An Accountable Privacy-Preserving Scheme for Public Information Sharing systems

Youcef Imine, Ahmed Lounis, Abdelmadjid Bouabdallah

Version of Record: https://www.sciencedirect.com/science/article/pii/S0167404820300717
Manuscript_5d9ef61ea19843be11124e2c3b2bfcfc

An Accountable Privacy-Preserving Scheme for Public Information Sharing systems

Youcef Imine, Ahmed Lounis, Abdelmadjid Bouabdallah
{imineyou,lounisah,bouabdal}@utc.fr
Sorbonne Universités, Université de Technologie de Compiègne, CNRS
HEUDIASYC UMR 7253 CS 60319 60203 Compiègne Cedex France

***Abstract*—Due to the emergence of data externalization technologies, as cloud and fog computing, setting up public information-sharing applications has become much easier. Yet, many concerns related to information security need to be addressed. While sharing information, privacy is without any doubt one of the major concerns for all users. Several proposals in the literature treated privacy issues using existing anonymization techniques, but few of them considered accountability service. Whereas, when security systems do not adopt accountability mechanisms, full anonymity may encourage users to act maliciously.**

In this paper, we propose a novel accountable privacy-preserving solution for public information sharing in data externalization platforms. Based on signatures, our scheme allows externalization servers to authenticate any user in the system without violating its privacy. In case of misbehavior, our solution allows to trace malicious users thanks to an authority. Moreover, our solution ensures privacy-preserving and accountability services in a completely distributed manner, without a permanent resort to the authority. Finally, we show through experimentation that our solution outperforms existing accountable privacy-preserving solutions.

***Index Terms*—Cloud computing, fog computing, security, privacy, accountability.**

I. Introduction

Recently, many applications such as internet of things (IoT), smart cities, autonomous cars, etc. have emerged. It is predicted that these emerging applications will share a huge amount of information. According to Cisco [2], the Annual global IP traffic will reach 3.3 Zettabytes by 2021. Therefore, including data externalization platforms to handle this massive amount of shared information is inevitable.

Cloud-computing technology has shown its efficiency regarding data processing, high computational power, and data storage and management tasks. Therefore, it may be an important player in data supply/demand equation that the world is about to face in the coming years. However, cloud computing has a centralized operating mode and may be less suitable for real time applications since it endures more latency. Consequently, a new paradigm called fog computing has appeared recently to overcome these limitations [5] [26].

Fog computing paradigm aims to extend cloud services to the edge of the network while ensuring the interaction with the cloud. Therefore, computation, communication, storage and control operations are performed closer to the end user by pooling network's local resources. Thus, it enables computation-intensive applications at the resource-limited mobile devices. Moreover, fog computing paradigm promises a dramatic reduction in latency and mobile energy consumption, tackling the key challenges for materializing 5G vision [17], implementing vehicular network applications [12], etc.

Nevertheless, due to sharing information through cloud or fog servers, privacy becomes a serious concern for users. Indeed, data externalization in outsourcing servers might expose users' personal information to leakage threat. It is true that the threat of personal information leakage can be solved using cryptography, but preserving privacy is not limited to exposing users' identities or some of their private information to the public. It also concerns the detection of users' behavior pattern, activity tracking, interests and preference detection, etc. In fact, selling this kind of information to companies, interested in targeted advertising for example, may be much more useful for service providers than revealing users' identity.

To deal with these issues, data owners usually tend to anonymization techniques such as k-anonymity [30], l-diversity [16], t-closeness[13], group signatures [4], etc., which avoid to link data to its owner. However, in some cases such as public information-sharing applications, anonymization has a crucial drawback, which is the lack of accountability. Indeed, users could misuse the system anonymity feature and start sharing false information, assault other users, etc. Consequently, it is very hard to trace the origin of misbehaviors, and thus, malicious users cannot be punished for their actions.

It is clear that full anonymity without any accountability mechanism can be a serious issue in public information-sharing applications.

Therefore, the challenging problem can be stated as follows: given a network of communicating entities that share public information in cloud or fog architecture, *how can we preserve the communicating entities' privacy? Besides privacy-preserving service, how can we ensure that one member of the network, such as law authority, could trace any malicious entity, in case of abuse or anomaly detection?*

Several solutions in the literature have addressed privacy-preserving in information sharing applications, but few of them considered accountability along with privacy-preserving. Group-based solutions and pseudonym-based solutions are the most recurrent contributions that ensure both privacy and accountability features in the literature [10].

To correctly operate, group-based solutions [9], [15], [25]

require the establishment of groups where one member of each group is set as a manager. However, group structuring is not always practical or easy to set, especially in applications with high mobility such as VANET. Moreover, in group-based solutions, all information need to pass by the group manager to be verified and then signed with the group signature before being shared. It is true that this mechanism ensures accountability and anonymity, but it makes the group manager a single point of failure for all group members. Besides, this mechanism does not scale in the case where the group members are numerous.

On the other hand, in pseudonym-based solutions [11], [28], [29], users need to frequently contact the authority and change their pseudonym certificate to avoid tracking. However, this mechanism consumes the bandwidth and reduces the autonomy of the system.

II. Our contribution

In this paper, we propose a novel privacy-preserving solution with accountability service for public information sharing applications.

In our solution, communicating entities perform a first registration with the authority, which provides access credentials to each registered entity. These credentials will allow the authority to trace any entity in the network. We note that in our solution the authority is the only entity that is able to trace other communicating entities. Moreover, unlike most of existing solutions that rely on group signatures, the authority in our scheme is used only to register users and to trace misbehaving ones when needed. It does not intervene at any moment in the information sharing process.

To allow accountability in our information-sharing model, each information needs to be signed by its owner. Moreover, an application set up in the externalization server verifies the signatures, and checks out that the authority is able to trace the origin of the information without breaching users' privacy.

In order to fulfill accountability requirement while preserving the privacy of communicating entities, we propose to randomize the signatures provided with the public information. Randomizing the signature is an efficient manner to preserve entities' privacy. Indeed, if an entity submits a new secure random signature at each information-sharing event, externalization servers cannot trace the origin of the information. Nevertheless, since we also need to ensure accountability service in our information-sharing model, we propose to randomize the credentials provided during the registration phase. In fact, randomizing these credentials ensures the privacy-preserving feature, but it also allows the externalization server to find out whether the authority could trace the shared information or not, without violating the information owner's privacy.

What sets us apart from existing solutions are the following points:

- We propose a privacy-preserving scheme with accountability feature that operates at any data sharing architecture and does not require any group-based structure.
- We propose a solution that does not rely on any third party during the information sharing process. Indeed, due to our novel accountable and privacy-preserving method, we ensure the same advantages known in solution that rely on third parties while overcoming their limitations.
- We propose an accountability mechanism that is efficient, lightweight, scalable and does not require any cooperation between any entities in the architecture.

The remaining of the paper is organized as follows. In section III, we present the related works. In section IV, we give backgrounds on Shamir's secret sharing scheme, Schnorr signature scheme and bilinear maps. Next, we introduce our achitecture and the threat model in section V. After that, we present our solution in section VI. Then, we present our security analysis in section VII. We provide an application use case and evaluate the performance of our solution in section VIII. Finally, we conclude in section IX.

III. Related work

There have been several proposals which addressed the privacy issue in the literature. Liu et al. [14] proposed an anonymous payment system with privacy protection support. Their work provides the mechanisms to enhance location privacy of electric vehicles. In [6], de Fuentes et al. introduced PRACIS, a scheme that provides privacy-preserving data forwarding and aggregation for cybersecurity information sharing in the network. Nicanfar et al. [18] proposed a robust privacy-preserving authentication scheme for communication between the electric vehicles and power stations. Rottondi et al. proposed a security infrastructure for privacy-friendly vehicle to grid (V2G) interactions [20] [21]. Gope [7] proposed a privacy-preserving security architecture with a cooperative device-to-device communication support that operates in fog computing model. These previous proposals preserve privacy, but they did not provide any accountability service that allows to identify misbehaving entities.

In [31], the authors proposed an accountable privacy-preserving communication scheme in smart grids. In this scheme, there are three main components: the local aggregators (LAG), the central aggregators (CAG) and the electric vehicles. The vehicles use pseudonyms to hide their private information nearby the local aggregator. However, before formulating any request to the LAG, the vehicles need to contact the CAG in order to get its signature. Aslam et al. [3] proposed a distributed certificate architecture for VANETs. Each vehicle in this scheme has a temporary pseudonym that is valid in a specific area during a specific period. The vehicles can get these pseudonyms from components known as payment providers. However, the vehicles use the same pseudonym in a specific area, thus, they can easily be traced in this area. Moreover, since the payment providers generate pseudonyms, they will be able to trace the vehicles and violate their privacy. Sucasas et al. [28] proposed a pseudonym-based privacy-preserving authentication for Vehicular Ad-Hoc Networks (VANET). In this scheme, a trusted authority issues credentials to the vehicles, and each vehicle is able to generate a number n (where n is a system parameter) of pseudonyms by its own during a specific time slot. Compared

to existing pseudonym-based solutions, this solution reduces the frequency of requests to the trusted authority during the same time slot. However, as it was stated by the same author in [27], it is possible to link the pseudonyms generated with the same credential in different time slots. Guan et al. [8] proposed a device-oriented anonymous privacy-preserving scheme with authentication in fog-enhanced IoT system. The anonymity of the devices is preserved by using pseudonym certificates. However, at each information sharing event, the devices present the same pseudonym certificate to fog servers, so it can easily be traced. Salem et al. [22] proposed a non-interactive authentication scheme providing privacy among drivers in Vehicle-to-Vehicle (V2V) Networks. In this solution, drivers are assembled in V2V communication groups. Each driver gets a pair of keys (public and private) from a trusted third party (TTP). Group members could frequently change their own set of public keys, and thus they ensure their privacy. Note that group members generate the new set of public keys without requiring a control from the TTP. However, to trace the drivers in case of misbehavior, the TTP need to check each private key stored in its database until it finds a match with the malicious driver's public key. This task is time-consuming, especially in large scale applications.

