PISA: Privacy-Preserving Smart Parking

Sayon Duttagupta
COSIC, KU Leuven
Leuven, Belgium
sayon.duttagupta@esat.kuleuven.be

Dave Singelée
COSIC, KU Leuven
Leuven, Belgium
dave.singelee@esat.kuleuven.be

***Abstract*—In recent years, urban areas have experienced a rapid increase in vehicle numbers, while the availability of parking spaces has remained largely static, leading to a significant shortage of parking spots. This shortage creates considerable inconvenience for drivers and contributes to traffic congestion. A viable solution is the temporary use of private parking spaces by homeowners during their absence, providing a means to alleviate the parking problem and generate additional income for the owners. However, current systems for sharing parking spaces often neglect security and privacy concerns, exposing users to potential risks. This paper presents PISA, a novel Privacy-Preserving Smart Parking scheme designed to address these issues through a cryptographically secure protocol. PISA enables the anonymous sharing of parking spots and allows vehicle owners to park without revealing any personal identifiers. Our primary contributions include the development of a comprehensive bi-directional anonymity framework that ensures neither party can identify the other, and the use of formal verification methods to substantiate the soundness and reliability of our security measures. Unlike existing solutions, which often lack a security focus, fail to provide formal validation, or are computationally intensive, PISA is designed to be both secure and efficient.**

***Index Terms*—Smart parking, Privacy-preserving, V2X, CPS**

I. Introduction

The escalating number of vehicles in urban areas, combined with the relatively slow expansion of public parking infrastructure, has led to a significant shortage of available parking spaces. This growing scarcity poses challenges for urban mobility and increases congestion, leading to longer transit times and heightened frustration among drivers [1], [2], [3], [4], [5]. As cities develop, the construction of additional parking facilities becomes increasingly difficult due to high costs and limited land availability. These factors exacerbate the existing imbalance between parking supply and demand. The problem is further complicated by the variability in parking needs across different neighbourhoods and times of day. For example, private parking spaces in residential areas are often left vacant during the day when residents are away at work. This situation highlights a potential opportunity to better utilise these private spaces, yet existing solutions for parking spot sharing frequently fall short in terms of security and privacy.

Current literature on smart parking systems often lacks rigorous formal verification of their security models and fails to adequately address privacy concerns. Many systems either do not offer sufficient security assurances or are encumbered by computational inefficiencies, leaving significant gaps in their effectiveness and reliability. The need for a robust framework that ensures both security and efficiency in parking management remains unmet. Addressing these challenges requires innovative approaches that not only enhance the utilisation of parking resources but also safeguard user privacy. By bridging the gap between the increasing demand for parking and the limitations of existing systems, we can improve urban mobility and reduce congestion.

In this paper, we want to explore a different approach to alleviate the problem of parking congestion and propose PISA: Privacy preservIng Smart pArking, an Airbnb for parking spots, while preserving anonimity. In our scheme, a parking spot which is owned by someone can benefit by hosting their available parking spot when they are not there to someone who might be looking for a free spot and gain some monetary benefits.

Contribution

In this paper, we present PISA, a novel Privacy-Preserving Smart Parking scheme, and provide a framework for the efficient sharing of private parking spots, addressing the scarcity of parking spaces in urban environments. The main contributions of this paper are the following:

- Provide by-directional anonymity by ensuring mutual anonymity between vehicle owners and parking spot providers, and protecting the identities of both parties involved.
- Design a novel cryptographically secure protocol to secure communications and data exchanges within the parking system.
- Defines a comprehensive set of security and privacy requirements for smart parking systems and demonstrates how PISA satisfies these properties.
- Finally, we formally prove the security of our solution using Verifpal [6].

Availability Statement. Our formal verification code and the analysis log are published in the following link, under a permissive open-source licence: https://gitlab.esat.kuleuven.be/Sayon.Duttagupta/PISA

II. Related works

Numerous research efforts have focused on privacy-preserving smart parking systems using frameworks like

blockchain, Private Information Retrieval (PIR), and cryptographic techniques. However, these solutions often face limitations in security, practicality, efficiency, and robustness.

Amiri et al. [7] propose a system that uses blockchain and PIR to protect user privacy, allowing drivers to find parking without revealing their location. This system queries a blockchain-based ledger of parking offers via PIR. Several studies explore similar approaches using blockchain to address parking issues [8], [9], [10]. However, these methods depend heavily on blockchain nodes' computational and storage capacities, leading to scalability issues as users and transactions increase. They also require extensive infrastructure, which may not be feasible in all urban areas. While these systems offer strong privacy guarantees, they often do not address denial-of-service attacks or dynamic pricing models effectively and lack security proofs for their protocols, making it hard to assess their robustness.

Another study introduces a decentralised, anonymous smart parking scheme combining blockchain with cryptographic signatures and zero-knowledge proofs [11]. Although this approach anonymises transactions and ensures parking record integrity, it introduces substantial computational overhead, potentially slowing transactions in real-time scenarios. While bilinear pairings have been used in IoT contexts [12], [13], the authors in these works have also argued only the specific use cases where it can be accommodated, and unfortunately, in vehicular networks, the latency is very high.

Some researchers have investigated the use of traditional cryptographic techniques without blockchain to create secure and private parking systems [14], [15], [16], [17], [18]. These solutions often involve complex key management and encryption schemes to protect user data and ensure secure communications between vehicles and parking infrastructure. While effective in theory, these approaches can be cumbersome to implement and manage, especially in large-scale deployments. They also tend not to focus on security at all, and when they do, they focus on specific aspects of privacy or security, such as data encryption or user anonymity, rather than providing a comprehensive solution that addresses multiple threats simultaneously. Furthermore, commercially deployed solutions, such as Prked [19] in the US and JustPark [20] in the UK, failed to launch in the EU as they were not GDPR compliant.

