHA3 by using the Analytic Hierarchy Process (AHP), we first constructed a pairwise comparison matrix based on a chosen criterion, such as computational speed. In this matrix, the two hash functions were compared against each other, assigning a value to indicate how much more one is preferred over the other in terms of speed. SHA256 is considered moderately faster than SHA3, so it is assigned a value of 3, whereas SHA3 would conversely have a value of 1/3 compared to SHA256. Next, we normalized this matrix by dividing each element by its column total, effectively scaling the comparisons. The average of the values in each row of this normalized matrix was then calculated, resulting in a priority vector. This vector provided the weights for each hash function, reflecting their relative importance or preference based on speed. Finally, the average for SHA256 was higher than that for SHA3, which indicated a preference for SHA256 in terms of computational speed according to the AHP analysis. This method provided a systematic and quantitative approach to evaluating and comparing the two hash functions based on the selected criterion.

2.2. Evaluate the Best Data Structure for Video Blockchain

In this article, we carefully choose data structures and utilize their synergies to develop a strong framework for a blockchain-based computational approach. A fundamental objective in the development of blockchain applications is to uphold the integrity and confidentiality of the data. If the implementation can achieve the confidentiality of the data, it is possible to achieve the data privacy requirements [13]. Therefore, we explore a range of data structures, including the Merkel tree [14], Hash list [15], H-tree [16], and SM-Tree (Sparse Merkle tree) [17] approaches. After a comprehensive evaluation and comparison of these technologies, we identify the most suitable data structure that aligns with our desired level of security.

To perform a comparative analysis using Merkle tree performance and other similar three data structures, in order to ensure a fair and objective evaluation, we utilize the same equation employed in Equation (1) to measure the computational efficiency of each approach. This performance assessment enables us to not only select the most secure

method but also the one that offers the best balance between security and computational speed. This multi-faceted approach ensures that the blockchain-based computational solution we propose in this paper is both robust in its security features and efficient in its operations. As the landscape of blockchain continues to evolve, it is imperative to strike the right balance between security and performance, and our research aim is to achieve it precisely.

The corresponding results of using Equation (1) are shown in Figures 1 and 2; it is very hard to discern the differences from Figures 3 and 4. Therefore, we have employed the pairwise comparison matrix to show the scales for each comparison by measuring the average time. In addition, to obtain a more effective answer for Hypothesis 2 mentioned in Section 1, we have used Equation (2) for the MCDM method to find out the most suitable cryptographic data structures for our implementation within the IoT network.

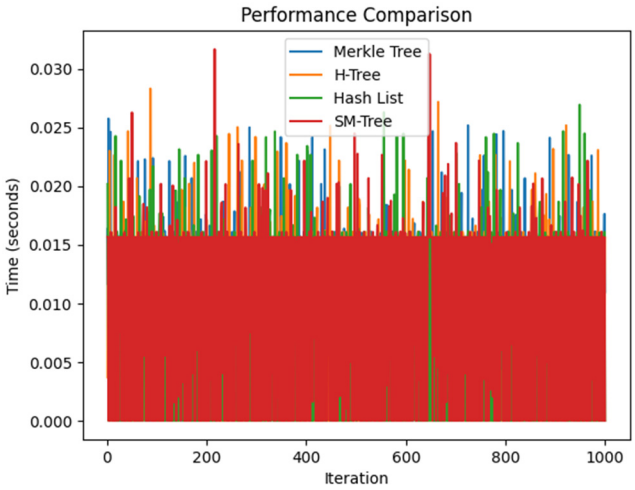

Figure 4. Performance comparison cryptographic data structures.

In this article, we employed the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method to assess and contrast different cryptographic data structures, including Merkle tree, Hash List, H-tree, and SM-Tree as show in the Table 2. The primary criterion for comparison was the performance, measured in execution time (seconds). We began by normalizing the performance data to ensure a fair comparison across different structures. Following this, we assigned equal weights to the performance criterion, acknowledging its crucial role in cryptographic operations. Subsequently, we calculated the ideal best and worst solutions to establish reference points for each criterion

Table 2. (TOPSIS) method to evaluate and compare various cryptographic data structures.

