Autonomous Certification with List-based Revocation for Secure V2V Communication

Anup Kumar Bhattacharya (*)
Abhijit Das (*)
Dipanwita Roychoudhury (*)
(*) Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur, India
anup, abhij, drc@cse.iitkgp.ernet.in,
Aravind Iyer (#)
Debojyoti Bhattacharya (#)
(#) General Motors Technical Centre India
India Science Lab, Bangalore
debojyoti.bhattacharya, aravind.iyer@gm.com

Abstract. Privacy and authenticity are two essential security attributes of secure Vehicle-to-Vehicle communications. Pseudonymous Public Key Infrastructure (PPKI), an extension of standard PKI, has been proposed to achieve these security attributes. In Pseudonymous PKI, a user needs certificates or pseudonyms periodically from the Certificate Authority (CA) to authenticate messages anonymously. But the infrastructure presence to communicate with the CA may not be ubiquitous, at least in the initial development phases of vehicular communication. Another proposal, PKI+ reduces dependence on the CA by allowing users to generate pseudonyms autonomously. However, user revocation in PKI+ is rather inconvenient, since it requires the entire network of non-revoked users to be reconfigured after each such event. In this paper, we propose PKI++, an improvement over PKI+, which brings together the desirable features of PKI and PKI+, namely autonomous certification and list-based revocation. We compare the proposed algorithm with PKI and PKI+, and show revocation to be less costly in PKI++.

Keywords: Authentication, Privacy, VANET, Revocation, PKI, PKI+

1 Introduction

Automotive driver assistance applications based on Vehicle-to-Vehicle (V2V) wireless communications have attracted a lot of attention in the past few years [3]. Driver assistance applications aim to assist drivers in avoiding accidents by providing early warnings and advisories. Security becomes critical in these applications, since drivers of vehicles are expected to rely on the advisories these applications would provide. One important requirement of secure V2V applications is that it must ensure message authentication and data integrity. These attributes are important because, based on the received messages, drivers take important decisions. The IEEE 1609.2 standard [7] specifies V2V security protocols and recommends the use of public-key certificates to sign messages.

Privacy is another important issue in V2V communications. The notion of privacy in V2V settings refers to different things in different protocol layers. In security layer, privacy is achieved by attaining anonymity and unlinkability. At higher layers, privacy may be compromised due to some information in the message or some other reasons (GPS, license plate, etc). By privacy, we here refer to privacy in the security layer.

A public-key infrastructure (PKI) satisfies security requirements like authenticity and data integrity. But signing multiple messages with the same key pair and certificate endangers the privacy of the vehicle. Multiple messages signed using the same private key are linkable to an adversarial observer. In order to protect user privacy, pseudonymous PKI has been proposed by Parno and Perrig [10] and Calandriello et al. [4]. Pseudonymous PKI provides privacy through the use of multiple certificates or pseudonyms obtained from the infrastructure. By infrastructure, we mean the Certification authority (CA) or some Road Side Units (RSU) which act as communication interfaces to the CA. These pseudonyms do not contain any direct identity information of the vehicle. To appear anonymous to other vehicles and adversaries, a vehicle may use each pseudonym for signing a small number of messages so that only a limited number of its messages may be linked to one another. Each such pseudonym must be certified by the CA. Therefore, in pseudonymous PKI, vehicles need to communicate frequently with the infrastructure to obtain new sets of pseudonyms.

In [13], Studer et al. consider another certification scheme called Temporary Anonymous Certified Keys (TACKs) where vehicles communicate with the CA through Regional Authorities (RA). RAs store updated revocation information from the group manager, and grants short-lived keys to vehicles only if they are not revoked by the group manager.

Both the schemes mentioned above require extensive communication between vehicles and the infrastructure and are not suitable except when the presence of the infrastructure is ubiquitous.

To eliminate frequent communication with the CA to obtain pseudonyms, autonomous certification schemes are proposed by researchers [4, 8]. In an autonomous certification scheme, a vehicle obtains an initial set of master credentials from the CA and generates certificates or pseudonyms, when required, without any involvement of the CA.

In [1], Armknecht et al. propose a method for a completely autonomous generation of pseudonyms for V2V, based on the PKI+ concept proposed by Zeng [15]. In PKI+, a user can generate distinct public keys and the corresponding certificates for these public keys without any involvement or direct communication with the CA. However, communication with the CA is required during the initial registration phase to obtain a master public key and a master certificate, and to obtain updated parameters after each revocation.

Weerasinghe and Fu propose an efficient and scalable authentication protocol [14] for VANET. Vehicles generate keys and certificates using a short-term authorization certificate from the CA. This scheme ensures anonymity, conditional and location privacy of the vehicles. Tracing and revocation of the malicious vehicles are achieved by publishing μ-CRL which contains a list of vehicles revoked within a short time period. Fan et al. propose another autonomous certificate generation scheme [6] using bilinear pairings. The scheme revokes vehicles by publishing RL.

