future internet

MDPI

Article

Threshold Cryptography-Based Secure Vehicle-to-Everything (V2X) Communication in 5G-Enabled Intelligent Transportation Systems

Nuwan Weerasinghe [1], Muhammad Arslan Usman [1], Chaminda Hewage [2,*], Eckhard Pfluegel [1] and Christos Politis [1]

[1] Faculty of Science, Engineering and Computing, Kingston University London, London KT1 2EE, UK; nuwan.weerasinghe@kingston.ac.uk (N.W.)

[2] Cybersecurity and Information Networks Centre (CINC), Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff CF5 2YB, UK

* Correspondence: chewage@cardiffmet.ac.uk

Abstract: Implementing 5G-enabled Vehicle-to-Everything (V2X) intelligent transportation systems presents a promising opportunity to enhance road safety and traffic flow while facilitating the integration of artificial intelligence (AI) based solutions. Yet, security and privacy concerns pose significant challenges that must be addressed. Therefore, researchers have focused on improving the security and integrity of vehicle data sharing, with a particular emphasis on V2X application layer security and privacy requirements. This is crucial given that V2X networks can consist of vehicles manufactured by different companies and registered in various jurisdictions, which may only be within communication range for a few seconds. Thus, it is necessary to establish a trusting relationship between vehicles quickly. The article proposes a threshold cryptography-based key exchange protocol that meets the key requirements for V2X data sharing and privacy, including the rapid establishment of trust, the maintenance of vehicle anonymity, and the provision of secure messages. To evaluate the feasibility and performance of the proposed protocol, a tailored testbed that leverages the NS-3 network simulator, a commercial 5G network, and public cloud infrastructure is used. Overall, the proposed protocol provides a potential solution for addressing security and privacy concerns in V2X networks, which is essential for successfully implementing and adopting this technology.

Keywords: V2X; VANET; threshold cryptography; 5G; 3GPP sidelink; cloud computing; computational efficiency; cyber-physical systems

check for updates

Citation: Weerasinghe, N.; Usman, M.A.; Hewage, C.; Pfluegel, E.; Politis, C. Threshold Cryptography-Based Secure Vehicle-to-Everything (V2X) Communication in 5G-Enabled Intelligent Transportation Systems. *Future Internet* **2023**, *15*, 157. https://doi.org/10.3390/fi15050157

Academic Editor: Ming-Chin Chuang

Received: 17 March 2023
Revised: 17 April 2023
Accepted: 20 April 2023
Published: 23 April 2023

1. Introduction

Vehicle-to-Everything (V2X) is an intelligent transportation system [1] which interconnects vehicles, cycles, pedestrians, and roadside infrastructure. The V2X is an evolution of Vehicular ad hoc Networks (VANETs), which aim to provide connectivity among vehicles on roads and nearby fixed equipment. The main aim of the V2X network is to enhance traffic safety by providing up-to-date information to relevant authorities, drivers, and pedestrians. Furthermore, as part of an intelligent transportation system, the V2X network aims to improve road safety, boost traffic flow, reduce congestion, and deliver multimedia content to roadside users. In recent years, the advancement of artificial intelligence (AI)-based solutions such as self-driving vehicles and machine learning (ML) based traffic management has widened the requirement of V2X network. The communication type depends on the equipment involved with communication and is broadly categorized as Vehicle-to-Sensors (V2S), Vehicle-to-Vehicle (V2V), Vehicle-to-Pedestrian (V2P), Vehicle-to-Grid (V2G), Vehicle-to-Cloud (V2C), and Vehicle-to-Infrastructure (V2I) [2].

A V2X network consists of vehicles fitted with wireless communication devices called onboard units (OBUs). Each OBU contains a hardware security module, a tamper-proof device for storing security information. The OBU on a vehicle communicates with a roadside unit (RSU), roadside users (RU) such as pedestrians and cyclists, as well as other vehicles (OBUs) [3].

Earlier VANET networks were proposed ad hoc Wi-Fi networks and were standardized in IEEE 802.11p [4]. This standard enables mobile nodes to communicate in a dynamic topology with intermittent connection, supporting non-line of sight. Although this technology can be deployed with the minimum investment, it lacks scalability, quality of service (QoS), and has limited coverage due to short radio changes [5].

Fifth generation network-based V2X is promised to provide low latency, high reliability, and high bandwidth. Moreover, 5G-based V2X can operate as a cellular-based communication [6] where V2X devices acting as user equipment (UE) communicate with the 5G Next Generation NodeB (gNB) for uplink and downlink connectivity. Furthermore, 5G Core can provide authentication and network resource management, while cloud hosted V2X applications can provide V2X specific functionality. On the other hand, Device-to-Device communication (D2D), defined in 3GPP side-link [7], allows two UE's (two vehicles or vehicle to RSU) to communicate without passing data through the 5G core network.

Security is a critical challenge in V2X communication. The data are transmitted over insecure wireless channels. They can be readily intercepted or tampered with by an attacker, thereby resulting in severe threats to the safety and privacy of vehicles, e.g., illegal tracking or remote hijacking. Further, they are susceptible to unauthorized vehicles sending falsified data to disrupt the standard transmission of data and providing incorrect messages to cause traffic mismanagement or even a severe traffic accident. Thus, improving the security and integrity of vehicle data sharing has become a primary focus for researchers.

The IEEE 802.11p-based network can use IEEE P1609.2 [8], which defines a certificate-based cryptographic algorithm to secure transaction messages between two entities and broadcast messages that do not direct to a particular entity. On the other hand, the 5G-based V2X can use the existing authentication mechanism of the 5G network [9–11]. This generally addresses physical layer to network-layer security.

However, there is still a gap in V2X application layer security and privacy requirements. At any given time, a V2X network can have vehicles manufactured by different manufacturers and even registered in different countries or jurisdictions. While the RSU part of V2X remains static, vehicles are continuously joining and leaving the network. Especially in V2V communication, depending on their speed and direction of travel, they may be in the communication range for a few seconds. Therefore, it is required to create a trusting relationship between vehicles quickly.

Data privacy is of utmost importance in V2X data sharing, as vehicles must share accurate information about vehicle and road conditions while maintaining anonymity. Additionally, different countries may have varying data privacy laws. To address these concerns, we have summarized the key requirements for V2X data sharing, particularly for AI-driven applications, as follows:

- Establishing trust quickly between vehicles and roadside units;
- Maintaining anonymity of vehicle identity while providing a means for validating messages and identifying/tracking misbehaving vehicles;
- Ensuring secure messages are only accessible by intended users;
- Implementing a threshold cryptography system where no single intermediary can reveal secret information, but a group of authorized entities can access it.

The primary contributions of this research work include proposing a novel threshold cryptography-based key exchange protocol for V2X networks to address the outlined data sharing requirements. This proposed protocol employs a threshold secret-sharing scheme that integrates a certificateless scheme, which is based on the approach developed by Saxena et al. This approach has been discussed in a series of publications [12–16] for a certificateless key management scheme using verifiable secret sharing.

Additionally, the feasibility of the proposed protocol was evaluated in a tailored hybrid V2X network testbed, leveraging the NS-3 network simulator, a commercial 5G network, and public cloud infrastructure. In this testbed, the cloud platform and 5G network were implemented in actual environments, while the RSUs and vehicles were simulated.

Finally, this research work provides recommendations for future research and development in V2X security systems.

2. Related Work

Cho et al. [17] discussed the various security challenges in V2X communication, including authenticity, confidentiality, integrity, availability, and privacy. It emphasizes the need for secure key management, secure message exchange, and secure vehicle authentication to mitigate these challenges. Garcia-Saavedra et al. [18] discussed the security challenges and opportunities of 5G-enabled V2X communication. They emphasize the need for secure and reliable communication, including confidentiality, integrity, and authenticity, to ensure user privacy and safety. Therefore, this section further investigates various secure key management approaches and their weaknesses.

Key management is a crucial aspect of any cryptographic security scheme. It encompasses creating, distributing, updating, and removing cryptography keys [19]. Secure communication is established by exchanging keys through an insecure channel or using pre-existing keys. Asymmetric encryption techniques necessitate the sharing of a public key, which is done through a trusted third-party entity known as a Certificate Authority (CA) that is recognized by all parties involved in the data communication.

Symmetric encryption algorithms utilize a shared secret key for encryption and decryption [20]. This feature makes symmetric encryption algorithms faster than asymmetric encryption algorithms. However, the drawback is that it can be challenging to establish the symmetric key without a secure channel in V2X networks.

In addition, the parties need to develop a framework of trust relationships to authenticate the ownership of the keys before starting a secure communication. The public key infrastructure (PKI)-based scheme [21–23] is commonly used in V2X networks to validate the ownership of keys [24]. However, these PKI-based systems require a trusted authority (TA) to store the certificates for all registered vehicles. Nevertheless, if the anonymity of vehicles is needed, the vehicle will have more than one certificate. This increases the number of certificate TA must store, which increases storage overheads, and makes it difficult to search for a certificate in a sizeable V2X network.

