

# A Comparative Analysis of Anonymous and Non-Anonymous Authorization Architectures for IoT Environments in Cooperative Intelligent Transport Systems

Hannes Salin<sup>a</sup>

Swedish Transport Administration, Department of Information and Communication Technology, Borlänge, Sweden  
fi

**Keywords:** Trust Model, Trust Architecture, Cryptographic Accumulators, Group Signatures, Anonymous Credentials, C-ITS, Authentication, Authorization.


**Abstract:** In this paper, we present a comparative analysis of two distinct authorization architectures with a focus on their applicability in dynamic and cooperative intelligent transportation networks (C-ITS) suitable for low-powered IoT devices. Both architectures leverage accumulators for authorization and secret key storage, while our modification of the original architecture introduces an enhanced privacy feature enabling anonymous device access via a proxy node. This modification results in increased communication complexity, trading off anonymity against increased interaction overhead. We provide a proof of concept implementation with performance experiments, and conclude that the cryptographic computational performance remains relatively unaffected between the two architectures. Our findings suggest a potential for different deployment strategies of these architectures; in urban settings with a dense presence of proxy nodes, but also in sparser regions where privacy is not paramount due to minimal vehicle presence.

## 1 INTRODUCTION

Connected transport infrastructures, also referred to Cooperative Intelligent Transport Systems (C-ITS), consist of both mobile and stationary nodes, where these nodes may be personal devices, vehicles, roadside- or central units. Also, having drones and other types of UAV devices are not excluded from C-ITS environments (Valle et al., 2021). C-ITS in the automotive sector primarily aims to improve communication between vehicles and nearby structures, leading to increased road safety and smoother traffic flow (Zeddini et al., 2022). C-ITS employs established short-range communication methods (such as ETSI ITS G5 (European Telecommunications Standards Institute, 2018)) and will be supplemented by broader communication technologies like 3G, 4G, and 5G. This setup will enable vehicles to interact with other vehicles, traffic lights, road infrastructure, and other types of infrastructure participants using Internet of Things (IoT) or Industrial IoT (IIoT) devices. In essence, the wireless communication among various nodes and ITS stations, along with their associated functions, is termed cooperative V2X communica-

tion. The utilization of the connectedness thus enables a range of informational, alert, and support services, which are set to roll out progressively in the forthcoming innovation stages (C2C-CC, 2023).

In these connected ad-hoc type of environments, the utilization of IoT and IIoT creates fruitful opportunities for novel and more efficient use cases. As an implication, involving more devices and allowing more nodes to interact, the need for secure and scalable trust architectures is crucial (Shahab et al., 2022; Galego and Pascoal, 2022). For the European context, not only must the General Data Protection Regulation be considered (European Commission, 2016) but also ITS-specific safety and security standardizations (European Telecommunications Standards Institute, 2018). However, there is currently no detailed specifications of authorization nor anonymization implementations for C-ITS environments, leaving the industry and academia open to work towards technological harmonization. Previously, the secure accumulator-based architecture AccA was proposed (Salin, 2023a) to address a suitable authorization mechanism for IoT-driven C-ITS environments. Expanding that work, we now develop a stronger and even more secure architecture conceptualized via *anonymous credentials* (Chaum, 1985). For this rea-

<sup>a</sup>  <https://orcid.org/0000-0001-6327-3565>

son, our work addresses two main areas: we propose a secure and anonymized authorization architecture based on cryptographic accumulators for C-ITS purposes, and we provide a performance comparison evaluation and analysis of the original AccA architecture and our new, anonymized architecture.

### 1.1 Anonymous Credentials

Anonymous credentials (AC), was originally introduced by Chaum in 1985 (Chaum, 1985) and later realized as the first fully anonymous scheme in 2001 (Camenisch and Lysyanskaya, 2001). These type of schemes are cryptographic mechanisms that let users validate attributes like their identity or group membership without disclosing personal details. AC schemes can be realized in various ways, where the group signature-based approach seems to be the more efficient and solid approach (Kakvi et al., 2023). This, however, requires a set of participants being able to collectively sign an authorization access.

### 1.2 Problem Statement

Trust management and authorization mechanisms in IoT environments remains a challenge in general (Sharma et al., 2020; Hammi et al., 2022). Considering the context of short-lived ad-hoc vehicle networks where IoT and IIoT devices may be utilized in both moving and stationary nodes, we need to investigate authorization mechanisms that also provides anonymity. This includes finding an efficient and secure authorization architecture that allows for easy implementations. We also consider the case where the accessing nodes, given an authorization access, can retrieve data or services from another node in a completely anonymous way. This is desired since C-ITS environments can be open for location privacy threats, e.g., (Liao et al., 2018) and (Salin, 2023b). We summarize our research goals, addressing these challenges as follows:

- Investigate the extension of an accumulator-based authorization mechanism, providing anonymous access.
- Compare the anonymous and non-anonymous authorization mechanisms using a performance analysis.

