Pseudorandom pseudonym implicit certificates for network-efficient V2X communications

Mattia Trabucco, Luca Ferretti
University of Modena and Reggio Emilia
{mattia.trabucco, luca.ferretti}@unimore.it

***Abstract*—In standard Vehicle-to-Everything (V2X) communications, privacy against tracking is guaranteed by provisioning each vehicle with multiple *conditionally unlinkable pseudonym certificates*. While these certificates are bound to independent keys, they contain pseudorandom indexing values called *linkage values* that allow authorities to efficiently link and revoke all certificates associated with a misbehaving vehicle, at the cost of increased network overhead of all V2X messages. The adoption of *additive key derivation functions* (AKD) for generating *conditionally unlinkable pseudorandom asymmetric key pairs* has been recently proposed to remove linkage values while keeping revocation at constant costs, but only in the context of explicit certificates. We design a PseudoRandom Pseudonym Implicit Certificate (PRPIC) protocol based on AKD which represents the pseudonym-based authentication protocol with lowest network overhead and constant revocation costs. We demonstrate our claims through an analytical evaluation based on settings recommended by standards, comparing PRPIC with SCMS and with pseudorandom explicit certificates.**

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

I. Introduction

Vehicle-to-Everything (V2X) communications are essential to Intelligent Transportation Systems (ITS), yet their design remains challenging due to the need of balancing security and privacy with the strict latency requirements of safety-related applications, resource constraints of vehicular networks, and economic costs of vehicles and infrastructures [1].

From a privacy perspective, vehicles need *conditional anonymity* (also called *conditional unlinkability*) to prevent tracking by malicious parties, as long as they behave honestly. However, if a vehicle behaves maliciously, such as by sending false information due to sensor faults or to a cyber attack, it should be possible to efficiently revoke it from the network, thus invalidating its privacy guarantees. Since complete anonymity is not affordable in this context, the accepted trade-off is to adopt certificates based on pseudonyms distributed by a trusted Vehicular Public Key Infrastructure (VPKI) [2]: vehicles are provisioned with a set of pseudonym certificates issued by a Certificate Authority (CA) which are bound to the vehicle and the CA public keys (either explicitly or implicitly), and which may include additional metadata (e.g., the validity period). As opposed to the Web PKI, certificates issued by the VPKI are considered pseudonyms of the vehicle because they omit all user-identifiable information. 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 key. Managing the large volume of pseudonym certificates per vehicle remains a fundamental challenge, which requires existing VPKI architectures to balance the trade-off between minimizing V2X message sizes via compact certificates and managing the overhead of revocation mechanisms [3]–[7].

In this paper, we propose a novel *PseudoRandom Pseudonym Implicit Certificates* (PRPIC) protocol for generating *conditionally unlinkable implicit certificates*, achieving the smallest pseudonym certificates with constant revocation costs which fit vehicular networks bandwidth constraints. Our approach is based on *additive key derivation schemes* (AKD) [8] which are mostly known for deterministic wallets in the context of Bitcoin [9], but which have never been used in the context of implicit certificates. The adoption of AKD schemes in the context of a VPKI based on *explicit certificates* has been recently proposed to remove the need for linkage values while keeping revocation at constant costs [10], however the generated certificates are still larger than existing techniques based on implicit certificates.

The most popular approach for efficient revocation has been included within the US Security Credential Management System (SCMS) [11], where a pseudorandom indexing identifier called *linkage value* is added within each certificate exchanged over the network. Authorities can revoke all pseudonym certificates associated with a misbehaving vehicle at constant network costs, by just releasing the seed used to derive all pseudorandom linkage values. While the benefit is the distribution of small CRLs, the drawback is the need to send the linkage value within all V2X messages. Although such a value may seem quite small (9 Bytes in SCMS), in the context of V2X communications where network bandwidth is the major bottleneck it represents a significant overhead. Moreover, its size depends on the workload of the vehicular network and on the required privacy levels, including the number of vehicles that impact on the number of pseudonyms used within communications, and the number of pseudonyms assigned to each vehicle. Thus, future changes to such parameters may require difficult software and hardware upgrades for accommodating larger linkage values.

Other approaches [12], [13] for preventing tracking in V2X communications include pseudonym swapping techniques [14], [15], identity-based solutions [16], [17], and

decentralized key management mechanisms [4], [18], but have several drawbacks due to heavy cryptographic operations, overheads in distributing CRLs and/or requesting new pseudonyms, inability to revoke misbehaving vehicles, scaling issues, or requirements for frequent communication with infrastructure nodes.

The difficulty of designing pseudorandom implicit certificates is related to the need of complying with mathematical structures of derivation schemes and of implicit certificates themselves, without introducing vulnerabilities. In particular, the owner of the certificate should obtain a set of pseudorandom key pairs which have been jointly computed with the CA, such that the CA is not able to compute the secret key. We propose such a design by integrating ECQV [19] implicit certificates with non-hardened BIP32 [9] asymmetric key derivation schemes and show that authenticated messages save 66 bits for each V2X message exchanged over the network considering workloads estimated by SCMS, which increases if considering stronger privacy requirements and future higher workloads.

