Network-efficient authenticated pseudonym-based V2X communications with constant revocation costs

Mattia Trabucco
Department of Physics, Informatics and Mathematics
University of Modena and Reggio Emilia
Modena, Italy
mattia.trabucco@unimore.it

Giovanni Gambigliani Zoccoli
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
giovanni.gambiglianizoccoli@unimore.it

Mirco Marchetti
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
mirco.marchetti@unimore.it

Luca Ferretti
Department of Physics, Informatics and Mathematics
University of Modena and Reggio Emilia
Modena, Italy
luca.ferretti@unimore.it

***Abstract*—Standard Vehicle-to-everything (V2X) communications guarantee privacy against tracking by provisioning each vehicle with many *conditionally unlinkable pseudonym certificates*, which can be linked to each other only with knowledge of a secret information, disclosed by ad-hoc authorities in case of misbehavior for efficient revocation. Certificates are bound to independent keys, but include pseudo-random indexing values which enable such an efficient mechanism at the expense of increasing network overhead of all V2X messages. We propose a novel network-efficient protocol where each vehicle is provisioned with *conditionally unlinkable* pseudo-random asymmetric key pairs, thus removing the need for linkage values and reducing network overhead while still supporting revocation at constant network costs. Our approach represents a novel application of *hierarchical deterministic homomorphic key derivation schemes*, which are mostly known for deterministic wallets in the context of blockchains. Compared to standards based on explicit certificates, our approach has lower network overhead, no computational overhead for securing communications, and same cryptographic assumptions. Computational costs for key management operations are higher, but still affordable. We analyze the costs of our proposal both asymptotically and analytically when instantiated with the NIST p-256 elliptic curve recommended by standards.**

***Index Terms*—pseudonym certificate, hierarchical key derivation, deterministic wallet, V2X, vehicular communication, VANET**

I. Introduction

Vehicle-to-Everything (V2X) communications are key enablers for Intelligent Transportation Systems, but their design includes many challenges due to the need of balancing security and privacy with tight latency-requirements of safety-related features, resource constraints of vehicular wireless communication protocols, and economic costs of infrastructures and, in particular, of vehicles [1].

Privacy relates to the peculiar need of guaranteeing *conditional anonymity* to vehicles, that is, preventing a vehicle from being tracked by malicious parties, only as long as it behaves honestly. As soon as a vehicle behaves maliciously, e.g., by sending false information, it should be possible to efficiently revoke it from the network, thus invalidating its privacy guarantees. As completely anonymous authentication schemes are very expensive and not affordable in the context of vehicular networks, the typical design choice is to adopt a PKI-based approach based on pseudonyms [2]. Vehicles are provisioned with a set of certificates issued by a Certificate Authority (CA) which are similar to those used in the Web PKI, as they are bound to the vehicle and the CA public keys (either by explicitly including them, or via implicit mathematical properties), and optional metadata such as the validity period of the certificate itself. However, certificates issued in the Vehicular PKI (VPKI) omit the real identity of the vehicle or other user-identifiable information, and are instead considered pseudonyms of the vehicle. Each vehicle can authenticate its messages with different secret keys, thus a message is supposed to be unlinkable to the real vehicle identity and to other messages authenticated by the same vehicle with a different certificate. This allows a vehicle to maintain its privacy while still being able to authenticate messages within the network. The major standards for VPKI are the US Security Credential Management System (SCMS) [3] and the European ETSI ITS standard [4], where each vehicle is provisioned with a set of pseudonym certificates and corresponding key pairs that are valid for a specific time period (e.g., one week). Vehicles select one of the pseudonym certificates from its set to authenticate outgoing messages, and rotates at regular intervals the pseudonym certificate to ensure unlinkability between messages sent over the time period (e.g., every 10 minutes or after sending a certain number of messages) [5].

Consequently, a fundamental challenge in the VPKI is the management of a large volume of pseudonym certificates per vehicle. The majority of approaches [1], [2] favor occasional pseudonym refills, where vehicles periodically receive a batch of pseudonyms from the VPKI authority. Although this approach requires, depending on pseudonym change rate and validity period, recurrent communication with the VPKI

issuing authority, it allows some sort of protection against Sybil attacks [6]. Revocation of certificates (and, more in general, of cryptographic keys) is quite always complex and possibly expensive [7]. Since pseudonym certificates do not include any vehicle identity information, the revocation of all the certificates associated with a misbehaving vehicle is not straightforward if no other additional mechanisms are implemented. Strawman approaches where the CA broadcasts large Certificate Revocation Lists (CRLs) including all the pseudonym certificates incur in linear costs with regard to the number of pseudonyms, and thus do not scale well. Standards avoid such high costs by using pseudo-random indexing identifiers (e.g., *linkage values* in SCMS [3]) included within the certificates exchanged in each V2X message, to enable authorities to reference the pseudonym certificates associated with a misbehaving vehicle by just releasing a single secret information. While the benefit is the distribution of small CRLs, the drawback is the need to send the indexing value within all V2X messages. Although such a value is currently quite small (9-Bytes in SCMS [3]), it still represents a significant disadvantage for V2X communications where network bandwidth is the major bottleneck, and, due to its dependence on hash functions' collision resistance, resizing may be needed in the future to support higher traffic workloads, in terms of number of vehicles and revocations, and privacy guarantees, in terms of number of pseudonyms assigned to each vehicle.

In this paper, we propose a novel method that uses *hierarchical deterministic homomorphic key derivation schemes* (HKD) to derive the key pairs used for signing V2X messages that do not require the inclusion of linkage values within messages. The messages exchanged by using this technique are smaller than those of standards based on explicit certificates, addressing one of the main bottlenecks of V2X communications. Network revocation costs are constant with regard to the number of pseudonyms assigned to vehicles and independent from traffic workloads, and no computational overhead is introduced for securing communications when compared to standards based on explicit certificates. Intuitively, HKD schemes allow to derive pseudo-random key pairs from a single master key pair in a deterministic way, while still ensuring that derived keys are unlinkable to each other and to the master derivation key pair. Also, when combining subsequent calls to the derivation routines, it is possible to build a hierarchical tree of key pairs that can be used to manage multiple pseudonyms in different time periods, while still being able to maintain unlinkability between keys and the same forward security property of standard linkage values. By leveraging these properties, our method allows to derive a large number of pseudonym key pairs from a single master key, thus reducing the storage requirements on both the vehicle and authority side and allowing efficient certificate issuance and revocation. We analyze the costs of our proposal both asymptotically and analytically when instantiated with the NIST p-256 elliptic curve recommended by standards [3], [4] and show that it is more efficient than standards based on explicit certificates in terms of network overhead, while still being affordable in terms of computational costs for key management operations. As a limitation, our approach can only be applied with explicit certificates, and we leave investigating the integration with implicit certificates as a future work.