The revocation process in PKI+ is cumbersome. In PKI+, revocation is achieved by updating the parameters of non-revoked vehicles. After each revocation event, the CA updates its own public key and some public parameters. Each non-revoked vehicle has to contact the CA to receive the new parameters. As a result, non-revoked vehicles cannot communicate with one another until they come across some RSU. Therefore, the revocation process can be very costly, particularly if the number of vehicles in the network is large or if connectivity with the infrastructure is sparse.

In PKI+, revocation may also be achieved by publishing the tracing result of the vehicle to be revoked. But active adversaries may use this result to determine whether one particular pseudonym has been used by the revoked vehicle to sign a message and to link all previous messages signed by the revoked vehicle.

Our Contribution: In this paper, we propose PKI++, a modification of PKI+, which overcomes the above disadvantages of PKI+. Assuming loose time-synchronization among vehicles, PKI++ combines the autonomous certification feature of PKI+ with the list-based revocation of PKI, for obtaining improved V2V privacy with minimal CA support. We summarize our contributions as follows.

- In PKI++, all non-revoked vehicles are not forced out of the group at every revocation event. Revocation is achieved by publishing revocation lists. Thus, a non-revoked vehicle without the latest revocation list can still authenticate its messages, though it may accept messages from some revoked vehicles.
- PKI++ also ensures backward unlinkability of revoked vehicles. Thus, compared to PKI+, revocation in PKI++ is made less costly, albeit at the expense of additional storage and computation overhead.

The rest of this paper is organized as follows. Section 2 provides some basic background on PKI and PKI+. Section 3 describes the PKI+ algorithm and its disadvantages. In sections 4–6, we present our scheme (PKI++), and analyze its performance, and compare it with PKI and PKI+. Finally, Section 7 concludes the paper.

2 Background

We now briefly describe the security attributes of a vehicle participating in V2V communications.

1. Authentication: A receiver of a message should be able to determine whether the signer is trusted. This is important because, based on the received mes-

sage, the driver would take some action and, therefore, the message should be authentic.

2. Data integrity: A message sent by a vehicle cannot be tampered with by an active adversary without being detected by a recipient. If the message is corrupt, it may lead to some fatal consequences for the driver.
3. Anonymity: An adversary should not be able to map the digital identity of a vehicle to its physical identity. Preserving privacy is an important issue in V2V communications.
4. Non-repudiation/Traceability: Non-repudiation (Traceability) requires that after signing a message, no sender will be able to deny that to any third party (infrastructure). This incorporates accountability in V2V communications.
5. Unlinkability: An adversary should not be able to relate multiple messages sent by the same vehicle. Casual observers should not be able to track down a drivers trajectory in the long term.
6. Backward unlinkability: A vehicle may be revoked from the system due to sensor malfunctioning or private key compromise. Therefore, past messages generated by that revoked vehicle should remain unlinkable.

In PKI, every vehicle has a private key and a public key, issued by a certification authority (CA), along with a digital certificate to authenticate the public key. To authenticate a message, a vehicle signs the message with its private key using a digital signature algorithm. A verifier of the message-signature pair first validates the authenticity of the public key of the sender using the certificate, and then verifies the signature using the authenticated public key. IEEE 1609.2 [7] mandates the use of the Elliptic Curve Digital Signature Algorithm (ECDSA) for V2V message authentication.

Signing/verification algorithms ensure authentication, data integrity and non-repudiation for PKI. Revocation in PKI is achieved by broadcasting Certificate Revocation Lists (CRL). CRLs store the certificates of all revoked vehicles. When a signed message is received, the receiver compares the certificate of the signer with certificates in the CRL to ensure that the message has not come from a revoked vehicle. After each revocation event, the CA updates the CRL. If some vehicle is not in the immediate vicinity of some RSU just after a revocation event, there will be some delay for the vehicle to obtain the updated CRL. This delay should be as small as possible, as this is the time during which a vehicle would treat messages from newly revoked vehicles as authentic. There has been considerable work on user revocation for V2V communication, and also to minimize its associated delay. In [9] , Laberteaux et al. show how the delay for revocation can be minimized by letting the vehicles broadcast the updated CRL. In [12] , Ren et al. propose the delta-RL method to achieve revocation efficiently in group-signature-based privacy-aware PKI.