He et al. [9] proposed an accountable, privacy-preserving authentication framework for wireless access networks. This scheme considers an architecture composed of a network operator; a group manager at the head of each user group; and finally a set of access points allowing the access to the network. Users can anonymously be authenticated in access points and can be traced through a cooperation between the network operator and the group managers. However, and as it has been stated in [9], this authentication framework is only applicable to group-based architectures.

Liu et al. [15] proposed Mona, a multi-owner data sharing solution for dynamic groups in the Cloud. Both anonymity and accountability are well supported in this scheme. However, as long as the group manager did not verify the data signature, the cloud cannot make it available for group members. Jian et al. [25] proposed an anonymous and accountable group data sharing and storage scheme in the cloud, which is similar to Mona. In this scheme, a group manager defines a group signature that is used to achieve anonymity. On the other hand, group members need to register with the group manager and receive a secret key. When a user wants to share data into the cloud, he first signs the data using its secret key and sends it to the group manager. The group manager verifies the signature and then replaces it with the group signature. Finally, the data will be uploaded to the Cloud. This scheme ensures both anonymity and accountability, but the group members need to pass through the group manager at any data-sharing event.

IV. Background

In this section, we present some mathematical notions and security models that we are going to use in our accountable privacy-preserving scheme.

A. Bilinear Maps

Let G_0 and G_1 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_0 and e be a bilinear map,

$$e : G_0 \times G_0 \rightarrow G_1$$

The bilinear map e has the following properties:

Bilinearity: for all $u, v \in G_0$ and $a, b \in Z_p^*$, we have:

$$e'(u^a, v^b) = e'(u, v)^{ab}$$

Non-degeneracy: $e'(g, g) \neq 1$

We say that G_0 is a bilinear group if the group operations in G_0 along with the bilinear map $e : G_0 \times G_0 \rightarrow G_1$ are efficiently computable.

Notice that the map e' is symmetric since

$$e'(g^a, g^b) = e'(g, g)^{ab} = e'(g^b, g^a)$$

B. Review on Shamir's secret sharing scheme

In cryptography, secret sharing refers to a method for distributing a secret amongst a group of participants. In this method, each participant possesses a part of the secret which can be reconstructed only if enough parts are combined together. Otherwise, individual parts are of no use on their own.

Based on the fact that the collection of at least k different points can reconstruct a polynomial of degree $k-1$, Shamir [24] suggested to consider a polynomial $q(x)$ of degree $k-1$, in which a_0 represents the secret S.

The secrect S is devided into pieces $(x_i, S_i = q(x_i))$ using the polynomial q:

$$q(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{k-1} x^{k-1}$$

$(1, S_1 = q(1)), (2, S_2 = q(2))(k, S_k = q(k))$ are called shares in Shamir's scheme.

The polynomial q can be reconstructed using Lagrange interpolation as:

$$q(x) = \sum_{i=1}^{k} S_i \times \prod_{j=1, j \neq i}^{k} \frac{x - x_j}{x_i - x_j}$$

Consequently, the secret S can be calculated as $S = q(0)$

C. Schnorr signature scheme

Let G be a group of prime order q, with generator g and in which the discrete logarithm problem is assumed to be hard.

Let Z_q^* be a multiplicative finite field of prime order q. Let $H()$ denotes a collision resistant hash function.

Assume that a signer S has a private key x with its corresponding public key $y = g^x$. To sign a message m, S chooses a random number $k \in Z_q^*$ and computes $r = g^k, s = k - x.H(m, r)$. Then, the tuple (m, r, s) becomes a valid signed message.

The validity of signature is verified by $g^s.y^e = h(m, r)$. Schnorr signature [23] has been proven to be secure under the random oracle model in [19]; where the authors have shown that existential forgery under the adaptive chosen message attack is equivalent to the discrete logarithm problem.

V. System and Threat models

A. System model

In this paper, we consider a system model composed of three different types of components presented as follows:

- **Communicating entities** such as connected vehicles, connected objects or any entity interested in sharing public information in the network. These components are completely untrustworthy.
- **Externalization servers** such as fog servers or Cloud, responsible for information sharing. In terms of security, these components are not trusted. Therefore, they aim to violate communicating entities' privacy. However, it is assumed that they correctly execute the verification tasks described in section VI of our protocol.
- **Registration authority** responsible for the management of security parameters in the network and the detection of misbehaviors. In terms of security, this component is trusted.

Figure 1 illustrates our system model.

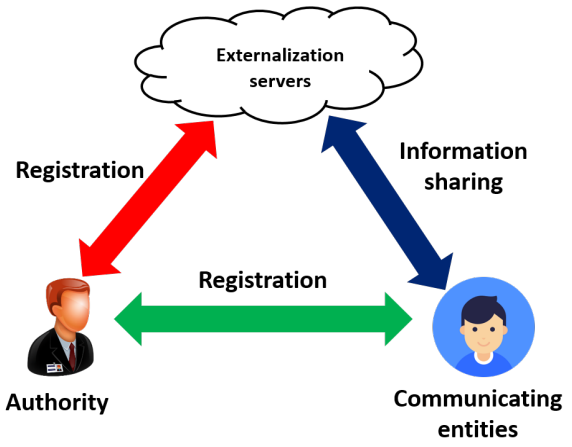

Fig. 1. The system architecture of our scheme

B. Threat model

In our protocol, we distinguish two different adversarial models where each model reflects a specific situation defined as follows:

C. The case where the externalization server aims to trace an entity

let Adv be a polynomial time adversary who interacts with a signature oracle. Adv can submit as much arbitrary tokens as he wants to the oracle. For each reception, the oracle randomizes the token using a secure pseudo-random function (PRF) and sends it back to the adversary. Finally, the adversary outputs an arbitrary message m to the oracle. The oracle chooses three random values a, b and R from Z_p^* and sends back a randomized token $T = ((g^a)^R, (g^b)^R, g^R)$ along with the signature of the message m.

Adversary Adv wins the security game if he can compute the values g^a or g^b given the randomized token T.

D. The case where an entity aims to forge the anonymization token

let Adv be a polynomial time adversary which interacts with a signature oracle. Thus, Adv submits arbitrary messages m_i to the oracle. The oracle provides the signature of these messages along with valid anonymization tokens. Finally, the adversary outputs a message m that has never been submitted to the oracle along with its signature and anonymization token. Adversary Adv wins the security game if the anonymization token along with message m signature are valid.

VI. Our solution

In this section, we present our proposed solution which ensures accountable privacy-preserving information sharing in data outsourcing architectures.

A. Our construction basis

Privacy preserving and Accountability features in our solution are two sides of the same coin. Indeed, in one hand, the users could share their data in a completely anonymous manner without being known neither by the externalization servers, nor any other regular member, within the information-sharing group. Therefore, users cannot be tracked in their eventual information-sharing activities.

On the other hand, and despite the fact that users' signatures are anonymous in our scheme, externalization servers are able to find out whether the user is allowed to share information in the sharing group or not. Moreover, if the system detects any anomaly in the sharing group, our solution ensures that the trusted authority will trace the origin of any shared information.

Our construction is based on the following idea:

Given two polynomials P_1 and P_2 defined as follows:

$$P_1(x) = R_1 x + S$$

$$P_2(x) = R_2 x + S$$

Where R_1, R_2 are random values in Z_p^* and $S \in Z_p^*$ is a common random value used in both polynomials. If we consider two random values x_1 and $x_2 \in Z_p^*$, the points $P_1(x_1), P_2(x_1)$ and $P_1(x_2), P_2(x_2)$ will result different random values. However, as it is shown in equations 1 and 2, even if we use two different polynomials to generate points $(P_1(x_1), P_2(x_1)$ for example), computing the polynomial interpolation at $x = 0$ will always result the same value S, given the same values (x_1, x_2).

$$P_1(0) = \sum_{i=1}^{2} P_1(x_i) \times L_i = \sum_{i=1}^{2} P_1(x_i) \times \prod_{j=1}^{2, j \neq i} \frac{-x_j}{x_i - x_j} = S \quad (1)$$

$$P_2(0) = \sum_{i=1}^{2} P_2(x_i) \times L_i = \sum_{i=1}^{2} P_2(x_i) \times \prod_{j=1}^{2, j \neq i} \frac{-x_j}{x_i - x_j} = S \quad (2)$$

Therefore, we conclude that even with two different polynomials as defined above, we can always find the same secret S if we use the same x_i values to generate points, and then we compute polynomial interpolation at $x = 0$. In our solution, we provide the externalization servers with constant values (computed using x_1 and x_2). On the other hand the users submit points generated using x_1 and x_2 but through a new random polynomial at each information sharing event. As it was discussed above, using different polynomials will result different points. However, if the externalization server performs polynomial interpolation (at $x = 0$) using its constant shares (computed based on x_1 and x_2) and the random points (computed based on the same values as well), it will result the same secret.

Submitting new points at each sharing event will preserve the privacy of the user, since the externalization server cannot trace the user in that case. However, it will allow the externalization server to verify that the user is a valid group member, if the polynomial interpolation results the group key S.

B. Overview

In what follows, we present an overview of our protocol that is composed of the following phases:

- *Setup phase*.
- *Externalization servers registration phase*.
- *Users registration phase*.
- *Information sharing phase*.
- *Tracking phase*.

Our protocol starts with a setup phase in which a registration authority defines a master key and a set of other security parameters. After that, each externalization server must perform a first registration to get the verification parameters that allow them to authenticate any communicating entities in the architecture, in a completely anonymous manner. Similarly, to externalization servers, the communicating entities perform a first registration with the authority. The aim of that phase is to provide access credentials that are going to authenticate the entities during the information-sharing phase. Note that during the users' registration phase, the credentials are sent through a secure communicating channel.