Data Structure	Performance (s)	Norm. Performance	Weighted Norm. Perf.	Sep. from Best	Sep. from Worst	Relative Closeness	Rank
Merkle tree	0.001500	0.402329	0.402329	0.159135	0.025899	0.139969	1
Hash List	0.001800	0.482795	0.482795	0.229808	0.006475	0.027403	2
H-tree	0.002000	0.536439	0.536439	0.284117	0.000719	0.002526	3
SM-Tree	0.002100	0.563261	0.563261	0.313430	0.000000	0.000000	4

The separation measures for each cryptographic structure from these ideal solutions were then determined, leading to the computation of the relative closeness to the ideal solution for each structure. Based on these calculations, the Merkle tree emerged as the most efficient structure, demonstrating the lowest execution time and hence the highest relative closeness to the ideal solution. It was followed by the Hash List, H-tree, and SM-Tree, in descending order of efficiency [18]. This ranking provides valuable insights into the

comparative performance of these cryptographic structures, highlighting the Merkle tree's superiority in terms of speed and efficiency in our analysis.

2.3. The Integration of Video Blockchain Technology into IoT Networks

In this section, we introduce a novel approach aimed at strengthening data integrity within smart cities [19]. Hence, in Sections 2.1 and 2.2, we have tested Hypothesis 1 and Hypothesis 2 to establish a strong base for establishing its connection with the IoT network. Our method seamlessly integrates SHA 256 hashing functions and Merkle tree cryptographic data structures. Also, in this section, we test the third hypothesis before launching our implementation. Therefore, a block matrix is used for data storage in IoT networks to ensure the utmost security and privacy of surveillance data [20–23].

The core of our methodology lies in the verification process, a meticulous procedure designed to identify alterations in image frame sequences and pinpoint specific image modifications. To accomplish this, we generate a dedicated Merkle tree for each data block, securely storing its root hash within the blockchain. This architectural design acts as a safeguard for data integrity, laying the foundation for a robust and secure blockchain implementation [23–25].

To rigorously test our methodology, we have meticulously curated multiple datasets containing sample videos for system integration. These surveillance videos, conventionally recorded at 25 frames per second [26], have been augmented to 30 frames per second in our study to include more comprehensive content in our experiments. Our research utilizes a dataset consisting of 7000 video frames centered around Auckland city. The primary aim is to create hash values for these video frames to increase their security and resilience against possible threats.

Our article serves as a pivotal link between surveillance video data and blockchain technology. We have established a decentralized repository for storing this critical data, with a primary focus on enhancing security and privacy. This enhancement is primarily driven by the strategic use of cryptographic algorithms for hashing and cryptographic data structure, setting our work apart from previous research. These algorithms play a pivotal role in ensuring the seamless connection of video frames, thereby facilitating the detection and localization of any frame alterations. Moreover, our verification procedure, incorporating Merkle trees and hashing functions, adds another layer of security.

Acknowledging the crucial role of maintaining privacy in blockchain applications, we introduce a solution based on blockchain technology designed to not only secure but also enhance the integrity of surveillance data within smart cities. Our goal is to build greater trust, provide dependable outcomes, and manage data disclosures with precision. By integrating computational techniques with video blockchain technology, we effectively manage data security, restrict unauthorized access, and facilitate detailed monitoring in sectors like law enforcement, insurance, and traffic management systems. Consequently, this aids in the necessary improvements for heightened security and compliance within the smart city video surveillance sector [21,22]. To verify the integrity of frames, we create a Merkle tree using the hash values from the block matrix. By saving the root hash of the Merkle tree in the blockchain, we can identify any modifications through comparisons between block hashes and Merkle tree hashes, thereby incorporating a critical layer of tamper-proofing into our system. Furthermore, we investigate the capabilities of block matrix operations, such as matrix multiplication and inversion, for video processing functions including compression, filtering, and restoration. These processes are carried out on block matrices preserved in the blockchain, facilitating highly efficient and secure video processing tasks [27]. The deployment of the Merkle tree function is crucial to our system. This function processes an array of data to iteratively build a Merkle tree. For the simplest scenario, when there is just a single piece of data, the function directly returns that piece. For more complex scenarios, it forms the left and right subtrees, combines their hashes using the SHA-3 algorithm, and outputs the aggregate hash. This final hash serves as the Merkle tree's root. The composite provides a comprehensive visualization of the

blockchain privacy-secure methodology by integrating SHA-256, Merkle tree, and block matrix processes, as shown in Figure 5. In SHA-256, the blue line depicts the output of the SHA-256 hashing process over time, showcasing how input data blocks are transformed into cryptographic hashes. In Merkle tree, represented by the green line, the evolution of the Merkle root hash as adjacent data chunk hashes are paired and hashed, with the final point indicating the unique identifier for the entire set of data blocks. In Figure 5 Subplot 3 (block matrix), the orange line portrays the compression of data blocks organized in a matrix, highlighting the reduction in data size over time. The overall trends and interdependencies between these processes offer valuable insights into the efficiency of cryptographic hashing, Merkle tree construction, and data compression, contributing to a more secure and private blockchain methodology.

