# PUCA: A Pseudonym Scheme with User-Controlled Anonymity for Vehicular Ad-Hoc Networks (VANET)

David Förster
Robert Bosch GmbH
david.foerster@de.bosch.com

Frank Kargl
University of Ulm, Germany &
University of Twente, NL
frank.kargl@uni-ulm.de

Hans Löhr
Robert Bosch GmbH
hans.loehr@de.bosch.com

*Abstract*—**Envisioned vehicular ad-hoc networks (VANET) standards use pseudonym certificates to provide secure and privacy-friendly message authentication. Revocation of long-term credentials is required to remove participants from the system, e.g. in case of vehicle theft. However, the current approach to revocation puts the users' privacy at risk if the backend systems are not fully trusted.**

**We propose PUCA – a scheme that provides full anonymity, even against colluding backend providers, until the owner of a vehicle triggers revocation himself. The scheme uses anonymous credentials for authentication with the backend while leaving the communication among vehicles and with road side units unchanged and in compliance with existing standards.**

**With PUCA, we put drivers back in charge of their privacy while still allowing revocation of long-term credentials.**

## I. Introduction

Vehicular ad-hoc networks (VANET) have been extensively studied in research [19], [21] and several car manufactures are planning to include Car2X functionality in some of their models starting in 2015 [12]. The technology is expected to improve road safety as well as to deliver a more pleasant driving experience. To ensure rapid deployment user acceptance is a crucial success factor. Privacy concerns have been raised in the media repeatedly and need to be addressed.

A core feature of the envisioned Car2X systems are cooperative awareness messages (CAM) sent via dedicated short-range radio communication (DSRC). Broadcast at a high frequency (1-10 Hz) these "beacon messages" contain information such as the vehicle's current GPS position, velocity, and direction. This information can be used by other vehicles for safety features such as Cooperative Collision Avoidance (CCA), as well as by traffic control infrastructure to implement traffic efficiency applications.

It is crucial that CAM messages can only be sent by legitimate vehicles and cannot be tampered with. Forged messages could endanger travelers' physical safety, e.g. by faking an imminent collision and provoking an autonomous emergency braking. To ensure only authorized parties can participate in the network, all messages are signed cryptographically. Unfortunately, this threatens the users' privacy as the signing keys are unique identifiers that expose them to tracking attacks by anybody who receives their messages (no matter if the receiver is a legitimate participant of the Car2X network or not). Tracking users' movements by their messages, an attacker could infer frequently visited locations such as work place and residence as well as personal preferences.[1]

In order to protect the users' privacy, a scheme employing changing "pseudonym certificates" has been proposed [21] and is included in the recent standards of the ETSI Technical Committee on ITS[2] for Europe and the IEEE 1609 working group[3] for the USA. Instead of using one fixed certificate per user, messages are signed using short-lived pseudonym certificates. These are changed periodically and ensure that a user can be tracked only until the next pseudonym change.

The users' privacy towards authorities can be protected by a separation of duties between the "Pseudonym CA" (PCA) and the "Long-term CA" (LTCA) as suggested by the CAR 2 CAR Communication Consortium (C2C-CC) [3]. If required, they can cooperate to resolve a user's identity from his pseudonyms and exclude him from the system. The privacy offered by this approach obviously depends on the authorities' correct behavior and can easily be subverted, e.g. by fraudulent operators. If regulations change, the user may be faced with unexpected use of his mobility data. In particular, the approach is insufficient in an environment where the government fails to adequately protect the rights of individuals. Beyond, car manufacturers in the US have expressed their favor of driver's anonymity over liability in order to protect themselves from lawsuits by drivers who's identity has been resolved [22].

In order to provide optimal privacy protection and prevent the problems stated above, we should aim for a system where privacy of vehicle owners has priority even over interest of other stakeholders like law enforcement. Nobody else but the owner should be able to identify a vehicle just based on recorded message signatures and pseudonyms. Still, there may be situations where a single vehicle should be removed from the Car2X network, one case being theft of the vehicle. With the consent of the legitimate vehicle owner, it should be possible to mark a vehicle as revoked or stolen, so that it can no longer participate in communication.

[1] Tracking based on Car2X communication. Other means of tracking users, e.g. based on their cell phones, are not in the scope of this work.

[2] http://www.etsi.org/index.php/technologies-clusters/technologies/intelligent-transport

[3] http://standards.ieee.org/develop/wg/1609 WG.html

978-1-4799-7660-7/14/$31.00 ©2014 IEEE

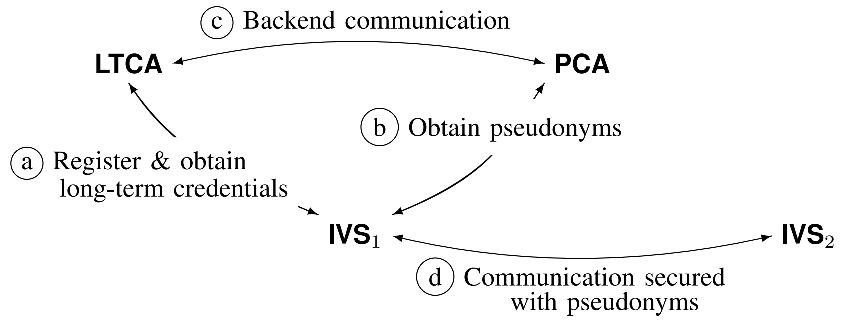


Figure 1. Interaction of IVS$_1$ with the LTCA, the PCA, and IVS$_2$ in our generic system model: IVS$_1$ is registered with the LTCA and obtains its long-term credentials ⓐ. Pseudonyms are obtained from the PCA ⓑ. The PCA may rely on the LTCA in order to validate the IVS's authentication ⓒ. The IVS then uses the pseudonyms to secure its communication with other participants ⓓ. (Communication between IVS$_2$ and the backend systems is omitted.)

