Privacy-preserving data aggregation in Intelligent Transportation Systems

Cătălin Goşman*, Ciprian Dobre*, Florin Pop*
*University Politehnica of Bucharest
Spl. Independentei 313, Bucharest, Romania
E-mails: catalin.gosman@cti.pub.ro, ciprian.dobre, florin.pop@cs.pub.ro

***Abstract*—Intelligent Transportation Systems (ITS) demonstrate innovative services relating to different modes of transport and traffic management. For this, ITS rely on data collected by volunteer users in traffic. The aggregation of information collected periodically offers the possibility to learn general statistics and provides ITS users with a common traffic view. For user's sensitive data, it is desired to hide individual values from other participants, but also from the ITS data aggregator. Therefore, in this paper, we propose a schema for privacy-preserving aggregation based on symmetric cryptography of time-series data applicable for ITS services/applications.**

I. Introduction

Intelligent transportation systems (ITS) rely on data collected from various sensors available within the road infrastructure or contributed by volunteer users, to derive models of traffic, automate recognition of transport related safety events, with the end objective of reducing congestions, making traffic safer, monitor and optimize transportation for car fleets, etc.

Participants periodically share their personal collected information for the construction and maintenance of ITS services. Using this data, an aggregator computes results for each time interval. A mandatory condition for the data aggregator is that it can only calculate and use the aggregated result obtained from all users (and nothing else, like individual user data) without any further interaction with the users. Many existing privacy preserving time-series data aggregation schemas focus on the sum, since it is the essential statistic used in many ITS services such as fastest route, parking lot discovery, etc. For example, authors in [1] assume users send their data to the data aggregator in an encrypted form. Then, the aggregator decrypts the sum interacting with users but without learning the data of any individual. Besides the impractical user-interaction ([1], [2], [3]), current privacy-relatd schemas for data aggregation of time-series share an important drawback: they rely on the trustworthiness of the data aggregator.

Therefore, it is necessary to create privacy schemas in which undesirable changes in the data at the aggregator side (due to maliciousness or malfunctioning) can be noticed by any verifier. Otherwise, data aggregation raises important security risks in ITS ([4], [5]). For example, an adversarial ITS aggregator can distribute incorrect aggregating result to ITS users without anyone noticing it. Wrongly calculated outputs can lead to incorrect routing results or false parking discovery lots. An adversarial aggregator can output a fake sum without any ITS source noticing anything. Therefore, the way to improve reliability of the data aggregation schema is to make the aggregation result verifiable ([6], [7]). For example, for a traffic navigation system, the aggregator can output traffic routes that can be verified by users who contributed with personal data to the aggregated result. In this manner, the result is verified and confirmed as the aggregate of the values sent by the users. So, undesirable changes in the data on the server side can be noticed by any verifier/ITS source.

978-1-5090-0058-6/16/$31.00 © 2016 IEEE

The data aggregator is usually considered a reliable component that provides correct results. In this paper, we assume that the data aggregator can be compromised, as it can provide incorrect or misleading information. Therefore, under this circumstances, *we propose a schema for privacy-preserving aggregation of time-series data in the presence of an untrusted aggregator*. Contributing ITS participants collectively add a small noise to the aggregated result to ensure differential privacy. The schema leverages on a ring-based interleaved grouping technique that deals with ITS users' dynamic joins and leaves. Specifically, when an ITS user joins or leaves the system, only a small number of other participants need to update their cryptographic keys.

We continue our previous work on the topic. In [8] we proposed a new model for users to define data sharing policies for their personal information captured by an ITS. Under this model, ITS applications might need to use specific information under various quality and context-based constraints. On the other hand, the user-oriented security policies could specify additional usage constraints. We propose a solution to mediate between the two sides. Our approach ensures that users have the possibility to control access to sensitive data before is is shared with various ITS applications. In [9], we proposed a solution to quantify the level of trust in information for participants sharing data in ITS. Without being able to determine the level of trust for the information disseminated in ITS, it is difficult to choose whether or not to use the data. In the proposed mechanism we consider all shared data reported about different events in the system, together with their information quality parameters.