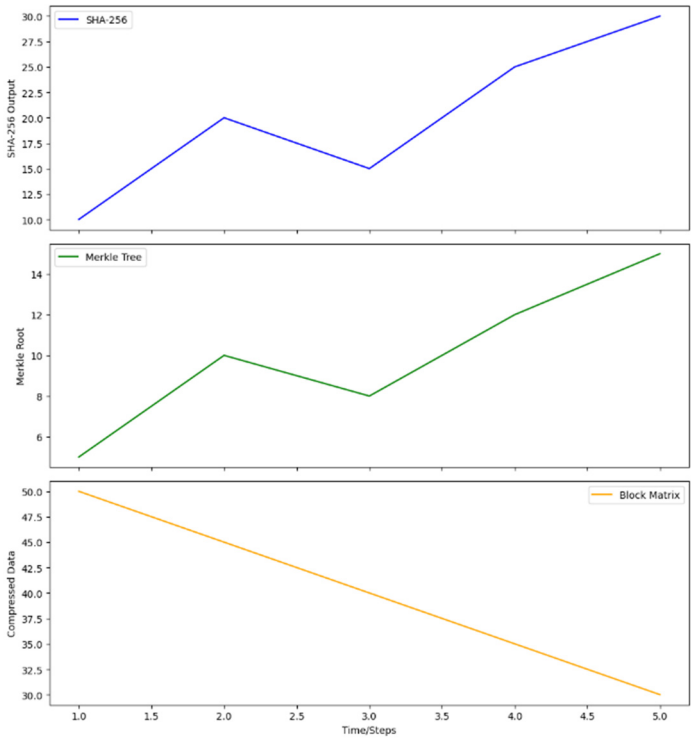

Figure 5. Comparison of SHA-256, Merkle tree, and block matrix.

The results portrayed in the composite plot provide several significant conclusions for the blockchain privacy-secure methodology. Firstly, the SHA-256 hashing process demonstrates the consistent and efficient transformation of input data blocks into cryptographic hashes, indicating the robustness of the chosen hashing algorithm. The evolving trend in the Merkle Tree construction reveals a systematic pairing and hashing of data chunk hashes, culminating in a unique Merkle root hash that serves as a reliable identifier for the entire dataset.

Additionally, the block matrix subplot underscores the effectiveness of compression algorithms, such as JPEG, in reducing the data size of video frames organized in a matrix. The interconnectedness of these processes, as evidenced by the integration points in the plot, signifies a cohesive and secure methodology. The reduction in data size through compression, coupled with the cryptographic integrity provided by SHA-256 and the Merkle Tree, collectively contribute to the enhancement of privacy and security [28–30] within the blockchain framework.

Overall, the results affirm the efficacy of the integrated approach, offering insights into the efficiency of individual components and their collaborative impact on the privacy and security attributes of the blockchain system.

The pivotal element in our system is the implementation of the Merkle Tree function. This function takes an array of data and recursively constructs a Merkle tree (Algorithm 1). In the context of Merkle Tree, denoted as D_i for the i-th data block, and $H(D_i)$ representing the cryptographic hash function applied to D_i, the process involves breaking down the data blocks into consistent fixed-size chunks:

$$D_i = [C_{i1}, C_{i2}, \ldots . C_{in}] \tag{5}$$

Subsequently, the hash for each data chunk is calculated using the cryptographic hash function:

$$H_i = [H(C_{i1}\ ), H(C_{i2}), \ldots, H(C_{in})] \tag{6}$$

Pairs of neighboring data chunk hashes are formed and the hash for each pair is computed:

$$P_{ij} = H_i[j] \# H_i[j+1] \tag{7}$$

This process iterates until only one hash remains:

$$P_i = P_{i-1,1} \# P_{i-1,2} \# ... \# P_{i-1,n/2} \tag{8}$$

The iteration continues until P_i becomes the Merkle root hash: $MerkleRootHash = P_{final}$.