Trust frameworks are classified as a centralized trusted third-party or fully distributed trust. However, due to dynamic and temporary relationships between vehicles, the centralized approach is not practical or secure in networks like V2X networks. This has been discussed in detail in [25,26]. Therefore, recent research has tried to find distributed trust solutions. Integrity and authenticity of the key is achieved using digital signature and hash functions. In a centralized system, a public key can be signed by a certificate authority trusted by both parties. Then, the receiving party can validate the public key sent via an unsecure communication channel. In distributed systems, threshold cryptography-based algorithms can be used.

Boneh and Franklin [27] proposed a bilinear pairing scheme that assumes the key generator is trusted. However, using a unique device ID can make it challenging to update keys and requires a secure channel for private key transfer. Self-certified public keys [28] combine PKI with the identity-based public key, which is generated by using the unique node id and a random number. Then, the node broadcasts its witness values so others can use witness values and unique id to calculate the public key. However, this approach is not scalable, requiring a reliable broadcasting protocol and large storage to store witnesses. Certificateless public key schemes are variants of ID-based schemes [29,30], which allow networks to initialize with threshold-based secret sharing schemes so that the master private key is agreed upon among the nodes. A new node can join the network by contacting a threshold number of nodes. Nodes can calculate public keys by using publicly available information, hence avoiding the requirement of the certificate authority.

Shamir Secret Sharing [31] is an early information protection threshold cryptographic scheme based on polynomial interpolation. This allows the dealer to distribute partial secrets $(s_1, s_2, \ldots\ s_n)$ of secret S to n parts such that minimum $t < n$ is required to reconstruct secret S. This assumes the dealer is trusted, generates valid partial secrets, and ensures that partial secrets can be securely communicated to a node. This scheme does not reveal any information to fewer than t participants and minimizes share size. In addition, it allows update/change shares while maintaining the secret or changing the threshold. However, in this method, there is no way to validate the partial secret share received by an entity is a valid partial secret. The Verifiable Secret Sharing (VSS) schemes, first introduced by Zhou and Haas [32], and then, by Chor et al. [33], were further developed by Feldman [34]. They enable participants to validate shares received from a semi-trusted dealer and detect any malicious changes during the transfer.

Identity-Based public key management schemes utilize a node's unique identity as a part of its public key, which could be software-based such as IP address and email or hardware-based Physical Unclonable Functions (PUFs) [35,36]. This eliminates the need for a certificate to validate the public keys. Shamir presents a scalable solution [37] for ID-based cryptography, which includes an ID-based encryption scheme where the sender encrypts a message using a public key derived from the recipient's unique identity. The receiver then decrypts the message using a private key obtained from a key generation center (KGC), which corresponds to the recipient's identity.

Nitesh Saxena proposed an ID-based certificateless public key framework [14] based on standard (discrete logarithm) assumptions. This scheme is based on Feldman's VSS scheme [34] and was discussed earlier in the section. Dealers use VSS to create partial shares and commitments, which act as a private group key. The generated partial share is securely transferred to devices, and it will become the device's private key. This will be the foundation of the proposed key exchange protocol, and these concepts will be explored further later in this work.

3. System Model

As illustrated in Figure 1, there are four entities in the V2X network: local and cloud-based trusted authority (TA), 5G base station (5G-BS), roadside units (RSU), and vehicle (V). First, we describe the functions of these entities.

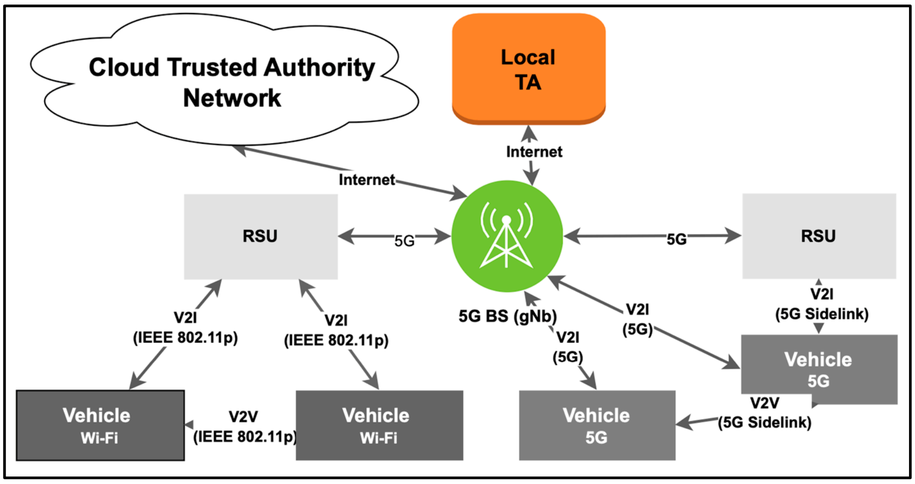


Figure 1. System model of V2X network.

Trusted authority (TA): In the proposed V2X network architecture, we defined three types of TAs. Local TA is overall responsible for the V2X network and the authentication of RSUs. Each vehicle belongs to a particular TA called Home TA, which holds all information about the vehicle and can validate and authenticate the vehicle's identity. In real life, Home TA could be a country, state, or council vehicle registration authority. When the network is initialized, the Local TA will connect with all available Home TAs to form the Cloud TA network. Third-party TA is in a vehicle prospective is all Cloud TA's, excluding Home and Local ones. TAs have high computational and storage capacity and can be connected via low-latency secure networks. In an actual situation, these could be achieved using any public cloud environment such as AWS or Azure. All the TAs initialize themself with secret and commitment coefficients, while global cryptographic parameters, described in Table 1, will be the same for all TAs. RSUs and vehicles must receive partial share and commitment coefficients from all TAs to be securely enrolled on the V2X network.

Table 1. Global cryptographic parameters.

Parameter	Description
T	This is the threshold value users need to supply based on specific security requirements.
p	This should be a large prime number ($p > 100$).
q	This is a factor of p-1 which satisfies p-1 is divisible by q.
g	This is a generator for a group $\mathbb{G}$, if the group of elements $\{ g^0, g^1, g^2, \ldots. \}$ is precisely the group $\mathbb{G}$; that is, every element $h \in \mathbb{G}$ can be expressed as $h = g^i$ for some i, and conversely, for every i, $g^i \in \mathbb{G}$.

Fifth generation base station (5G-BS): 5G-BS provides a communication link to connect RSU, and 5G enables vehicles to TAs and V2X applications. This is also responsible for providing authentication services to 5G-enabled vehicles to use side links, i.e., V2V communication.

Roadside units (RSU): RSU provide connectivity to vehicles which do not have 5G. Vehicles and RSU can communicate with ad hoc Wi-Fi using IEEE 801.11 P. In addition, RU and Vehicle provide multi-hop communication, which enables V2V multi-hop communication. Additionally, RSU can help vehicles without 5G to securely enroll to the V2X network and applications.

Vehicle (V): This is the main entry that consumes the V2X application and generates a vast amount of data that must be securely transferred to other vehicles, cloud-based data analytic platforms, and AI and ML algorithms. The vehicles can have only Wi-Fi, only 5G, or both, and they can participate in V2V and V2I communication.

3.1. Network and Security Assumptions

Vehicles are already registered with the Home TA, and the vehicle and the Home TA can establish a secure communication link. In practice, this can be achieved via PKI and secure network technologies such as VPN, TLS/SSL, and HTTPS. Moreover, new vehicles entering the V2X network area have a way of identifying V2X network details, which can be achieved by broadcasting network information from RSU, especially RSUs in the edge of the V2X coverage area. An example can be from the tall plaza of a highway for motorway entries and entry locations of a city.

The V2X network provides a way to communicate with the Home TA for new vehicles. Fifth generation-enabled vehicles can directly connect to the Home TA via the 5G network, and Wi-Fi-only vehicles will use a RSU to connect to the Home TA.

It is assumed that each vehicle carries a unique identity UID, which is based on Physical Unclonable Functions (PUFs) [35,36] that are unique and tamper-resistant during their lifetime.

The work of Saxena used the random oracle model [38] to validate the cryptographic scheme, and the security of the scheme is based on the computational Diffie–Hellman (CDH) assumption [39]. That relies on the problem of computing the discrete logarithm in cyclic groups. Consider a cyclic group $\mathbb{G}$ of order q The CDH assumption states that, given g, g^a, g^b for a randomly chosen generator g and random $a,\ b\ \in \{0, \ldots., \ q-1\}$, it is computationally intractable to compute the value g^{ab}. The CDH assumes this cannot be solved quickly with current computational power. However, if computing the discrete logarithm (base g) in $\mathbb{G}$ is straightforward (such as availability of a supercomputer or quantum computer), the CDH problem can also be solved effortlessly.

3.2. Network Initialization

The threshold-based secret sharing scheme proposed by Saxena can be initialized by a single trusted authority (TA) or a group of TAs. However, the drawback of using a single TA is that if an adversary can access the TA and retrieve the network's private key, this will result in the whole network being compromised. An alternative to this is using multiple TAs working independently and providing multiple partial shares to a joining vehicle. Then, the combined partial share will be the vehicle's private key, while the group private key is not known to any individual.

The local TA decides the global cryptographic parameters shown in Table 1 and selects the cloud TAs required in the V2X network. The selection of cloud TAs will be based on the type of vehicles you want to support in the V2X network; at the minimum, all vehicles' Home TAs should be part of cloud TAs. Then, upon sharing global cryptographic parameters, each TA (i) initializes itself by generating a large random number as its secret (a_{i_0}) with generated witness values as shown in Equations (1)–(7).