Algorithm 1 BIP32 HKD specification using abstract algebra

```
1 : keygen() :
2 :   sk ←$ Z_q
3 :   PK ← sk · B
4 :   return ⟨sk, PK⟩

1 : sderive(sk, PK, info) :
2 :   sk_info ← sk + KDF(PK, info)
3 :   return sk_info

1 : pderive(PK, info) :
2 :   PK_info ← PK + KDF(PK, info) · B
3 :   return PK_info
```

In Section II we provide the notation and background knowledge. In Section III we present our system and threat model. In Section IV we define an abstract protocol framework for PKI-based V2X communications. In Section V, we present our efficient protocol specification for explicit pseudonym certificates. In Section VI, we discuss some asymptotic and analytic performance results. In Section VII we discuss related work, and in Section VIII we conclude the paper.

II. Notation and Background knowledge

A. Notation

We denote as $\mathbb{G}$ an additive cyclic group of prime order q built over an elliptic curve, where the Elliptic-Curve Discrete Logarithm Problem (ECDLP) is hard with regard to a security parameter λ. Let $\mathcal{O}$ be the neutral element of $\mathbb{G}$ and $B \in \mathbb{G}$ be the generator of $\mathbb{G}$. We slightly abuse notation and adopt $+$ and $\cdot$ both for operations on scalars belonging to $\mathbb{Z}_q$ and to elliptic curve points belonging to $\mathbb{G}$. Thus, $X+Y$ denotes the point addition operation for any $X, Y \in \mathbb{G}$, $s \cdot X$ denotes the point scalar operation for any $s \in \mathbb{Z}_q$ and $X \in \mathbb{G}$, and $a+b$ and $a \cdot b$ denote scalar addition and multiplication modulo q for any two $a, b \in \mathbb{Z}_q$. We assume that all parameters are public and agreed by all participants. For ease of notation, we omit them from inputs of routines and assume them as implicit.

We denote as $[a, b]$ the set of integers $[a, a+1, \ldots, b]$, and as $[a]$ the set of integers $[1, a]$.

We denote as $\sigma \leftarrow \mathsf{sign}(sk, m)$ the digital signature σ on message m with secret key sk, and as $\{\mathsf{accept}, \mathsf{reject}\} \leftarrow \mathsf{verify}(PK, m, \sigma)$ the routine for verifying a signature σ on message m with public key PK. We observe that sign may be probabilistic or deterministic, which may depend on the concrete specification, without affecting the validity of our proposal.

B. Hierarchical Deterministic Homomorphic Key Derivation and BIP32

Hierarchical Deterministic Homomorphic Key Derivation (HKD) schemes have recently become popular in the context of blockchains for managing so-called *deterministic wallets*. We consider an abstract version of the most popular scheme defined within BIP32 [8] for non-hardened addresses. An

HKD scheme is composed of three routines: *key generation* (keygen), for generating a freshly new key pair which acts as the root of the key hierarchy; *public key derivation* (pderive) and *secret key derivation* (sderive), for computing derived public and secret keys, respectively, with regard to some information $info$, which is a bitstring which acts as the scope of the key derivation, as for more typical symmetric key derivation schemes such as HKDF [9]. In Algorithm 1 we show specifications of the three routines, where $\mathsf{KDF} : \{0,1\}^* \rightarrow \mathbb{Z}_q$ denotes a standard key derivation function for symmetric keys [10] (e.g., implemented through HKDF in BIP32). For modeling correctness, we denote as $\{\mathsf{accept}, \mathsf{reject}\} \leftarrow \mathsf{check}(sk', PK')$ a routine which is able to verify whether PK' is a legitimate public key for secret key sk'. Let $\langle sk, PK\rangle \leftarrow\!\!\$\ \mathsf{keygen}()$ be the master key pair: correctness of HKD holds if, given $sk' \leftarrow \mathsf{sderive}(sk, PK, info)$ and $PK' \leftarrow \mathsf{pderive}(PK, info)$, $\mathsf{check}(sk', PK')$ accepts with probability 1. Intuitively, security requires the *unlinkability* of keys, i.e., secret and public keys generated from the same master key pair and different $info$ are computationally indistinguishable from freshly generated keys as long as the master key pair is not leaked. Other security guarantees are implied for unlinkability, among which the most important is the unforgeability of master keys and other derived keys by only knowing some derived key. However, it is important to remark that not all HKD specifications may generate derived keys which are *universally composable* with any other cryptographic scheme, in the sense of not introducing some type of vulnerability when used as if they were freshly generated random keys. In particular, while BIP32 is secure when used with ECDSA digital signatures [11], it may not be secure when straightforwardly combined with other types of cryptographic systems. Note that while other universally composable HKD schemes exist, their design prevent their adoption in the context of efficient communications (see Section VII). Combining subsequent calls to pderive and sderive allows to build a hierarchical tree of key pairs which can be used for managing multiple keys or construct deterministic wallets. The hierarchical approach starts from a root, the master key pair $\langle sk, PK\rangle$. By evaluating sderive and pderive for different $info$ values, one can obtain a number of level-1 derived nodes. Since each derived node can be used as a master key for further derivation, one can build a sub-tree of derived keys by recursively applying sderive and pderive to a derived node. We assume that, for each level i of the hierarchy, there is only one special $info$ value, that we call *master derivation label* (`mdl`), used to derive the child master key pair for the subsequent level $i+1$.

III. System and threat model

We consider a Vehicular credential management system based on a Public Key Infrastructure (VPKI) composed of a Certificate Authority (CA) and multiple vehicles which act as senders and/or receivers within the communication network[1]. Each vehicle communicates within the network by using many *pseudonyms*, and without ever using its own identity information. The privacy of the vehicle is preserved by using different pseudonym key pairs over time, as it is assumed that it is difficult to link two different pseudonyms to the same vehicle (see Section VII). To this aim, a vehicle must obtain a set of *pseudonym certificates* from the CA, which are cryptographic attestations, each binding a pseudonym (and possibly additional metadata) to cryptographic material used for authenticating messages. For a good security and performance trade-off, the CA divides time into time periods such that each pseudonym is valid only within a specific time period, and each vehicle can obtain a maximum fixed number $\mathtt{N}$ of pseudonym certificates for each time period. We identify time periods through enumeration using index $t = 0, 1, 2, \ldots$ to denote the t-th time period, and τ the current time period. We assume that CA and vehicles share a synchronized clock[2]. Moreover, the CA may define a parameter $\mathtt{T} \in \mathbb{N}$, such that a vehicle can request sets of pseudonym certificates for up to $(\mathtt{T}-1)$ future time periods[3]. Thus, a vehicle can obtain a maximum of $\mathtt{N} \cdot \mathtt{T}$ pseudonym certificates.

A CA may revoke all pseudonym certificates associated with a vehicle by distributing *revocation material* (rm), that is received by vehicles to build and maintain *Certificate Revocation Lists* ($\langle crl\rangle$). Note that, in comparison to well-known standard Web PKI where a $\langle crl\rangle$ is directly downloaded from a CA, for network efficiency a vehicle may need some extra processing to build each crl records from each rm and build $\langle crl\rangle$. Also note that a vehicle may need to update its $\langle crl\rangle$ for each new time period, to discard old records or build new ones. The CA owns a key pair $\langle sk_{CA}, PK_{CA}\rangle$, where sk_{CA} is the secret key for authenticating certificates and PK_{CA} is the public key used for verifying authenticity of certificates. Thus, the CA may determine the real identity of a vehicle from its pseudonym certificate even if the vehicle has not been revoked [4, Section 6.2.3 and 6.2.4].