The rest of the paper is structured as follows: In Section II we present our approach for protecting user privacy. We present our results in Section III. Section IV presents conclusions and future work.

II. Privacy preserving time-series data aggregation schema

A. Overview

The data is being provided by ITS sources S_i . In order to protect their data, sources S_i add noise to their shared information. The realistic assumption when constructing an ITS service is that the aggregator is untrusted. A number of sources S_i in ITS may collude with the data aggregator. So, the aggregator can expose sources' shared data together with their noise values. We assume that the maximum fraction of compromised sources in ITS is γ. We also assume the existance of an entity, key dealer, which issues keys to the sources and to the aggregator using secure communication. In order to construct an ITS service, an aggregator has to obtain the aggregated result of n sources S_i periodically. The sum aggregate for time period t is $\sum_{i=1}^{i=n}(x_i(t))$. Since the accurate sum may leak source privacy in presence of side information, the aggregator is only allowed to obtain a noisy sum (i.e., the accurate sum plus some noise). In each time period t, each source S_i adds noise r_i i to her data x_i, encrypts the noisy data $\hat{x}_i$(t) = x_i(t) +r_i(t) with her key k_i(t) and sends the ciphertext to the aggregator. The aggregator uses the capability k_0 to decrypt the noisy sum $\sum_{i=1}^{i=n}(x_i(t)+r_i(t))$. Here, k_i(t) and k_0(t) change in every time period. In order to ensure privacy for sources S_i that share data, but also in order to create a mechanism through which sources can verify if the aggregator can be trusted during the creation of ITS services, the following conditions need to be met: first, the aggregator only learns the noisy sum but nothing else (e.g., intermediate results). Second, a source S_i different than the aggregator learns nothing about the information shared in ITS. Thirdly, the sum obtained by the aggregator should be roughly the same no matter if a specific source S_i in ITS is in the system or not.

B. Initial privacy schema

The key dealer generates a set K of nc random secrets s_1,..., s_{nc}. It divides them into n random disjoint subsets K_1,..., K_n. Each group has assigned c secrets. Clearly, K= $\bigcup K_i$. The key dealer randomly selects a subset $\hat{K}$ of q secrets and assigns them to the aggregator. Then, it evenly divides $K-\hat{K}$ into n random disjoint subsets $\overline{K_1}$,..., $\overline{K_n}$. So, K = $(\bigcup\overline{K_i})\bigcup\hat{K}$. A source S_i in ITS has assigned the secrets in K_i and $\overline{K_i}$.

Shared data encryption In order to protect its data, in time period t, source S_i generates key:

$$k_i = (\sum_{\forall s'\in K_i} h(f_{s'}(t) - \sum_{\forall s'\in \overline{K_i}} h(f_{s'}(t))modM.(1) \qquad (1)$$

$$M = 2^{[log_2 n\delta]} \qquad (2)$$

We consider that $f_{s'}(t)$ is the HMAC of t with s' as the key.

$$h : f_{s'} \rightarrow [0, M-1] \qquad (3)$$

A simple construction for h is to truncate the output of $f_{s'}$ into shorter bit strings of length $\log_2 M$ and use the exclusive-OR of all these strings as the output. Source S_i encrypts its noisy data $\hat{x}_i$ by computing c_i = (k_i + $\hat{x}_i$) mod M.

Shared data decryption In time period t , the aggregator generates key:

$$k_0 = (\sum_{\forall s'\in\hat{K}} h(f_{s'}(t))modM. \qquad (4)$$

and decrypts the noisy sum $\hat{K}=\sum_{i=1}^{i=n}\hat{x}_i$ by computing $\hat{K}=(\sum_{i=1}^{i=n}c_i-k_0)modM$. It is easy to verify that $k_0=(\sum_{i=1}^{i=n}k_i)modM$. In order to add noise to each source S_i in ITS we are using geometric distribution [10]. Let $\alpha>1$. Geom(α) denotes the symmetric geometric distribution with parameter α. Let $0<\beta\leq 1$. A random variable follows β-diluted Geometric distribution $Geom^{\beta}(\alpha)$ if it is sampled from Geom(α) with probability β, and is set to 0 with probability 1 - β. Parameters α and β are set as:

$$\alpha = e^{\frac{\varepsilon}{\delta}} \qquad (5)$$

$$\beta = min(\frac{1}{(1-\gamma)n} ln\frac{1}{\delta}, 1) \qquad (6)$$

where ε and δ are privacy parameters.