Let the number of dealers be d and define

$$f_1(x) = a_{1_0} + \sum_{n=1}^{t-1} (a_{1_n} x^n) \bmod p \tag{1}$$

$$f_2(x) = a_{2_0} + \sum_{n=1}^{t-1} (a_{2_n} x^n) \bmod p \ldots \tag{2}$$

$$f_d(x) = a_{d_0} + \sum_{n=1}^{t-1} (a_{d_n} x^n) \bmod p \tag{3}$$

$$f(x) = \sum_{n=1}^{d} f_n(x) \tag{4}$$

$$W_1 = \{\, g^{a_{1_0}},\, g^{a_{1_1}}, \ldots, g^{a_{1_t}} \,\} \tag{5}$$

$$W_2 = \{\, g^{a_{2_0}},\, g^{a_{2_1}}, \ldots, g^{a_{2_t}} \,\} \tag{6}$$

$$W_d = \{\, g^{a_{d_0}},\, g^{a_{d_1}}, \ldots, g^{a_{d_t}} \,\} \tag{7}$$

Then, TAs are ready to enroll vehicles to the V2X network. The network-wide private key is unknown to anyone. Using threshold cryptography, a collaboration of any T number of TAs could compute the network-wide private key. A summary of the network's cryptographic information is shown in Table 2.

Table 2. Summary of network cryptographic information.

Cryptographic Information	Notation	Description
Scheme information	$\{\, p, q, g, T \,\}$	Public information available to everyone
Network Private Key (N_{prk})	$a_{1_0} + a_{2_0} + \ldots + a_{d_0}$	No one has this information; collaboration of T nodes can compute
Network Public Key (N_{pk})	$W_1 + W_2 + \ldots + W_d$	Public information available to everyone
Commitment coefficients (W)	$\{\, W_1, W_2, \ldots W_d \,\}$	Public information available to everyone

PUF could be used to generate a unique id associated with a vehicle. However, direct use of PUF-generated unique id Equation(8) will break the anonymity and the uniqueness of the message in the network.

$$\text{UID}_i = PUF(i) \tag{8}$$

where i is the node number and *PUF* is Physical Unclonable Function

Hence, it is proposed that each vehicle generates a self-identity using a one-way hash algorithm such as SHA-256 (H) with a network name (N) and current time (T) as parameters Equation (9).

$$\text{ID}_i = H\ (\ \text{UID}_i + \text{N} + \text{T}) \tag{9}$$

3.3. Join the Network

When the vehicle enters the V2X network, it will receive a broadcast message from RSU with network information. If the vehicle is 5G-enabled, it would directly send Join Request to Home TA with network information. If the vehicle does not have 5G connectivity, it will use Wi-Fi ad hoc to connect to RSU and send the Join request via RSU. The initial interaction between the joining Vehicle and the V2X network is shown in Figure 2.

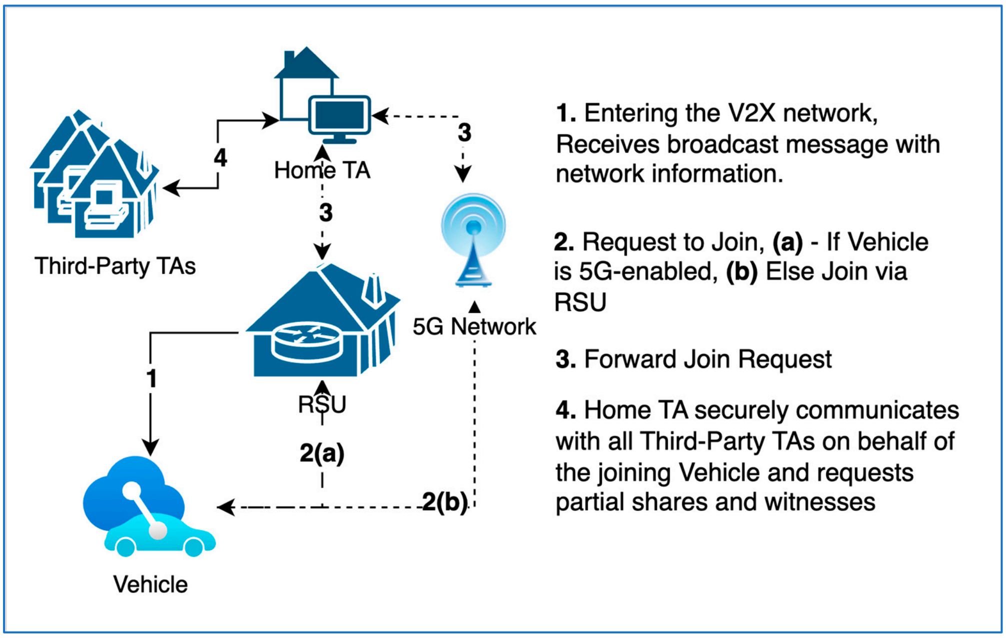

Home TA
3
4
Third-Party TAs
3
5G Network
1
RSU
2(a)
2(b)
Vehicle
1. Entering the V2X network, Receives broadcast message with network information.
2. Request to Join, **(a)** - If Vehicle is 5G-enabled, **(b)** Else Join via RSU
3. Forward Join Request
4. Home TA securely communicates with all Third-Party TAs on behalf of the joining Vehicle and requests partial shares and witnesses

Figure 2. Initial interaction between the joining vehicle and the V2X network.

Home TA exposes the Join request functionality via REST API over HTTPS. Since the vehicle is registered with Home TA, it can authenticate the join request. In practice, this can be done using any API security mechanism. However, we proposed to use a PKI-based authentication scheme. Then, the Home TA securely communicates with all third-party TAs on behalf of the joining vehicle and requests partial shares and witnesses. Communication between cloud TAs could be via REST API over HTTPS and could be done in an internal cloud network. This enables the secure transfer of partial secrets to the Home TA. Then, the Home TA can respond to the joining vehicle with all partial shares and witness sets. Since communication between the vehicle and the Home TA is via HTTPS, we propose that it does not need any extra encryption. However, V2X networks such as emergency services or the military, where a higher level of security is required for the Join response, could be encrypted using any asymmetric encryption algorithm. Figure 3 shows the sequence diagram of a vehicle joining the network via an RSU using home TA, local TA, and n number of third-party TAs.

Let TA_i i = 0 . . . n, where n is number of trusted authorities and TA_0 is the Home TA. JV is a new vehicle that wants to access the V2X network. The vehicle joining algorithm is shown in Algorithm 1.

```
Algorithm 1 REST API based network join
JV: Join request (JV)
When a Vehicle (JV) enters the V2X network, it receives a broadcast message from RSU with
network information. Then, it creates a Join request (JR) using the following steps:
Step 1: JV creates its id ($\mathrm{ID}_{jv}$) using Equations (8) and (9)
Step 2: JV creates Join request ($\mathrm{JOIN}_{\mathrm{REQ}jv}$)with $\mathrm{ID}_{jv}$. An example JSON Join request is shown
below.
{ "id": ID_jv, "networkId": <V2X network Id> }
Step 3: JV sends $\mathrm{JOIN}_{\mathrm{REQ}jv}$ to the $TA_0$ (Home TA) as a HTTPS POST request.
Home TA: Join response.
Home TA calls REST API provided by all third-party TAs on behalf of the joining vehicle and
request partial shares and witnesses. An example JSON partial share request is shown below.
{ "id": ID_jv, "networkId": <V2X network Id>}
Step 4: Home TA sends partial share request to Each $TA_i$ as a HTTPS POST request.
Step 5: Upon receipt a partial share request, each $TA_i$ calculates partial share ($Prk_{jn_i}$) using
Equation (3).
Step 6: Then, each $TA_i$ sends a JSON response to the Home TA with partial share and witnesses.
An example of a JSON partial share response is shown below.
{ "id": ID_jv, "share": $Prk_{jn_i}$, "witnesses": [$g^{a_{i_0}}$, $g^{a_{i_1}}, \ldots, g^{a_{i_t}}$] }

Step 5: Upon receipt of all partial share responses, the Home TA constructs a Join response
($\mathrm{JOIN}_{\mathrm{RSP}jv}$). An example of a JSON Join response is shown below.

{ "id": ID_jv,
" shares": [
{ "share": $Prk_{jn_1}$, "witnesses": [$g^{a_{1_0}}$, $g^{a_{1_1}}, \ldots, g^{a_{1_t}}$],
"share": $Prk_{jn_2}$, "witnesses": [$g^{a_{2_0}}$, $g^{a_{2_1}}, \ldots, g^{a_{2_t}}$],
. . .
"share": $Prk_{jn_i}$, "witnesses": [$g^{a_{i_0}}$, $g^{a_{i_1}}, \ldots, g^{a_{i_t}}$],
}
]
}
Step 6: Send $\mathrm{JOIN}_{\mathrm{RSP}jn}$ to JV.
JV: Join response processing.
Step 7: For each $Prk_{jn_i}$ in $\mathrm{JOIN}_{\mathrm{RSP}jn}$, verify a validity of partial share received using verifiable
secrete sharing, using Equation (10).

    $$g^{Prk_{jn_i}} = \prod_{j=1}^{t} w_j^{ID_i^j} \ mod \ p \qquad (10)$$

    Step 8: If verification is successful, JV adds all the partial secret to calculate its private key
$ID_{jv\,prk}$ using Equation (11).

    $$ID_{jv\,prk} = Prk_{jn_0} + Prk_{jn_1} + \ldots + Prk_{jn_{d_\mathrm{i}}} \qquad (11)$$
```