Signing multiple messages using the same pseudonym may endanger privacy for the vehicle and render the messages signed by the vehicle linkable. Using each pseudonym to sign a small number of messages, a vehicle can ensure anonymity and unlinkability. To change the pseudonyms frequently, a vehicle either has to frequently communicate with the CA to obtain new sets of pseudonyms, or has to store a large number of pseudonyms on-board. PKI+ intends to reduce communication with the CA by autonomous certificate generation. In PKI+, each vehicle can create a very large number of pseudonyms without any involvement of the CA. Communication with the CA is required during the initial registration phase and after each revocation event. During the registration phase, the CA registers a vehicle after verifying its credentials, and provides a root public key and a root certificate. After each revocation event, the CA updates its public key, and all vehicles need to communicate with the CA to obtain the new set of parameters. We describe PKI+ in more detail in the next section, since our scheme is based on the same mathematical constructs as used in PKI+.

3 PKI+

In the following we describe some mathematical constructs which are needed for understanding of our modified scheme.

3.1 Bilinear pairings

Let g_1, g_2, g be generators of finite multiplicative cyclic groups G_1, G_2 and G respectively with $|G_1| = |G_2| = |G| = p$, where p is some large prime. A bilinear map $e : G_1 \times G_2 \rightarrow G$ satisfies the following three properties [5]:

1. *Bilinearity*: For all $h_1 \in G_1$ and $h_2 \in G_2$ and for all $a, b \in Z_p$, the following relation holds.

$$e({h_1}^a, {h_2}^b) = e(h_1, h_2)^{ab}$$

2. *Non-degeneracy*: There exists $h_1 \in G_1$, $h_2 \in G_2$ such that $e(h_1, h_2)$ is not the identity of G.
3. *Ease of computation*: There exists an efficient algorithm to compute e.

3.2 Signature of knowledge proof

Let G_1 and G_2 be finite cyclic groups, g_1, h_1 generators of G_1, and g_2 a generator of G_2. A signature of knowledge proof indicates that the signer knows some secret, without which the signature cannot be computed. The signature can be computed only when the discrete logarithms u_1 and u_2 of y_1 and y_2 are known with bases g_1, g_2 and h_1, where $y_1 = {g_1}^{u_1} \cdot {h_1}^{u_2}$ and $y_2 = {g_2}^{u_2}$. A signature of knowledge proof [2] for signing the message m is given by

$$SKP\{(x_1, x_2) : y_1 = {g_1}^{x_1}.{h_1}^{x_2} \wedge y_2 = {g_2}^{x_2}\}(m).$$

3.3 The PKI+ algorithm

In [15], Zeng designs a new pseudonymous PKI algorithm called PKI+ which allows autonomous certificate generation. The basic advantage of using PKI+ is that any vehicle can generate a very large number [1] of pseudonymous key pairs using the root public key and the root certificate issued by the CA. Therefore, PKI+ can provide privacy and unlinkability with reduced communication with the infrastructure.

Our scheme is based on the construction of PKI+, so we describe the main phases of PKI+.

1. *Setup*: The setup algorithm, on input of a security parameter 1^k, outputs a bilinear map along with some associated parameters.

$$(p, G_1, G_2, G, g_1, g_2, e) \leftarrow Setup(1^k)$$

   Here, p is a prime of bit length k, G_1, G_2 and G are three groups of order p, $e : G_1 \times G_2 \to G$ is a bilinear map, and g_1 and g_2 are some generators of the groups G_1 and G_2 respectively. The private key and the public key of the CA are given respectively by a and (p, G_1, G_2, G, g_1, g_2, e, h_1, h, A), where $A = {g_2}^a$. The version of the key pair is denoted by Ver.
2. *Peer registration*: A vehicle computes its root public key (h, h^x), where x is the private key of the vehicle and $h \in G_1$. The vehicle sends (ID, y, y') to the CA, where ID is the identity of the vehicle, $y = h^x$ and $y' = {g_2}^x$. It obtains the root certificate (t_g, t_h, z), where $t_g, t_h \in G_1^2$ and $z \in Z_p$, given by the following.
   (a) $z = \text{Hash}(ID \mid y \mid s_{id} \mid \xi)$, s_{id} is a proof of the ownership of ID and the root public key, and $\xi \in Z_p$.
   (b) $t_g = {g_1}^{\frac{1}{a+z}}$ and $t_h = (h_1.h^x)^{\frac{1}{a+z}}$.
3. *Peer-to-peer authentic communication*: A vehicle can generate a new set of public keys and the corresponding certificates by itself. It uses its root public key, its root certificate, and its own private key to generate a new public key and the associated certificate without any direct contact with the CA. The sender, with its pseudonymous public key (PPK) (t, t_y) where $t = (t_g.t_h)^r$ and $t_y = t^x$ for a random $r \in Z_p$ and with the public-key certificate (PCert) (Ver, s) where Ver is the version and s is the signature of knowledge proof, generates the signature s_m on a message m.
   (a) The signer computes signature of knowledge proof as
   $s = SKP\{(x_1, x_2, x_3) : e(t, A) = {v_1}^{x_1} \cdot ({v_2}^r)^{x_2} \cdot {v_3}^{x_3} \wedge 1 = (t^{-1})^{x_3} \cdot {t_y}^{x_1}\}$ $(Ver) \in {Z_p}^4$,
   where $v_1 = e(g_1 \cdot h_1, g_2)$, $v_2 = e(t_g \cdot t_h, {g_2}^{-1})$, and $v_3 = e(h, g_2)$.
   (b) The signer computes signature s_m on message m as
   $s_m = SKP\{(x_1) : t_y = t^{x_1}\}(m) \in {Z_p}^2$