### 1.3 Contribution

We provide a comparison between two accumulator-based authorization architectures tailored for C-ITS environments: the first with no anonymity but decentralization of authorization, and a second which is

our modification of the latter with anonymity but relying on a proxy node. We provide a proof of concept implementation of both architectures, coupled with performance experiments. From the analysis we are then able to suggest when the two different architectures are suitable for implementation.

## 2 RELATED WORK

Some research in using cryptographic accumulators have previously been investigated, related to the domain of ad-hoc networking and C-ITS. However, the use-cases are more relevant for other scenarios, e.g., (Zuo et al., 2021; Salin et al., 2021; Förster et al., 2014; Heng et al., 2022). No group signature based accumulator architecture for authorization was found in the current literature. Lauinger et al. proposed an authorization scheme for Internet of Vehicles (IoV), building on cryptographic accumulators and zero-knowledge proofs. However, the proposed architecture needs a managing root authority for the accumulators and is not self-contained in the vehicles (Lauinger et al., 2021). In a recent survey of accumulators, Ren et al., shows that accumulator based schemes exists for anonymous credentials, but these are developed for cloud data sharing and mobile payments (Ren et al., 2022). Also, in the comprehensive study by Khan et al. on IoT-specific authorization schemes, no accumulator-based solutions utilizing anonymous credentials were to be found either (Khan et al., 2022). The main difference of the accumulator-based architectures with other type of technologies is primarily on the chosen underlying cryptographic primitives whereas the accumulator-based architecture uses an interactive protocol to enable the de-centralization of authentication and authorization between nodes.

## 3 PRELIMINARIES

We designate  $\mathcal{H}$  as a secure hash function  $\mathcal{H} : \{1,0\}^* \rightarrow \{1,0\}^n$  for a predetermined  $n$ , which is secure against preimage attacks. The resultant hash digest  $\mathcal{H}(m)$  belongs to a secure group  $G$ . The generator of this secure group  $G$  is denoted as  $g$ . A security parameter  $\lambda$  is a value that determines the key sizes during the initial setup of a particular scheme. A key pair is denoted  $(sk, pk)$  where  $sk$  is the secret key and  $pk$  the corresponding public key. The analyzed architectures use the efficient Boneh-Lynn-Shacham (BLS) short signature scheme (Boneh et al., 2001). This scheme is based on bilinear pairings where a

signature  $\sigma$  of a message  $m$  is then computed by  $\mathcal{H}_g(m)^{sk} = \sigma$  and the signature is verified by checking that  $\hat{e}(\sigma, g) \stackrel{?}{=} \hat{e}(\mathcal{H}(m), pk)$  holds.

### 3.1 System Settings

In this section we describe the system C-ITS setting, which includes multiple IoT nodes and vehicles. Any type of device capable of sending or receiving data using wireless communication channels, are referred to as *objects*, and we denote such object as *OBJ*. Each object have a *hardware security module*, denoted HSM. This is a memory segment of the device's architecture being able to securely store secret keys and generate pseudo random numbers to be used in secure protocols.

A vehicular ad-hoc network (VANET) with multiple connections (at least two objects) is denoted NET. Although our C-ITS system is primarily a VANET, we do not limit the NET to be exclusively such network; any cluster of IoT devices capable of short range communication can be considered. The underlying logical network of the NET is thus bounded for a designated area, e.g. a smart home network, a clustered drone network or a standard VANET. Moreover, in the NET there is at least one *registration authority object* denoted *OBJ<sub>R</sub>*. This object allows other objects to register into the NET and is trusted by the NET holder, e.g. the infrastructure or network segment in the eco-system. Therefore, *OBJ<sub>R</sub>* is considered trusted in this particular setup. However, in contrast to the scenarios found in isolated public key infrastructure (PKI) settings where various functions might be overseen by a centralized authority (CA), in our system architecture, object registration holds a singular focus: it serves exclusively for the registration of the object. This is a crucial distinction because, following this registration process, all subsequent procedures involving authentication and authorization are handled autonomously by the objects themselves, without a need for an overseeing authority, including revocation. This decentralization arises from the capability of these objects to facilitate a range of accessible functionalities within their own individual devices. Thus, instead of relying on a centralized system to manage authentications and authorizations, this task is dispersed amongst the network of objects, each holding its own control and maintaining the network's functionality. This implies a self-sustained ecosystem where IoT devices independently manage the validations, bringing both efficiency and robustness to the system's operations.