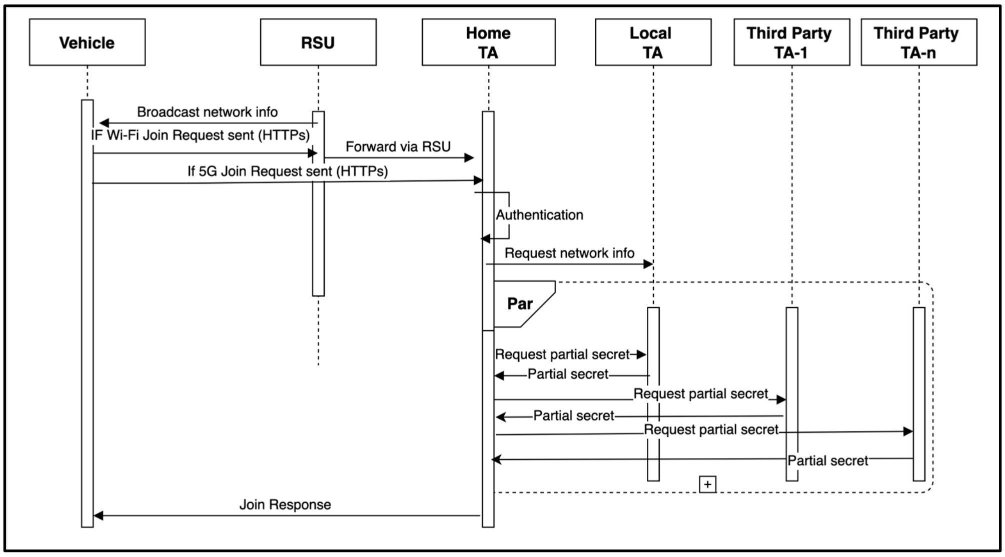


Figure 3. Join the network.

After successfully joining the V2X network, the vehicle computes its private key for this network using Equation (11). In addition, it can calculate the public key (Pk_{ov}) of any other vehicle (OV) in the V2X network by using OV's Id. The network information as shown in Equation (12).

$$Pk_{ov} = \sum_{k=0}^{d} \Big(\prod_{j=1}^{t} w_{k_j}^{ID^i_{ov}} \bmod p \Big) \tag{12}$$

3.4. Key Establishment

Any two entities joined to the V2X network can establish a shared key (Session Key), which can then be used with any symmetric encryption algorithm. Suppose A and B want to establish a session key. Given that A knows about the unique id of B (ID_b), it calculates the public key of B (Pk_b). Given that B knows about unique id of A (ID_a), it calculates the public key of A (Pk_a). Then, A calculates session key (S_{ab}) using Equation (13), and B also calculates the session key (S_{ba}) using Equation (14). Here, pri_a and pri_b represent the partial secret shares of A and B, respectively.

$$S_{ab} = {Pk_b}^{Pri_a} = g^{Pri_b \cdot Pri_a} \bmod p \tag{13}$$

$$S_{ba} = {Pk_a}^{Pri_b} = g^{Pri_a \cdot Pri_b} \bmod p \tag{14}$$

From this, we confirm $S_{ab} = S_{ba}$. Hence, given two entities, the shared key or session key can be established without any interaction.

On the other hand, a broadcast encryption scheme requires a key which the broadcaster and all receiving vehicles can compute. Assume that vehicle J wants to broadcast a message. The message can be received by any other vehicle or RSU within the wireless range. In our broadcast encryption scheme, the message is encrypted using its private key (Prk_j). Then, any legitimate entity could compute the corresponding public key Pk_j using network information already received when joining the network. This allows any message broadcast from a legitimate entity to be decrypted by any other legitimate entity.

In practice, a random session key (PS_j) was generated to improve the scheme's efficiency, and the message was encrypted using the AES algorithm with the PS_j as an encryption key. Then, the PS_j will be encrypted using the ElGalmal encryption algorithm with Prk_j as encryption key. Once an entity receives an encrypted message, the decryption key (DK_j) is calculated in Equation (15) as follows.

$$DK_j = \prod_{i=1}^{t} w_i^{UID_j^i} \bmod p \tag{15}$$

A potential drawback of this approach is that if a legitimate entity is compromised, an intruder can retrieve network information through that entity. The intruder would be able to eavesdrop on every broadcast message. However, it will not be able to act as a legitimate entity and send messages since it did not receive a partial secret from the V2X network. Hence any message encrypted by the intruder will not be able to be decrypted by legitimate entity.

4. Simulator Design, Performance Evaluation and Results

As discussed in the system model, our proposed V2X network required cloud servers and computation, a 5G network, static RSU with ad hoc Wi-Fi, and moving vehicles with 5G/Wi-Fi ad hoc access. Building a simulator for all of them is complicated and time-consuming. Additionally, our proposed changes are only at RSU, vehicle and cloud-based servers. Therefore, we decided to set up our simulator as a hybrid system: the cloud platform and 5G network parts use the actual implementation, and RSU and Vehicles are in simulated environments. The system architecture of the proposed system is shown in Figure 4 and described in the next subsection.

4.1. System Architecture

The NS-3 simulator is a network simulator used primarily for research and education in Wi-Fi, LTE, and 5G. It is written in C++; however, the network simulation script can be written in either C++ or python. It consists of a collection of core libraries called NS-3 modules, developed within an open-source project. It is licensed under the GNU GPLv2 license [40].

NS-3 simulator has a 5G NR millimeter wave module [41] and a module for the V2X network [42]. It can simulate a V2X network with different mobility models, routing protocols and wireless communication technologies. Additionally, NS-3 supports hybrid devices where the lower part of the network stack is simulated while the upper layers can be implemented in a Linux container. In our simulator, RSU and vehicles were implemented in NS-3 with a hybrid 5G network interface using the tap bridge module [43] of NS-3. In this hybrid 5G interface, one side is connected to NS-3 and the other end is connected to a commercial 5G network via a Linux container. Using a commercial 5G network restricts the capability of our simulator to the application's layer protocol evaluation. However, it was enough for the scope of this research.

TA functionality is developed as Microsoft.net Core REST API and hosted in the AWS Cloud platform. To demonstrate the different locations of TAs, we hosted TAs in different AWS regions. Client functionality of key exchange and other cryptographic functionalities were developed as a .net Core application and hosted in a Linux container, and it is connected to NS3 simulated Wi-Fi ad hoc network.

4.2. Performance Evaluation of Network Initialisation

Our proposed solution should contain at least two TAs, i.e., Local TA and Home TA. As all TAs are cloud-based and can be initialized in parallel, we evaluated the initialization process of a single TA only. The time taken for network initialization is influenced by the size of the private key, the threshold value T, and the computational power of the machine executing the process. However, we did not assess different cloud servers, as server optimization is beyond the scope of our study and computational power can be easily increased in cloud computing. Thus, we conducted our simulation using AWS's minimum available cloud server (t3a.nano) [44].