The proposed protocol has two main limitations. First, it does not support a decentralized governance where CAs are separated from linkage authorities as in SCMS. We leave supporting multi-authority architectures as future work. Second, the size of CA public keys is linear in the number of issued pseudonym certificates. However, we show that these costs are in the order of tens of kilobytes when considering vehicular networks workloads and requirements, and thus vehicles should be able to store them without issues. Computational costs for key management operations are also higher but still practical: the most expensive operations are performed by the CA, and there is no additional overhead for everyday message with regard to standards based on implicit certificates.

Section II provides notation and base knowledge. Section III describes system and threat models. Section IV describes the details of the proposed protocol. Section V discusses asymptotic and concrete costs. Section VI concludes the paper.

II. Notation and Base knowledge

A. Notation

Let $\mathbb{G}$ be an additive cyclic group of prime order q built over an Elliptic Curve where the Elliptic-Curve Discrete Logarithm Problem is hard with regard to security parameter λ, and $B \in \mathbb{G}$ be the generator of $\mathbb{G}$. We use $+$ and $\cdot$ both for operations on field $\mathbb{Z}_q$ and on group $\mathbb{G}$. Thus, $X+Y$ denotes point addition for any $X, Y \in \mathbb{G}$, $s \cdot X$ point scalar multiplication for any $\langle s, X\rangle \in \mathbb{Z}_q \times \mathbb{G}$, and $a+b$ and $a \cdot b$ denote integer addition and multiplication modulo q for any $a, b \in \mathbb{Z}_q$. We denote as $[a, b]$ the ordered set of integers $[a, a+1, \ldots, b]$ and shorten $[1, a]$ as $[a]$.

B. Implicit Certificates

Implicit certificates allow a receiver to compute the public key for verifying a digital signature by using the public key of the certification authority, reducing the size of exchanged

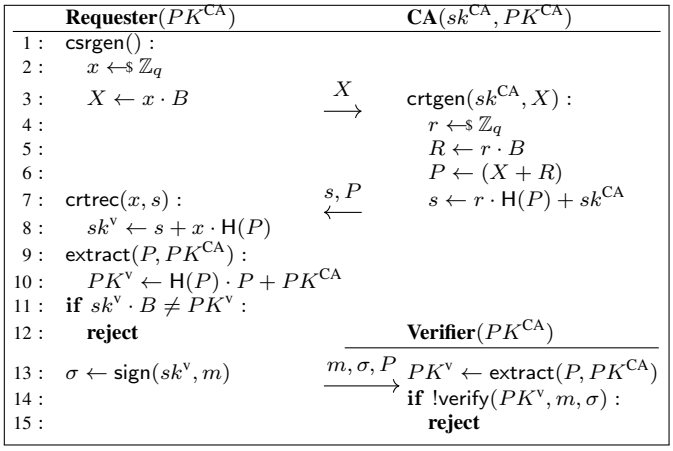

Fig. 1: ECQV implicit certificates information flow.

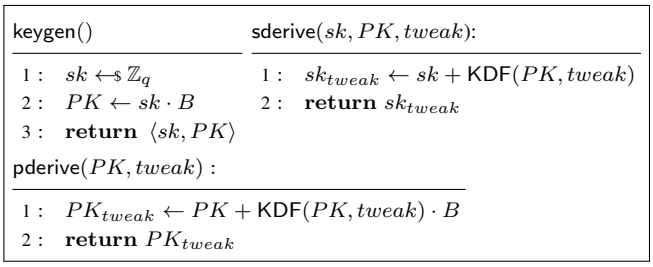

Fig. 2: Non-hardened BIP32 AKD specification using abstract algebra.

cryptographic material with regard to traditional explicit certificates. We consider the Elliptic Curve Qu-Vanstone (ECQV) scheme [19] adopted by vehicular communication standards instantiated with the NIST p256 ECC curve, where each implicit certificate is 32-Byte long and which allow to save the 64-Byte digital signature generated by the CA included in explicit certificates. ECQV includes four routines detailed in Figure 1: *certificate sign request (CSR) generation* (csrgen) generates the *requester secret key contribution* x and the related public contribution X; *certificate generation* (crtgen) generates the *CA secret contribution* s and *implicit certificate* P; *certificate reception* (certrec) computes the "actual" secret key sk^{v}; *public key extraction* (extract) computes the related public key PK^{v}. Let $\sigma \leftarrow \mathsf{sign}(sk, m)$ create digital signature σ on message m and secret key sk, and $\{\mathsf{accept}, \mathsf{reject}\} \leftarrow \mathsf{verify}(PK, m, \sigma)$ verify σ and m with public key PK. Key sk^{v} can be directly used to sign messages, while verification requires first to compute PK^{v} via extract.

C. Additive Key Derivation (AKD)

An AKD scheme is composed of three routines: *key generation* (keygen), for generating a new key pair which acts as the root of the key derivation hierarchy; *public key derivation* (pderive) and *secret key derivation* (sderive), for computing derived public and secret keys, respectively, with regard to some tweak $tweak$, which acts as the scope of the key derivation as for well-known symmetric key derivation schemes (e.g., HKDF [20]). In Figure 2 we show a generalized version of the three routines as specified by BIP32 [9] for non-hardened addresses, where $\mathsf{KDF} : \{0,1\}^{*} \times$

$\{0,1\}^* \to \mathbb{Z}_q$ denotes a key derivation function for symmetric keys [21]. Subsequently calling pderive and sderive allows to build a hierarchical tree of derived key pairs from a master key pair $\langle sk_0, PK_0 \rangle$. Let $\texttt{mdl}$ be some dedicated metadata for scoping a hierarchy unambiguously and $PK_i = \mathsf{pderive}(\mathsf{pderive}(\ldots \mathsf{pderive}(PK_0, \texttt{mdl})))$ for i chained calls of pderive, then $PK_i = PK_0 + (\sum_{j=0}^{i-1} \mathsf{KDF}(PK_j, \texttt{mdl})) \cdot B$, and similarly $sk_i = sk_0 + (\sum_{j=0}^{i-1} \mathsf{KDF}(PK_j, \texttt{mdl}))$.