Given that during the decryption the aggregator only learns the noisy sum but nothing else, the data perturbation procedure achieves (ε,δ)-differential privacy. Having this mechanism in place, the aggregator can get the correct noisy sum. Also, no other source S_i in ITS can learn the sum, since only the trusted key dealer and the aggregator know the secrets used by the aggregator. The technique is efficient in computation due to the use of HMAC. The encryption/decryption mechanism depends on how large c and q are. If each source S_i in ITS is assigned a large-enough number of secrets (also this is available for the aggregator), it becomes impossible for an adversary to discover the secrets assigned to a particular source S_i in ITS or the aggregator. More formally, for l-bit security, the probability of a successful guess is smaller than 2^{-l}.

The mechanism described does not take into consideration frequent joins and leaves specific to ITS. Changing keys for all sources existing in ITS is not feasible due to high communication costs. Thus, next, we are describing an efficient technique to deal with joins/leaves specific in ITS.

C. Interleaved Grouping

In order to deal with dynamic joins and leaves specific in ITS, we have chosen interleaved grouping for our privacy-preserving schema for aggregation of time-series data. The schema is based on the studies described in [10]. Interleaved group structures have the property that each group shares some of its component sources S_i with other groups. Those sources S_i which are part of multiple groups will receive secrets from all groups in which they take part. In order to calculate the encryption key (decryption capability), a source S_i that shares data uses all the secrets it has been assigned from all the groups it belongs to, unioning them. The encryption (decryption) process is the same as the one described above. The aggregator has the cability to decrypt the aggregate sum. Additionally, the structure of interleaved grouping guarantees that the aggregator cannot learn the sum of any individual group or the sum of any subset of (not all) sources S_i. Sources S_i collectively add just one copy of geometric noise to the aggregated data, minimizing the aggregation error. Having an interleaved grouping structure provides benefits in case of dynamic joins/leaves from the group: the key dealer runs the setup phase again only for the affected group. Therefore, the communication cost for obtaining the new keys is low.

In order to construct interleaved groups and spread accordingly the sources S_i to different interleaved groups, we will use the following contruction condition 1: Let S denote an arbitrary strict and non-empty subset of groups. There exists a good source S_k, group $G \in S$ and group $G' \notin S$, such that $S_k \in G$ and $S_k \in G'$ [10]. It is nontrivial to satisfy this condition taking into consideration the formal structure of interleaved groups. With a total of g groups, there are about 2^g possible subsets of groups. Unless g is very small, it is infeasible to adjust each possible subset to satisfy the condition.

D. *Ring-based group structure*

In order to construct interleaved grouping we are setting up the ITS sources S_i into a ring structure. The ring structure is reflected by two virtual rings, the outer ring and inner ring. Each virtual ring is divided into segments. The ITS sources S_i that belong to a specific segment compose a group. Each ITS source S_i belongs to two groups, one group is in the inner ring, and one group is in the other ring. Groups in the same virtual ring do not intersect each other. According to the interleaved structure, groups in inner and outer rings may overlap (i.e., they share at least one source S_k).

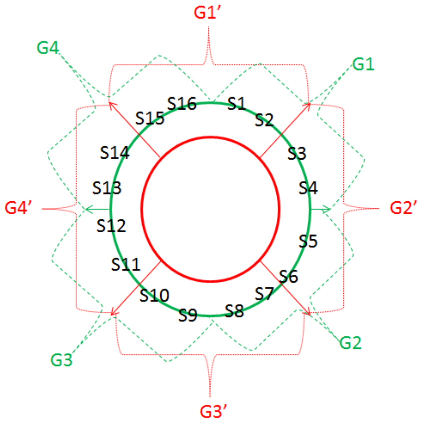

Fig. 1. Interleaved grouping. Groups on different rings may overlap. Group G1 and G1' overlap, sharing ITS sources S1 and S2.

