Efficient and Privacy-Aware Data Aggregation in Mobile Sensing

Padma Saikrishna¹, B. Hari Krishna², Obulesh³

¹PG Scholar, Dept of CSE, Avanthi Institute of Engineering and Technology, Hyderabad, India, Email: Krishna.padmasai@gmail.com.
²Associate Professor & HOD, Dept of CSE, Avanthi Institute of Engineering and Technology, Hyderabad, India.
³Associate Professor, Dept of CSE, Avanthi Institute of Engineering and Technology, Hyderabad, India.

Abstract: The works on sensor data aggregation assume a trusted aggregator, and hence cannot protect user privacy against an untrusted aggregator in mobile sensing applications. Several recent works consider the aggregation of time series data in the presence of an untrusted aggregator. To protect user privacy, they design encryption schemes in which the aggregator can only decrypt the sum of all users’ data but nothing else. Rastogi and Nath use threshold Paillier cryptosystem to build such an encryption scheme. To decrypt the sum, their scheme needs an extra round of interaction between the aggregator and all users in every aggregation period, which means high communication cost and long delay. The proliferation and ever-increasing capabilities of mobile devices such as smart phones give rise to a variety of mobile sensing applications. This paper studies how an untrusted aggregator in mobile sensing can periodically obtain desired statistics over the data contributed by multiple mobile users, without compromising the privacy of each user. Although there are some existing works in this area, they either require bidirectional communications between the aggregator and mobile users in every aggregation period, or have high-computation overhead and cannot support large plaintext spaces. Also, they do not consider the Min aggregate, which is quite useful in mobile sensing. To address these problems, we propose an efficient protocol to obtain the Sum aggregate, which employs an additive homomorphic encryption and a novel key management technique to support large plaintext space. We also extend the sum aggregation protocol to obtain the Min aggregate of time-series data. To deal with dynamic joins and leaves of mobile users, we propose a scheme that utilizes the redundancy in security to reduce the communication cost for each join and leave. Evaluations show that our protocols are orders of magnitude faster than existing solutions, and it has much lower communication overhead.

Keywords: Mobile Sensing, Privacy, Data Aggregation.

I. INTRODUCTION

A sensor network consists of large number of sensors deployed in a region for the purpose of event monitoring or detection. The sensors are preprogrammed to listen for specific events. For example, a sensor network deployed in a high security region might be programmed to detect infrared heat signals to indicate an intruder. Figure 1 shows a typical sensor network deployment. Each node in a sensor network is responsible for observing and reporting various dynamic properties of their surroundings in a time critical manner. These mobile and miniaturized information devices are equipped with embedded processors, wireless communication circuitry, information storage capability, smart sensors and actuators. These sensor nodes networked in an ad hoc way, with little or no fixed network support, to provide the surveillance and targeting information for dynamic control. Sensor devices are mobile, subject to failure, deployed spontaneously and repositioned for more accurate surveillance. Despite these dynamic changes in configuration of the sensor network, critical real-time information must still be disseminated dynamically from mobile sensor data sources through the self-organizing network infrastructure to the components that control dynamic re-planning and re-optimization of the theatre of operation based on newly available information. With large number of sensor devices being quickly and flexibly deployed in most impromptu networks, each sensor device must be autonomous and capable of organizing itself in the overall community of sensors to perform coordinated activities with global objectives.

When spontaneously placed together in an environment, these sensor nodes should immediately know about the capabilities and functions of other sensor Nodes and work together as a community system to perform cooperative tasks and networking functionalities. Sensor networks need to be self-organizing since they are often formed spontaneously from large number of mixed types of nodes and may undergo frequent configuration changes. Some sensor nodes may provide networking and system services and resources to other sensor nodes. Others may detect the presence of these nodes and request services from them. The characteristics of sensor nodes necessary for creating self-organizing sensor networks are agility, self-awareness, self-configurability and autonomy. Sensor nodes with these features will have capabilities for self assembling impromptu networks that are incrementally extensible and dynamically adaptable to device failure and degradation, mobility of sensor nodes and changes in task and network requirements. Nodes are aware of their own capabilities and those of other nodes around them which
may provide the networking and system services or resources that they need. Although nodes are autonomous, they may cooperate with one another to disseminate information or assist each other in adapting to changes in the network configuration. An impromptu community of these nodes may cooperate to provide continual coordinated services while some nodes may be newly deployed or removed from the spontaneous community.