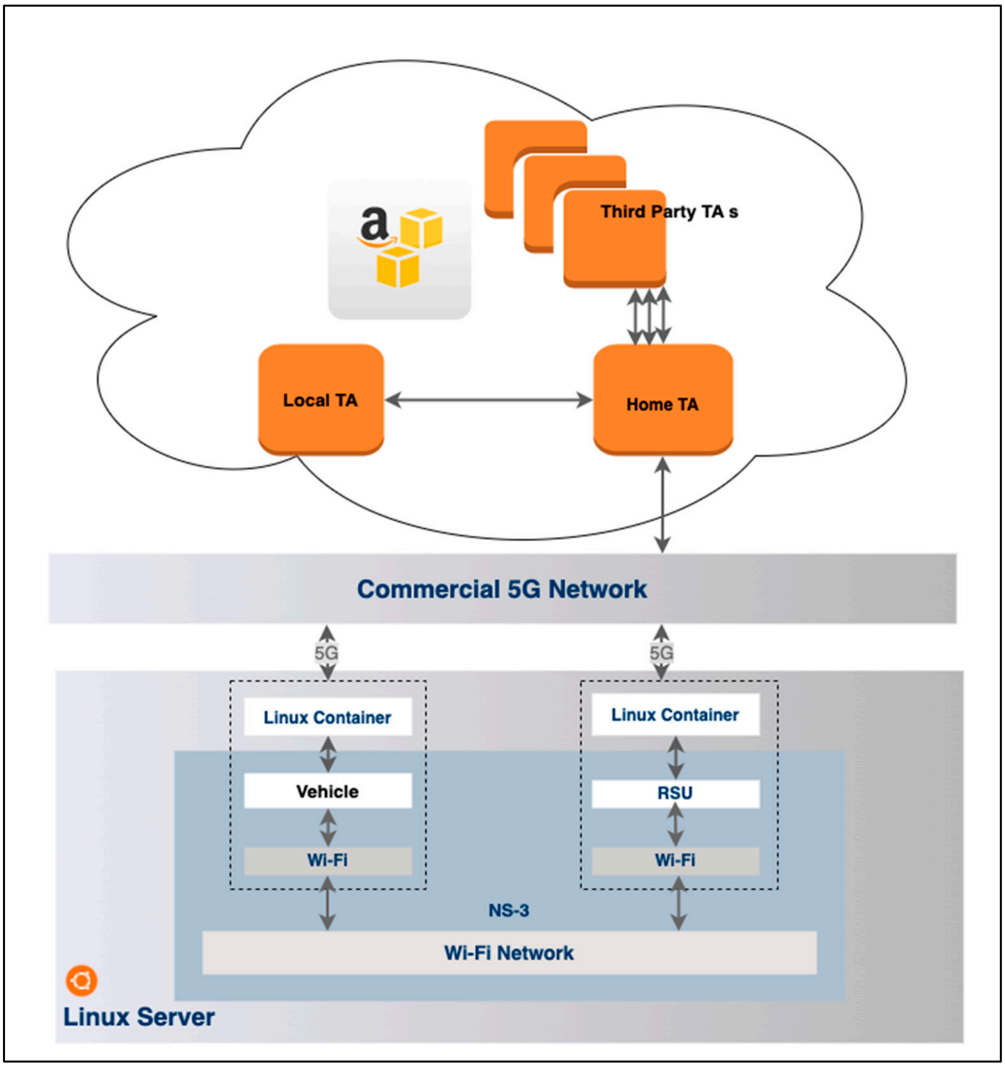


Figure 4. Proposed system architecture.

As described in Section 3.2, the network initializes by generating a large random number as its secret, along with randomly generated witness values, as illustrated in Equations (1)–(7). We utilized a range of simulation parameters, as shown in Table 3, and each configuration was executed 10,000 times, with the average time taken for initialization recorded. We simulated T values from 3 to 12 and private key sizes of 64 to 512. We kept the size of P and Q as 64-bit and the ID size as 32-bit for the entire simulation. Figure 5 depicts the network initialization time per TA for various private key sizes and threshold values.

Table 3. Range of simulation parameters.

Parameter	Value
Threshold value (T)	3 to 12
Large prime number (P)	64-bit random number
Large prime number (Q)	64-bit random number
Private Key—Secret (a_{i_0})	64–512-bit random number
Vehicle Id	32-bit random number
Commitment coefficients $\left(a_{1_1}, a_{1_2} \ldots a_{1_{t-1}}\right)$	8-bit random number

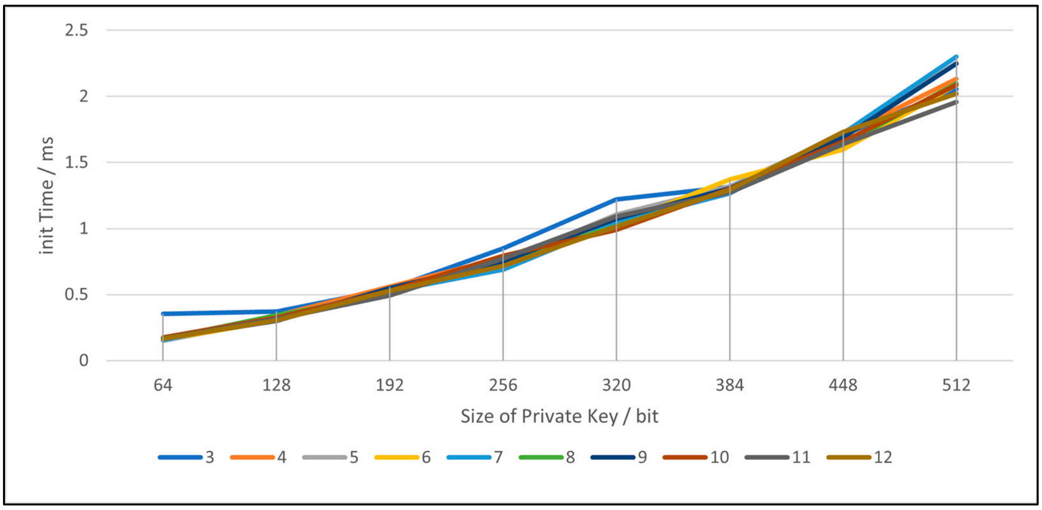


Figure 5. Network initialization time by the size of the private key and threshold number.

The result shows that with the increase of private key size, initialization time increases linearly. However, even for a larger private key size of 512, it only takes 2–3 milliseconds. Considering network initialization is one operation, it is possible to use larger private key sizes without considerable overhead while significantly increasing the network's security. In addition, notice that the threshold value T does not have much impact on network initialization. Therefore, network designers should try to use large private key sizes as well as large T values.

4.3. Performance Evaluation of Vehical Joining a V2X Network

Once the vehicle receives shares and witnesses, it must be verified as described in Equation (12). Share verification is a critical security condition in a distributed threshold-based secret-sharing scheme and is computationally intensive. However, our proposed system model defined a vehicle as having a pre-existing trust relationship with their Home TA. Therefore, we defined two verification types, client-verify, where verification is done in the vehicle, and server-verify, where the Home TA inside the cloud server does verification. Vehicle joining time will include the V2X network delay, combined share calculation time of all TAs, and network delay between the Home TA and other TAs. We used a stationary 5G-connected vehicle simulated in NS-3 in our V2X network to evaluate this and tried to join the network using two verification methods. First, we ran a simulator with a different number of TAs from 2 to 12 and a threshold value of 3 to 12. We also varied private key sizes from 64-bit to 512-bit with 64-bit increments. Other parameters are kept the same as stated in Table 3. Each configuration was run 100 times. The share calculation time, verification time, V2X network delay, and cloud network delay were measured.

Figure 6 shows the breakdown of joining time for both verification types. Here, we exclude V2X network delay and cloud network delay in the figure. The impact of communication delay is evaluated in the next section. This concludes that using server verification reduces verification time by 15%. This was performed using the least powerful AWS server. Increasing computational power in cloud computers is easy; therefore, server verification time can be further reduced by using powerful computers available in the cloud. Hence, it reduces the overall time that the vehicle required to join the V2X network.

Figure 7a shows the impact of the number of TAs on the total join time calculation for both verification types, while Figure 7b shows the impact based on the threshold value. In both cases, with an increase of TAs and threshold value, verification time is increasing. However, if it shows using server verification, we can reduce the overall verification time.

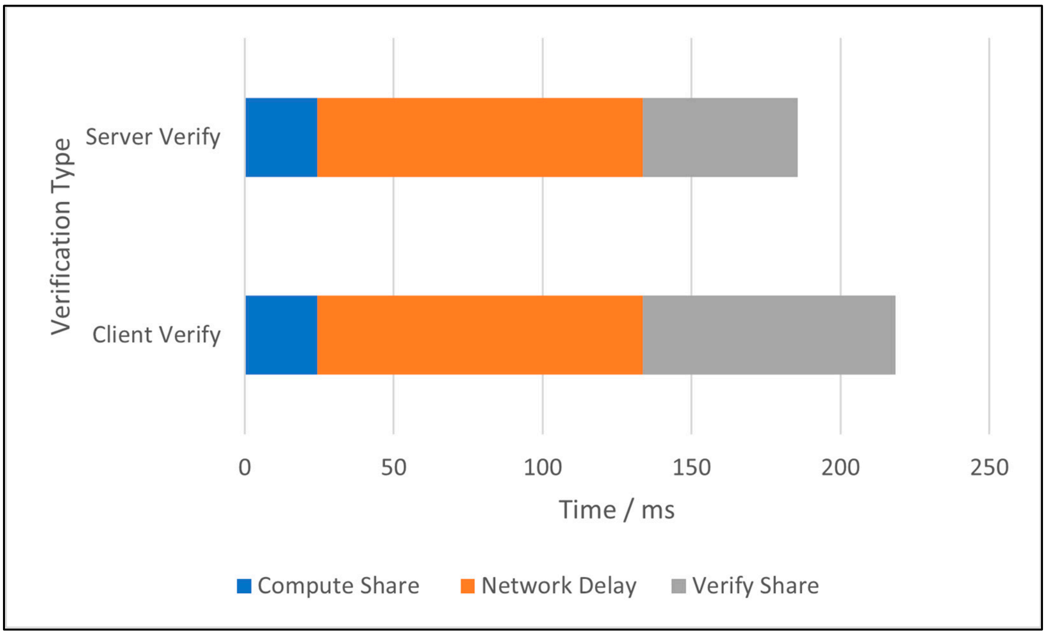
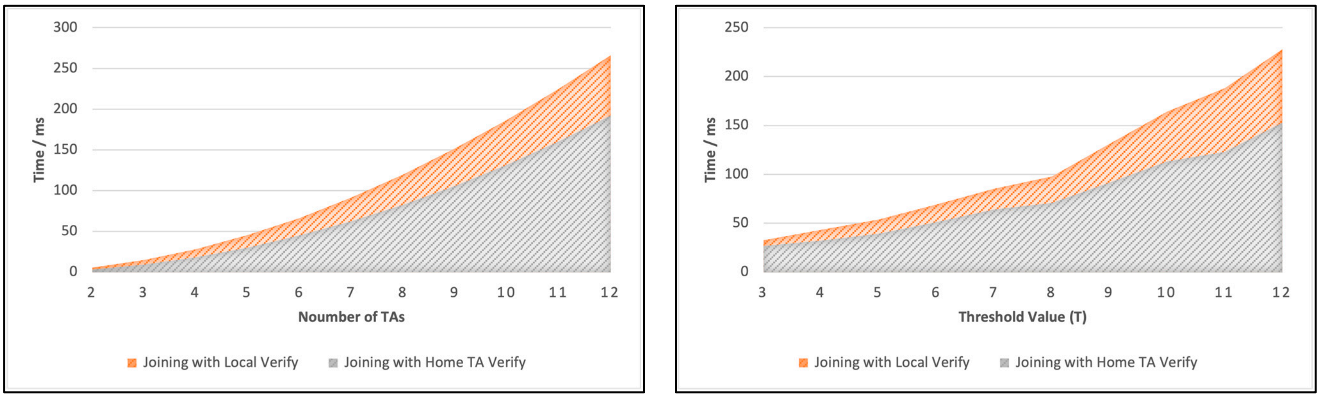


Figure 6. Breakdown of joining time.



