Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 2 (2025)

Design of a Decentralized Authentication and Off-Chain Data Management Protocol for VANETs Using Blockchain

Mrs. Misbah Kousar[1] , Dr. Sanjay Kumar[2] , Dr. Mohammed Abdul Bari[3]
Ph. Scholar in Kalinga University; Email: mkouar11@gmail.com
Associate Professor, Kalinga University (CSE DEPT)
Professor –CSE-ISL Engineering College; Email: abdulbarimohammed11@gmail.com

Article History:

Received: *17-07-2024*

Revised: *08-09-2024*

Accepted: *29-09-2024*

Abstract:

In the evolving landscape of Intelligent Transportation Systems (ITS), the need for secure and reliable data sharing is more critical than ever. As ITS increasingly rely on the exchange of sensitive information, challenges related to data privacy, security, and the integrity of communication have emerged. Traditional approaches to securing vehicular networks often depend on centralized Trusted Third Parties (TTP) and cloud-based infrastructure, which introduce vulnerabilities such as single points of failure and potential latency issues. This paper proposes a novel decentralized architecture for Vehicular Ad-Hoc Networks (VANETs) that enhances security by eliminating the dependency on centralized cloud servers. The proposed protocol leverages blockchain technology to facilitate secure off-chain storage of event data and implements a decentralized authentication mechanism. This approach ensures that the privacy and integrity of shared information are maintained without compromising the efficiency of the system.

Keywords: Decentralized Authentication, Vehicular Ad-Hoc Networks (VANETs), Blockchain Technology, Off-Chain Data Management, Intelligent Transportation Systems (ITS), Data Privacy and Security.

1. Introduction

In recent years, Intelligent Transportation Systems (ITS) have significantly enhanced road safety and optimized traffic management by enabling efficient sharing of critical information, such as traffic congestion alerts and accident notifications. These advancements have led to improved decision-making and reduced response times, ultimately contributing to safer and more efficient transportation networks [1, 2]. However, as these systems increasingly rely on the exchange of sensitive data, concerns regarding the privacy of vehicle information and the security of shared event data have come to the forefront.

The key challenge lies in securely sharing event information while maintaining the privacy of the involved parties and ensuring the integrity of the data, all without relying on a centralized Trusted Third Party (TTP) [3]. Traditional approaches often depend on centralized cloud servers for data storage and authentication, which introduces potential vulnerabilities, such as single points of failure and susceptibility to external attacks. Furthermore, the dependence on cloud servers raises concerns about latency and the potential for data breaches, especially as the volume of data and the number of connected vehicles continue to grow [4].

Blockchain technology has emerged as a promising solution to these challenges by offering a decentralized and tamper-resistant platform for secure data exchange [5]. Building on the foundation of blockchain, previous research has proposed various protocols for secure event sharing and vehicle authentication within ITS [6]. These protocols address critical issues related to the safe storage of event information and the authentication of vehicles, ensuring that only authorized entities can access sensitive data. However, many existing solutions still rely heavily on cloud-based infrastructure, particularly for vehicle authentication, which undermines the decentralized nature of the system and reintroduces some of the vulnerabilities associated with centralized models [7].

To address these issues, this paper proposes a novel decentralized architecture for Vehicular Ad-Hoc Networks (VANETs) that eliminates the dependency on cloud servers for vehicle authentication and data management [8]. The proposed protocol leverages blockchain technology to ensure secure and efficient off-chain storage of event data, while also implementing decentralized authentication mechanisms that do not rely on a TTP. By decentralizing both the authentication process and data storage, this approach enhances the security and privacy of ITS, making it more resilient to attacks and more reliable in delivering timely and accurate information to road users.

This paper aims to build upon existing research by introducing a comprehensive solution that integrates decentralized authentication with off-chain data management, creating a more secure and efficient ITS framework. The proposed architecture not only addresses the limitations of current cloud-dependent models but also lays the groundwork for future advancements in decentralized transportation systems.

2. Literature Review

Trusted Distributed Information Sharing Protocol: The TDISP (Trusted Distributed Information Sharing Protocol) scheme attempts to introduce decentralization in Vehicular Ad-Hoc Networks (VANETs) by involving Roadside Units (RSUs) in the data collection process. In this scheme, RSUs collect registration details from vehicles (VEHs) and forward them to a central cloud server (CS), where all critical data is stored in a centralized ledger [9]. Although this approach incorporates some decentralization by engaging RSUs, the ultimate control remains with the CS, which manages and stores all the sensitive information. This centralization introduces significant vulnerabilities, as it creates a single point of control that, if compromised, could expose or manipulate the entire network's data.

The primary limitation of this partially decentralized system is the inherent risk associated with centralized data control [10]. The CS's role as the sole repository for all RSU and VEH data means that any breach or failure at the CS level could have catastrophic consequences for the network. Additionally, as the number of vehicles in the network grows, the CS faces scalability challenges, with increased data processing and storage demands potentially leading to performance issues [11]. Furthermore, the centralization of the authentication process within the CS introduces latency and creates a single point of failure, where the entire network could be disrupted if the CS goes offline.

To address these issues, a fully decentralized approach is necessary, where control and data storage are distributed across multiple nodes, such as RSUs or even the vehicles themselves, using blockchain or distributed ledger technology [12]. In such a system, no single entity would hold complete control, enhancing security and resilience against attacks or failures. This fully decentralized model would not only eliminate the single point of failure but also improve scalability and ensure that data integrity is maintained through consensus mechanisms, making the network more robust and reliable.

Single-Point-of-Failure: The TDISP scheme's centralized architecture introduces a significant vulnerability known as the Single-Point-of-Failure (SPoF) problem. In this system, the entire functionality of the Vehicular Ad-Hoc Network (VANET) relies heavily on the availability and operational integrity of the central cloud server (CS) [13]. The CS is responsible for managing and distributing critical data, including vehicle authentication details and event information. If the CS becomes unavailable due to technical issues, cyber-attacks, or maintenance, the entire network's operation is jeopardized [14]. This reliance on a single central entity poses a serious risk, as the failure of the CS could lead to a complete breakdown in communication and functionality across the network.