At each information-sharing event, the entity randomizes the credentials provided by the authority and use them to sign the shared information. Randomizing the credential provided by the authority is an efficient manner to preserve entities' privacy. Indeed, sharing information signed by a securely randomized signature makes it very hard for the externalization server to trace its origins. Nevertheless, even with an anonymous signature, the server uses the parameters (delivered by the authority during its registration) to verify whether the information is shared from an authentic source or not. Note that in the verification phase, the server tries to extract the group key from the random credentials submitted with the shared information based on the idea presented in sub-section VI-A. Finally, our protocol provides a tracking mechanism that is used when a misbehavior is detected in the system. In that case, the authority uses its master key and the credentials submitted with the shared information to disclose the anonymity from the randomized signature.

Table I summarizes the main notations used to describe our protocol.

Notation	Description
Z_p^*	A finite field of prime order p.
G_1, G_2	two multiplicative cyclic groups.
g_1, g_2	Group G_1 and G_2 generators respectively.
$H(*)$	Hash Function.
$P_i(x)$	Polynomial of degree one.
S'	The authority's master key.
P	The authority's public key.
(L_1, L_2)	Complementary shares used to compute users' credentials and the verification parameters.
CE_{s_i}	Communicating entity i's share.
T_j	User j's trace stored in the users registry along with j's identity.
S_j	User j's identifier.
P_{ES_i}	Externalization server i's public key.
S_{Es_i}	Externalization server i's identifier.
(Y_1, Y_2)	Randomized credentials used to authenticate the users.
(e, s)	Anonymous digital signature.
D	The shared data.
$e'(*)$	A bilinear map.

TABLE I
TABLE OF NOTATIONS

C. Our proposed protocol

In what follows, we describe the main phases of our accountable privacy-preserving scheme.

1) Setup phase: in this phase, the authority sets up the system parameters that are going to be used in the eventual registration, authentication and tracking processes.

During the setup phase, the authority executes the following tasks:

- Define a finite field Z_p^* of a prime order p.
- Define two cyclic group G_1 and G_2 of prime order p_1 and p_2.
- Define g_1 and g_2 as group generators for G_1 and G_2 respectively.
- Define a bilinear map $e' : G_1 \times G_2 \rightarrow G_T$
- Choose a random master key $S' \in Z_p^*$ and compute the group public key as: $P = e'(g_1, g_2)^{S'}$
- Choose two random values $x_1, x_2 \in Z_p^*$.
- Compute $L_1 = \frac{-x_2}{x_1 - x_2}$ and $L_2 = \frac{-x_1}{x_2 - x_1}$
- Choose a random $K \in Z_p^*$ and compute the following values:

$$T_1 = g_2^{\frac{L_1 \times L_2}{K}}, T_2 = g_2^{L_1}, T_3 = g_2^{S'} \quad (3)$$

$$T_4 = g_1^{\frac{K \times S'}{L_1}}, T_5 = g_2^{\frac{-1}{K \times x_1}}, T_6 = g_2^{\frac{1}{K \times x_2}} \quad (4)$$

- Create the users registry in which the authority will store the identity of any registered entity in the network. We can see the users' registry as a Hash table that maps a given key to a value.

2) Externalization servers registration phase: in order to be able to authenticate any communicating entity in the architecture, externalization servers must request the verification parameters from the authority.

For each request coming from an externalization server, the authority executes the following tasks:

- Generate a random and unique identifier S_{Es_i} for the externalization server Es_i.
- Send $(T_1^{S_{Es_i}}, T_2^{S_{Es_i}}, T_3^{S_{Es_i}}, T_4, T_5, T_6, P^{S_{Es_i}}, P^{S_{Es_i} \times L_1})$ to the externalization server Es_i.

3) Users registration phase: each communicating entity which wants to join the information-sharing group must perform a registration with the authority. First, the entity sends its digital certificate or any information that proves its identity. Once the authority verifies the entity's identity, it performs the following operations:

- Generate a unique and random value $S_j \in Z_p^*$ specified for entity CE_j.
- Compute entity CE_j's trace $T_j = g_1^{S_j}$
- Store the trace T_j and entity CE_j's identity in the users registry.
- Compute the entity CE_j's share

$$CEs_j = (g_1^{\frac{S'x_1}{S_j} + \frac{S'}{L_1}}, g_1^{\frac{KS'x_2}{S_j L_1}}, A = \frac{S'}{S_j L_2}, B = \frac{S'}{S_j L_1})$$

- Send CEs_j to entity CE_j through a secure communicating channel.

4) Information sharing phase: in our solution, when an entity decides to share information into the externalization servers, it needs to provide two main pieces of information. The first piece of information is the digital signature while the second piece represents the anonymization token. This token proves that the entity is a valid group member without divulging its identity. In addition, the anonymization token links the entity to the signature provided with the shared information. In other words, it proves that the entity who signed the data is the same that provided the token. Beside the entity's anonymity and authenticity features that the token ensures, it allows on the other hand the registration authority to trace communicating entities in the case of any detected misbehavior. The information sharing process in our solution works as follows:

- Choose a random value $R' \in Z_p^*$.
- Request the externalization server's public parameter $P_{Es_i} = T_2^{S_{Es_i}} = g_2^{S_{Es_i} \times L_1}$.
- Using the shares provided by the authority and the R' value, generate the anonymization token as $T = (g_1^{R'}, Y_1, Y_2)$ where:

$$Y_1 = (g_1^{\frac{KS'x_2}{S_j L_1}})^{R'}$$

$$Y_2 = e'(g_1^{\frac{S'x_1}{S_j} + \frac{S'}{L_1}}, P_{Es_i})^{R'}$$
$$= e'(g_1, g_2)^{\frac{R'S_{Es_i} \times S'x_1}{S_j} \times L_1 + S'R'S_{Es_i}}$$

- Generate a digital signature $Sig = (s, e)$ for data D according to Schnorr scheme [23] as follows:
  1) set $r = Y_1$.
  2) Compute $e = H(r||D)$.
  3) Compute $s = R' \times (A - e \times B)$, where $A = \frac{S'}{S_j L_2}$ and $B = \frac{S'}{S_j L_1}$.
- Upload the data, its digital signature and the anonymization token into the externalization server as (T, Sig, D).

We note that our solution aims to achieve an accountable privacy preserving signature scheme. Therefore, we do not consider data confidentiality service in this paper.

5) Authenticity and signature verification step: in order to make shared information visible for public, the externalization server starts to verify the information owner's authenticity. The authenticity verification process aims to make sure that the owner is a valid member who could be accountable by the authority. We note that this verification process preserves the privacy of the information owner since it prevents the externalization server from discovering its identity. Besides, it does not allow to trace the owner's activity as well. Once the server achieves the anonymous authenticity verification process, it also verifies that the information used to prove the authenticity of the communicating entity is related to the signature provided with the information.

The authenticity verification process runs in two steps. In the first step:

- Compute V_1 as:

$$V_1 = e'(Y_1 \times g_1^{\frac{KS'}{L_1}}, g_2^{S_{Es_i} \times L_1 \frac{L_2}{K}})$$
$$= e'(g_1, g_2)^{\frac{S_{Es_i}S'}{S_j} L_2 (R'x_2 + S_j)}$$

- Compute V_2 as:

$$V_2 = Y_2 \times e'(g_1, g_2)^{S_{Es_i}S'L_1}$$
$$= e'(g_1, g_2)^{\frac{R'S_{Es_i} \times S'x_1}{S_j} \times L_1 + S'R'S_{Es_i}} \times e'(g_1, g_2)^{S_{Es_i}S'L_1}$$
$$= e'(g_1, g_2)^{\frac{S_{Es_i}S'}{S_j} L_1 (R'x_1 + S_j) + R'S'S_{Es_i}}$$

- Compute V as

$$V = V_1 \times V_2$$
$$= e'(g_1, g_2)^{\frac{S_{Es_i}S'}{S_j} \times (L_1(R'x_1 + S_j) + L_2(R'x_2 + S_j)) + R'S_{Es_i}S'}$$

- Compute V' as

$$V' = \frac{V}{e'(g_1^{R'}, g_2^{S_{Es_i}S'})}$$
$$= \frac{e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_j}\times(L_1(R'x_1+S_j)+L_2(R'x_2+S_j))+R'S_{Es_i}S'}}{e'(g_1,g_2)^{R'S_{Es_i}S'}}$$
$$= e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_j}\times(y_1\times L_1+y_2\times L_2)+R'S_{Es_i}S'-R'S_{Es_i}S'}$$
$$= e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_j}\times S_j} = e'(g_1,g_2)^{S_{Es_i}S'}$$

Where $y_1 = R'x_1 + S_j$ and $y_2 = R'x_2 + S_j$.
Note that, $(y_1 \times L_1 + y_2 \times L_2 = S_j)$ represents the polynomial interpolation at $x = 0$.
If the value V' computed in this first step is equal to the externalization server key $P = e'(g_1, g_2)^{S_{Es_i}S'}$, received from the authority, then the information owner is considered as a valid member of the information-sharing group.
Moreover, this also proves that the owner could be accountable by the authority. In fact, the externalization servers perform polynomial interpolation using values computed with L_1 and L_2. (L_1, L_2) are computed using two secret values x_1 and x_2 that are known only by the authority. Therefore, in order to correctly perform the polynomial interpolation, the entity must provide shares generated using the same pieces of coordinates used to compute (L_1, L_2). Since anonymization tokens, provided with the shared information, do not reveal the values (x_1, x_2, S_j, S', K), the only way that allows any entity to be authenticated is to get valid credentials from the authority. As a result, a succesfull authentication means that the authority is able to trace the communicating entity.
In the second step, the externalization server proceeds to the signature verification process. This process ensures that the information owner who have provided the anonymization token is the same who signed data D.
In order to verify the signature $Sig = (e, s)$, the server executes the following steps:

1) Let $r_v = g_1^s$.
2) Let $e_v = H(Y_1||D)$
3) If $(e'(Y_1, g_2^{\frac{-1}{K\times x_1}}) = e'(r_v, g_2) \times e'(Y_1^{e_v}, g_2^{\frac{1}{K\times x_2}}))$ then the signature is verified
4) Otherwise, the signature is not verified.