   The verifier can check the authenticity of the received signature by checking the version of the certificate. If the version is less than that of its own certificate, the verifier rejects the signature. If the version number is more than that of its own, it contacts the CA to obtain the new set of public keys and certificates. If the versions match, it verifies the signature using the signer's public key as follows.

(a) The verifier checks that s is a valid signature of knowledge proof with respect to

$$s = SKP\{(x_1, x_2, x_3) : e(t, A) = e(g_1 \cdot h_1, g_2)^{x_1} \cdot e(t, {g_2}^{-1})^{x_2} \cdot e(h, g_2)^{x_3} \wedge \\ 1 = (t^{-1})^{x_3} \cdot {t_y}^{x_1}\}(Ver)$$

4. *Tracing*: The CA can trace a vehicle by using the vehicle's PPK and PCert values and using the following relation.

$$e(t, y'_i) = e(t_y, g_2).$$

The process involves matching y'_i of the vehicle in the database of all vehicles, maintained by the CA.

5. *Revocation*: In order to revoke a vehicle, the CA generates a new set of public parameters with increased version Ver, and broadcasts these parameters. Based on the new public key of the CA, each non-revoked vehicle updates its own parameters and the version number.

Some parameters of PKI+ are listed in Table 1 for ready future references.

Table 1. Scope of parameters in PKI+

Name	**Description**
CA public-key	(p, G_1, G_2, G, g_1, g_2, e, h_1, h, A) of version Ver, these parameters are known to everybody.
a	CA private-key, accessible only to CA.
x	Private key of a vehicle.
y, y', ID	Values sent to the CA by a vehicle during registration. Known to only that vehicle. CA maintains a database of y' for all vehicles.
t_g, t_h, z	Root certificate of a vehicle, computed by CA; known by the vehicle and the CA.
v_1, v_3	Can be computed and known to every vehicle.
v_2	Computed and known to the signing vehicle
PPK (t, t^y)	Pseudonymous public key of the vehicle, accessible to everybody.
PCert (Ver, s)	Pseudonymous public-key certificate of the vehicle, accessible to everybody.
s	Signature of knowledge proof.
s_m	Signature on message m.

PKI+ derives its security from the presumed intractability of certain computational problems such as the discrete-log and the bilinear Diffie-Hellman problems (see [15] for the details). We will not study these topics in this paper.

3.4 Disadvantages of PKI+

Although PKI+ ensures anonymity and unlinkability of the vehicles, its revocation mechanism is very costly and has a number of shortcomings. Whenever a vehicle is revoked, every non-revoked vehicle has to update its parameters in order to continue communication in the network. This imposes a huge burden to all non-revoked vehicles. The time period between the revocation of a vehicle and the updating of the version across the entire network can be referred to as *window of outage*. After one revocation event, all non-revoked vehicles will not be able to communicate until they receive the new set of parameters from the CA through an RSU, and for this time period, they essentially remain out of the group. Although PKI+ eliminates the necessity of communication with the infrastructure for refreshing pseudonyms, it necessitates communication between each non-revoked node and the infrastructure after each event of revocation. In situations where the presence of the infrastructure is not ubiquitous, this window of outage may be quite large. It is worthwhile to note here that this problem does not pertain to the original PKI, that is, non-revoked vehicles continue to communicate securely with each other irrespective of the availability of an updated CRL. Below, we describe our proposed algorithm PKI++ which overcomes the above disadvantages of PKI+.

4 PKI++

4.1 Motivation and our approach

The problem with the window of outage in PKI+ may be handled in two ways.