We model the persistent storage of each vehicle by distinguishing four types of data:

- $\langle crt\rangle$, the set of pseudonym certificates that is valid in the current time period;
- $\langle crl\rangle$, the list of all pseudonym certificates that have been revoked by the CA in the current time period;
- crm (*certificate refresh material*), all the due data to refresh and/or prepare $\langle crt\rangle$ for the next time period;
- rrm (*revocation refresh material*), all the due data to refresh and/or prepare $\langle crl\rangle$ for the next time period.

[1]While the efforts of protecting the privacy of a participant in a vehicular network are typically focused on vehicles, the same principles can be applied to other participants, such as infrastructure nodes or even pedestrians. For simplicity, throughout the paper we will refer to vehicles as the only participant of the network that requires pseudonym certificates.

[2]Note that, like standards for vehicular communications, we are assuming a *low-precision* clock synchronization as typical for Web communication networks, since a typical period lasts a week [5].

[3]This happens within SCMS [3], but may not be defined in other standards. In this latter case, our analyses still apply by setting $\mathtt{T} = 1$.

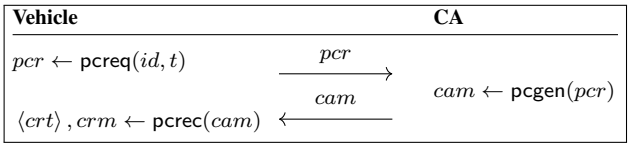

Fig. 1: Pseudonym certificates release

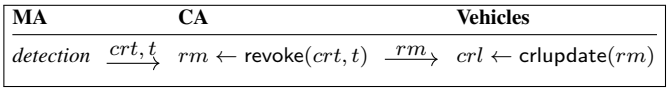

Fig. 2: Revocation

We consider passive and active network adversaries for communications among vehicles, while for ease of presentation we assume trusted communications between vehicles and the CA. We assume that PK_{CA} is known by all vehicles and that the distribution of public keys associated with vehicles depends on protocols specifications.

IV. PPV PROTOCOL FRAMEWORK

We propose an abstract protocol framework for describing the management of certificates by a PKI in the context of Pseudonym-based authenticated Vehicular communications, which we denote as PPV. PPV is composed of three sub-protocols: *pseudonym certificate release*, where a vehicle requests and obtains a set of pseudonym certificates from the CA; *revocation*, where a CA receives a notification about a vehicle misbehavior and revokes all the pseudonym certificates associated with the vehicle; *refresh*, where a vehicle updates its certificates and revocation lists for the following time period. Note that, while we only focus on the main features of VPKI management, for which we propose novel contributions, other sub-protocols may be included to cover other features of existing VPKI standards (e.g., the initial registration of vehicles through an Enrollment Authority [4]).

Pseudonym certificates release, shown in Figure 1, is composed of three routines:

- *pseudonym certificates request* (pcreq): the vehicle creates a homonym data structure (that we distinguish from the routine name by denoting it as pcr) from the vehicle identifier id and time span information Δ_t (i.e., the number of total subsequent time periods, *overall covered time span*, for which the vehicle is requesting pseudonym certificates, including the current time period τ);
- *pseudonym certificates generation* (pcgen): the CA creates a *certificate approval material* (cam) comprising data which assess the approval of the input pcr by the CA, but which may not include complete pseudonym certificates due to the need for further processing by the vehicle (for security and/or performance reasons);
- *pseudonym certificates reception* (pcrec): the vehicle processes cam and produces a set of pseudonym certificates $\langle crt \rangle$, which can be used by the vehicle to authenticate messages within the current time period τ, and certificate refresh material crm, which is used during refresh to generate $\langle crt \rangle$ for the following $(\Delta_t - 1)$ time periods;

Revocation, shown in Figure 2, is composed of two routines:

- *revoke* (revoke): the CA creates *revocation material* (rm) upon reception of the pseudonym certificate crt of a misbehaving vehicle at some time period t (which we show as the output of an abstract *detection* event observed by the Misbehavior Authority, MA);
- *CRL update* (crlupdate): the vehicle processes rm to create a crl record containing all the pseudonym certificates associated with a misbehaving vehicle.

Note that all crl output data obtained from multiple executions of the *Revocation* protocol are appended to the $\langle crl \rangle$ data structure, which is maintained within the persistent storage of the vehicle, as anticipated in Section III.

Refresh is composed of two routines, both executed by a vehicle:

- *certificate refresh* ($\{\perp \mid \langle crt \rangle'\} \leftarrow \mathsf{crtrefresh}(\langle crt \rangle, crm, t)$) updates the current set of pseudonym certificates $\langle crt \rangle$ for the next time period t, thus creating $\langle crt \rangle'$. The routine may fail and output $\perp$ if crm is empty, i.e., there is no certificate refresh material, which was released by the CA, to create a new set of pseudonym certificates for the next time period.
- *revocation list refresh* ($\{\perp \mid \langle crl \rangle'\} \leftarrow \mathsf{crlrefresh}(\langle crl \rangle, rrm, t)$) updates the current $\langle crl \rangle$ for the next time period t, possibly deleting crl associated with pseudonym certificates released more than T time periods ago. The routine may fail and output $\perp$ if rrm is empty, i.e., there are no revoked vehicles in the future time period.

V. NETWORK-EFFICIENT HKD-BASED PPV PROTOCOL SPECIFICATION

We describe a network-efficient PPV protocol specification (Section IV) based on homomorphic key derivation (HKD) functions (Section II). We describe certificates release in Section V-A, revocation in Section V-B, and refresh in Section V-C. We conclude with some final remarks on design choices and potential variants in Section V-D.

We recall from Section III that a vehicle can request a set of N pseudonym certificates for each time period, for a total of T time periods, starting from the current time period τ and up to $\mathtt{T} - 1$ time periods in the future.