Different entities are categorized as *types*. A specific object within this network could establish a stan-

dard trust configuration of certain types that it can engage with and respond to.

We summarize the system setting in the following components:

**OBJ:** an object, manifested as a device with short range communication capabilities and computational power equally strong as IoT or IIoT devices. Each object can store and execute authentication and authorization protocols.

**HSM:** a hardware security module, i.e. a memory area within the device architecture, being able to store secret keys or generate random values.

**RA:** a registration authority object, trusted by the network facilitator and have the ability to register new objects into the network.

**NET:** the overall network where registered objects *OBJ<sub>R</sub>*, *OBJ<sub>1</sub>*, ..., *OBJ<sub>n</sub>* communicate, consisting of the registration object as a minimum. Can be manifested as a stationary road side unit at a certain road- and network segment in a larger C-ITS eco-system.

### 3.2 Threat Model

We consider the following threat model where an adversary  $\mathcal{A}$  has two different capabilities:  $\mathcal{A}$  either passively observe the NET from the outside, or  $\mathcal{A}$  is itself registered and connected to the NET as a normal participant, i.e., the adversary is an object within the given system setting, as described in Sec. 3.1.  $\mathcal{A}$  also have the ability to capture data transfers between any two nodes in the NET. However,  $\mathcal{A}$  does not have computational capacity to steal or manipulate any memory segments nor data within any other objects. It cannot access any secret keys. The remaining participants in the NET are considered honest and non-tampered objects *OBJ<sub>1</sub>*, ..., *OBJ<sub>k</sub>*. The formalized threat model is found in (Salin, 2023a) and defined as two security experiments  $\text{Exp}_{A_1}$  and  $\text{Exp}_{A_2}$ , where the former captures the case where  $\mathcal{A}$  tries to gain access to function  $f_i$  in object *OBJ<sub>k</sub>* but have no registered access, and the latter captures the case where  $\mathcal{A}$  is not even registered in the NET.

## 4 AUTHORIZATION ARCHITECTURE

In this section we introduce a modified AccA architecture for authentication and authorization of in-vehicle services and data, namely AccA – GS. The core part of the architecture is built on the what is referred to as

the *authorization matrix*  $\mathcal{M}$ . It embeds an *authentication vector*  $\mathcal{V}$ . These two are combined and securely stored in the object's HSM as an accumulated element of a secure accumulator  $\mathcal{Z}$ . Due to space constraints, we refer to (Salin, 2023a) for a complete description of the architecture.

#### 4.1 Group Signatures for a Modified Architecture with Anonymity

In this section we will detail an architecture using a standard AC, but adjusted for the previously described scenario of C-ITS authorization. The main goal is to construct a simple alternative to AccA for comparison, in order to analyze the differences of using non-collaborative but non-anonymous access, with collaborative but anonymous access. We modify the AccA architecture to handle group signatures to incorporate an AC approach (Kakvi et al., 2023), which implies a stronger dependency on RA. Therefore, our proposed architecture is the natural modification of AccA into a group signature version we will call AccA – GS. This modification is inspired by the original work proposed by Camenisch and Lysyanskaya, incorporated as an AC system (Camenisch and Lysyanskaya, 2004). The main difference between the two architectures is that AccA is self-contained after object registration with the RA, where AccA – GS relies heavily on the RA as a proxy to ensure privacy.

#### 4.2 Modified Architecture

We propose a modified AccA architecture with the following re-defined sub-protocols:

$\text{Reg}_{\text{RA}}(\text{OBJ}_j, \text{RA})$ : this protocol runs just as in the original AccA architecture. However, after the computation of  $\alpha_j$  and storage of  $\text{pk}_j$ ,  $\text{OBJ}_j$  and RA executes the Join protocol as a second step.

$\text{Reg}_{\text{OBJ}}(\text{OBJ}_1, \text{OBJ}_2, \text{RA})$ : this protocol is adjusted as follows:

1.  $\text{OBJ}_1$  sends a request  $\rho_{\text{register}} = (\sigma_{i,1} = \mathcal{H}(\text{OBJ}_1 || \alpha_1)^{\text{sk}_1}, \alpha_1, f_i, \sigma_{\text{GS}_1})$  to the RA requesting function  $f_i$  in object  $\text{OBJ}_2$ . This includes the group signature  $\sigma_{\text{GS}_1}$  which is computed by  $\text{OBJ}_1$  using the Sign protocol where the message  $m = \sigma_{i,1}$ .
2. The RA sends  $(\sigma_{i,1}, \sigma_{\text{GS}_1})$  to  $\text{OBJ}_2$ .
3.  $\text{OBJ}_2$  run the Verify protocol to verify the signature  $\sigma_{\text{GS}_1}$ .
4.  $\text{OBJ}_2$  creates (or updates) a binary string  $b_{i,1}$  that corresponds access to  $f_i$  for the specific signature  $\sigma_{i,1}$ . Note that  $\text{OBJ}_2$  only knows that it

is a valid signature but not from whom it is created, i.e. the index 1 is not signifying a particular identity.