1. *Revocation by publishing Certificate Revocation List*: The CA can always purge compromised users by publishing a certificate revocation list (CRL). Though this approach is suitable for PKI, it does not work in autonomous certification scheme such as PKI+, where a vehicle can generate a huge number of new certificates by itself. It is impractical, if not infeasible, to let the CRL contain all these revoked certificates.
2. *Revocation by publishing initial commitment to knowledge* (y'): In PKI+, the CA may purge or revoke a compromised or bad vehicle by publishing the root commitment of the vehicle, y'. However, on publication of the root commitment y', revoked vehicles do not remain backward unlinkable, as the knowledge of y' can be used by any adversary to link earlier messages. The problem seems to happen due to the absence of any time element in the root commitment y'. If the CA publishes y', then other vehicles can verify whether a public-key (t', ${t_y}' = t'^{x}$) has been generated by that vehicle, since for any t', t'' and $x' \neq x$, we have
   (a) $e(t', y') = e({t_y}', g_2)$
   (b) $e(t'', y') \neq e(t''^{x'}, g_2)$

In order to circumvent the difficulties pointed out above, our scheme divides a reasonably long time span (like a day) into time intervals and uses different generators g_2 for group G_2 in different time intervals. Our scheme assumes loose time-synchronization among vehicles. Revocation in the i-th interval is achieved by publishing the $y'^{(j)}$ values for the subsequent intervals j. This way our scheme can achieve forward linkable revocation and backward unlinkability. We follow the steps mentioned below.

1. *Time-slot division*: Let T denote the total number of time intervals. We number the intervals as 1, 2, ..., T.
2. *Parameters for each slot*: We use different generators $g_2^{(1)}$, $g_2^{(2)}$, ..., $g_2^{(T)}$ of G_2 and different y' values $y'^{(1)}$, $y'^{(2)}$, ..., $y'^{(T)}$ for the T intervals. The generators $g_2^{(i)}$s are so chosen that they are not trivial multiples/powers of one another.
3. *Revocation*: To revoke a vehicle during the i-th interval, the CA publishes $y'^{(i)}$, $y'^{(i+1)}$, $y'^{(i+2)}$, ..., $y'^{(T)}$ of the vehicle for all subsequent intervals. Forward linkable revocation is thus achieved. Backward unlinkability is also preserved up to interval $(i-1)$, as $y'^{(j)}$ are not published for intervals $j =$ 1, 2, ..., $(i-1)$. Non-revoked vehicles continue to use the same parameters ${g_2}^{(j)}$ and $y'^{(j)}$ for $j \geq i$.
4. *Version update*: After T time intervals (such as after a day), all parameters of the CA and the vehicles are altered and the version of the certificates is updated.

4.2 The PKI++ algorithm

The algorithm PKI++ is described as follows. We highlight only the changes introduced in our scheme.

1. *Setup*: We choose groups G_1, G_2 and G of prime order p together with a bilinear pairing $e : G_1 \times G_2 \to G$. The only change with respect to PKI+ in this phase is that here we use different generators g_2 of the group G_2 and different y' values for different time intervals. Thus, the modified public key of the CA is given by $(p, G_1, G_2, G, g_1, (g_2^{(1)}, g_2^{(2)}, \ldots, g_2^{(T)}), e, h_1, h,$ $(A_1, A_2, \ldots, A_T))$ with $A_i = \left({g_2}^{(i)}\right)^a$, where a is the private key of the CA.
2. *Peer registration*:
   (a) To authenticate itself to the CA, a vehicle sends $(ID$, y, $y'^{(1)}$, $y'^{(2)}$, ...,$y'^{(T)})$ to the CA, where $y = h^x$ and $y'^{(i)} = ({g_2}^{(i)})^x$ are to be used in the i-th time interval.
   (b) The vehicle obtains from the CA the proof of linkage between its ID and its root public key (h, h^x).
   (c) The CA verifies that for each $y'^{(i)}$ and $g_2^{(i)}$, the following relation holds.

$$e(y, g_2^{(i)}) = e(h, y'^{(i)}).$$

(d) The CA stores the vehicle identifier in its database and sends the root certificate to the vehicle as (t_g, t_h, z) which remains the same for T time intervals, where

i) $z = \text{Hash}(ID \mid y \mid s_{id} \mid \xi)$, s_{id} is proof of the ownership of ID and the root public key, and $\xi \in Z_p$.

ii) $t_g = {g_1}^{\frac{1}{a+z}}$ and $t_h = (h_1 \cdot h^x)^{\frac{1}{a+z}}$.

(e) The vehicle validates that for all i the following relations hold.

$$e(t_g, A_i \cdot {(g_2^{(i)})}^z) = e(g_1, g_2^{(i)})$$
$$e(t_h, A_i \cdot {(g_2^{(i)})}^z) = e(h_1 \cdot h^x, g_2^{(i)})$$

(f) The vehicle stores the CA's public key $(p,\, G_1,\, G_2,\, G,\, g_1,\, (g_2^{(1)},\, g_2^{(2)},\, \ldots,\, g_2^{(T)}),\, e,\, h_1,\, h,\, (A_1, A_2, \ldots, A_T))$ and its version Ver.