A. Pseudonym certificates release

Figure 3 shows the specification of the *pseudonym certificates release* protocol including three routines: pcreq, pcgen, and pcrec. We assume that the vehicle is requesting pseudonym certificates for the first time at some time period τ, thus $\langle crt \rangle$, $\langle crl \rangle$, crm and rrm are empty at the beginning of the protocol. The vehicle generates a master key pair ($\langle sk, PK \rangle \leftarrow\!\!\$\ \mathsf{keygen}()$), and creates a pseudonym certificates request containing the master public key PK and the number of future time periods for which the vehicle is requesting pseudonym certificates Δ_t, where $\Delta_t \leq \mathtt{T}$. If the vehicle

	Vehicle	**CA**
	$\langle crt\rangle, \langle crl\rangle, crm,$	$sk_{CA}, PK_{CA},$
	rrm, PK_{CA}	$\mathtt{S}_t : [\langle PK_n^\tau\rangle_{n\in[\mathtt{N}]}^v]_{v\in[V]} \rightarrow [PK_0^v]_{v\in[V]}$
1 :	$\langle sk, PK\rangle \leftarrow\!\$$ keygen()	
2 :	$pcr = \langle PK, \Delta_t\rangle$	
	$\longrightarrow$	
3 :		**if** $\Delta_t > \mathtt{T}$: **reject**
4 :		$PK_0^\tau = PK$
5 :		**foreach** $t \in [\tau+1, \tau+\Delta_t-1]$:
6 :		$PK_0^t \leftarrow$ pderive$(PK_0^{t-1}, \mathtt{mdl})$
7 :		**foreach** $t \in [\tau, \tau+\Delta_t-1]$:
8 :		**foreach** $n \in [\mathtt{N}]$:
9 :		$PK_n^t \leftarrow$ pderive(PK_0^t, n)
10 :		$\sigma_n^t \leftarrow$ sign$(sk_{CA}, \langle PK_n^t, t\rangle)$
11 :	$cam = \left\langle \langle \sigma_n^t\rangle_{n\in[\mathtt{N}]}\right\rangle_{t\in[\tau,\tau+\Delta_t-1]}$	
	$\longleftarrow$	
12 :	**foreach** $n \in [\mathtt{N}]$:	
13 :	$PK_n^\tau \leftarrow$ pderive(PK, n)	
14 :	$\langle crt\rangle \leftarrow \langle PK_n^\tau, \sigma_n^\tau\rangle_{n\in[\mathtt{N}]}$	
15 :	$crm \leftarrow \left\langle \langle \sigma_n^t\rangle_{n\in[\mathtt{N}]}\right\rangle_{t\in[\tau+1,\tau+\Delta_t-1]}$	

Fig. 3: Specification of *pseudonym certificates release*

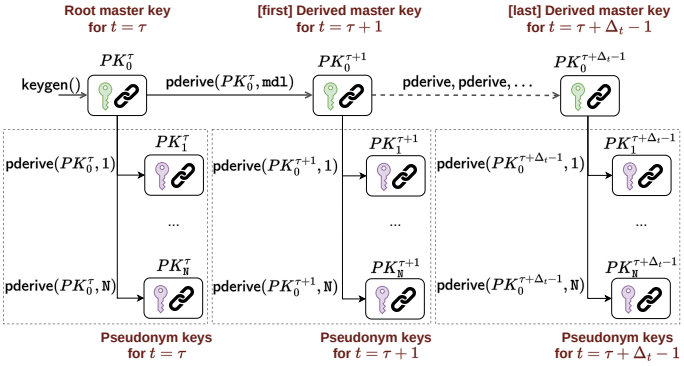

Fig. 4: Derivation of public keys spanning Δ_t time periods.

is requesting pseudonym certificates for more than one time period ($\Delta_t > 1$), the CA must derive the master public key PK_0^t for each time period $t \in [\tau+1, \tau+\Delta_t-1]$ by using pderive, where $info$ is set to the reserved value mdl (Lines 4-6). A graphical example of key derivation operated by the pseudonym certificates release protocol is shown in Figure 4. Then, for each time period $t \in [\tau, \tau+\Delta_t-1]$ (where $PK_0^\tau = PK$), the CA derives N pseudonym public keys PK_n^t from PK_0^t using pderive, where $info = n, \forall n \in [\mathtt{N}]$, and signs them (together with the related time period t for which the keys are valid) (Lines 7-10). For network efficiency, only digital signatures are sent back to the vehicle within *certificate approval material* cam. The vehicle receives cam and computes pseudonym public keys for the current time period (Lines 12-13), postponing computation of public keys for following time periods during *refresh*. Thus, certificates for the current time period ($\langle PK_n^\tau, \sigma_n^\tau\rangle_{\forall n\in[\mathtt{N}]}$) are saved in $\langle crt\rangle$, while other signatures ($\langle \sigma_n^t\rangle_{t\in[\tau+1,\tau+\Delta_t-1], n\in\mathtt{N}}$) are stored in crm (Lines 14-15).

Remarks. Note that, for computational efficiency at revocation time, our protocol requires the CA to maintain all the master public keys PK that each registered vehicle has sent during the pseudonym certificate release using a reverse hash map $\mathtt{S}_t$. At time period t, $\mathtt{S}_t$ maps each pseudonym public key PK_n^t to the corresponding master public key PK_0^t, for each vehicle $v \in [V]$ (where V is the total number of vehicles registered in the VPKI). In Figure 3 (and Figure 5) we denote this reverse hash map as a function $\mathtt{S}_t : [\langle PK_n^\tau\rangle_{n\in[\mathtt{N}]}^v]_{v\in[V]} \rightarrow [PK_0^v]_{v\in[V]}$, slightly abusing notation when writing PK_0^v to denote the master public key of the v-th vehicle. Also, in our specification we omit verification of the vehicle's identity id by the CA, which is orthogonal to the protocol (e.g., checking inclusion within a database of registered vehicles). Moreover, in order to minimize the size of messages sent over the network, the CA does not send the derived pseudonym public keys PK_n^t to the vehicle, but only the signatures σ_n^t and let the vehicle derive the pseudonym public keys on its own. This is possible because of the deterministic nature of the HKD scheme, but variants may let the CA also send public keys for easing computation at the vehicle side. Finally, note that at the end of the pseudonym certificates release protocol we let the vehicle only derive public keys, and not secret keys. This is due to the very low computation cost of the sderive routine, which includes the computation of a KDF function (that is, the execution of a few hash functions or block ciphers, depending on the specification [12]) and of a scalar addition operation, which is negligible compared to message signing. Thus, secret key derivation can be executed at runtime before sending the message, and vehicles can avoid storing (up to) N secret keys (see Section VI), which is a clear advantage with regard to standard VPKI protocols.

	CA	**Vehicle**
	$\langle crt\rangle, \langle crl\rangle, crm,$	$sk_{CA}, PK_{CA},$
	rrm, PK_{CA}	$\mathtt{S}_t : [\langle PK_n^\tau\rangle_{n\in[\mathtt{N}]}^v]_{v\in[V]} \rightarrow [PK_0^v]_{v\in[V]}$
1 :	$PK_n^t \leftarrow crt$	
2 :	$PK_0^t \leftarrow \mathtt{S}_t[hash(PK_n^t)]$	
3 :	$rm = PK_0^t$	
	$\longrightarrow$	
4 :		**foreach** $n \in [\mathtt{N}]$:
5 :		$PK_n^t \leftarrow$ pderive(PK_0^t, n)
6 :		$\langle crl\rangle$ .add$(\langle PK_n^t\rangle_{n\in[\mathtt{N}]})$
7 :		$PK_0^t \leftarrow$ pderive$(PK_0^t, \mathtt{mdl})$
8 :		rrm.add(PK_0^t)

Fig. 5: Specification of *pseudonym certificates revocation*.