The SPoF problem not only threatens the network's resilience but also impacts the reliability of the Intelligent Transportation System (ITS). For instance, in critical scenarios such as emergency vehicle coordination or real-time traffic management, any disruption caused by a failure at the CS could have severe consequences [15]. The centralization of data and authentication processes means that if the CS is compromised or becomes a target of a distributed denial-of-service (DDoS) attack, the entire network could be paralyzed, leaving vehicles unable to authenticate and communicate effectively. This vulnerability highlights the inadequacies of relying on a single server to manage and safeguard the network's operations [16-18].

Addressing the SPoF problem requires a shift towards a decentralized architecture, where the responsibilities of the CS are distributed across multiple nodes in the network. By decentralizing the data storage and authentication processes using blockchain or distributed ledger technology, the network can eliminate its reliance on a single entity [19]. This approach ensures that even if one or more nodes fail, the network can continue to function without interruption, as other nodes would maintain the necessary data and processing capabilities. Decentralization not only mitigates the risk of a single-point-of-failure but also enhances the overall security, reliability, and robustness of the VANET.

Centralized Authentication Mechanism: The TDISP scheme’s reliance on a centralized authentication mechanism poses another critical limitation within Vehicular Ad-Hoc Networks (VANETs). In this scheme, vehicle (VEH) authentication is entirely managed by the central cloud server (CS). The process involves vehicles sending their event information to Roadside Units (RSUs), which then relay this data to the CS. The CS is responsible for performing all necessary computations to authenticate the vehicles and subsequently informs the RSUs of the authentication outcome [20]. This centralized approach introduces several challenges, including increased latency, as all authentication requests must be processed by the CS before vehicles can be authorized to participate in network activities.

One of the major drawbacks of this centralized authentication process is its vulnerability to attacks targeting the central server [21]. Since the CS is the sole entity responsible for verifying the authenticity of vehicles, any successful breach of the CS could compromise the entire network’s security. An attacker who gains control of the CS could potentially authorize unauthorized vehicles or deny legitimate ones, leading to significant disruptions in network operations. Additionally, the centralization of authentication duties increases the likelihood of bottlenecks, particularly in scenarios involving high volumes of vehicles, as the CS must handle all authentication requests, potentially slowing down the process and impacting the network's efficiency [22-24].

To overcome these limitations, a decentralized authentication mechanism is necessary. By distributing the authentication process across multiple nodes, such as RSUs or even the vehicles themselves, the network can reduce its dependency on a single central server, thereby enhancing security and reducing latency [25,26]. In a decentralized system, authentication responsibilities are shared, making it much more difficult for an attacker to compromise the entire network. This approach not only improves the speed and reliability of the authentication process but also ensures that the network remains resilient in the face of potential security threats, offering a more robust solution for secure vehicle authentication in VANETs [27].

Storage Scalability Constraint: The TDISP scheme also faces significant challenges related to storage scalability, particularly as it mandates that Roadside Units (RSUs) must have the capacity to store extensive information for all vehicles (VEHs) within the network. This requirement imposes a heavy burden on RSUs, especially in large-scale networks with a high volume of vehicles constantly generating and transmitting data [28]. As the number of vehicles grows, the amount of data that needs to be stored and managed by the RSUs increases exponentially, leading to potential performance bottlenecks and difficulties in maintaining data integrity across the network [29].

The centralized storage model within the TDISP scheme not only creates issues related to storage capacity but also raises concerns about data redundancy and consistency. Since the RSUs are tasked with storing vast amounts of data, there is a risk that some RSUs may become overloaded, leading to delays in data retrieval and processing [30]. Moreover, the reliance on RSUs for storage without sufficient redundancy measures could result in data loss or corruption if an RSU fails or is compromised. This lack of a scalable and reliable storage infrastructure undermines the overall effectiveness of the VANET, making it difficult to ensure the continuous and secure management of vehicular data [31-33].

To address these storage scalability constraints, the proposed decentralized authentication and off-chain storage protocol introduces a more efficient and scalable approach. By leveraging off-chain storage solutions, the system can significantly reduce the storage burden on RSUs. Data is stored in a distributed manner across multiple nodes, with blockchain technology ensuring data integrity and consistency through consensus mechanisms [34,35]. This decentralized approach not only alleviates the storage demands on individual RSUs but also enhances the network's resilience by preventing data bottlenecks and ensuring that data remains accessible even if some nodes fail [36]. The result is a more scalable, efficient, and secure VANET that can effectively manage large volumes of data without compromising performance or security.

3. Proposed System

The proposed system aims to enhance the TDISP protocol by eliminating the need for a trusted cloud server and leveraging decentralized technologies such as the InterPlanetary File System (IPFS) and blockchain. This solution addresses the key weaknesses identified in the TDISP protocol by introducing a blockchain-based data sharing and authentication scheme, ensuring secure and reliable communication within the vehicular network. By utilizing a decentralized authentication protocol, the proposed system mitigates the risks associated with centralized control, enhancing the overall security and resilience of the network.

In addition to blockchain, the system incorporates IPFS for distributed storage of event information, further decentralizing data management and reducing reliance on a single point of failure. The use of Ethereum smart contracts ensures that only authorized users can access event data, enforcing strict data access policies. A consensus procedure for establishing a common key among network participants is also integrated, ensuring consistency and security across the network. Security analyses confirm that the proposed protocol is robust against various attacks, such as Sybil and man-in-the-middle attacks, while also being cost-effective in terms of computation, communication, and storage compared to the original TDISP scheme. This innovative approach offers a more secure, scalable, and reliable solution for managing vehicular networks, paving the way for future advancements in decentralized transportation systems.

