Sustainable and round-optimized group authenticated key exchange in vehicle communication

Sustainable and Round-Optimized Group Authenticated Key Exchange in Vehicle Communication

Zengpeng Li, Mei Wang, Vishal Sharma, and Prosanta Gope

***Abstract*—Vehicle authentication is an essential component validating the vehicle's identity and ensuring the integrity of transformed data for intelligent transport vehicles (ITS) in the vehicular ad hoc network (VANET). Easy to deploy and operate privacy-enhancing vehicle authentication mechanisms are the mainstay for the widespread ITS in the VANET. Very recently, VANET security architectures are constituting by IEEE 1609.2 group, NoW project, the SeVeCom project. However, these approaches heavily depend on the consuming public key infrastructure (PKI) and certification authorities (CA). In this work, walking along the research line, we attempt to design authentication protocols with two diverse factors for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) networks, respectively, without depending on the stumbling block PKI/CA. In addition, a smooth projective hash function (SPHF) (a.k.a., a special case of the designated-verifier zero-knowledge proof system) guarantees any recipient can confirm the authenticity and integrity of the received messages without knowing the authentication factors. Thus, to optimize the communication round, SPHF is used to design a (group) two-factor authenticated key exchange (AKE) with low-interactive communication rounds. The proof-of-concept implementation indicates that the computation and communication overheads introduced by our solution are acceptable in real-world deployments. The security of the proposed approach is validated using Bellare-Pointcheval-Rogaway (BPR) model along with the experimental evaluation and the theoretical analysis.**

***Index Terms*—Intelligent Transport Vehicle; Vehicular Ad Hoc Network; Two-Factor Authentication; Two-Factor Authenticated Key Exchange.**

I. Introduction

Industry 4.0 is a digital transformation project that was launched by Germany in 2011 and is widely referenced in Europe (ISP-4IR) [1]. Intelligent Transport System (ITS) technology is an integration of the industrial internet of things (or industrial IoT). The essential duty of ITS is to manage the deployment of cooperative networks among the vehicles and the communication between the vehicles and the fixed infrastructures. Normally, the former setting is abbreviated to the Vehicle-to-Vehicle (V2V), and the latter one is abbreviated to the (Vehicle-to-Infrastructures (V2I)), the V2I and V2X setting are collectively known as V2X [2], [3].Indeed, various ITS applications are proposed successively for V2V networks and V2I networks, respectively, such as [4], [5]. Particularly, in the V2V networks, the vehicles equipped with the onboard unit (OBU) exchange messages under the support of the Global Navigation Satellite System (GNSS). In V2I networks, it is necessary for the public infrastructure (*e.g.*, road-side unit or RSU) and the vehicles and their users to communicate, for example, to share the news about road conditions and vehicle trajectory. Further, some applications require information exchange between the vehicles and the fixed infrastructure in V2V and V2I networks. An illustration of connected applications and the relation to V2V and V2I is shown in Figure 1.

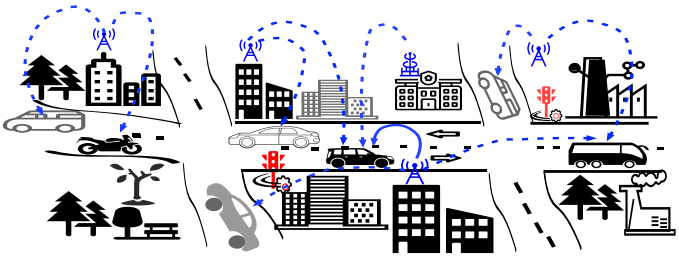

Figure 1: V2X Networks Enabled by Industrial Internet of Things

Nowadays, the global demand for ITS equipment in the vehicular ad hoc network is growing rapidly [6], [7]. The importance of ITS is also reflected in the resources major technology brands like Tesla, NIO, and BYD are pouring into enterprise all-electric clean energy platform development. Regrettably, the promising ITS technology lacks security measures that make it vulnerable to sorts of cyberattacks. In that case, to enhance the quality of experience for the vehicle drivers and passengers without fear for their safety and personal privacy, the intelligent security and privacy-preserving authentication protocol as the first line of defense are suggested as the candidate. As we know, authentication would be addressed easily if the problem of privacy is not regarded as an issue. In this case, each vehicle has the ability to sign messages simply using its corresponding certified private signing key. However, the main drawback is that the malicious participant is enabled to trace vehicles using the corresponding public keys throughout the road network. In addition, all vehicles would use the same signing key provided by the PKI/CA, assuming key compromise is not a weak point.

Z. Li is with the School of Cyber Science and Technology, Shandong University (Qingdao Campus) and Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University (Qingdao Campus), China (Email: zengpeng@email.sdu.edn.cn).

M. Wang is with the School of Cyber Science and Engineering, Wuhan University, China (Email: wangmeiz@whu.edu.cn).

V. Sharma is with the Queen's University Belfast, United Kingdom (Email: v.sharma@qub.ac.uk).

P. Gope is with the University of Sheffield, United Kingdom (Email: p.gope@sheffield.ac.uk)

However, anyone's vehicle is compromised, and the private signing key will reveal to the attackers. Thus, fresh keys have to provide to the vehicles. Further, authenticated vehicles would be more secure to establish the session key if authenticated vehicles are legitimate users. Nevertheless, these approaches heavily depend on the consumption PKI.

To our knowledge, password (or multiple factors) based authentication solutions are the primary choice as Internet-scale authentication. They are even best for IoT-scale vehicle authentication in the Industry 4.0 era. In addition, we note that most of the existing (password) authentication protocols for V2V communications require a multi-path communication environment so that it can be separated into two types. One is to design end-to-end authentication, and another is to design group authentication. The end-to-end authentication is flexible in establishing the session key between two different partners, which is easy to extend the group authentication. Very recently, some practical asymmetric password-authenticated key exchange (or PAKE) protocols [8], [9] with random oracles are proposed. However, cryptographers [10] pointed out some limitations of random oracle when using it to make the reduction. Thus, how to design the cryptographic primitives without using random oracles has aroused high interest, and we would ask a natural question,

> *Is it possible to design a round-optional and more secure two-factor authentication based on several theoretical assumptions in the V2X networks?*

Inspired and motivated by recent single-factor, two-factor, and multi-factor authentication works, such as password harden services [11], [12], PAKE [9], [13], [14], two-factor authentication and two-factor AKE [15], device-enhanced password protocol [16], we consider to design a more secure two-factor authentication for V2X (*e.g.*, V2V and V2I) networks. We note that Haase and Labrique [9] gave a practical solution of the asymmetric-PAKE for industrial IoT. Following this line of the research problem, we give a completely different approach to obtaining asymmetric two-factor authenticated key exchange for V2I networks.

As shown in Figure 2, to systemize the two-factor authentication and two-factor AKE for V2I network, our proposed solutions are the secure composable protocols that contains three phases: **1).** *two-factor registration*, **2).** *(asymmetric) two-factor authentication* and **3).** *(two-factor authenticated) key-establishment phase, namely two-factor AKE protocol*. Our design contains three achievements: **1).** it prevents the password from exposing to the server directly by the transformation from a password to a randomized password, **2).** it achieves two-factor key establishment based on the smooth projective hash function for V2V network and V2I network respectively. **3).** it guarantees the privacy of authentication-factor of the vehicle clients using the non-interactive zero-knowledge (NIZK) proof.

A. Related Works

PAKE for V2X. V2X (*e.g.*, V2V and V2I) networks need challenging requirements to achieve the desired reliability, privacy, and security. Further, conventional regular IoT security solutions cannot be used directly to address the industrial IoT security flaws because industrial IoT devices [17], [18] have their associated natures. Very recently, various lightweight authentication solutions, including authenticated key exchanges (AKE) protocols PAKE protocols, are proposed in new customization to achieve the adapted security and privacy for industrial IoT. These kinds of authentication and key establishment protocols enable to perform the remote authentication securely with short low-entropy passwords while preventing know attacks no matter passive and active.

Notably, most of the existing secure remote password (or SRP) protocols are referred to as password authentication and PAKE [19], [8], which evidently is the typical representative of single-factor authentications. However, simple password-based authentication is an insecure way to authenticate devices in industrial IoT. Although multi-factor schemes may introduce more frictions which will decrease productivity and reduce user adoptions. Various schemes are proposed successively, such as two-factor authentication [16], two-factor AKE [15], multi-factor authentication [20], and multi-factor AKE [21], [22], [23], to make it more difficult for cybercriminals to breach our account.

Group PAKE for V2X. Abdalla and Pointcheval [24] proposed the first scalable group PAKE protocol without random oracles using the smooth projective hashing functions. Afterward, Abdalla *et al.*, [25] formalized group PAKE with the universal composable framework. Very recently, various optimizations are proposed in succession [26], [27]. Furthermore, an important application to V2X has aroused much concern, and various [28], [29]. However, these existing solutions are under random oracles with some uniform hashing functions. How to bypass the usage of random oracles for authentication in the V2X setting is an interesting open question.

B. Overview Contributions and Techniques

Compared with most of the existing solutions, which used hash functions as the building blocks with random oracles, the main difference is that we use the SPHF to realize two different solutions in the standard model for V2X. Below, we conclude our main achievements.