B. Pseudonym certificates revocation

Figure 5 shows details of the specification for *pseudonym certificates revocation*. Let crt be the certificate observed by the Misbehavior Authority (MA) sent within some malicious message at time period t, the CA only needs to publish a revocation material (rm) containing the master public key PK_0^t of the misbehaving vehicle, where t is the time period

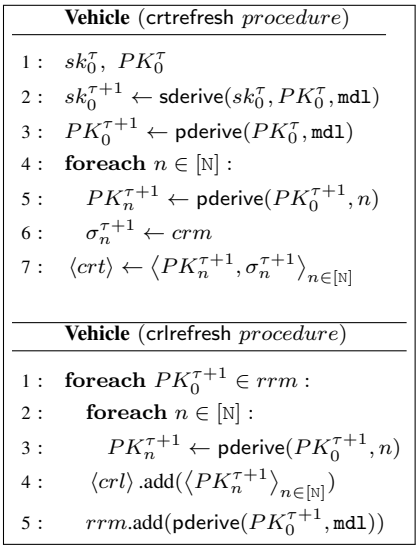

Fig. 6: Specification of *refresh*.

in which the misbehavior was detected, to revoke all its pseudonym certificates. The CA may also include an additional information if the revocation spans multiple time periods, i.e., if the misbehaving vehicle had requested pseudonym certificates for more than one time period, the revocation material must be valid also for those future time periods. The CA may also prevent the vehicle from requesting new pseudonyms in the future [3], [4]. A vehicle that receives rm can then derive all the pseudonym public keys associated with the misbehaving vehicle using the pderive routine for all $n \in \mathrm{N}$ with the master public key PK_0^t contained in the CRL. The vehicle stores all the derived pseudonym public keys for time period t in $\langle crl \rangle$, and stores in rrm the master public key PK_0^{t+1} for the next time period $t+1$, if required. When a vehicle receives a message, it can check if the message is signed with a pseudonym public key that is listed in $\langle crl \rangle$ and, possibly, discard the message.

Remarks. Note that, in Figure 5 (and Figure 6) we denote as .add() the addition of an element to a set without removing previous elements, if any. A critical advantage of adopting an HKD scheme is its support for efficient certificate revocation. Rather than referencing pseudonym certificates through global opaque identifiers (e.g., Serial Number in X.509 certificates [13] or linkage values in SCMS [3]), our protocol allows to derive all the pseudonyms of a vehicle by knowing only the master public key. This allows the CA to revoke all pseudonym certificates associated with a vehicle at some time period t (and for future time periods, if any) by simply publishing its master public key PK_0^t, without the need to send all the pseudonym certificates keys or identifiers. More importantly, V2X messages do not need to include any identifier, because at revocation time all vehicles can use pderive with a fixed set of known predefined indexes ($n \in [\mathrm{N}]$).

C. Certificate refresh

Figure 6 describes the specification of the *refresh* protocol which includes crtrefresh and crlrefresh. A vehicle executes crtrefresh if, at the end of the current time period τ, it has at least one set of signatures stored in crm. In other terms, if it successfully obtained pseudonym certificates during some time period t using as input $\Delta_t \geq (\tau - t)$. Otherwise, it must request a new set of explicit pseudonym certificates from the CA. The crtrefresh procedure allows the vehicle to derive a new master key pair $((sk_0^{\tau+1}, PK_0^{\tau+1}))$ for the next time period $\tau+1$ (i.e., invoking sderive and pderive using the current master key pair and the master derivation label mdl). Then, the vehicle derives all the pseudonym public keys $PK_n^{\tau+1}$ for the next time period, and generates the corresponding explicit certificates, to be stored in $\langle crt \rangle$, by using the signatures from crm. The unused signatures for time periods $t > \tau+1$ are kept stored in crm for future invocations of crtrefresh. Whether the vehicle has refreshed $\langle crt \rangle$ from a previous set of signatures stored in crm or from a new invocation of the pseudonym certificate release protocol, it must also update its set of revoked pseudonym public keys for the next time period. The crlrefresh procedure allows the vehicle to update $\langle crl \rangle$ as follows: for each revoked master public key (for $\tau+1$) stored in rrm the vehicle derives all the revoked pseudonym public keys $PK_n^{\tau+1}$ for the next time period, and stores them in $\langle crl \rangle$. The crlrefresh procedure ends by saving in rrm the master public keys for future time period $t > \tau+1$ (i.e., invoking pderive using the current master public key and the master derivation label mdl).

D. Final remarks

The proposed architecture reduces the storage requirements when compared to traditional VPKI systems. Vehicles only need to store a single master private key and a set of explicit certificates that are smaller than the pseudonym certificates containing global linkage identifiers [3], [14]. The scalability of the architecture is also improved, as the VPKI authority only needs to store the master public keys for each registered vehicle, rather than maintaining a large database of individual pseudonym certificates or revocation identifiers. Section VI provides a detailed analysis of the storage and communication overhead of the proposed architecture.

While implicit certificates (e.g., ECQV [15] or SIMPL [16]) offer superior performance in terms of bandwidth efficiency, they rely on elliptic curve cryptography and do not currently extend to post-quantum settings [17], while, in contrast, explicit certificates are compatible with post-quantum signing schemes. We leave the integration of implicit certificates as a future work. Nevertheless, the use of HKD schemes still provide benefits in terms of scalable pseudonym generation and efficient revocation, even when combined with explicit certificates in post-quantum scenarios.

VI. Cost evaluation

We discuss costs of the proposed HKD-based PPV protocol specification (see Section V) and compare them with existing

Protocol	Network cost
Release	$O(\mathtt{N} \cdot \mathtt{T})$
Revocation	$O(1)$
Refresh	-

TABLE I: Asymptotic costs for network

Protocol	Procedure	Computation cost	Made by
Release	pcreq	$O(1)$	Vehicle
	pcrec	$O(\mathtt{N})$	
	pcgen	$O(\mathtt{N} \cdot \mathtt{T})$	CA
Revocation	revoke	$O(1)$ or $O(\mathtt{N})$	
	crlupdate	$O(\mathtt{N})$	Vehicle
Refresh	crtrefresh	$O(\mathtt{N})$	
	crlrefresh	$O(\mathtt{N} \cdot \mathtt{T})$	

TABLE II: Asymptotic costs for computation

standards both asymptotically (Section VI-A) and analytically when instantiated with standard security parameters (Section VI-B).

A. Asymptotic analysis

We analyze asymptotic costs in terms of network and computation costs for the Release, Revocation and Refresh protocols, and in terms of storage size used by the vehicle for maintaining $\langle crt \rangle$, $\langle crl \rangle$, crm and rrm.

Table I shows asymptotic network costs, which we want to minimize because they are the most critical bottleneck in vehicular networks, especially when considering large number of vehicles. During Release, vehicles send a *pseudonym certificates request* to the CA including the master public key of the vehicle, the CA replies with all the signatures for each pseudonym certificate $\mathtt{N}$ in each time period $\mathtt{T}$, thus Release network cost is $O(\mathtt{N} \cdot \mathtt{T})$. During Revocation, the CA sends a single master public key to all the vehicles, which is used to revoke all the pseudonym certificates with public keys that have been derived from that master public key, regardless of their number, thus Revocation network cost is $O(1)$. Refresh does not require any network communication, as vehicles can update their local storage with the new pseudonym certificates and revocation lists without interacting with CAs.