(**a**) (**b**)

Figure 7. Total join time calculation for both verification types; (**a**) impact of the number of Tas; (**b**) impact based on the threshold value.

Another critical parameter to evaluate is the time an entity takes to calculate another entity's public key using Equation (13). For example, Figure 8 shows the time taken to compute a public key increase with the number of TAs and threshold values. Therefore, we need to select different parameters based on network performance and security requirements.

4.4. Communication Overhead of Joining V2X Network

As mentioned earlier, communication overhead occurs in two different locations: in V2X, where 5G-enabled vehicles connect to the Home TA via 5G network; or Wi-Fi-only vehicles which connect to the Home TA via RSU. In this case, vehicle to RSU communication will be via Wi-Fi, and RSU to Home TA is via 5G. Other network overhead occurs when the Home TA tries to communicate with the local TA and other third-party TAs. To evaluate this, we used 5G-enabled vehicles and Wi-Fi-only vehicles in the V2X network, where vehicles were randomly placed in a grid of 1000 m by 1000 m area and moving with a constant speed of 20 m/s using the NS-3 random walking module. We ran a simulator with 5 TAs, a threshold value of 8, and private key sizes of 256-bit. In addition, the size of P and Q were kept as 64-bit and the ID size was kept as 32-bit for the entire simulation. The simulation ran for 1000 s. With each second, RSU broadcasts network information, and the vehicle tried to join the network upon receiving the broadcast. We recorded V2X network delay as well as cloud network delay. In this simulation, server delay is the delay between maximum end-to-end delay between local TA and other TAs. Local TAs call all other TAs (REST API) parallelly, so we measured the average maximum delay as server delay. The V2X delay is the communication (network) delay between joining the vehicle and the local TA.

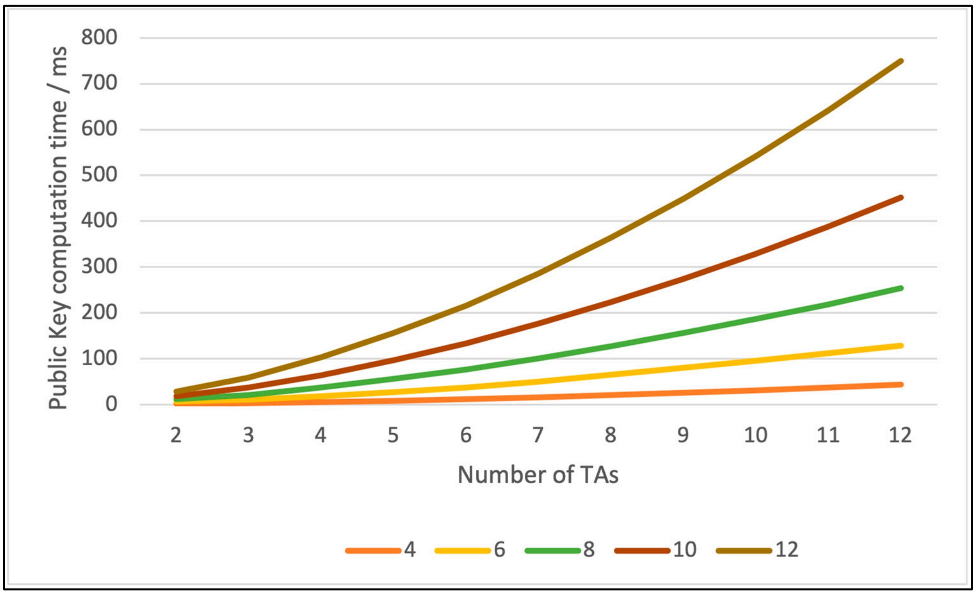
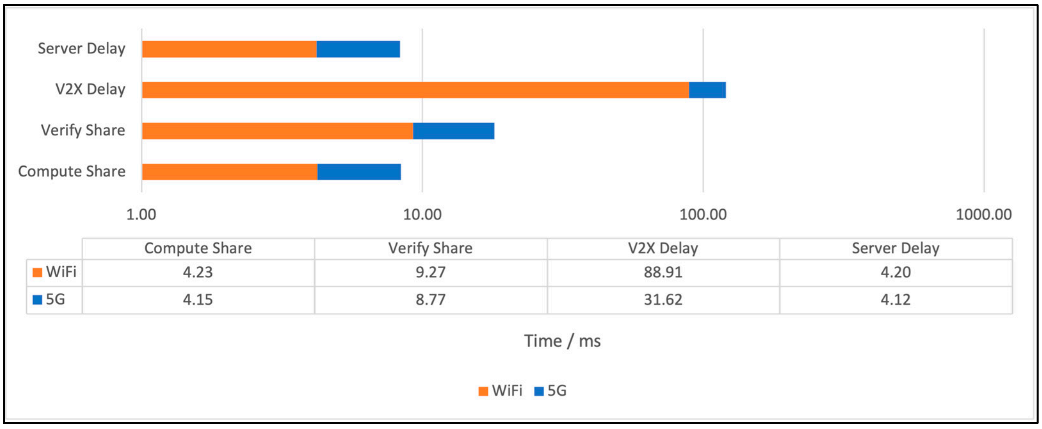


Figure 8. Time taken to compute a public key.

Figure 9 compares the joining time breakdown for 5G and Wi-Fi vehicles. It shows that the additional joining delay in the Wi-Fi vehicle is due to the V2X network delay, where the Wi-Fi vehicle connects to RSU via ad hoc Wi-Fi and then uses 5G. This shows the delay could be 100 ms. Figure 10 shows the CDF of the V2X network delay for both vehicles. Therefore, more focus should be on network designers to add more RSUs near the edge of the V2X network and potentially use RSUs with high transmission power to increase range.



	Compute Share	Verify Share	V2X Delay	Server Delay
🟧 WiFi	4.23	9.27	88.91	4.20
🟦 5G	4.15	8.77	31.62	4.12

Figure 9. The joining time breakdown for 5G and Wi-Fi vehicles.

Figure 11 shows the cloud network delay between TA communication and its average of less than 5 ms. Selecting a dedicated cloud network could potentially reduce this delay; however, it is small compared with the V2X delay. Therefore, more focus should be on improving V2X delay.

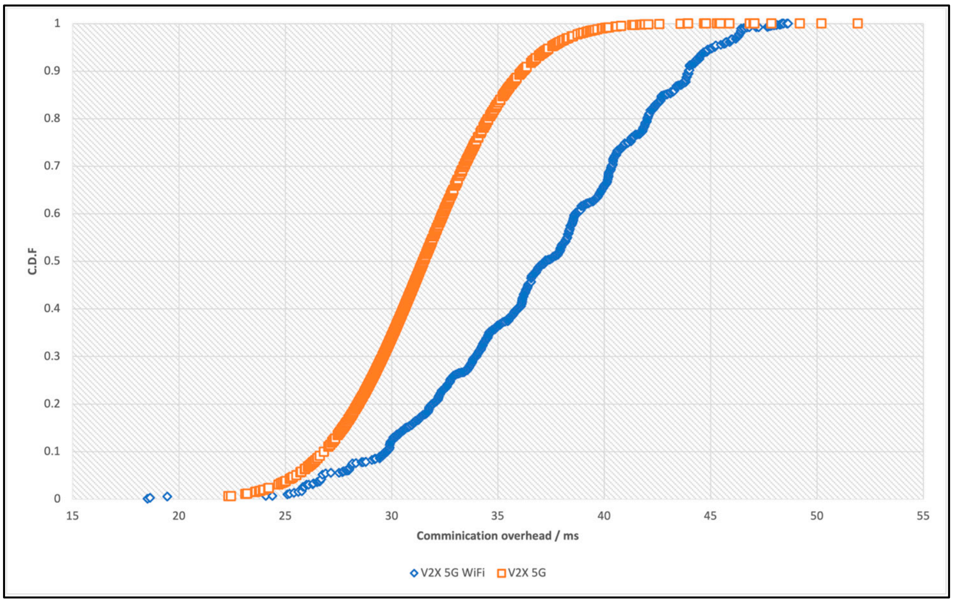


Figure 10. The CDF of the V2X network delay.