- **Round-Optimized Abdalla and Pointcheval Group AKE**. Abdalla and Pointcheval proposed an efficient group AKE protocol but with at least six communication rounds. However, we observed that communication rounds could be reduced one round by merging the SPHF hash key generation and projection key generation, and we optimize the original solution [24] and avoid verifying the correctness and legality at the end of each round. On the contrary, in our optimized solution, the parties only validate the legitimacy of the received message before the session key generation. Further, to enhance the security level, we upgrade the protocol with two authentication factors armed with oblivious pseudorandom function, that optimized solution is used in our following symmetric two factor authentication for V2V.
- **Symmetric Two-Factor Authentication for V2V**. In our vehicle-to-vehicle authentication setting, we require the

two (and multiple) legitimate vehicles with two authentication factors to establish the session key without utilizing random oracle models. In previous strategies, a designated-verifier proof system based on the number theorem is used so that several rounds of interactions are required for this construction [1]. It is not needed at all that each vehicle client proves to the others that he owns legitimate authentication factors. We only guarantee that the same authentication factors are known by both parties, then the same strong session key will be generated by them respectively. Hence, SPHF designed in the standard model is introduced to guarantee that any recipient vehicles can confirm the integrity and authenticity of transactions without knowing authentication factors.

- **Asymmetric Two-Factor Authentication for V2I**. An observation is that most of the existing authentication and authenticated key establishment proposals are provided under the random oracle model with idealized assumption [22], for instance, hash functions *e.g.* HMAC. In contrast to the symmetric two-factor authentication used for V2V authentication aforementioned the first contribution point, we give an asymmetric two-factor authentication along with two-factor AKE protocol for the vehicle to infrastructure authentication in the standard model using an exactly different approach. The methodology of PAKE inspires us over hash proof systems [30], [31], [13], and we utilize the SPHF to execute the two-factor authentication and generate the session key once the authentication is accomplished.

C. Paper Organization

In Section III we review related notions. In Section II, we describe the problem formally and outline out a solution with the security and communication models. In Section IV, we detail our two-factor authentication and key establishment protocol based on SPHF for the V2V communication. In Section V, we detail our solution for the V2I communication. In addition, we analyze the security and correctness of our proposed two-factor authentication protocol and authenticated key establishment. Finally, Section VII, concludes the article. Further, for easy reference, in Table I we provide a summary of the main terms and acronyms used throughout the paper.

Table I: Glossary.

Acronym	Description
AKE	Authenticated Key Exchange
TFA	Two-Factor Authentication
PAKE	Password-Authenticated Key Exchange
UH	Universal Hash Function Family
SPHF	Smooth Projective Hash Function (HashKG, ProjKG, Hash, Proj)
OBU	On-Board Unit
RSU	Road-side Unit

[1]Intuitively, each player could convince others that the correct authentication factors are committed and transferred. Then the two parties could execute a standard key exchange program after validating proofs.

II. System Model

A. System Model

The system model used in the proposed work is illustrated in Figure 2 that shows how IIoT devices establish a session key with multiple factors.

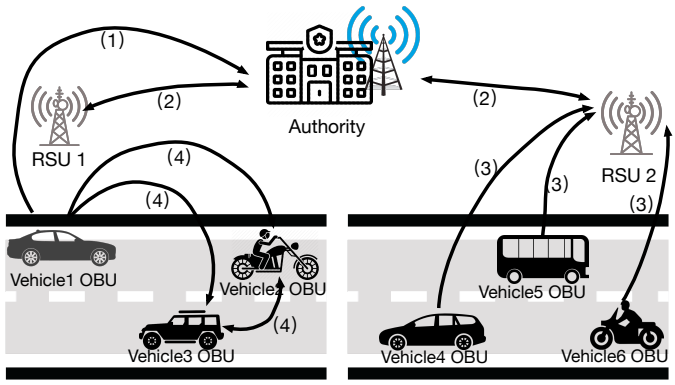

Figure 2: V2X Authentication: (1). Vehicles are registered before leaving the factory; (2) RSUs are registered before deploying on the road; (3) Vehicles controlled by the same RSU are first to synchronize the authentication factors with the RSU, and OBU enabled vehicle and RSU establish the session key; (4) Authenticated vehicles establish the session key between two different vehicles.

Entities. We sketch the considered scenario, giving an idea of the required flexibility and complexity of our framework. We target a Publicly Verifiable Vehicle Trajectory for Intelligent Transportation Systems. It involves the following entities:

- On-Board Unit (OBU) is an important component for the V2X, and the main duty of OBU is to transfer on-road messages during the period of the debt and dispute clearance. In reality, a OBU (*e.g.*, smartphone) is used alternatively to connect the service provider via 4G or the advanced 5G. Notably, each OBU is equipped with a tamper-proof device (TPD) that is used to store some sensitive records, for example, secret keys, location coordinates along with time provided by the Global Positioning System (GPS), and vehicle crashes recorded by an event data recorder.
- Road-Side Unit (RSU) is deployed on the road-side by the service provider as the fixed infrastructure, and RSU bridge the connection between the trusted authority by securing wire links, and it also enables to interact with the vehicle users who are integrated with OBUs by a wireless channel. If there is not much traffic in a couple of duration, then RSUs exchange data periodically for fraud detection with the service provider.
- Vehicle and Vehicle Clients. Informally, vehicles armed with OBUs are named as intelligent transport vehicles because the vehicle client enabled with a (portable or mounted) OBU has the ability to communicate with other OBUs and RSUs in a fixed domain.
- Service Provider (SP). In the whole transportation system, SP equipped with the PKI/CA is responsible for maintaining all of the communications of entities and would divide the transportation system into several geographic regions for convenience. In addition, SP is regarded as a

fully trusted entity that is infeasible to compromise for any opponent. In reality, SP might be a privately owned company. Thus, its main duty is to manage the registration of RSUs and OBUs.

Two kinds of communication. Vehicular wireless communications technologies are used by the vehicles to communicate among them (*i.e.*, V2V) while communicating with the roadside infrastructure (*i.e.*, Vehicle to RSU, or V2I), which is enabled to service a wide range of intelligent transport system (ITS) applications. In reality, each RSU and OBU could access the wireless channel in a directional or a unidirectional antenna, which is carried out via the Dedicated Short Range Communications (DSRC) standard radio (an important of IEEE 802.11p radio technology). For example, if a sensitive message will be transmitted by an RSU to a specific location, then a unidirectional antenna is recommended in this setting. Below, we present two types of communications in VANETs.

- V2V communication network, it means the moving vehicles enable to communicate with each other, in that case, we provide a solution as shown in Section IV and Figure 3.
- V2I communication network, it means the moving vehicles enable to exchange with the RSUs which are deployed aside the roads, we also provide a solution as shown in Section V.

Indeed, V2V and V2I communications are executed via an open wireless channel that is vulnerable to various attacks, such as interference, eavesdropping, and jamming, *etc*. These attacks are out of the scope of this work, so we omit V2V and V2I communications' security.

B. Problem Statements

A series of authentication limitations are pointed out recently, and the primary limitation is the RAM memory capability of industrial IoT devices. Indeed, most of these devices cannot satisfy the requirement of fast, re-writable, and non-volatile data storage. Additionally, it is hard to prohibit illegitimate access from the service provider while preventing malicious parties from accessing the sensitive resources and services at remote servers. An important observation is that most of these existing solutions, such as [21], [22], [23], cannot guarantee sufficient security along with practical efficiency for the two-factor AKE because they are based on the simple ElGamal encryption [32]. Hence, how to address or bypass these limitations is becoming a valued research topic.

To our knowledge, various authentication proposals are provided. However, most of the existing authentication approaches for V2V and V2I, the service provider equipped with PKI/CA requires the vehicle to send the corresponding low-entropy authentication factors (*e.g.*, password pw and randomness r (as the second authentication factor)) in cleartext with the associated identity. Then the pair of the identity and the derived hash value from the received factors are stored in the form of $\langle \mathsf{uid}, \mathsf{hash}(\mathrm{pw}, r)\rangle$. Notably, during the registration phase, one of the drawbacks of this conventional technique is that the service provider enables to operate the authentication factors without the client's permission because authentication factors are transferred in plaintext. To overcome these limitations, authentication factors of the vehicle client are blinded before sending to the service provider during the registration phase. In contrast, the service provider could validate these factors usually, and we detail the technique in Subsection V-A1.

In addition, RSUs are deployed by the service provider in different geographic regions, and then the legitimate RSUs are enabled to communicate with the service provider who stores the identity information of the vehicles and vehicle clients. In this setting, the public parameters of each RSU are interchanged with the RSU in different geographic regions, these kinds of RSU are used to validate the vehicles' legitimacy. In a nutshell, when a vehicle enabled with the OBU moves from one geographic region to another one, the vehicle would authenticate herself to the nearest RSU whenever the vehicle wants to get the service of ITS, and it will be authenticated by the nearby RSU in the new geographic region depending on the information provided by the service provider.

Another important observation is that various prevalent attacks, *e.g.*, man-in-the-middle and dictionary attacks, *etc*, can be launched by the malicious to steal the authentication factors stored at the service provider side or RSU. Further, when the vehicles want to communicate with the nearest RSU with the secure session channel, the vehicle has to convince the RSU that the vehicle is holding the correct authentication factors. Meanwhile, the RSU has to inquire the service provider, and the knowledge of the corresponding OBU's authentication factors are answered to the RSU by the service provider. Thus, exploring a new technique to guarantee the privacy of the authentication factors is a challenging problem. Below, as described in Subsection V-A2, to prevent the offline dictionary attacks, we use the hashing function with salt to store the salted authentication factors.

Finally, we show the V2I authentication protocol (including registration, login-authentication, and two-factor AKE) in Subsection V-A3 to generate a secure session key, and the proposed V2I authentication protocol combines user identities and desired services like two-factor authentication and two-factor AKE.

III. Preliminaries

A. Cryptographic Building Blocks

Below, we detail the building blocks associated with its corresponding instantiations.