Network Administrator (N_Admin): In the proposed protocol, the Network Administrator (N_Admin) is responsible for the initial registration of Roadside Units (RSUs) before they are deployed into the network. During this offline registration process, the N_Admin generates and distributes a common key (CommonKey_RSU_x) to all RSUs[37]. This key is used by the RSUs to securely exchange hash values received from the InterPlanetary File System (IPFS). The validity of the CommonKey_RSU_x is limited to a fixed duration TTT, after which the RSUs must regenerate the common key through a consensus procedure and distribute the new key (NewKey_RSU_x) to all participating RSUs to ensure continued secure communication.

Roadside Unit (RSU_x): RSU_x plays a crucial role in the registration and validation of vehicles (Vehicle_y) within its designated range, which typically spans three to four kilometers. When a Vehicle_y enters the network, it must register with the nearest RSU_x. Upon successful registration, the RSU_x selects a set of registration parameters and initiates the creation of a new data block (Block_i) in the blockchain. This block undergoes a validation process, and if validated successfully, the RSU_x issues an index for this block (Index_Block_i) to the Vehicle_y. For subsequent authentication, Vehicle_y must present this index along with its registration parameters to verify its identity within the network.

Vehicle (Vehicle_y): The Vehicle (Vehicle_y) is equipped with an On-Board Unit (OBU), Wi-Fi, GPS, and other necessary communication tools. Due to its limited computational and storage capabilities, Vehicle_y only retains essential data, such as its identity (ID_Vehicle_y), the identifier of the RSU_x (ID_RSU_x), and the temporary identifier assigned during registration (TempID_Vehicle_y). Vehicle_y gathers event information (Event_data) and forwards it to the nearest RSU_x for further processing. This streamlined process ensures that vehicles maintain secure communication while minimizing the load on their computational resources.

Blockchain and IPFS: The proposed methodology emphasizes decentralized authentication and distributed data storage (events) by leveraging IPFS and blockchain technologies. IPFS, through its content-addressable nature, provides a solution to the challenges posed by blockchain storage. The fundamental concept of IPFS is the decentralized storage of encrypted data or files across nodes, with IPFS offering the hash of this encrypted data as a retrievable service. Accessing the encrypted data from IPFS is accomplished using this hash.

Smart Contracts: In contrast to traditional contracts, smart contracts are automated programs that execute when certain conditions, agreed upon by the participating peers (or nodes), are fulfilled. Operating within a decentralized environment such as an Ethereum-based blockchain network[38-39], smart contracts ensure that transactions or asset validations occur without the need for third-party intervention. These contracts are immutable and cryptographically secure. Our proposed methodology utilizes Ethereum-enabled smart contracts to securely store event details in IPFS and retrieve these details via the RSU (as outlined in Algorithm 3.1). This ensures that the event details within IPFS remain immutable. Consequently, authorized users can access event details from blockchain-enabled IPFS servers in the future, provided that the smart contract conditions are satisfied.

Registration Phase

Roadside Unit (RSU) Registration: The registration process for a Roadside Unit (RSU) is executed through the following steps, with corresponding equations to describe the process mathematically:

Step 1: The RSU, denoted as **RSU_x**, begins the registration process by selecting a unique identifier, **ID_RSU_x**. This identifier is securely transmitted to the Network Administrator (NA) using a secure channel:

$$ID\_RSU\_x \rightarrow NA$$

Here, the arrow indicates the secure transmission of **ID_RSU_x** from **RSU_x** to **NA**.

Step 2: Upon receiving **ID_RSU_x**, the NA generates a corresponding public and private key pair for **RSU_x**. The public key generator (PKG) is used for this purpose:

{PubKey_{RSU_x}, PrivKey_{RSU_x}}= PKG(ID_RSU_x)

This equation shows that the key pair **PubKey_{RSU_x}** and **PrivKey_{RSU_x}** is generated based on the unique identifier **ID_RSU_x**.

Step 3: The NA securely delivers the generated key pair to **RSU_x**. This can be represented as:

{PubKey_{RSU_x}, PrivKey_{RSU_x}}→ RSU_x

The arrow indicates the secure delivery of the keys to **RSU_x**.

Step 4: Once **RSU_x** receives the key pair, it publicly declares its public key **PubKey_{RSU_x}**:

PubKey_{RSU_x}→Public

This indicates that the public key **PubKey_{RSU_x}** is broadcasted to other RSUs and entities in the system. Simultaneously, **RSU_x** securely stores its private key **PrivKey_{RSU_x}**:

PrivKey_{RSU_x}→Secure Storage

Step 5: Before deploying **RSU_x** in the system, it is necessary to agree upon a common key **K** that will be used for secure communication within the network. The common key **K** is generated and agreed upon through a consensus process, as outlined in Subsection 3.6:

K=Consensus({RSU_x})

This equation represents the generation of the common key **K** based on the consensus among participating **RSU_x** units.

Vehicle (VEH) Registration

The following steps are executed by the vehicle **VEH_b** during the registration process:

Step 1: The user of the vehicle **VEH_b** selects a unique identity **ID_VEH_b** and sends it along with a bio-hashed version of the user's biometric information **H(Binfo)** to the nearest Roadside Unit (RSU), denoted as **RSU_a**. This transfer can be performed offline or through in-person communication.

ID_VEH_b→RSU_a

H(Binfo)→RSU_a

These equations represent the secure transmission of the unique identity and the bio-hashed biometric information from the vehicle to the RSU.

Step 2: Upon receiving **ID_VEH_b** and **H(Binfo)**, the RSU **RSU_a** computes a temporary identity **TID_VEH_b** using the following equation:

TID_VEH_b=h(ID_VEH_b || N_VEH_b)

Where **N_VEH_b** represents a nonce generated for the vehicle, and **h(·)** denotes a cryptographic hash function. Additionally, the RSU computes the hash of the temporary identity combined with the biometric hash:

h(TID_VEH_b || H(Binfo))

This equation is used to create an intermediary hash value that will be used in the subsequent steps.

Step 3: After calculating the temporary identity, **RSU_a** computes the parameter **A_VEH_b** by further hashing the combination of the temporary identity **TID_VEH_b** and the biometric hash **H(Binfo)**:

A_VEH_b=h(TID_VEH_b || H(Binfo))

The RSU also computes an additional verification parameter by hashing **ID_VEH_b** with **H(Binfo)**:

V_VEH_b=h(ID_VEH_b || H(Binfo))

These parameters are crucial for verifying the identity of the vehicle during the registration process.