*Our contribution:* In this paper we present PUCA[4], a pseudonym scheme where the user's privacy is protected by cryptographic methods instead of separation of responsibilities. When obtaining and using pseudonyms he remains fully anonymous. His identity can only be revealed if he either violates the protocol and attempts to obtain more pseudonyms than he is entitled to, or if he voluntarily chooses to revoke his membership in the Car2X network, e.g. because his vehicle will be sold or was stolen. In case of theft additional measures can be triggered such as having the stolen vehicle report its location. PUCA is built on top of the basic pseudonym scheme and only changes how pseudonyms are obtained, not how they are used. Hence it is fully compatible with the currently standardized approach and can be deployed alongside existing solutions.

To the best of our knowledge, we are the first to propose such a system that gives vehicle owner's privacy absolute priority while still enabling revocation with the owner's consent.

In the next section we present the high-level system model of a Car2X network. The requirements for our approach are laid out in section III. In section IV we describe other pseudonym schemes and related work. The building blocks for our scheme are introduced in section V. The PUCA scheme is presented in section VI. We close with an evaluation and discussion in section VII and provide a conclusion in section VIII.

## II. System model and scenario

We use the following system model of an Intelligent Transport System (ITS). Participating vehicles are equipped with an ITS Vehicle Station (IVS) that contains all the required Car2X components. Prior to deployment, an IVS is registered with the Long-term CA (LTCA) that keeps track of all participants within the ITS. The pseudonym CA (PCA) issues pseudonym certificates to the participants which they can then use to secure their communication among each other. The interactions within an ITS can be split into five different phases, which we will later refer to.

1) *Initialization:* Global system setup; this phase is only executed once when the ITS is established.
2) *Setup-Vehicle:* Add a new IVS to the ITS and provide it with a long-term authentication token ⓐ.
3) *Obtain-Pseudonyms:* Is executed by the IVS to refresh its supply of pseudonyms. It obtains pseudonyms from the PCA, authenticating with its long-term credential ⓑ. The PCA may rely on the LTCA to validate the authentication ⓒ.
4) *Communication:* Vehicles communicate among each other using the pseudonym certificates to authenticate their messages ⓓ.
5) *Revocation:* Remove an IVS from the system and invalidate its long-term authentication token.

Figure 1 depicts the entities' interaction in the *Setup-Vehicle*, *Obtain-Pseudonyms* and *Communication* phase.

## III. Requirements and constraints

We base our requirements on the general requirements for Car2X pseudonym schemes outlined by Schaub et al. [24]. In particular, we put emphasis on strong anonymity to counter threats to users' privacy posed by malicious backend providers.

R.1 *Authentication* – Provide a way to distinguish which parties are allowed to participate in the system.
R.2 *Restricted credential usage* – Only allow one active pseudonym at a time to prevent impersonation of multiple vehicles ("sybil attack").
R.3 *Revocation* – Revoke a user's right to participate in the system. This must only be possible with the user's agreement or in case he tries to cheat.
R.4 *Strong anonymity* – For an honest user, interactions with both the authorities as well as other participants must not be linkable to the user's identity. Even the authorities may only resolve a user's identity with his consent or in case he tries to cheat.
R.5 *Perfect forward privacy* – Revocation must not impact anonymity of previous messages.
R.6 *Real-time constraints* – Allow to sign up to 10 messages per second, validate up to several hundred messages per second in the communication phase.
R.7 *Scalability* – Must work with a large number of participating nodes, the interactions required should be minimal.

[4] Pseudonyms with User Controlled Anonymity; pronounced *pooka*, Irish for spirit/ghost

Schaub et al. also list accountability as a requirement. However, accountability cannot be achieved together with strong anonymity that enables drivers/vehicle owners to control when privacy is breached. We argue that law enforcement should resort to traditional investigation methods and not rely on Car2X pseudonym resolution.

## IV. Related work

In this section we first describe the basic pseudonym scheme. Then we present related efforts to employ advanced cryptography in vehicular ad-hoc networks and point out how our approach differs.

### *A. The basic pseudonym scheme*

In the following we describe the basic pseudonym scheme due to the Car 2 Car Communication Consortium (C2C-CC) [3], [11]. The scheme uses elliptic curve cryptography (ECC) for the benefit of short keys and signatures. A single root CA acts as the trust anchor. Its public key is securely provided to all participating parties prior to their deployment. We describe the five phases according to our system model (cf. section II).

*1) Initialization:* Setup the root CA; the root CA issues certificates to LTCA and PCA. LTCA and PCA may be replicated and operated by different vendors.
*2) Setup-Vehicle:* The LTCA issues a long-term certificate (LTC) to the new vehicle.
*3) Obtain-Pseudonym:* The IVS generates a number of pseudonym certificates and sends them to the PCA. It authenticates by providing an encrypted signature created with its LTC. The PCA forwards the encrypted signature to the LTCA, which decrypts and validates it. Upon confirmation from the LTCA the PCA signs the pseudonym certificates and sends them back to the requesting IVS. Furthermore the PCA stores both the vehicle's encrypted signature and the pseudonyms issued for reference in case of revocation. Optionally, pseudonym certificates may be valid for a certain time period $t$ only, in order to restrict the number of pseudonyms an IVS can use at the same time.
*4) Communication:* Vehicles use the pseudonym certificates to sign (and optionally encrypt) outgoing messages using the ECDSA signature algorithm [18].
*5) Revocation:* The scheme considers only revocation of long-term certificates. Pseudonyms obtained prior to revocation can be used until they expire. A pseudonym's owner can be identified as follows: The PCA looks up the encrypted signature that was used for authentication when the pseudonym was obtained. It sends it to the LTCA, which can decrypt the signature and revoke the corresponding long-term certificate. The PCA periodically updates its revocation list (CRL) to make sure revoked entities can no longer obtain new pseudonyms.

