H3PC: Enhanced Security and Privacy-Preserving Platoon Construction Based on Fully Homomorphic Encryption

Badreddine Chah, Alexandre Lombard, Anis Bkakria, Abdeljalil Abbas-Turki, Reda Yaich

H3PC: Enhanced Security and Privacy-Preserving Platoon Construction Based on Fully Homomorphic Encryption

Badreddine Chah[1*], Alexandre Lombard[2*], Anis Bkakria[3◇], Abdeljalil Abbas-Turki[2*], Reda Yaich[5◇].

***Abstract*— With the increasing adoption of connected and autonomous vehicles, platooning services have emerged as a promising solution to enhance road traffic efficiency. However, the widespread deployment of platooning services raises concerns about the privacy and security of sensitive vehicle data. This paper proposes a novel privacy-preserving framework for platoon formation, named H3PC (Homomorphic Privacy-Preserving Platooning Construction), to address privacy and security challenges. The proposed approach is based on fully homomorphic encryption and order-preserving encryption, to enable secure and private operations within the platoon construction. In addition, the H3PC framework incorporates a safe vehicle control method that adheres to established norms in the literature. By striking a balance between security and computational efficiency, H3PC enables effective platooning construction. Furthermore, we present experimental results demonstrating the performance and latency of H3PC.**

I. Introduction

As an important part of a smart city, Intelligent Transportation Systems (ITSs) are the key to the integration of advanced technologies and communication systems into transportation infrastructure and vehicles to improve the safety, efficiency, and sustainability of transportation networks. Indeed, ITS is fundamental to the sustainable development of urban transport, taking into account fuel consumption and traffic efficiency. ITSs are highly related to Connected and Autonomous Vehicles (CAVs) as they involve the use of various technologies such as smart sensors, cameras, artificial intelligence (AI), and connectivity (Vehicle-to-Everything (V2X) communication). In the years to come, the roads will gradually be filled with autonomous vehicles that are able to drive themselves while cooperating with each other to form sustainable transportation systems. Chah, B. et al. [2] present and classify several innovative CAV services such as reducing road accidents, improving the quality of life, increasing the efficiency of transportation systems, and so on. In addition, ITS plays a critical role in providing the infrastructure and communication systems necessary for CAV to communicate, navigate, and assist in making informed decisions. As a representative driving pattern of CAV, the platooning use case [3] has great potential to improve safety, driver comfort, traffic efficiency, fuel efficiency, and emissions reduction.

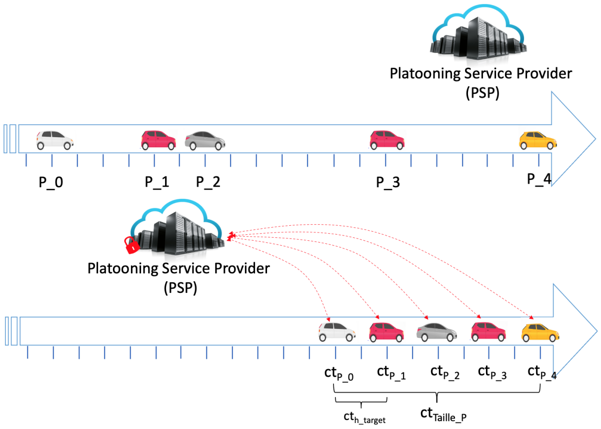

Fig. 1. Platooning illustration with encryption: the PSP has no access to the speed and position of vehicles

Imagine a scenario where a group of CAVs has an intersection in their path itinerary, as shown in Fig. 1, platooning services [3] allow them to travel closely together in a convoy-like formation. The CAVs in a platoon must maintain a short, constant, and safe distance from each other while moving synchronously. This is possible thanks to advanced technologies such as V2X communication and control systems. In addition, platoon service is more than just convoy formation. Other platooning shapes are harnessed to improve traffic performance. For example, virtual platoons can be formed at intersections [4] and lane changes to improve traffic flow and increase the infrastructure capacity.

In order to perform this promising service, it is needed to exchange real-time sensitive information between vehicles and interfaces. In this work, we consider sensitive data to be any information that leads to information related to a physical individual. Generally in the platooning service, the sensitive manipulated data are the vehicle identification, the real-time position and speed, the itinerary, the sensors' data, and the control. The service provider may add some other sensitive data that must be protected to ensure no entity can deduce sensitive information about the drivers. A strict framework is required to prevent possible abuses. Currently, there is general data protection legislation that addresses this issue. For example, the General Data Protection Regulation (GDPR) [5] which concerns European Union (EU) countries, and The California Consumer Privacy Act (CCPA) for the United States (US). These regulations govern the use of data and protect the users' privacy and

[1] Badreddine.chah@utbm.fr; [2] Alexandre.lombard@utbm.fr;
[3] Anis.bkakria@irt-systemx.fr; [4] Abdeljalil.abbas-turki@utbm.fr;
[5] Reda.yaich@irt-systemx.fr
[*] CIAD UMR 7533, Univ. Bourgogne-Franche-Comté, UTBM, F-90010 Belfort, France.
[◇] IRT SystemX, Palaiseau 91120, France.

personal information, providing a single legal framework for professionals. To comply with the legal requirements of the GDPR, car manufacturers and their suppliers must ensure that their development systems protect the privacy of users. To our knowledge, until now, no study has been conducted or proposed on this topic.

The rest of the paper is organized as follows. Section II introduces our main contribution. In Section III, we provide an overview of the platooning use case and introduce the concept of fully homomorphic encryption (FHE). Section IV focuses on our proposed framework, "H3PC: Homomorphic Privacy-Preserving Platooning Construction". Section V presents the findings of our conducted experiments, including the first experiment using homomorphic max/min operations, as well as its lighter technique that leverages order-preserving encryption (OPE) methods. Finally, Section VI resumes the main findings and contributions of the study and discusses the limitations and future directions.