5.  $\text{OBJ}_2$  accumulates  $\sigma_{i,1}$  into  $\mathcal{Z}$  and in particular  $\mathcal{V}_1$  using the Acc procedure, which in turn output corresponding witnesses  $\omega_1$  and  $w_i$ , respectively. The witness is encrypted as  $w_i^{\text{pk}_{\text{RA}}}$  and  $\omega_1$  is stored locally in  $\text{OBJ}_2$ .
6.  $\text{OBJ}_2$  sends  $w_i^{\text{pk}_{\text{RA}}}$  to RA.
7. RA decrypt  $w_i^{\text{pk}_{\text{RA}}}$  into  $w_i$  and sends  $w_i^{\text{pk}_1}$  to  $\text{OBJ}_1$ .
8.  $\text{OBJ}_1$  receive and decrypt  $w_i^{\text{pk}_1}$  and stores  $w_i$ .

$\text{Auth}(\text{OBJ}_1, \text{OBJ}_2, f_i)$ : this protocol runs between two registered objects where  $\text{OBJ}_1$  seek access to  $f_i$  in  $\text{OBJ}_2$ . The protocol is using the RA as a proxy for keeping the privacy of the original sender of the request, although the signature is not telling the original signer.

1.  $\text{OBJ}_1$  sends  $(\sigma_p, \rho_{\text{access}} = (\alpha_1, \sigma_{i,1}, w_i^{\text{pk}_{\text{RA}}}))$  to RA, where  $\sigma_p$  is a standard BLS-signature over  $\rho_{\text{access}}$ .
2. RA verifies that

$$e(g^{\frac{1}{z_1}}, \alpha_1) \stackrel{?}{=} e(\text{pk}_1, g). \quad (1)$$

3. RA verifies  $\sigma_p$  with the standard BLS signature verification procedure.
4. RA decrypt  $w_i^{\text{pk}_{\text{RA}}}$  into  $w_i$  and sends  $\rho'_{\text{access}} = (\alpha_1, \sigma_{i,1}, w_i^{\text{pk}_2})$  to  $\text{OBJ}_2$ .
5.  $\text{OBJ}_2$  receive  $\rho'_{\text{access}}$  and decrypt  $w_i^{\text{pk}_2}$  into  $w_i$ .
6.  $\text{OBJ}_2$  performs validation of two witness proofs in the accumulators, i.e. verifying  $\omega_1$  for the internal check and  $w_i$  for access to  $f_i$ .
7. If successful, either send back  $f_i^{\text{pk}_{\text{RA}}}$  and use same procedure as in  $\text{Reg}_{\text{OBJ}}$  to proxy the data back to  $\text{OBJ}_1$ , or if it was to trigger a certain function  $f_i$  in the device of  $\text{OBJ}_2$  that will then execute.

$\text{Rev}(\text{OBJ}_1, \text{OBJ}_2, f_i)$ : this protocol revokes access of  $f_i$  in  $\text{OBJ}_2$  for  $\text{OBJ}_1$ . In all its simplicity,  $\text{OBJ}_2$  runs the accumulator procedure Del to remove  $\sigma_{i,1}$ . Similarly, the standard Revoke protocol for the RA is run as in the original scheme of Camenisch and Lysyanskaya.

Note that even if the group signature scheme provides anonymity of the original signer, we cannot have  $\text{OBJ}_1$  directly communicate with  $\text{OBJ}_2$  since it would be obvious for  $\text{OBJ}_2$  who is requesting access. Therefore, the RA function as a proxy and ensure that the anonymous access request is verified by  $\text{OBJ}_2$ . Moreover, since we only use the RA public