Let's suppose there are $n \geq 2d$ sources S_i in the ring. According to the construction mechanism, for group G, l' and r denote the indexes of the leftmost and rightmost ITS source in the group. The boundaries of group G are: G.left = $l' - 0.5$ and G.right = $r + 0.5$, respectively. The number of ITS sources in G represents the group cardinality, abbreviated with $|G|$. Group G's cardinality is $|G| = (G.right - G.left) mod n$. Considering the interleaved grouping property specific for the ring based model, we must be aware of the number of sources S_i that can overlap between groups. Therefore, we will consider that any two groups that overlap share at least x sources. x is large enough to make it infeasible (i.e., with probability smaller than 2^{-l}) that none of the shared sources is valid. Also, due to the interleaved structure, if two neighboring ITS sources in the ring belong to two neighboring groups in one virtual ring, they belong to the same group in the other virtual ring. Regarding groups cardinality, each group has d ($d > 2x$) to $2d - 1$ sources. To minimize the communication cost of dynamic joins and leaves, parameter d can be set as the minimum value that satisfies $d > 2x$, i.e., $d = 2x + 1$. The algorithm below is used for group creation [10].

Algorithm 1 Group initialization

```
Require: ITS sources $0,\ldots,n$ - 1. Without loss of generality, $n$ is divisible by $d$.
Ensure: $\frac{2n}{d}$ groups $G_1,\ldots, G_{\frac{n}{d}}, G'_1,\ldots,G'_{\frac{n}{d}}$, each of size $d$.
FOR  i=1,$\ldots$,$\frac{n}{d}$
Add ITS sources: ($i$ - 1)$\frac{n}{d}$ , ($i$ - 1) $\frac{n}{d}$ + 1,$\ldots$, ($i$ - 1) $\frac{n}{d}$ + $d$ - 1 to group $G_i$
Add ITS sources ($i$- 1)$\frac{n}{d}$ + $\frac{d}{2}$ mod $n$, ($i$- 1)$\frac{n}{d}$ + $\frac{d}{2}$ +1 mod $n$,$\ldots$, ($i$- 1) $\frac{n}{d}$ + $\frac{d}{2}$ +$d$ - 1 mod $n$ to group $G'_i$
```

The key distribution is set up in accordance to the interleaved grouping model. Each ITS source gets secrets from the two groups that it is in. Let K_{i1} and $\overline{K_{i1}}$ denote the sets of secrets received from one group, and K_{i2} and $\overline{K_{i2}}$ denote the sets of secrets received from the other group. The ITS source S_k merges the secrets as follows: $K_i = K_{i1} \cup K_{i2}$ and $\overline{K_i} = \overline{K_{i1}} \cup \overline{K_{i2}}$. Let g denote the number of groups generated by algorithm 1. The aggregator obtains $\hat{K}_1,\ldots,\hat{K}_g$, where each $\hat{K}_j$ is the decryption capability of one group. It sets the overall decryption capability as $\hat{K} = \cup \hat{K}_j$. The encryption and decryption is done as in the original schema presented.

We can notice that after construction, each group overlaps with two other groups and shares $\frac{d}{2}$ ITS sources with each overlapping group. When a new ITS source joins and contributes with data for the creation of the ITS service/application, it is inserted to a random location in the ring, and added to the two groups that cover the location of insertion. However, the join activity of ITS sources may violate the group size property: we may end up having groups that contain more than $2d$ - 1 ITS sources. Similarly, when a ITS source S_k leaves, it is removed from the ring and from the two groups it belongs to, which may violate the group size property and overlap property. In these cases, the sources should be regrouped so that the three properties still hold. The algorithms below solves the dynamic joins and leaves of ITS sources [10]. Multiple cases must be analyzed due to the ring based model.