(g) The vehicle pre-computes the time-varying accelerators $({v_1}^{(i)},\, {v_2}^{(i)},\, {v_3}^{(i)})$ for every interval i where ${v_1}^{(i)} = e(g_1 \cdot h_1,\, {g_2}^{(i)})$, ${v_2}^{(i)} = e(t_g \cdot t_h,\, {{g_2}^{(i)}}^{-1})$, ${v_3}^{(i)} = e(h, {g_2}^{(i)})$ are elements of G.

(h) The vehicle stores the root certificate (t_g, t_h, z) and the time-dependent accelerators $({v_1}^{(i)}, {v_2}^{(i)}, {v_3}^{(i)})$ for all $i = 1,\, 2,\, \ldots,\, T$ according to Ver.

3. *Peer-to-peer communication*:

(a) *Signature generation and verification*: These steps are similar to the signature generation and verification steps of PKI+ as mentioned in Section 3.2. The only difference is that here we use time-dependent parameters.

(b) *Pseudonym computation*: The sender can generate a new pseudonym as in PKI+. Now, for the computation of the signature of knowledge proof for the new pseudonym, it uses the time-dependent accelerators.
$s = SKP\{(x_1, x_2, x_3) : e(t, A) = ({v_1}^{(i)})^{x_1} \cdot ({{v_2}^{(i)}}^r)^{x_2} \cdot ({v_3}^{(i)})^{x_3} \wedge 1 = (t^{-1})^{x_3} \cdot {t_y}^{x_1}(Ver)\}$
$\in {Z_p}^4$
During the verification of a pseudonym of the signer by the receiver, the algorithm remains the same as in PKI+, except that now the time-dependent parameters are used.

4. *Tracing*: In order to trace a misbehaving vehicle by its PPK (t, t_y) and PCert (Ver, s), m, s_m during the j-th interval, the CA uses the following relation to obtain the identity of the vehicle:

$$e(t, {y'_i}^{(j)}) = e(t_y, {g_2}^{(j)}).$$

The CA can recover the identity of the vehicle by matching the ${y'_i}^{(j)}$ values of all vehicles, stored in its database.

5. *Revocation*: In PKI+, revocation is achieved by changing parameters of all vehicles. On the contrary, revocation is here achieved by publishing a revocation list (RL). If the CA revokes a vehicle i during the j-th interval, the CA publishes ${y'_i}^{(j)}$ and also all y'_i values for subsequent intervals. Any verifier can run the tracing algorithm used by the CA for a particular interval to check whether a sender appears in the revocation list. Since no y'_i values

for intervals up to $(j-1)$ are published, backward unlinkability is achieved. To sum up, our modified algorithm achieves forward linkable revocation and backward unlinkability at the same time, and is, therefore, a combination of the positive features of both PKI and PKI+.

5 Analysis of the Proposed Algorithm

PKI++ achieves all of the required attributes for V2V communication, namely authentication, data integrity, privacy/anonymity, traceability and revocation. However, its storage and computation requirements are more compared to PKI+ [15]. We assume loose time-synchronization among all vehicles.

1. *Security*: The security of PKI++ is based upon the security of PKI+. Publishing multiple generators ${g_2}^{(i)}$ and corresponding public keys $A_i = ({g_2}^{(i)})^a$ does not reveal any additional information about the CA's secret. Indeed, given any single instance of g_2 and $A = {g_2}^a$, any entity can generate random generators g_2' and the corresponding public keys A' as ${g_2}' = {g_2}^r$ and $A' = A^r$ for any values of r co-prime to p.
2. *Storage and computation*: The vehicle now has to store the CA's public key of increased size and also some time-dependent parameters. The computation at the vehicle's end also increases due to the use of time-dependent parameters in different intervals.

   (a) The size of CA's public key is increased by $2T$ elements each of size $\log\ p$.
   (b) The computation at the vehicle's end also increases, since each vehicle needs to compute the time-dependent accelerators for each interval.
   (c) If the CA provides the vehicles with their respective precomputed accelerators for each interval, then the storage requirement increases by $2T$ elements of size $\log\ p$.

   Storage and computation overheads at the CA's end also increases but that is not a serious problem, since the CA is expected to have enough computational and storage capacity to accommodate the increase.
3. *Revocation list size*: This is no larger than $T \cdot \log\ p \cdot n$, where n is the number of revoked nodes.
4. *Verification time*: The verification time increases linearly with the number of revoked nodes.
5. *Restriction*: Our scheme has one limitation in the sense that any vehicle can exploit the published y' of the revoked vehicle to test the ownership of any other pseudonymous public-key generated by that same revoked vehicle in that particular time interval. Therefore, the duration of an interval cannot be made arbitrarily long, and it should be adjusted with change in the number of malicious vehicles.

6 Performance Comparison

We compare PKI, PKI+ and PKI++ with respect to four metrics.