II. Proposed Contribution

The focus of this work is on the Platooning Formation. Where platooning is initiated by selecting a vehicle leader, the group of CAVs, and computing the intersection itinerary path for all entities within the group. Our technique H3PC is employed after this initialization phase, where the vehicles come together to form the platoon 1. This formation process can be driven by either a human driver or autonomous driving capabilities. The CAVs within the group need to merge at the same x-position, which is a variable that changes based on the behavior of the CAVs. It is important to note that all CAVs within the group already know the path of the itinerary since it is their original trajectory. We represent it as a straight-ahead path in Figure 1. The CAVs only require the target information to synchronize with the other entities in the group. In a platoon, vehicles often position themselves behind the lead vehicle, maintaining a safe and constant distance between them. Vehicles establish communication links to share their position and speed and coordinate their movements. These data are sensitive information that can be linked to the drivers.

To meet the aforementioned objectives, we propose the following three main contributions:

(i) Development of a privacy-preserving protocol for Platooning Formation based on Homomorphic Encryption (HE): The study introduces a protocol that utilizes Homomorphic Encryption, a secure and widely used cryptographic technique, to perform calculations on encrypted data. This approach ensures the confidentiality of sensitive information while enabling the formation of platoons.

(ii) Provision of a service without requiring access to sensitive data: The protocol offers a service that does not rely on clear access to sensitive data such as Location and Speed. This allows for the grouping of vehicles in an x-position while maintaining data privacy and security.

(iii) Utilization of a Platooning Service Provider (PSP) for secure computations and platoon identification: The proposed protocol relies on a central server known as the Platooning Service Provider (PSP). The PSP performs heavy computations over the clients' encrypted data and delivers encrypted outputs of nearby platoons. Notably, the PSP does not have access to the client's sensitive data, ensuring the privacy of the individuals involved.

(iv) Proposition of the Lightweight Linear Control Equation: The proposed equation is a lightweight version that is based solely on addition and multiplication operations. The equation is suitable for homomorphic techniques.

III. Background

A. Platooning Use-Case Description

Let us consider a group of vehicles that drive on the same highway each day. Some drivers need to rest and others need to perform other tasks during this driving period. The platooning service [3] offers the possibility to automatically join and follow a nearby convoy instead of stopping for a certain time. To manage the CAVs groups, there is an entity that receives information about the group of vehicles and makes decisions, named Platooning Service Provider (PSP).

This paper focuses on the Platooning Formation (see Fig.1). To this end, PSP uses a method that allows vehicles to merge near one point based only on their respective speeds and positions. More precisely, based on the received speeds and positions, the PSP, must compute the suitable acceleration for each CAV in the group platoon. This control method should meet the following criteria:

- Safety: Control methods need to prioritize safety, maintain a safe distance between the vehicles, and be able to react appropriately to any potential hazard or change in the environment.
- Stability and Robustness: Control methods have to be stable and resistant to changing traffic conditions, vehicle dynamics, and communication delays.
- Smooth merger: Control methods must provide a smooth merging, coordinating speed and position adjustments to minimize disturbance to the platoon CAVs.
- Adaptability & Scalability: Control methods must be adaptable to different traffic scenarios, road conditions, and platoon sizes. It should be able to handle varying speeds and dynamic changes within the platoon.
- Energy Efficiency: Control methods should optimize energy efficiency by smartly managing vehicles' accelerations.

By meeting the aforementioned criteria, control methods can help establish stable behavior, ensuring safe and efficient operation while maintaining the desired formation and dynamics of the platoon. However, the platooning operation requires the exchange of specific real-time sensitive data, such as positions and speeds. The revelation of such sensitive data can therefore have severe consequences [2]. A solution to this issue can be found by having the PSP able to compute the platooning instructions (e.g. Target Acceleration) based on encrypted data, in compliance with the GDPR principles. This can be achieved using FHE techniques, as we discuss in the following.

B. Fully Homomorphic encryption (FHE)

In 2009, Gentry [14] made a discovery in the field of cryptography by defining the first fully homomorphic encryption scheme (FHE). The FHE is a powerful cryptographic primitive that allows calculations to be performed on encrypted data without having access to the secret key and without the need to decrypt. The hardness of FHE is based on a mathematical problem known as the Learning With Errors (LWE) problem. LWE is considered among the lattice-based cryptography problems, which is robust to quantum attacks. They are among the problems considered as *Difficult*, i.e. there is no polynomial algorithm to solve this problem.

In this work, we focus on the CKKS scheme implemented in the library OpenFHE [16]. The CKKS scheme is particularly designed for calculations on real or complex numbers, which makes it suitable for applications that involve calculations with real numbers, such as our use case (platooning services). In addition, the schema implemented in OpenFHE is the RNS variants of the CKKS scheme presented in [12]. The RNS variant uses a different scaling factor for each level. To do that, the authors utilize the Chinese Remainder Theorem (CRT) representation to break multi-precision integers in Z_q into vectors of smaller integers $(q_0, q_1, \ldots, q_L)$ to perform operations efficiently using native (64-bit) integer types. For $L > 0$. The CKKS security relies on the Ring Learning with Errors (RLWE) problem hardness.