Definition III.1 (Decisional Diffie-Hellman (DDH) problem)**.** *It implies that no probabilistic polynomial time (PPT) adversary has the ability to distinguish computationally (in the security parameter $\lambda = \|q\|$) the following two distributions over a q-order group $\mathbb{G}$ with a generator g:*

- (g, g^a, g^b, g^{ab}) *where we sample randomly a and b from $\mathbb{Z}_q$.*
- (g, g^a, g^b, g^c) *where we sample randomly $a, b, c \leftarrow \mathbb{Z}_q$.*

Universal Hash Function Families ($\mathcal{UH}$s). A universal hash function family $\mathcal{UH}$ is a map $\mathcal{K} \times \mathcal{G} \to \mathcal{R}$, where $\mathcal{K}$, $\mathcal{G}$, and $\mathcal{R}$ are denoted as the key (or seed) space, the domain, and the range. Particularly, let $\mathrm{UH}_{\mathrm{k_{UH}}}$ be a universal hash function with the fixed key $\mathrm{k_{UH}}$ drawn from $\mathcal{UH}$, and $\mathrm{UH}_{\mathrm{k_{UH}}} : \mathcal{G} \to \mathcal{R}$

is used in the hashing computation. Further, if an element g is drawn uniformly from $\mathcal{G}$, then the output of $\mathrm{UH}_{\mathrm{k_{UH}}}(g)$ is statistically close to the uniform in $\mathcal{R}$.

Smooth Projective Hash Functions (SPHFs). SPHFs are the special case of the designed-verifier zero-knowledge proof system, which is first introduced in [33]. In this setting, the verifier first generates the private hashing key $hk \leftarrow \mathsf{HashKG}(1^\lambda)$ and obtains the public projective key $ph \leftarrow \mathsf{ProjKG}(hk, \mathcal{L})$ by taking a hk and the language L, where $L \subset X$. Then the verifier broadcasts the ph. Upon receiving the ph, the prover is with the pair of statement and witness (wrd, w) as input under the received ph, and outputs $p \leftarrow \mathsf{Proj}(ph, L, \mathrm{wrd}, w)$. Additionally, SPHFs are with two important properties, and one is a projection (or correctness) which means the Hamming distance between the output of public hash and the output of the private hash is negligible. Formally, the verifier computes $h \leftarrow \mathsf{Hash}(hk, L, \mathrm{wrd})$ by inputting a private hk and the corresponding word wrd, and validates whether $h = p$ that is guaranteed by the property of projection (or correctness). Another one is smoothness that means the output of $\mathsf{Hash}(hk, L, \mathrm{wrd})$ is independent of ph. Finally, In particular, for any word wrd in the domain $L(\subset X)$, then we have the result of $h = p$ if there exists a witness $w \in L$. Otherwise, no existing witness satisfies $\mathrm{wrd} \in X \setminus L$.

Cramer-Shoup Public-Key Encryption. Cramer-Shoup encryption is a great example of a chosen-ciphertext attack (or CCA) secure scheme to instantiate our proposed symmetric and asymmetric two-factor authentications and two-factor AKE protocols for V2V and V2I settings. Below, we revisit it over a p-ordered group $\mathbb{G}$ of with two different generators g and h.

- $(pk_{\mathsf{CS}}, sk_{\mathsf{CS}}) \leftarrow \mathsf{CS.KeyGen}(\mathbb{G}, p, \mathbb{Z}_p)$: Inputs two independent g and h over a group $\mathbb{G}$ with the a p, then outputs $sk = (\alpha_1, \alpha_2, \beta_1, \beta_2, \rho)$ by sampling randomly five random scalars from $\mathbb{Z}_q$ and $pk = (g, h, c = g^{\alpha_1} h^{\alpha_2}, d = g^{\beta_1} h^{\beta_2}, f = g^\rho, H)$. Here, H is denoted as a random collision-resistant hash function.
- $\mathbf{c} \leftarrow \mathsf{CS.Enc}(pk_{\mathsf{CS}}, m)$: To output the encryption of $m \in \mathbb{G}$, a label ℓ is created firstly regarding the identity of the encryption entity and the session id for each encryption. Next, the algorithm picks a randomness $r \leftarrow \mathbb{Z}_p$ and obtains $c_1 = g^r$, $c_2 = h^r$, $c_3 = f^r \cdot m$. Then the algorithm computes $\Theta = H(\ell, c_1, c_2, c_3)$ and $\phi = (cd^\Theta)^r$. Finally, the algorithm outputs the ciphertext $\mathbf{c} = (c_1, c_2, c_3, \phi)$.
- $m := \mathsf{CS.Dec}(sk_{\mathsf{CS}}, \mathbf{c})$: Parses the inputed $\mathbf{c}$ into c_1, c_2, c_3, ϕ, then it validates whether it satisfies the equation $\phi \stackrel{?}{=} \mu^{\alpha_1 + \Theta\beta_1} \cdot v^{\alpha_2 + \Theta\beta_2}$. Finally outputs the decrypted $m := c_3 / c_1^\rho$, otherwise, outputs a $\perp$.

The correctness is easy to validate, so we omit it here, and the security is guaranteed in Theorem III.2.

Theorem III.2. *Cramer-Shoup encryption is against* IND-CCA *if the* DDH *assumption is hard.*

Schnorr's Protocol for Discrete Logarithm. Schnorr's protocol aims to prove that the verifier V knows the discrete logarithm $w = \log_g h$, where $\mathbb{G}$ be a q-order group with a g. Here, the prover P and the verifier V have the same statement $h \in \mathbb{G}$, and the prover P has the private witness w such that $h = g^w$. In this setting, the prover P first picks a random r to mask the witness w and sends the first flow $a = g^r$ to V. The verifier next samples a challenge $e \in \{0,1\}^t$ to answer to the prover. Subsequently, the prover P forms the third flow $\pi = ew + r \pmod q$ and sends it to V. Finally, V validates that $g^\pi = ah^e$. The correctness can be validated easily by $g^\pi = g^{r+ew} = g^r (g^w)^e = ah^e$.

B. Security/Threat Model

To guarantee efficiency, a weaker version of the Bellare-Pointcheval-Rogaway (BPR) model [34] is introduced. We bypass the traditional *Find-then-Guess* (FaG) game, which uses the indistinguishability methodology. In a nutshell, in the FaG game, we assume no adversary could tell the difference a real session key from a random one with an advantage significantly more significant than q_S/N, where the number of active sessions is denoted as q_S, and the size of the dictionary is denoted as N. Then we walk along with the line of *Real-or-Random* (RoR) scenario, in this setting, Reveal-query does not require, but it requires multiple Test-queries. Notably, the RoR model requires the session keys to meet two important properties,

1) the session keys should be independent of each one in different Test-queries,
2) the session keys should be indistinguishable globally from the random.

Then we illustrate the the adversary's behaviors using the following oracles:

- Execute-query models the *passive* attack executions.
- Send-query models the *active* attack executions.
- Corrupt-query is used to explore the property of *forward-secrecy*, and it could model the corruptions and the secrets' leakage.
- Test-query is used to output the results, either a real one or a random one, that models, in essence, the semantic security of the session key.

Additionally, both parties (*i.e.*, the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$) are interacted to illustrate: at the beginning of the game, the challenger $\mathcal{C}$ picks a random bit $b \in \{0,1\}$ to the adversary $\mathcal{A}$ when answering Test-queries. Notably, $b = 1$ means that $\mathcal{C}$ provides real keys to $\mathcal{A}$. Otherwise, random keys to $\mathcal{A}$. Subsequently, $\mathcal{A}$ is enabled to interact with the instances of the targeted protocol by using the above-defined Execute, Send, Corrupt, and Test oracles. Finally, $\mathcal{A}$ picks a random bit $b' \in \{0,1\}$, and checks if $b = b'$, the adversary $\mathcal{A}$ wins the game when $b = b'$, otherwise, $\mathcal{A}$ loses.

IV. Our Solution: Round-Optional Two-Factor Authentication and Key Establishment for V2V

In V2V authentication with requiring a multi-path communication environment, regarding the end-to-end authentication setting, the the legitimate client has the ability to know the content of the messages without maintaining the confidentiality of each message. However, in this strategy, we consider the authentication between two different OBUs. Thus, SPHF is introduced to guarantee that any receiver is enabled to confirm the authenticity and integrity of the received messages.

A. Two-Factor Authentication and Key Exchange for V2V

1) Vehicle Registration Phase: Notably, in the vehicle-to-vehicle setting, we introduced the two-factor authentication mechanism that removes the PKI/CA dependence. Thus, the vehicles have been issued identity to the unique OBU before leaving the factory. Further, when vehicle users pick up the new vehicle, they are obliged to generate a unique identity uid for the vehicle by using the registered information, *e.g.*, name, driver's license, email, phone number, *etc*. After the registration phase, the vehicle users are regarded as VANET legitimate users.

More concretely, all of RSUs first register to the service provider. Then the legitimate RSUs who have passed the service provider's validation will establish a secure communication channel with the service provider. In addition, the RSUs will store the index of the registered vehicles equipped with the OBU, so that the vehicles could contact the nearby RSU when the legitimate vehicle goes into the domain controlled by the RSU. Then the vehicles and RSU first initializes parameters, *e.g.*, public keys, and then they execute two-factor AKE to establish the shared session key.