Therefore, we extracted details on how the blockchain data structure operates in conjunction with the Merkle tree. In the future, we will further examine the video frame securing method of the block matrix algorithm and explore how it works with the Merkle tree to achieve our desired outcomes.

Algorithm 1: Block Matrix

Input: A video file consisting of frames.
(1) Divide each frame into uniform blocks of 16 × 16 pixels.
(2) Arrange the blocks from each frame into a matrix format, with rows indicating blocks and columns denoting frames.
(3) Apply compression techniques (e.g., JPEG) to each block for data size reduction.
(4) Store the resulting compressed block matrix in a binary format.
(5) To retrieve a specific frame, load the compressed matrix and select the appropriate column of blocks.
(6) To access a particular block within a frame, locate and decompress the corresponding row from the matrix.

The block matrix algorithm involves several key steps. Initially, each frame of a video file is divided into fixed-size blocks (16 × 16 pixels), creating a frame sequence:

$$F_i = [B_{i1},\ B_{i2}, \ldots, B_{in}] \tag{9}$$

These blocks are then organized into a matrix, M_i, where each row represents a block of the i-th frame. Compression algorithms, such as JPEG, are applied individually to each block, resulting in a compressed form C_{ij} for the j-th block in the i-th frame. The compressed block matrix is subsequently saved as a binary file using an appropriate method. While accessing a specific frame, the binary file is loaded, and the corresponding column of blocks is extracted. To access a particular block within a frame, the relevant row is retrieved from the block matrix, and the block is decompressed using the algorithm's specified decompression function. These steps collectively define the block matrix algorithm, facilitating efficient storage and retrieval of video data.

Algorithm 1 accepts an array of data along with a specified block size to create a matrix, where each matrix row corresponds to a data block. This process involves traversing the data array, dividing it into segments of the designated block size, and assigning each segment to the correct row within the matrix. Should the total length of the data array

not be divided evenly by the block size, padding is added to the matrix's final row to compensate for the shortfall.

This Figure 6 presents a comparative analysis of the performance of the SHA-256 and SHA-3 hashing functions. The graph illustrates the average time taken by each hashing function to process a set of sample data strings. The vertical axis represents the average time in milliseconds, while the horizontal axis indicates the number of iterations. The comparison highlights the efficiency of each hashing function, providing insights into their suitability for integration into the video blockchain system.

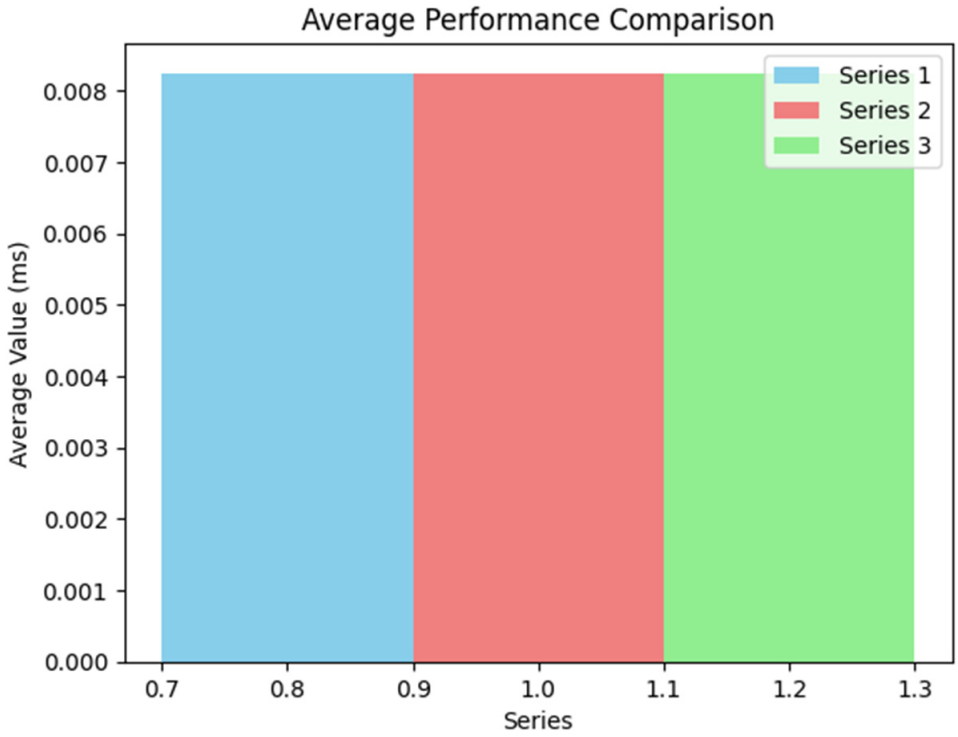

Figure 6. Average computational time (in milliseconds) for Merkle tree-based authentication according to data size.