We recall that $\frac{1}{L_2} = \frac{(x_1-x_2)}{x_1}$. Moreover, $\frac{1}{L_1} = \frac{(x_2-x_1)}{x_2}$ which means that $\frac{KS'\times x_2}{S_j\times L_1} = \frac{KS'(x_2-x_1)}{S_j}$.
Thus, $e'(Y_1, g_2^{\frac{-1}{K\times x_1}}) = e'(g_1,g_2)^{\frac{R'S'\times(x_1-x_2)}{S_j\times x_1}}$

$$= e'(g_1,g_2)^{\frac{R'S'}{S_jL_2}}$$

On the other hand:

$$e'(r_v, g_2) \times e'(Y_1^{e_v}, g_2^{\frac{1}{K\times x_2}}) = e'(g_1,g_2)^{\frac{R'S'}{S_jL_2}-\frac{R'S'e}{S_jL_1}+\frac{R'S'e_v}{S_jL_1}}$$

$$= e'(g_1,g_2)^{\frac{R'S'}{S_jL_2}} \text{ , if } e = e_v$$

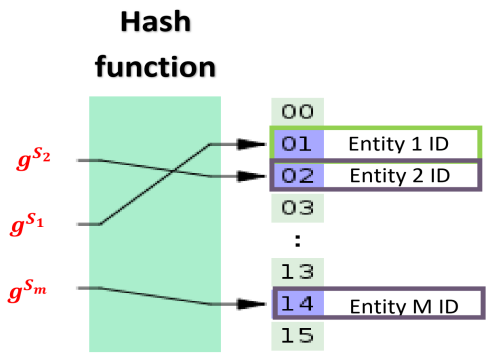

Fig. 2. A description of the lookup function during tracking phase

6) Tracking step: it is true that our solution ensures full anonymity in the externalization servers, but it also allows tracing any user, if the system detects any anomalies. In our solution, the trusted authority ensures the accountability service using the anonymization token uploaded with information. The tracking process runs as follows:

- Given the signature $Sig = (e, s)$, compute:

$$T' = \frac{s}{S'\times(\frac{1}{L_2}-\frac{1}{L_1}\times e)} = \frac{\frac{S'R'}{S_j}\times(\frac{1}{L_2}-\frac{1}{L_1}\times e)}{S'\times(\frac{1}{L_2}-\frac{1}{L_1}\times e)} = \frac{R'}{S_j}$$

- Compute $T'' = (g_1^{R'})^{\frac{1}{T'}} = g^{S_j}$
- As shown in figure 2, the authority stores both the traces and the user's identity in a Hash Table (users registry), it only needs to look up for T'' in the registry and gets the corresponding user's identity.

Figure 3 summarizes the different steps of our solution going from the setup phase to the tracking phase.

VII. Security Analysis

In this section, we discuss the security of our scheme and show that it ensures the expected privacy and accountability requirements. Moreover, we provide the proof of correctness of our protocol in the appendix of this paper.

A. Replay/impersonation attack

An attacker may want to intercept an information signed by another entity and replays it later. To avoid that kind of situation, communicating entities should include timestamps when they share public information. In that case, it will be easy for externalization servers to detect replayed information. An attacker may also try to impersonate one of the valid communicating entities in the network. To do so, the attacker can try to generate valid credentials using brute force. Applying brute force on a cyclic group of order p, where p is a safe prime, is a computational consuming task. Furthermore, the attacker may intercept a valid signed message and then try to extract valid credentials from token provided with the message, or to only change the message content in order to share it into the externalization server.

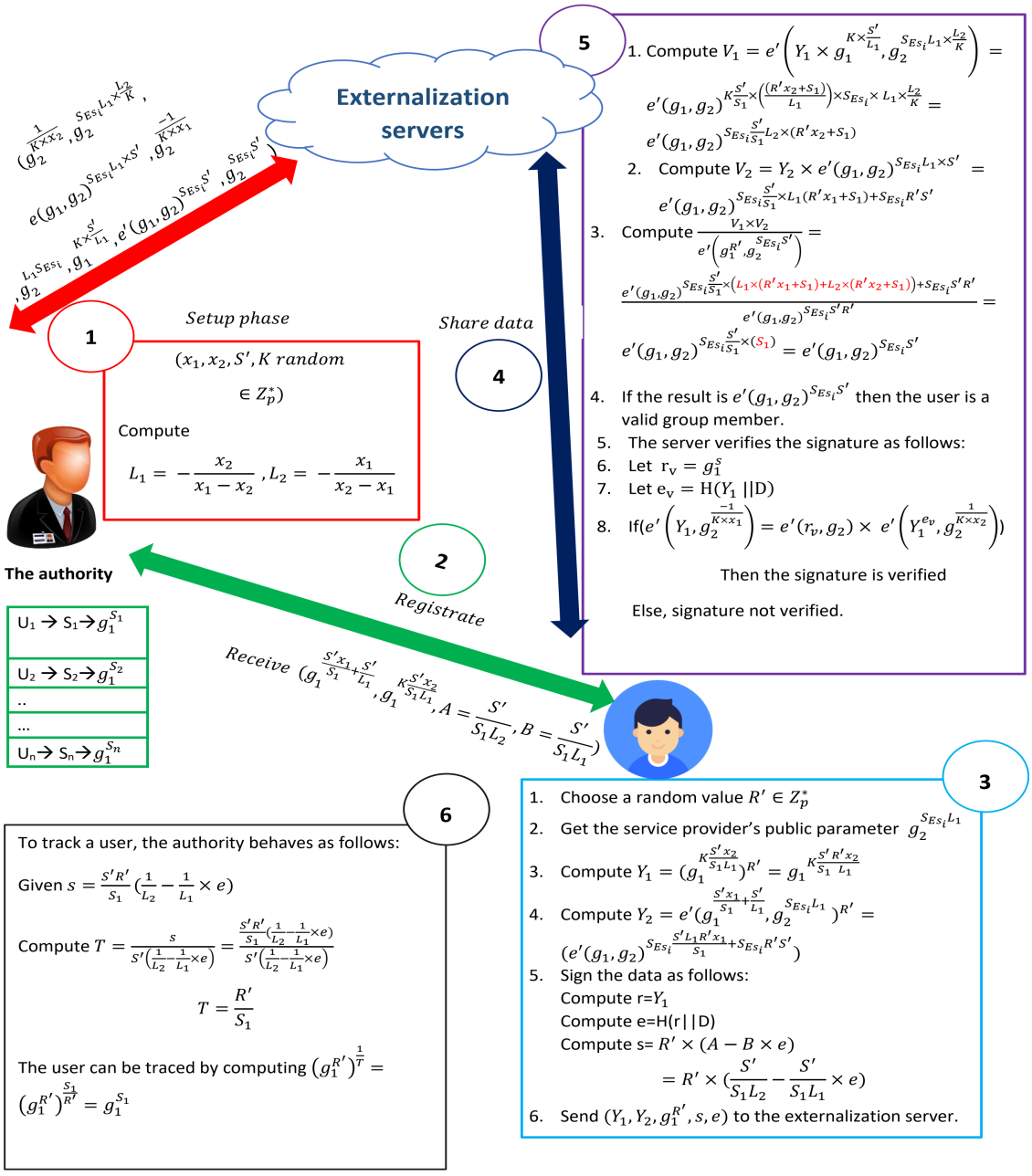

Fig. 3. A descriptive diagram of our scheme

Extracting credentials from a signed message means that the attacker is able to guess the output of the pseudo-random function (PRF) used by the entities. Moreover, it will also require from that attacker to solve discrete logarithm problem known by its hardness in multiplicative groups. Similarly, if an attacker tries to change only the content of the message, he will need to forge Schnorr signature. We note that, the security of Schnorr scheme has been proved in [19].

B. Privacy breach

In order to violate users' privacy in our scheme, the adversary needs to find out one of the unique values that the authority provides to each user.

Given the public information $(Y_1, Y_2, g_1^{R'}, s, e)$ made available to the adversary during each information-sharing event, we can deduce the following:

- The adversary will have no benefit from targeting the value e to extract useful information since e is computed based on two public values, namely Y_2 and the shared data D.
- The adversary cannot deduce any useful information from the signature s. In fact, s is computed based on values A,B and e. All these values are randomized, thus, as long as we use a secure pseudo-random function $S-PRF$, the adversary cannot distinguish between signatures randomized

through $S-PRF$.

Therefore, the adversary will try to use $(Y_1, Y_2, g_1^{R'})$ to breach the users' privacy. We recall that $Y_1 = (g_1^a)^{R'}$ and $Y_2 = (e'(g_1, g_2)^b)^{R'}$, where g_1^a and $e'(g_1, g_2)^b$ are two values that could identify the users. Hence, the adversary will aim to trace users using either Y_1 or Y_2 along with $g_1^{R'}$.

In what follows, we will prove the security of our scheme against an adversary who tries to identify users based on Y_1 and $g_1^{R'}$ values. Note that the same proof can be applied on adversaries who use the Y_2 instead of Y_1 in their attack.

Assumption 1: (*Computational Diffie-Hellman assumption*) given a multiplicative cyclic group G of order p with generator g_1, a probabilistic polynomial-time adversary has a negligible probability of computing g_1^{ab} from (g_1, g_1^a, g_1^b), where a, b are random values in Z_p^*.

Theorem 1: if our scheme is broken, we can construct a polynomial time adversary who breaks assumption 1.