While network costs are more critical in vehicular communications, it is important that computational costs are still affordable and kept below a certain threshold to comply with allowed latencies of safety-related features. Our proposal does not modify verification algorithms and thus safety-related features are not influenced. When dynamically deriving private keys, signing costs are only increased by a negligible amount (see remarks of Section V-A). As for the CA, we assume that has enough computational resources to perform all the cryptographic operations required to manage the pseudonym certificates, and thus we do not consider the CA as bounded by the same constraints as the vehicles (the same applies for network communication, where we assume that the CA is not constrained by the same limitations as the vehicles).

Type of data	Storage	Storage cost
Pseudonym certificates valid in τ	$\langle crt \rangle$	$O(\mathtt{N})$
Certificates revoked by the CA in τ	$\langle crl \rangle$	$O(\mathtt{N} \cdot \mathtt{T})$
Certificate refresh material	crm	$O(\mathtt{N} \cdot \mathtt{T})$
Revocation refresh material	rrm	$O(\mathtt{N} \cdot \mathtt{T})$

TABLE III: Asymptotic costs for storage

Table II shows the asymptotic computation costs for certificate management. During Release, the vehicle performs a pseudonym certificate request (pcreq) that computes the master key pair, whose computation is independent of any protocol parameter except the security level, and thus its computation cost is $O(1)$. Then, CA performs the pseudonym certificate generation (pcgen) that requires up to $O(\mathtt{N} \cdot \mathtt{T})$ computation cost to generate all the pseudonym certificates for each time period requested by the vehicle. Note that we assume the worst case where the CA has to generate all the pseudonym certificates for the maximum allowed number of time periods $\mathtt{T}$. This assumption is justified by our main design goal of minimizing the network communication cost, thus reducing interactions among vehicles and CAs. Finally, the vehicle processes the certificate approval material (pcrec) to generate the $\mathtt{N}$ pseudonym certificates for the current time period, thus requiring $O(\mathtt{N})$ computation cost. During Revocation protocol, the CA performs the revocation operation (revoke) that requires a constant computation cost of $O(1)$ to revoke a master public key, thanks to the reverse hash map (see Section V-C). Different approaches where the CA does not maintain such a data structure, and thus has to perform more expensive search operations, could also be chosen to reduce the CA storage requirements, however we do not see any realistic scenario where a CA has such constrained storage. The vehicle then processes the certificate revocation list update (crlupdate) to update its local revocation list with the new revoked vehicle, thus requiring $O(\mathtt{N})$ computation cost to reconstruct all the pseudonym public keys of the revoked vehicle. During Refresh, the vehicle operates the certificate refresh routine (crtrefresh) and the revocation list refresh (crlrefresh) to update its local storage with new pseudonym certificates and revocation lists to be used in the next time period, which cost $O(\mathtt{N})$ and (up to) $O(\mathtt{N} \cdot \mathtt{T})$, respectively.

Table III shows storage asymptotic costs. Vehicles store all the pseudonym certificates valid in the current time period in $\langle crt \rangle$, that is, derived public keys and their associated signatures, thus storage cost is $O(\mathtt{N})$. Vehicles also store revocation list $\langle crl \rangle$, which contains $\mathtt{N}$ pseudonym certificates for each vehicle that has been revoked by the CA in the current time period. However, since each vehicle can request pseudonym certificates for up to $\mathtt{T}$ time periods, the total storage required is $O(\mathtt{N} \cdot \mathtt{T})$ for each revoked vehicle. Furthermore, vehicles store certificate refresh material crm, which includes the signatures of pseudonym certificates that are valid in future time periods, thus its storage cost is also $O(\mathtt{N} \cdot \mathtt{T})$. Finally, vehicles store revocation refresh material rrm, which includes

master public keys associated with the next time periods, thus storage cost is again $O(\mathtt{N} \cdot \mathtt{T})$.

B. Analytic evaluation for 128-bit security

We provide an analytic evaluation of our protocol and we compare it with the SCMS standard [3]. We consider an instantiation based on the ECDSA signature scheme [18] with the NIST p-256 elliptic curve [19], thus with a 128-bit security level, as recommended by current standards for V2X communications [4], [20].

Certificate size. The SCMS architecture [3] relies on opaque indexing identifiers called *linkage values*, which are included in each pseudonym certificate released by the CA (similar to the Serial Number in X.509 certificates [13]). Linkage values *globally* identify pseudonym certificates among all vehicles, except for a certain probability of collision which depends on its size. The current recommended size for linkage values is 72 bits. The authors of SCMS suggest a flexible design of the linkage value to account for the increasing number of connected vehicles and potential weakness of the underlying cryptographic primitives in terms of collision resistance [3]. However, allowing such flexibility in real-world embedded systems may not be practical. Modifying the size of the linkage value after the system has been deployed would require a significant effort in terms of software and hardware updates, which is not feasible in many cases. $\mathtt{N}$ is defined by the VPKI authority at setup time, and allows a trade-off between privacy, security and scalability: the larger the value, the more pseudonym identities a vehicle can use within a time period, thus increasing anonymity (i.e., preventing identification and tracking based on a series of eavesdropped messages [21]), but also increasing storage requirements and risk of Sybil attacks [6]. We refer to the example from the discussion about the SCMS linkage value length in [3, Section V.C] to better understand the impact of this design choice. Consider 2.5×10^8 vehicles, each having $\mathtt{N} = 40$ pseudonym certificates per time period of one week, then in any time period there are $2.5 \times 10^8 \times 40 = 10^{10}$ pseudonym certificates in use. Revocation rate is assumed below 1%. In this setting, exploiting the birthday paradox [22], the chance of a collision between linkage values (i.e., the linkage value of a revoked vehicle gets assigned also to another vehicle in the same time period) is $\approx 1 - \exp(-((10^{10}/10^2) \times (10^{10}/10^2 - 1))/(2 \times 2^{72})) \approx 1.06 \times 10^{-6}$. While this probability is small, it is not *cryptographically* negligible [23], especially considering that the SCMS system is designed to be used for many years, and must account for the growth in the number of vehicles and pseudonym certificates over time. To obtain a collision probability that is *cryptographically* negligible, the size of the linkage value should be increased to at least 85 bits, which would result in a collision probability of $\approx 2^{-32} \approx 1.29 \times 10^{-10}$, further increasing the size of all certificates and thus the network congestion. In contrast, our protocol does not require inclusion of any identifier within a certificate, because certificate revocation is only based on cryptographic keys (Section V-B). Since our approach aims to minimize the network communication cost, a reduction of 72 bits per certificate is significant, especially when considering the large amount of messages exchanged in a V2X network. Moreover, our approach is more *future-proof*, because its parameters are independent of workloads.