Step 4: Next, **RSU_a** issues a challenge **C_VEH_b** to the vehicle and uses a Physical Unclonable Function (PUF) to generate a response **R_VEH_b**:

C_VEH_b→VEH_b

R_VEH_b=PUF(C_VEH_b)

This step ensures that the vehicle's identity is verified in a secure manner.

Step 5: The RSU **RSU_a** stores the computed values **TID_VEH_b**, **ID_RSU_a**, **C_VEH_b**, **PUF**, **A_VEH_b**, in the On-Board Unit (OBU) of the vehicle **VEH_b** for future authentication processes:

Store(TID_VEH_b, ID_RSU_a, C_VEH_b, PUF, A_VEH_b)

Additionally, the RSU creates a new block **B_1** by hashing the concatenated parameters:

B1=h(A_VEH_b || ID_RSU_a || R_VEH_b)

Transaction-generation (Event Detection) Phase

During this phase, one of the vehicles **VEH_b** collects critical events (e.g., accidents on the road, traffic congestion, etc.) through its On-Board Unit (OBU) and Wi-Fi units. The vehicle then sends the collected event data **E_m** along with the parameters **TID_VEH_b** and **ID_RSU_a** to the nearest Roadside Unit (RSU). The RSU verifies the integrity of the event and determines the necessary actions. We assume that the event **E_m** is collected by **VEH_b** and subsequently forwarded to **RSU_a**.

Step 1: The vehicle **VEH_b** computes the hash of the event data **E_m** and the response **R_VEH_b** received from the Physical Unclonable Function (PUF), forming a parameter **B**:

B = h(E_m || R_VEH_b)

The vehicle **VEH_b** then sends the following parameters to **RSU_a**: **TID_VEH_b**, **ID_RSU_a**, **A_VEH_b**, **C_VEH_b**, **R_VEH_b**, **B**, **E_m**, and an integrity block **IB_1**. These parameters are encrypted using the public key of **RSU_a**, denoted as **P_{RSU_a}**, ensuring secure transmission:

{TID_VEH_b, ID_RSU_a, A_VEH_b, C_VEH_b, R_VEH_b, B, E_m, IB_1} = E_P_RSU_a (TID_VEH_b, ID_RSU_a, A_VEH_b, C_VEH_b, R_VEH_b, B, E_m, IB_1)

The Physical Unclonable Function (PUF) is used by **VEH_b** to generate **R_VEH_b**, ensuring that the response is unique to the specific challenge issued by **RSU_a**.

Step 2: Once **RSU_a** receives the encrypted parameters, it decrypts the data using its private key **PR_{RSU_a}** to obtain the original parameters:

{TID_VEH_b, ID_RSU_a, A_VEH_b, C_VEH_b, R_VEH_b, B, E_m, IB_1} = D_PR_RSU_a(E_P_RSU_a (TID_VEH_b, ID_RSU_a, A_VEH_b, C_VEH_b, R_VEH_b, B, E_m, IB_1))

This decryption process ensures that the data sent by **VEH_b** remains confidential and unaltered during transmission.

Step 3: The RSU **RSU_a** computes a new hash value **B'** based on the event data **E_m** and the response **R_VEH_b**:

B' = h(E_m || R_VEH_b)

The RSU then verifies whether the computed hash **B'** matches the hash **B** received from **VEH_b**:

B'=?B

This comparison helps verify the integrity and authenticity of the event data **E_m**.

Step 4: If the condition **B' == B** holds true, **RSU_a** confirms that the event **E_m** provided by **VEH_b** is correct and has not been altered by any adversary in the public channel. This ensures the integrity of the data:

If $B' == B$, then E_m is validated.

Step 5: After confirming the accuracy of **E_m**, **RSU_a** further verifies the authenticity of **VEH_b** by referencing the blockchain, which is detailed in Subsection 3.4. This step ensures that the event and the vehicle's identity are securely and reliably authenticated.

Block Creation Phase

In this phase, **RSU_a** generates blocks **B_1** and **B_2**, which are created through a proof-of-work consensus mechanism. The blocks are formed using specific parameters: **A_VEH_b**, **ID_RSU_a**, **C_VEH_b**, and **R_VEH_b** for **B_1**, and **TID_VEH_b** and **E_m** for **B_2**. These parameters are utilized in the computation of the Merkle root **root_mer** for each block. The **root_mer** is stored in the block header and is essential for calculating the current block hash, **hash_cur.**

$$hash\_cur = h(hash\_pre \,||\, root\_mer \,||\, N \,||\, TS\_i)$$

Where: **hash_pre** is the hash of the previous block, **N** is the nonce used for computing the hash value of the current block, **TS_i** is the timestamp.

For block **B_1**, the Merkle root **root_mer** is computed as follows:

$$root\_mer = h(H\_1 \,||\, H\_2)$$

Where:

$$H\_1 = h(h(A\_VEH\_b) \,||\, h(ID\_RSU\_a))^{2}$$

$$H\_2 = h(h(C\_VEH\_b) \,||\, h(R\_VEH\_b))^{3}$$

This computation involves hashing the combined values of **A_VEH_b** and **ID_RSU_a** to form **H_1**, and similarly combining and hashing **C_VEH_b** and **R_VEH_b** to form **H_2**. The final Merkle root **root_mer** is derived by hashing the concatenation of **H_1** and **H_2**.

For block **B_2**, the Merkle root **root_mer** is calculated as:

$$root\_mer = h(H\_3 \,||\, H\_4)$$

Where:

$$H\_3 = h(TID\_VEH\_b)^{4}$$

$$H\_4 = h(E\_m)^{5}$$

In this case, **H_3** is obtained by hashing **TID_VEH_b**, and **H_4** is derived from hashing **E_m**. The Merkle root **root_mer** for **B_2** is then obtained by hashing **H_3** and **H_4** together.