Proof 1: let us call Adv_1, the adversary who breaks users' privacy in our scheme. A_1 plays the following security game: given $(g_1, Y_1 = (g_1^a)^{R'}, g_1^{R'})$ as input, Adv_1 tries to output g_1^a. If Adv_1 has a non-negligible advantage in the security game above, we can reconstruct an adversary Adv_2 which uses Adv_1 as a sub-routine, and has a non-negligible advantage in breaking assumption 1.

We recall that Adv_2 takes (g_1, g_1^a, g_1^b) as inputs and tries to output g_1^{ab}.

The construction of Adv_2, given a polynomial adversary Adv_1 who breaks users' privacy in our scheme with a non-negligible probability, is as follows:

1) Adv_2 receives the input values (g_1, g_1^a, g_1^b).
2) Adv_2 calls Adv_1 with (g_1, g_1, g_1^b) as input.
3) If Adv_1 has a non-negligible advantage in breaking users' privacy in our scheme, it will output $g_1^{1/b}$.
4) Adv_2 calls Adv_1 with $(g_1, g_1^a, g_1^{1/b})$ as input.
5) If Adv_1 has a non-negligible advantage in breaking users' privacy in our scheme, it will output $g_1^{\frac{a}{1/b}} = g_1^{ab}$.
6) Finally, Adv_2 outputs g_1^{ab} and breaks assumption 1.

Conclusion 1: according to theorem 1, the existence of an adversary who breaks users' privacy in our scheme implies the existence of an adversary who breaks assumption 1. Thus, as long as assumption 1 holds our scheme is secure.

C. Accountability breach

A malicious user may try to submit a token that allows him to be authenticated in the externalization servers but not to be tracked by the authority. In that kind of attacks, we can distinguish two scenarios:

In the first one, the attacker tries to generate valid credentials based on the information available in public (the anonymization tokens submitted with the shared information), without resorting to the authority. This means that the attacker needs to reveal the values $MK = (K, S', x_1, x_2)$ known only by the authority. Note that, in the values available in public, MK components are protected according to the hardness of the discrete logarithm problem in multiplicative cyclic groups. Therefore, the attacker will have to solve discrete logarithm problem in order to generate valid credentials.

In the second, the attacker is a valid group member who possesses valid credentials, but he tries to modify them in a way that allows its authentication at the externalization servers but does not allow the authority to trace him. In that case, we can distinguish two possibilities: In the first possibility, the attacker combines its valid credential components in order to generate fake ones. We note that fake credentials need to allow the user to be successfully authenticated at the externalization servers, so it needs to have the following form:

$$FC = (y_1 = g_1^{\frac{S'x_1}{S_F} + \frac{S'}{L_1}}, y_2 = g_1^{\frac{KS'x_2}{S_F L_1}}, A = \frac{S'}{S_F L_2}, B = \frac{S'}{S_F L_1})$$

Given the original credentials:

$$OC = (y_1 = g_1^{\frac{S'x_1}{S_j} + \frac{S'}{L_1}}, y_2 = g_1^{\frac{KS'x_2}{S_j L_1}}, A = \frac{S'}{S_j L_2}, B = \frac{S'}{S_j L_1})$$

We can clearly notice that the attacker has one particular challenge that consists of replacing S_j by S_F. Faking the values A, B and y_2 of OC is an easy task. However, applying the same changes on y_1 requires the knowledge of $\frac{S'}{L_1}$ or $g_1^{\frac{S'}{L_1}}$. Since both values are known only by the authority, the attacker can only use brute force in order to reveal them.

In the second case, the attacker may collude with other users or malicious externalization servers and fake its credentials. Similarly, to the first possibility, the attacker needs to get rid of the value $\frac{S'}{L_1}$ available in y_1. The challenge in that case consists of finding the value $g_1^{\frac{S'}{L_1}}$ given $g^{\frac{KS'}{L_1}}$. Thus, he needs to solve discrete logarithm problem.

D. Discussion

In our solution, we propose an anonymous signature scheme that prevents any other entity than the authority from tracking users through their signature. However, if the users share personal or sensitive information in a public context, they may be identified even if they sign data with an anonymous signature. In the context of public information sharing, that we consider in this paper, users usually share information that has a public nature such as traffic information, incident reporting, etc. Therefore, it is unlikely to share personal or sensitive information in that same context. Thus, an attacker in that case will track users activities based on the signature provided with the shared information rather than the information itself.

Nevertheless, if the users are in a context where they may share personal or sensitive information, our solution will ensure the anonymity of the signature, but it will be more secure either to encrypt the shared information or to use data anonymization techniques such as k-anonymity [30].

Note that, if users tend to anonymazation techniques, it is important to anonymize data in a way that is resilient against attacks such as homogeneity attacks or background knowledge attacks [13].

VIII. Application and performance evaluation

In this section, we apply our accountable privacy-preserving scheme on event-reporting application use case. Then, we evaluate, through simulations, its performance on the proposed use case.

Accountable and privacy-preserving are among the most important requirements to ensure while reporting events. Indeed, users cannot agree to report events by themselves or to allow their connected objects to diffuse data that may violate their privacy and expose their identities or ease tracking them. On the other hand, law authorities need to be able to ensure order and track users in case of misbehavior, which makes accountability as important as privacy-preserving.

The minimal architecture of any secure event-reporting application is composed of three main components: 1) **the users** reporting events occurring in the architecture; 2) **the authority** which manages security on the architecture and ensures accountability service if it is requested by law authorities; 3) **externalization servers** responsible for information collecting, aggregation and publishing.

Our accountable privacy-preserving solution fits perfectly with the requirements of event-reporting applications, and operates directly on its minimal architecture without requiring any additional component.

Given that most of event-reporting situations require a real time treatment, adopting fog computing paradigm becomes more suited. Thus, without loss of generality, fog nodes are going to play the role of externalization servers in our use case.

Note that the eventual event indexation and aggregation problems are not in the scope of this paper. Moreover, we do not address in our application use case the problems related to fog computing architecture and which do not have a direct relationship with privacy-preserving and accountability.

To measure the performance of our solution, we implemented an event-reporting environment, in which:

1) we emulate the setup-launcher module (available in the authority) as a program that runs the setup phase as described in sub-section VI-C1.
2) we emulate the fog-subscriber module (available in the authority) as a program that intercepts registration requests formulated by fog servers, and sends back the verification parameters as described in sub-section VI-C2.
3) we emulate the subscription module (available in each fog server) as a program which requests the verification parameters from the fog-subscriber module.
4) we emulate the users-registration module (available in the authority) as a program that intercepts registration requests formulated by the users, and sends back the registration credentials as described in sub-section VI-C3.
5) we emulate the registration module (available in each connected object) as a program which requests credentials from the users-registration module.
6) we emulate the event-reporting module (available in each connected object) as a program that generates and signs random events, as described in our information sharing phase (sub-sectionVI-C4), before sending it to fog servers.
7) we emulate the event-collecting module (available in each fog server) as a program that executes our signature verification algorithm (sub-sectionVI-C5) to verify the signature of the reported events, before making them available to the public.
8) we emulate the identity disclosure module as a program (available in the authority) that executes our tracking algorithm (sub-section VI-C6) in order to break the anonymity of misbehaving users. This module interacts with the setup-launcher module to get some setup parameters. Moreover, it interacts with the users-registration module to get information related to users registration.

In our event-reporting environment, the authority first executes our setup-launcher module to generate the system parameters. Later on, each fog server runs its subscription module to get the verification parameters from the authority. On the other hand, each user, willing to report events in our environment, needs to call its registration module to get its registration credentials.

Once this task is successfully executed, the event-reporting module can proceed to report events. To do so, we randomly schedule a set of events to sign and report to the fog server. When public information is received, the event-collecting module uses the verification parameters, brought from the authority, to verify the signature of each reported event in order to publish it.

In the case where a misbehaving event has been pointed out to the authority, the tracking module uses the signature available in the reported event to disclose the identity of its reporter.

Figure 4 describes the sequence of the main actions adopted in our event-reporting environment.

To evaluate privacy-preserving and accountability features in our event-reporting environment and compares it with existing solutions, we first measure the computational time of the credential generation phase. Then, we provide comparative tables in which we compare our solution with existing accountable privacy-preserving solutions according to: 1) the number of operations performed in the credential generation phase; 2) the number of operations performed during the signature process as well as its computational cost; 3) the signature sizes.

We also provide a comparison between our solution and existing solutions in terms of: 1) signature verification time; 2) the computational operations performed during this phase and 3) the number of computational operations performed during the tracking phase.

Finally, we simulate the arrival of reported-event requests in one fog server, and compare our solution to existing ones, according to the number of events waiting to be verified and published in that server.