### *B. Advanced cryptography in Car2X*

Gañán et al. propose to use a one-way accumulator for checking pseudonym certificates for revocation during the communication phase [14]. Our approach, in contrast, only checks for revocation when obtaining new pseudonym certificates where performance is not critical. Furthermore we implement strong privacy towards backend systems while they focus on privacy among Car2X participants.

Singh [25] use CL signatures and anonymous credentials to authenticate Car2X messages and provides an implementation based on the `idemix` system [10]. Similarly, Huang [17] proposes CLIBA, a broadcast authentication scheme based on `idemix`. However, the performance analysis of both systems show that verification of messages is prohibitively inefficient.

Similarly, Guo et al. propose to use group signatures to authenticate Car2X message [16]. To deal with the inefficiency of group signatures they propose a "probabilistic verification of group signatures" where only a small fraction of incoming messages is validated. However, it is unclear whether this approach provides sufficient protection against forged messages.

Using anonymous credentials (or group signatures) between vehicles may lead to complete unlinkability between individual messages. This is undesirable as many application rely on a certain degree of linkability. For example, unlinkability of messages would make it much harder for cars to maintain their "local dynamic map", in which nearby vehicles and their trajectories are recorded, and which is the basis for functions such as for collision avoidance. Traffic efficiency applications, that want, e.g., to calculate traffic density may become right out impossible, if they are unable to count vehicles based on their pseudonyms.

A more promising approach (which we also follow) is to use anonymous credentials for authentication with the PCA when obtaining new pseudonyms. In this case performance requirements are much more relaxed. Furthermore, messages are linkable in between pseudonym changes, which ensures that safety applications are not affected.

Calandriello et al. present an approach for vehicles to generate pseudonym certificates for themselves using group signatures [4]. The scheme offers conditional anonymity which can be revoke by the LTCA which acts as group manager.

Schaub et al. created a scheme that enables a user to anonymously obtain pseudonym certificates through the use of intermediate "V-Tokens" using blind signatures [23]. The user's identity is encoded in the pseudonym certificates and can be recovered by a distributed resolution authority (RA).

In contrast to the two proposals above our scheme is fully anonymous and protects the user's privacy even when the authorities collaborate. We implement revocation, however, unlike in other approaches revocation can only be performed with the user's cooperation or in case he tries to cheat while obtaining pseudonyms.

## V. Building blocks

In this section we give an informal description of the cryptographic primitives used in our construction. The building blocks use advanced cryptographic concepts such as zero-knowledge proofs of knowledge. An introduction to these topics is given by Goldreich [15].

### *A. Dynamic Accumulators*

The concept of a one-way accumulator was originally introduced by Benaloh and de Mare [2]. It allows to aggregate and store multiple values in an accumulated hash of constant length. Any value that was added can later be demonstrated to be contained in the accumulator by providing a corresponding "witness". Whenever the accumulator is changed, all witnesses obtained previously must be updated. Camenisch and Lysyanskaya extend the basic concept and present a "dynamic accumulator" [7] that also allows to remove elements. Additionally they provide an efficient zero-knowledge protocol that can be used to prove knowledge of a value that is stored in the accumulator without revealing the value or the corresponding witness.

An instance of a dynamic accumulator consists of the accumulated value $v$, a trapdoor function $f$, some auxiliary information $aux_f$ that can be used to revert $f$, and the following operations.

$(v, f, aux_f) := \mathit{Initialize}(1^k)$ chooses $f$ and $aux_f$ according to the security parameter $1^k$ and initializes $v$.

$(v', w_i) := \mathit{Add}_f(v, x_i)$ adds the value $x_i$ to the accumulator. It returns the updated accumulated value $v'$ and a witness $w_i$. All witnesses for values that have been added to $v$ previously must be updated to work with the new accumulated value $v'$.

$w'_i := \mathit{UpdateAdded}_f(w_i, x_j)$ updates the witnesses $w_i$ for a value $x_i$ after a new value $x_j$ has been added to the accumulator. It returns the updated witness $w'_i$.

$res := \mathit{Contained}_f(v, x_i, w_i)$ checks whether $x_i$ is contained in the accumulated value $v$ using the witness $w_i$.

$v' := \mathit{Remove}_f(v, x_i, aux_f)$ remove $x_i$ from the accumulator using the auxiliary input $aux_f$. It returns the updated accumulated value $v'$. All witnesses for values that are contained in $v$ must be updated to work with the new accumulated value $v'$.

$w'_i := \mathit{UpdateRemoved}_f(w_i, x_i, x_j, v_{old}, v_{new})$ updates the witness $w_i$ for the value $x_i$ after some other value $x_j$ has been removed from the accumulator. The operation returns the updated witness $w'_i$. Note that $\mathit{UpdateRemoved}_f$ fails in case $x_i = x_j$. Obviously, it must not be possible to update a witness for a value that has been removed from the accumulator.

### B. CL signatures

The "CL signature" scheme presented by Camenisch and Lysyanskaya [8] was specifically created to be used as a building block in anonymity-enhancing cryptographic systems. The authors provide an efficient protocol for proving knowledge of a signature without revealing it. The scheme offers the following operations.

$(PK, SK) := \mathit{Keygen}(1^k)$ generates public key $PK$ and secret key $SK$.

$\sigma := \mathit{Sign}(m, PK, SK)$ signs the message $m$. This involves choosing a random prime number $e$ that is part of the resulting signature $\sigma$.

$res := \mathit{Verify}(\sigma, m, PK)$ checks whether $\sigma$ is a valid signature on the message $m$.