- *Connectivity Requirements*: Communication with the infrastructure is required to obtain pseudonyms and updated CRLs. We now introduce two important notions in this context.
  - *Window of Vulnerability* can be defined as the time across a network during which a vehicle may accept messages as authentic from revoked nodes.
  - *Window of Outage* can be defined as the time across a network during which one non-revoked vehicle with old parameters cannot communicate with another non-revoked vehicle with updated parameters.

  The Window of Vulnerability is the same for all the three crypto-systems. It depends only on the time elapsed between the latest revocation event and obtaining the latest revocation information. The advantage of any autonomous certificate generation scheme such as PKI+ and PKI++ over PKI is that connectivity to the CA is not required to obtain new pseudonyms. In PKI+, when a vehicle is revoked, the system version gets updated, and new parameters are disseminated by the CA. Therefore, any non-revoked vehicle that has not received updated parameters from the CA after a revocation event, cannot communicate with a non-revoked vehicle which has updated its parameters, and experiences the Window of Outage. Window of Outage applies specifically to PKI+, since PKI and PKI++ follow revocation list based approach to disseminate revocation information. The availability of the infrastructure may be sparse, particularly in the initial phases of deployment of V2V. PKI++ combines the autonomous certification feature of PKI+ and the revocation-list approach of PKI to enjoy the benefits of both of the schemes.
- *Implementation Cost*: From an implementation point of view, the three crypto-systems can be compared as shown in Table 2.

Table 2. Comparison of PKI, PKI+, and PKI++

Protocol	Privacy	Infrastructure connectivity	Cost
PKI (ECDSA)	Pseudonyms	Pseudonyms and CRLs	Per-certificate and per-revocation list
PKI+	Participant generates pseudonyms	Parameter update on each revocation event	Per-revocation event
PKI++	Participant generates pseudonyms	Revocation list	Per-revocation list

- *Storage and Computation Overhead*: It is worthwhile to take a critical look at the storage and computation overheads associated with the RL of PKI++. Each pseudonym may be assumed to be used for one minute. If we take each interval of PKI++ as one hour, this calls for 60 autonomously generated pseudonyms for a vehicle per interval. Finally, we assume that the version of PKI++ changes after 24 hours, that is, $T = 24$. These values are some suggestive choices meant for comparing PKI++ with PKI.

  If n is the number of revoked vehicles in a day, the revocation list contains at most $24n$ values of y'. Since each y' is a member of a group of size having 160–320 bits, the storage requirement for the RL is no more than $960n$ bytes. Even for an unreasonably high value of n like 1000, this is less than 1 Megabyte. In PKI, on the other hand, all pseudonyms to be used in a day by a revoked vehicle need to be stored in the CRL. This calls for a storage of $1440n$ pseudonyms for n revoked vehicles, that is, the RL of PKI++ is significantly smaller in size than the CRL of PKI.

  Now, let us consider the time for search in the RL. In PKI, the CRL may be kept sorted with respect to the pseudonyms. Binary search in that sorted list requires a few tens of comparisons of short strings (few hundred bits), even when the list is quite large (like having a total of 10^9 pseudonyms of all revoked vehicles). As pointed out in [11], one may use probabilistic data structures like Bloom filters for reducing the memory needed for the storage of the CRL. A search in a Bloom filter requires computation of a few hash functions. These hash functions need not be cryptographically strong. So fast hash algorithms can be used. Even cryptographic hash functions like SHA-1 are not too expensive. To sum up, we may expect each search in the CRL to take time no more than a few tens or hundreds of microseconds.

  On the contrary, the search complexity in the RL is significantly enhanced in PKI++. For each search for a pseudonym, one needs to make (at most) $2n$ pairing computations, where n is the number of revoked vehicles. Although this is not an infeasible task, doing it periodically in a vehicle may become impractical except only for small values of n. For example, Beuchat et al. [2] report multi-core implementations of pairing taking time 2–3 milliseconds. For $n = 10$, we then require a running time of about 50 ms for searching a pseudonym in the RL. This is somewhat on the higher side.

  Small values of n are, however, not unrealistic in the context of PKI++, since n stands for the number of vehicles revoked in a single day (as per our choices of the parameters). PKI++ is designed to refresh the system version and the parameters after T time intervals, and use an RL which is valid within the T time intervals only. After this period, the version of PKI++ changes, prohibiting the vehicles revoked earlier from participating further in V2V communications. Indeed, the parameter T can be adjusted so as to keep the maximum value of n below a tolerable limit. During the initial stages of V2V deployment, it is expected that the infrastructure availability would be sparse. But it is also likely that there would be only few revocations. In this case, the value of T can be quite high. As the V2V technology becomes more widely available in vehicles, incidents of revocation would increase. But it is also expected that more communication infrastructure would be deployed over time. In this case, the value of T can be decreased to limit the maximum value of n.