Algorithm 2: Re-grouping after a ITS source S_k joins two groups overlapping in the pattern shown in figure 2.

```
Require: G, A: two groups that ITS source $S_k$ joins, $|G| > |A|$.
if $|G| < 2d$ then return;
else split G in the middle into two groups, each of size $d$;
```

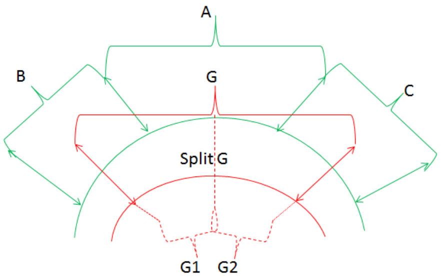

Fig. 2. Re-grouping when a source joins G and A which overlap. G is split into G1 and G2 if it has too many sources.

In ring-based interleaved grouping, when a source joins (leaves), at most three (four) existing groups are updated and the number of updated sources has an upper bound $4d$ ($6d$).

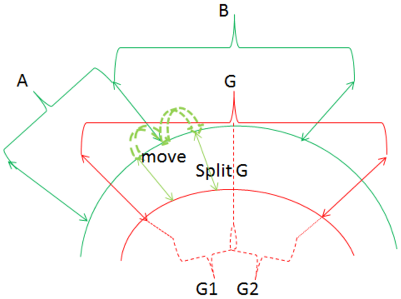

Fig. 3. Re-grouping when a source joins G and A which overlap. B is the neighbor of A; it overlaps with G.

```
Algorithm 3 adjust(JOIN, II): Re-grouping after a ITS source $S_k$ joins two groups overlapping in the pattern in figure 3.
Require: G, A: two groups that ITS source $S_k$ joins, $|G| \geq |A|$.
Require: B: A's neighbor group which also overlaps with G.
Require: Without loss of generality, G overlaps with the right part of A, i.e., $G.left > A.left$ and $G.right > A.right$. In this case, B is the right neighbor of A.
if $|G| < 2d$ then return;
else split G in the middle into two groups, each of size $d$;
move A.right to position P = max(G.left + $x$, A.left + $d$);
if $|B| < 2d$ then return;
else create group C, with C.left = A.right and C.right = C.left + $d$ and $B.left \leftarrow C.right$;
```

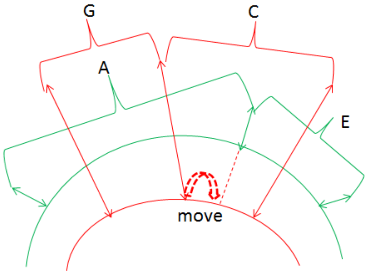

Fig. 4. Re-grouping when a source leaves G and A which overlap.

```
Algorithm 4: Re-grouping after a ITS source $S_k$ leaves two groups which overlap in the pattern shown in figure 4.
Require: G, A: the groups that source leaves, with $|G| < |A|$.
Require: C, E: the right neighbor of G and A, respectively.
if $|G| \geq d$ then return;
else if $|C| = d$ then merge G and C;
else $u \leftarrow |C \cap A|$;
if $u \geq x+1$ then move G.right to the right by 1;
else if $u$ = $x$ and $|C| \geq d+2x$ move G.right to right by $2x$;
else if $u$ = $x$ and $|C| < d+2x$ then move both G.right and A.right to the right by 1;
```

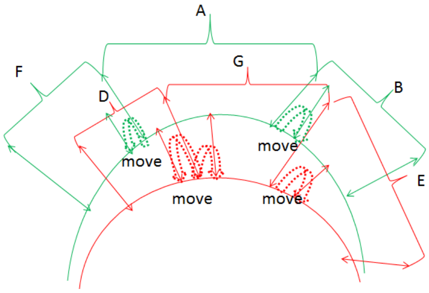

Fig. 5. Re-grouping when a source leaves G and A which overlap.

When an source joins/leaves, the key dealer runs algorithms 2-5 to adjust grouping. The setup phase for a new key distribution is re-run only for the group(s) affected. Because it affects only a small portion of all sources, it is not expensive.