Certificate Revocation List size. In SCMS, the revocation of a vehicle is performed by distributing a Certificate Revocation List (CRL) that allows vehicles to reconstruct all the linkage values of the pseudonym certificates issued to the revoked vehicle. Two Linkage Authorities (LAs)[4] are involved in the revocation process: the CRL includes two linkage value seeds (i.e., hash chains) and the identities of the two LAs. The total size of each CRL is $2 \times 128 + 2 \times 32 = 320$ bits, where 128 bits are the linkage seeds created by the LAs and 32 bits are identity strings associated with the LAs. When a vehicle receives a CRL, it can reconstruct all the linkage values of the certificates of the revoked vehicle. On the vehicle side this means that a single CRL expands to $\mathtt{N}$ linkage values, each of size 72 bits, for a total of $72 \cdot \mathtt{N}$ bits that must be stored to check the revocation status of each V2X message received. The authors of SCMS assume that all vehicles will provide at least enough storage for 10000 CRL entries (where a single CRL entry corresponds to a revoked vehicle), which translates to $320/8 \times 10000 = 400$ KB. But, when expanding each CRL entry, the total required storage space is $72\mathtt{N}/8 \times 10000 = 90\mathtt{N}$ KB, which, for $\mathtt{N} = 40$, results in 3600 KB. In our architecture, the revocation of a vehicle is performed by distributing a CRL that includes the master public key of that vehicle, which is of size 256 bits for the NIST p-256 elliptic curve [19]. When a vehicle receives a CRL, it can reconstruct all the pseudonym public keys issued to the revoked vehicle by deriving the set of pseudonym public keys from the master pseudonym public key. Assuming the same number of CRL entries, the total required size is $256/8 \times 10000 = 320$ KB, before expansion. When expanding each CRL entry, the total required storage space for $\mathtt{N} = 40$ is $256/8 \times 40 \times 10000 = 12800$ KB. This is $\sim 3.5\times$ larger than the SCMS approach for explicit certificates, however, this is still acceptable in practice, as the increased storage space requirements are manageable for modern vehicles. This difference in storage requirements holds also when compared to the SCMS approach for implicit certificates, as implicit certificates in SCMS use the same linkage value mechanism as explicit certificates, thus requiring the same CRL storage space for 10000 entries.

VII. Related Work

Vehicular Public Key Infrastructures. The infrastructure for vehicular networks in US and European standards is based on PKI, and the differences in the (hierarchical) architecture from the root CA to the vehicles allow for multifold approaches to certificate distribution, management and revocation, with different trade-offs in terms of security and privacy [1], [2].

[4]This is due to the different VPKI architecture: in SCMS there are four different authorities, and the authority involved in certificate revocation is the Linkage Authority.

Existing Vehicular PKI (VPKI) architectures balance the trade-off between minimizing V2X message size via compact certificates (e.g., ECQV implicit certificates) [3], [16] and managing the overhead of revocation mechanisms for misbehaving vehicles [7], [24]–[27]. The cost of administration and revocation of pseudonym certificates is a key factor in the design of VPKI architectures, as it directly impacts the scalability and efficiency of the system. Literature which proposed privacy improvements typically, even with affordable computational costs, have drawbacks in terms of certificate revocation and CRL distribution [3], [16], [28]–[30]. Our protocol design takes into account trade-offs between the size of a V2X message and the size of the revocation lists, tackling network and revocation costs, and creating V2X messages that are smaller than those of SCMS [3] while providing the same privacy-preserving features based on pseudonyms to guarantee unlinkability. Moreover, as our protocol parameters do not depend on communications workload as for SCMS linkage value size, given enough vehicle storage and communication bandwidth during certificate management operations, our protocol can scale better when provisioning more pseudonyms to each vehicle. Other approaches for preventing tracking of vehicles in V2X communications include pseudonym swapping techniques [31], [32], identity-based solutions [33], and decentralized key management mechanisms [24], [34]. These approaches have several drawbacks: non-practical performance in real-world scenarios due to heavy cryptographic operations, pseudonym collisions or incomplete swaps due to frequent changes in the vehicular network topology, additional complexity and overheads in distributing CRLs and/or requesting new pseudonyms, inability to remove misbehaving vehicles from the network, scaling issues with an increasing number of connected vehicles, or requiring frequent communication with infrastructure nodes. We differ from these approaches by proposing a solution that is simple, centralized and highly scalable, achieving the same level of unlinkability and privacy as current V2X standards, while allowing for a more efficient V2X message size and bandwidth usage. The centralized nature of our architecture follows the approach of [25], which aims to simplify the VPKI architecture and reduce the attack surface by using a single CA authority.

Hierarchical deterministic key derivation schemes. Hierarchical deterministic key derivation schemes (HKD) are widely used in the context of cryptocurrencies and blockchain technologies to manage multiple keys from a single master key [8], [11], [35]. The Arcula [36] HKD scheme provides a robust solution for Bitcoin wallets that brings identity-based signatures to the blockchain, and evaluate its usage in a real-world scenario. Parallel research efforts have been made to extend the use of HKD schemes to other contexts, such as WebAuthn authenticators [37], [38], and to improve their security properties [36], [39]–[41], also in distributed [42], [43] and post-quantum [44] settings. ARKG [38] firstly demonstrated the use of HKD schemes for generating unforgeable signatures in WebAuthn. Follow up works [39], [41] improved the security properties of ARKG, allowing its use with a wider range of signature schemes and pairing-based cryptosystems. [40] tackles global key revocation in FIDO2, presenting a revocation procedure based on the BIP32 standard, which can be efficiently implemented and addresses a key challenge in decentralized systems. As for distributed settings, the authors of [42] present the design of a HKD protocol for DL-based threshold cryptosystems, and the scheme by [43] enhances ARKG by allowing the reconstruction of private keys in distributed settings, though they suffer from high communication or verification costs, respectively. Due to the great interest in HKD for blockchain applications, similar schemes are under study for post-quantum contexts, while the creation of implicit certificate schemes seems more uncertain [17]. [44] introduces alternative security definitions for ARKG, focusing on secure instantiations from KEMs and standard signature schemes, with an emphasis on post-quantum security. To the best of our knowledge, HKD have not been previously applied to the context of PKI-based communications, certificate management and vehicular networks.

VIII. Conclusions

We proposed a novel protocol for communications based on pseudonyms with low network costs for authenticated messages and certificate management operations, including revocation. Our work can be applied to V2X communications, where network costs represent major bottlenecks, and represents a novel application of hierarchical deterministic homomorphic key derivation schemes for distributing *conditionally unlinkable* pseudonym certificates. Previous works on hierarchical key derivation schemes do not consider the application to vehicular networks, nor the problem of certificate revocation. Our solution can be used to design scalable and efficient vehicular credential management systems that meet the requirements of V2X communications. We demonstrated that our approach significantly reduces network overhead by removing the need for linkage values in all V2X messages, while supporting efficient revocation at constant network costs. The higher computational costs for key management operations are still affordable in practice, and there is no computational overhead for securing communications, all under the same cryptographic assumptions of current standards. Future work includes the evaluation of further optimizations to allow even more efficient certificate management, such as using implicit certificates, and application to future V2X communication protocols based on post-quantum cryptography.

Acknowledgments

This work has been partially supported by the project "FuSeCar" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2022 - grant 2022W3EPEP.