The idea behind the construction of FHE is based on adding noise during encryption and key generation to obtain the hardness properties. In addition, the system includes multiple parameters that determine the level of security, functionality, performance, and precision supported by the scheme. These parameters are as follows: 1) Number of fractional bits f, corresponding to the accuracy of the computation. 2) Plaintext modulus p. 3) Ciphertext modulus q. 4) Ciphertext dimension n. We suppose that each plaintext value is represented as a fixed-point binary number that has f fractional bits after the radix point. Rescale operation is a tool that allows adjusting the value of f for a ciphertext, which is a distinctive feature of CKKS. It always follows a homomorphic multiplication in the OpenFHE library. To determine the computing capabilities of the system, the cipher text module q is the main functional parameter for this task. The higher the q parameter, the more operations can be performed on the encrypted data and the higher the accuracy. For a chosen value of q, the dimension of the cipher text n determines the security level of the system. A larger n represents higher security (see article for more details [15]).

FHE allows a third party, such as a PSP, to perform calculations on encrypted data without having access to the plaintext. The result of the calculation is encrypted, and only the owner of the data can decrypt it. In [12], the authors detail the construction of their scheme with the necessary mathematical proofs. Their algorithms for homomorphic operations satisfy the following properties:

- **Homomorphic Addition** – $Add(c_1, c_2)$, for given encrypts of m_1 and m_2, output is an encryption of $m_1 + m_2$ which satisfies $En(m_1 + m_2) = En(m_1) + En(m_2)$.
- **Homomorphic Multiplication** – $Mult_{evk}(c_1, c_2)$,for given encrypts of m_1 and m_2, output is an encryption of $m_1 * m_2$. Which satisfies $En(m_1 * m_2) = En(m_1) * En(m_2) + e_{mult}(mod\ q)$ for some additional error $e_{mult} \in R$.

Note that the noise grows as the number of encrypted computations increases. This is especially an issue with homomorphic multiplication. So, the number of operations needs to be limited otherwise the consequence is a false decryption. By increasing q, the modulus in the ciphertext space, there is more room for noise, which allows more operation. However, the latency of the algorithm increases exponentially with respect to the value of q. Another solution is presented in the literature, under the name of *Bootstrapping Technique* [17]. This technique increases the level of a cipher text in order to perform more homomorphic operations. It is used to reset the noise and keep the value of q relatively small. As we know from the literature, FHE, even with bootstrapping technique, has a severe handicap that is confirmed in this paper: It has limited functionality, in terms of the operations to be performed. Some operations on encrypted messages such as computing the root, inverting, maximum, and minimum are possible but take a significant amount of time (see V-A).

IV. H3PC: Homomorphic Privacy-Preserving Platoon Construction

Since we have discussed both the technical aspects of platooning formation and the cryptography considerations, this section provides a detailed presentation of our proposed algorithm. We begin by describing the privacy-preserving protocol, followed by an explanation of the equations used to form the platoon based on encrypted data.

A. Threat Model and Security Requirements

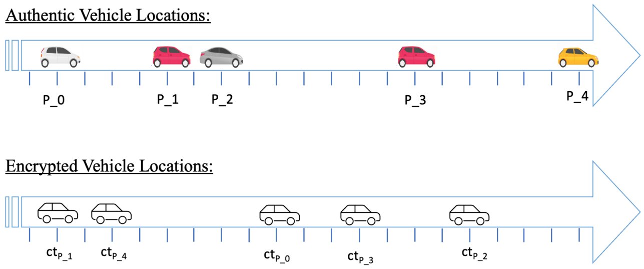

Fig. 2. Encrypted perspective: what PSP sees

The system architecture studied in this work comprises two entities: the vehicles (V_i) and the PSP. The task of each vehicle is to provide the necessary data to calculate the instruction. Each vehicle is considered an honest but curious entity that provides correct data and follows the designed protocol and the received instructions. They will eventually merge into the intersection itinerary and be close enough to visually identify each other, such as the vehicle's number plate. The PSP is the entity that provides the instructions, such as the target acceleration. It is an honest but curious entity, which follows the protocol properly but is curious to deduce information about the vehicles. It provides the services and computing infrastructure that will be used to perform operations on sensitive encrypted data.

Our priorities are the following:

- The PSP should not have any access to sensitive information about the vehicles (see Fig. 2).
- No entity should be able to learn the control equation used by the PSP.
- Malicious entities should not be able to intercept the data flow between the vehicles and the PSP.

B. Notation

In accordance with the construction provided by the authors in [12], the following protocol formally describes the vehicle control platoon on encrypted data. We define R_{q_L} as the residue ring of R modulo an integer q_L. Its elements are polynomials of degrees less than n with coefficients in $Z/q_L Z$. More precisely $q_L < n$. In the CKKS construction, we focus on the following parameters:

- Plaintext modulus (p): This parameter represents the modulus used for plaintext values. It should be a positive integer.
- Ciphertext modulus (q_ℓ): The ciphertext modulus is the modulus used for encrypted values. The primes q_ℓ for $i = 1, ..., \ell$ are chosen to be as close to 2^p as possible to minimize the error introduced by rescaling (see section 2 in [12]).
- Level ciphertext (ℓ): It refers to a vector in $R^k_{q_\ell}$, where k is a fixed integer.
- Real value σ: It follows the χ distribution. It is discrete Gaussian with standard deviation σ if all coefficients of $a \leftarrow \chi$ are selected from discrete Gaussian distribution with standard deviation σ.
- Number of fractional bits f: is the representation as a fixed-point binary number.

At level ℓ, the modulus is given by $q_\ell = 2^{p_0 + \ell \cdot p} = q_0 \cdot \Delta^\ell$. In the construction presented in [12], the authors utilize a zero-level modulus q_0 and a chain of prime modulo $q_1, q_2, \ldots, q_L$ of the same size, satisfying $q_i \equiv 1 \mod 2n$ for $i = 1, \ldots, L$. The modulus Q_ℓ is computed as the product of this modulo up to level ℓ, i.e., $\prod_{i=0}^{\ell} q_i$. The value of L corresponds to the highest level of the ciphertext modulus.