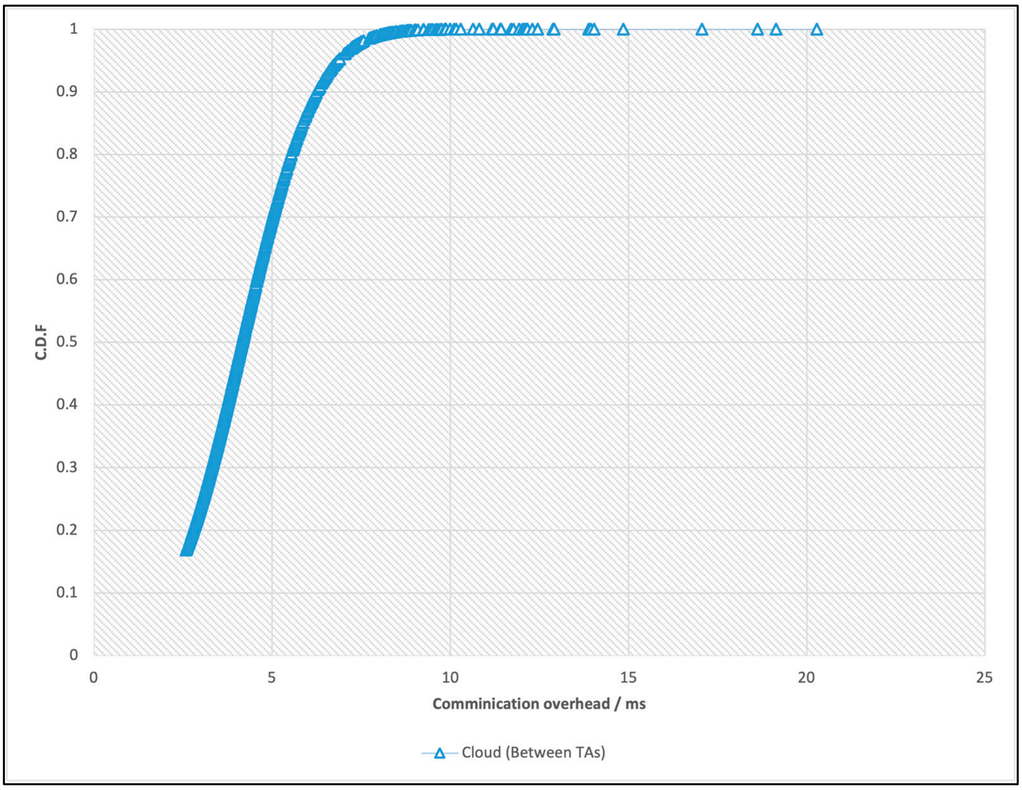


Figure 11. The cloud network delay between TA communication.

As shown in Figure 12, encryption and decryption time is not huge compared to network delays. For example, a 1024-byte message can be encrypted or decrypted within 1 ms. However, as shown in Figure 8, the public key calculation is expansive. Therefore, based on network requirements, we need to select the correct security parameters.

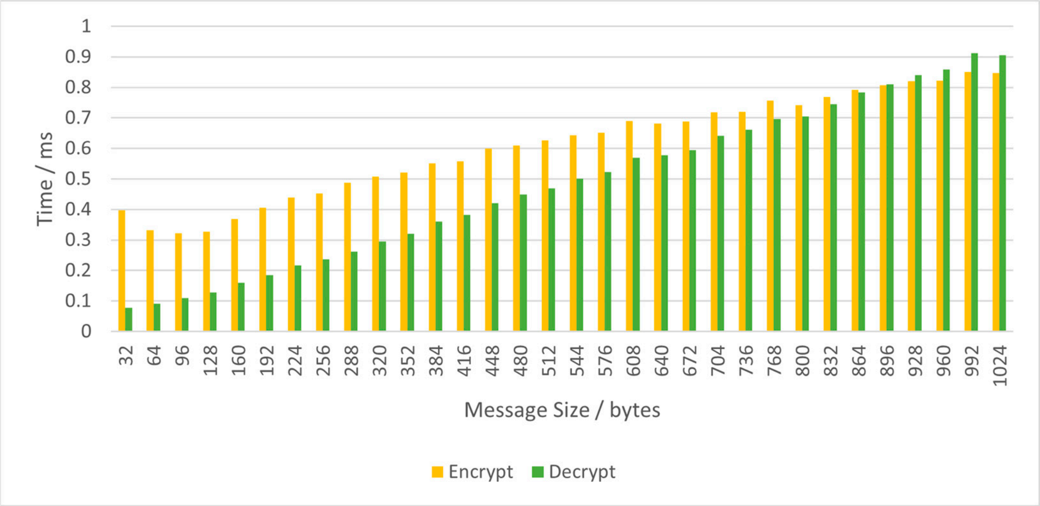


Figure 12. Comparison of encryption and decryption time.

5. Discussion

The previous section evaluated the proposed system in two ways. First, we evaluated the time taken for network initialization in Section 4.2, which is influenced by the size of the private key, the threshold value T, and the computational power of the machine executing the process. Second, we evaluated the performance of a vehicle joining a V2X network in Section 4.3. We used a stationary 5G-connected vehicle simulated in NS-3 in our V2X network to evaluate this and tried to join the network using two verification methods: client-verify and server-verify. In this section, we will discuss the key findings from the performance evaluations of the proposed system.

The use of larger private key sizes in V2X communication systems has been an ongoing discussion due to the potential benefits it could offer in terms of security. However, concerns regarding the potential overhead that may arise from the use of larger key sizes have been raised.

Our simulation that was discussed in Section 4.2 demonstrates that the initialization time increases linearly with the increase of the private key size. Nevertheless, our research results in Figure 5 have suggested using private key sizes can be larger than 512 bits without incurring considerable overhead, with an average processing time of 2–3 milliseconds. Another parameter that has been considered in the context of V2X network initialization is the threshold value T. Research has indicated that increasing the value of T does not have a significant impact on the network initialization process, allowing for larger T values to be used (greater than 12) without adverse effects. Therefore, network designers should try to use large private key sizes as well as large T values.

One approach we considered to reduce the overall time required for vehicles to join the V2X network is server verification, as shown in Figure 6. Performance analysis has shown that server verification can reduce verification time by 15% compared to local verification. Moreover, this time can be further reduced by increasing the computational power available in cloud-based systems.

The time taken to compute a public key is another factor that can impact V2X network performance. As shown in Figure 8, as the number of TAs and threshold values increase, the time required to compute a public key also increases.

Typically, V2X applications share information with neighbors, for example, traffic information and road conditions using message broadcast. In addition, V2X multi-hop routing protocols use the broadcast message to discover its one-hop neighbors. These messages are generated regularly and need to be delivered and consumed quickly. Therefore, we propose that broadcast messages be encrypted with a partial share received from the local TA with ElGalmal encryption. The receiving node can then compute the corresponding decryption key and decrypt the message with ElGalmal decryption. According to Figure 8, using only one TA reduces the calculation time to 10 ms. We also propose for the node to cache the calculated public key so that subsequential broadcast messages from the same source can be decrypted quickly.

6. Conclusions and Future Work

The performance evaluations presented in this research demonstrate that the proposed V2X communication system achieves efficient and secure communication with optimal parameter settings. Specifically, larger private key sizes, higher threshold values, server verification, and computational optimization techniques can significantly reduce the time required for vehicles to join the V2X network. Additionally, improvements to the network infrastructure, such as adding high transmission power RSUs near the edge of the V2X network and optimizing the multi-hop routing protocol, can help reduce V2X network delays.

In the future, we suggest enhancing the proposed solution by utilizing parallel and graphics processing units (GPUs) commonly available in modern devices. This would significantly reduce the computational overhead, allowing V2X network operators to use high-strength cryptographic information, leading to a more secure network. Additionally, using system-on-a-chip (SoC) hardware solutions can help reduce computational overhead while increasing security.

Our proposed security scheme can be applied to different V2X scenarios which need to be further studied. One such scenario is V2C: messages can be sent to cloud servers, where vehicles and RSU will collect information about road conditions, traffic patterns, weather information, driver and vehicle performance, and other data required by AI and ML algorithms. Usually, they are collected at vehicles and RSU and transferred to cloud servers for processing. These kinds of data can be encrypted using the full public key of an entity with ElGalmal encryption. Then, cloud servers can decrypt the data later. Another potential scenario is direct messages to vehicles. In AI and ML-based V2X, applications might need to send messages to vehicles, for example, change routes or request lane change. These kinds of data can be encrypted using the full public key of an entity with ElGalmal encryption. Then, the receiving vehicle can decrypt the message with its private key.

A key challenge in implementing this protocol on a large scale is deploying V2X infrastructure in the real world, ensuring most vehicles are connected to the V2X network via 5G or Wi-Fi, and ensuring an adequate number of RSUs are installed in road infrastructure. While this challenge is outside the scope of this research, it is an essential consideration for the practical implementation of the proposed protocol. Furthermore, provided that the V2X infrastructure is in place, implementing and scaling the proposed protocol in cloud infrastructure are expected to be easy.