Our proposed PISA scheme addresses these issues by offering a formally verified, cryptographically secure protocol ensuring bi-directional anonymity. Unlike existing solutions, PISA undergoes rigorous verification with Verifpal, delivering a more reliable and privacy-preserving solution for urban parking.

III. Structure of PISA

To realize our PISA scheme of a friendly privacy-preserving parking solution, we need to explicitly declare all the parties and entities involved in the process and describe the overall architecture and structure.

Registration Authority $\mathcal{R}$. It can be assumed that there exists a database which is securely stored and maintained by the government/local authority (such as the police, or DMV) and contains personal identifier details of governmental documents, such as driver's license, national card number, house documents, etc. We denote this entity as Registration Authority. The security of such databases is out of scope, and are assumed to be honest parties. The registration authority is only used during the registration phase at the start to register all different entity members and at the end to de-anonymize the identity of such members if required, and hence, is not required to be connected to the overall ecosystem and can be offline. It can also, in practice, be multiple authorities (e.g. one for the address of a citizen, and another one for the car registrations). But conceptually, it can be modelled as a single entity.

Service Provider $\mathcal{S}$. The Service Provider is an integral part of the smart-parking system and is the main responsible entity for running the protocol and regulating messages between the other three entities. In theory, the service provider is an honest-but-curious party, being the main central entity in the protocol. But no personal identifiers, except the parking spot's location and the home owner's UID, are being shared with the service providers. The service provider also knows nothing about the identities of the car owners, but only their signed public keys from the registration authority. The service provider is also responsible for allocating the parking-spot listings into a neighbourhood group with enough diversity to preserve their privacy. The service provider is also responsible for matching a car owner with the desired parking spot and sending the monetary provider an account of all the transactions. It is also responsible for keeping a note of when a car has left its spot and calculating the amount that needs to be charged.

Monetary Provider $\mathcal{M}$. The Monetary Provider is used as an accountant and to facilitate sending/receiving money for the transactions. The monetary provider allows payment via privacy-friendly electronic-coins to use as a financial commodity in the transactions. The monetary provider also allows for purchasing the tokens via normal payment gateways, such as bank transfers or using their credit/debit card. The monetary provider does not leak the identities of such persons to other entities, and only sends the required user ID and transaction number to the concerned entity. Although the inclusion of the monetary provider is not necessary for the working of the protocol, and can be merged with the service provider if necessary, we decided to keep it a separate entity to provide complete anonymity, even with the financial transactions. Using this, we can make the service provider an *honest-but-curious* model, and not a trusted third party, which reduces the security and trust of the system. We discuss this in detail in Sect. VI.

Home Owner $\mathcal{H}$. The Home Owner is the entity that lists out the available parking spots to rent. They need to authorize themselves with the registration authority and with the receiving UID, need to ask the service provider to list their given address up for renting. The home owners are required

to install a lightweight sensor, such as a roadside unit (RSU), near the parking spot for the cars to communicate with it. The home owner knows nothing about the identities of the car which comes to their spot to park their vehicles.

Car Owner $\mathcal{C}$. The car owners are the vehicle owners who want to reserve and book a parking spot, without compromising their identities or personal details. They need to also install a sensor in their car to communicate with the parking sensor, for example an onboard unit (OBU) if the car does not come with one. They have to register themselves with the registration authority first, and with the received credentials can ask the service provider for a list of parking spots.

IV. Threat model and assumptions

The focus of our PISA scheme is to provide a platform that facilitates the secure and anonymous leasing of parking spaces within urban environments. The primary security threats we address include the risk of unauthorised vehicles impersonating legitimate vehicles to gain access to parking spots, and the potential for homeowners to charge higher fees than permitted by exploiting a session established between the adversary and the legitimate party. Our threat model encompasses both types of adversaries: those posing as home owners and those posing as vehicle owners. Additionally, our model protects against adversaries aiming to breach the anonymity of either the homeowner or the car owner. Such adversaries may attempt to compromise the system by running multiple instances of the protocol to deduce and expose the identities of the participants. We consider both passive adversaries, who monitor and eavesdrop on unsecured channels, and active adversaries, who may inject and tamper with messages in transit.

We assume that the Registration Authority is a trusted entity, such as a governmental agency. All other participants are assumed to be *honest-but-curious*, and collusion among entities is not anticipated. Homeowners are expected to install roadside units (RSUs) at their parking spots, while car owners will equip their vehicles with onboard units (OBUs). These OBUs and RSUs are designed as lightweight, tamper-resistant devices that remain fixed in place to ensure secure vehicle-to-roadside communication, and can be produced in large quantities at a low cost [21], [22]. Communication between RSUs and OBUs is inherently secure; however, all other interactions between entities such as $\mathcal{C}$/$\mathcal{H}$ and $\mathcal{R}$/$\mathcal{S}$/$\mathcal{M}$ are conducted over secure channels, such as TLS. Additionally, use-cases where $\mathcal{C}$ ceases to send tokens to $\mathcal{H}$—analogous to scenarios where an unauthorised vehicle parks in a designated spot—are considered out of scope for this scheme. In such instances, it is expected that the homeowner would contact law enforcement to address the issue of unauthorised parking.

V. Working of the PISA protocol

To fully realise our privacy-friendly parking scheme, we first need to initialise all the above-mentioned entities and then communicate securely between the parking spot and the vehicle.

A. PISA – Registration phase

In this subsection, we will first initialise all the involved parties with the central Registration Authority $\mathcal{R}$, as shown in Fig. 1.