<b>Setup procedure:</b> each object initializes their own accumulator (if not already setup), i.e. running $\text{Generate}(\lambda, N)$ , and the RA runs KeyGen.			
<b>Setup keys:</b> each object $OBJ_i$ generates $(i, i)$ .			
<b>Modified Authentication and Authorization protocols using group signatures</b>			
Sub-protocol	$OBJ_j$	RA	$OBJ_k$
Reg <sub>RA</sub>	1. Sends $pk_j = g^{sk_j}$ to RA	2. Receive $pk_j$ . 3. Generate $\alpha_j = (g^{sk_j})^{z_j}$ where $z_j \leftarrow \mathbb{Z}$ 4. Store $(\frac{1}{z_j}, z_j, j)$ 5. Sends $\alpha_j$ to $OBJ_j$ .	
Reg <sub>OBJ</sub>	6. Receive and store $\alpha_j$ 7. <b>Run protocol</b> Join( $OBJ_j, RA$ ) 1. <b>Sends</b> $\rho_{register} = (\sigma_{i,j} = \mathcal{H}(OBJ_j    \alpha_j^{sk_j}, \alpha_j, f_i, \sigma_{GR_j}))$ to RA where $f_i$ is the function to access in $OBJ_k$ .	2. <b>Sends</b> $(\sigma_{i,j}, \sigma_{GS_j})$ to $OBJ_k$ .  7. <b>Decrypt</b> $w_i^{RA}$ into $w_i$ and send $w_i^j$ to $OBJ_j$ .	3. <b>Run protocol</b> Verify( $\sigma_{GS_j}, \sigma_{i,j}$ ). 4. If successful, update $\mathcal{M}$ . 5. Run Acc( $\sigma_{i,j}$ ) to generate $w_i$ . 6. <b>Send</b> $w_i^{RA}$ to RA.
Auth	8. <b>Receive and stores</b> $w_i^j$ . 1. <b>Sends</b> $(\sigma_p, \rho_{access} = (\alpha_1, \sigma_{i,1}, w_i^{RA}))$ to RA	2. <b>Receive</b> $(\sigma_p, \rho_{access})$ and verifies $\sigma_p$ using BLS. 3. <b>Verifies</b> $e(g^{\frac{1}{z_j}}, \alpha_1) \stackrel{?}{=} e(g, \rho_{access})$ . 4. <b>Decrypt</b> $w_i^{RA}$ and send $\rho'_{access} = (\alpha_1, \sigma_{i,1}, w_i^j)$ to $OBJ_k$ .  8. <b>Decrypt</b> $f_i^{RA}$ into $f_i$ and send $f_i^j$ to $OBJ_j$ .	5. <b>Receive</b> $\rho'_{access}$ and decrypt $w_i^j$ . 6. Run accumulator Verify. 7. <b>If successful</b> , send $f_i^{RA}$ to RA.
Rev	9. <b>Receive</b> $f_i^j$ from RA and decrypt.	Run Rev protocol.	

Figure 1: The proposed modified authentication and authorization architecture with group signatures. Additional steps in the protocols are highlighted in bold.

key for anonymous credentials of all registered users there is no scalability issue as in traditional PKI. The modified architecture protocol is shown in Fig. 1.

## 5 SECURITY ANALYSIS

The security analysis for the original AccA scheme is provided by (Salin, 2023a). Therefore, we limit this paper to include a complementary analysis for the AccA – GS scheme, showing correctness and privacy, based on the previous analysis and from the original scheme of Camenisch and Lysyanskaya.

**Theorem 1.** *The verification  $e(g^{\frac{1}{z_j}}, \alpha_j) \stackrel{?}{=} e(g, pk_j)$  in AccA – GS is correct and authenticates  $OBJ_j$ .*

*Proof.* The verification is correct since:

$$e(g^{\frac{1}{z_j}}, \alpha_j) = e(g^{\frac{1}{z_j}}, g^{sk_j z_j}) \quad (2)$$

$$= e(g, g^{sk_j})^{\frac{z_j}{z_j}} = e(g, pk_j). \quad (3)$$

The only party who knows  $\frac{1}{z_j}$  is the RA who also generated that value (and stores it in a HSM internally). Moreover, since the RA also generated  $\alpha_j$  during the setup phase, there is no way to construct an  $\alpha'_j$  which is based on another public key than  $pk_j$ . Finally, since the RA receives a BLS signature  $\sigma_p$  over the request, the only way to forge it is to break the BLS scheme,

namely the security of the privacy is reduced to the security of BLS.  $\square$

**Corollary 1.** *For the modified version Acca – GS the correctness and security provided in the original paper (Camenisch and Lysyanskaya, 2004) implies the privacy of the proposed architecture since we only applied this already secure construction on the already proved secure AccA architecture (Salin, 2023a), under the threat model of security experiments  $\text{Exp}_{A_1}$  and  $\text{Exp}_{A_2}$ .*

We note that in the modified request and response interactions for the REG<sub>OBJ</sub> and Auth sub-protocols, witnesses and functions  $f_i$  are sent encrypted using the public keys of the receiving parties; the exponentiation encryption is secure under the discrete logarithm problem. Moreover, we underscore that the protocol is secure under the defined security experiments  $\text{Exp}_{A_1}$  and  $\text{Exp}_{A_2}$  as defined in (Salin, 2023a), hence we have not analyzed Acca – GS utilizing stronger models or different attack vectors.

