W. Lalouani, M. Younis and D. Tan, "Lightweight and Anonymity-preserving Secure Group Communication Mechanism for Cooperative Driving," 2023 32nd Wireless and Optical Communications Conference (WOCC), Newark, NJ, USA, 2023, pp. 1-5, doi: 10.1109/WOCC58016.2023.10139566.

https://doi.org/10.1109/WOCC58016.2023.10139566

Lightweight and Anonymity-preserving Secure Group Communication Mechanism for Cooperative Driving

Wassila Lalouani
Department of Computer and Information Science,
Towson University,
Towson, Maryland, USA
wlalouani@towson.edu

Mohamed Younis, Dayuan Tan
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore county,
Baltimore, Maryland, USA
younis@umbc.edu, dayuan1@umbc.edu

***Abstract*— Platooning is gaining much attention due to its potential for improving road safety, and increasing vehicular throughput. Given the required fine-grained coordination among the involved vehicles, resilience to cyberattacks is very crucial. Moreover, the information sharing among the vehicles should not be at the expense of the user's privacy. Additionally, the platoon operation is based on broadcast and requires a lightweight method to support secure group communication. This paper presents a novel protocol that utilizes lightweight hardware fingerprinting primitives and the Chinese Remainder Theorem (CRT) to automate the key generation and management process. Our protocol eliminates the need for pre-loaded keys and enables vehicles to infer the group key on-the-fly. CRT is utilized to help the transportation authority generate a group key for each platoon and broadcast an obscured version of such a key. Using our scheme, only the vehicles involved in platoon can recover the key using their respective hardware primitives. The validation results confirm the resilience of our protocol to attempts for unveiling the keys by single and collusive actors, while also providing reduced computational complexity compared to competing schemes.**

Keywords: Secure Group communication, PUF, Privacy, Platoon, Chinese reminder theorem, Intelligent transportation systems.

I. INTRODUCTION

Intelligent Transportation Systems (ITS) are deemed as a vital part of smart cities. ITS offers numerous benefits such as improved safety, energy efficiency, and user convenience. In essence, smart vehicles can partially/fully self-drive, collect data from sensors, and make decisions based on close coordination among vehicles [1]. Such a new technology relies on two main concepts: (1) cooperative adaptive cruise control. and (2) Vehicular Ad hoc Networking (VANET). A vehicle can share various information such as location, speed, and acceleration, etc., with the road infrastructure to facilitate traffic management. Typically, a road-side unit (RSU) is employed to interface a vehicle with the traffic authority. In addition, the RSU can play a role in traffic flow optimization, e.g., by dynamic signal time adjustment. Using Vehicle-to-Vehicle (V2V) communication, a vehicle can also communicate with a wide range of different nearby vehicles that work cooperatively to travel in an orchestrated manner. Platooning is a way to closely coordinate the motion of a group of vehicles, which fundamentally decrease of inter-vehicle spacing, which boosts road capacity and vehicular throughput [2].

Despite the advantages of platooning, it exposes the networked vehicles to cyberattacks. Attackers may be external eavesdroppers or malicious actors that exploit the inter-vehicle communication to access sensitive information, inject corrupted data to causing unsafe driving conditions [1]. The attacker can also use the captured information about the vehicular platoon or platoon members to prevent additional members from joining. Asymmetric cryptographic techniques are usually used to ensure the safety and secrecy of information exchange. Yet, these techniques do not protect against traceability attacks that allow correlating vehicle positions and consequently disclosing the user's travel path. Data is broadcasted to all platoon members and hence secure group communication needs to be supported. Specifically, lightweight group management is needed to support the dynamic nature of platoons and cope with the limited computational resources, especially with no stable connection to a trusted authority.