1) Initially, all the participating parties must register themselves with the Registration Authority, which we will henceforth mention in this paper using its notation $\mathcal{R}$. All the participating parties need to contact and authenticate themselves to $\mathcal{R}$, using existing governmental identifications like driver's license, etc, prior to the protocol execution. $\mathcal{R}$ will then distribute the already generated public parameters required to establish the cryptographic keys, such as the choice of elliptic-curve and the curve parameters, and the public key formats. For the purposes of generalisation within this paper, we utilise Curve25519 [23] for all public key operations. Nonetheless, our protocol is not restricted to any specific public key algorithm or curve and can be implemented using alternative public key algorithms as per the implementer's preference. We have, however, provided cryptographic recommendations that balance algorithmic compatibility with considerations of speed and performance.
2) The Home Owners $\mathcal{H}$ can send their free parking spot location along with their identification to receive a randomly generated UID, along with a $\mathcal{R}$'s signed version of it (the signature also contains some randomness, to avoid replay attacks), via a secure channel. We use Ed25519 [24] as our signature algorithm, which is the EdDSA signature algorithm using SHA-512 and Curve25519.
3) The Car Owners $\mathcal{C}$ can also authenticate themselves by sending their names and vehicular details to $\mathcal{R}$. $\mathcal{R}$ will, in turn, generate a random number x and issue $\mathcal{C}$ its public key along with its signature of the public key.
4) The Registration Authority $\mathcal{R}$ then sends these values to the Car Owner $\mathcal{C}$ using a secure channel like TLS.
5) The Registration Authority also keeps a record of every UID to ID and public key to ID mapping, stored securely in its database.
6) After the registration phase has concluded, $\mathcal{R}$ informs a central entity, the Service Provider $\mathcal{S}$, of all new public keys it issued, and also sends the public key of the service provider $\mathcal{S}$, to $\mathcal{H}$ and $\mathcal{C}$, for the following steps in the protocol phase.

B. PISA – Protocol phase

In this subsection, we will provide a detailed description of the working of the protocol, as shown in Fig. 2.

1) There exits a central body known as the Service Provider $\mathcal{S}$ which acts as a regulator between the Home Owner $\mathcal{H}$ and the Car Owner $\mathcal{C}$. The Home Owner and the Car Owner have installed a device on their parking spot and vehicle respectively, which would communicate with each other and with the Service Provider's central server. It is also assumed that all the entities would be running

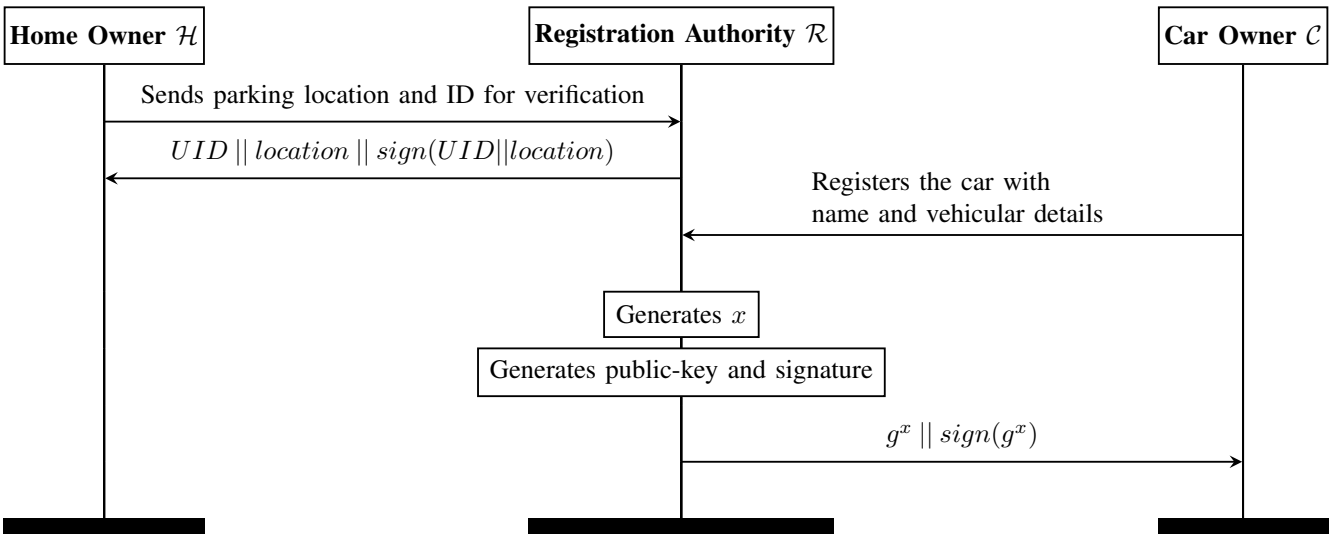

Fig. 1: Registration Phase

a user-friendly application on their smartphone, which would communicate with the Service Provider's server, to facilitate key-exchanges.