III. System and threat models

As in [10], 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 vehicular network. The CA divides time into time periods, each identified by an index $t = 0, 1, 2, \ldots$, and each vehicle can obtain a fixed maximum number $\texttt{N}$ of pseudonym certificates for each time period starting from the *current* time period τ.

Capturing design choices of the SCMS standard [11], the CA may define a parameter $\texttt{T} \in \mathbb{N}$, such that a vehicle can request pseudonym certificates for up to $(\texttt{T} - 1)$ future time periods. Thus, a vehicle can obtain a maximum of $\texttt{N} \cdot \texttt{T}$ pseudonym certificates. Upon receiving a pseudonym certificate associated with a vehicle, a CA may revoke all the related pseudonym certificates for the current and following time periods by distributing *revocation material* (rm), that vehicles use to build and maintain a *certificate revocation list* (crl). A vehicle may need to update its crl for each new time period, to discard or update records.

Persistent storage of vehicles includes four types of data:

- *current cryptographic material* (ccm) includes secret keys and certificates used within the current time period;
- *certificate revocation list* (crl) includes pseudonym certificates that are revoked within the current time period;
- *certificate refresh material* (crm) includes data needed to update ccm for the next time period;
- *revocation refresh material* (rrm) includes data needed to update crl for the next time period.

We consider passive and active network adversaries on communications among vehicles, while for ease of presentation we assume trusted communications between vehicles and the CA. We also assume that CA public key material is known by all vehicles, and that CA and vehicles share a loosely synchronized clock and agree on the current time period τ. As typical when using implicit certificates, we also assume/recommend that the adopted signature scheme is resistant to *key substitution attacks* [22].

IV. PRPIC: Pseudorandom pseudonym implicit certificates specification

PRPIC includes four sub-protocols: *pseudonym certificate release*, where a vehicle requests and obtains a set of pseudonym certificates from the CA (Section IV-A); *revocation*, where a CA receives a notification about a vehicle misbehavior and revokes all the pseudonym certificates associated with the vehicle (Section IV-B); *refresh*, where a vehicle updates its certificates and revocation lists for the following time period (Section IV-C); *message authentication and verification*, where vehicles exchange messages within the network (Section IV-D).

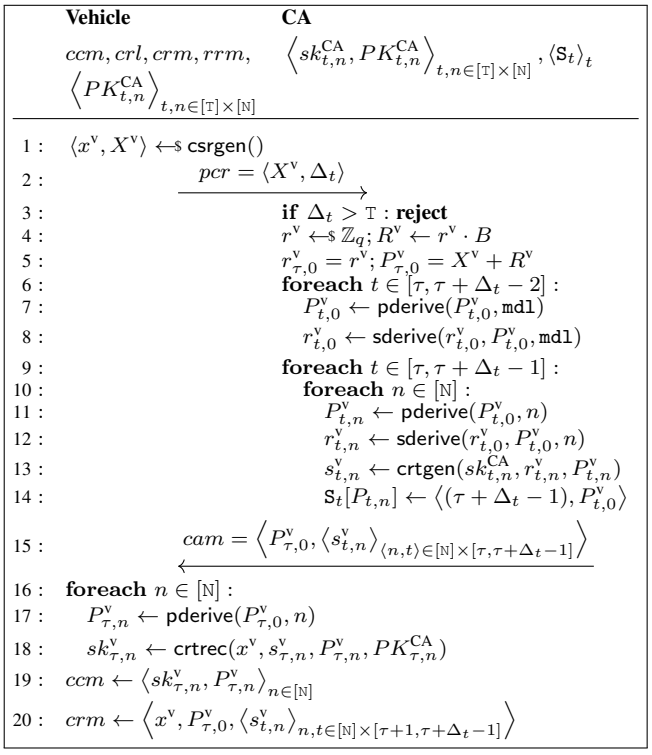

Fig. 3: Specification of *pseudonym certificates release*

A. Pseudonym certificates release

Details are in Figure 3. For ease of presentation and without loss of generality, we omit application-related metadata and procedures (e.g., verification of the vehicle identity), and we assume that, at the beginning of the protocol, the CA owns key pairs $\left\langle sk^{\text{CA}}_{t,n}, PK^{\text{CA}}_{t,n} \right\rangle_{t,n \in [\texttt{T}] \times [\texttt{N}]}$, that ccm, crl, crm and rrm are empty. Also, we write $sk^{\text{CA}}_{t,n}$ and $PK^{\text{CA}}_{t,n}$ instead of $sk^{\text{CA}}_{(t \bmod \texttt{T}),n}$ and $PK^{\text{CA}}_{(t \bmod \texttt{T}),n}$. The important design choice is deriving pseudorandom secret nonces $\left\langle r^{\text{v}}_{t,i} \right\rangle_{t,i}$ from a single fresh nonce r^{v}, and using implicit certificates $\left\langle P^{\text{v}}_{t,0} \right\rangle_t$ for derivation instead of related nonce commitments $\left\langle R^{\text{v}}_{t,0} \right\rangle_t$.