The CL signature scheme can be extended to support revocation by using a dynamic accumulator (cf. last section) as follows. This is an adaption of the approach presented in [7] to revoke CL credentials. The random value $e$ that is part of the signature is stored in a dynamic accumulator $A$ and the verification procedure is extended to check for $e$'s presence in $A$. The signature can be invalidate by removing $e$ from $A$. By using the zero-knowledge protocols given in [7], [8] the holder of a signature can demonstrate to another party that he holds a valid signature $\sigma$ and that the value $e$ (that is part of $\sigma$) is contained in a (public) accumulated value $v$. Neither the signature $\sigma$, the value $e$, nor the message $m$ are revealed during the proof.

### C. Periodic n-show credentials

Anonymous credentials were originally conceived by Chaum in 1985 [13]. They enable anonymous authentication, i.e. proving some entitlement without revealing any additional information such as the user's identity. Our scheme uses so called "periodic n-show credentials" proposed by Camenisch et al. [5]. They implement the additional restriction that a credential can be used at most $n$ times per time period. The scheme is constructed using the CL signature scheme. As outlined in the original paper it can be extended to support revocation. This is achieved by invalidating the CL signature $\sigma$ that is part of the so-called "e-token dispenser" as outlined in the previous section.

The credential scheme consist of an issuer $\mathcal{I}$ that provides a dispenser of e-tokens to each user $\mathcal{U}$ and a verifier $\mathcal{V}$ towards which $\mathcal{U}$ authenticates using the tokens. $n$ is a global system parameter. It offers the following operations and protocols.

$(pk_{\mathcal{I}}, sk_{\mathcal{I}}) := \mathit{IKeygen}(1^k, \mathit{params})$ generates the issuer's key pair.

$(pk_{\mathcal{U}}, sk_{\mathcal{U}}) := \mathit{UKeygen}(1^k, pk_{\mathcal{I}})$ generates the user's key pair.

$(A, MD, DS) := \mathit{VSetup}(1^k)$ initializes the verifier's dynamic accumulator $A$ and sets up the mapping database $MD$ and the double spending database $DS$ as empty lists.

$\mathit{Obtain}(\mathcal{U}(pk_{\mathcal{I}}, sk_{\mathcal{U}}, n), \mathcal{I}(pk_{\mathcal{U}}, sk_{\mathcal{I}}, n), \mathcal{V}(A, MD))$ Interactive protocol for the user to obtain an e-token dispenser $D$ that can be used $n$ times per period. $D$ contains (among other information) $pk_{\mathcal{U}}$, a CL signature $\sigma$ on $sk_{\mathcal{U}}$, and a list of counters $(J_{t_0}, J_{t_1}, ...)$ that indicate how many tokens have already been spent in each period $t_i$.[5] The value $e$ from the signature $\sigma$ is the user's revocation token. It is provided to $\mathcal{V}$ and stored in $A$ running $\mathit{Add}_f(v, e)$. $\mathcal{U}$ obtains the corresponding witness $w$. Furthermore $\mathcal{V}$ stores $(pk_{\mathcal{U}}, e)$ in its mapping database $MD$.

$\mathit{Show}(\mathcal{U}(D, pk_{\mathcal{I}}, t, n, w), \mathcal{V}(pk_{\mathcal{I}}, t, n, A, DS))$ Interactive protocol for the user to authenticate for the time period $t$ using the e-token dispenser $D$. The protocol involves proving knowledge of a signature $\sigma$ and proving that the value $e$ from $\sigma$ is contained in $A$, i.e. $D$ has not been revoked. $\mathcal{V}$ obtains a token serial number (TSN) $S$ and a transcript $\tau$. $S$ is formed by a deterministic one-way function using $\mathcal{V}$'s input $t$ and $J_t$. The verifier checks that $J_t < n$ and that $S$ has not been used before. (If a user was trying to cheat he had to either set $J_t \geq n$ or re-use a TSN.) Upon successful authentication ($S$, $\tau$) is stored in the double spending database $DS$. Finally $\mathcal{U}$ increments $J_t$.

[5] The original publication suggests using only one counter $J$ for the currently "active" time period $T$. However, this only works if $T$ is never decreased. As we do not require the user to request the pseudonyms in the order of their validity period, we modify the scheme to use a list of counters instead.

$pk_{\mathcal{U}} := \mathit{Identify}(pk_{\mathcal{I}}, S, \tau, \tau')$ With input of a TSN $S$ and two corresponding transcripts $\tau \neq \tau'$ the verifier can calculate the public key $pk_{\mathcal{U}}$ from the the dispenser $D$ that was used to create $S$. Using the mapping in $MD$, $\mathcal{V}$ can obtain $e$ and revoke the dispenser. Note that this only works if a user reused a TSN, i.e. he tried to authenticate more than $n$ times within one time period.

$A' := \mathit{Revoke}(A, e, aux_f)$ removes the value $e$ from the accumulator $A$ running $\mathit{Remove}_f(v, e, aux_f)$ thus invalidating the corresponding signature $\sigma$ which effectively revokes the corresponding dispenser $D$. The operation returns the updated accumulator.

## VI. The PUCA pseudonym scheme

In the following we present the PUCA pseudonym scheme. Our key contribution are modifications to the *Obtain-Pseudonyms* and *Revoke* phases from the basic scheme. The other phases are modified only as far as needed to setup the required cryptographic primitives. To obtain pseudonyms the user authenticates to the PCA using a periodic n-show credential, thus remaining fully anonymous. He can request pseudonyms for arbitrary time periods $t_i$. However, only up to $n$ pseudonyms can be requested in total for any time period. The only way a user's authentication credential can be revoked is if he (1) tries to cheat and requests more than $n$ pseudonyms for one time period or if he (2) voluntarily submits his revocation token to the PCA.

The roles are as follows: The LTCA acts as issuer $\mathcal{I}$, each of the participating vehicles is a user $\mathcal{U}$ and the PCA performs the role of the verifier $\mathcal{V}$. The value $n$ is a global system parameter which specifies how many pseudonyms a user may request for one time slot. Time is divided into discrete slots which are referenced by their start time $t$. The length of the time slots is another system parameter which controls the granularity of pseudonym validity and can be set to anything from a few minutes to several days.