Nodes will act in response to environmental events and relay collected and possibly aggregated information through the multi-hop wireless network in accordance with desired system functionality. The inherently dynamic and distributed behavior of these networks, coupled with inherent physical limitations such as small instruction and data memory, constrained energy resources, short communication radii and a low bandwidth medium in which to communicate, make developing communication protocols difficult. Using these sensors as a basis for development, the software architecture and communication stack residing on these devices are built taking into consideration the prolific research in the areas of ad-hoc networking, data aggregation, cluster formation, distributed services, group formation, channel contention and power conservation. An event is an abstraction, identifying anything from a set of sensor readings, to the nodes processing capabilities. For the purpose of the simulation studies in this project, events are assumed to be localized phenomenon, occurring in a fixed region of space. This assumption will hold for a wide variety of sensor-net applications, since many external events are localized themselves.

II. SYSTEM ANALYSIS

A. Existing System

Implementation-based measurements show that operations at user and aggregator in our protocol are orders of magnitude faster than existing work. Although there are some existing works in this area, they either require bidirectional communications between the aggregator and mobile users in every aggregation period, or have high-computation overhead and cannot support large plaintext spaces. In these operations, the random subset of existing and remaining users is called helper users for convenience. This section compares our Sum and Min aggregation protocols against existing work.

B. Proposed System

Based on the Sum aggregation protocol, we also proposed two schemes to derive the Min aggregate of time-series data. We propose a new protocol for mobile sensing to obtain the sum aggregate of time-series data in the presence of an untreated aggregator. We propose an efficient protocol to obtain the Sum aggregate, which employs an additive homomorphism encryption and a novel key management technique to support large plaintext space. We propose a scheme that utilizes the redundancy in security to reduce the communication cost for each join and leave. We also propose a scheme that employs the redundancy in security to reduce the communication cost of dealing with dynamic joins and leaves. One building block of our solution is the additive homomorphism encryption scheme proposed by Castelluccia.

Advantages:
- It reduces the Communication cost of dealing with dynamic joins and leaves.
- Users may frequently join and leave in mobile sensing.
- In each time period, a mobile user sends her encrypted data to the aggregator via WiF, 3G or other available access networks.
- No peer-to-peer communication is required among mobile users, since such communication is nontrivial in mobile sensing scenarios due to the high mobility of users and users may not be aware of each other for privacy reasons.

Fig.1. System Architecture.

The intuition behind the straw man construction the aggregator computes the sum of a set of random numbers as the decryption key as shown in Fig.1. These numbers are secretly allocated to the users, and each user computes the sum of its allocated numbers as the encryption key. The aggregator does not know which random numbers are allocated to each user, and thus does not know any user’s key.

III. MODULE

A. Module Description

1. Mobile Sensing: This paper studies how a un trusted aggregator in mobile sensing can periodically obtain desired statistics over the data contributed by multiple mobile users, without compromising the privacy of each user. This enables various mobile sensing applications such as environmental monitoring, traffic monitoring, healthcare, and so on. Since such communication is nontrivial in mobile sensing scenarios
due to the high mobility of users and users may not be aware of each other for privacy reasons.

2. Privacy: Note that this paper focuses on thwarting attacks against users’ privacy. Our goal is to guarantee the privacy of each user’s data against the untrusted aggregator. Dynamic joins and leaves should be properly dealt with to protect each user’s privacy and ensure the secrecy of the aggregate statistics. Differential privacy for Sum differential privacy provides strong privacy guarantee for users such that a user’s participation in the system only leaks negligible information about the user.