Correctness. For all $t, n \in [\texttt{T}] \times [\texttt{N}]$, the CA computes:

$$P^{\text{v}}_{t,0} = P^{\text{v}}_{\tau,0} + \left(\sum_{i=\tau}^{t-1} \mathsf{KDF}(P^{\text{v}}_{i,0}, \texttt{mdl}) \right) \cdot B$$

$$P^{\text{v}}_{t,n} = P^{\text{v}}_{\tau,0} + \left(\sum_{i=\tau}^{t-1} \mathsf{KDF}(P^{\text{v}}_{i,0}, \texttt{mdl}) + \mathsf{KDF}(P^{\text{v}}_{t,0}, n) \right) \cdot B$$

$$r^{\text{v}}_{t,0} = r^{\text{v}} + \sum_{i=\tau}^{t-1} \mathsf{KDF}(P^{\text{v}}_{i,0}, \texttt{mdl})$$

$$r^{\text{v}}_{t,n} = r^{\text{v}} + \sum_{i=\tau}^{t-1} \mathsf{KDF}(P^{\text{v}}_{i,0}, \texttt{mdl}) + \mathsf{KDF}(P^{\text{v}}_{t,0}, n)$$

$$s^{\text{v}}_{t,n} = r^{\text{v}}_{t,n} \cdot \mathsf{H}(P^{\text{v}}_{t,n}) + sk^{\text{CA}}_{t,n}$$

CA $(\langle \mathtt{S}_t \rangle_{t\in[(\tau-\mathtt{T}+1),\tau]})$ | **Vehicle** (ccm, crl)

$P^{\mathrm{v}}_{t',n'} \leftarrow crt^{\mathrm{v}}_{t'}$
foreach $t \in [(\tau - \mathtt{T} + 1), \tau]$:
  if $P^{\mathrm{v}}_{t',n'} \in \mathtt{S}_t$: / Check if certificate is related to time period t
    $\left\langle t_{na}, P^{\mathrm{v}}_{t',0} \right\rangle \leftarrow \mathtt{S}_t[P^{\mathrm{v}}_{t',n'}]$
    break
else : / Enter if previous loop did not exit with `break`
  abort
if $t_{na} < \tau$: / Vehicle cannot derive secret keys anymore
  abort
foreach $t \in [t', (\tau - 1)]$:
  $P^{\mathrm{v}}_{(t+1),0} \leftarrow \mathsf{pderive}(P^{\mathrm{v}}_{t,0}, \mathtt{mdl})$
  $rm = \left\langle t_{na}, P^{\mathrm{v}}_{\tau,0} \right\rangle$ ⟶
  foreach $n \in [\mathtt{N}]$:
    $P^{\mathrm{v}}_{\tau,n} \leftarrow \mathsf{pderive}(P^{\mathrm{v}}_{\tau,0}, n)$
    $crl.\mathsf{add}(P^{\mathrm{v}}_{\tau,n})$
  $rrm.\mathsf{add}(\langle t_{na}, P^{\mathrm{v}}_{\tau,0} \rangle)$

Fig. 4: Specification of *pseudonym certificates revocation*.

For all $t, n \in [\mathtt{T}] \times [\mathtt{N}]$, the vehicle computes:

$$sk^{\mathrm{v}}_{t,n} = s^{\mathrm{v}}_{t,n} + x^{\mathrm{v}} \cdot \mathsf{H}(P^{\mathrm{v}}_{t,n}) = (x^{\mathrm{v}} + r^{\mathrm{v}}_{t,n}) \cdot \mathsf{H}(P^{\mathrm{v}}_{t,n}) + sk^{\mathrm{CA}}_{t,n}$$

Since $P^{\mathrm{v}}_{\tau,0} = (X^{\mathrm{v}} + R^{\mathrm{v}})$, then $P^{\mathrm{v}}_{t,n} = (x^{\mathrm{v}} + r^{\mathrm{v}}_{t,n}) \cdot B$ also holds, and thus $PK^{\mathrm{v}}_{t,n} = sk^{\mathrm{v}}_{t,n} \cdot B = \mathsf{extract}(P^{\mathrm{v}}_{t,n}, PK^{\mathrm{CA}}_{t,n})$.

Security (sketch). For each certificate release procedure, a vehicle receiving $\left\langle s^{\mathrm{v}}_{t,n} \right\rangle_{t,n}$ can build a system of $\mathtt{T} \cdot \mathtt{N}$ linear equations dependent on the same secret r^{v}, instead of $\mathtt{T} \cdot \mathtt{N}$ fresh secrets as in standard ECQV. However, since the CA uses $\mathtt{T} \cdot \mathtt{N}$ independent keys $\left\langle sk^{\mathrm{CA}}_{t,n} \right\rangle_{t,n}$, the system still depends on $(\mathtt{T} \cdot \mathtt{N} + 1)$ random secrets as in ECQV, and thus is indeterminate with uniformly distributed solutions, perfectly hiding r^{v} and $\left\langle sk^{\mathrm{CA}}_{t,n} \right\rangle_{t,n}$. For multiple certificate release procedures, the same CA secret key is only re-used with a fresh nonce r^{v}, even in presence of malicious vehicles possibly submitting the same x^{v}, as in standard ECQV.