2) The Home Owner activates the application to list their parking spot and transmits the UID received from $\mathcal{R}$ to the Service Provider, along with the available time duration and a randomly generated nonce α. The Service Provider verifies the UID using the provided signature and subsequently lists the parking spot as available on its platform. The spot is assigned to a specific geographical group (neighbourhood), and a uniform price-per-hour charge (€$/hr$, \$$/hr$) is established for all spots within that neighbourhood. This geographical grouping is determined based on the number of parking spots available in each neighbourhood to ensure sufficient diversity and anonymity. In cases where there are only a few spots in a neighbourhood, adjacent geographical groups may be merged to mitigate de-anonymisation risks. Although this merging may involve a minor trade-off in pricing, it enhances the overall privacy of the system. Following these steps, the Service Provider sends an acknowledgement message back to the Home Owner.
3) The Car Owner will open the app to find a parking spot. It will send the Service Provider a freshly generated ephemeral public key along with its signature, which will be only valid and used in this instance of the protocol execution. The Service Provider will verify the signature of its public key to authenticate the Car Owner. It will then compute a nonce β and send it to the Car Owner by encrypting it using the Car Owner's public key. The Car Owner then needs to decrypt the ciphertext and send the β back to the Service Provider, thus guaranteeing mutual authentication. Meanwhile, the Car Owner can also purchase tokens to pay for the parking spot from the Monetary Provider $\mathcal{M}$. The Monetary Provider is a separate entity, hence even if the Car Owner wants to purchase tokens via orthodox methods of bank transfer or using their bank card, the Monetary Provider does not disclose their identities and will delete the personal identifier linked to a transaction right after the transaction is completed, only storing the UID/public key. To ensure additional privacy, the Car Owner can opt to buy the tokens using privacy-friendly cryptocurrencies [25], [26].
4) The Car Owner after receiving the tokens from the Monetary Provider and authenticating itself to the Service Provider, will send a request to the Service Provider to show all available parking spots in an approximate area. The Service Provider will reply back by showing a list of such spots in a geo-location and with its approximate distance range to the Car Owner. The Car Owner then selects the desired location, generates a random secret k, and sends these values to the Service Provider. The Service Provider then removes the listing from the application, sends the actual address to the Car Owner and charges the Car Owner for the first hour.
5) Along with the location, the Service Provider also sends the Car Owner the nonce α sent by that specific Home Owner to the Service Provider beforehand, the total time t available to park their vehicles, the first hash $H(r)$ of a random seed r and a signature of the t^{th} hashchain $sign(H^t(r))$. The Service Provider will also send k to the Home Owner. These hashchain values will be used by

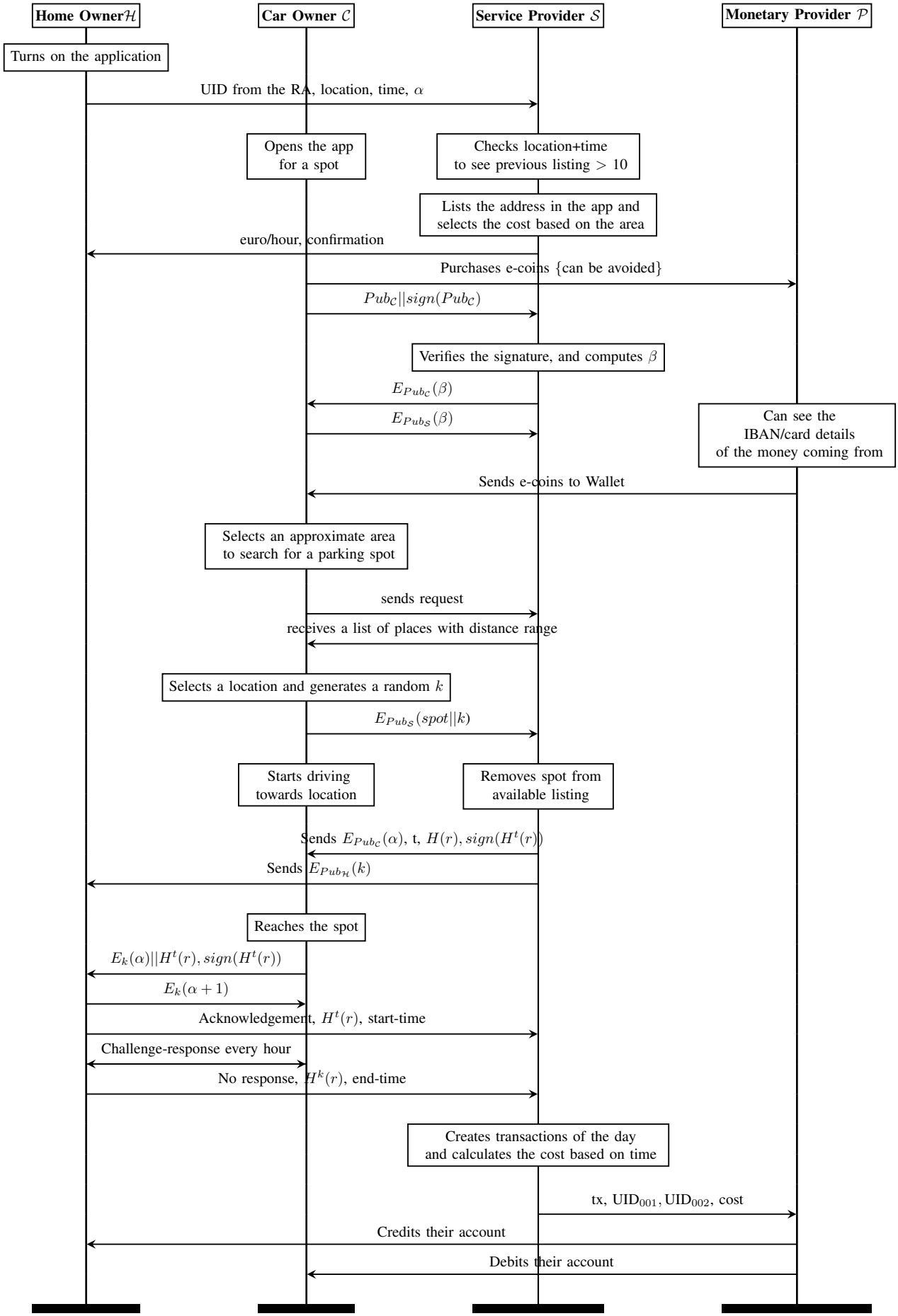

Fig. 2: Protocol Phase

the Home Owner to authenticate the vehicle to be parked, and will also prevent any adversarial home owners from charging more rent for that available parking spot. We recommend using BLAKE2 [27] as the hash function algorithm.