This paper proposes a novel secure and privacy-preserving intra-platoon communication (SePIP) protocol to achieve data confidentiality, maintain vehicle anonymity, prevent travel path traceability, and ensure data integrity. SePIP employs a lightweight hardware-based fingerprinting primitive, namely, a Physically Unclonable Function (PUF), for authenticating vehicles and establishing an encryption key for data sharing within a platoon. A trusted authority (TA) relies on the uniqueness and unclonability of PUFs to validate the vehicle's identity prior to joining or forming a platoon. To support secure group communication within a platoon, the TA generates a session key that is a function of the ID of the RSU in the vicinity. CRT is used to securely share the session key with only the platoon members. By leveraging the properties of PUF and CRT, the TA obscures the session key such that only the vehicles within the platoon recover the key using their respective PUF and by using just a single modulo operation. Such a key extraction process is very lightweight and suits the resource constrained vehicles On-board units (OBUs). The simplicity of the key generation process enables SePIP to efficiently support dynamic platoon membership and key management under varying platoon configurations. The privacy of the communicating parties is preserved as the session keys are independent of the identity and the location of vehicles. The session key is varied across RSU's coverage areas to counter trajectory tracking. Furthermore, SePIP eliminates the need for pre-loading a master key onto the vehicle's which alleviate any tampering attacks. SePIP's is validated using FPGA-based PUF implementation and is shown to be resilient

to attacks by a single or multiple collusive actors. The computation complexity is also shown to be much lower than competing schemes. The contribution summary is as follows:

- Develop a novel protocol that enables vehicles to securely share data and exchange messages for cooperative driving.
- Leverage the properties of PUFs and CRT to ensure key secrecy, efficiently handle dynamic platoon membership, and reduce the computational complexity.
- Achieve data confidentiality and integrity while sustaining user privacy and protecting against vehicle tracking and contemporary cyberattacks.
- Validate SePIP using vehicular network simulator and an FPGA-generated dataset. In addition, the security properties of SePIP are verified using AVISA [3].

II. Related Work

Given the scope of this paper, we focus on existing methods for secure data sharing in VANET. PKI-based digital signature schemes have been the popular means for supporting data confidentiality, authentication, and non-repudiation [4]. The asymmetric cryptographic nature of PKI makes it quite robust. However, PKI requires each vehicle to store a large number of certificates. Also, certificate verification imposes significant computational overhead. Employing symmetric cryptographic primitives avoids such overhead, where for all group members uses the same key. Group key management schemes can be classified into: centralized, decentralized, and distributed [5]. Centralized schemes rely on a server, and hence do not scale well. In VANETs the server may not be reachable at all times.

Decentralized and distributed schemes are more practical. For example, Wong et al. [6] proposed a tree-based scheme for optimize rekeying. SePIP employs distributed group management schema without assuming trusted RSUs. Beyond the key management architecture, distinct group key generation has been applied in the literature [7]. Identity-Based Group Key Generation [8] is a cryptographic technique used to generate a shared secret key among a group. This cryptographic technique is useful for setting up secure communication channels between multiple users; yet it does not sustain privacy. To ensure security and privacy, some schemes use shared secrets that are accessible through trusted RSUs, something that SePIP does not assume. Additionally, some schemes rely on dynamic secret sharing which imposes significant communication overhead due to the vehicle's motion speeds.

Some work uses hardware-based primitives to generate keys and authenticate nodes [9]. However, most existing approaches focus on PUF modeling attack mitigation rather than group key management. Moreover, CRT has been leveraged for group key management [11]. VijayaKumar et al. [10] proposed a CRT-based group key distribution scheme that optimizes the computation complexity in IoT. Compared to the existing approaches, SePIP leverages hardware fingerprints and CRT to enable the control platoon membership, access to group keys, and preserve user's privacy.

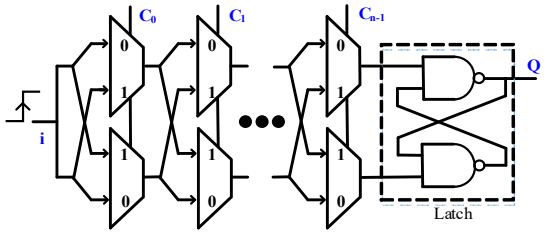

Fig. 1. Schematic diagram of an Arbiter PUF, where the challenge bits control the individual multiplexers and cause the input signal to experience different delays on distinct devices and consequently the latched value (Q) would differ.