To simplify the notation, for given $p > 0$ and a modulus q_0, we set $q_\ell = q_0 \cdot \Delta^\ell$ for ℓ such that $0 < \ell \leq L$. The integer p serves as the scaling base for approximate computation. For security considerations, we select a parameter $M = M(\lambda, q_L)$ representing a cyclotomic polynomial of degree $n = \phi(M)$, where λ is the security parameter. These parameters play a crucial role in CKKS construction and determine the security, precision, and efficiency of the scheme. The scaling factor is denoted as $\Delta = 2^p$, and the zero-level modulus is defined as $q_0 = 2^p$. It is important to ensure that q_0 is greater than Δ to enable correct decryption.

C. Proposed Protocol

Since we are familiar with the entities involved in the communication and the various notations used. Let's assume that each vehicle will establish individual communication channels with the PSP. So, all communications are over TLS 1.3 protocol with a unique symmetric private key for each vehicle [23]. In the upper layer (over TLS 1.3), all vehicles need to establish a common public and private key that is exclusive to the platoon. To accomplish this, during the initial setup phase, the vehicles will engage in a collaborative communication protocol to generate a common public and private key. The protocol ensures that the keys are securely distributed among the vehicles while maintaining confidentiality. Once the common public key is derived, it will be communicated to the PSP, enabling secure interactions and information exchange between the vehicles and the PSP.

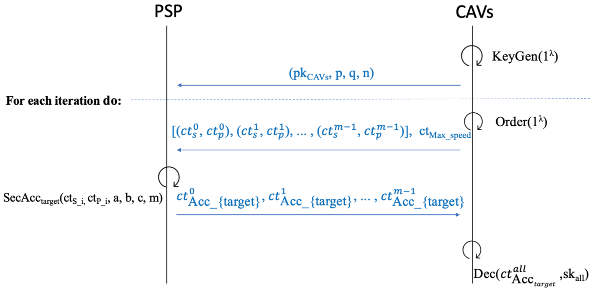
PSP
CAVs
KeyGen(1^λ)
(pk_{CAVs}, p, q, n)
For each iteration do:
Order(1^λ)
$[(ct_s^0, ct_p^0), (ct_s^1, ct_p^1), \ldots, (ct_s^{m-1}, ct_p^{m-1})]$, ct_{Max_speed}
$SecAcc_{target}(ct_{S\_i}, ct_{P\_i}$, a, b, c, m)
$ct^0_{Acc\_\{target\}}, ct^1_{Acc\_\{target\}}, \ldots, ct^{m-1}_{Acc\_\{target\}}$
$Dec(ct^{all}_{Acc_{target}}, sk_{all})$

Fig. 3. H3PC Encrypted Communication Protocol

As illustrated in the Fig. 3. Our construction concentrate on seven algorithms ($Setup$, $KeyGen$, Enc, $EvalBootstrap$, $Order$, $SecAcc_{target}$, Dec). The protocol for generating the shared asymmetric homomorphic pair keys between vehicles is out of the scope of this paper. The details of our construction are as follows:

- `Setup`$(1^\lambda, \Phi)$ Any vehicle in the platoon must agree to respect the parameters $Fixed\_Parm$ and CKKS shame chosen, so these parameters will be omitted. Let's select a FHE environment (n, p, q, σ). Set the small distributions χ_{key}, χ_{err}, and χ_{enc} over R for secret, error, and encryption, respectively. The Setup algorithm returns $Fixed\_Parm \leftarrow (n, p, q, \sigma, \Phi)$.
- `KeyGen`(1^λ) : The algorithm performs the key generation algorithm of the FHE defined [12]. In this context, it is assumed that the vehicles collaboratively establish the common keys, while adhering to the following properties:
  - Secret key (sk) generation: The secret key s choose randomly from χ_{key} Set the secret key as $sk \leftarrow (1, s)$.
  - Public key (pk) generation: The public key $pk = (p_1, p_2) \in R^2_{q_L}$ is computed by $p_2 \leftarrow -p_1 \cdot sk + e \ (mod \ q_L)$. And by choosing $p_1 \leftarrow R_{q_L}$ and $e \leftarrow \chi_{err}$.
- `Enc`(m_i, pk) : Each ith-vehicle encrypt it message m_i, which is the position P_i and the speed S_i. The Encryp-

tion algorithm return $ct_m = v \cdot pk + (m_i + e_0, e_1) mod q_L$ where v is choosing randomly from χ_{enc} and $(e_0, e_1) \leftarrow \chi_{err}$.

- `EvalBootstrap`$(ct_i, numIterations, precision)$: Let $ct_i = (ct_{i1}, ct_{i2})$ be a ciphertext at level $\ell = 0$. The goal of the bootstrapping operation is to increase the ct_i level, i.e. to increase the number of homomorphic multiplications to apply on ct_i. The Bootstrapping algorithm returns the same ciphertext but with a higher level (See [16], [17] for more details).
- $Order(S_1, S_2, ..., S_m)$: To order the set of speeds from minimum speed to maximum speed, we need to perform the Max or Min operation. However, it has been tested that the homomorphic Max/Min function is heavy in terms of computation time. Therefore, for each iteration t, we utilize an alternative function to secretly compute a vector of speeds in sorted order. For more information, please refer to the article [18]. The result of the private ordering algorithm is denoted as $OrderVec_{Speeds} \leftarrow PrivateOrder(S_1, S_2, ..., S_m)$. In addition to encrypting the speed and position homomorphically, the vehicles send the encrypted positions using Order-Preserving Encryption (OPE) to order them.
- $SecAcc_{target}(ct_{S_i}, ct_{P_i}, a, b, c, m)$: The inputs a, b, c and m are constant. Where $a \in R_{qL}$ is a ciphertext of the max car speed in real-time. $b \in R_{q_L}$ is the encrypted inverse of the number of the vehicles m. $c \in R_{qL}$ is the ciphertext of the inverse of the number 3. For a given encrypted speed ct_{S_i} and position ct_{S_i} of the vehicle i. This algorithm returns the target acceleration $Acc_{targetV_i}$ instruction to be respected, $Acc_{targetV_i} \leftarrow SecAcc_{target}(ct^i_S, ct^i_P, a, b, c, m)$. After several iterations, the respect of this target acceleration should lead the m vehicles to form a platoon. (see more details in the section IV-D)
- `Dec`(m, sk): The same as the common Public key, each vehicle has the common secret key of the platoon to decrypt the result, by computing $m' = ct_2 + ct_1 \cdot s\ mod\ q_L$, $m' \leftarrow MPC_{Dec}(ct, s)$.