These algorithms were utilized to securely and efficiently store data from video frames. By dividing the video frames into blocks and building a Merkle tree across these blocks, the system ensures data integrity and authentication. This approach facilitates distributed storage, enabling the secure transfer of data from surveillance systems. The throughput (*TP*) is defined mathematically as follows:

$$TP = \frac{T}{\Delta t} \tag{10}$$

where *TP* is the throughput, measured in transactions (or blocks) per second (TPS).

Figure 7 depicts the efficiency of the SHA-256 and SHA-3 hashing functions in terms of computational resource utilization. The graph showcases the relationship between the size of the input data and the time required for hashing. The results indicate how each hashing function scales with increasing data size, which is critical for evaluating their performance in real-world IoT applications.

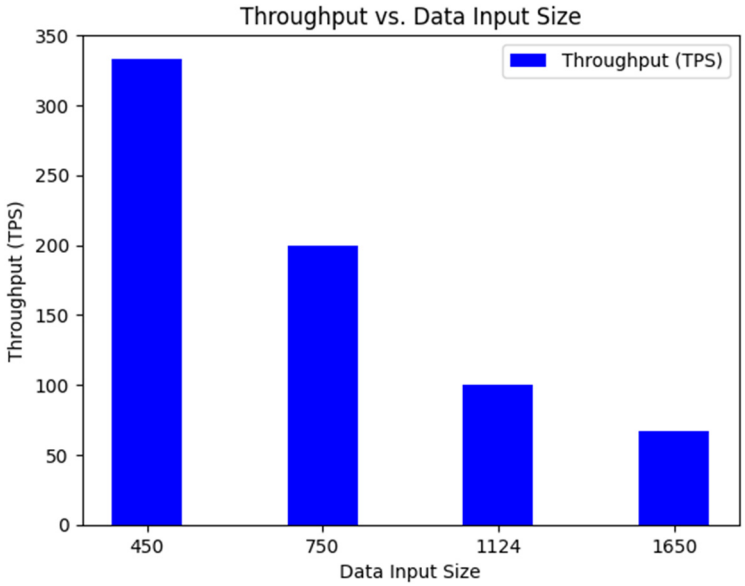

Figure 7. Analysis of the throughput vs. data input size.

3. Results Analysis and Discussion

In this study, we explore the use of video blockchain technology within IoT networks. Our method entails transforming recorded videos into discrete frames, each ranging in size from 50 KB to 1024 KB. These frames were used to create a private blockchain system on a Windows 11 64-bit platform. We set up an experimental framework to evaluate the effectiveness of this innovative technique in defending against various cyber threats. This led to the formulation of new computational strategies for video blockchains, integrating distinct cryptographic algorithms within the video blockchain architecture. This paper makes a novel contribution to the field of video blockchain by proposing a unique strategy for enhancing the privacy of video data within IoT networks. The findings of this research have the potential to lay the groundwork for future studies on video blockchain technology and cryptographic methodologies.

In Figure 8, we analyze the impact of varying data sizes on the execution time. The graph provides a detailed view of the time taken by each hashing function as the input data size increases. This information is vital for understanding the behavior of the hashing functions under different data load conditions and assessing their potential scalability in a blockchain-based IoT network.