Parameter β used when calculating the noise added for the aggregated result is based on the total number of ITS sources n. Considering ITS is a dynamic system in which sources frequently join and leave, the interleaved grouping benefits in communication costs are nullified if we are to consider the exact value of n. In order to obtain a tradeoff between the communication cost in case of joins/leaves and the value of n needed for calculating the noise for the aggregated result, we are changing the basic schema by adding a small extra noise. Specifically, each ITS source records the value of n according to its knowledge. Let u denote the value recorded by the ITS source S_k. This may be different compared with real number of ITS source in the system. However, each source uses u to set parameter β for data perturbation, i.e., $\beta = min(\frac{1}{(1-\gamma)u} ln\frac{1}{\delta}, 1)$. The noise varies in accordance with the reflected value of u. An ITS source S_k will add more noise, therefore more aggregation error, if it will not observe that there are leaves in the ITS. However, u should not be set higher than n to ensure that at least one copy of geometric noise is added to provide differential privacy. Thus, the values of u reflected at the level of ITS source S_k should be updated in accordance with the dynamic joins or leaves in the ITS, but without incurring too much communication cost.

Algorithm 6 guarantees that $\forall i, u_i \in (\frac{n}{2}, n]$. Since $u \leq n$, at least one copy of geometric noise is added to the sum aggregate to provide differential privacy. Since $u > \frac{n}{2}$, at most one more copy of geometric noise is added. Thus, the average aggregation error is roughly within twice of the geometric

Algorithm 5 adjust: Re-grouping after a ITS source S_k leaves two groups which overlap in the pattern shown in figure 5.

```
Require: G, A: the two groups that the ITS source $S_k$ leaves.
Require: D, E: G's neighbors. D overlaps with A; E does not.
Require: B, F: A's neighbors. B overlaps with G; F does not.
Require: Without loss of generality, suppose $G.left > A.left$ and $G.right > A.right$.
if $|G \cap A| \geq x$ then
if $|G| \geq d$ and $|A| \geq d$ then return;
if $|G| = d-1$ then
if $|E| = d$ then merge G and E;
else move G.right to the right by 1;
if $|A| = d-1$ then
if $|F| = d$ then merge A and F;
else move A.left to the left by 1;
else if $|G| \geq d$ and $|A| \geq d$ then
if $|B| \geq d+1$ then move A.right to the right by 1;
else if $|D| \geq d+1$ then move G.left to the left by 1;
else move D.right to the right by $2x$ - 1;
if $|G| = d-1$ then
if $|D| = d$ then merge G and D;
else move G.left to the left by 1;
if $|A| = d-1$ then
if $|B| = d$ then merge A and B;
else move A.right to the right by 1;
```

Algorithm 6 Procedures run by the key dealer to manage the values of u for ITS sources [10].

```
Require: $n$: the real number of ITS sources
Require: $u_i$: the number of ITS sources that source $S_i$ uses to set parameter $\beta$
Initialization:
if $n$ is even then
$u_1, u_2, \ldots, u_n \leftarrow [\frac{n}{2}]+1, [\frac{n}{2}]+1, [\frac{n}{2}]+2, [\frac{n}{2}]+2, \ldots, n, n$;
else: $u_1, u_2, \ldots, u_n \leftarrow [\frac{n}{2}]+1, [\frac{n}{2}]+2, [\frac{n}{2}]+2, \ldots., n, n$;
Join: if ITS source $S_i$ joins then: $n \leftarrow n+1$ and $u_i \leftarrow n$;
Find an ITS source $S_j$ with $u_j$ = min($u_1$, $u_2$,..., $u_n$); then $u_j \leftarrow n$;
Leave: if ITS source $S_i$ leaves then: $n \leftarrow n-1$;
Find an ITS source $S_j$ with $u_j$ = max($u_1$, $u_2$,..., $u_n$);
if there exists another ITS source $S_m$ with $u_m = u_j$ then: $u_m \leftarrow u_i$ and $u_j \leftarrow [\frac{n}{2}]+1$;
```

noise required for differential privacy. Algorithm 6 only incurs very small communication cost. When a ITS source joins, the joining source and another source with the minimum u are updated; when a ITS source leaves, (at most) two remaining sources with the maximum u are updated. Thus, the u of at most two sources are updated for each join or leave.