Performance. For efficiency at revocation time, for each time period t the CA maintains a reverse hash map $\mathtt{S}_t$ mapping each pseudonym implicit certificate $P^{\mathrm{v}}_{t,n}$ to the corresponding master implicit certificate $P^{\mathrm{v}}_{t,0}$ and the last time period $(\tau + \Delta_t - 1)$ for which the vehicle obtained pseudonym certificates. We also observe that the proposed protocol is meant to minimize network overhead, but other variants may be instantiated to obtain different trade-offs. As an example, the CA may send all implicit certificates $\left\langle P^{\mathrm{v}}_{t,n} \right\rangle_{t,n}$ so that the vehicle does not have to compute them at its side. Finally, note that the vehicle does not need to derive and store secret keys for following time periods because crm stores all the due data to derive them during *refresh* (see Section IV-C).

B. Pseudonym certificates revocation

Figure 4 shows details of the specification for *pseudonym certificates revocation*. Let $crt^{\mathrm{v}}_{t'}$ be a certificate associated to a misbehaving vehicle at time period t' and sent to the CA by some authority for revocation. If the certificate is not found within $\mathtt{S}_t, \forall t$, then the CA simply aborts because it means

Vehicle (crm, rrm)

.....Refresh of ccm.....
$\left\langle x^{\mathrm{v}}, P^{\mathrm{v}}, \langle s^{\mathrm{v}}_{t,n} \rangle_{n,t\in[\mathtt{N}]\times[\tau+1,\tau+\Delta_t-1]} \right\rangle \leftarrow crm$
$P^{\mathrm{v}}_{(\tau+1),0} \leftarrow \mathsf{pderive}(P^{\mathrm{v}}, \mathtt{mdl})$
foreach $n \in [\mathtt{N}]$:
  $P^{\mathrm{v}}_{(\tau+1),n} \leftarrow \mathsf{pderive}(P^{\mathrm{v}}_{(\tau+1),0}, n)$
  $sk^{\mathrm{v}}_{(\tau+1),n} \leftarrow \mathsf{crtrec}(x^{\mathrm{v}}, s^{\mathrm{v}}_{(\tau+1),n}, P^{\mathrm{v}}_{(\tau+1),n}, PK^{\mathrm{CA}}_{(\tau+1),n})$
$ccm \leftarrow \left\langle sk^{\mathrm{v}}_{(\tau+1),n}, P^{\mathrm{v}}_{(\tau+1),n} \right\rangle_{n\in[\mathtt{N}]}$
$crm \leftarrow \left\langle x^{\mathrm{v}}, P^{\mathrm{v}}_{(\tau+1),0}, \langle s^{\mathrm{v}}_{t,n} \rangle_{n,t\in[\mathtt{N}]\times[\tau+2,\tau+\Delta_t-1]} \right\rangle$
.....Refresh of crl.....
$crl \leftarrow \emptyset$ / Deleting crl of previous time period
$rrm' \leftarrow \emptyset$ / Temporary data structure
foreach $\langle t_{na}, P^{\mathrm{v}}_{\tau,0} \rangle \in rrm.\mathsf{pop}()$:
  if $t_{na} \leq \tau$:
    continue
  $P^{\mathrm{v}}_{(\tau+1),0} \leftarrow \mathsf{pderive}(P^{\mathrm{v}}_{(\tau),0}, \mathtt{mdl})$
  $rrm'.\mathsf{add}(\left\langle t_{na}, P^{\mathrm{v}}_{(\tau+1),0} \right\rangle)$
  foreach $n \in [\mathtt{N}]$:
    $P^{\mathrm{v}}_{(\tau+1),n} \leftarrow \mathsf{pderive}(P^{\mathrm{v}}_{(\tau+1),0}, n)$
    $crl.\mathsf{add}(P^{\mathrm{v}}_{(\tau+1),n})$
$rrm \leftarrow rrm'$

Fig. 5: Specification of *refresh* from time period τ to $(\tau + 1)$.

that it is associated with a time period older than $\tau - \mathtt{T}$, thus revocation is no longer needed. The same holds if the certificate is found but it is associated with an expire time $t_{na} < \tau$. Instead, if $t_{na} > \tau$, the CA derives the master certificate for the current time period and sends it as *revocation material* rm to the vehicle, else if $t_{na} = \tau$ the first retrieved certificate is directly sent without any derivation. The vehicle then derives all the pseudonym implicit certificates associated with the same vehicle within the current time period, and stores the received master certificate within rrm for future *refresh*.

C. Certificate refresh

Figure 5 shows details of the *refresh* protocol. The refresh of ccm is executed only if, at the end of τ, the vehicle has at least one set of $\langle s_{t,n} \rangle$ for $t = \tau + 1$ stored in crm. Whether the vehicle has refreshed ccm from previous data stored in crm or from a new invocation of the pseudonym certificate release protocol, it must also update crl for the subsequent time period using the revoked master certificates stored in rrm. If the master certificate is associated with an expire time $t_{na} \leq \tau$, then the vehicle skips the derivation because it means that it is associated with a time period older than $\tau - \mathtt{T}$, thus refresh is no longer needed.