We note that the experiments run on an adhoc network composed of an HP, i7 laptop with a CPU frequency of 2.7 GHZ and 16 GB of RAM, and a Toshiba i5 laptop with CPU frequency of 2.4 GHZ and 6 GB of RAM. We used pbc-0.5.14

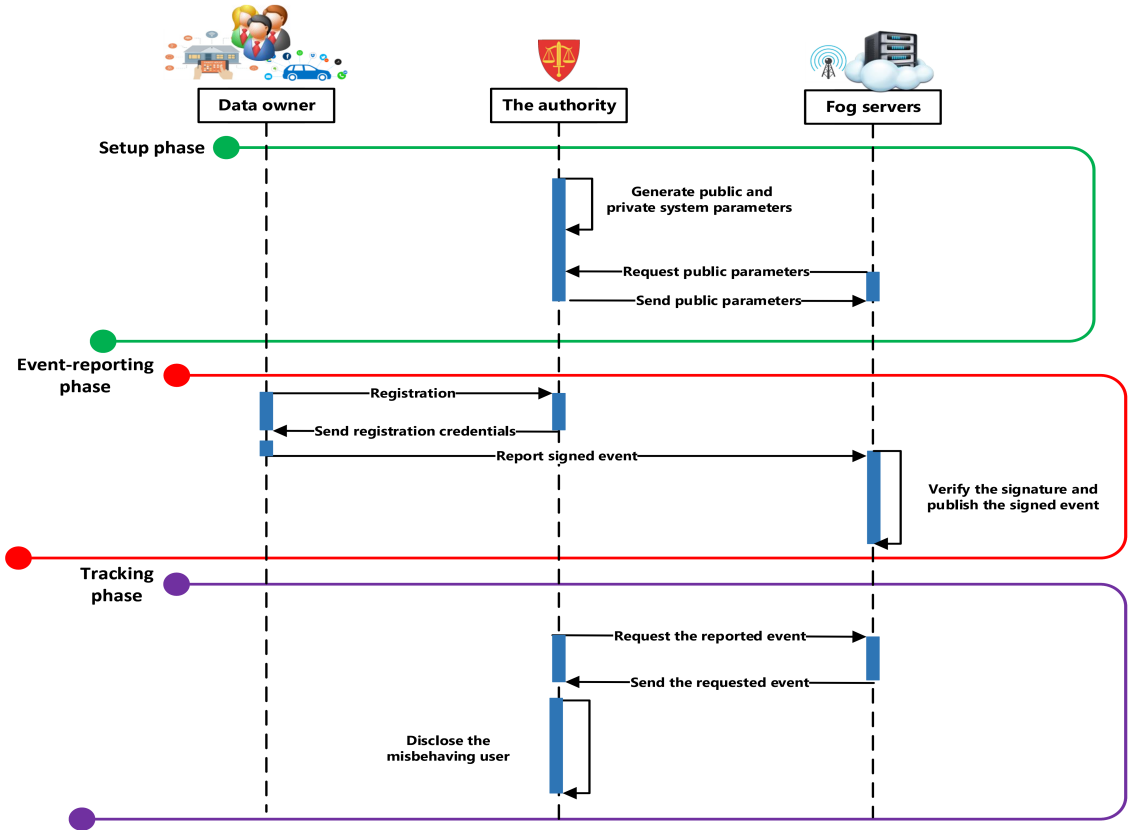

Fig. 4. The major sequences executed in our event-reporting environement

security library in our implementation. The sizes of elements G_1^*, G_T^* and Z_p^* used in our implementation are $21, 61$ and 20 bytes respectively.

Moreover, we have ran 50 executions in each measurement, and the presented results represent the average of the computational time collected in these multiple executions.

A. Credential generation

In our scheme, all communicating entities execute the registration phase at the authority. Once the authority verifies the identity of the communicating entity, it generates a valid token that the entity will use to sign public information. Given that the authority in our scheme performs a constant number of operations in each registration, the complexity of this process is in the order of $O(n)$, where n is the number of registration requests received in parallel. Table II, provides a comparison between our solution and existing accountable privacy-preserving solutions, according to the number of operations performed during the credential generation phase. As shown in table II, our scheme proposes a constant and less heavier credential generation process compared to existing solutions.

TABLE II
A COMPARATIVE TABLE OF THE COMPUTATIONAL OPERATIONS PERFORMED IN THE CREDENTIAL GENERATION PHASE

Ours	Mona [15]	TPP [31]	Anonymous [25]
$2div + 2Me$	$3P + (r + 5)Pm + 1Me$	$2P + 4Pm + 4Me + 1Pa$	$1P + 12kPa$

(div, add) refer to modular division, and addition resp. (P, Me, Pm, Pa) refer to paring operation, modular exponentiation, elliptic curve point multiplication and point addition resp. (k, r) are two parameters defined in [15] and [25] schemes resp.

B. Signature process

In our event-reporting environment, all communicating entities that want to share public information through fog servers, must sign the reported events. Our signature process adopted in each event-reporting module, requires the computation of one pairing operation, three modular exponentiations, two multiplications and a Hash function. On the hand, existing solutions perform a considerable number of pairing operations, elliptic curve point multiplications (going up to eleven in the case of Mona [15]), additions and modular exponentiations in their signature process. Table III, provides a comparison between our solution and the existing accountable privacy-preserving schemes in terms of the average computational cost of the signature process, signature sizes and the number of operations performed during the same process. Moreover, we show in figure 5, the communication overhead resulting from the transmission of signed information to the fog servers. To compute the communication overhead, we first measured the transmission time of full data (the payload), given a

TABLE III
A COMPARATIVE TABLE OF THE OPERATIONS AND COMPUTATIONAL COST OF THE SIGNATURE AND ITS VERIFICATION PHASES, ALONG WITH SIGNATURE SIZES

	Our	**Mona**	**TPP**	**Anonymous**
Signature time (ms)	**8.55**	**33.40**	**31.13**	**33.40**
Signature computational operations	$1P+3Me+2mult+1add$	$11Pm + 3P + 3Me + 3Pa + 7add + 5mult$	$4P + 3Pm + 1Pa$	$11Pm + 3P + 3Me + 3Pa + 7add + 5mult$
Signature size	$1G_T^* + 2G_1^* + 2Z_p^*$	$3G_1^* + 6Z_p^*$	$3G_1^* + 2G_T^* + 1Z_p^*$	$3G_1^* + 6Z_p^*$
Signature verification computational operations	$5P + 4Pm + 1Me$	$5P+12Pm+6Me+4Pa$	$6P + 3Pm + 1Pa$	$5P+11Pm+4Me+4Pa$
Verification time (ms)	**40**	**66**	**77**	**70**

$(mult, add)$ refer to modular division, and addition resp. (P, Me, Pm, Pa) refer to paring operation, modular exponentiation, elliptic curve point multiplication and point addition resp.

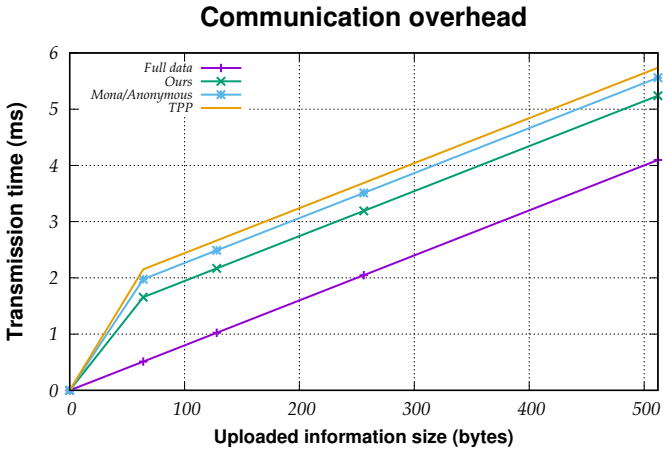

Fig. 5. The communication overhead resulting from the information-sharing process

network bandwidth of 10 Mbps. Then, we measured the extra transmission time induced by the signature in each scheme. As shown in figure 5, our solution has the lowest communication overhead since it offers the smallest signature size compared to existing solutions.

C. Signature verification process

In our event-reporting environment, the fog servers verify the authenticity of any reported event before making it available to the public. Figure 6, shows the verification time spent by the fog server to verify the authenticity of events received in parallel. As we can see, our solution outperforms existing accountable privacy-preserving solutions in terms of computational time consumed in the verification process. These results can be explained through table III, where we notice that our signature verification process does not require as much point multiplications as it is required in [25] and [15]. Moreover, it does perform less pairing operations than [31].

In addition, we show in figure 7, a comparison between our solution and existing accountable privacy-preserving solutions according to the number of information waiting to be verified by the fog server. The results in figure 7 have been obtained through a simulation, in which the reporting of events follows a Poisson distribution with an arriving rate $(\lambda = \frac{1}{6})$. Thus, the fog server will receive one signed information each six milliseconds. In our simulation, each fog server defines a single Queue Q that will contain signed events waiting for the signature verification process. Finally, we observe the evolution of Q during the simulation time (7 seconds in our case). Figure 7 results show that our signature verification process achieves an average of seven reported events waiting to be verified and published along the simulation time, while Mona [15] and TPP [31] achieve an average of eleven and thirteen waiting events respectively.

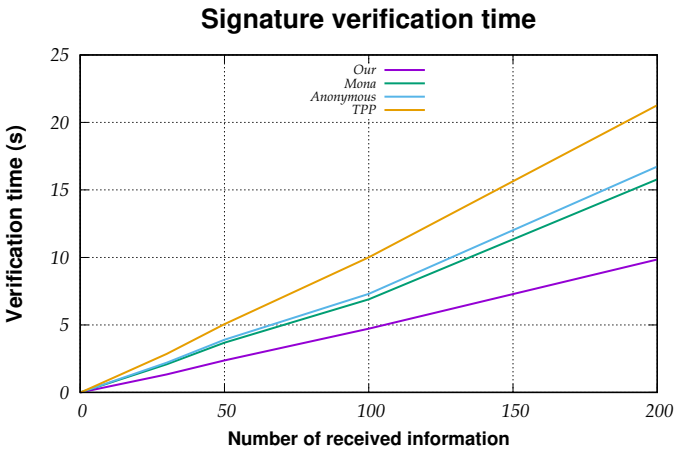

Fig. 6. The computational time consumed in signature verification

D. Tracking process

In our event-reporting environment, the authority tracks users and reveal their identities in case of misbehavior. As shown in the benchmarks of JPBC library [1], the computational operations of the tracking process performed in our solution have more or less the same computation time as the operations performed in solutions [15], [25], [31], in all elliptic curve configurations. In terms of complexity, table IV shows that each execution of the tracking process in the compared

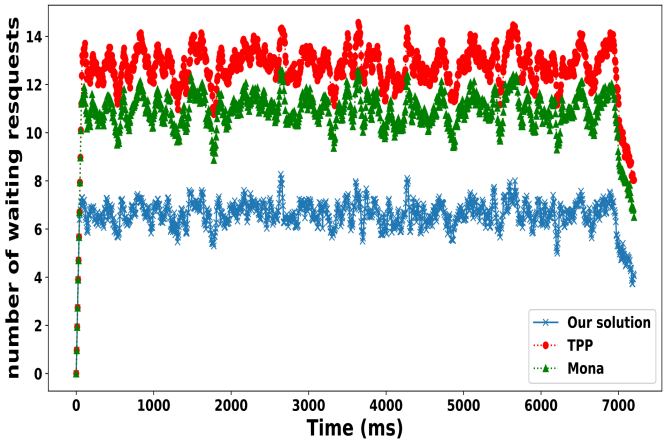

Fig. 7. The number of non-verified sharing requests as a function of time

TABLE IV
A COMPARATIVE TABLE OF THE COMPUTATIONAL OPERATIONS PERFORMED IN TRACKING PHASE

Ours	Mona [15]	TPP [31]	Anonymous [25]
$1sub + 2mult + 2div + 1Me$	$2Pm + 2Pa$	$2sub + 1div + 1Pm + 1Pa$	$2Pm + 2Pa$

(div, add, sub) refer to modular division, addition and substruction resp. (Me, Pm, Pa) refer to modular exponentiation, elliptic curve point multiplication and point addition resp.

solutions requires the computation of a constant number of arithmetic operations. Therefore, given n tracking requests formulated in parallel to the tracking module, we can conclude that the complexity is in the order of $O(n)$.