## 6 EXPERIMENTS

We implemented the main parts of the AccA and AccA – GS architectures, and ran several tests to evaluate their performances. We used the Python wrapper of the MCL library (Mitsunari, 2019) with a type A curve, BLS12\_381. We also used the same approach

for the hashing to group elements and byte conversion as in (Salin, 2023a), namely the *hashAndMapTo* function in the MCL-library. Another technical detail, that was particularly considered in the AccA – GS architecture, is that the underlying signature scheme by Camenisch and Lysyanskaya recommends a Fiat-Shamir heuristic in the proof signature step (Camenisch and Lysyanskaya, 2004), but for many implementations it is replaced by a hash function (Chen et al., 2021). We used the standard SHA-256 in our experiments, configured as follows from the Hazmat library:

```
c = public_key.encrypt(message,
padding.OAEP(mgf=padding.MGF1
(algorithm=hashes.SHA256()),
algorithm=hashes.SHA256()),
label=None)
```

The first experiments were sub-protocol performance tests that were executed 1000 times, and the average timings are noted in Tab. 1 for the AccA architecture, and Tab. 2 for the AccA – GS architecture. The tests included the computational crypto functions in each sub-protocol. Tests were executed on two different devices, D<sub>1</sub>: an Intel Core i5, 2.7GHz platform, and D<sub>2</sub>: a lightweight single board device with a dual core Cortex-A72, 1.5GHz platform.

Table 1: Computational performance in each object (ms) in the AccA architecture on device D<sub>1</sub> and device D<sub>2</sub>.

Protocol (D <sub>1</sub> )	$OBJ_j$	RA	$OBJ_k$
RegRA	-	0.0901	-
RegOBJ	0.0233	-	1.5101
Auth	-	-	2.7701
Rev	-	-	0.1103
Protocol (D <sub>2</sub> )	$OBJ_j$	RA	$OBJ_k$
RegRA	-	0.1053	-
RegOBJ	0.0275	-	2.1970
Auth	-	-	4.0361
Rev	-	-	0.0932

Table 2: Computational performance in each object (ms) in the AccA – GS architecture on device D<sub>1</sub> and device D<sub>2</sub>.

Protocol (D <sub>1</sub> )	$OBJ_j$	RA	$OBJ_k$
RegRA	0.0013	0.0831	-
RegOBJ	0.1241	0.0211	1.0113
Auth	0.2621	2.7131	1.6761
Rev	-	0.3217	0.0799
Protocol (D <sub>2</sub> )	$OBJ_j$	RA	$OBJ_k$
RegRA	0.0069	0.1012	-
RegOBJ	0.2151	0.0329	1.8280
Auth	0.3800	4.6123	2.8493
Rev	-	0.4601	0.1142

As demonstrated in Tab. 1, the Auth protocol emerges as the most resource-intensive, primarily ow-

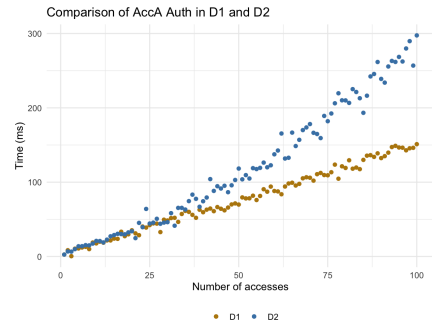


Figure 2: Comparison of running the Auth sub-protocol in AccA up to 100 times on devices D<sub>1</sub> and D<sub>2</sub>, only the computational part for the verifying object in the protocol.

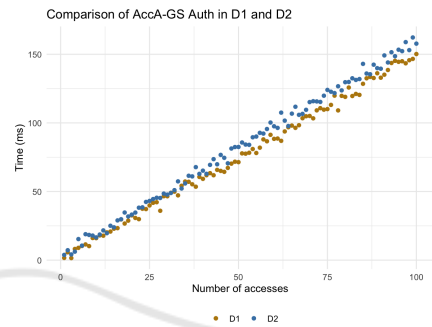


Figure 3: Comparison of running the Auth sub-protocol in AccA – GS up to 100 times on devices D<sub>1</sub> and D<sub>2</sub>, only the computational part for the verifying object in the protocol.

ing to its verification processes. We also did a scalability experiment comparing the two devices' ability to run an increased number of authentications. Since we exclude the communication complexity we did not investigate how much request and response traffic would jam a device if the number of accesses increased. However, from a computational perspective, we measured how each device handled an increase of running the Auth sub-protocol which contains 4 pairing operations. In Fig. 2 we show the difference of execution time (y-axis) when increasing the number of runs of Auth in  $OBJ_k$ , i.e., the device that verifies an authorization of an accessing device  $OBJ_j$ . In D<sub>2</sub> we see that the performance decreases after a while and we hypothesise that it is due to that it may have less optimizations in the execution environment such as efficient caching. Both devices had network traffic turned on in the background in a first round of the experiment, and when turning off all WiFi connections and running the experiment again, we got similar results. Therefore, this was not a significant factor to the computations on neither device.