Initially, the nonce **N** is set to zero. In each iteration, it is incremented by one, and the values of **hash_pre**, **root_mer**, and **TS_i** are repeatedly hashed with different values of **N** until the computation of the current block hash **hash_cur** is achieved. After successfully creating the block, **RSU_a** hands over the index of block **B_1** (i.e., **IB_1**) to the vehicle **VEH_b**.

As a result, the blocks created by **RSU_a** are used by other **RSU**s in the system to accomplish the decentralized proof-of-work consensus mechanism. This ensures that **VEH_b** can securely store and validate the block in the blockchain, maintaining the integrity and security of the data.

Vehicle Authentication and Event Storage Process of TDAOP Scheme

In a previous section, we considered a scenario where the vehicle **VEH_b** collects **E_m** and sends the parameters **⟨TID_VEH_b, ID_RSU_a, A_VEH_b, C_VEH_b, R_VEH_b, B, E_m, IB_1⟩** to **RSU_a**. The following steps describe the vehicle authentication process:

Step 1: Upon receiving the parameters **⟨TID_VEH_b, ID_RSU_a, A_VEH_b, R_VEH_b, B, E_m, IB_1⟩**, **RSU_a** utilizes the block index **IB_1** to locate the corresponding Merkle root **root_mer**. This search is efficient, requiring constant time, denoted as **O(constant).**

Step 2: After identifying **root_mer**, **RSU_a** recalculates a new root, denoted as **root_mer′**, based on the received parameters. The equation for this recalculation is:

$$\mathbf{root\_mer'=h(H\_5||H\_6)}$$

where:

$$H\_5=h(h(A\_VEH\_b)||h(ID\_RSU\_a))$$

$$H\_6=h(h(C\_VEH\_b)||h(R\_VEH\_b))$$

Step 3: After calculating **root_mer′**, **RSU_a** checks whether **root_mer′** matches **root_mer**.

Step 4: If **root_mer′** equals **root_mer**, **RSU_a** concludes that **VEH_b** is authentic.

Step 5: Upon successful authentication of **VEH_b** and verification of **E_m**, **RSU_a** stores the event in the decentralized InterPlanetary File System (IPFS).

Step 6: For the event **E_m** uploaded to the IPFS, **RSU_a** creates a hash of **E_m** and provides this hash to **RSU_a**.

Step 7: **RSU_a** records this hash in its ledger and shares it with other **RSU_s** using a common key **K**.

Step 8: **RSU_a** selects a nonce **N_a** and calculates **PR1_RSU_a** as:

$$\mathbf{PR1\_RSU\_a=h(PR\_RSU\_a||N\_a)}$$

Step 9: **RSU_a** encrypts the hash using **PR1_RSU_a** and stores the encrypted hash in its local ledger.

Step 10: Steps 8 and 9 are repeated by all **RSU_s** to share and store the hash across the network.

Step 11: Finally, **PR1_RSU** is used by the **RSU_s** to securely transmit the event data among each other, maintaining the integrity and security of the event information throughout the network.

Procedure for Generating the New Key K_1

The existing key is shared among the **RSU_s** using public and private keys generated by **N_A ⟨PU_RSU, PR_RSU⟩**. However, there's a potential risk that **N_A** could expose **PR_RSU** to an unauthorized party, enabling them to track the supplied hash. To mitigate this risk, it is necessary to periodically update the key.

Let's consider a scenario where **RSU_a** aims to generate a new common key **K_1**. To achieve this, **RSU_a** adds the current key into the existing pool of **N** hashes and then randomly selects **M** hashes, where **M≤N+1**. Following this selection, the new common key is generated by repeating the same steps as described previously.

The key difference in this process is that, instead of the previously used key **K**, the newly generated key **K_1** is employed to compute parameters **S_1** and **S_2**, and the encryption and decryption operations are carried out using **K_1**. Once consensus is reached on **K_1**, the **RSU_s** utilize it for exchanging the hash provided by the IPFS.

4. Results And Discussion

Performance Analysis of TDAOP Scheme

In this section, we evaluate the performance of the proposed TDAOP scheme and compare it with the existing TDISP scheme. The evaluation focuses on security properties such as the computational cost, storage requirements for nodes, and communication costs between nodes. These metrics are essential for assessing the performance of both schemes.

To conduct this comparison, we considered a scenario involving a single vehicle **VEH_b** and a single roadside unit **RSU_a**. The proposed approach primarily utilizes encryption **E(.)** and decryption **D(.)** operations alongside a lightweight hashing function **h(.)**. A 256-bit hash function is employed to compute **T_h**, and the **PKG** is utilized (for the first time only) to generate the keys for the **RSU**. For comparison with the existing scheme, we assume that the vehicle and **RSU** identities **ID_RSU_a**, **ID_VEH_b** and the parameters **⟨TID_VEH_b, A_VEH_b, C_VEH_b⟩** are each 128 bits in length.

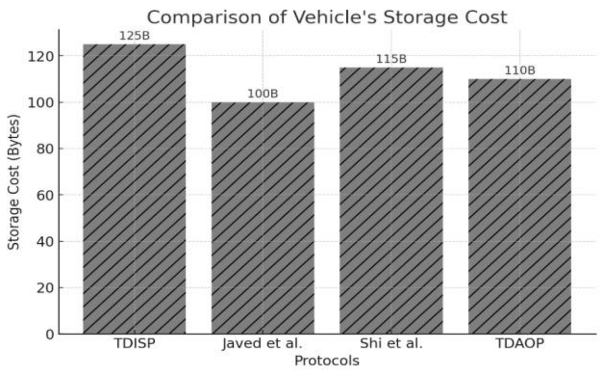

Figure 1. Comparison of Vehicles Storage Cost.

The bar chart in Figure 1 illustrates a comparative analysis of the storage costs associated with different protocols for vehicle storage. Among the protocols, TDISP incurs the highest storage cost at 125 bytes, while the Javed et al. protocol demonstrates the lowest storage cost of 100 bytes. The Shi et al. protocol shows a storage cost of 115 bytes, positioning it between the extremes. The TDAOP protocol, introduced as part of the current work, has a storage cost of 110 bytes, indicating a more optimized approach compared to TDISP, yet slightly more storage-intensive than the Javed et al. protocol. This comparison underscores the efficiency of the TDAOP protocol in balancing storage requirements while maintaining robust functionality.