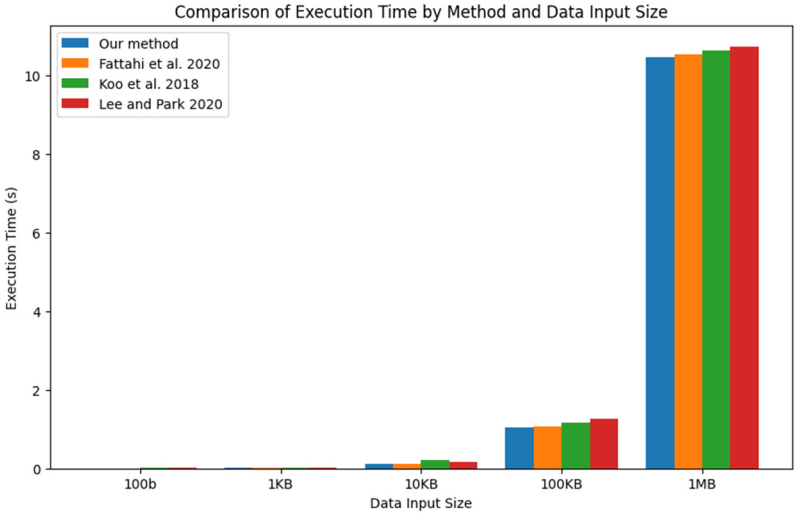

Figure 8. Comparison of computational time between our work and similar studies [31–33].

The analysis of the hashing functions revealed that SHA-256 and the Merkle tree provide robust security features suitable for securing video data in IoT networks. The performance benchmarking showed that SHA-256 offers a balance between security and computational efficiency, making it a viable option for real-time video data processing. The Merkle tree, on the other hand, demonstrated its effectiveness in handling large datasets, which is essential for managing the high volume of video data generated in IoT environments.

The implications of these findings on IoT network security and data privacy are significant. The adoption of SHA-256 and the Merkle tree in video blockchain systems can enhance the integrity and confidentiality of video data, protecting it from unauthorized access and tampering. Furthermore, the efficient performance of these cryptographic functions ensures that the security measures do not impede the real-time processing and transmission of video data, which is crucial for maintaining the functionality and responsiveness of IoT networks.

Overall, our results underscore the importance of selecting appropriate cryptographic functions for video blockchain integration in IoT networks. By ensuring both security and efficiency, we can create a more secure and privacy-preserving IoT ecosystem, where video data are protected without compromising the network's performance.

Each root structure in the Merkle tree validates the sequence and hashing order of video frames, safeguarding against any unauthorized modifications to the sequence unless the entire root structure is altered. We are actively working to integrate a feature for detecting changes in real time into the system, thereby increasing its reliability and strengthening its defense against attacks that compromise privacy and those conducted by quantum computers. The findings from this research provide critical insights into the development of web interfaces for video blockchain systems, establishing a solid foundation for enhancing the dependability and security of such systems in the future.

In this article, we concentrate on a hypothesis based on a testing mechanism to develop methodologies for data structure. In Section 2, the hypotheses were rigorously tested and validated to ensure the robustness of our methodology, ultimately contributing to achieving the objectives of this research work. The evaluation involved measuring the computational time and data size for each experiment, with each iteration repeated 100 times to mitigate errors stemming from outliers. Figure 8 illustrates an ascending trend in computational time concerning the increase in data size for the three Merkle tree-based approaches. However, beyond 100 KB of data, these approaches display only minor differences, as the generation of a Merkle tree for a 1 MB data file constitutes 99.9% of the computational time required by the prover. We graphically depict the results to ascertain time complexity, which is contingent on varying input sizes and block sizes, contributing to the determination of the function's time complexity.

The incorporation of blockchain into IoT networks confronts several challenges, including scalability, interoperability, and regulatory compliance. Addressing these challenges entails scaling blockchain to manage substantial data volumes and transactions, seamlessly integrating it with existing systems, and navigating complex regulatory frameworks. In our forthcoming work, we plan to devise solutions to tackle these issues.

4. Conclusions

Our central objective revolves around establishing a symbiotic relationship between video frames, as captured by the IoT video network, and the blockchain. Our innovative approach lies in the seamless integration of this data into a decentralized storage platform purpose-built for IoT video network security. What sets our work apart from the existing work is its heavy reliance on cryptographic functions, which are instrumental in extracting hash values and signatures from video blockchains. This, in turn, fortifies the security of surveillance data, ensuring its integrity in a tamper-resistant environment.

Notably, the focus of our research work is primarily on enhancing the robustness of data storage within IoT systems rather than centering on the mitigation of potential risks

posed by quantum computer attacks on blockchains. While our current emphasis is on bolstering data security, we acknowledge that the landscape of blockchain technology is evolving. In the future, we intend to delve into the solutions outlined in Section 3 to fortify blockchains against quantum threats.

Privacy concerns remain a significant challenge in blockchain implementation, we acknowledge this aspect as a crucial consideration in our work. However, our vision extends beyond this immediate concern. In the future, there will be a need to address broader challenges such as scalability, interoperability, and regulatory issues that affect the effective deployment of blockchain technology.