III. System Model and Preliminaries

A. Attack Model

The main objectives of an adversary are to: (1) disrupt the normal operation of a platoon, and (2) uncover the identity of platoon members to track their trajectory. To achieve these objectives, two attack scenarios are considered. First, the attacker may intercept the transmitted messages to gain unauthorized access to intra-platoon data exchange. Basically, platooning relies heavily on wireless communications among vehicles which make them prone to passive attacks, e.g., eavesdropping, or active attacks such as message replay, and impersonation. The second scenario involves hacking vehicles and/or RSUs, either individually or in a collusive manner to uncover device secrets that are used for data encryption. Additionally, some vehicles may collude with an RSU to reveal the identity and travel path of a target platoon member.

B. Physically Unclonable Functions

SePIP calls for the incorporation of a PUF in the OBD of each vehicle. A PUF design is founded on the typical variations among integrated circuits that often happens during fabrication. These variations are unintended and uncontrollable, and hence become specific for each device [9]. A PUF circuit exploits the potential variations to define a hardware-based fingerprint of the underlying device. Fig. 1 shows the arbiter PUF, which leverages the difference in the delay experienced by an input signal when propagating through the sequence of multiplexers until reaching the arbiter. The multiplexers are configured using a set of bits, referred to as the challenge. In Fig. 1, the challenge, $c_0, c_1, \ldots, c_{n-1}$, configure the multiplexers and determine the path that the input signal traverses, and consequently the latched value (the PUF response). Since the delay is random in nature, the response for the same challenge bit string is device specific. In other words, a PUF cannot be cloned and its challenge-to-response mapping constitutes a device signature. Even hacking a device will not reveal such a signature. However, intercepting some challenge-response pairs (CRPs) and applying machine learning (ML) techniques make the PUF subject to modeling attacks [13]. SePIP alleviates such vulnerability to modeling attacks by using the algebraic properties of CRT. To generate m response bits, either the PUF circuit is replicated or is queried m times. To simplify the presentation, we will assume that PUF maps n challenge bits to m response bits.

C. Solution Strategy

To counter the aforementioned attacks while coping with the limited computation resources of an OBD and the scale of a VANET, it is critical to establish secure communication among platoon members through a lightweight secret key generation process. Moreover, key sharing with vehicles should not reveal important information that allows traceability of their travel

paths and thus violates the user's privacy. SePIP strives to fulfill the needs and achieves the following design goals:

1. *Data integrity and freshness*: The communication mainly supports decision making in ITS. Malicious manipulation could thus risk user safety. Thus, the exchanged message should be subject to forward secrecy provisions.
2. *Physical protection*: An OBU may be hacked by an adversary. Thus, the device secret should not be stored.
3. *Impersonation and collusion resistance*: A vehicle identity should not be falsified to participate in the platoon decision making or enable illegitimate data retrieval that involves individual or multiple vehicles and RSUs.
4. *Travel path anonymity*: a vehicle should not be tracked by eavesdroppers, other vehicles, or any manipulated RSU.

The design strategy for SePIP is based on embedding a PUF in the OBU and pursuing a multi-level key management architecture. A TA, e.g., the department of motor vehicles, is employed to conduct registration, facilitate vehicle authentication, and generate session keys. The registration process is done using a secure communication channel or by physical access to the vehicle by the authority, e.g., when issuing the title and tag. In the registration, a vehicle V_x shares a set $\Gamma_x(C_x, R_x)$ of CRPs of its PUF; such a set is to be stored at the TA and is assumed to be safeguarded. The TA uses Γ_x to: (i) authenticate the vehicle; the TA picks a challenge bit- string at random and query the vehicle for the corresponding response. (ii) obfuscate the keys during transmission; CRT is used to share the session key with the platoon members where each vehicle will use its own CRP to extract the key. The RSU is not playing a security role and simply acts as a relay to ensure vehicle reachability to the TA. Communication between the RSU and TA is often wireline-based and is secure. The TA also varies the platoon key and assigns a distinct vehicle pseudo-identifier (PID) based on the RSU coverage area in order to mitigate any potential RSU compromise and ensure privacy. SePIP is explained in detail in the next section.

IV. DETAILED SEPIP DESIGN

A. Vehicle Authentication and Trip Setup

When starting a trip, each vehicle, V_x, needs to inform the TA about the planned travel route. Upon contacting the TA, possibly through the RSU, V_x is authenticated as follows:

1. The TA randomly picks a challenge C_i^x and sends the following message to V_x: $\{C_i^x, [Nonce]_{R_i^x}\}$, where $(C_i^x, R_i^x) \in \Gamma_x$, *Nonce* is a randomly generated bit-string that is then encrypted using R_i^x.
2. Upon receiving the message, V_x applies C_i^x to its PUF to generate R_i^x and decrypts the message to extract the *Nonce*.