3. Data Aggregation: The data generated by these sensors provide opportunities to make sophisticated inferences about not only people but also their surrounding and thus can help improve people’s health as well as life. In each time period, a mobile user sends her encrypted data to the aggregator via Wi-Fi, 3G or other available access networks.

4. Models and Assumptions: The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 presents system models and assumptions. An aggregator wishes to get the aggregate statistics of n mobile users periodically.

5. Underlying Encryption Scheme:
   • Represent message m as an integer within range [0, M-1], where M is a large integer.
   • Let k be a randomly generated key, k ∈ {0, 1}^λ, where λ is a security parameter.
   • Output ciphertext mod M, where f_k is a pseudorandom function (PRF) that uses k as a parameter, h is a length-matching hash function and r is a nonce for this message.

6. A Straw-Man Construction for Key Generation: The intuition behind the straw-man construction the aggregator computes the sum of a set of random numbers as the decryption key. These numbers are secretly allocated to the users, and each user computes the sum of its allocated numbers as the encryption key. The aggregator does not know which random numbers are allocated to each user, and thus does not know any user’s key.

7. Our Construction for Key Generation: The users collectively generate the summands on the left side and add them to the aggregate, while the aggregator alone generates the summands on the right side and subtracts them from the perturbed aggregate.

8. Low-Cost Min Aggregation: It may not be necessary to get the exact Min, but an approximate answer is good enough. For such scenarios, the basic scheme can be extended to get an approximate Min with much smaller cost. Only two instances are needed and each user only sends 128 bytes of cipher texts to the aggregator in each time period.

9. Dealing with Dynamic Joins and Leaves: We compare the communication cost of our scheme against EXP in dealing with dynamic joins and leaves. The computation cost of our scheme is increased when dealing with dynamic joins and leaves. To evaluate the computation cost, we measured the running time when there is redundancy in security. We also propose a scheme that employs the redundancy in security to reduce the communication cost of dealing with dynamic joins and leaves.

10. The Cost of Sum and Min Aggregation:
   • Cost of Sum Aggregation
   • Cost of Min Aggregation
   • Practical Performances

IV. EVALUATIONS
This section evaluates the cost of our aggregation protocols for Sum and Min. We compare our solution against three existing privacy-preserving aggregation protocols for time series data: the protocol proposed in [8] (denoted by EXP), Collapse [10], and the spatial aggregation protocol proposed (denoted by Spatial).

A. The Cost of Sum and Min Aggregation
This section compares our Sum and Min aggregation protocols against existing work.

Cost of Sum Aggregation: EXP is a Sum aggregation protocol based on the decisional Diffie-Hellman assumption. In EXP, encryption (decryption) requires two (√nA) modular exponentiations (see [8] for details). Similar to our protocol, Collapse also uses the homomorphic encryption scheme to derive Sum, but in a different way (see [10] for details). In Collapse, each user (the aggregator) computes s PRFs to encrypt her data (decrypt the sum). Here, s denotes the number of colluding users that the protocol can tolerate. Spatial is based on the Paillier cryptosystem. Table 1 shows the computation, storage, and communication cost of the four aggregation protocols for Sum, where the cost is derived under the same condition that they can tolerate γn colluding users. Compared with Collapse, our protocol has much smaller computation and storage cost at both the users and the aggregator, especially for a large system with possibly many colluding users. Compared with EXP and Spatial, our protocol has slightly higher storage cost (i.e., around 10 secrets each with just tens of bytes), but our computation overhead is much lower because in practice PRF (when implemented with HMAC) can run orders of magnitude faster than modular exponentiation, Pailler encryption, and Pailler decryption. We elaborate this point further in Section 4.1.3. Our protocol also has much lower communication cost than Spatial.