6) The Car Owner will generate the t-hashchains of the received initial hash message, i.e., $H(H(...H(r))) = H^t(r)$. After the vehicle reaches the parking spot, it sends out the nonce α encrypted using the secret k, the t^{th} hash, and its signature received from the service provider: $E_k(\alpha) \mid\mid H^t(r) \mid\mid \ sign(H^t(r))$. The Home Owner will verify the nonce received with the one initially generated and shared with the Service Provider. It will also verify the signature of the last hash message. If succeeded, then the Home Owner will send an acknowledgement message to the Car Owner by encrypting using AES-GCM-128 and sending an incremented nonce $E_k(\alpha+1)$, thus providing mutual authentication. The Home Owner will also notify the Service Provider that the vehicle has parked by sending the last t^{th} hash and the start-time. To verify the vehicle is still present. every hour, there will be a challenge-response between the parking sensor (RSU) and the vehicle (OBU), with the vehicle sending the $(t-m)^{th}$ hash every hour, where m is the number of hours passed, and the Home Owner can verify this by first computing the hash of the received token m times and then verifying its signature. In the end, the Home Owner will send the last received hash token $H^{t-m}(r)$ and the end-time to the Service Provider, and it will calculate the time difference and charge the Car Owner.
7) At the end, the Service Provider sends a record of all the transactions with the UID numbers to the Monetary Provider, and it then distributes their payments.

VI. Security & Privacy properties

The main goal of PISA is to provide a platform for both the home owners and car owners to anonymously park their cars at available personal parking spots while maintaining a degree of security and privacy guarantees. We formally define these parameters and claim that our solutions provide these properties.

Definition 6.1 (Cryptographically secure): We claim that our protocol is cryptographically secure as long as the underlying mathematically-hard assumption holds true. The security of our scheme is based on the computational Diffie-Hellman (CDH) problem [28] over the elliptic curve – curve25519. It also assumes that the cryptographic hash function *H* being used is a pre-image resistant one-way hash function and the key sizes are long enough, that is, 256-bit for ECC operations and 128-bit for AES operations. In Sect. VII, we detail and prove how our protocol is secure against all known attacks, for example, but not limited to, man-in-the-middle attacks, nonce validation, nonce reuse attacks, duplication verification, authentication attacks, replay attacks, spoofing, to name a few.

Definition 6.2 (Bidirectional anonymity): The Home Owner $\mathcal{H}$ does not have any information about the parked car or its owner $\mathcal{C}$ and neither does the Car Owner $\mathcal{C}$ know about the personal details of the parking spot's owner. The Car Owner $\mathcal{C}$ also only knows the exact address of only one spot per transaction.

Definition 6.3 (Entity anonymity): The other entities which take part in the execution of the protocol, namely the monetary provider $\mathcal{M}$ and the service provider $\mathcal{S}$, also have very limited knowledge of the personal identifiers of $\mathcal{C}$ and $\mathcal{H}$. The service provider $\mathcal{S}$ only knows a random ID assigned to the home $\mathcal{H}$ and the car $\mathcal{C}$ owners. $\mathcal{S}$ only knows the list of locations with available parking spots per time interval and the UID of $\mathcal{H}$. The monetary provider $\mathcal{M}$ only knows a randomly assigned UID per user who is purchasing the e-coins and only the identity if purchasing via bank transfer.

Definition 6.4 (Reliability): The protocol's soundness and correctness remain unaffected even if either of the participating entities goes offline or tries to become corrupt by intentionally manipulating the synchronisation in place. Due to the use of hash tokens and both entities verifying each other every hour, neither party can claim an advantage in case one of them goes offline.

Definition 6.5 (Traceability): The Registration Authority $\mathcal{R}$ is able to recover the real identities of the Car Owners $\mathcal{C}$ and the Home Owners $\mathcal{H}$ who are acting malicious and need to be reported while participating in the smart parking protocol.

Definition 6.6 (Unlinkability): None of the tokens, keys, or passwords used in any protocol execution can be linked to any entity in this state or the previous state or to another entity in future executions of the protocol. Our protocol guarantees the freshness of all keys and the unlinkability of all parameters.

VII. Formal Analysis

This section provides a detailed explanation of our formal analysis of PISA. Verifpal [6] is a new software tool used to formally verify the security of cryptographic protocols. It has already been employed to analyse the security of complex protocols such as Signal, Zoom, the DP-3T decentralised pandemic-tracing protocol and a lot more academic protocols in the literature [29], [30], [31], [32], [33], [34], [35], [36]. Verifpal relies on a symbolic model that does not check for computational soundness, assuming that the cryptographic primitives and functions are perfectly secure. Besides its very intuitive language, another key property of Verifpal is its prohibition on designing custom cryptographic primitives, thus avoiding well-known user errors.

The first step in verifying the security of the preliminary registration phase and the vehicular protocol phase was to model them using Verifpal. We have included the formal verification code both in the Appendix A and in an anonymous repository for the analysis to be re-run locally and compared with the provided output logs, as shown in Fig. 3. Additionally, we have uploaded both protocols to Verifpal's repository, VerifHub*, for easy verification. In our formal analysis, we consider both

*https://verifhub.verifpal.com/744a07ae9daf9053466fe71b20437c1e
https://verifhub.verifpal.com/db5a1b827275d45eec620c3566e1b0e9