III. Validation and results for the proposed schema

A. Comparison schemas and results

In order to evaluate our proposed solution, we look by comparison at the following approaches: the schema proposed in [11] (denoted by SCRCS), the Binary schema [12], the schema proposed in [13] (denoted by JK). The communication cost of the proposed schema is smaller than SCRCS and Naive Grouping, and is close to Binary and JK. This is because the ring-based interleaved grouping can effectively reduce the number of ITS sources that should be updated for dynamic joins/leaves. Also, the communication cost of our schema does not change with parameter n, hence it can scale to large systems. The aggregation error is O(1) for all schemas, except for Binary schema, where it is O($(logn)^{\frac{3}{2}}$). In case of ITS dynamic joins/leaves, binary and [13] have O(1) communication cost, while our schema has O(d) costs. However, these schemas are based on public-key encryption, which is not appropriate for mobile sensing scenarios with resource-constrained devices, short aggregation periods, and many aggregate statistics collected simultaneously. Our proposed schema is based on symmetric-key cryptography. Moreover, the computation in our proposed schema is efficient since it is based on HMAC.

TABLE I
COMPUTATIONAL COST OF DIFFERENT SCHEMA

	ITS sources	Aggregator
Proposed schema	4c Pseudorandom function	$\frac{2nq}{d}$ Pseudorandom function
JK	2 Paillier encryptions and 1 signature generation	2 Paillier decryptions and 2n mod. multiplication and n sig. verifications
SCRS	2 mod. exp.	$\sqrt{n\Delta}$ mod. exp.
Binary	2([$log_2 n$] + 1) mod. ep.	$\sqrt{n\Delta}$ mod. exp.

The table shows the computation overhead of these schemas (see [6], [11], [12], [13] for details). HMAC-SHA256 is used as the pseudorandom function of the proposed schema. For JK, the Paillier cryptosystem uses a 1024-bit modulus, and RSA (1024-bit) is used as the digital signature algorithm. For SCRCS and Binary, the high-speed elliptic curve 'curve25519' is used for modular exponentiation. Compared with JK, the proposed schema is faster in both encryption and decryption. This is because JK uses the Paillier cryptosystem which is very expensive in computation, while the proposed schema is based on HMAC which is very efficient. Compared with SCRCS (Binary), the proposed schema is faster in encryption and decryption when the plaintext space is 1000 or larger. SCRCS and Binary are very slow in decryption because they need to traverse the possible plaintext space to decrypt sum, computing one modular exponentiation for each possible value. The communication cost of ring-based interleaved grouping is affected by parameter γ (i.e., the maximum fraction of sources compromised). Parameter γ depends on x and d. Therefore, if x is larger, the group size is larger, γ and the communication cost increase. The variance of the number of sources in ITS has impact on the aggregation error. Binary and Naive Grouping schemas have direct relationship with the number of sources in ITS: in case the number of sources increases, the aggregation error of Binary and Naive Grouping also increases quickly.

On the other side, this variance does not influence very much the proposed solution, neither SCRCS and JK. The reason is that all these schemas add on average a constant number of copies of geometric noise to the aggregate. If we are to look for stricter requirements for differential privacy, the privacy parameter γ must be decreased. In this case, all schemas have a higher aggregation error, since each ITS source needs to add more noise to meet the stronger privacy requirement.

B. Relaxing the trusted key dealer assumption

The proposed solution is based on a trusted key dealer. However, we can relax this condition without loss of generality. We can assume that the key dealer may attempt to obtain some data values from ITS sources by knowing how the proposed schema works, but also from eavesdropping communications in ITS. Considering this, we can protect our schema by adding one extra encryption/decryption to the data that each ITS sources submits to the aggregator. Specifically, each source first encrypts its noisy data as previously specified with the secrets received from the key dealer, deriving an intermediate result c. The intermediary results is encrypted using a pre-shared key shared with the aggregator. This final ciphertext is sent by the ITS source to the aggregator. The aggregator decrypts the results by performing the operations in a reverse manner: first, it decrypts each source' intermediate result c using the pre-shared key. Afterwards, the aggregator decrypts the noisy sum as it was described in the initial schema. Therefore, the proposed schema does not need to be based on a trusted key dealer as long as the key dealer does not collude with the aggregator. The reason is that the key dealer is unable to understand the intermediate result c. Therefore it can't obtain the actual source information.