### A. Protocols

We assume a secure, anonymous channel for all communication involving the PCA and the LTCA – e.g. a TLS connection over an anonymization network such as Tor[6]. Furthermore, we assume that a globally trusted root CA is in place, like in the basic scheme (cf. section IV-A).

*Initialization: Global system setup and key generation*

The LTCA executes the protocol $\mathit{IKeygen}(1^k, \mathit{params})$ and obtains the key pair $(pk_{\mathcal{I}}, sk_{\mathcal{I}})$. The PCA is provided with $pk_{\mathcal{I}}$ which it later needs to verify requests from the users. It runs $\mathit{VSetup}$, obtains $A$, $MD$ and $DS$ and initializes the two lists $E_{add}$ and $E_{delete}$. Furthermore, the PCA generates an ECDSA key pair $(sk_{PCA}, pk_{PCA})$ to sign pseudonym certificates. (Like in the basic scheme, the key pair is certified by the root CA.)

*Setup-Vehicle: Add a vehicle $\mathcal{U}$ to the system* (cf. Figure 2)

$\mathcal{U}$ executes the protocol $\mathit{UKeygen}(1^k, pk_{\mathcal{I}})$ and obtains the key pair $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$. Running the *Obtain* protocol, $\mathcal{U}$ obtains a token dispenser $D$ and a witness $w$. The value $e$ from the signature $\sigma$ that is part of $D$ serves as the user's revocation token. It should be kept in a safe place, e.g. stored as a printout together with the vehicles certificate of ownership. The PCA also obtains $e$ and stores it in $A$. The PCA adds $e$ to $E_{add}$ in order to enable other parties to update their witnesses.

*Obtain-Pseudonyms* (cf. Figure 3)

$\mathcal{U}$ updates $w$ by running $\mathit{UpdateAdded}(w, e_i)$ for all values $e_i$ in $E_{add}$ that have been added since the last protocol run and $\mathit{UpdatedDeleted}(w, e_j)$ for all value in $E_{delete}$ respectively.

For each pseudonym to be requested: $\mathcal{U}$ creates a pseudonym key pair $(ppk_k, psk_k)$ and runs the *Show* protocol to authenticate with the PCA specifying $t_k$ as the time period. It sends the pseudonym public key $ppk_k$ signed with $psk_k$ to prove ownership. As part of the *Show* protocol the PCA obtains $(S, \tau)$ and verifies that $S$ is not already stored in $DS$. This ensures that no more than $n-1$ pseudonyms have already been requested for the specified time period. If the protocol exits successfully, $(S, \tau)$ is added to $DS$, the PCA signs and returns $(ppk_k, t_k)$ to $\mathcal{U}$. Finally, $\mathcal{U}$ increments $J_{t_k}$. If validation fails because $\mathcal{U}$ was "overspending", his identity can be revealed using the *Identify* protocol and the PCA can take appropriate actions.

*Communication*

We do not make any changes to the communication phase from the basic scheme (cf. section IV-A).

*Revocation*

In order to revoke a user's e-token dispenser the revocation token $e$ is required. It can either be obtained through the *Identify* protocol (if the user tried to cheat) or be provided by the user voluntarily (e.g. if the vehicle will be sold or was stolen). To revoke a user's dispenser $D$, the PCA runs $\mathit{Revoke}(A, e)$ and adds $e$ to $E_{delete}$. Note that revocation of an honest user without his consent is not possible as $e$ is never revealed during the regular protocol runs. Like in the basic scheme we do not consider revocation of pseudonyms.

If a revoked vehicle tries to obtain pseudonyms, it cannot be identified by the PCA due to the anonymity of the credential scheme. However, once an IVS discovers that its credential has been revoked, it can act accordingly, e.g. contact its owner, report its location and possibly execute further anti-theft measures.