References

[1] M. AlMarshoud, M. Sabir Kiraz, and A. H. Al-Bayatti, "Security, privacy, and decentralized trust management in vanets: A review of current research and future directions," *ACM Computing Surveys*, 2024.

[2] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE Comm. Surveys & Tutorials*, 2014.

[3] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for V2X communications," *IEEE Trans. Intelligent Transportation Systems*, 2018.

[4] ETSI TS 102 731 V2.0.0, "Intelligent Transport Systems (ITS); Security; Security Services and Architecture," 2022.

[5] N. Bißmeyer, H. Stübing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc, "A generic public key infrastructure for securing car-to-x communication," in *18th ITS World Congress, Orlando, USA*, 2011.

[6] J. R. Douceur, "The sybil attack," in *Int'l. workshop peer-to-peer systems*, 2002.

[7] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE journal selected areas in communications*, 2007.

[8] P. Wuille, "Bitcoin Improvement Proposal (BIP) 32: Hierarchical Deterministic Wallets," 2012. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

[9] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," RFC 5869, 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5869

[10] L. Chen, "Recommendation for key derivation using pseudorandom functions," 2024. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=957462

[11] P. Das, S. Faust, and J. Loss, "A formal treatment of deterministic wallets," in *Proc. ACM CCS*, 2019.

[12] L. Chen and L. Chen, *Recommendation for key derivation using pseudorandom functions*. US Department of Commerce, National Institute of Standards and Technology, 2024.

[13] P. E. Yee, "Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 6818, 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6818

[14] ETSI TS 103 097 V2.1.1, "Intelligent Transport Systems (ITS); Security; Security header and certificate formats," 2021.

[15] C. Research, "SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)," 2004, Standards for Efficient Cryptography. [Online]. Available: https://www.secg.org/sec4-1.0.pdf

[16] P. S. L. M. Barreto, M. A. Simplicio, J. E. Ricardini, and H. K. Patil, "Schnorr-based implicit certification: Improving the security and efficiency of vehicular communications," *IEEE Trans. Computers*, 2021.

[17] G. Twardokus, N. Bindel, H. Rahbari, and S. McCarthy, "When cryptography needs a hand: Practical post-quantum authentication for v2v communications," in *NDSS Symp.*, 2024.

[18] N. I. of Standards, T. (NIST), L. Chen, D. Moody, A. Regenscheid, and A. Robinson, "Digital signature standard (dss)," 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935202

[19] L. Chen, D. Moody, K. Randall, A. Regenscheid, and A. Robinson, "Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters," 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935198

[20] V2X Core Technical Committee, *SAE J 2735 - V2X Communications Message Set Dictionary*, 2024.

[21] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *Int'l. Conf. wireless on-demand network systems and services*, 2010.

[22] H. Davenport, *The higher arithmetic: An introduction to the theory of numbers*. Cambridge University Press, 1999.

[23] M. Dworkin, "Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac," 2007. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51288

[24] S. Bao, A. Lei, H. Cruickshank, Z. Sun, P. Asuquo, and W. Hathal, "A pseudonym certificate management scheme based on blockchain for internet of vehicles," in *IEEE Int'l. Conf. Dependable, Autonomic and Secure Computing*, 2019.

[25] M. A. Simplicio, E. L. Cominetti, H. K. Patil, J. E. Ricardini, L. T. Ferraz, and M. V. M. Silva, "Privacy-preserving certificate linkage/revocation in vanets without linkage authorities," *IEEE Trans. Intelligent Transportation Systems*, 2020.

[26] S. Khan, F. Luo, Z. Zhang, M. A. Rahim, S. Khan, S. F. Qadri, and K. Wu, "A privacy-preserving and transparent identity management scheme for vehicular social networking," *IEEE Trans. Vehicular Technology*, 2022.

[27] H. Wan, Q. Wang, C. Ma, Y. Teng, J. Lin, and D. Ye, "Escort: Efficient status check and revocation transparency for linkage-based pseudonym certificates in vanets," in *IEEE Symp. Computers and Communications*, 2023.

[28] P. Vijayakumar, M. Azees, and L. J. Deborah, "Cpav: Computationally efficient privacy preserving anonymous authentication scheme for vehicular ad hoc networks," in *IEEE Int'l. Conf. cyber security and cloud computing*, 2015.

[29] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intelligent Transportation Systems*, 2017.

[30] S. Wang and N. Yao, "Liap: A local identity-based anonymous message authentication protocol in vanets," *Computer Communications*, 2017.

[31] P. K. Singh, S. N. Gowtham, S. Nandi *et al.*, "Cpesp: Cooperative pseudonym exchange and scheme permutation to preserve location privacy in vanets," *Vehicular Communications*, 2019.

[32] A. P. Mdee, M. T. R. Khan, J. Seo, and D. Kim, "Security compliant and cooperative pseudonyms swapping for location privacy preservation in vanets," *IEEE Trans. Vehicular Technology*, 2023.

[33] L. Zhang, "Otibaagka: A new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE Trans. Information Forensics and Security*, 2017.

[34] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for vanet with blockchain," *IEEE Trans. Vehicular Technology*, 2020.

[35] G. Gutoski and D. Stebila, "Hierarchical deterministic bitcoin wallets that tolerate key leakage," in *Int'l. Conf. Financial Cryptography and Data Security*, 2015.

[36] A. Di Luzio, D. Francati, and G. Ateniese, "Arcula: A secure hierarchical deterministic wallet for multi-asset blockchains," in *Int'l. Confer. Cryptology and Network Security*, 2020.

[37] E. Lundberg and D. Nilsson, "Webauthn recovery extension," 2019, yubico. [Online]. Available: https://github.com/Yubico/webauthn-recovery-extension

[38] N. Frymann, D. Gardham, F. Kiefer, E. Lundberg, M. Manulis, and D. Nilsson, "Asynchronous remote key generation: an analysis of yubico's proposal for w3c webauthn," in *Proc. ACM CCS*, 2020.

[39] N. Frymann, D. Gardham, and M. Manulis, "Unlinkable delegation of webauthn credentials," in *European Symp. Research in Computer Security*, 2022.

[40] L. Hanzlik, J. Loss, and B. Wagner, "Token meets wallet: Formalizing privacy and revocation for fido2," in *IEEE Symp. Security and Privacy*, 2023.

[41] N. Frymann, D. Gardham, M. Manulis, and H. Nartz, "Generalised asynchronous remote key generation for pairing-based cryptosystems," in *Int'l. Conf. Applied Cryptography and Network Security*, 2023.

[42] S. Das, Z. Xiang, L. Kokoris-Kogias, and L. Ren, "Practical asynchronous high-threshold distributed key generation and distributed polynomial sampling," in *USENIX Security Symp.*, 2023.

[43] M. Manulis and H. Nartz, "Distributed asynchronous remote key generation," in *Int'l. Conf. Applied Cryptography and Network Security*, 2025.

[44] J. Brendel, S. Clermont, and M. Fischlin, "Post-quantum asynchronous remote key generation for fido2," in *Int'l. Conf. Theory and Application of Cryptology and Information Security*, 2024.