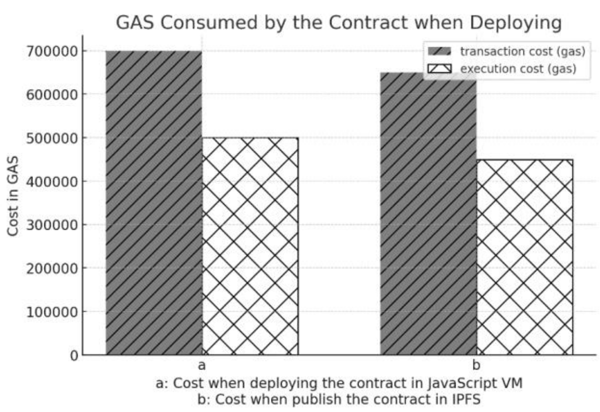

Figure 2. Deploying Cost of Smart Contract.

Figure 2 illustrates the gas consumption associated with deploying a smart contract in two scenarios: (a) within a JavaScript VM and (b) when publishing the contract in IPFS. The chart highlights two types of costs—transaction and execution costs. Deploying the contract in the JavaScript VM incurs a higher gas cost, with approximately 700,000 gas units for transactions and 500,000 gas units for execution. In comparison, publishing the contract in IPFS shows a slightly lower transaction cost of around 650,000 gas units and an execution cost of 450,000 gas units. This comparison suggests that while both deployment methods are gas-intensive, using IPFS may offer a more efficient approach in terms of gas consumption.

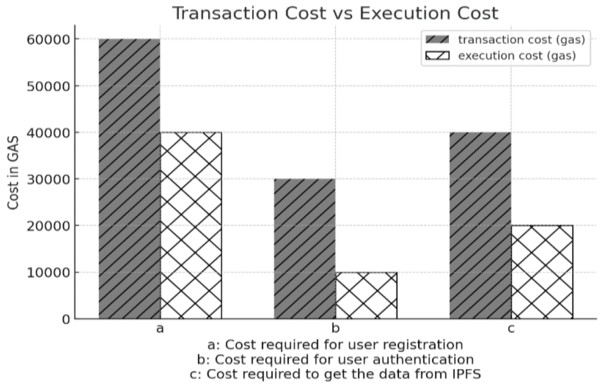

Figure 3. Transaction and Execution Cost of Smart Contract.

Figure 3 compares the transaction and execution costs in gas for three operations related to a smart contract: user registration (a), user authentication (b), and retrieving data from IPFS (c). The chart shows that user registration incurs the highest transaction cost at around 60,000 gas units, with an execution cost of about 40,000 gas units. User authentication is less demanding, with transaction costs at approximately 30,000 gas units and execution costs around 10,000 gas units. Retrieving data from IPFS has a moderate transaction cost of 40,000 gas units and a lower execution cost of 20,000 gas units. Overall, the figure highlights that user registration is the most gas-intensive operation, particularly in terms of transaction costs, compared to the other two operations.

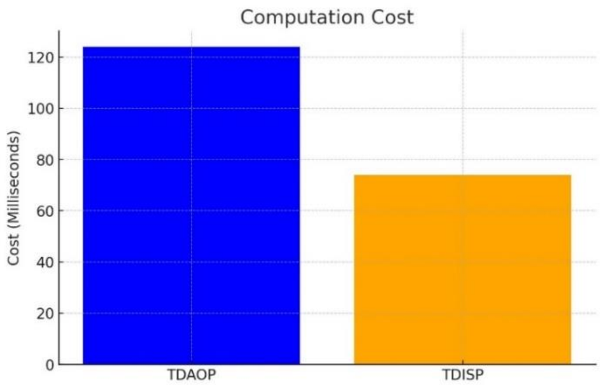

Figure 4. Comparison between TDAOP and TDISP concerning computation cost.

Figure 4 provides a comparison between the TDAOP and TDISP protocols concerning their computation costs, measured in milliseconds. The bar chart shows that the TDAOP protocol has a significantly higher computation cost, approximately 120 milliseconds, compared to the TDISP protocol, which has a lower computation cost of around 80 milliseconds. This suggests that while TDAOP might offer enhanced security or additional features, it does so at the expense of higher computational time. In contrast, TDISP is more efficient in terms of computation, making it a faster option, albeit potentially with trade-offs in other areas such as security or functionality.

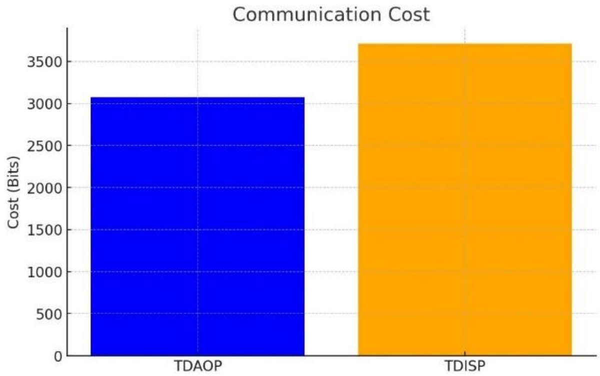

Figure 5. Comparison between TDAOP and TDISP concerning communication cost.

Figure 5 illustrates a comparison between the TDAOP and TDISP protocols in terms of their communication costs, measured in bits. The chart indicates that the TDISP protocol incurs a higher communication cost, approximately 3,500 bits, compared to the TDAOP protocol, which has a communication cost of around 3,000 bits. This suggests that the TDAOP protocol is more efficient in terms of data transmission, requiring less communication overhead than TDISP. The lower communication cost of TDAOP could be beneficial in scenarios where minimizing data exchange is critical, potentially leading to faster operations and reduced network load compared to TDISP.