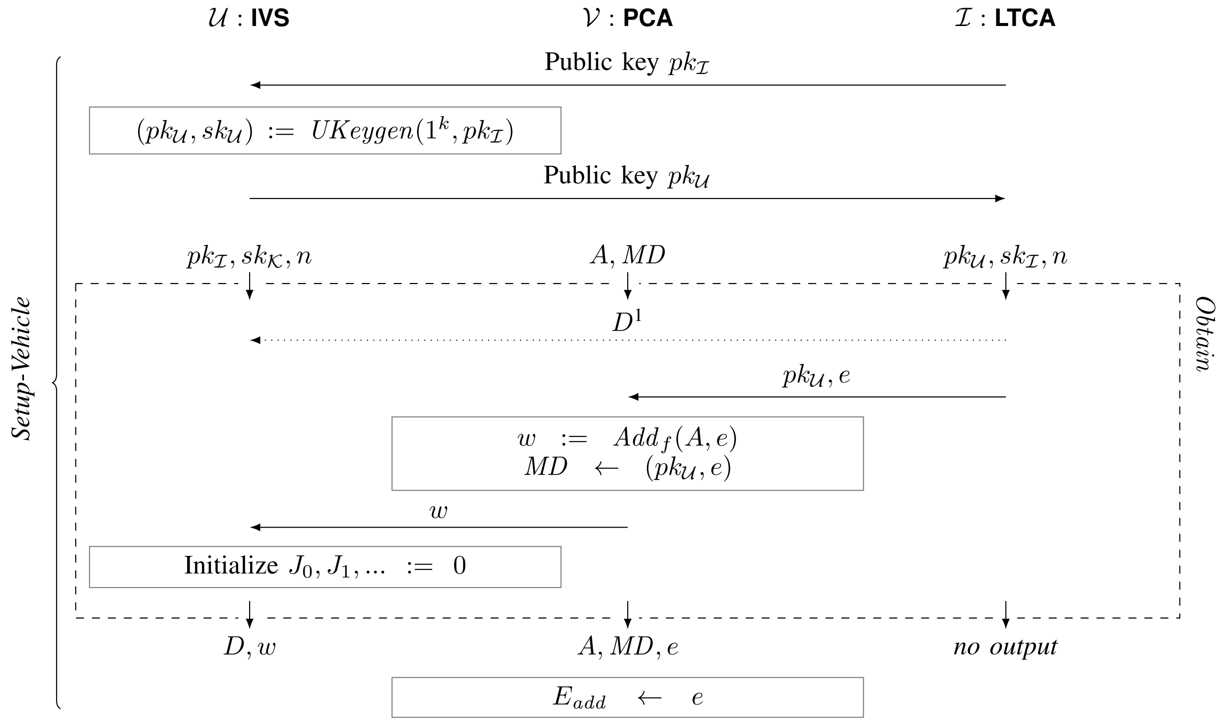
### B. Extensions and modifications

The PUCA scheme is quite flexible and several modifications can be made.

*1) Multiple PCA instances:* In the basic scheme it is possible to run several PCA instances, e.g. to handle the load of a large number of users. This is also possible in our scheme. In that case the values $MD$, $E_{add}$ and $E_{delete}$ must be synchronized among all PCAs. Note that the accumulator $A$ need not be synchronized as modifications to it can be done locally based on $E_{add}$ and $E_{delete}$.

*2) Merge LTCA and PCA:* Separation of the LTCA and the PCA is not a requirement to guarantee the users' privacy in contrast to the basic scheme. In order to reduce communication and management overhead, the two entities can be merged into one central authority.

[6] https://www.torproject.org/


$\mathcal{U}$ : IVS
$\mathcal{V}$ : PCA
$\mathcal{I}$ : LTCA
Public key $pk_{\mathcal{I}}$
$(pk_{\mathcal{U}}, sk_{\mathcal{U}}) := UKeygen(1^k, pk_{\mathcal{I}})$
Public key $pk_{\mathcal{U}}$
$pk_{\mathcal{I}}, sk_{\mathcal{K}}, n$
$A, MD$
$pk_{\mathcal{U}}, sk_{\mathcal{I}}, n$
$D^1$
$pk_{\mathcal{U}}, e$
$w := Add_f(A, e)$
$MD \leftarrow (pk_{\mathcal{U}}, e)$
$w$
Initialize $J_0, J_1, ... := 0$
$D, w$
$A, MD, e$
no output
$E_{add} \leftarrow e$
Setup-Vehicle
Obtain

[1] $D$ is jointly created by $\mathcal{U}$ and $\mathcal{I}$ in an interactive protocol.

Figure 2. Add a new vehicle to the system in the *Setup-Vehicle* phase. The e-token dispenser $D$ is created and initialized; the PCA stores information required in case of revocation.

### C. Integration into existing systems

As our scheme only modifies the pseudonym issuance procedure it can be deployed alongside the basic scheme. PUCA users can securely communicate with Car2X participants that use a different backend to obtain their pseudonyms and vice versa. To establish interopability, the trust hierarchy must be set up such that all PCA certificates are signed by a globally trusted root CA. The compatibility enables both a gradual deployment as well as the coexistence of the schemes on the long-term.

## VII. Evaluation and discussion

We evaluate our scheme against the requirements from section III, first with regard to security and privacy, and then with regard to performance and communication overhead.

### A. Security and privacy evaluation

R.1) *Authentication* is implemented using periodic n-show credentials. R.2) *Credential usage* is restricted as the credential scheme allows only $n$ pseudonyms to be requested for any time period. R.3) *Revocation* is possible with the user's cooperation or in case of "overspending" i.e. requesting too many pseudonyms for the same time period. R.4) *Strong anonymity* is provided by the anonymous n-show credential scheme. As no identifiers are exchanged and stored, anonymity is protected both against colluding backend providers as well as against attackers that might compromise backend systems during operation. Note that we assume an anonymous communication channel between the IVS and the PCA for all protocols, so that the PCA does not learn the IVS's identity based on communication identifiers. R.5) *Perfect forward privacy* is established as even after revoking an e-token dispenser, transactions performed previously remain anonymous.