D. V2X message authentication and verification

We show protocol details in Figure 6. Given current time period τ and a certain index $i \in [\mathtt{N}]$ which is previously selected with some logic that is orthogonal to our protocol, a *sender* vehicle v can send a message m by signing it as $\sigma \leftarrow \mathsf{sign}(sk^{\mathrm{v}}_{\tau,i}, \langle \mathsf{m}, i, PK^{\mathrm{CA}}_{\tau,i} \rangle)$. Then, v sends $\langle \mathsf{m}, \sigma, P^{\mathrm{v}}_{\tau,i}, i \rangle$.

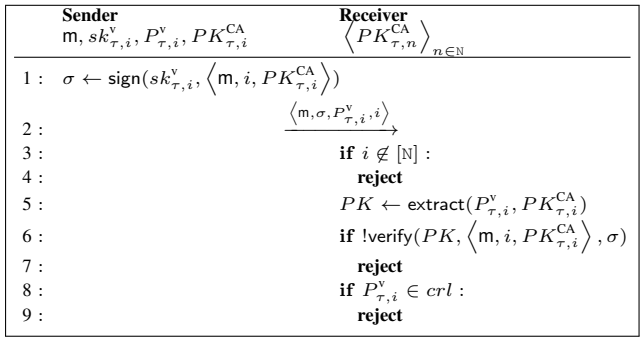

Fig. 6: Message authentication and verification in τ.

The receiver first checks the validity of i. As we assume a loosely shared clock among participants of the vehicular network (Section III), the receiver already shares the same current period τ, and uses index i explicitly sent over the network to select public key $PK^{\mathrm{CA}}_{\tau,i}$. Then, it proceeds with key extraction, signature verification, and validation of the certificate revocation status.

Note that signing $PK^{\mathrm{CA}}_{\tau,i}$ together with m and i is meant to prevent attacks aiming at making the receiver correctly verify the signed message by using a different CA key for extraction, and is analogous to recommendations related to preventing key substitution attacks in other more traditional scenarios [22]. Instead, we do not explicitly sign the vehicle public key because we assume that the sender already uses a signature scheme resistant to key substitution attacks. If this assumption does not hold, then the sender should also derive its own public keys during *refresh* and include $PK^{\mathrm{v}}_{\tau,i}$ within the signed message, thus requiring additional storage at the vehicle side[1].

V. Cost evaluation

We discuss the PRPIC protocol costs asymptotically (Section V-A) and concretely when instantiated with the NIST p256 curve, comparing them with current standards and with related literature (Section V-B).

A. Asymptotic analysis

Table I shows network and computation costs for *release*, *revocation* and *refresh*. We consider the worst case for a single execution of the protocol where $\Delta_t = \mathtt{T}$. Moreover, we evaluate dominant computational costs by considering the most expensive underlying cryptographic operations, which in PRPIC are represented by point scalar multiplications with a fixed point (csrgen, crtgen, pderive) and, even more expensive, with a variable point (extract). Other types of operations are considered negligible.

Release. CA computation costs are dominated by $\mathtt{N} \cdot \mathtt{T}$ pderive and crtgen routines, while for the vehicle they are dominated by one csrgen and $\mathtt{N}$ pderive[2]. Vehicle costs may be reduced to the computation of $\mathtt{N}$ crtrec if the CA sends all pseudonym certificates, however network costs would almost double. Network is dominated by the exchange of a secret contribution value $s^{\mathrm{v}}_{t,n}$ for each $t, n \in [\mathtt{T}] \times [\mathtt{N}]$.

Revocation. Computation costs of the CA depend on the presented certificate *age*, because the CA must derive the master implicit certificate obtained from its hash map S_t for the current time period τ using pderive. Considering a worst case scenario, its computation cost is $O(\mathtt{T})$. The vehicle computes $\mathtt{N}$ pderive for each revocation. Network costs are constant for each newly revoked vehicle, because the CA sends only a single master implicit certificate independently of $\mathtt{N}$ and $\mathtt{T}$.

Refresh. The vehicle performs $\mathtt{N}$ pderive to refresh ccm and $\mathtt{N} \cdot \mathtt{T}$ pderive for each revoked vehicle to refresh crl. Network costs are absent because the procedure is executed locally.

Phase	Entity	Computation		Network
Release	CA	$O(\mathtt{N} \cdot \mathtt{T})$	pderive +crtgen	$O(\mathtt{N} \cdot \mathtt{T})$
	Vehicle	$O(\mathtt{N})$	pderive	
Revocation	CA	$O(\mathtt{T})$	pderive	$O(1)$
	Vehicle	$O(\mathtt{N})$	pderive	
Refresh	Vehicle	$O(\mathtt{N} + \mathrm{rev} \cdot \mathtt{N} \cdot \mathtt{T})$	pderive	–

TABLE I: Asymptotic costs for computation and network

Type of data	Storage	Storage cost
Pseudonym certs and private keys valid in τ	ccm	$O(\mathtt{N})$
Certificates revoked by the CA in τ	crl	$O(\mathtt{N} \cdot \mathtt{T})$
Certificate refresh material	crm	$O(\mathtt{N} \cdot \mathtt{T})$
Revocation refresh material	rrm	$O(\mathtt{T})$
CA public key size	PK^{CA}	$O(\mathtt{N} \cdot \mathtt{T})$