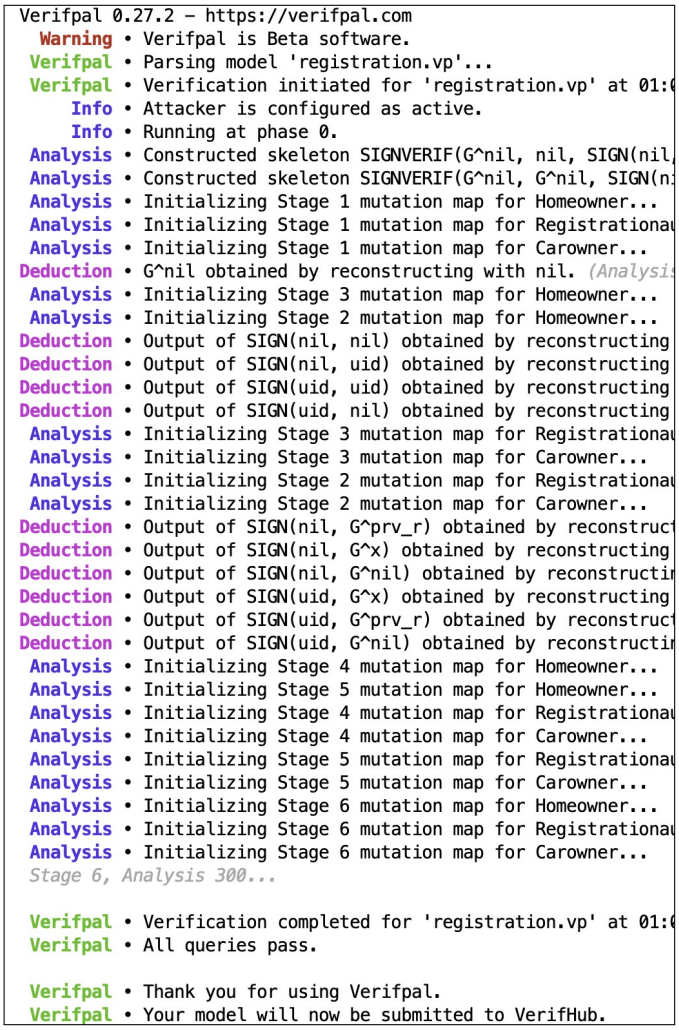

Fig. 3: The output log of `Registration.vp`. Verifpal runs multiple instances of the protocol and tries to mutate the attacks. At the end, it outputs the result of all the queries performed, and show the possible attack found, if any.

passive adversaries who can only observe the protocol as well as *active adversaries* who can modify, inject, or tamper with the exchanged messages. We made a few minor modifications to our protocol codes to tailor them to Verifpal's model. For instance, instead of using point multiplication over an elliptic curve for a Diffie-Hellman key exchange, which is not supported by formal verification tools, we used exponentiation over a finite cyclic group. This adjustment is not problematic, as the underlying discrete logarithm problem remains the same in both cases. Our Verifpal code does not include any counters since Verifpal is unable to check for inequality, greater/lesser than, or increment/decrement of variables. Instead of generating the entire hash chain $H(H(H(....H(r))))$, we hashed five times and used the fifth value to create the hash-based tokens and simulate the model. In further analyses, the fourth token, then the third token, and so on, were used.

We verified the protocols in the Dolev-Yao model, primarily testing and analysing our models for secrecy, authentication, spoofing attacks, replay attacks, trace, freshness, unlinkability, and equivalence properties. Trace properties are defined for each protocol run. The protocol satisfies such a property when it holds true for all traces. For example, the fact that some states are unreachable is a trace property. Equivalence properties mean that the adversary cannot distinguish between two processes. For instance, one of these processes can be the protocol under study, and the other can be its specification. Equivalence then implies that the protocol satisfies its specification. Equivalences can thus be used to model many subtle security properties. Our model passed all the tests and queries challenged to the verifier in both passive and active attacker modes, achieving 100% success in confidentiality, authentication, freshness, and unlinkability for all values and parameters. Therefore, given the correctness of the cryptographic primitives being used, our solution provides a secure key exchange protocol, with no confidentiality losses and with mutual authentication between all parties. Given Verifpal's goal of mimicking and resembling attacks on "real-world protocols", these outcomes support our claim that the protocol is secure against most known types of confidentiality and authentication attacks, such as man-in-the-middle attacks, replay attacks, and key compromise impersonation, across multiple unbounded protocol session executions.

VIII. Conclusion

The availability of parking spots in urban environments is an escalating concern, worsened by the continuous increase in the number of vehicles. Traditional solutions often fail to address the critical aspects of security and privacy, leaving users vulnerable. In this paper, we have proposed `PISA`, a novel Privacy-Preserving Smart Parking scheme designed to mitigate these issues through a secure and anonymous protocol. By employing a cryptographically secure scheme, it ensures bi-directional anonymity between vehicle owners and parking spot providers, thereby protecting the identities of both parties. This level of privacy is crucial in fostering trust and encouraging the adoption of such systems in urban settings. Our contributions are multifaceted. First, we introduced a comprehensive framework that facilitates the anonymous sharing of parking spots, effectively addressing the pressing need for more efficient use of existing parking infrastructure. Second, we utilised the formal verification tool Verifpal to rigorously prove the correctness and soundness of our security algorithms within the symbolic model. Additionally, we defined a set of security and privacy requirements that such solutions should ideally meet and evaluated how our model satisfies these properties.

Through detailed analysis and simulation, we demonstrated the practicality and effectiveness of `PISA` in real-world scenarios. Our results indicate that `PISA` not only enhances the availability of parking spaces but also maintains a high level of user privacy and security. This makes the model a viable solution for modern urban environments where privacy concerns are paramount. In conclusion, `PISA` sets a new standard

for privacy-preserving smart parking solutions by combining robust cryptographic techniques with formal verification. It addresses the dual challenges of parking scarcity and user privacy, offering a scalable and secure alternative to traditional parking systems.

ACKNOWLEDGMENT

This work was supported by CyberSecurity Research Flanders with reference number VR20192203.

REFERENCES

[1] "Shuffling the pack: optimising car parking in Amsterdam (The Netherlands)," https://www.eltis.org/discover/case-studies/shuffling-pack-optimising-car-parking-amsterdam-netherlands.