To facilitate the presentation of our protocol, we only focus on establishing the session key and ignore the registration phase. In this setting, an essential assumption for the registration, it happens in a reliable and secure environment. A vehicle user U who has the device $\mathcal{D}_{\mathsf{dvc}}$ interacts with RSU as below:

- The vehicle client first creates the unique vehicle identity uinfo according to different scenarios. Next, the vehicle client types in his password pw to the special physical device $\mathcal{D}_{\mathsf{dvc}}$, and a new randomized password rpw along with a one-time passcode (OTP) $\mathsf{otp} \leftarrow \mathbb{Z}_p^*$ will respond to the vehicle client.
- To register to the RSU, the vehicle client then sends a unique identifier uid with the corresponding quarter message $(\mathsf{rpw}, \mathsf{otp}, \mathsf{uinfo})$ to complete the registration.
- Once received the uid along with the triple $(\mathsf{rpw}, \mathsf{otp}, \mathsf{uinfo})$, the RSU concatenates rpw and otp to create $\theta = \mathsf{rpw} \| \mathsf{otp}$, then the RSU computes $\vartheta = H_p(\theta, s)$ under the salt s, and finally he calculates $Z = g^{\vartheta}$, where H_p is onto $\mathbb{Z}_p$.
- Later on, the RSU forwards uid along with $(\mathsf{rpw}, \mathsf{otp}, \mathsf{uinfo})$ to the service provider and checks whether there is a record in the service provider. If there is a record, then the service provider will let the RSU know. Otherwise, the RSU will insert $(id, (\mathsf{rpw}, \mathsf{otp}, \mathsf{uinfo}))$ into the local database.

2) Authentication Phase: The authentication phase happened at which the vehicle user C_1 (with the unique identity uid_1) authenticates his partner vehicle user C_2 (with the unique identity uid_2). During the authentication phase, two peer vehicle clients C_1 and C_2 first input his (or her) passwords pw and fetch rpw using their device, respectively. Then they share the authentication factors and validate if the received authentication factors match the locally stored authentication factors. Finally, they create $\theta' = \mathsf{rpw} + \mathsf{otp}$, then they calculate $Z' = g^{\vartheta'}$ and $\vartheta' = H(\theta', s)$ separately for the salt s.

3) Two-Factor Authenticated Key Establishment Phase: After finishing the phase of authentication, the two-vehicle clients are going to the phase of the session key establishment, as depicted in Figure 3.

Below, we instantiate our V2V authentication protocol via the Cramer-Shoup-based SPHF [19].

1) $hk \leftarrow \mathsf{HashKG}(pk_{\mathsf{CS}})$: The algorithm outputs the hashing key $hk := k = (a_1, a_2, a_3, a_4)$ by sampling four randomness $a_1, a_2, a_3, a_4 \leftarrow \mathbb{Z}_q$ and taking as input the $pk_{\mathsf{CS}} = (g, h, c = g^{\alpha_1} h^{\alpha_2}, d = g^{\beta_1} h^{\beta_2}, f = g^{\rho}, H)$ of Cramer-Shoup encryption.
2) $ph \leftarrow \mathsf{ProjKG}(hk, pk_{\mathsf{CS}})$: *After receiving the hashing key* hk, the algorithm outputs the projection key $ph := p = (g^{a_1}, h^{a_2}, f^{a_3}, (cd^{\Theta})^{a_4})$, where $\Theta = H(\ell, (c_1, c_2, c_3))$.
3) $\mathsf{Hash}(hk, \mathrm{wrd} := (\mathsf{ct}_{\mathsf{CS}}, m))$: *After receiving the ciphertext* $\mathsf{ct}_{\mathsf{CS}}$ *(i.e.,* $c_1 = g^r$*,* $c_2 = h^r$*,* $c_3 = f^r \cdot m$*)* from Cramer-Shoup scheme, the algorithm creates a $\mathrm{wrd} = (\mathsf{ct}_{\mathsf{CS}}, m)$ over L as the input, then it outputs
$$\begin{aligned} & \mathsf{Hash}(k = (a_1, a_2, a_3, a_4), ((c_1, c_2, c_3, \phi), m) \\ = \; & c_1^{a_1} \cdot c_2^{a_2} \cdot (\frac{c_3}{m})^{a_3} \cdot \phi^{a_4} \\ = \; & g^{ra_1} \cdot h^{ra_2} \cdot f^{ra_3} \cdot \phi^{ra_4}, \end{aligned}$$
where $\phi = (cd^{\Theta})^r$.
4) $\mathsf{Proj}(ph, \mathrm{wrd} := (\mathsf{ct}_{CS}, m); w)$: *After receiving the public* ph *and seeing the ciphertext* $\mathsf{ct}_{\mathsf{CS}}$, the algorithm uses the witness w to calculate
$$\begin{aligned} & \mathsf{Proj}(p = (g^{a_1}, h^{a_2}, f^{a_3}, (cd^{\Theta})^{a_4}), w := r) \\ = \; & (g^{a_1})^r \cdot (h^{a_2})^r \cdot (f^{a_3})^r \cdot ((cd^{\Theta})^{a_4})^r. \end{aligned}$$

Claim IV.1. *The above-mentioned* SPHF *enabled by the Cramer-Shoup scheme is a smooth projective hash function.*

Proof. The SPHF instantiation can be proved to satisfy the properties of *projective* and *smoothness*. Here we ignore them here, and more details will be found in [19]. □

B. Round-Optimized Group Two-Factor AKE for V2V

To our knowledge, group key exchange protocols aim to provide a pool of players communicating over an open network with a shared secret key [35]. Katz and Yung [26] have proposed an efficient ring-based group key-exchange solution. Recently, Apon *et al.* has extended the solution of [26] to the lattice-based setting [36]. In a nutshell, they proceed [24], [26], [27] as follows: when n participants $U_1, U_2, \cdots, U_n$ would like to establish a session key. Remarkably, the indices are taken modulo n so that the participant U_0 is U_n and U_{n+1} is U_1.

- *Round 1*. Each participant U_i picks up a random $r_i \in \mathbb{Z}_q$ and broadcasts $z_i = g^{r_i}$.
- *Round 2*. Each participant U_i computes and broadcasts $X_i = \left(\frac{z_{i+1}}{z_{i-1}}\right)^{r_i} = g^{(r_{i+1} - r_{i-1}) \cdot r_i}$.
- *Finalization*. Each U_i calculates the session key
$$\begin{aligned} \mathsf{skey} \; = \; & (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i+n-2} \\ = \; & (g^{r_{i-1} r_i})^n \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i+n-2} \\ = \; & g^{r_1 r_2 + r_2 r_3 + \cdots + r_{n-1} r_n + r_n r_1} \end{aligned}$$

However, these two solutions [26], [36] have to rely on the computing-consuming PKI. Thus, removing the PKI and establishing the session only depending on the authentication factors (*e.g.*, password) is enough to be noticed recently. Inspired by the work of [37], the authors introduced the password to reduce the usage of PKI, and they proposed a transformation

from two-party PAKE setting to the group PAKE setting. We can follow their methodology to transform our (symmetric) two-factor AKE to our (symmetric) group two-factor AKE for multiple vehicle clients. However, their solution has to use a random oracle model to complete the security reduction. Indeed, SPHF is a particular hash function over the standard model without using any pseudorandom functions, it achieves the functionalities of designated verifier zero-knowledge proof that could be used to establish the session key only with the password, such as PAKE [13], [38] for two parties and PAKE for group setting [24]. Abdalla and Pointcheval gave the first group PAKE using SPHF that contains five rounds.

Key Establishment Phase. In our solution, we follow the solution of Abdalla and Pointcheval [24], and we combine two authentication factors (*e.g.*, password and one-time passcode) to design the two-factor group PAKE for the V2X setting.