TABLE II: Asymptotic costs for storage

Table II shows vehicle storage costs for maintaining ccm, crl, crm, rrm and the set of CA public keys for certificate verification (PK^{CA}). In each time period t, ccm includes all private keys and pseudonym certificates $\left\langle sk^{\mathrm{v}}_{t,n}, P^{\mathrm{v}}_{t,n} \right\rangle_n$ for each $n \in [\mathtt{N}]$, thus storage cost is $O(\mathtt{N})$. crl includes $\mathtt{N}$ pseudonym certificates for each revoked vehicle, but considering that pseudonym certificates can be refreshed for up to $\mathtt{T}$ time periods, we estimate storage costs to be $O(\mathtt{N} \cdot \mathtt{T})$ for each revoked vehicle. crm includes a CA private contribution $s^{\mathrm{v}}_{t,n}$ for each $t, n \in [\mathtt{T}] \times [\mathtt{N}]$, thus its storage cost is also $O(\mathtt{N} \cdot \mathtt{T})$. rrm includes a master implicit certificate for each revoked vehicle, which in the worst case can be valid for up to $\mathtt{T}$ time periods, thus storage cost is $O(\mathtt{T})$. Finally, PK^{CA} includes the public keys used to verify all the pseudonym certificates issued by the CA (see Section IV-D), thus storage cost is $O(\mathtt{N} \cdot \mathtt{T})$.

B. Concrete evaluation and comparison

We analytically evaluate storage and network costs for the NIST p256 elliptic curve [23] recommended by current standards [24], [25], and compare them with SCMS both based on implicit and on explicit certificates [11], and with

[1]Note that while public key extraction may also be performed at runtime during message authentication, we consider it to introduce too much computational overhead due to expensive cryptographic operations.

[2]In our specification, we assume that the vehicle trusts the CA and the communication channel is authenticated, and thus it does not execute extract, however if this operation is also executed then vehicle costs would almost double.

		Network overhead		Storage overhead				
		V2X message relative size	rm	ccm	crl	crm	rrm	PK^{CA}
SCMS [11]	**implicit certificate**	72-bit linkage value	320 bits	2.9 KB	3.6 MB	131 KB	**2 KB**	**32 B**
	explicit certificate	72-bit linkage value + 256-bit PK^{CA}	320 bits	4.2 KB	3.6 MB	195 KB	2 KB	32 B
This paper: PRPIC (***pseudorandom* pseudonym implicit certificate**)		**6-bit local index**	**256 bits**	**2.6 KB**	12.8 MB*	**65 KB**	8 KB	66560 B
Pseudorandom explicit pseud. cert. [10]		256-bit PK^{CA} (no index)	256 bits	2.6 KB	12.8 MB*	65 KB	8 KB	32 B

TABLE III: Network and storage overheads for NIST p256 elliptic curve protocol instantiation by considering settings and workload proposed within SCMS [11]. *V2X message relative size* shows only information related to data which differ in the different approaches (e.g., does not show digital signature size). rm shows the size of a single *revocation material* required to revoke all the pseudonym certificates of a misbehaving vehicle. In crl, the asterisk symbol ($*$) reminds that PRPIC and [10] sizes may be reduced to those of SCMS at the cost of introducing the same false positives rate. crm shows the maximum amount of data that must be stored for refresh, which is that stored just after completing pseudonym certificates release.

pseudorandom explicit certificates [10]. Table III shows the resulting costs considering the following parameters. Size of both scalars and compressed elliptic curve points for NIST p256 curve is 32 Bytes. We size other parameters according to discussions proposed by SCMS for sizing their linkage value [11, Section V.C], that is, $\mathtt{N} = 40$ pseudonym certificates associated with each time period and, considering 2.5×10^8 vehicles, about $2.5\times10^8\times40 = 10^{10}$ total pseudonym certificates in use within each time period, with a revocation rate assumed below 1%. The certificate preload time is one year and each time period is one week long, thus $\mathtt{T} = 52$. Finally, SCMS recommends to plan support for 10000 revocation entries.

V2X message size. The second column of the table (*V2X message relative size*) shows sizes of data which differ among the considered approaches. SCMS [11] relies on *global* indexing identifiers called *linkage values*, which are included in each pseudonym certificate (similar to the Serial Number in X.509 certificates [26]). Linkage values *globally* identify pseudonym certificates among all vehicles, except for a certain probability of collision which depends on its size. SCMS uses 72-bit linkage values to get a probability of collision (i.e. of re-assigning the same linkage value to more vehicles) below 10^{-6}, which is estimated considering the birthday paradox. If we consider a negligible probability related to cryptographic standards, that is $2^{-32} \approx 10^{-10}$, a size of 85 bits would be preferable [10]. In PRPIC, authenticated messages include opaque indexing value $i \in [\mathtt{N}]$ which is used by receivers to select the due CA public key (Section IV-D), which is not required to be unique among vehicles and time periods and thus can be sized as $\lceil \log_2(\mathtt{N}) \rceil$. Since we consider $\mathtt{N} = 40$, then size of i is 6 bits. In addition to the information included in the table, strictly related to the parameters proposed by SCMS, we observe that our index is also more scalable than SCMS linkage value, because it scales linearly with regard to the number of pseudonym certificates, which is much better than the quadratic increase of global linkage values (again, due to the birthday paradox). As an example, if $\mathtt{N} = 4096$, then size of i is 12 bits and that of the linkage value is 92 or 105 bits respectively for 10^{-6} and 2^{-32} collision probability with 2.5×10^8 vehicles, resulting in overhead gain increasing from $66/79$ bits to $80/93$ bits for each message. However, it must be reminded that PRPIC needs to maintain a CA public key that also scales linearly to $\mathtt{N}$ and $\mathtt{T}$, and for $\mathtt{N} = 4096$ and $\mathtt{T} = 52$ key size would be ≈ 6.8MB. We observe that using pseudorandom explicit certificates [10] do not need to include any index because there is just a single CA public key to verify all messages and benefits of being completely independent of the network workload and settings, however they incur in the high overhead of sending the explicit digital signature of the CA within the certificate, thus achieving worse overall network overhead. Deploying SCMS with explicit certificates has downsides of both having a linkage value and the explicit digital signature, and thus achieves worse network overhead.