We note that there is no need for V_x to confirm successful extraction of the *Nonce* since using a wrong *Nonce* will deprive V_x from being part of the ITS network. V_x will then inform the TA about its travel path, which consists of a set of road segments, P_x. The message will be encrypted using R_i^x, which implicitly confirms the identity of V_x. Based on the route, the TA determines the RSUs that V_x will interact with and note them in the trip record. Based on the number or encountered RSUs, k, the TA will use the *Nonce* as a seed to a pseudo random number generator (PRNG) to determine a sequence of k challenge bit-strings for V_x. We denote such a sequence by, $C_1^x, C_2^x, \ldots, C_k^x$. Such a sequence is neither shared with any RSU nor sent back to V_x. SePIP requires each vehicle to have the same PRNG that the TA uses. With successful extraction of the *Nonce*, V_x should thus be able to regenerate the same sequence of challenges. It is important to note that an RSU does not know any of these challenges and cannot thus associate them with V_x. In addition, the RSU does not have the TA's PRNG and does not know the *Nonce* that the TA provided to V_x.

Because of the unique process variation of integrated circuits, each CRP of a PUF will act as a vehicle fingerprint. Only the challenge bit-string is used as a vehicle's PID and hence the real vehicle ID is not revealed to the RSU and other vehicles to sustain the user's privacy. To prevent a compromised RSU from being exploited to track a vehicle, SePIP requires that a distinct CRP, and consequently PID, is used for V_x at different RSUs. Therefore, V_x will switch between $C_1^x, C_2^x, \ldots, C_k^x$ when it encounters a different RSU, where V_x will send a message to the TA, probably through the RSU, with the new challenge. By using the same PRNG and seed, V_x will be synchronized with the TA. Hence, the TA will be able to match the new challenge and the RSU ID to uniquely identify V_x. We note that with a relatively long challenge bit-string, e.g., 32 or 64 bits, the probability that two vehicles will have the same challenges within the coverage area of one RSU is extremely low. Consequently, V_x stays unidentifiable by the RSU, and cannot be tracked. In other words, SePIP prevents tracing V_x's trip by any eavesdropper, its neighboring vehicles, or malicious RSUs and hence sustains the user's privacy.

B. Group Key Generation

To achieving confidentiality and support data sharing within a platoon, SePIP employs a dynamic process for the generation and management of data encryption keys. Such a process ensures forward secrecy, enables quick rekeying, and imposes little overhead. Specifically, the TA generates a group key and sends it to the platoon member through the RSU in an obscured form. Only legitimate vehicles, i.e., platoon members, are able to retrieve the key using their hardware fingerprints. The idea is to exploit the properties of CRT to restrict key retrieval through a system of modulo equations. CRT states that if we know the remainder of the Euclidean division of an integer δ by several coprime integers, then we can determine uniquely the remainder of the division of δ by the product of these integers. In the context of SePIP, the product of the PUF responses of the individual vehicles is used for such a purpose. Since the PUF response can be generated by the specific device and is not known to any other vehicle, the key can be inferred only by a platoon member. The following explains the detail of the key generation and retrieval process.

Given a platoon formed by Q vehicles $V_1, \ldots V_Q$ within the coverage range of RSU_k, the TA uses the responses $R_{\mathrm{k}}^1, \ldots . R_{\mathrm{k}}^Q$, to generate a group key for the platoon members. As noted earlier, $R_{\mathrm{k}}^1, \ldots . R_{\mathrm{k}}^Q$ reflect the PUF output of the individual the platoon members to $C_{\mathrm{k}}^1, \ldots . C_{\mathrm{k}}^Q$ which are generated by TA during trip registration of each platoon member. Since $R_{\mathrm{k}}^1, \ldots . R_{\mathrm{k}}^Q$ are not necessarily coprimes, the TA first finds the

closest prime number for each response. Let $\ddot{R}$ be the closest prime number to the value of R. Then, the TA computes:

$$x_i = \frac{\partial_g}{\ddot{R}_i}, \text{ where } \partial_g = \prod_{i=1}^{Q} \ddot{R}_i \quad (1)$$