- *Round 0 (a.k.a, Initialization)*. A trusted server runs the key generation algorithm $(ek_{\mathsf{CS}}, dk_{\mathsf{CS}}) \leftarrow \mathsf{CS.KeyGen}(1^\lambda)$. Then the server broadcasts ek_{CS} along with the universal hash functions $\mathrm{UH}_{\mathrm{k_{UH}}}$ and $\mathrm{UH}'_{\mathrm{k'_{UH}}}$. Each participant keeps their private password pw and one-time passcode otp. Then each participant executes password-to-random protocol discussed in [39] and obtains the random password $\mathsf{rpw} = \mathsf{OPRF}(\mathrm{pw})$, then calculates $\vartheta := H(\theta := (\mathsf{rpw} \| \mathsf{otp}), s := salt)$ locally. Here, we omit the detailed computation, and please refer to [39], [14] for more details.
- *Round 1*. Each participant U_i for $i = 1, 2, \cdots, n$ starts by setting the partner identifier pid_i to $\{U_1, U_2, \cdots, U_n\}$ and proceeds as follows.
  1) Generates a signature key-pair
  $$(vk_i, sk_i) \leftarrow \mathrm{Gen}(1^\lambda)$$
  for a signature scheme, and creates a label
  $$\ell_i := vk_i \| U_1 \| U_2 \| \cdots \| U_n.$$
  2) Encrypts the joint group password $\bar{Z}$ sampled from a dictionary of size $\|D\|$ using the encryption algorithm $\mathsf{CS.Enc}(\cdot)$ under the public key ek_{CS} with respect to the label ℓ_i and the randomness r_i^R. Finally, outputs the resulting ciphertext
  $$\mathrm{ct}_i^R \leftarrow \mathsf{CS.Enc}(ek, \ell_i, \bar{Z}; r_i^R).$$
  3) Encrypts once more the joint group authentication factor $\bar{Z}$ using $\mathsf{CS.Enc}(\cdot)$ under the common pk with respect to the received label ℓ_i and a new randomness r_i^L, then outputs a resulting ciphertext
  $$\mathrm{ct}_i^L \leftarrow \mathsf{CS.Enc}(ek, \ell_i, \bar{Z}; r_i^L).$$
  4) Generates three hashing key
  $$hk_i^L \leftarrow \mathsf{HashKG}(ek_{\mathsf{CS}}),$$
  $$hk_i \leftarrow \mathsf{HashKG}(ek_{\mathsf{CS}}),$$
  $$hk_i^R \leftarrow \mathsf{HashKG}(ek_{\mathsf{CS}}).$$
  At the end of the round, each participant U_i broadcasts the flow $(U_i, \ell_i, \mathrm{ct}_i^R, \mathrm{ct}_i^L)$.
- *Round 2*. Upon receiving the first flow $(\ell_i, \mathrm{ct}_i^R, \mathrm{ct}_i^L)$, each participant U_i for $i = 1, 2, \cdots, n$ proceeds as follows.
  1) generates three projective keys for the SPHF family,
  $$ph_i^L \leftarrow \mathsf{ProjKG}(hk_i^L, \ell_{i-1}, \mathrm{ct}_{i-1}^R),$$
  $$ph_i \leftarrow \mathsf{ProjKG}(hk_i, \ell_{i+1}, \mathrm{ct}_{i+1}^L),$$
  $$ph_i^R \leftarrow \mathsf{ProjKG}(hk_i^R, \ell_{i+1}, \mathrm{ct}_{i+1}^L).$$
  At the end of the round, each participant U_i broadcasts the flow (ph_i, ph_i^L, ph_i^R).
- *Round 3*. Upon receiving the first flow $(\mathrm{ct}_i^L, ph_i^L)$, each participant U_i for $i = 1, 2, \cdots, n$ proceeds as follows.
  1) Computes
  $$\mathsf{h}_i^R \leftarrow \mathsf{Hash}(hk_i^R, \ell_{i+1}, (\mathrm{ct}_{i+1}^L, \bar{Z}))$$
  $$\mathsf{p}_{i+1}^L \leftarrow \mathsf{Proj}(ph_{i+1}^L, \ell_{i+1}, (\mathrm{ct}_{i+1}^R, \bar{Z}); r_{i+1}^R),$$
  where $\mathrm{ct}_{i+1}^R \leftarrow \mathsf{CS.Enc}(pk, \ell_{i+1}, \bar{Z}; r_{i+1}^R)$.
  Then computes a test master key
  $$X_i^R = \mathsf{p}_{i+1}^L \cdot \mathsf{h}_i^R$$
  for its successor.
  2) Computes a test $\mathrm{test}_i^R = \mathrm{UH}_{\mathrm{k_{UH}}}(X_i^R)$ and sets the transcript
  $$T_i^R = U_i \| U_{i+1} \| \mathrm{ct}_i^R \| \mathrm{ct}_{i+1}^L \| ph_i^R \| ph_{i+1}^L \| \mathrm{test}_i^R,$$
  3) Validates if $\{0, 1\} \leftarrow \mathrm{Vrfy}(vk_{i-1}, \sigma_{i-1}^R)$, if fails the check, the participant U_i halts and sets $\mathsf{acc}_i = \mathrm{FALSE}$. Otherwise, the participant U_i computes
  $$\mathsf{h}_i^L \leftarrow \mathsf{Hash}(hk_i^L, \ell_{i-1}, (\mathrm{ct}_{i-1}^R, \bar{Z}))$$
  and computes
  $$\mathsf{p}_{i-1}^R \leftarrow \mathsf{Proj}(ph_{i-1}^R, \ell_i, (\mathrm{ct}_i^L, \bar{Z}); r_i^L),$$
  where $\mathrm{ct}_i^L \leftarrow \mathsf{CS.Enc}(pk, \ell_i, \bar{Z}, r_i^L)$.
  4) Generates the test master key
  $$X_i^L = \mathsf{h}_i^L \cdot \mathsf{p}_{i-1}^R$$
  for its predecessor.
  5) Verifies if
  $$\mathrm{test}_{i-1}^R = \mathrm{UH}_{\mathrm{k_{UH}}}(X_i^L).$$
  Once again, it fails this check, then participant U_i halts and sets $\mathsf{acc}_i = \mathrm{FALSE}$. If succeeds this test, then the participant U_i computes a test
  $$\mathrm{test}_i^L = \mathrm{UH}'_{\mathrm{k'_{UH}}}(X_i^L)$$
  for its predecessor, and computes an auxiliary key
  $$X_i = \mathsf{h}_i / \mathsf{p}_{i-1},$$
  where $\mathsf{h}_i \leftarrow \mathsf{Hash}(hk_i, \ell_{i+1}, (\mathrm{ct}_{i+1}^L, \bar{Z}))$ and $\mathsf{p}_{i-1} \leftarrow \mathsf{Proj}(ph_{i-1}, \ell_i, (\mathrm{ct}_i^L, \bar{Z}); r_i^L)$ for the ciphertext $\mathrm{ct}_i^L \leftarrow \mathsf{CS.Enc}(pk, \ell_i, \bar{Z}, r_i^L)$.
  At the end of this phase, each participant U_i broadcasts the pair $(X_i, X_i^L, X_i^R, \mathrm{test}_i^L, \mathrm{test}_i^R)$.
- *Round 4*. Upon receiving the flow (X_i, test_i^L), each participant U_i for $i = 1, 2, \cdots, n$ proceeds as follows.
  1) Validates if $\mathrm{test}_{i+1}^L = \mathrm{UH}'_{\mathrm{k'_{UH}}}(X_i^R)$ and if $\Pi_{l=1}^n X_l = 1$.
  2) If any of these tests fails, then participant U_i halts and sets $\mathsf{acc}_i = \mathrm{FALSE}$. Otherwise, each participant U_i sets.
  $$T_j = vk_j \| U_j \| \mathrm{ct}_j \| ph_j \| ph_j^L \| ph_j^R \| X_j \| X_j^L$$
  for $j = 1, 2, \cdots, n$ and
  $$T = T_1 \| T_2 \| \cdots \| T_n$$
  and signs it
  $$\sigma_i \leftarrow \mathrm{Sign}(sk_i, T)$$
  At the end of this phase, each U_i broadcasts σ_i.
- *Finalization*. Upon receiving the σ_i, each participant U_i for $i = 1, 2, \cdots, n$ proceeds as follows.
  1) Checks for $j \neq i$ if $1 \leftarrow \mathrm{Vrfy}(vk_j, \sigma_j)$, where the signature σ_j is on the $T = T_1 \| T_2 \| \cdots \| T_n$.
  2) If any of these checks fails, then the participant U_i halts

and sets acc_i = FALSE. Otherwise, the participant U_i sets acc_i = TRUE and computes the master key

$$\begin{aligned} msk &= \mathsf{h}_i{}^n \cdot \Pi_{j=1}^{n-1} \left(\frac{\mathsf{h}_{i+j}}{\mathsf{p}_{i+j-1}}\right)^{n-j} \\ &= \Pi_{j=1}^{n} \mathsf{Hash}(hk_j, \ell_{j+1}, (\mathrm{ct}_{j+1}, \bar{Z})) = \Pi_{j=1}^{n} \mathsf{h}_j. \end{aligned}$$

C. Correctness and Security for V2V

In the V2V setting, we assume the vehicles complete successfully the registration and authentication, thus, we so we omit the analysis of the correctness and security.

Correctness. The correctness of the registration and login-authentication are evident and can be verified easily. In the phase of session-key establishment, the symmetric two-factor authenticated key exchange can be achieved relying on the projective property of SPHF and labelled Cramer-Shoup encryption, which is similar to the conventional symmetric-PKAE protocol based on the SPHF except for the OTP usage. Thus, the correctness can be verified in the V2V setting, if the session key of three parties setting, then computes the master key

$$\begin{aligned} msk_i &= \mathsf{h}_i{}^n \cdot \Pi_{j=1}^{n-1} \left(\frac{\mathsf{h}_{i+j}}{\mathsf{p}_{i+j-1}}\right)^{n-j} \\ &= \Pi_{j=1}^{n} \mathsf{Hash}(hk_j, \ell_{j+1}, (\mathrm{ct}_{j+1}, \bar{Z})) \\ &= \Pi_{j=1}^{n} \mathsf{h}_j \end{aligned}$$

and validates whether $msk_1 \stackrel{?}{=} msk_2 \stackrel{?}{=} msk_3$.

Security. Regarding the security of the symmetric two-factor AKE, we adopt the modularity analysis strategy that follows the methodologies of Katz and Vaikuntanathan [30] and Benhamouda *et al.* [40, Theorem 4]. In that case, we only need to check that the SPHF is associated with Cramer-Shoup encryption. Below, we sketch the RoR analysis strategy. We use the Execute oracle to model the passive security (*i.e.*, eavesdropping) for the two participants, respectively. Then we use the Send oracle to model the active security (*i.e.*, MITM attacks, insertion, deletion, or arbitrarily modification, *etc*) for the two participants. After that, we use the Corrupt oracle to model the set of server corruption. Finally, we use the Test oracle to answer the adversary whether his guess is correct or not. To facilitate our main contribution, we omit the detailed security analysis for the symmetric two-factor AKE.

Theorem IV.2. *The proposed two-factor group PAKE protocol for the V2V setting over a prime q-order group are secure under the DDH assumption, where the password dictionary is drawn from* $\mathsf{D} \in \mathbb{Z}_q^*$.

The two-factor group PAKE for the V2V setting is secure if for all PPT adversary $\mathcal{A}$ making at most $Q(\lambda)$ on-line dictionary attacks, it holds that $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{GPAKE}}(\lambda) \leq Q(\lambda)/D + \mathrm{negl}(\lambda)$.