Finally, PKI+ does not use any revocation list and is, therefore, free from these storage and computation overheads.

- *Resistance against certificate stealing attack*: Digital certificates or pseudonyms are not usually stored securely as they are meant to be public. An adversary who steals the certificates from a vehicle before they are used can link multiple messages signed by the vehicle using the stolen pseudonyms. Traditional pseudonym-based PKI stores pseudonyms and certificates issued by the CA. Therefore, if an active adversary steals pseudonyms and its certificates, the privacy and unlinkability of the victim are jeopardized. For autonomous certification schemes such as PKI+ or PKI++, the vehicles can themselves generate pseudonyms and the certificates for those pseudonyms, and so there is no need to pre-compute and store certificates a priori, reducing the threats posed by certificate stealing.

7 Conclusion

Privacy with limited infrastructure support is one of the prime requirements for secure V2V communications. To achieve that, PKI+ has been proposed as an improved alternative to PKI. But user revocation has been shown to be expensive in PKI+ (particularly bad for non-revoked users). In this paper, PKI++ which overcomes many shortcomings with user revocation in PKI+ is proposed. PKI++ combines the autonomous certification feature of PKI+ to reduce CA involvement, and the revocation-list-based approach of PKI to remove the revocation complexity of PKI+. The increased storage overhead, in PKI++, associated with public keys and revocation lists, is tolerable. However, the increased computation overhead associated with searching pseudonyms in revocation lists, particularly when the number of revoked vehicles is large, calls for further research in this area.

References

1. Frederik Armknecht, Andreas Festag, Dirk Westhoff, and Ke Zeng. Cross-layer privacy enhancement and non-repudiation in vehicular communication. In *Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN)*, Bern, Switzerland, March 2007.
2. Jean-Luc Beuchat, Emmanuel López-Trejo, Luis Martínez-Ramos, Shigeo Mitsunari, and Francisco Rodríguez-Henríquez. Multi-core Implementation of the Tate Pairing over Supersingular Elliptic Curves. In *Cryptology and Network Security*, Lecture Notes in Computer Science, pages 413–432. Springer Berlin / Heidelberg, 2009.
3. G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. On the performance of secure vehicular communication systems. *IEEE Transactions on Dependable and Secure Computing*, 8:898 –912, 2011.

4. Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in VANET. In *Vehicular Ad Hoc Networks (VANET'07)*, pages 19–28, September 2007.
5. Jan Camenisch and Anna Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In M. Franklin, editor, *Proceedings of Crypto 2004*, LNCS, pages 56–72. Springer-Verlag, August 2004.
6. Chun-I Fan, Ruei-Hau Hsu, and Chun-Hao Tseng. Pairing-based message authentication scheme with privacy protection in vehicular ad hoc networks. In *International Conference on Mobile Technology, Applications, and Systems*, Mobility '08, pages 82:1–82:7. ACM, 2008.
7. IEEE. IEEE trial-use standard for wireless access in vehicular environments - security services for applications and managemnet messages, July 2006. `http://ieeexplore.ieee.org/xpl/standardstoc.jsp?isnumber=34648`.
8. Pandurang Kamat, Arati Baliga, and Wade Trappe. An identity-based framework for VANETs. In *Third ACM International Workshop on Vehicular Ad Hoc Networks (VANET'06)*, Los Angeles, California, USA, September 2006.
9. Kenneth P. Laberteaux, Yih-Chun Hu, and Jason J. Haas. Security Certificate Revocation List Distribution for VANET. In *Proceedings of ACM Mobicom International Workshop on Vehicular Ad Hoc Networks (VANET)*, pages 88–89. ACM, September 2008.
10. Bryan Parno and Adrian Perrig. Challenges in securing vehicular networks. In *Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.
11. Maxim Raya, Daniel Jungels, Panos Papadimitratos, Iman Aad, and Jean-Pierre Hubaux. Certificate revocation in vehicular networks. In *Tech. report*, 2006.
12. Wei Ren, Kui Ren, Wenjing Lou, and Yanchao Zhang. Efficient User Revocation for Privacy-aware PKI. In *Proceedings of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness ICST*, pages 1–7, 2008.
13. Ahren Studer, Elaine Shi, Fan Bai, and Adrian Perrig. Tacking together Efficient Authentication, Revocation, and Privacy in VANETs. In *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2009)*, 2009.
14. H. Weerasinghe and Huirong Fu. ESAP: Efficient and scalable authentication protocol with conditional privacy for secure vehicular communications. In *GLOBECOM Workshops, 2010 IEEE*, pages 1729 –1734, 2010.
15. Ke Zeng. Pseudonymous PKI for ubiquitous computing. In *Proceedings of EuroPKI*, pages 207–222, 2006.