Cost of Min Aggregation: The Min aggregation scheme presented in derives Min from 2^((logΔ+2)/2) parallel Sum aggregates of 1-bit data, where each sum is obtained using our Sum aggregation protocol. Here, each sum can also be obtained using EXP, and we refer to the Min aggregation scheme that uses EXP as a building block as EXP-Min. In EXP-Min, each user computes 2^((logΔ+2)/2) modular exponentiations to encrypt her data, and the aggregator computes 2^((logΔ+2)/2) parallel Sum aggregates of 1-bit data, where each sum is obtained using our Sum aggregation protocol. Here, each sum can also be obtained using EXP, and we refer to the Min aggregation scheme that uses EXP as a building block as EXP-Min. In EXP-Min, each user computes 2^((logΔ+2)/2) parallel Sum aggregates of 1-bit data, where each sum is obtained using our Sum aggregation protocol. Here, each sum can also be obtained using EXP, and we refer to the Min aggregation scheme that uses EXP as a building block as EXP-Min. In EXP-Min, each user computes 2^((logΔ+2)/2) parallel Sum aggregates of 1-bit data, where each sum is obtained using our Sum aggregation protocol. Here, each sum can also be obtained using EXP, and we refer to the Min aggregation scheme that uses EXP as a building block as EXP-Min.
these protocols in Java. The cost is 2\(e\)−1(\(\gamma\)n+1)/(\(\log\) \(\Delta\)+2) PRFs for both encryption and decryption, considering that the concatenation technique also works for Collapse.

### TABLE I: Comparisons between Our Sum Aggregation Protocol and Existing Protocols

<table>
<thead>
<tr>
<th></th>
<th>Encryption (use)</th>
<th>Decryption (use)</th>
<th>Storage (use)</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>3 Mod. Exp.</td>
<td>3/4AS Mod. Exp.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>EXP2</td>
<td>((\gamma)+1) Prs</td>
<td>((\gamma)+1) Prs</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Spatial</td>
<td>1 Pallier Enc. + ((s)−1) Prs</td>
<td>1 Pallier Dec. + ((s)−1) Mod. Med</td>
<td>1</td>
<td>0(s)</td>
</tr>
<tr>
<td>Our</td>
<td>2 Prs</td>
<td>q/(\Phi)hs</td>
<td>2c</td>
<td>q</td>
</tr>
</tbody>
</table>

Mod. Exp. (Mul) stands for modular exponentiation (multiplications). In most practical settings, \(c<7\) and \(q<15\) (see Table 5 and 6).

\(\Delta\) is the plaintext space. Each user’s data value is from \([0, 1, \ldots, \Delta]\).

Compared with Collapse-Min, our Min aggregation scheme improves the computation cost of encryption and decryption by a factor of \(\frac{\gamma n+1}{2e}\) and \(\frac{\gamma n+1}{q}\), respectively. Our scheme is also much more efficient than EXP-Min in computation, as shown in Section 4.1.3.

### Practical Performances: In Table 1, the computation costs of our Sum aggregation protocol, EXP, and Spatial are measured by different units. Here, we elaborate the comparison between them with results in running time. For this purpose, we implemented these protocols in Java. The function of mobile user is implemented on Nexus S Smartphone with Android 4.0.4 OS, 1-GHz CPU, and 512-MB RAM. The function of aggregator is implemented on a Windows Laptop with 64-bit Windows 7 OS, 2.6-GHz CPU, and 4-GB RAM. For EXP, the elliptic curve “curve25519” is adopted for modular exponentiation. For Spatial, the Paillier cryptosystem uses a 1,024-bit modulus, and random numbers are generated using the standard function provided by Java. For our protocol, PRF is implemented with HMAC-SHA256.

Table 2 shows the running time of our Sum aggregation protocol, EXP, and Spatial. Our protocol is much faster than EXP and Spatial in both encryption and decryption. Specifically, encryption is at least one order of magnitude faster. When the plaintext space \(\Delta\geq10^2\), decryption is at least three orders of magnitudes faster. In our protocol, the computation cost decreases as the system scale increases, and it does not change with the plaintext space (so long as the size of plaintext data does not exceed the size of an HMAC output). Thus, our protocol can support large systems and large plaintext spaces. Table 3 shows the running time of our Min aggregation protocol and EXP-Min. Here, the plaintext space is set as \(\Delta=10^7\). The parameters of our protocol are set according to when \(\gamma=0.2\). In all the shown cases, our protocol is at least four (seven) orders of magnitude faster than EXP-Min in encryption (decryption). Especially, as the system scale increases, the running time of decryption in EXP-Min increases quickly, which shows the poor scalability of EXP-Min, but the running time of our protocol decreases and is always very low, which shows that our protocol is scalable.