The proposed protocol focuses on establishing a public/private key pair for each connected device, allowing any connected node to compute and validate the public key of any other connected device. This key pair can be used to secure decentralized or centralized awareness and warning messages, either for message encryption or adding a digital signature. However, integrating decentralized or centralized awareness and warning messages into the proposed protocol requires further evaluation in future research to determine its feasibility and effectiveness.

Author Contributions: Conceptualization, N.W.; Investigation, N.W.; Methodology, N.W. and E.P.; Software, N.W.; Validation, M.A.U. and C.H.; Writing—original draft, N.W.; Writing—review & editing M.A.U. and C.H.; Supervision C.P. and E.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Paul, A.; Chilamkurti, N.; Daniel, A.; Rho, S. *Chapter 2—Intelligent transportation systems, Intelligent Vehicular Networks and Communications*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 21–41.
2. 3GPP Universal Mobile Telecommunications System (UMTS); LTE. *Architecture enhancement for V2X Services (3GPP TS 23.285 version 14.2.0 Release 14) Technical Report*; ETSI: Sophia Antipolis, France, 2017; Available online: https://www.etsi.org/deliver/etsi_ts/123200_123299/123285/14.02.00_60/ts_123285v140200p.pdf (accessed on 19 April 2023).
3. Hamida, E.B.; Noura, H.; Znaidi, W. Security of cooperative intelligent transport systems: Standards, threats analysis and cryptographic countermeasures. *Electronics* **2015**, *4*, 380–423. [CrossRef]
4. IFilippi, A.; Moerman, K.; Martinez, V.; Haran, O.; Toledano-Autotalks, R. IEEE802.11p ahead of LTE-V2V for safety applications. In Proceedings of the 2017 IEEE Vehicular Networking Conference (VNC), Torino, Italy, 27–29 November 2017; pp. 1–8. Available online: https://www.nxp.com.cn/docs/en/white-paper/LTE-V2V-WP.pdf (accessed on 19 April 2023).
5. Dawood, M.; Fuhrmann, W.; Ghita, B.V. Assay of White Space Technology Standards for Vehicular Cognitive Access. In Proceedings of the Tenth International Network Conference (INC 2014), Plymouth, UK, 8–10 July 2014; Plymouth University: Plymouth, UK, 2014; pp. 23–33, ISBN 9781841023731.
6. Asadi, A.; Wang, Q.; Mancuso, V. A survey on device-to-device communication in cellular networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1801–1819. [CrossRef]
7. Lien, S.-Y.; Deng, D.-J.; Lin, C.-C.; Tsai, H.-L.; Chen, T.; Guo, C.; Cheng, S.-M. 3GPP NR Sidelink Transmissions Toward 5G V2X. *IEEE Access* **2020**, *8*, 35368–35382. [CrossRef]
8. *IEEE Std 1609.2-2013*; IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages, in IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006). IEEE: New York, NY, USA, 2013; pp. 1–289. [CrossRef]
9. Schlienz, R.A.; Whitepaper, J. *Device to Device Communication in LTE Technical Report*; Rohde & Schwarz: Coppell, TX, USA, 2015. Available online: https://www.rohde-schwarz.com/us/applications/device-to-device-communication-in-lte-white-paper_230599.html (accessed on 19 April 2023).
10. Zhang, M.; Fang, Y. Security analysis and enhancements of 3gpp authentication and key agreement protocol. *IEEE Trans. Wireless Commun.* **2005**, *4*, 734–742. [CrossRef]
11. Hsu, R.H.; Lee, J. Group anonymous D2D communication with end-to-end security in LTE-A. In Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS), Florence, Italy, 28–30 September 2015; pp. 451–459.
12. Saxena, N. Public key cryptography sans certificates in ad hoc networks. In Proceedings of the International Conference on Applied Cryptography and Network Security, Singapore, 6–9 June 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 375–389.
13. Saxena, N.; Tsudik, G.; Yi, J.H. Efficient Node Admission and Certificateless Secure Communication in Short-Lived MANETs. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *20*, 158–170. [CrossRef]
14. Saxena, N.; Tsudik, G.; Yi, J.H. Identity-based access control for ad hoc groups. In Proceedings of the International Conference on Information Security and Cryptology, Jeju Island, Korea, 5–9 December 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 362–379.
15. Saxena, N.; Yi, J.H. Noninteractive Self-Certification for Long-Lived Mobile Ad Hoc Networks. *IEEE Trans. Inf. Forensics Secur.* **2009**, *4*, 946–955. [CrossRef]
16. Elgamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **1985**, *31*, 469–472. [CrossRef]
17. Cho, J.K.; Lee, J.G.; Kim, Y.J. Security issues in V2X communication. *IEEE Commun. Mag.* **2018**, *56*, 94–100.
18. Garcia-Saavedra, A.F.; Serrano, P.; Banchs, A.; Rodriguez, P.S. 5G V2X Communications: Challenges and Opportunities. *IEEE Commun. Mag.* **2019**, *57*, 129–135.
19. Wu, B.; Wu, J.; Cardei, M. A survey of key management in mobile ad hoc networks. In *Handbook of Research on Wireless Security*; IGI Global: Hershey, PA, USA, 2008; pp. 479–499.
20. Bellovin, S.; Housley, R. Guidelines for Cryptographic Key Management. In *Request for Comments (RFC) 4107*; Internet Engineering Task Force: Fremont, CA, USA, 2005. Available online: https://tools.ietf.org/html/rfc4107 (accessed on 19 April 2023).
21. Raya, M.; Hubaux, J.-P. Securing vehicular ad hoc networks. *J. Comput. Secur.* **2007**, *15*, 39–68. [CrossRef]
22. Lu, R.; Lin, X.; Zhu, H.; Ho, P.-H.; Shen, X. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In Proceedings of the IEEE INFOCOM 2008—The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 1229–1237.
23. Wasef, A.; Jiang, Y.; Shen, X. ECMV: Efficient certificate management scheme for vehicular networks. In Proceedings of the IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference, New Orleans, LA, USA, 30 November 2008–4 December 2008; pp. 1–5.
24. Cui, J.; Wei, L.; Zhang, J.; Xu, Y.; Zhong, H. An Efficient Message-Authentication Scheme Based on Edge Computing for Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1621–1632. [CrossRef]
25. Capkun, S.; Buttyan, L.; Hubaux, J.-P. Self-organized public-key management for mobile ad hoc networks. *IEEE Trans. Mob. Comput.* **2003**, *2*, 52–64. [CrossRef]
26. Hegland, A.M.; Winjum, E.; Mjolsnes, S.F.; Rong, C.; Kure, O.; Spilling, P. A survey of key management in ad hoc networks. *IEEE Commun. Surv. Tutor.* **2006**, *8*, 48–66. [CrossRef]

27. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. [CrossRef]
28. Girault, M. Self-certified public keys. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, 8–11 April 1991; Springer: Berlin/Heidelberg, Germany, 1991; pp. 490–497.
29. Al-Riyami, S.S.; Paterson, K.G. Certificateless public key cryptography. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 30 November–4 December 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 452–473.
30. Rana, S.K.; Singh, M. Certificateless Efficient Group Key Management Scheme in Mobile Adhoc Networks. *Int. J. Comput. Sci. Issues (IJCSI)* **2011**, *8*, 343.
31. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
32. Zhou, L.; Haas, Y.J. Securing ad hoc networks. *IEEE Netw.* **1999**, *13*, 24–30. [CrossRef]
33. Chor, B.; Goldwasser, S.; Micali, S.; Awerbuch, B. Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proceedings of the 26th Annual Symposium on Foundations of Computer Science (sfcs 1985), Portland, OR, USA, 21–23 October 1985; pp. 383–395.
34. Feldman, P. A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), Los Angeles, CA, USA, 12–14 October 1987; pp. 427–438.
35. Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
36. Al Ibrahim, O.; Nair, S. Cyber-physical security using system-level PUFs. In Proceedings of the 2011 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, 4–8 July 2011; pp. 1672–1676.
37. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Paris, France, 9–11 April 1984; Springer: Berlin/Heidelberg, Germany, 1984; pp. 47–53.
38. Bellare, M.; Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; pp. 62–73.
39. Joux, A.; Nguyen, K. Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups. *J. Cryptol.* **2003**, *16*, 239–247. [CrossRef]
40. Stallman, R. "GNU General Public License Version 2." Free Software Foundation. 1991. Available online: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html (accessed on 19 April 2023).
41. Su, Z.; Liu, Y.; Xian, C.; Zhao, Y. Pre-specified-time coordination algorithm for convex optimization problems over weight-unbalanced networks. In Proceedings of the 2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanchang, China, 28–30 May 2021; pp. 156–161. [CrossRef]
42. Razavi, S.M.; Yuan, D. Mitigating mobility signaling congestion in LTE by overlapping tracking area lists. In Proceedings of the 14th ACM international Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '11), Association for Computing Machinery, New York, NY, USA, 24–28 October 2022; pp. 285–292. [CrossRef]
43. Tap Bridge Class Reference, 10. 2016. Available online: https://www.nsnam.org/docs/doxygen/classns3_1_1_tap_bridge.html (accessed on 19 April 2023).
44. Amazon Web Services. (n.d.) Amazon Elastic Compute Cloud (EC2) T3a Instances. Available online: https://aws.amazon.com/ec2/instance-types/t3/ (accessed on 19 April 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.