Then, the TA finds an integer y_i such that:

$$x_i . y_i \equiv 1 \; mod \; \ddot{R}_i \quad (2)$$

The TA chooses a random key λ, and defines:

$$\gamma = \lambda . \sum_{i=1}^{Q} x_i . y_i \quad (3)$$

where γ is the obfuscated version of the platoon key. The TA then sends γ to the RSU_k which in turn passes it to the platoon members. The key can be recovered by a vehicle V_i using:

$$\lambda = \gamma \; mod \; \ddot{R}_\iota \quad (4)$$

Specifically, V_j uses the PRNG to generate C_{k}^{j} and applies to its own PUF to find R_{k}^{j}. The latter is then used to get $\ddot{R}_j$ before applying eq. (4). To elaborate:

$$\left(\lambda . \sum_{i=1}^{Q} x_i . y_i\right) mod \; \ddot{R}_j = \lambda (x_1 . y_1 + \cdots + x_n . y_n) \; mod \; \ddot{R}_j$$

$$= \lambda \left(\frac{\partial_g}{\ddot{R}_1} . y_1 \; + \cdots + \frac{\partial_g}{\ddot{R}_j} . y_j + \cdots + \frac{\partial_g}{R_Q} . y_Q \right) mod \; \ddot{R}_i$$

$$= \lambda (0 + \cdots + 1 + \cdots + 0 \;) = \; \lambda$$

Thus, SePIP does not reveal the vehicle's CRPs to either RSUs or other vehicles. Finally, we note that picking a large *m*, i.e., size of PUF response, the number of all possible prime numbers will be very large and hence the probability of launching a brute force attack by explore all prime numbers to extract the key will be quite low, especially with the frequent change of the key. Also, the value of *m* is unknown and may vary across vehicles.

C. Platoon Establishement and Management

When a platoon is formed the RSU will inform the TA about the PID of each member vehicle. Recall that the PID for V_j in the coverage area of RSU_k is in essence C_{k}^{j}. Hence, the TA will have to use both RSU_k and C_{k}^{j} to identify V_j since there may be other vehicles in the ITS network that use the same challenge bit-string, which by itself is not unique (only the combined challenge and response is deemed as a signature). After identifying V_j, the TA will find R_{k}^{j} for each platoon vehicle and generate the group key as detailed in the previous subsection. If more vehicles join the platoon at a later time, the TA will share the key depending on the number of these vehicles. We note that the current key, λ, will not change in this case, and just needs to be securely shared with the new platoon members. If only one vehicle, V_r, joins, the key can be encrypted using R_{k}^{r}, and sent to RSU_k to be relayed to V_r. On the other hand, if multiple vehicles are to be added, the steps in the previous subsections are followed with the exception of generating λ.

On the other hand, when a vehicle leaves the platoon, the TA must update the key to ensure that the departing member cannot access the exchanged data in future intra-platoon communications, i.e., preserving forward secrecy. The update simply reflects the generation of a new key where the departing vehicle will be excluded from Eq. (1), (2) and (3), above, which deprives it from knowing the new key. Generally, the key update requires special attention since it could be susceptible to message replay attacks. Basically, an attacker could replay an old key sharing message to allow a former member to have the group key. Therefore, we make a refinement to the steps in Eq. (3) and (4) by adding a signature that defines the version of the key. Specifically, we add the following signature to the message that provides γ to the platoon members:

$$Sig_k = [\lambda_{k,v-1}, RSU_k]_{\lambda_{k,v}} \quad (5)$$

Sig_ksimply includes the previous group key that is shared by RSU_k to confirm the key freshness to the platoon member. The variable v reflects the version of the key, where $\lambda_{k,0}$ will be simply none. The new key, $\lambda_{k,v}$, is used to encrypt the signature and hence a former member cannot create a new key for the platoon. A replayed key sharing message will be thus detected.