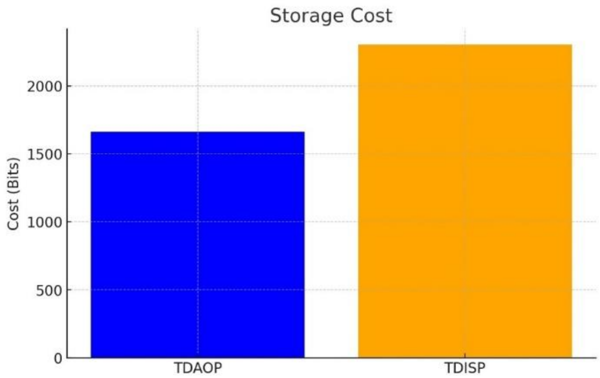

Figure 6. Comparison between TDAOP and TDISP concerning storage cost.

Figure 6 compares the storage costs of the TDAOP and TDISP protocols, measured in bits. The chart reveals that the TDAOP protocol has a lower storage cost, around 1,500 bits, whereas the TDISP protocol incurs a higher storage cost, approximately 2,000 bits. This indicates that TDAOP is more storage-efficient, requiring less space to store the necessary data compared to TDISP. The reduced storage cost of TDAOP can be advantageous in environments where storage resources are limited or where minimizing data footprint is crucial, making it a more efficient choice over TDISP in terms of storage requirements.

5. CONCLUSION

In conclusion, this paper presented a blockchain-based decentralized architecture designed to securely share event information among RSUs without relying on trusted third parties, such as cloud servers. The analysis began by reviewing the weaknesses identified in the TDISP scheme discussed earlier in the paper. To overcome these security challenges, the paper introduced an enhanced event-sharing and vehicle authentication protocol, leveraging the IPFS and blockchain technologies. The proposed protocol effectively stores event information in a distributed manner through IPFS and authenticates vehicles via blockchain. Additionally, the paper detailed the consensus mechanism used for generating the common key, which ensures robust security. The security analysis conducted confirms that the proposed protocol is resilient against potential attacks and meets essential security requirements. Furthermore, the paper compared the computation, communication, and storage costs of the new protocol against the previous scheme, demonstrating its superior efficiency. The performance evaluation clearly indicates that the proposed protocol is more capable and efficient than existing protocols.

References

[1] S. Pal and A. Islam, "Variation tolerant differential 8T SRAM cell for ultralow power applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 4, pp. 549–558, 2015.

[2] J. M. Dutertre, V. Beroulle, P. Candelier, S. De Castro, L.-B. Faber, M.-L. Flottes, P. Gendrier, D. Hély, R. Leveugle, P. Maistri, G. D. Natale, A. Papadimitriou, and B. Rouzeyre, "Sensitivity to laser fault injection: CMOS FD-SOI vs. CMOS bulk," *IEEE Transactions on Device and Materials Reliability*, vol. 19, no. 1, pp. 6–15, 2018.

[3] G. Torrens, S. A. Bota, B. Alorda, and J. Segura, "An experimental approach to accurate alpha-SER modeling and optimization through design parameters in 6T SRAM cells for deep-nanometer CMOS," *IEEE Transactions on Device and Materials Reliability*, vol. 14, no. 4, pp. 1013–1021, 2014.

[4] E. Ibe, H. Taniguchi, Y. Yahagi, K.-i. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule," *IEEE Transactions on Electron Devices*, vol. 57, no. 7, pp. 1527–1538, 2010.

[5] X. Liu, M. Mao, X. Bi, H. Li, and Y. Chen, "Exploring applications of STT-RAM in GPU architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 238–249, 2020.

[6] E. Lee, T. Han, D. Seo, G. Shin, J. Kim, S. Jeong, J. Rhe, J. Park, J. H. Ko, and Y. Lee, "A charge-domain scalable-weight in-memory computing macro with dual-SRAM architecture for precision-scalable DNN accelerators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 8, pp. 3305–3316, 2021.

[7] L. Lu and T. T.-H. Kim, "A high reliable SRAM-based PUF with enhanced challenge-response space," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 589–593, 2021.

[8] L. D. T. Dang, J. S. Kim, and C. I. Joon, "We-quatro: Radiation-hardened SRAM cell with parametric process variation tolerance," *IEEE Transactions on Nuclear Science*, vol. 64, no. 9, pp. 2489–2496, 2017.

[9] T. W. Oh, H. Jeong, K. Kang, J. Park, Y. Yang, and S.-O. Jung, "Power-gated 9T SRAM cell for low-energy operation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1183–1187, 2017.

[10] A. Sachdeva and V. Tomar, "A schmitt-trigger based low read power 12T SRAM cell," *Analog Integrated Circuits and Signal Processing*, vol. 105, no. 2, pp. 275–295, 2020.

[11] M. Wang, W. Lv, F. Yang, C. Yan, W. Cai, D. Zhou, and X. Zeng, "Efficient yield optimization for analog and SRAM circuits via Gaussian process regression and adaptive yield estimation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 10, pp. 1929–1942, 2017.

[12] N. Yadav and S. Jadav, "Efficient energy recovery in 9T adiabatic SRAM cell using body bias," *International Journal of VLSI and Embedded Systems*, vol. 5, pp. 778–784, 2014.

[13] J. Jiang, Y. Xu, W. Zhu, J. Xiao, and S. Zou, "Quadruple cross-coupled latch-based 10T and 12T SRAM bit-cell designs for highly reliable terrestrial applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 3, pp. 967–977, 2018.

[14] A. Yan, Y. Ling, J. Cui, Z. Chen, Z. Huang, J. Song, P. Girard, and X. Wen, "Quadruple cross-coupled dual-interlocked-storage-cells-based multiple-node-upset-tolerant latch designs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 3, pp. 879–890, 2020.

[15] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 305–316, 2005.

[16] P. E. Dodd, "Physics-based simulation of single-event effects," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 343–357, 2005.