E. Comparaison in terms of security requirements

In addition to the comparison that we provided above in terms of computational performance, we also compare our solution to existing solutions in terms of security requirements. To do so, we identified the following security features as criteria of comparison:

1) **Authentication requirement** in which we note whether the privacy-preserving solution allows to authenticate users or not.
2) **Unlinkability requirement** in which we note if there is any unauthorized entity in the architecture able to track users.
3) **Accountability requirement** in which we note whether there is an authorized and trusted entity that is able to track legitimate users in case of misbehavior detection.
4) Immunity against the **single point of failure** problem.

As shown in table V, our solution provides all the requirements cited above. On the other hand, we notice that there are solutions in the literature such as [15], [25], [31] that provide privacy-preserving and accountability at the same time. However, these solutions are centralized and thus they suffer of the single point of failure problem. Besides, we also find other solutions in the literature such as [6], [14], [28] that operate in a completely decentralized manner and preserve users' privacy, but they do not provide any accountability service that deals with any detected misbehavior.

TABLE V
A COMPARATIVE TABLE BETWEEN EXISTING SOLUTIONS IN TERMS OF SECURITY

	Ours	[15]	[31]	[25]	[14]	[28]	[6]
Users' authentication	+	+	+	+	+	+	-
Unlinkability (Strong anonymity)	+	+	+	+	+	-	+
Accountability	+	+	+	+	-	+	-
Immune against single point of failure	+	-	-	-	+	-	+

IX. CONCLUSION

In this paper, we have proposed a new secure, accountable privacy-preserving scheme. Based on the secret sharing method and randomization techniques, our solution allows anonym and accountable public information sharing in information sharing architectures. In our scheme, communicating entities perform one registration with the registration authority. Then, they will be able to share information through the externalization servers without resorting to the registration authority or any third party. Each communicating entity signs shared information with an anonymous token. That token will allow the externalization servers to verify the entity's authenticity without violating its privacy. In the case of anomaly detection, the authority is able to trace any communicating entity in the system, in spite of the anonymity of the provided signature. In addition to security features, our solution does not indulge a considerable overhead in terms of storage and communication. Indeed, our information-sharing process does not require several exchanges between the servers and the communicating entities. Furthermore, externalization servers do not need to store users' pseudonyms or any temporary digital certificates; they only hold a constant set of values that are going to allow them to verify the authenticity of any entity in the system. Besides, our scheme deals efficiently with situations where an entity tries to impersonate and share information on behalf of another one. Finally, our experimental results show that our proposal outperforms existing accountable privacy-preserving solutions. In the future, we intend to address the problem of conditional revocation, where the authority provides mechanisms for temporary or permanently prevent malicious users from sharing public information. Moreover, we believe that it will be interessting to prove that our system is secure under a more standard model than the random oracle.

X. AKNOWLEDGMENT

This work was supported by both the Algerian government, and carried out in the framework of the Labex MS2T, which

was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02)

REFERENCES

[1] Jpbc - java pairing-based cryptography library : Benchmark. http://gas.dia.unisa.it/projects/jpbc/benchmark.html, 4 december 2013 [consulted on 20 may 2019].

[2] Cisco visual networking index: Forecast and methodology, 20162021 - complete-white-paper-c11-481360.pdf, "2018". https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf, 6 november 2018 [consulted on 6 november 2018].

[3] B. Aslam and C. Zou. Distributed certificate and application architecture for vanets. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–7. IEEE, 2009.

[4] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Annual International Cryptology Conference*, pages 41–55. Springer, 2004.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[6] J. M. de Fuentes, L. Gonzlez-Manzano, J. Tapiador, and P. Peris-Lopez. Pracis: Privacy-preserving and aggregatable cybersecurity information sharing. *Computers & Security*, 69:127 – 141, 2017. Security Data Science and Cyber Threat Management.

[7] P. Gope. Laap: Lightweight anonymous authentication protocol for d2d-aided fog computing paradigm. *Computers & Security*, 2019.

[8] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, and J. Hu. Appa: An anonymous and privacy preserving data aggregation scheme for fog-enhanced iot. *Journal of Network and Computer Applications*, 125:82–92, 2019.

[9] D. He, S. Chan, and M. Guizani. An accountable, privacy-preserving, and efficient authentication framework for wireless access networks. *IEEE Transactions on Vehicular Technology*, 65(3):1605–1614, 2016.

[10] J. Heurix, P. Zimmermann, T. Neubauer, and S. Fenz. A taxonomy for privacy enhancing technologies. *Computers & Security*, 53:1–17, 2015.

[11] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien. Abaka: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks. *IEEE Transactions on vehicular technology*, 60(1):248–262, 2010.

[12] A. A. Khan, M. Abolhasan, and W. Ni. 5g next generation vanets using sdn and fog computing framework. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2018.

[13] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[14] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Enhancing location privacy for electric vehicles (at the right time). In *European Symposium on Research in Computer Security*, pages 397–414. Springer, 2012.

[15] X. Liu, Y. Zhang, B. Wang, and J. Yan. Mona: Secure multi-owner data sharing for dynamic groups in the cloud. *ieee transactions on parallel and distributed systems*, 24(6):1182–1191, 2013.

[16] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24. IEEE, 2006.

[17] R. Mahmud, R. Kotagiri, and R. Buyya. Fog computing: A taxonomy, survey and future directions. In *Internet of everything*, pages 103–130. Springer, 2018.

[18] H. Nicanfar, S. Hosseininezhad, P. TalebiFard, and V. C. Leung. Robust privacy-preserving authentication scheme for communication between electric vehicle as power energy storage and power stations. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 55–60. IEEE, 2013.

[19] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 387–398. Springer, 1996.

[20] C. Rottondi, S. Fontana, and G. Verticale. Enabling privacy in vehicle-to-grid interactions for battery recharging. *Energies*, 7(5):2780–2798, 2014.

[21] C. Rottondi, S. Fontana, and G. Verticale. A privacy-friendly framework for vehicle-to-grid interactions. In *International Workshop on Smart Grid Security*, pages 125–138. Springer, 2014.

[22] F. M. Salem, M. H. Ibrahim, and I. Ibrahim. Non-interactive authentication scheme providing privacy among drivers in vehicle-to-vehicle networks. In *Networking and Services (ICNS), 2010 Sixth International Conference on*, pages 156–161. IEEE, 2010.

[23] C.-P. Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.

[24] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[25] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo. Anonymous and traceable group data sharing in cloud computing. *IEEE Transactions on Information Forensics and Security*, 13(4):912–925, 2018.

[26] I. Stojmenovic and S. Wen. The fog computing paradigm: Scenarios and security issues. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1–8. IEEE, 2014.

[27] V. Sucasas, G. Mantas, S. Althunibat, L. Oliveira, A. Antonopoulos, I. Otung, and J. Rodriguez. A privacy-enhanced oauth 2.0 based protocol for smart city mobile applications. *Computers & Security*, 74:258–274, 2018.

[28] V. Sucasas, G. Mantas, F. B. Saghezchi, A. Radwan, and J. Rodriguez. An autonomous privacy-preserving authentication scheme for intelligent transportation systems. *Computers & Security*, 60:193–205, 2016.

[29] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Transactions on Vehicular Technology*, 59(7):3589–3603, 2010.

[30] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[31] H. Wang, B. Qin, Q. Wu, L. Xu, and J. Domingo-Ferrer. Tpp: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids. *IEEE Transactions on Information Forensics and Security*, 10(11):2340–2351, 2015.

XI. APPENDIX

A. Proof of correctness

To prove the correctness of our protocol, we need to prove the following:

- **Claim 1**: given an anonymous token T provided by a legitimate user u, $Auth(T) = e(g_1, g_2)^{S_{Es_i}S'}$, where $Auth()$ is a function that executes the token authentication process and $e(g_1, g_2)^{S_{Es_i}S'}$ is the public key of server Es_i.
- **Claim 2**: given the anonymous token T and the signature $Sig = (s, e)$ provided by a legitimate user u, $Verif(T, Sig) = true$, where $Verif()$ is a function which verifies that T has been used to sign the shared information.
- **Claim 3**: given the anonymous token T and the signature $Sig = (s, e)$ provided by a legitimate user u, $Trace(T, Sig) = g_1^{S_u}$, where $Trace()$ is a function which executes our traking process and $g_1^{S_u}$ represents the user u's trace computed during his registration.