The contributions stemming from this paper open new avenues for necessary advancements in the realm of heightened security and adaptability for IoT video network implementations in the dynamic landscape of smart urban environments. Our work aligns with the evolving needs of these cities, where IoT video devices or networking is an integral component of public safety and urban management.

Author Contributions: Conceptualization, K.M. and W.Y.; methodology, K.M.; software, K.M.; validation, K.M.; formal analysis, K.M.; investigation, K.M.; resources, K.M. and W.Y.; data collection, K.M.; writing—original draft preparation, K.M.; writing—review and editing, K.M., W.Y., M.N., and X.L.; visualization, K.M.; supervision, W.Y. and M.N.; project administration W.Y. and M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aldairi, A.; Tawalbeh, L. Cyber security attacks on smart cities and associated mobile technologies. *Procedia Comput. Sci.* **2017**, *109*, 1086–1091. [CrossRef]
2. Alketbi, A.; Nasir, Q.; Abu Talib, M. Novel blockchain reference model for government services: Dubai government case study. *Int. J. Syst. Assur. Eng. Manag.* **2020**, *11*, 1170–1191. [CrossRef]
3. Gedara, K.M.; Nguyen, M.; Yan, W.Q. Enhancing privacy protection in intelligent surveillance: Video blockchain solutions. In *Blockchain and Applications, 5th International Congress*; Machado, J.M., Prieto, J., Vieira, P., Peixoto, H., Abelha, A., Arroyo, D., Vigneri, L., Eds.; BLOCKCHAIN 2023. Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 778. [CrossRef]
4. Gedara, K.M.; Nguyen, M.; Yan, W.Q. *Visual Blockchain for Intelligent Surveillance in a Smart City*; IGI Global: Hershey, PA, USA, 2022; pp. 210–222. [CrossRef]
5. Moolikagedara, K.; Nguyen, M.; Yan, W.Q.; Li, X.J. Video Blockchain: A decentralized approach for secure and sustainable networks with distributed video footage from vehicle-mounted cameras in smart cities. *Electronics* **2023**, *12*, 3621. [CrossRef]
6. Hu, R.; Yan, W.Q. Design and implementation of visual blockchain with Merkle tree. In *Handbook of Research on Multimedia Cyber Security*; IGI Global: Hershey, PA, USA, 2020; pp. 282–295.
7. Shu, Y.; Yu, J.; Yan, W.Q. *Blockchain for Security of a Cloud-Based Online Auction System*; IGI Global: Hershey, PA, USA, 2019; pp. 189–210.
8. Fu, J.; Qiao, S.; Huang, Y.; Si, X.; Li, B.; Yuan, C. A Study on the Optimization of Blockchain Hashing Algorithm Based on PRCA. *Secur. Commun. Netw.* **2020**, *2020*, 8876317. [CrossRef]
9. Priyadharshini, K.; Canessane, R.A. Blockchain-Based Security Algorithm on IoT Framework for Shielded Communication in Smart Cities. In Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021, Tirunelveli, India, 4–6 February 2021; pp. 320–327. [CrossRef]
10. Lin, I.C.; Liao, T.C. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659. [CrossRef]
11. Swaminathan, A.; Mao, Y.; Wu, M. Robust and secure image hashing. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 215–230. [CrossRef]
12. Huang, J.-J. Consistent Fuzzy Analytic Hierarchy Process by Considering Fuzzy Input and Output Data. In Proceedings of the 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS), Sapporo, Japan, 25–28 August 2016. [CrossRef]
13. Sayogo, D.S.; Najafabadi, M.M.; Tayi, G.K.; Pardo, T.A. Privacy, confidentiality, and security challenges for interoperable data platforms in supply chains. *Public Adm. Inf. Technol.* **2016**, *26*, 109–128. [CrossRef]