V. Validation Experiments

We evaluated the effectiveness of SePIP using Omnet++ on a 1000m ×1000m map. We used a CRP dataset collected from an arbiter-PUF implemented in Xilinx ARTIX-7 FPGA. The inter-vehicle distance was set to 2m. We considered: (i) an eavesdropper that targets the communication links among vehicles, and between vehicles and RSUs, (ii) malicious vehicles exploiting their previous keys generation mechanism, and (iii) multiple collusive vehicles and RSU from different platoon. Performance is assessed in term of modeling attack accuracy and the computational overhead due to the key management scheme. We have compared SePIP to PKI. In the simulation, we consider a dynamic platoon setting, where a platoon is formed based on the vehicle proximity. When some vehicles become close to each other in the same direction, the RSU invites them to form a platoon and informs the TA to apply SePIP. When another vehicle gets close and travels the same direction, RSU requests it to join the platoon. We note that the platoon formation in practice could be different, e.g., triggered by the vehicles; yet it is not within the scope of SePIP.

We have verified the security properties of SePIP using a formal verification framework, namely, AVISPA [3], which assesses the vulnerability to active and passive attacks. We have defined all players, namely, the TA, RSUs and vehicles, and described the vehicle enrollment, and key and data exchange protocols using the High-Level Protocol Specification Language (HLPSL). We have specified all the steps and defined the security goals in terms of key secrecy and vehicle authentication. The environment role involves multiple sessions of vehicles, and one session for the RSU, TA and an intruder. Fig. 2 shows the output confirming that SePIP is safe.

To assess the modeling accuracy, we have used a multi-layer neural network (NN) with 3 layers, where a set of γ is fed as an input and λ is observed as an output. The NN is trained using 4,000 messages, unless otherwise specified. Tables 1-3 compare the performance of SePIP in terms of modeling accuracy. Table 1 studies the effect of the number of vehicles in the platoon, i.e., the platoon size on the accuracy. We also compare versions of SePIP with different key sizes. The results clearly show that SePIP guarantees the unpredictability of the key for all platoon sizes. Even with decreased key sizes, SePIP stays robust. For large keys, e.g., 32 bits or more, the modeling accuracy is extremely low due to the large number of possible keys. Yet, SePIP sustains robustness for small keys where the accuracy of predicting 8-bit keys does not exceed 15%.

We have also studied SePIP's resilience against collusive attacks where multiple vehicles in the ITS networks collaborate

to predict the platoon key. The attacking vehicles themselves are aware of how SePIP operates; yet they do not know the underlying fingerprints of the platoon members, i.e., the CRP of each vehicle that the TA used during the key establishment process. Table 2 captures the effect of varying the key size, and the number of colluding vehicles, which reflects the scope of the attack. The results clearly confirm that SePIP's robustness is not impacted much by the group key size. This is attributed to the randomness introduced by the PUF of the platoon participants. As indicated by the results, SePIP is not affected by collusion where the accuracy stays almost flat regardless of the growth in attack scope. The most relevant observation here is that a key size of 8 bits could suffice for countering collusion.

Table 3 assesses the modeling accuracy under increased message exchange. Here an eavesdropper is assumed to be capturing γ to train the NN model. The increase in the number of messages will thus grow the training set. As indicated by the results, SePIP is not impacted by the training set size as λ is randomly selected. Finally, Table 4 compares the runtime of SePIP and PKI. The number of vehicles is varied between 100 and 800; we have observed that about 50% of them join a platoon and hence apply either SePIP or PKI. In SePIP, the computational overhead is insignificant due to the very low complexity of modulo operations. Our results indicate that SePIP is 1000 times faster than PKI.

VI. Conclusions

This paper has presented, SePIP, a secure privacy-preserving group communication protocol for ITS. SePIP employs PUFs and the Chinese remainder theorem to generate and manage group keys. The PUF response to a certain challenge bit pattern is used to not only ensure the vehicle user's anonymity but also to introduce randomness in the group key management process. SePIP also leverages the advantages of PUFs in terms of tamper-resistance and low overhead. To effectively distribute the key to group members, SePIP relies on the properties of CRT. The validation results have confirmed that SePIP is resilient to ML-based modeling attacks conducted by a single or multiple collusive vehicles. We have also shown that SePIP is lightweight and is 1000 faster than PKI.

Reference

[1] A. Lamssaggad, N. Benamar, A. S. Hafid and M. Msahli, "A Survey on the Current Security Landscape of Intelligent Transportation Systems," *IEEE Access*, vol. 9, pp. 9180-9208, 2021.