[2] "The future of European parking," https://www.easyparkgroup.com/news/the-future-of-european-parking-demand-for-electric-cars-and-free-parking-spots/.

[3] M. Bencekri, D. Ku, N. Sungyong, S. Lee, and S. Lee, "Parking policies review: Europe study case," *International Journal of Transportation*, pp. 1–10, 07 2019.

[4] K. W. Axhausen, M. Chikaraishi, and H. Seya, "Parking: Learning from japan," *Arbeitsberichte Verkehrs-und Raumplanung*, vol. 1095, 2015.

[5] "Yes, Parking in New York Has Gotten Worse," https://www.nytimes.com/2021/01/06/nyregion/nyc-parking.html.

[6] N. Kobeissi, G. Nicolas, and M. Tiwari, "Verifpal: Cryptographic Protocol Analysis for the Real World," in *INDOCRYPT*, 2020, pp. 151–202.

[7] W. A. Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmary, and K. Akkaya, "Privacy-preserving smart parking system using blockchain and private information retrieval," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2019.

[8] Z. Zinonos, P. Christodoulou, A. Andreou, and S. Chatzichristofis, "Parkchain: An iot parking service based on blockchain," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019.

[9] G. Brenner, M. Baza, A. Rasheed, W. Lalouani, M. Badr, and H. Alshahrani, "Dpark: Decentralized smart private-parking system using blockchains," *Journal of Grid Computing*, vol. 21, no. 3, 2023.

[10] C. Zhang, L. Zhu, C. Xu, C. Zhang, K. Sharif, H. Wu, and H. Westermann, "Bsfp: Blockchain-enabled smart parking with fairness, reliability and privacy protection," *IEEE Transactions on Vehicular Technology*, 2020.

[11] L. Xu, N. Shah, L. Chen, N. Diallo, Z. Gao, Y. Lu, and W. Shi, "Enabling the sharing economy: Privacy respecting contract based on public blockchain," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, ser. BCC '17, 2017.

[12] S. Kumar, Y. Hu, M. P. Andersen, R. A. Popa, and D. E. Culler, "JEDI: Many-to-Many End-to-End encryption and key delegation for IoT," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019.

[13] S. Duttagupta, D. Singelée, and B. Preneel, "T-hibe: A novel key establishment solution for decentralized, multi-tenant iot systems," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022.

[14] D. An, Q. Yang, D. Li, W. Yu, W. Zhao, and C.-B. Yan, "Where am i parking: Incentive online parking-space sharing mechanism with privacy protection," *IEEE Transactions on Automation Science and Engineering*, 2022.

[15] O. Tran Thi Kim, N. H. Tran, C. Pham, T. LeAnh, M. T. Thai, and C. S. Hong, "Parking assignment: Minimizing parking expenses and balancing parking demand among multiple parking lots," *IEEE Transactions on Automation Science and Engineering*, 2020.

[16] P. Yan, X. Cai, D. Ni, F. Chu, and H. He, "Two-stage matching-and-scheduling algorithm for real-time private parking-sharing programs," *Computers & Operations Research*, vol. 125, 2021.

[17] I. Chatzigiannakis, A. Vitaletti, and A. Pyrgelis, "A privacy-preserving smart parking system using an iot elliptic curve based security platform," *Computer Communications*, vol. 89-90, 2016, internet of Things Research challenges and Solutions.

[18] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Privacy-preserving smart parking navigation supporting efficient driving guidance retrieval," *IEEE Transactions on Vehicular Technology*, 2018.

[19] "Prked," https://prked.com/.

[20] "JustPark," https://www.justpark.com/.

[21] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, 2008.

[22] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: implementation, performance, and research challenges," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 110–118, 2008.

[23] D. J. Bernstein, "Curve25519: New diffie-hellman speed records," in *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography*, ser. Lecture Notes in Computer Science, vol. 3958. Springer, 2006, pp. 207–228. [Online]. Available: https://iacr.org/archive/pkc2006/39580209/39580209.pdf

[24] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," in *CHES*, ser. Lecture Notes in Computer Science, vol. 6917. Springer, 2011, pp. 124–142. [Online]. Available: https://www.iacr.org/archive/ches2011/69170125/69170125.pdf

[25] N. van Saberhagen, "Cryptonote v 2.0," in *Monero Whitepaper*, 2013.

[26] E. Ben Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, 2014.

[27] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "Blake2: Simpler, smaller, fast as md5," in *Applied Cryptography and Network Security*, 2013, pp. 119–135.

[28] M. Bellare and P. Rogaway, *Introduction to Modern Cryptography*. UCSD, 2005.

[29] "Encryption in Zoom calls verified by NGI-supported Verifpal," https://nlnet.nl/news/2020/20200611-SecureVideochatVerifpal.html.

[30] S. P. Rao and A. Bakas, "Authenticating mobile users to public internet commodity services using sim technology," in *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '23, 2023.

[31] O. Basem, A. Ullah, and H. R. Hassen, "Stick: An end-to-end encryption protocol tailored for social network platforms," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[32] T. Lauser, D. Zelle, and C. Krauß, "Security analysis of automotive protocols," in *Proceedings of the 4th ACM Computer Science in Cars Symposium*, ser. CSCS '20, 2020.

[33] S. Duttagupta, E. Marin, D. Singelée, and B. Preneel, "Hat: Secure and practical key establishment for implantable medical devices," in *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '23, 2023.

[34] M. Nakkar, R. AlTawy, and A. Youssef, "Gase: A lightweight group authentication scheme with key agreement for edge computing applications," *IEEE Internet of Things Journal*, 2023.

[35] J. Zhang, L. Yang, X. Gao, G. Tang, J. Zhang, and Q. Wang, "Formal analysis of quic handshake protocol using symbolic model checking," *IEEE Access*, 2021.