The algorithm $SecAcc_{target}(ct_{S_i}, ct_{P_i}, a, b, c, m)$ is based on a linear equation a linear equation, which will be presented in the following subsection.

D. H3PC Linear Control Technique

In this work, we propose a method that relies on a linear model of vehicle acceleration control to form a consistent and safe platoon. This method has the advantage to be only based on addition, multiplication, and their inverse operation to ensure that vehicles maintain a specified safety and following distance, adjust their acceleration accordingly, and react in real-time to changes in platoon configuration or external conditions. The method is inspired from [1], to form a consensus between agents in a distributed system.

The equation 3 considers the position and speed of all vehicles to compute the target speed S_{target} and the target position P_{target}. And, the equation 5 calculates the target acceleration Acc^i_{target} for each vehicle i. It takes into account various factors, such as the difference in speed and position compared to the mean values, the deviation from the ideal platoon size, and the desired position and speed for each vehicle.

To achieve this, we first compute the average speed $S_{Average}$ and position $P_{Average}$ using follow two equations : $P_{Average}(P_0, P_1, ..., P_{m-1}) = \frac{\sum_{i=0}^{m-1} P_i}{m}$ and $S_{Average}(S^0, S^1, ..., S^{m-1}) = \frac{\sum_{i=0}^{m-1} S_i}{m}$. Next, we determine the actual size of the ideal platoon after construction by first computing the ideal target distance h_{target_P} (see Fig. 1), using Equation 1. In this equation, p_{sec} represents the security distance between vehicles in the ideal platoon, which can be set to 1 meter. Additionally, p_{conv} represents the convoy position and can be 2 meters. Finally, h_t denotes the driver's reflex time, which is assumed to be 2 seconds.

$$h_{target_P} = \frac{(p_{sec} \cdot S_{max} + p_{sec} \cdot S_{Average})}{(p_{sec} + p_{conv})} \cdot h_t \tag{1}$$

Then, the size of the ideal platoon after construction is calculated using Equation 2. After the merging of vehicles, the variable $Taille_P$ represents the desired size of the platoon (see Fig. 1).

$$Taille_P = h^P_{target} \cdot m - 1 \tag{2}$$

Based on this information, we deduce the target speed for each vehicle, which is set to the average speed. Furthermore, To get the target position for the first vehicle P^0_{target}, we need to compute the equation 3.

$$\begin{aligned} S_{target} &= S_{Average} \\ P^0_{target} &= P_{Average} - \frac{Taille_P}{2} \end{aligned} \tag{3}$$

Since the PSP has knowledge of the vehicle order, we can calculate the target position of the ith-vehicle in the ideal platoon. By computing the equation 4.

$$For\ k = m - 1\ to\ 1 : P^i_{target} = P^{i+1}_{target} + h^P_{target} \tag{4}$$

And to compute the target position of the other vehicles following the 0th vehicle, we can use the following sequential calculation: starting from the last vehicle to the leader, for $k = 1$ to $m - 1$, we have $P^i_{target} = P^{i-1}_{target} + h_{target_P}$. This equation holds for the vehicles ordered as $(P_0, P_1, ..., P_{m-1})$.

In the second part of the technique, we compute the target acceleration Acc^i_{target} for ith-vehicle using the results obtained so far, and the equation 5. Where $\alpha_s = 2$ and $\alpha_p = 0,1$ are the speed and the control gains, respectively.

$$Acc^i_{target} = \alpha_s(S^i_{target} - S_i) + \alpha_p(P^i_{target} - P_i) \tag{5}$$

Note that all operations performed thus far are computed on encrypted data, ensuring that the PSP does not gain any sensitive information about the results or even the intermediate computations. Algorithm 1 outlines our secure target acceleration algorithm, which achieves polynomial complexity $O(m)$. Furthermore, the equations are known only by the PSP, as some companies may need to conceal their operations to maintain a competitive advantage. Regarding the constants used in the equations, they can be adjusted to achieve the desired specifications and requirements of the PSP, thereby allowing control over the final shape of the platoon.

Algorithm 1 $SecAcc_{target}$ algorithm

```
Input: $(ct_S^0, ct_P^0)$, $(ct_S^1, ct_P^1)$, ..., $(ct_S^{m-1}, ct_P^{m-1})$, $a$, $b$, $c$, $m$,
    $S_{max}$
Output: $ct_{Acc^0_{target}}, ct_{Acc^1_{target}}, ..., ct_{Acc^{m-1}_{target}}$
1: $ct_{P_{Aver}} \leftarrow P_{Average}(ct_P^0, ct_P^1, ..., ct_P^{m-1})$
2: $ct_{S_{Aver}} \leftarrow S_{Average}(ct_S^0, ct_S^1, ..., ct_S^{m-1})$
3: $ct_{S_{target}} \leftarrow ct_{S_{Aver}}$
4: $ct_{h^P_{target}} \leftarrow \frac{(p_{sec} \cdot ct_{S_{max}} + p_{sec} \cdot ct_{S_{Aver}})}{(P_{sec} + P_{conv})} \cdot h_t$
5: $ct_{Taille_P} \leftarrow ct_{h^P_{target}} \cdot (m-1)$
6: $ct^0_{P_{target}} \leftarrow ct_{P_{Aver}} - \frac{ct_{Taille_P}}{2}$
7: for $(i = 0,\ i{+}{+},\ i < m)$ do
8:     $ct^i_{Acc_{target}} \leftarrow Acc_{target}(ct_{P^i_{Aver}}, ct_{S^i_{Aver}}, ct^i_{P_{target}})$
9: end for
```