[2] V. Lesch, M. Breitbach, M. Segata, C. Becker, S. Kounev and C. Krupitzer, "An Overview on Approaches for Coordination of Platoons," *IEEE Trans. on Intelligent Transportation Sys.*, 23(8), pp. 10049-10065, 2022.

[3] Armando, A. et al. "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," *Lecture Notes in Computer Science*, vol 3576. Springer, Berlin, Heidelberg, 2005

[4] A. Wasef, Y. Jiang, and X. Shen, "ECMV: Efficient certificate management scheme for vehicular networks," *Proc. IEEE Global Telecom. Conf.*, 2008.

[5] M. Ge, K.-K. R. Choo, H. Wu, and Y. Yu, "Survey on key revocation mechanisms in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 63, pp. 24-38, 2016.

[6] C. K. Wong, M. Gouda and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. on Networking*, 8(1), pp. 16-30, 2000.

[7] T. Prantl, et al., "A Survey on Secure Group Communication Schemes With Focus on IoT Communication," *IEEE Acc.*, vol.10, pp.99944-99962, 2022.

[8] S.-F. Tzeng, et al., "Enhancing security and privacy for identity-based batch verification scheme in VANETs," *IEEE Trans. Veh. Technol.*, 66(4), pp. 3235–3248, 2017.

[9] A. Shamsoshoara et al., "A Survey On Physical Unclonable Function (PUF) Based Security Solutions for Internet of Things," *Computer Networks*, 183, pp. 107593, 2020

[10] P. VijayaKumar, S. Bose, and A. Kannan, "Centralized key distribution protocol using the greatest common divisor method," *Comput. Math. Appl.*, 65(9), pp. 1360–1368, 2013.

[11] X. Zou, B. Ramamurthy, and S. S. Magliveras, "Chinese Remainder Theorem Based Hierarchical Access Control for Secure Group Communication," *Proc. of the 3rd Int'l Conf. on Info. and Comm. Security (ICICS '01)*, Springer-Verlag, Berlin, Heidelberg, 381–385, 2001

[12] P. Sharma, and B.R. Purushothama, "BP-MGKM: An efficient multi-group key management scheme based on bivariate polynomial," *Computer Networks*, vol. 216, pp. 109244, 2022,.

[13]W. Lalouani, and M. younis, "Collusion-resistant, Lightweight and Privacy-preserving Authentication Protocol for IoV," *Proc. IEEE Consumer Comm. and Networking Conf. (CCNC)*, Las Vegas, NV, 2023.

Figure 2: Results of the formal verification using AVISPA

Table 1: Accuracy of modeling attack of the platoon key.

	Modeling accuracy				
# Vehicles	10	20	30	50	100
SePIP-8bits	0.1225	0.1325	0.15125	0.11	0.12125
SePIP-16	≈ 0.06625	≈ 0.07125	≈ 0.06	≈ 0.06875	≈ 0.0575
SePIP-32	0.0275	0.035%	0.0325%	0.0275%	0.03%
SePIP-128	≈0	≈0	≈0	≈0	≈0

Table 2: Effect of multi-vehicle collusion on key modeling accuracy

	Modeling accuracy				
# Vehicles	2	4	5	7	8
SePIP-8bits	≈ 0.1187	≈ 0.1175	≈ 0.1487	≈ 0.125	≈ 0.1162
SePIP-16	≈ 0.0625	≈ 0.055	≈ 0.0562	≈ 0.0662	≈ 0.0675
SePIP32	0.0325	0.0437	0.025	0.0412	0.0362
SePIP-128	≈0	≈0	≈0	≈0	≈0

Table 3: Effect of message count on the modeling attack accuracy

	Modeling accuracy			
#messages	500	1000	2000	4000
SePIP-8bits	0.1	≈0.115	≈0.1133	≈ 0.1162
SePIP-16	=0.05	≈0.0950	≈0.073	≈ 0.0675
SePIP32	0.02	0.04	0.013	0.0362
SePIP-128	≈0	≈0	≈0	≈0

Table 4: Runtime of key management as a function of the network density

	Latency Overhead (s)			
#vehicle	100	200	500	800
SePIP	0.008	0.035	0.157	0.323
PKI	10.4	39.1	195.4	393.1