Size of a CRL sent for revocation. The third column of the table shows sizes of *revocation material* (rm) sent by the CA for revoking a vehicle from the network. In SCMS, the revocation of a vehicle is performed by distributing seeds and metadata that allows vehicles to reconstruct all pseudorandom linkage values associated with the revoked pseudonyms. Two authorities (Linkage Authorities) are involved in the revocation process: the CRL includes two linkage value seeds (i.e., hash chains) and the identities of the two authorities. The total size of the revocation material is $2 \times 128 + 2 \times 32 = 320$ bits, where 128 bits are the linkage seeds created by the authorities and 32 bits are identity strings associated with the authorities.

Current cryptographic material ccm includes signing keys and pseudonym certificates for the current time period in all approaches. Overall, sizes are not too different. More in detail, both SCMS approaches and PRPIC need to store $\mathtt{N}$ 32B secret keys, while [10] can store just one secret and derive the due pseudorandom keys at authentication time. All approaches need to store $\mathtt{N}$ certificates, however implicit certificates are smaller (as described above for *V2X message relative size*) and SCMS also require vehicles to maintain linkage values.

Current refresh material crm includes very different data depending on the approaches. All approaches must store $\mathtt{N} \times (\mathtt{T}-1)$ secret keys for all time periods (or, in PRPIC, all partial secret values $s^{\text{v}}_{t,n}$ provided by the CA plus the vehicle partial secret value x^{v}). However, while SCMS approaches must also store all $\mathtt{N}\times(\mathtt{T}-1)$ certificates and thus requires higher storage, PRPIC and [10], adopting pseudorandom keys, must only store one master key which is derived at each refresh.

Certificate revocation lists crl include the due material for rejecting an authenticated message that would be otherwise be accepted. The total size obviously depends on the number of revoked vehicles. In our analysis, we considered SCMS recommendations of providing storage for 10000 revoked vehicles. For each pseudonym of all revoked vehicles, SCMS approaches store a 9-Byte linkage value. Instead, PRPIC and [10] both store a 32-Byte EC point which acts as implicit certificate and public key, respectively. Note that PRPIC and [10] do not suffer from the false positives characterizing SCMS revocation mechanisms (as discussed above), however they can achieve the same storage overhead of SCMS at the cost of allowing the same false positives by storing a truncated 9-Byte hash of each EC point. Clearly, different trade-offs in terms of storage overhead and false positive rate can be obtained by choosing a different digest size.

Revocation refresh material rrm includes material to refresh crl for the next time period. All approaches use derivation techniques to save storage: SCMS includes data to derive linkage values; PRPIC and [10] store EC points which act as master implicit certificates and master public keys, respectively, similarly to crl.

CA public key material (PK^{CA}). SCMS approaches and [10] are conservative and only need one CA public key for verifying all messages. Instead, PRPIC requires vehicles to store $\mathtt{N} \times \mathtt{T}$ public keys. While the resulting cryptographic material is obviously much larger, for the settings recommended by SCMS it is comparable or smaller than other data that must still be stored within vehicles, as described before.

VI. Conclusions

We proposed a novel protocol for communications based on pseudonyms with low network overhead for authenticated messages and certificate management operations, including revocation with constant costs with regard to the number of issued pseudonyms. Our work achieves the lowest network overhead for any known authenticated communication system based on asymmetric cryptography and pseudonyms, and thus fits perfectly V2X communications, where network bandwidth represents the most critical bottleneck. Computational overhead for key management operations is affordable in the context of vehicular communications, and so is the major limitation of requiring CA public key size that is linear in the number of pseudonyms associated with each vehicle. Future work will try to overcome this limitation, but also to experimentally evaluate the protocol in real scenarios, to investigate variants involving different families of key derivation schemes, and to deploy the protocol in a decentralized architecture governed by multiple authorities.

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] 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.

[4] 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.

[5] 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. ITS*, 2020.

[6] 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 TVT*, 2022.

[7] 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.

[8] J. Groth and V. Shoup, "On the security of ECDSA with additive key derivation and presignatures," in *EUROCRYPT: 41st Int'l. Conf. Theory and Applications of Cryptographic Techniques*, 2022.

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

[10] M. Trabucco, G. Gambigliani Zoccoli, M. Marchetti, and L. Ferretti, "Network-efficient authenticated pseudonym-based v2x communications with constant revocation costs," in *IEEE Int'l. Symp. Network Computing Applications*, 2025.

[11] 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. ITS*, 2018.

[12] 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.

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

[14] 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.

[15] 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 TVT*, 2023.

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

[17] L. Zhang, "Otibaagka: A new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE TIFS*, 2017.

[18] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE TVT*, 2020.

[19] 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

[20] 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

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

[22] A. Menezes and N. Smart, "Security of signature schemes in a multi-user setting," *Design, Codes and Cryptography*, 2004.

[23] 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

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

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

[26] 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