When running the same experiment for AccA – GS we got as expected more aligned results since the  $OBJ_k$  does not run a sequence of costly pairings in this protocol; instead much of the

computations lies on the RA. This would be more efficient if the RA is a RSU which may have more computational power than a typical low-powered IoT device. The result is shown in Fig. 3. We also note that there are very small shifts in the timing measurements for both architectures, although following a linear pattern. This is also expected since there were running operation system processes in the background on both devices which should affect the measurements slightly; however the scattering along the linearity is on a magnitude of a few  $\frac{1}{100}$ 's of a millisecond, thus negligible.

We also compared the AccA – GS against the performance assessment carried out by Ometov et al., since the original AccA architecture was theoretically compared in the original paper. Simplifying the theoretical comparison, an Intel Edison, 500 MHz Dual-Core device would execute a pairing verification and a curve point multiplication in 580 ms and 0.1 ms respectively. For the AccA – GS architecture the computation does not require any pairings on the  $OBJ_k$  side, but instead the decryption of a witness and a accumulator verification. A theoretical conversion of these operations on a Intel Edison device gives 0.4 ms. However, the RA node will have a similar computational performance as the  $OBJ_k$  in the AccA architecture since it includes a BLS verification which is a pairing verification, and an additional pairing verification.

## 6.1 Analysis

Although not run as an experiment, we conclude that the communication complexity for each architecture differs as detailed in Tab. 3. The AccA – GS has in total 3 more send/receive operations. Given the computational complexity found in tables 1 and 2 we note that there is a generally a larger factor between  $D_1$  and  $D_2$  in the sub protocols where pairings are included, e.g., the Auth protocol. This is noted in both architectures and we hypothesise that it could be related to how the MCL library runs on different (computer) architectures. However, in  $REG_{RA}$  where the sub-protocol need to generate random numbers the differences are not too big. This should be natural since the protocol does not exhaust the device entropy by generating only a few values during the protocol execution.

Now, given the computational complexity and the theoretical communication complexity we conclude that applying the group signature based architecture is not significantly more expensive, given the assumption that the wireless channels in the NET between the devices is efficient. Having 3 more interaction steps is

Table 3: Communication complexity in terms of interactions for each architecture, we denote one sending/receiving operation as one  $T$ .

Protocol	AccA	AccA – GS
$Reg_{RA}$	2T	3T
$Reg_{OBJ}$	4T	4T
Auth	2T	4T

not optimal, and also relying on a third party for every request is not optimal in the sense of not being able to access a device if the RA is down. This could happen due to DDoS attacks or similar events. For that reason, if the NET environment does not require full privacy of the objects' identities, the original AccA architecture is more suitable.

## 7 CONCLUSION

Our proposed modified version of AccA have an additional privacy property where each access to a device is made anonymously but require a proxy node, i.e., the RA. Thus, we trade anonymity for an increased communication complexity. Our experiments shows that the cryptographic computational performance between the protocols does not differ significantly, but instead the additional interactions in AccA – GS gives the architecture a slower overall performance as expected. Depending on the wireless technology used in the NET environments, infrastructure and vehicle stakeholders may need to balance anonymity with the reliance on proxy nodes, where urban areas with dense proxy nodes can support collaborative setups, while in sparse regions, deploying the AccA architecture may be more advantageous.

## REFERENCES

- Boneh, D., Lynn, B., and Shacham, H. (2001). Short signatures from the weil pairing. In *Advances in Cryptology—ASIACRYPT '01, LNCS*, pages 514–532. Springer.
- C2C-CC (2023). CAR 2 CAR Communication Consortium. <https://www.car-2-car.org/about-c-its>. Accessed: 2023-08-23.
- Camenisch, J. and Lysyanskaya, A. (2001). An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Pfizmann, B., editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 93–118, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Camenisch, J. and Lysyanskaya, A. (2004). Signature schemes and anonymous credentials from bilinear maps. In Franklin, M., editor, *Advances in Cryptol-*