[36] S. Paul, P. Scheible, and F. Wiemer, "Towards post-quantum security for cyber-physical systems: Integrating PQC into industrial M2M communication," Cryptology ePrint Archive, Paper 2021/1563, 2021, https://eprint.iacr.org/2021/1563. [Online]. Available: https://eprint.iacr.org/2021/1563

APPENDIX A
FORMAL VERIFICATION CODE

Listing 1: PISA – Registration Phase

```
attacker[active]

principal HomeOwner[
   knows public location
   knows public ID
]

principal RegistrationAuthority[
   generates prv_R
```

```
   pub_R = G^prv_R
]

RegistrationAuthority -> HomeOwner: [pub_R]
RegistrationAuthority -> CarOwner : [pub_R]

principal RegistrationAuthority[
   knows private UID
   sign1 = SIGN(prv_R, UID)
]

RegistrationAuthority -> HomeOwner: UID, sign1

principal CarOwner[
   knows public car_name
   knows public vehicular_details
]

principal HomeOwner[
   _ = SIGNVERIF(pub_R, UID, sign1)
]

principal RegistrationAuthority[
   generates x
   pub_C = G^x
   sign2 = SIGN(prv_R, pub_C)
]

RegistrationAuthority -> CarOwner: pub_C, sign2

principal CarOwner[
   _ = SIGNVERIF(pub_R, pub_C, sign2)
]

// Queries
queries [
   // Confidentiality queries
   confidentiality? prv_R
   confidentiality? location
   confidentiality? car_name
   confidentiality? vehicular_details
   confidentiality? x

   // Mutual authentication queries
   authentication? RegistrationAuthority -> HomeOwner: sign1
   authentication? RegistrationAuthority -> CarOwner: sign2

   // Freshness queries
   freshness? sign1
   freshness? sign2
   freshness? location
   freshness? car_name
   freshness? vehicular_details
   freshness? x
]
```

Listing 2: PISA – Protocol Phase

```
attacker[active]

principal HomeOwner[
   knows public UID
   knows public location
   knows public time
   generates alpha
   generates Priv_H
   Pub_H = G^Priv_H
]

principal CarOwner[
   generates x
   Pub_key = G^x
   sign1 = SIGN(x, Pub_key)
]

principal ServiceProvider[
   knows public price
   generates ack
   generates Priv_S
   Pub_S = G^Priv_S
]

HomeOwner -> ServiceProvider: [alpha], [Pub_H]
ServiceProvider -> HomeOwner: [ack], [Pub_S]
ServiceProvider -> CarOwner: [Pub_S]
CarOwner -> ServiceProvider: [Pub_key], sign1

principal ServiceProvider[
   _ = SIGNVERIF(Pub_key, Pub_key, sign1)?
   generates beta
   msg1 = PKE_ENC(Pub_key, beta)
]

ServiceProvider -> CarOwner: [msg1]

principal CarOwner[
   ack2 = PKE_ENC(Pub_S, PKE_DEC(x, msg1))
]

CarOwner -> ServiceProvider: [ack2]

principal ServiceProvider[
   _ = ASSERT(beta, PKE_DEC(Priv_S, ack2))?
]

principal CarOwner[
   generates k
   sign2 = SIGN(x, k)
   msg2 = PKE_ENC(Pub_S, k)
]

CarOwner -> ServiceProvider: [msg2], sign2

principal ServiceProvider[
   _msg2 = PKE_DEC(Priv_S, msg2)
   _ = SIGNVERIF(Pub_key, _msg2 , sign2)
   generates r
   hash1 = HASH(r)
   hash2 = HASH(hash1)
   hash3 = HASH(hash2)
   hash4 = HASH(hash3)
   hash5 = HASH(hash4)
   sign3 = SIGN(Priv_S, hash5)
   msg3 = PKE_ENC(Pub_key, alpha)
   msg4 = PKE_ENC(Pub_H , _msg2)
]

ServiceProvider -> CarOwner: [msg3], hash1, sign3
ServiceProvider -> HomeOwner: [msg4]

principal CarOwner[
   _hash2 = HASH(hash1)
   _hash3 = HASH(_hash2)
   _hash4 = HASH(_hash3)
   _hash5 = HASH(_hash4)
   _ = SIGNVERIF(Pub_S, _hash5, sign3)
   msg5 = ENC(k, PKE_DEC(x, msg3))
]

CarOwner -> HomeOwner: msg5, _hash5, sign3

principal HomeOwner[
   _k = PKE_DEC(Priv_H, msg4)
]

// HomeOwner -> CarOwner: acknowledgements will follow
// HomeOwner -> ServiceProvider: challenge-response will
     follow

queries [
   // Confidentiality queries
   confidentiality? x
   confidentiality? k
   confidentiality? _k
   confidentiality? UID
   confidentiality? location
   confidentiality? time
   confidentiality? Priv_H
   confidentiality? Priv_S

   // Mutual authentication queries
   authentication? CarOwner -> ServiceProvider: sign1
   authentication? CarOwner -> ServiceProvider: Pub_key
   authentication? CarOwner -> ServiceProvider: ack2
   authentication? ServiceProvider -> CarOwner: msg1
   authentication? CarOwner -> ServiceProvider: msg2
```

```
   authentication? CarOwner -> ServiceProvider: sign2
   authentication? ServiceProvider -> CarOwner: msg3
   authentication? ServiceProvider -> CarOwner: sign3
   authentication? ServiceProvider -> HomeOwner: msg4

   // Freshness queries
   freshness? x
   freshness? k
   freshness? _k
   freshness? beta
   freshness? sign1
   freshness? sign2
   freshness? sign3

   // unlikability queries
   unlinkability? x, k, _k, Priv_H, Priv_S
]
```