Sketched Proof. Below, the security analysis can be sketched here, that is following the analysis strategy of [41] in the BRP model [42]. Notably, we operate two distinct authentication factors to establish the session key. Thus, in order to analyze easily, we regard the cascaded θ via rpw and otp as a common factor, then we can analyze it in the approach of AKE. Please see the appendix for our detailed security analysis of IV.2.

V. Our Solution: Two-Factor Authentication and Key Establishment for Vehicle-to-Infrastructure

Traditional key exchange protocols are not good candidates for secure V2I communications. We cannot adapt these approaches in the V2V setting straightforwardly to the V2I communications. The main reasons are concluded as follows:

- Vehicle is a moving entity, and it is easy to join (or leave) a nearby RSU domain. In general, the vehicle's velocity is assumed 60 km/h (or 16.6 m/s). Thus, to estimate the time interval, we assume the communication interval between the OBU and RSU in a straight road, in this setting, that can be roughly calculated as $36.1s (= 2 * 300/16.6)$.
- RSUs are deployed at the fixed coordinates by the service provider, and the vehicle users may check them for a certain time. However, an eavesdropper could be installed close to the RSU device by an attacker so that no one could notice it for an extended period.

Further, to our knowledge, the first multi-factor AKE scheme in the random oracle model (ROM) was proposed by Pointcheval and Zimmer [21], their computing cost is acceptable in real applications because it depends on the ElGamal encryption and pseudorandom functions. But their communication complex is not considered optimal with four rounds of interaction, and their solution on how to establish the session key is desperately vulnerable because some known attacks had launched on it. Thus, in the V2I environment, our solution would enhance the security level (*i.e.*, robustness and balanced security) to prevent known attacks while achieving the quasi-optimal (even optimal) communication rounds. In this setting, the CPA-secure ElGamal encryption adopted from [21] can not satisfy the strong security requirement as pointed out in [10]. Further, the conventional pseudorandom functions enable to obtain of the session key by designing the quasi-optimal (even optimal) communication rounds two-factor AKE, while these kinds of solutions rely heavily on the ROM. Therefore, the above two-mentioned observations promote us to explore a solution of quasi-optimal (even optimal) two-factor group AKE without using the ROM.

Below, we detail our construction to different steps for the key establishment during the V2I environment. In this work, we instantiate our solution of the key establishment for the V2I setting by adopting the CCA-secure Cramer-Shoup encryption scheme with an associated SPHF. Further, we highlight the main techniques for each phase in Figure 2 to depict our construction in the V2I communication.

- When the vehicle user would authenticate the server, he will first fetch the randomized password rpw from the physical device D. Notably, the vehicle user types in a "randomized password" $\mathsf{rpw} = \mathsf{F}_{\mathsf{key}}(\mathrm{pw})$ rather than a memorable regular pw where rpw is generated by using a pseudorandom function F under a secret key key provided by a physical device (*e.g.*, smartphone) D. An important observation is that the randomized rpw is in the range set of F with has full entropy while it is without knowledge of key. We need to stress that, during this phase, the physical device is unavailable to learn the knowledge of pw.

Figure 3: An illustration of Two-Factor Group (Three-Party) Password-Authenticated Key Exchange for Vehicle-to-Vehicle

- After authentication, the vehicle would generate a common session key with the server to guarantee the security of the message transmission. Thus, our designed asymmetric two-factor AKE protocol is introduced here to establish a session key by using two distinct factors (*i.e.*, rpw and otp). In addition, to prevent leakage of the stored rpw and otp at the server-side, we introduce a zero-knowledge proof of $\vartheta = H_p(\theta)$ to ensure that offline attacks are infeasible.

A. Two-Factor Authentication and AKE for V2I

1) Registration Phase: The registration phase of V2I setting is similar with the V2V setting. In a nutshell, all of RSUs first register to the service provider. Then the legitimate RSUs who have been validated by the service provider will establish a secure communication channel with the service provider. In addition, the RSUs will store the index of the registered vehicles equipped with the OBU so that the vehicles could contact the nearby RSU when the legitimate vehicle goes into the domain controlled by the RSU. Then the vehicles and RSU first initializes parameters, *e.g.*, public keys, and then they execute two-factor AKE to establish the common session key.

2) Authentication Phase: The authentication phase is happening at a time when the vehicle users want to authenticate the RSU to access the service. Here, the vehicle users with the unique uid should be equipped with OBU and a registered device $\mathcal{D}_{\mathsf{dvc}}$. Below, we detail the vehicle how to communicate with the RSU.

- The vehicle client with the unique OBU sends an authentication request to the RSU. Then the vehicle inputs his (or her) password pw, and responses the randomized password rpw from the special device $\mathcal{D}_{\mathsf{dvc}}$ who is armed with the salt s. Next, the vehicle client picks the stored otp, then the vehicle sends the triple $(\mathsf{rpw}, \mathsf{otp}, \mathsf{uinfo})$ to the RSU.
- Once received the uid along with the triple $(\mathsf{rpw}, \mathsf{otp}, \mathsf{uinfo})$, the RSU proceeds as the registration phase, he first concatenates rpw and otp to create a new $\theta' = \mathsf{rpw}\|\mathsf{otp}$, then the RSU computes $\vartheta' = H(\theta', s)$ under the salt s, and finally he calculates $Z = g^{\vartheta'}$. Finally, the RSU checks whether the received $\mathsf{HMAC}(Z') \stackrel{?}{=} \mathsf{HMAC}(Z)$, where the $\mathsf{HMAC}(Z)$ stores in the database. Notably, if the new generated $\mathsf{HMAC}(Z')$ matches the registered $\mathsf{HMAC}(Z)$, the vehicle client is authenticated and enabled to access the service provided by the RSU.

3) Session Key Establishment Phase: After authentication, the unqualified vehicle users are refused to access the services provided by the RSU, only the legitimate vehicle is available to generate a high-entropy session key with the designed RSU.

In the V2I authentication, the SPHF for the asymmetric two-factor AKE is instantiated by the labeled Cramer-Shoup scheme [19], which is similar to the setting of V2V authentication for the symmetric two-factor AKE. Further, we introduce the NIZK proof to enable the service provider (or RSU) to validate the legality of the authentication factors on the client-side without compromising their privacy. Below we detail how the NIZK works for the pre-hash of the authentication factors at the vehicle client-side.

1) Firstly, the vehicle client first calculates the hash of $\vartheta = H(\theta, s)$ under the designed salt s, where the ϑ is regarded as the witness in the following steps. Then the vehicle client creates a statement $Z = g^{\vartheta}$ and broadcasts it to the service provider and the RSU in the special domain.
2) Next, the vehicle client samples a masking $\gamma \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$ to prevent the witness ϑ from revealing. Thus, the vehicle client follows the procedures on how to hind the witness and broadcast the statement in the first steps and generates a fake masking statement $\tau = g^{\gamma}$. Then the vehicle client calculates creates $\tau' = \mathsf{HMAC}(\tau)$ and use τ' to generate a challenge $\varepsilon = H_{\mathsf{zk}}(\tau')$.
3) Finally, the vehicle client forms the zero-knowledge proof $\pi := \varepsilon \cdot \vartheta + \gamma \pmod q$ for the witness ϑ with supporting the statement Z and the calculated challenge ε. Then, the vehicle client sends the fake statement $\tau' = \mathsf{HMAC}(\tau)$ and the corresponding challenge ε to the service provider and the RSU in the special domain.

1) firstly, the vehicle client first calculate the hash of $\vartheta = H(\theta, s)$ under the designed salt s, where the ϑ is regarded as the witness in the following steps. Then the vehicle client creates a statement $Z = g^{\vartheta}$ and broadcast it to the service provider and the RSU in the special domain.
2) Next, the vehicle client samples a masking $\gamma \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$ to prevent the witness ϑ from revealing. Thus, the vehicle client follows the procedures on how to hind the witness and broadcast the statement in the first steps, and generates a fake masking statement $\tau = g^{\gamma}$. Then the vehicle client calculates creates $\tau' = \mathsf{HMAC}(\tau)$ and use τ' to generate a challenge $\varepsilon = H_{\mathsf{zk}}(\tau')$.

Below, we describe our solution in the V2I environment on how to establish the session key via two different authentication factors, As the above explanation, we leverage the NIZK to validate the legality of the authentication factors in the client-side without compromising the privacy of them. After generating a challenge $\varepsilon = H_{\mathsf{zk}}(\tau')$ and a proof π, the vehicle client sends them along with the public ph_C and the ciphertext (c_1, c_2, c_3, ϕ) to the RSU. Subsequently, the RSU continues to validate whether meets the requirement $\tau \stackrel{?}{=} g^{\pi} \cdot Z^{-\varepsilon} = g^{\pi} \cdot (g^{\vartheta})^{-\varepsilon}$. If passes, then the RSU execute the $\mathsf{Proj}_S(ph_C, \mathrm{wrd}_S; w_S)$ to gain $\mathsf{p}_C := (ph_S)^{w_C}$. Synchronously, the vehicle client executes the operation $\mathsf{Proj}_C(ph_S, \mathrm{wrd}_C; w_C)$ to gain $\mathsf{p}_C := (ph_S)^{w_C}$ after receiving $(\mathsf{ph}_S, c_0', c_1')$.

B. Correctness and Security for V2I

Correctness and security of the two-factor authentication and two-factor group AKE for V2I communication are analyzed in this part, respectively. In the following correctness and security analysis, we only focus on the key establishment (*i.e.*, two-factor group AKE) phase because we have assumed that the registration and authentication are completed in the secure channel. Thus, we ignore these analyses in this part. Importantly, in the key establishment phase, the two-factor group PAKE is achieved heavily relying on the SPHF and Cramer-Shoup encryption.