C. Dealing with source failures

Looking at dynamic systems like ITS, source failure is an important factor that must be considered. In our proposed schema, the key dealer is the entity that knows the secrets assigned to every ITS source. In case an ITS source suffers a failure, for whatever reason, we must provide a way by which the aggregator manages to correctly obtain the results. The solution is based on the aggregator asking the key dealer to behave as the faulty source. Under these circumstance, the key dealer will act as the the faulty source and submit empty payload data with large noise, encrypt it as the faulty source would and send it to the aggregator. So, the aggregator can correctly decipher the information received. The extra cost for dealing with faulty sources is the fact that the key dealer must communicate with the aggregator. However, the complexity added to the communication is linear, as we have a finite number of n sources in ITS. If a source has failed for a long time, it can be removed from the system as a leave. The recovery of a failed source can be processed as a join.

IV. Conclusions

Verifiability of data aggregation is essential: without verifiability, the trustworthiness of the output can be questionable. We stress that being able to verify the correctness of the data aggregation is even more important when the underlying shared data is encrypted. In this paper, we have proposed a schema for privacy-preserving aggregation of time-series data in presence of untrusted aggregator, which provides differential privacy for the aggregated result. The privacy preserving time-series data aggregation schema facilitates ITS users to share data, but also illustrates a manner in which participants can verify the data aggregator output. The schema deals with dynamic joins and leaves specific to ITS. The proposed technique's efficiency in computation (due to HMAC) makes it a better choice for ITS applications with high aggregation loads (e.g., many parallel aggregation tasks per source or per aggregator, short aggregation period), resource-constraint mobile devices (e.g., personal healthcare devices besides smartphones) and rich statistics.

Acknowledgment

The research is supported by projects MobiWay: Mobility beyond Individualism (PN-II-PT-PCCA-2013-4-0321), and DataWay: Real-time Data Processing Platform for Smart Cities (PN-II-RU-TE-2014-4-2731).

References

[1] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 735–746.

[2] M. Joye and B. Libert, "A scalable scheme for privacy-preserving aggregation of time-series data," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 111–125.

[3] I. Leontiadis, K. Elkhiyaoui, and R. Molva, "Private and dynamic time-series data aggregation with trust relaxation," in *International Conference on Cryptology and Network Security*. Springer, 2014, pp. 305–320.

[4] D. Christin, "Privacy in mobile participatory sensing: current trends and future challenges," *Journal of Systems and Software*, vol. 116, pp. 57–68, 2016.

[5] B. G. Bakondi, A. Peter, M. Everts, P. Hartel, and W. Jonker, "Publicly verifiable private aggregation of time-series data," in *Availability, Reliability and Security (ARES), 2015 10th International Conference on*. IEEE, 2015, pp. 50–59.

[6] Q. Li and G. Cao, "Efficient and privacy-preserving data aggregation in mobile sensing," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2012, pp. 1–10.

[7] R. Zhang, J. Shi, Y. Zhang, and C. Zhang, "Verifiable privacy-preserving aggregation in people-centric urban sensing systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 268–278, 2013.

[8] C. Gosman, T. Cornea, C. Dobre, C. X. Mavromoustakis, and G. Mastorakis, "Security policies in intelligent transportation systems," in *The 18th Mediterranean Electrotechnical Conference (MELECON 2016), 2016. Proceedings.* IEEE, 2016.

[9] A. Castiglione, "Putting the user in control of the intelligent transportation system," in *Information Security and Privacy: 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings*, vol. 9722. Springer, 2016, p. 231.

[10] Q. Li and G. Cao, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in *Intern. Symp. on Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 60–81.

[11] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society., 2011.

[12] T.-H. H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," in *International Conference on Financial Cryptography and Data Security*. Springer, 2012, pp. 200–214.

[13] M. Jawurek and F. Kerschbaum, "Fault-tolerant privacy-preserving statistics," Nov. 4 2014, uS Patent 8,880,867.