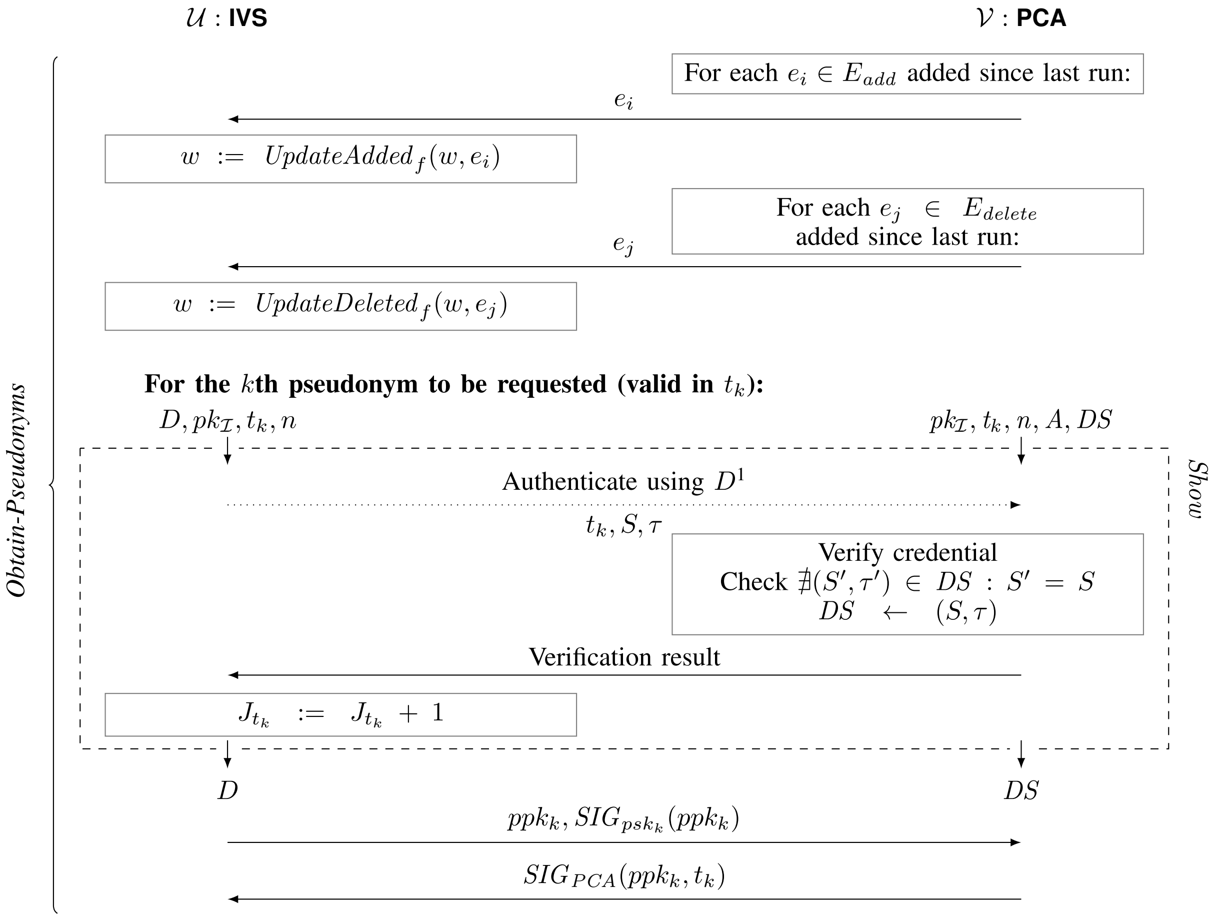
The security of PUCA is based on the cryptographic assumptions the respective schemes make [5], [7], [8]. Note that our scheme could also be implemented using different instantiations of the respective cryptographic building blocks, e.g. based on bilinear mappings [6], [9].

### B. Performance evaluation

R.6) *Real-time constraints* in the *Communication* phase are satisfied by using efficient ECDSA signatures. We make no change to the *Communication* phase from the basic scheme. R.7) *Scalability* on the server-side can be achieved by replicating the PCA as illustrated in section VI-B1. Furthermore, several instances of PUCA can be deployed in parallel to cope with a very large number of participants.

In the following we elaborate on our scheme's performance and communication overhead in the *Obtain-Pseudonyms* phase, where the n-show credential scheme is used. We assume a typical usage pattern of an IVS requesting pseudonyms every few days with less than a hundred pseudonyms per request.

Lapon et al. evaluated the performance of the CL credential scheme and the use of a dynamic accumulator for revocation [20]. They found that showing a credential takes under 400 ms for the prover and under 300 ms for the verifier. Updating the witness for a credential after 500 values have been added or removed from the accumulator takes less than one second.


$\mathcal{U}$ : IVS
$\mathcal{V}$ : PCA
For each $e_i \in E_{add}$ added since last run:
$e_i$
$w := UpdateAdded_f(w, e_i)$
For each $e_j \in E_{delete}$ added since last run:
$e_j$
$w := UpdateDeleted_f(w, e_j)$
For the $k$th pseudonym to be requested (valid in $t_k$):
$D, pk_{\mathcal{I}}, t_k, n$
$pk_{\mathcal{I}}, t_k, n, A, DS$
Authenticate using $D^1$
$t_k, S, \tau$
Verify credential
Check $\nexists(S', \tau') \in DS : S' = S$
$DS \leftarrow (S, \tau)$
Verification result
$J_{t_k} := J_{t_k} + 1$
$D$
$DS$
$ppk_k, SIG_{psk_k}(ppk_k)$
$SIG_{PCA}(ppk_k, t_k)$
Obtain-Pseudonyms
Show

[1] Interactive protocol between $\mathcal{U}$ and $\mathcal{V}$. Note that the verifier learns only the values $t_k$, $S$ and $\tau$.

Figure 3. An IVS obtains new pseudonym certificates in the *Obtain-Pseudonyms* phase. After updating its witness $w$, pseudonyms are obtained one by one. Authentication is repeated for each pseudonym and its validity period $t_k$.

In table I we compare their analysis to the n-show credential scheme with respect to the number of exponentiations and multi-exponentiations performed. While the results for the verifier are similar, the prover has to do about 70 percent more exponentiations in our case.

Table I. NUMBER OF (MULTI-BASE) EXPONENTIATIONS PERFORMED BY THE CL CREDENTIAL AND N-SHOW CREDENTIAL SCHEMES

| | **CL credentials** | **n-show credentials**[7] |
|---|---|---|
| Show credential | 54 | $55 + 18$ |
| *Prover* | 27 | $35 + 11$ |
| *Verifier* | 27 | $20 + 7$ |
| Update witness | 1 | 1 |

Table II shows our analysis of the communication overhead based on the zero-knowledge protocols given in [5], [7], [8] and the implementation of the respective proofs in the idemix library [1]. We use the same length parameters as Lapon et al. [20], which are based on a 2048 bit RSA modulus. To show the credential, the IVS has to send about 11 kB to the PCA and receive about 210 B. Updating the witness takes 63 B sent from the PCA to the IVS.