[17] O. A. Amusan, A. F. Witulski, L. W. Massengill, B. L. Bhuva, P. R. Fleming, M. L. Alles, A. L. Sternberg, J. D. Black, and R. D. Schrimpf, "Charge collection and charge sharing in a 130 nm CMOS technology," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6, pp. 3253–3268, 2007.

[18] T. Heijmen, D. Giot, and P. Roche, "Factors that impact the critical charge of memory elements," in *Proc. IEEE International On-Line Testing Symposium (IOLTS'06)*, Italy, 2006, pp. 6–pp.

[19] A. Yan, Y. Chen, Y. Hu, J. Zhou, T. Ni, J. Cui, P. Girard, and X. Wen, "Novel speed-and-power-optimized SRAM cell designs with enhanced self-recoverability from single-and double-node upsets," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4684–4695, 2020.

[20] K. P. Rodbell, D. F. Heidel, J. A. Pellish, P. W. Marshall, H. H. Tang, C. E. Murray, K. A. LaBel, M. S. Gordon, K. G. Stawiasz, J. R. Schwank, M. D. Berg, H. S. Kim, M. R. Friendlich, A. M. Phan, and C. M. Seidleck, "32 and 45

nm radiation-hardened-by-design (RHBD) SOI latches," *IEEE Transactions on Nuclear Science*, vol. 58, no. 6, pp. 2702–2710, 2011.

[21] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, pp. 2874–2878, 1996.

[22] W. Chen, X. Guo, C. Wang, F. Zhang, C. Qi, X. Wang, X. Jin, Y. Wei, S. Yang, and Z. Song, "Single-event upsets in SRAMs with scaling technology nodes induced by terrestrial, nuclear reactor, and monoenergetic neutrons," *IEEE Transactions on Nuclear Science*, vol. 66, no. 6, pp. 856–865, 2019.

[23] S. Pal, S. Mohapatra, W. H. Ki, and A. Islam, "Design of soft-error-aware SRAM with multi-node upset recovery for aerospace applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2470–2480, 2021.

[24] J. Guo, L. Zhu, Y. Sun, H. Cao, H. Huang, T. Wang, C. Qi, R. Zhang, X. Cao, L. Xiao, and Z. Mao, "Design of area-efficient and highly reliable RHBD lT memory cell for aerospace applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 5, pp. 991–994, 2018.

[25] C. I. Kumar and B. Anand, "A highly reliable and energy-efficient process-variation-aware 12T SRAM cell design," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 1, pp. 58–66, 2019.

[26] S. Pal, S. Mohapatra, W.-H. Ki, and A. Islam, "Soft-error-aware highly reliable SRAM cell with enhanced multi-node-upset tolerance for aerospace applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 2353–2365, 2021.

[27] S. M. Jahinuzzaman, M. Sharifkhani, and M. Sachdev, "An analytical model for soft error critical charge of nanometric SRAMs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 9, pp. 1187–1195, 2009.

[28] S. M. Jahinuzzaman, D. J. Rennie, and M. Sachdev, "A soft error tolerant 10T SRAM bit-cell with differential read capability," *IEEE Transactions on Nuclear Science*, vol. 56, no. 6, pp. 3768–3773, 2009.

[29] A. Yan, Z. Wu, J. Guo, J. Song, and X. Wen, "Novel double-node-upset-tolerant memory cell designs through radiation-hardening-by-design and layout," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 354–363, 2018.

[30] R. C. Lacoe, "Improving integrated circuit performance through the application of hardness-by-design methodology," *IEEE Transactions on Nuclear Science*, vol. 55, no. 4, pp. 1903–1925, 2008.

[31] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 148–156, 2012.

[32] I. Alouani, W. M. Elsharkasy, A. M. Eltawil, F. J. Kurdahi, and S. Niar, "As8: static random access memory (SRAM): asymmetric SRAM architecture for soft error hardening enhancement," *IET Circuits, Devices & Systems*, vol. 11, no. 1, pp. 89–94, 2017.

[33] J. Guo, L. Zhu, W. Liu, H. Huang, S. Liu, T. Wang, L. Xiao, and Z. Mao, "Novel radiation-hardened-by-design (RHBD) 12T memory cell for aerospace applications in nanoscale CMOS technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1593–1600, 2017.

[34] C. Peng, J. Huang, C. Liu, Q. Zhao, S. Xiao, X. Wu, Z. Lin, J. Chen, and X. Zeng, "Radiation-hardened 14t SRAM bitcell with speed and power optimized for space application," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 2, pp. 407–415, 2018.

[35] M. S. M. Siddiqui, S. Ruchi, L. Van Le, T. Yoo, I.-J. Chang, and T. T.-H. Kim, "SRAM radiation hardening through self-refresh operation and error correction," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, pp. 468–474, 2020.

[36] S. Pal, D. D. Sri, W.-H. Ki, and A. Islam, "Soft-error resilient read decoupled SRAM with multi-node upset recovery for aerospace applications," *IEEE Transactions on Electron Devices*, vol. 68, no. 5, pp. 2246–2254, 2021.

[37] Mohammed Sameer, Mohd Abdul Faizan, Mohammad Altaf Hussain, Dr. Mohammed Abdul Bari, "Avoidance of Redundant Data In Cloud With Sha-512 Sheltered Approach ", Journal of Engineering Science (JES), ISSN NO:0377-9254, Vol 13, Issue 05, MAY/2022

[38] Farhan Ali Baig, Hajera Zia, Dr. Mohammed Abdul Bari, "Decentralised Social Media Platform Using Blockchain Technology", International Journal Of Research In Electronics And Computer Engineering (IJRECE), Vol. 10 Issue 2 April-June 2022; ISSN: 2393-9028

[39] Salwa Sayeedul Hasan, Mohamad Misbah Uddin Zia, Mohammed Shoeb Qureshi, Dr.Mohammed Abdul Bari, "Effect Of A Dynamic/Static Scan On The Response Times Of An Application Running On The Cloud", Journal of Engineering Science (JES), ISSN NO:0377-9254, Vol 13, Issue 05, MAY/2022