To prove the correctness of claim 1, let us recall that following our protocol, the user u submits an anonymous token T expressed as follows:

$$T = (g_1^{R'}, Y_1 = (g_1^{\frac{KS'x_2}{S_u L_1}})^{R'}, Y_2 = e'(g_1^{\frac{S'x_1}{S_u} + \frac{S'}{L_1}}, P_{Es_i})^{R'})$$

On the other hand, the server ES_i authenticates u through the following steps:

Given the public parameter $PP_1 = g_1^{\frac{KS'}{L_1}}$, compute $V_1' = Y_1 \times PP_1$ as:

$$V_1' = g_1^{\frac{R'KS'x_2}{S_uL_1}} \times g_1^{\frac{KS'}{L_1}} = g_1^{\frac{R'KS'x_2}{S_uL_1}+\frac{KS'}{L_1}}$$

By taking $\frac{KS'}{S_uL_1}$ as a common factor, we get:

$$V_1' = g_1^{\frac{KS'}{S_uL_1}\times(R'x_2+S_u)}$$

Given the public parameter $PP_2 = g_2^{S_{Es_i}\times L_1\frac{L_2}{K}}$, compute:

$$V_1 = e'(V_1', PP_2) = e'(g_1,g_2)^{\frac{KS'}{S_uL_1}\times(R'x_2+S_u)\times S_{Es_i}\times L_1\frac{L_2}{K}}$$

By eliminating factors L_1 and K from the power, we get:

$$V_1 = e'(g_1,g_2)^{\frac{S'S_{Es_i}}{S_u}\times L_2\times(R'x_2+S_u)} \quad (5)$$

Given the public parameter $PP_3 = e'(g_1,g_2)^{S_{Es_i}S'L_1}$, compute $V_2 = Y_2 \times PP_3$ as:

$$V_2 = e'(g_1,g_2)^{\frac{R'S_{Es_i}\times S'x_1}{S_u}\times L_1+S'R'S_{Es_i}} \times e'(g_1,g_2)^{S_{Es_i}S'L_1} = e'(g_1,g_2)^{\frac{R'S_{Es_i}\times S'x_1}{S_u}\times L_1+S_{Es_i}S'L_1+S'R'S_{Es_i}}$$

By taking $\frac{S'S_{Es_i}}{S_u} \times L_1$ as a common factor from the first two factors appearing in the power, we get:

$$V_2 = e'(g_1,g_2)^{\frac{S'S_{Es_i}}{S_u}\times L_1\times(R'x_1+S_u)+S'R'S_{Es_i}} \quad (6)$$

Based on the results V_1 (eq.5) and V_2 (eq.6), the server computes $V = V_1 \times V_2$ as:

$$V = e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_u}L_2(R'x_1+S_u)+\frac{S_{Es_i}S'}{S_u}L_1(R'x_2+S_u)+R'S_{Es_i}S'}$$

By taking $\frac{S_{Es_i}S'}{S_u}$ as a common factor from the first two factors of the power, we get:

$$V = e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_u}\times(L_2(R'x_2+S_u)+L_1(R'x_1+S_u))+R'S_{Es_i}S'}$$

Finally, given the public parameter $PP_4 = g_2^{S_{Es_i}S'}$ and the value $g_1^{R'}$ submitted with the anonymous token, the server computes:

$$V' = \frac{V}{e'(g_1^{R'}, PP_4)} = \frac{V}{e'(g_1^{R'}, g_2^{S_{Es_i}S'})} = \frac{e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_u}\times(L_1(R'x_1+S_u)+L_2(R'x_2+S_u))+R'S_{Es_i}S'}}{e'(g_1,g_2)^{R'S_{Es_i}S'}}$$

By eliminating $e'(g_1,g_2)^{R'S_{Es_i}S'}$ form the numerator and the denominator, we get:

$$V' = e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_u}\times(L_1(R'x_1+S_u)+L_2(R'x_2+S_u))} \quad (7)$$

Based on Shamir secret sharing method described in sub-section IV-B, providing two points generated through the same polynomial $P_R(x)$ of degree 1, allows to reconstruct $P_R(x)$.

The reconstruction process is done through polynomial interpolation as follows:

$$P_R(x) = \sum_{i=1}^{2} Y_i \times L_i(x)$$

By considering the interpolation at $x = 0$, we get:

$$P_R(0) = \sum_{i=1}^{2} Y_i \times L_i(0)$$

Where:

$$L_i(0) = \frac{-x_j}{x_i - x_j},\ i,j \in \{1,2\}, i \neq j$$
$$\text{And } Y_i = P_R(x_i),\ i \in \{1,2\}$$

Recall that in the setup phase of our protocol (defined in sub-section VI-C1), L_1 and L_2 are defined as follows:

$$L_1 = \frac{-x_2}{x_1 - x_2},\ L_2 = \frac{-x_1}{x_2 - x_1}$$

From eq.7, we can clearly see that the factor $L_1(R'x_1 + S_u) + L_2(R'x_2 + S_u)$ (available in the power) is nothing else than the polynomial interpolation of the points $\{(x_1, P_R(x_1), (x_2, P_R(x_2))\}$ and which results $P_R(0) = S_u$.

Therefore, eq.7 results:

$$V' = e'(g_1,g_2)^{\frac{S_{Es_i}S'}{S_u}\times S_u}$$

By eliminating S_u from the power, we get:

$$V' = e'(g_1,g_2)^{S_{Es_i}S'}$$

which is stated in claim 1.

To prove the correctness of claim 2, let us recall that during information sharing process, user u submites a signature s along with the anonymous token T. The signature s is expressed as follows:

$$s = R' \times (A - e \times B)$$

Where, $A = \frac{S'}{S_jL_2}$ and $B = \frac{S'}{S_jL_1}$.

Once the server Es_i verifies the authenticity of the token T, it verifies that the same token T has been used to sign the shared information. Therefore, given the signature s the server computes:

$$r_v = g_1^s = g_1^{R'\times(A-e\times B)} = g_1^{\frac{R'S'}{S_uL_2}-\frac{R'S'}{S_uL_1}\times e}$$

Given the value Y_1 available in the anonymous token T, the server computes:

$$e_v = H(Y_1||D)$$

Where D represents the shared data.

The verification of the signature consists to compare between two values C_1 and C_2. Given the public parameter $PP_5 = g_2^{\frac{-1}{Kx_1}}$, the value C_1 is expressed as follows:

$$C_1 = e'(Y_1, PP_5) = e'(g_1^{\frac{R'KS'x_2}{S_uL_1}}, g_2^{\frac{-1}{Kx_1}})$$

Recall that:

$$\frac{1}{L_1} = \frac{x_2 - x_1}{x_2} \quad (8)$$

$$\text{and } \frac{1}{L_2} = \frac{x_1 - x_2}{x_1} \quad (9)$$

Thus, by represting $\frac{1}{L_1}$ as expressed in eq.8, we get:

$$C_1 = e'(g_1, g_2)^{\frac{R'KS'x_2}{S_u} \times \frac{x_2 - x_1}{x_2} \times \frac{-1}{Kx_1}}$$

By eliminating K and x_2 from the power, we get:

$$C_1 = e'(g_1, g_2)^{\frac{R'S'}{S_u} \times \frac{x_1 - x_2}{x_1}}$$

As we can notice, the factor $\frac{x_1 - x_2}{x_1}$ appearing in the power is nothing else than $\frac{1}{L_2}$ (according to eq.9). Therefore, C_1 can be represented as:

$$C_1 = e'(g_1, g_2)^{\frac{R'S'}{S_uL_2}} \quad (10)$$

Given the public parameter $PP_6 = g_2^{\frac{1}{Kx_2}}$, the value C_2 involved in the signature verification process is computed as follows:

$$\begin{aligned} C_2 &= e'(r_v, g_2) \times e'(Y_1^{e_v}, PP_6) \\ &= e'(g_1, g_2)^{\frac{R'S'}{S_uL_2} - \frac{R'S'}{S_uL_1} \times e + \frac{R'KS'x_2}{S_uL_1} \times e_v \times \frac{1}{Kx_2}} \end{aligned}$$

By eliminating K and x_2 from the the third factor appearing in the power, we get:

$$\begin{aligned} C_2 &= e'(r_v, g_2) \times e'(Y_1^{e_v}, PP_6) \\ &= e'(g_1, g_2)^{\frac{R'S'}{S_uL_2} - \frac{R'S'}{S_uL_1} \times e + \frac{R'S'}{S_uL_1} \times e_v} \end{aligned}$$

Recall that in whenever user u signs the shared information with the anonymous token T, e is supposed to be equal to $H(Y_1||D)$ which is equal to e_v. Therefore, C_2 becomes:

$$C_2 = e'(g_1, g_2)^{\frac{R'S'}{S_uL_2} - \frac{R'S'}{S_uL_1} \times e + \frac{R'S'}{S_uL_1} \times e}$$

By eliminating $\frac{R'S'}{S_uL_1} \times e$ from the power, we get:

$$C_2 = e'(g_1, g_2)^{\frac{R'S'}{S_uL_2}} \quad (11)$$

As we can notice, when we consider that e and e_v are equal, the results of equations 10 and 11 become equal as well, which refers to a succesfull verification of the signature. Otherwise, when e is not equal to e_v, it means that the user u did not sign the shared information with token T and thus, eq.10 is not equal eq.11, which refers to a failure in the verification process.

To prove the correctness of claim 3, let us recall that the authority in our protocol is the only entity that holds the master key S' and the values L_1 and L_2. Given the signature (s, e) submitted with the shared information, the authority computes:

$$T' = \frac{s}{S' \times (\frac{1}{L_2} - \frac{1}{L_1} \times e)} = \frac{\frac{S'R'}{S_u} \times (\frac{1}{L_2} - \frac{1}{L_1} \times e)}{S' \times (\frac{1}{L_2} - \frac{1}{L_1} \times e)}$$

Given the parameter $g_1^{R'}$ submitted in the anonymous token T, the user u's trace Tr_u as:

$$Tr_u = (g_1^{R'})^{\frac{1}{T'}} = g_1^{R' \times \frac{S_u}{R'}} = g_1^{S_u}$$

which is stated in claim 3.