### B. The Cost of Dynamic Joins and Leaves

This section evaluates our scheme for dealing with joins and leaves. Each data point is the average result of ten simulation runs with different random seeds.

### TABLE III: The Running Time of Our Min Aggregation Protocol and EXP-Min Which Uses EXP As a Building Block

<table>
<thead>
<tr>
<th></th>
<th>Encryption</th>
<th>Decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n=10^9)</td>
<td>1 Meg.</td>
<td>1 Meg.</td>
</tr>
<tr>
<td>(n=10^{10})</td>
<td>2 Meg.</td>
<td>2 Meg.</td>
</tr>
<tr>
<td>(n=10^{11})</td>
<td>3 Meg.</td>
<td>3 Meg.</td>
</tr>
<tr>
<td>(n=10^{12})</td>
<td>4 Meg.</td>
<td>4 Meg.</td>
</tr>
</tbody>
</table>

\(\Delta\) is the plaintext space. Each user’s data value is from \([0, 1, \ldots, \Delta]\).

\(\gamma\) in EXP-Min is set to 0.2. In all shown cases, our protocol is at least four (seven) orders of magnitude faster than EXP-Min.

**Fig. 2.** The effects of parameters \(K\), \(n\), and \(\gamma\) on the number of joins/leaves supported by each setup phase. By default, \(K = 50\), \(\gamma = 0.2\), and \(n = 3,000\) (i.e., the number of users at the beginning of simulation).

The Effect of \(K\), \(n\), and \(\gamma\): In our scheme, after the setup phase, a number of joins and leaves can be processed before the setup phase has to be run again. We first run simulations to evaluate the effect of parameters \(K\), \(n\), and \(\gamma\) on this number. Fig. 2 shows the results. As can be seen from Fig. 2a, when \(K\) increases, more joins and leaves can be supported (where only \(\phi'\) users are updated) without rerunning the setup phase again. This is because when \(K\) is larger there is more redundancy in security, and hence, the communication cost is lower. For similar reasons, when \(n\) increases, more joins and leaves can also be supported (see Fig. 2b). When \(\gamma\) increases, the setup phase needs to be run again sooner (see Fig. 2c). The reason is that when \(\gamma\) is higher, \(\phi'\) is larger, which means more users are updated and more secrets are transited from black to white in each join. The computation and storage overhead at mobile user linearly increases with \(K\). Thus, in practice, the value of \(K\) should be selected based on the...
Efficient and Privacy-Aware Data Aggregation in Mobile Sensing

To facilitate the collection of useful aggregate statistics in mobile sensing without leaking mobile users’ privacy, we proposed a new privacy-preserving protocol to obtain the Sum aggregate of time-series data. The protocol utilizes additive homomorphic encryption and a novel, HMAC based key management technique to perform extremely efficient aggregation. Implementation-based measurements show that operations at user and aggregator in our protocol are orders of magnitude faster than existing work. Thus, our protocol can be applied to a wide range of mobile sensing systems with various scales, plaintext spaces, aggregation loads, and resource constraints. Based on the Sum aggregation protocol, we also proposed two schemes to derive the Min aggregate of time-series data. One scheme can obtain the accurate Min, while the other one can obtain an approximate Min with provable error guarantee at much lower cost. To deal with dynamic joins and leaves, we proposed a scheme that utilizes the redundancy in security to reduce the communication cost for each join and leave. Simulation results show that our scheme has much lower communication overhead than existing work.

VI. REFERENCES


Author’s Profile:

**Saikrishna Padma, Dept of CSE, Avanthi Institute of Engineering and Technology, Hyderabad, India,**
Email ID: Krishna.padmasai@gmail.com.