14. Khan, P.W.; Byun, Y.C.; Park, N. A data verification system for cctv surveillance cameras using blockchain technology in smart cities. *Electronics* **2020**, *9*, 484. [CrossRef]
15. George, R.V.; Harsh, H.O.; Ray, P.; Babu, A.K. Food quality traceability prototype for restaurants using blockchain and food quality data index. *J. Clean Prod.* **2019**, *240*, 118021. [CrossRef]
16. Chen, J.; Ruan, Y.; Guo, L.; Lu, H. BCVehis: A Blockchain-based service prototype of vehicle history tracking for used-car trades in China. *IEEE Access* **2020**, *8*, 214842–214851. [CrossRef]
17. Deepak, K.; Badiger, A.N.; Akshay, J.; Awomi, K.A.; Deepak, G.; Harish Kumar, N. Blockchain-Based Management of Video Surveillance Systems: A Survey. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020, Coimbatore, India, 6–7 March 2020; pp. 1256–1258. [CrossRef]
18. Zajac, P. Ephemeral keys authenticated with merkle trees and their use in iot applications. *Sensors* **2021**, *21*, 2036. [CrossRef] [PubMed]
19. Michelin, R.A.; Ahmed, N.; Kanhere, S.S.; Seneviratne, A.; Jha, S. Leveraging Lightweight Blockchain to Establish Data Integrity for Surveillance Cameras. In Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020, Toronto, ON, Canada, 2–6 May 2020; pp. 3–5.
20. Gergely, A.M.; Crainicu, B. Randadminsuite: A new privacy-enhancing solution for private blockchains. *Procedia Manuf.* **2020**, *46*, 562–569. [CrossRef]
21. Fitwi, A.; Chen, Y. Secure and Privacy-Preserving Stored Surveillance Video Sharing Atop Permissioned Blockchain. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021; pp. 1–8.
22. Hasan, O.; Brunie, L.; Bertino, E. Privacy-Preserving reputation systems based on blockchain and other cryptographic building blocks: A survey. *ACM Comput. Surv. (CSUR)* **2023**, *55*, 1–37. [CrossRef]
23. Du, J.; Jiang, C.; Gelenbe, E.; Xu, L.; Li, J.; Ren, Y. Distributed data privacy preservation in iot applications. *IEEE Wirel. Commun.* **2018**, *25*, 68–76. [CrossRef]
24. Ali, M.S.; Dolui, K.; Antonelli, F. IoT data privacy via blockchains and IPFS. In *ACM International Conference Proceeding Series*; Association for Computing Machinery: New York, NY, USA, 2017.
25. Loukil, F.; Ghedira-Guegan, C.; Boukadi, K.; Benharkat, A.N.; Benkhelifa, E. Data privacy based on iot device behavior control using blockchain. *ACM Trans. Internet Technol.* **2021**, *21*, 1–20. [CrossRef]
26. Kalbo, N.; Mirsky, Y.; Shabtai, A.; Elovici, Y. The security of ip-based video surveillance systems. *Sensors* **2020**, *20*, 4806. [CrossRef] [PubMed]
27. Zhu, G.; Ding, Y.; Cao, Y. The Effect of block-matrix interface of srm with high volumetric block proportion on its uniaxial compressive strength. *Appl. Sci.* **2023**, *13*, 3463. [CrossRef]
28. Majdoubi, D.E.L.; El Bakkali, H.; Sadki, S. Towards Smart Blockchain-Based System for Privacy and Security in a Smart City Environment. In Proceedings of the 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications, CloudTech, Marrakesh, Morocco, 24–26 November 2020. [CrossRef]
29. Drijvers, M.; Edalatnejad, K.; Ford, B.; Kiltz, E.; Loss, J.; Neven, G.; Stepanovs, I. On the Security of Two-Round Multi-Signatures. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 1084–1101. [CrossRef]
30. Anajemba, J.H.; Tang, Y.; Iwendi, C.; Ohwoekevwo, A.; Srivastava, G.; Jo, O. Realizing efficient security and privacy in IoT networks. *Sensors* **2020**, *20*, 2609. [CrossRef] [PubMed]
31. Fattahi, S.M.; Makanju, A.; Milani Fard, A. SIMBA: An efficient simulator for blockchain applications. In Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks: Supplemental Volume, DSN-S 2020, Valencia, Spain, 29 June–2 July 2020; pp. 51–52. [CrossRef]
32. Koo, D.; Shin, Y.; Yun, J.; Hur, J. Improving security and reliability in Merkle tree-based online data authentication with leakage resilience. *Appl. Sci.* **2018**, *8*, 2532. [CrossRef]
33. Lee, D.; Park, N. Blockchain based privacy preserving multimedia intelligent video surveillance using secure Merkle tree. *Multimed. Tools Appl.* **2020**, *80*, 34517–34534. [CrossRef]