The H3PC technique stands out from others for several reasons. Firstly, the equations used in this technique involve only multiplication and addition operations, making it highly compatible with the FHE technique. This allows for the secure computation of encrypted data without compromising the privacy of sensitive information. Secondly, the H3PC technique requires only two essential data from the vehicles: the position and the speed. By utilizing these key parameters, the technique effectively captures the necessary information for controlling the platoon formation and maintaining the desired behavior. Thirdly, the H3PC technique incorporates a crucial aspect known as the maximum time to maintain stable driving behavior. This parameter is carefully set within the range of $[100\ ms, 600\ ms]$, ensuring that the platoon operates smoothly and maintains safe and consistent driving patterns over time.

The technique meets most of the criteria mentioned in Section III-A (see Table I). However, it still requires some additional tests. There is only one case that did not mark: Adaptability. Our technique is currently being tested on ideal roads and platoon sizes, but it still needs to be evaluated in various traffic scenarios, and road conditions.

	S	S&R	SM	A	Scal	EE
H3PC	✓	✓	✓	✗	✓	✓

TABLE I
CRITERIA ASSESSMENT OF H3PC TECHNIQUE. WHERE S IS SAFETY, S&R IS STABILITY&ROBUSTNESS, SM IS SMOOTH MERGER, A IS ADAPTABILITY, SCAL IS SCALABILITY AND EE IS ENERGY EFFICIENCY

V. EXPERIMENTAL RESULTS

In this section, we experimentally evaluate the performance of the H3PC construction based on two techniques. Our aim is to present and compare two experiments in terms of calculation costs, demonstrating their practicality in real-life scenarios.

In order to elect the performance and accuracy of the CKKS scheme for privacy-preserving platoon construction. We present the configuration details that were used. We fix the number of fractional bits $f = 3$, to represent a fixed-point binary number. Then, We choose the ring dimension $n = 1024$, and plaintext modulus $p = 50$. n and p are fixed, then the coefficient modulus q_i is chosen to be as close to 2^p as possible to minimize the error introduced by rescaling. The security parameter λ is equal to $64bits$, which we build $128 - bit$ CKKS schemes. In other words, a malicious entity would need to perform 2^128 operations to exhaustively search the key space and find the correct key. We choose the lowest possible parameter because for HE, the larger the values, the longer it takes to implement the code (see [16], [19] for more details).

H3PC algorithm is implemented using OpenFHE cryptographic library [16] over the CKKS schema. And run on a Mac OS interface with a dual-core Intel Core i5 processor at 2.7 GHz (Turbo Boost up to 3.1 GHz), and 8 GB of integrated LPDDR3 memory at 1866 MHz. The construction is also usable on other operating systems as Linux and Windows.

For the equation 4 of the H3PC algorithm, PSP needs to determine the order of the CAVs based on their encrypted positions. The hard approach is to perform the homomorphic Max/Min operation V-A. Alternatively, a lighter approach is to utilize the OPE technique V-B.

A. Experiment 1: Homomorphic Max/Min

Max/Min is homomorphically difficult to perform and time-consuming. We have utilized mathematical skills (The Convergence) to convert it into addition and multiplication. As shown by Chon, J. et al. [20], the authors employ mathematical skills to convert a complex concept into a simpler form 6. However, the conditions for a and b require them to be in the range of [0, 1), for d is the number of iterations. Knowing that positions and speeds used in this work are in the real number ensemble $\mathbb{R}$. Solution that we propose is to convert into $a' = a/(a+b)$ and $b' = b/(a+b)$ for $(a', b') \in [0, 1)$.

$$\begin{aligned} Max(a, b; d) &= \frac{a+b}{2} + \frac{sqrt((a-b)^2; d)}{2} \\ Min(a, b; d) &= \frac{a+b}{2} - \frac{sqrt((a-b)^2; d)}{2} \end{aligned} \tag{6}$$

Up until now, we have considered two new operations, namely the inverse and the square root. For the inverse operation, we relied on the algorithm presented in the article [21]. The idea is based on the Goldschmidt division algorithm, where the algorithm works by multiplying the divisor repeatedly by a correction factor until the quotient is obtained (in our simulation, $d = 7$ iterations for the initialized factor equal to 0,5). So we get $a' = a * inv(a + b; d)$ and $a' = a * inv(a + b; d)$ for $(a', b') \in [0, 1)$. As for the square root, we relied on the algorithm described in the article [20]. The technique is based on Wilkes' method proposed in 1951 [22]. It allows us to converge towards the square root result after performing d iterations (in our simulation, $d = 12$). Final, to compare a and b we multiply the result of $Max(a', b'; d)$ with $(a+b)$, which have the follow equation: $Max(a, b; d) = Max(a', b'; d) * (a+b)$. These computations require at least 14 homomorphic multiplication, utilizing the Bootstrapping technique [12] to increase the multiplication depth of the plaintext.