Correctness. The correctness of V2I setting is captured via the following Lemma V.1.

Lemma V.1. *If the correctness of two-factor authentication for V2I communication holds, then the session key of vehicle client and RSU satisfy* $\mathsf{key}_C := \mathsf{p}_S \cdot \mathsf{h}_C = \mathsf{p}_C \cdot \mathsf{h}_S = \mathsf{key}_S$.

The correctness can be proved by using the property of projection. Regarding the security, the following Theorem IV.2 is introduced here.

Security. The security of V2I setting is captured via the following Theorem V.2.

Theorem V.2. *The proposed two-factor authentication and two-factor group PAKE protocol for V2I setting over a prime q-order group are secure under the DDH assumption, where the password dictionary is drawn from* $\mathsf{D} \in \mathbb{Z}_q^*$.

Here we omit the detailed security analysis of two-factor group PAKE for V2I setting. Indeed, the detailed analysis could refer to [14], however, it is with two factors to achieve the authentication. Please refer to the appendix to see the detailed analysis of IV.2 that is covered the setting of V2I and V2V.

VI. Performance Evaluation

Below, the theoretical and the experimental analyses are provided for our proposed two-factor authentication and two-factor AKE for intelligent transport vehicles.

Theoretical Analysis. Compared with the traditional password-authenticated key exchange (*e.g.*, secure remote protocol, or SRP) that only supports the single-factor (*i.e.*, password) authentication and single-factor AKE protocols, our two-factor authentication and two-factor group PAKE support two factors to synergistic fulfill the requirement of the two-factor authentication and two-factor group PAKE schemes. Further, our two-factor solutions are computationally efficient while our communication is round quasi-optional. As shown in Table II, we compared with some existing group password-authenticated key exchange protocols, such as [24], [27], [43], [44]. Concretely, our protocol follows the research line of [24] and reduces four communication rounds by removing the legality verification at the end of each round and merging the generations of SPHF into one round instead of two rounds. Further, the building blocks are updated by optimizing the OPRF and SPHF with two authentication factors. Compared with [27], [43], [44], our solution is two factors scheme based on DDH assumption without random oracle model. Notably, the session key could be established in four founds. Even [44] could be generated the session key in two rounds, but their authors only provided a generic solution without detailed instances under the universal composable framework.

Additionally, a fact is that the knowledge of authentication factors will disclose if the malicious attacker compromises the server successfully. Thus, storing the passwords in cleartext is not practical for the V2I setting. In that case, some kinds of literature have begun to consider how to guarantee the hashed authentication factors' privacy after the server is compromised.

However, the hashed authentication factors at the server-side could increase the difficulty for the service provider (or RSU) to validate the authentication factors' legality because the vehicle client has required to type in the authentication factors cleartext without any protection. Thus, we introduce the NIZK proof to enable the service provider (or RSU) to validate the legality of the authentication factors on the client-side without compromising their privacy.

Experimental Analysis. To specify the execution time of different cryptographic operations, we separate each round of communication complexity by different cryptographic operations. In more detail, in the V2V setting, each participant only has to perform two encryptions, three hashing key-generation, three projection key-generation, three hashing computations, three projected hashing computations, and five universal hash computations. In the V2I setting, each participant only has to perform one encryption, one hashing key-generation, one projection key-generation, 1 hashing computing, one projected hashing computing. Additionally, the client-side has to perform one universal hash computation.

For simplicity, we only assume that the collected biometric data could be encoded into N-bits in a bit-by-bit manner without considering how to extract the encodable information from the biometric data. It can be observed that the reason behind the expensive computation of the existing biometric authentication schemes is the biometric bit-by-bit encoding approach. Indeed, our solution is with much stronger robustness, so that the communication traffic is not optimal. To evaluate the actual performance, we wrote a proof-of-concept two-factor AKE based on the Cramer-Shoup encryption and its associated SPHF, which are based on the GMP (*i.e.*, GNU Multiple Precision Arithmetic) libraries. In order to illustrate the difference of the performance on a different platform, we conduct a series of evaluations on the Raspberry Pi 3 Model B+ to simulate the vehicle client sides that are equipped with OBU.

Our experiments encode the biometric templates into six bytes with the security parameter varies from 128 to 1024 bits. According to our experiment, we estimate the computation overheads from 128 bits to 1024 bits are as the following Table III. Further, the space costs on both sides are the same, while the execution time on both sides has a big gap, and the overhead of encryption and NIZK is near to 1 second. Thus, the total overheads can be tolerated by the industry IoT devices. Indeed, the most expensive part of the V2V setting, which is linear in the group size, is the number of zero-knowledge proof generation and verification (including signature verification) and the master session key generation.

As shown in Table IV, we detailed the computation and communication complexity for each round. Then we could observe the following points. Firstly, the parties establish the session key using at least 1.034 ms with the laptop or at least 4.5048 ms with the Raspberry if the security parameter is 128 bits. Analogously, the parties establish the session key using at most 17.702 ms with the laptop or 93.31 ms with the Raspberry if the security parameter is 1024 bits. Thus, the time consumption in security parameters from 128 bits to 512 bits could be acceptable by industry. Secondly, no matter how many parties join the vehicle community to establish the session key, the time and computation consumption are likely to produce the same result.

Table II: Overview of Differences Group PAKE Systems.

Scheme	Round	Assumption	Model	Factor(s)	Auth	Building Blocks
Abdalla and Pointcheval [24]	5	DDH	Std	1	✗	CS enc+ SPHF
Abdalla *et al.* [27]	6	DDH	UC & ROM	1	✗	2PAKE+one-time Sign+MAC
Abdalla *et al.* [43]	7(+1)	CDH	ROM	1	✗	PRF+ MAC
Fiore *et al.* [44]	2	(Generic Construction)	ROM+Ideal Cipher	1	✗	Sym.Enc + Hash
Ours	4	DDH	Std	2	✓	CS enc + SPHF + Oblivious-PRF

- ✓ denotes that the scheme does have this property; ✗ denotes that the scheme does not have this property.
- UC implies that universal composable framework; ROM implies that random oracle model.

Table III: Execution Analysis (average) for each Algorithm at Different Security Parameter.

Performance	128-bit Execution (ms)	128-bit Storage (bit)	256-bit Execution (ms)	256-bit Storage (bit)	512-bit Execution (ms)	512-bit Storage (bit)	768-bit Execution (ms)	768-bit Storage (bit)	1024-bit Execution (ms)	1024-bit Storage (bit)
KeyGen	0.0732	317+383	0.1692	704+768	0.9084	1118+1536	3.3633	2570+2304	4.7755	5819+3069
Encryption	0.1117	509	0.2100	1022	0.3462	2046	0.9191	3070	1.5332	4092
HashGen	0.0033	304	0.0054	598	0.0068	1059	0.0093	1745	0.0071	2385
ProjGen	0.0813	510	0.1542	1023	0.2861	2047	0.8670	3071	1.3013	4094
SPHF.Hash	0.0162	128	0.0655	256	0.1514	511	0.6110	768	0.8851	1024
SPHF.Proj	0.0144	128	0.0515	256	0.1767	511	0.6348	768	0.7652	1024
Performance	**Raspberry in 128-bit Execution**	**Raspberry in 128-bit Storage**	**Raspberry in 256-bit Execution**	**Raspberry in 256-bit Storage**	**Raspberry in 512-bit Execution**	**Raspberry in 512-bit Storage**	**Raspberry in 768-bit Execution**	**Raspberry in 768-bit Storage**	**Raspberry in 1024-bit Execution**	**Raspberry in 1024-bit Storage**
KeyGen	0.3001	317+383	0.6937	704+768	3.724	1118+1536	13.7895	2570+2304	19.5795	5819+3069
Encryption	0.5697	509	1.0711	1022	1.7241	2046	2.3435	3070	7.6353	4092
HashGen	0.0132	304	0.0216	598	0.0279	1059	0.0272	1745	0.0284	2385
ProjGen	0.4391	510	0.8342	1023	1.5421	2047	4.6991	3071	7.0141	4094
SPHF.Hash	0.1391	128	0.5621	256	1.2995	511	5.2430	768	7.951	1024
SPHF.Proj	0.1802	128	0.6474	256	2.2176	511	7.9541	768	9.5727	1024

Table IV: Communication Analysis (average) for each Flow at Different Security Parameter with Three Parties.