Table II. COMMUNICATION OVERHEAD FOR THE N-SHOW CREDENTIAL SCHEME

| | **IVS → PCA** | **PCA → IVS** |
|---|---|---|
| Show credential + Check for revocation | $10099 + 3597$ Bytes | $189 + 20$ Bytes |
| Update witness | 0 Bytes | 63 Bytes |

The *Show* protocol is executed for every pseudonym that is requested. We estimate that the computational overhead for requesting 100 pseudonyms (and thus executing the *Show* protocol 100 times) will be less than one minute for the PCA and a bit more than a minute for the IVS. The communication overhead will be about 1.3 MB of data sent from the IVS to the PCA and about 210 kB received. The time needed to update the credential's witness prior to running the *Show* protocol is linear to the number of updates to the accumulator since the last pseudonym request. We expect that at most 10.000 vehicles will be added to or removed from the system between two pseudonym requests, which would result in an estimated computational overhead of less than 20 seconds and 615 kB transferred for the PCA to the IVS.

The measurements in [20] were taken on a 2.53 GHz, 4 GB RAM laptop computer. As current automotive hardware is not quite as powerful, we expect the performance on a typical

[7] Numbers for showing the credential (taken from [5]) plus for the revocation check (our own analysis of the protocol given in [7]).

car PC[8] to be less but still practical. For our scenario we estimate that the pseudonym update will take no more than five minutes. The load on the server side can be handled using standard techniques of replication (c.f. section VI-B1) and load balancing.

We see that the use of advanced cryptography certainly incurs elevated performance requirements and communication overhead. However, also with regard to the expected increase in computing power in the next years and broad availability of 3G/4G networks, we conclude that an implementation of PUCA is practical. Note that the credential scheme is only used during the *Obtain-Pseudonyms* phase where the vehicle may be stationary and performance is not critical, e.g. when at a gas station or parked at home. The performance-critical *Communication* phase suffers no degradation as we do not change the way pseudonyms are used to secure messages with ECDSA signatures.

## VIII. Conclusion

We present PUCA, a pseudonym issuance scheme with strong guarantees for user-privacy. The scheme employs advanced cryptography that protects the users' anonymity even against colluding backend providers. Revocation is possible, however only with the user's cooperation or in case he tries to cheat while requesting pseudonyms. As we do not change the communication phase from the widely-accepted "basic pseudonym scheme", PUCA can be deployed alongside existing solutions or can be used as a drop-in replacement with minimal changes.

Privacy concerns are more and more common. Skepticism towards public authorities rises due to recent revelations about massive surveillance measures being in place in some countries. This shows an increasing demand for a solution which puts the users back in control of their privacy. With our scheme we show that strong privacy protection also against authorities is possible while still maintaining full functionality, with only accountability left out intentionally. By adopting our solution ITS developers and standardization bodies could demonstrate their strong commitment to privacy protection.

## Acknowledgments

This work was partially funded within the project CONVERGE by the German Federal Ministries of Education and Research as well as Economic Affairs and Energy.

## References

[1] "Specification of the Identity Mixer Cryptographic Library – Version 2.3.40," *IBM Research – Zurich*, 2013.

[2] J. Benaloh and M. De Mare, "One-way accumulators: A decentralized alternative to digital signatures," in *Advances in Cryptology—EUROCRYPT'93*. Springer, 1994, pp. 274–285.

[3] N. Bissmeyer, H. Stubing, E. Schoch, S. Gotz, 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.

[4] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. ACM, 2007, pp. 19–28.

[5] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to win the clonewars: efficient periodic n-times anonymous authentication," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 201–210.

[6] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *Public Key Cryptography–PKC 2009*. Springer, 2009, pp. 481–500.

[7] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Advances in Cryptology—CRYPTO 2002*. Springer, 2002, pp. 61–76.

[8] ——, "A signature scheme with efficient protocols," in *Security in communication networks*. Springer, 2003, pp. 268–289.

[9] ——, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology–CRYPTO 2004*. Springer, 2004, pp. 56–72.

[10] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 21–30.

[11] CAR 2 CAR Communication Consortium, "C2C-CC public key infrastructure memo," *CAR 2 CAR Communication Consortium, Technical Report*, Feb. 2011.

[12] ——, "Memorandum of understanding on deployment strategy for cooperative ITS in europe," Jun. 2011.

[13] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[14] C. Gañán, J. L. Muñoz, O. Esparza, J. Mata-Díaz, and J. Alins, "PPREM: Privacy preserving REvocation mechanism for vehicular ad hoc networks," *Computer Standards & Interfaces*, vol. 36, no. 3, pp. 513–523, Mar. 2014.

[15] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*, 1st ed. Cambridge University Press, Jan. 2007.

[16] J. Guo, J. P. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *2007 Mobile Networking for Vehicular Environments*. IEEE, 2007, pp. 103–108.

[17] L. Huang, "Secure and privacy-preserving broadcast authentication for IVC," Master's thesis, University of Twente, the Netherlands, Jul. 2012.

[18] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

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

[20] J. Lapon, M. Kohlweiss, B. De Decker, and V. Naessens, "Performance analysis of accumulator-based revocation mechanisms," in *Security and Privacy–Silver Linings in the Cloud*. Springer, 2010, pp. 289–301.

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

[22] P. Papadimitratos and J.-P. Hubaux, "Report on the secure vehicular communications: results and challenges ahead workshop," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 2, pp. 53–64, 2008.

[23] F. Schaub, F. Kargl, Z. Ma, and M. Weber, "V-tokens for conditional pseudonymity in VANETs," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[24] F. Schaub, Z. Ma, and F. Kargl, "Privacy requirements in vehicular communication systems," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*. IEEE, 2009, pp. 139–145.

[25] A. Singh, "Restricted usage of anonymous credentials in VANET for misbehavior detection," Master's thesis, University of Applied Sciences, Frankfurt am Main, Germany, Jun. 2012.

[8] e.g. NEXCOM VTC 7220-BK, Intel®Core$^{\text{TM}}$i7 1.7 Ghz, 2 GB RAM