Unfortunately, this method suffers from significant computational overhead, leading to prolonged computation times. For the same configurations mentioned in subsection V, the computation of the homomorphic maximum between only two vehicle positions can take at least $800ms$. Considering the need to compare all vehicles within the platoon group to get their order. For instance, the time taken to compute the order of 4 vehicles based on the Maximum between two values is $6 \cdot 800ms \approx 4800ms$. In addition to this computation, the Bootstrapping, Encryption, $SecAcc_{target}$, and Decryption operations take approximately $360ms$, $1ms$, $35ms$, and $2ms$, respectively II. This means the total for only one iteration of communication is $\approx 5198ms >> 600ms$. Note that when the communication time (i.e., vehicle sending data and receiving target acceleration) exceeds $600ms$, timely decision-making becomes crucial to maintain optimal platoon dynamics (see Figure 4). That is Why in the following section, we present a lightweight method to efficiently and quickly determine the CAVs' order.

B. Experiment 2: Order-Preserving Encryption (OPE) technique

To address the time issue of maximum calculation in our techniques, we incorporate another type of encryption called Order-Preserving Encryption (OPE) [18]. OPE is a cryptographic technique that maintains the order of encrypted data. In other words, if two plaintext values are arranged in a particular order, their corresponding ciphertexts will also maintain the same order. Recalling our protocol 3, in addition to sending the positions encrypted homomorphically with CKKS, the vehicles included in the message the encryption of the same value P_i using OPE. This allows the PSP to order the encrypted positions of the vehicles while preserving their privacy. For the configurations mentioned in subsection V, to order m have a Constant Complexity $O(1)$.

In order to evaluate the privacy-preserving capabilities of our proposed system, we choose to conduct our simulation using homomorphic encryption and OPE encryption technique. The simulation was performed using selected example values based on our criteria. Table II presents the performance of key operations in our work. It is important to note that the time step range must be maintained at less than $600ms$ to have stable vehicle control. For the same parameters presented in Section V, the computation of Secure Target acceleration depends on the number of vehicles and has a polynomial complexity of $O(m)$. The multiplication depth in the CKKS scheme is set on 10 holomorphic multiplication, without the need for the Bootstrapping technique. The total time is the cumulative sum of the individual times taken for each operation involved in the process. $Total = Enc + SecAcc_{target} + Dec$, see table II. Using the OPE technique, the time required to compute the encrypted target acceleration for 4 vehicles is $1 + 6 + 2 = 9ms <<< 4800ms$, which is significantly lower than the time required by the Homomorphic Max/Min technique discussed in the previous Subsection V-A.

V_{Numr}	Enc	$SecAcc_{target}$	Dec	$Total$
m = 4	1 ms	$\approx 6ms$	2 ms	9 ms
m = 8	1 ms	$\approx 10ms$	2 ms	13 ms
m = 10	1 ms	$\approx 14ms$	2 ms	17 ms
m = 15	1 ms	$\approx 18ms$	2 ms	21 ms
m = 20	1 ms	$\approx 26ms$	2 ms	29 ms
m = 25	1 ms	$\approx 40ms$	2 ms	43 ms

TABLE II

PERFORMANCE EVALUATION OF OUR SYSTEM FOR A SINGLE ITERATION.

In Fig. 5, 6, and 7, we present the behaviors of four vehicles that adhere to Platooning service. The simulation is conducted with a time step of $9ms$, and the initial positions and speeds of the vehicles are as follows: $Vehicle_1$ (S_1, P_1) = (10, 120), $Vehicle_2$ (S_2, P_2) = (12, 70), $Vehicle_3$ (S_3, P_3) = (20, 30), and $Vehicle4$ (S_4, P_4) = (12, 0). Fig. 5 visually depicts how the vehicles smoothly adapt their acceleration to navigate their trajectories. Each vehicle adjusts its acceleration dynamically, ensuring a seamless transition in its movement. Fig. 6 showcases the speed adaptation process, demonstrating how the vehicles synchronize their speeds to converge towards a common value. This synchronization plays a vital role in maintaining coordination among the vehicles. Lastly, in Fig. 7, we observe the vehicles gradually approaching their respective target positions while maintaining a safe distance between each other. An example of this behavior can be seen in Vehicle 2, which adapts its acceleration by decelerating when necessary. This enables it to align its speed closer to the target speed (V_{target}) and its position closer to the desired target position. The value of S_{target} and P_{target} are dynamically adjusted in each iteration to keep the smooth behavior. Overall, these figures provide insights into the effectiveness of our system in facilitating coordinated and safe vehicle movements while prioritizing privacy preservation and efficiency.

VI. CONCLUSION

In conclusion, this research presents an advanced solution to address the Platooning Formation problem by harnessing the power of homomorphic encryption. To our knowledge, this is the first approach to tackle this problem using homomorphic encryption techniques, and the proposed solution, H3PC: Homomorphic Privacy-Preserving Platooning Con-

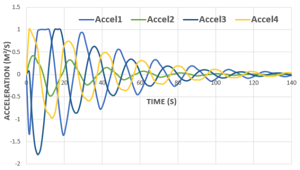

Fig. 4. Experiment 1: Disrupted and Irregular Acceleration Behavior.

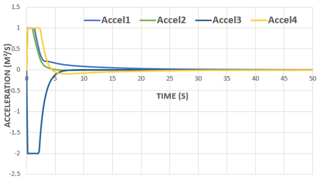

Fig. 5. Experiment 2: Encrypted Accelerations of 4 Vehicles During Their Trajectory.

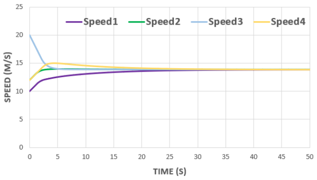

Fig. 6. Experiment 2: Encrypted Speeds of 4 Vehicles During Their Trajectory.