Performance	128-bit Execution (ms)	128-bit Msg-Size (bit)	256-bit Execution (ms)	256-bit Msg-Size (bit)	512-bit Execution (ms)	512-bit Msg-Size (bit)	768-bit Execution (ms)	768-bit Msg-Size (bit)	1024-bit Execution (ms)	1024-bit Msg-Size (bit)
First-Flow	0.1849	700	0.3792	1472	1.2545	2654	4.2824	4874	6.3087	8888
Second-Flow	0.1963	1323	0.3696	2643	0.6390	5152	1.7954	7886	2.8417	10571
Third-Flow	0.3304	1533	0.5933	2558	1.1001	4606	3.2213	6654	4.5951	8700
Forth-Flow	0.1264	256	0.3125	256	0.7493	256	2.6031	256	3.4645	256
Verification	0.0653		0.0785		0.0932		0.1115		0.1640	
Performance	**Raspberry in 128-bit Execution**	**Raspberry in 128-bit Storage**	**Raspberry in 256-bit Execution**	**Raspberry in 256-bit Storage**	**Raspberry in 512-bit Execution**	**Raspberry in 512-bit Storage**	**Raspberry in 768-bit Execution**	**Raspberry in 768-bit Storage**	**Raspberry in 1024-bit Execution**	**Raspberry in 1024-bit Storage**
First-Flow	0.9148	700	1.8098	1472	5.4931	2654	16.1783	4874	27.2598	8888
Second-Flow	1.0221	1323	1.9269	2643	3.2941	5152	6.0698	7886	14.6778	10571
Third-Flow	1.3139	1533	3.0111	2558	6.7471	4606	22.7398	6654	31.6987	8700
Forth-Flow	0.3643	256	1.2545	256	3.5621	256	13.2421	256	17.5237	256
Fifth-Flow	0.5176	256	0.6433	256	0.7642	256	0.9143	256	1.3087	256
Verification	0.3721		0.3949		0.4781		0.5675		0.8413	

- First flow means the pre-authentication stage that contains the overhead of KeyGen, Enc, and HMAC.
- Second flow means the authentication stage that contains the overhead of HashKG, ProjKG and Enc.
- Third flow means the stage that contains the overhead of twice of HashKG and ProjKG, and one time Hash, Proj, UH and Sign.
- Forth flow means the stage that contains the overhead of two times of Hash, Proj and UH.
- Fifth flow means the stage contains one-time UH and Sign.
- Verification contains one time UH and two times multiplication operations.

VII. Conclusion

In our work, we presented two practical two-factor authentications and two-factor AKE protocols for ITS in the V2X networks, in a nutshell, symmetric two-factor authentication for the V2V networks and asymmetric two-factor authentication for the V2I networks, which are two composable protocols containing the registration phase, (asymmetric) two-factor based authentication phase and (two-factor authenticated) key-exchange phase. To reduce the communication complexity, during the two-factor AKE phase, we introduced the SPHF based on the Cramer-Shoup scheme to establish the session key between the two vehicles and between the vehicles and the public infrastructure. Finally, under the BPR model, the balanced security of the two-factor authentication and two-factor group AKE protocols are achieved along with experimental evaluations and sketched security analysis.

VIII. Acknowledgements

This research was supported in part by the National Key Research and Development Program of China No. 2021YFA1000600, the National Natural Science Foundation of China under Grant No. 61802214, in part by the National Natural Science Foundation of Shandong province, China under Grant No. ZR2019BF009, the Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, and Shandong University Young Scholars Future Program.

References

[1] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial internet of things and industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[2] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1015–1028, 2016.

[3] M. Azees, P. Vijayakumar, and L. J. Deborah, "EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2467–2476, 2017.

[4] S. Jiang, X. Zhu, and L. Wang, "An efficient anonymous batch authentication scheme based on HMAC for vanets," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2193–2204, 2016.

[5] J. Camenisch, M. Drijvers, A. Lehmann, G. Neven, and P. Towa, "Zone encryption with anonymous authentication for V2V communication," *IACR Cryptology ePrint Archive*, vol. 2020, p. 43, 2020. [Online]. Available: https://eprint.iacr.org/2020/043

[6] A. Jindal, G. S. Aujla, N. Kumar, R. Chaudhary, M. S. Obaidat, and I. You, "Sedative: Sdn-enabled deep learning architecture for network traffic control in vehicular cyber-physical systems," *IEEE Netw.*, vol. 32, no. 6, pp. 66–73, 2018.

[7] R. Lu, X. Lin, X. Liang, and X. S. Shen, "A dynamic privacy-preserving key management scheme for location-based services in vanets," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 127–139, 2012.

[8] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks," in *Proc. EUROCRYPT 2018*, 2018, pp. 456–486.

[9] B. Haase and B. Labrique, "Aucpace: Efficient verifier-based PAKE protocol tailored for the iiot," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 2, pp. 1–48, 2019.

[10] J. Katz, R. Ostrovsky, and M. Yung, "Efficient and secure authenticated key exchange using weak passwords," *J. ACM*, vol. 57, no. 1, pp. 3:1–3:39, 2009.

[11] A. Everspaugh, R. Chatterjee, S. Scott, A. Juels, and T. Ristenpart, "The pythia PRF service," in *Proc. USENIX Security 2015*, 2015, pp. 547–562.

[12] J. Schneider, N. Fleischhacker, D. Schröder, and M. Backes, "Efficient cryptographic password hardening services from partially oblivious commitments," in *Proc. ACM CCS 2016*, 2016, pp. 1192–1203.

[13] Z. Li and D. Wang, "Achieving one-round password-based authenticated key exchange over lattices," *IEEE Trans. Service Computing*, 2019. [Online]. Available: https://doi.org/10.1109/TSC.2019.2939836

[14] Z. Li, Z. Yang, P. Szalachowski, and J. Zhou, "Building low-interactivity multi-factor authenticated key exchange for industrial internet-of-things," *IEEE Internet of Things Journal*, 2020. [Online]. Available: https://doi.org/10.1109/JIOT.2020.3008773

[15] V. Kolesnikov and C. Rackoff, "Password mistyping in two-factor-authenticated key exchange," in *Proc. ICALP 2008*, 2008, pp. 702–714.

[16] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Two-factor authentication with end-to-end password security," in *Proc. PKC 2018*, 2018, pp. 431–461.

[17] K.-K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial internet-of-things: Research challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, 2018.

[18] H. Mouratidis and V. Diamantopoulou, "A security analysis method for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4093–4100, 2018.

[19] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in *Proc. EUROCRYPT 2003*, 2003, pp. 524–543.

[20] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, p. 1, 2018.

[21] D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," in *Proc. ACNS 2008*, 2008, pp. 277–295.

[22] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[23] N. Fleischhacker, M. Manulis, and A. Azodi, "A modular framework for multi-factor authentication and key exchange," in *Proc. SSR 2014*, 2014, pp. 190–214.

[24] M. Abdalla and D. Pointcheval, "A scalable password-based group key exchange protocol in the standard model," in *Proc. 12th ASIACRYPT 2006*, vol. 4284. Springer, 2006, pp. 332–347.

[25] M. Abdalla, D. Catalano, C. Chevalier, and D. Pointcheval, "Password-authenticated group key agreement with adaptive security and contributiveness," in *Proc. AFRICACRYPT 2009*, vol. 5580. Springer, 2009, pp. 254–271.

[26] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," *J. Cryptol.*, vol. 20, no. 1, pp. 85–113, 2007.

[27] M. Abdalla, C. Chevalier, L. Granboulan, and D. Pointcheval, "Contributory password-authenticated group key exchange with join capability," in *Proc. CT-RSA 2011*, vol. 6558. Springer, 2011, pp. 142–160.

[28] M. Azees, P. Vijayakumar, and L. J. Deborah, "EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2467–2476, 2017.

[29] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 2, pp. 722–735, 2021.

[30] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," in *Proc. TCC 2011*, 2011, pp. 293–310.

[31] M. Abdalla, F. Benhamouda, and D. Pointcheval, "Public-key encryption indistinguishable under plaintext-checkable attacks," in *Proc. PKC 2015*, 2015, pp. 332–352.

[32] A. Bhargav-Spantzel, A. C. Squicciarini, S. K. Modi, M. Young, E. Bertino, and S. J. Elliott, "Privacy preserving multi-factor authentication with biometrics," *Journal of Computer Security*, vol. 15, no. 5, pp. 529–560, 2007.

[33] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Proc. EUROCRYPT 2002*, 2002, pp. 45–64.

[34] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT 2000*, 2000.

[35] B. Poettering, P. Rösler, J. Schwenk, and D. Stebila, "Sok: Game-based security models for group key exchange," Cryptology ePrint Archive, Report 2021/305, 2021, https://eprint.iacr.org/2021/305.

[36] D. Apon, D. Dachman-Soled, H. Gong, and J. Katz, "Constant-round group key exchange from the ring-lwe assumption," in *Proc. 10th PQCrypto 2019*, vol. 11505. Springer, 2019, pp. 189–205.

[37] J.-M. Bohli, M. I. G. Vasco, and R. Steinwandt, "Password–authenticated group key establishment from smooth projective hash functions," *International Journal of Applied Mathematics and Computer Science*, vol. 29, no. 4, pp. 797 – 815, 01 Dec. 2019. [Online]. Available: https://eprint.iacr.org/2006/214.pdf

[38] Z. Li, D. Wang, and E. Morais, "Quantum-safe round-optimal password authentication for mobile devices," *IEEE Transactions on Dependable and Secure Computing*, 2020. [Online]. Available: https://doi.org/10.1109/TDSC.2020.3040776

[39] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Device-enhanced password protocols with optimal online-offline protection," in *Proc. 11th ACM AsiaCCS 2016*. ACM, 2016, pp. 177–188.

[40] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "New techniques for sphfs and efficient one-round PAKE protocols," in *Proc. CRYPTO 2013*, 2013, pp. 449–475.

[41] O. Blazy, C. Chevalier, and D. Vergnaud, "Mitigating server breaches in password-based authentication: Secure and efficient solutions," in *Proc. CT-RSA 2016*, 2016, pp. 3–18.

[42] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "New techniques for sphfs and efficient one-round PAKE protocols," in *Proc. CRYPTO 2013*, 2013, pp. 449–475.

[43] M. Abdalla, E. Bresson, O. Chevassut, B. Möller, and D. Pointcheval, "Strong password-based authentication in TLS using the three-party group

diffie-hellman protocol," *Int. J. Secur. Networks*, vol. 2, no. 3/4, pp. 284–296, 2007.

[44] D. Fiore, M. I. G. Vasco, and C. Soriente, "Partitioned group password-based authenticated key exchange," *Comput. J.*, vol. 60, no. 12, pp. 1912–1922, 2017.