- ogy – *CRYPTO 2004*, pages 56–72, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Chaum, D. (1985). Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044.
- Chen, Y., Lombardi, A., Ma, F., and Quach, W. (2021). Does fiat-shamir require a cryptographic hash function? In Malkin, T. and Peikert, C., editors, *Advances in Cryptology – CRYPTO 2021*, pages 334–363, Cham. Springer International Publishing.
- European Commission (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance).
- European Telecommunications Standards Institute (2018). ETSI TS 123 501 V15.3.0: 5G; System Architecture for the 5G System. <https://www.etsi.org/standards>. Accessed: 2023-08-25.
- Förster, D., Kargl, F., and Löhr, H. (2014). Puca: A pseudonym scheme with user-controlled anonymity for vehicular ad-hoc networks (vanet). In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 25–32.
- Galego, N. M. C. and Pascoal, R. M. (2022). Cybersecurity in smart cities: Technology and data security in intelligent transport systems. In Mesquita, A., Abreu, A., and Carvalho, J. V., editors, *Perspectives and Trends in Education and Technology*, pages 17–33, Singapore. Springer Singapore.
- Hammi, B., Monteuiis, J.-P., and Petit, J. (2022). Pkis in c-its: Security functions, architectures and projects: A survey. *Vehicular Communications*, 38:100531.
- Heng, X., Qin, S., Xiao, Y., Wang, J., Tao, Y., and Zhang, R. (2022). A strong secure v2i authentication scheme from pki and accumulator. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 98–103.
- Kakvi, S. A., Martin, K. M., Putman, C., and Quaglia, E. A. (2023). Sok: Anonymous credentials. In Günther, F. and Hesse, J., editors, *Security Standardisation Research*, pages 129–151, Cham. Springer Nature Switzerland.
- Khan, A., Ahmad, A., Ahmed, M., Sessa, J., and Anisetti, M. (2022). Authorization schemes for internet of things: requirements, weaknesses, future challenges and trends. *Complex & Intelligent Systems*, 8(5):3919–3941.
- Lauinger, J., Ernstberger, J., Regnath, E., Hamad, M., and Steinhorst, S. (2021). A-poa: Anonymous proof of authorization for decentralized identity management. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9.
- Liao, D., Li, H., Sun, G., Zhang, M., and Chang, V. (2018). Location and trajectory privacy preservation in 5g-enabled vehicle social network services. *Journal of Network and Computer Applications*, 110:108–118.
- Mitsunari, S. (2019). "MCL cryptolibrary". <https://github.com/herumi/mcl>.
- Ren, Y., Liu, X., Wu, Q., Wang, L., Zhang, W., et al. (2022). Cryptographic accumulator and its application: A survey. *Security and Communication Networks*, 2022.
- Salin, H. (2023a). Acca: A decentralized and accumulator-based authentication and authorization architecture for autonomous iot in connected infrastructures. In Wills, G. B., Buttyán, L., Kacsuk, P., and Chang, V., editors, *Proceedings of the 8th International Conference on Internet of Things, Big Data and Security, IoTBDs 2023, Prague, Czech Republic, April 21-23, 2023*, pages 170–177. SCITEPRESS.
- Salin, H. (2023b). Pseudonym swapping with secure accumulators and double diffie-hellman rounds in cooperative intelligent transport systems. In Kallel, S., Jmaiel, M., Zulkernine, M., Hadj Kacem, A., Cuppens, F., and Cuppens, N., editors, *Risks and Security of Internet and Systems*, pages 223–238, Cham. Springer Nature Switzerland.
- Salin, H., Fokin, D., and Johansson, A. (2021). Authenticated multi-proxy accumulation schemes for delegated membership proofs. In Luo, B., Mosbah, M., Cuppens, F., Othmane, L. B., Cuppens, N., and Kallel, S., editors, *Risks and Security of Internet and Systems - 16th International Conference, CRIStIS 2021, Virtual Event, Ames, USA, November 12-13, 2021, Revised Selected Papers*, volume 13204 of *Lecture Notes in Computer Science*, pages 162–171. Springer.
- Shahab, S., Agarwal, P., Mufti, T., and Obaid, A. J. (2022). Siot (social internet of things): A review. In Fong, S., Dey, N., and Joshi, A., editors, *ICT Analysis and Applications*, pages 289–297, Singapore. Springer Nature Singapore.
- Sharma, A., Pilli, E. S., Mazumdar, A. P., and Gera, P. (2020). Towards trustworthy internet of things: A survey on trust management applications and schemes. *Computer Communications*, 160:475–493.
- Valle, F., Cooney, M., Mikhaylov, K., and Vinel, A. (2021). The integration of uavs to the c-its stack. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, pages 1–6.
- Zeddini, B., Maachaoui, M., and Inedjaren, Y. (2022). Security threats in intelligent transportation systems and their risk levels. *Risks*, 10(5).
- Zuo, X., Li, L., Luo, S., Peng, H., Yang, Y., and Gong, L. (2021). Privacy-preserving verifiable graph intersection scheme with cryptographic accumulators in social networks. *IEEE Internet of Things Journal*, 8(6):4590–4603.