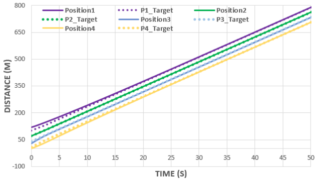

Fig. 7. Experiment 2: Comparison of Real Positions of 4 Vehicles with their Encrypted Target Positions.

struction, demonstrates its effectiveness in achieving secure and privacy-preserving platooning. Several key aspects of the proposed methodology are noteworthy. H3PC enables clients participating in the platooning service to secure merge with other vehicles at intersections while maintaining safe and smooth dynamics. The control method ensures safety distances, preserves client data privacy, and upholds the integrity of operations performed by the PSP. Importantly, this service maintains confidentiality without compromising its ability to provide platoon formation.

In our future work, we aim to simulate our vehicle control technique to ensure adaptability to various traffic scenarios and road conditions. More analytical results of the platoon stability and safety will be provided to adjust the platoon according to the computation and communication performances. Additionally, on the security front, we intend to strengthen the shared common key mechanism to withstand potential attacks from malicious clients. By addressing these aspects, we can further evaluate the performance and efficacy of our techniques in real-life scenarios involving CAV. These advancements will contribute to the practical implementation and deployment of our proposed solution, making significant strides towards privacy-preserving.

ACKNOWLEDGMENT

This research is funded by the German-French joint collaboration program on “Cybersecurity” funded by the MESRI (via ANR) and BMBF (via VDI/VDE IT) (ANR-20-CYAL-0008), within the frame of AUTOPSY project. In addition, we thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] MARDEN, Jason R. et SHAMMA, Jeff S. Game-theoretic learning in distributed control. In : Handbook of dynamic game theory. Springer, Cham, 2017. p. 1-36.

[2] CHAH, Badreddine, LOMBARD, Alexandre, BKAKRIA, Anis, et al. Privacy Threat Analysis for Connected and autonomous vehicles. Procedia Computer Science, 2022, vol. 210, p. 36-44.

[3] Kavathekar, Pooja, and YangQuan Chen. ”Vehicle platooning: A brief survey and categorization.” International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 54808. 2011.

[4] ABBAS-TURKI, Abdeljalil, MUALLA, Yazan, GAUD, Nicolas, et al. Autonomous Intersection Management: Optimal Trajectories and Efficient Scheduling. Sensors, 2023, vol. 23, no 3, p. 1509.

[5] EU General Data Protection Regulation (2016), http://www.eugdpr.org/.

[6] ENISA good practices for security of Smart Cars, November 25, 2019.

[7] Commission National for Information and Freedom, “CONNECTED VEHICLES AND PERSONAL DATA,” OCTOBER 2017 EDITION.

[8] C. Gentry et al., “Fully homomorphic encryption using ideal lattices.” in STOC, vol. 9, no. 2009, 2009, pp. 169–178.

[9] Brakerski, Zvika, and Vinod Vaikuntanathan. ”Fully homomorphic encryption from ring-LWE and security for key dependent messages.” Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings 31. Springer Berlin Heidelberg, 2011.

[10] Brakerski, Zvika, Craig Gentry, and Vinod Vaikuntanathan. ”(Leveled) fully homomorphic encryption without bootstrapping.” ACM Transactions on Computation Theory (TOCT) 6.3 (2014): 1-36.

[11] Cheon, Jung Hee, et al. ”Homomorphic encryption for arithmetic of approximate numbers.” Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. Springer International Publishing.

[12] Kim, Andrey, Antonis Papadimitriou, and Yuriy Polyakov. ”Approximate homomorphic encryption with reduced approximation error.” Topics in Cryptology–CT-RSA 2022: Cryptographers’ Track at the RSA Conference 2022, Virtual Event, March 1–2, 2022, Proceedings. Cham: Springer International Publishing, 2022.

[13] Fan, Junfeng, and Frederik Vercauteren. ”Somewhat practical fully homomorphic encryption.” Cryptology ePrint Archive (2012).

[14] J. H. Cheon, A. Costache, R. C. Moreno, W. Dai, N. Gama, M. Georgieva, S. Halevi, M. Kim, S. Kim, K. Laine, Y. Polyakov, and Y. Song. Introduction to homomorphic encryption and schemes. In K. Lauter, W. Dai, and K. Laine, editors, Protecting Privacy through Homomorphic Encryption, pages 3–28, Cham, 2021. Springer International Publishing. 4

[15] Lauter, Kristin Estella, Wei Dai, and Kim Laine, eds. Protecting Privacy Through Homomorphic Encryption. Springer International Publishing AG, 2022.

[16] Al Badawi, Ahmad, et al. ”OpenFHE: Open-source fully homomorphic encryption library.” Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. 2022.

[17] Bossuat, Jean-Philippe, et al. ”Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys.” Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I. Cham: Springer International Publishing, 2021.

[18] Boldyreva, Alexandra, et al. ”Order-preserving symmetric encryption.” Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28. Springer Berlin Heidelberg, 2009.

[19] Albrecht, Martin, et al. ”Homomorphic encryption standard.” Protecting privacy through homomorphic encryption (2021): 31-62.

[20] Chon, Jung Hee, et al. ”Numerical method for comparison on homomorphically encrypted numbers.” Advances in Cryptology–ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part II. Cham: Springer International Publishing, 2019

[21] Goldschmidt, Robert E. Applications of division by convergence. Diss. Massachusetts Institute of Technology, 1964.

[22] M. V. Wilkes. The Preparation of Programs for an Electronic Digital Computer: With special reference to the EDSAC and the Use of a Library of Subroutines. Addison-Wesley Press, 1951.

[23] Bhargavan, Karthikeyan, Bruno Blanchet, and Nadim Kobeissi. ”Verified models and reference implementations for the TLS 1.3 standard candidate.” 2017 IEEE Symposium on Security and Privacy (SP).