Efficient Privacy-preserving Aggregation for Mobile Crowdsensing

Mengdi Huai*†, Liusheng Huang*†, Yu-e Sun‡§ and Wei Yang*†
*School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China
†Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou 215123, China
‡ School of Urban Rail Transportation, Soochow University, Suzhou 215006, China
§Nanjing University of Information Science & Technology, Nanjing 210044, China
Email: mdhuai@mail.ustc.edu.cn

***Abstract*—Mobile crowdsensing applications can learn the aggregate statistics over personal data to produce useful knowledge about the world. Since personal data may be privacy-sensitive, the aggregator should only gain desired statistics without learning anything about the personal data. Differential privacy, the state-of-the-art privacy mechanism, can provide strong protection to ensure parties' privacy in such scenarios. Correspondingly, based on the differential privacy, many collusion-tolerant aggregation schemes have been proposed. However, those collusion-tolerant schemes usually incur high accumulated error and also require the priori knowledge of the fraction of those colluded parties. In this paper, we propose a differential-private collusion-tolerant aggregation protocol, while incurring no additionally error except the noise required for providing the differential privacy guarantee. Another salient characteristic of the proposed protocol is that it need not to have an priori estimation of those colluded parties. In addition, we also design an efficient aggregation encryption scheme to support those mobile crowdsensing applications where large plaintext is required. We also make some extensions to make the proposed protocol more applicable in realities, such as the fault tolerant. The analysis shows that the proposed protocol can achieve desired goals, and the performance evaluation demonstrates the protocol's efficiency in reality.**

***Index Terms*—Mobile Crowdsensing, Aggregation, Differential Privacy.**

I. Introduction

The increasing deployment of sensors on current smartphones and other daily devices, paves the way for an new exciting paradigm for accomplishing large-scale sensing, known as participatory sensing and/or mobile crowdsensing [8]. The key idea behind mobile crowdsensing is to empower ordinary users to collect and share sensed data learned from the measured environment, to gain advanced knowledge.

A plethora of novel and exciting mobile crowdsensing applications, which perform the aggregated statistics over these sensing data, have emerged in recent years. For example, Hull et al. [11] propose a system that uses mobiles phones carried in vehicles to aggregate statistic results about traffic, quality of en-route WiFi access points, and potholes on the road.

Within the scope of this manuscript, we discuss the typical mobile crowdsensing system where a potentially malicious aggregator aggregates sensing data related to the parties and/or their environment. In other words, this mobile crowdsensing system operates in a centralized fashion, i.e., the sensed data collected by parties are reported (e.g., using wireless data communications) to the untrusted aggregator. However, during information aggregation for knowledge discovery, it also brings privacy and security concerns, because private information leakage can have serious consequences.

The authors in [5] define the notion of aggregation privacy in mobile crowdsensing system and highlight threats to privacy resulting from the disclosure of parties' sensed data to the untrusted aggregator. Here, we briefly introduce the defined aggregation privacy notion [5] in the mobile crowdsensing system: aggregation privacy in mobile crowdsensing is the guarantee that parties maintain control over the release of their sensitive information. This includes the protection of information that can be inferred from both parties' submitted data as well as from the final aggregation statistic results in the mobile crowdsensing system [5]. This implies that in addition to the direct protect of the sensing data being submitted to the aggregator, the protection of the parties' privacy against the malicious inference from the aggregation results should also be achieved in the mobile crowdsensing system.

A. Related Work

In order to tackle the privacy concerns in different aggregation applications, such as mobile crowdsensing, various private aggregation schemes have been proposed [12], [10], [1], [21], [18], [3], [20], [4], [15], [2].

The primitive private aggregation schemes, such as [12], [10], only consider how to securely perform the aggregation task over multiple parties. More specifically, those secure aggregation schemes allow the aggregator to learn the accurate aggregation statistic results while preserving each party' submitted data via specific techniques, such as homomorphic encryption. However, the accurate aggregation results always disclose private information of parties [6]. That is to say, parties' data privacy can be violated by malicious inference from the accurate aggregation statistic results, which contracts with the aggregation privacy notion in the mobile crowdsensing system defined in [5].

To protect the privacy of parties' sensitive information against the illegitimate inference from the accurate aggregation results, parties can add noises to their individual data to make the aggregator only derive noisy aggregation summation results. Meanwhile, two goals during the period aggregation

CFP1552Z-CDR/15 $31.00 © 2015 IEEE
DOI 10.1109/BDCloud.2015.54

procedure have to be ensured: preserving each party' data privacy and providing useful aggregation results. Otherwise, the high added noise make the learned aggregation results useless in reality. Fortunately, differential privacy [6] currently is a strong privacy guarantee mechanism that can provide both good utility and privacy guarantee.

Given that, lots of differentially private aggregation schemes have been emerged. In 2010, Rastogi et al. [20] firstly propose the differentially private aggregation scheme. Yet, this scheme sacrifice the aggregation accuracy for resisting the collusion attack. More specifically, the aggregator finally decrypt the noisy aggregation summation result: $\bar{Q} = Q + Lap(\lambda) + ExtraNoise$, where Q is the accurate summation over distributed parties, $Lap(\lambda)$ is the noise generated by honest parties to provide ϵ-differential privacy guarantee and $ExtraNoise$ is that generated by malicious parties.

Shi et al. [21] in 2011 propose an computational differential-privacy [18] (i.e., the (ϵ, δ)-differential privacy, a relaxtion of the standard ϵ-differential privacy) aggregation scheme, which is also collusion-tolerant. Subsequently, some works [15], [3] adopt this scheme. However, firstly, this private aggregation scheme [21] only satisfies the relaxed $(\epsilon, \delta)-$differential privacy, since δ ($\delta > 0$) decreases the utility. Secondly, on the extreme situation where $\gamma = O(\frac{1}{n})$, the accumulated error would be $O(\frac{\Delta}{\epsilon}\sqrt{n})$, which makes no sense in practice where more accurate aggregation results are required.

Besides, the authors in [1] propose a ϵ-differential privacy aggregation scheme, which can achieve the two privacy requirement in the mobile crowdsensing system [5] and is also collusion-tolerant. However, the added noise is larger than $Lap(\lambda)$, the Laplace noise which is needed to ensure ϵ-differential privacy. In 2013, Barthe et al. [2] propose a aggregation protocol, which only ensures the computational differential privacy, i.e., (ϵ, δ)-differential privacy. In 2014, Chen et al. [4] propose a private aggregation scheme, which only provides 2ϵ-differential privacy.

Most importantly, those differential-private aggregation schemes [20], [21], [3], [15], [1], [5] mentioned above all require that the system should have a priori estimate over the fraction of those corrupted parties, which is unpractical and inflexible in many mobile crowdsensing applications. And, they usually resist collusion attacks with a specific probability.

B. Our Result

Given those, we propose a private aggregation scheme, which can provide ϵ-differential privacy guarantee without incurring any additional error and is collusion-tolerant. The proposed protocol ensures that the accumulated error in the sum statistics is only an copy Laplace noise required for ϵ-differential privacy, but the magnitude of the noise incorporated to each party' data is not large enough to the data privacy. On the other hand, communication channels are insecure, making parties' data suffer from eavesdropping attacks. So, it can not guarantee the security of parties' sensed data if only based on the proposed differentially-private scheme. And, we improve this by designing an efficient cryptographic scheme. Besides, several other problems should also be addressed, such as data pollution.

To sum up, the contributions of this paper can be summarized as follows: firstly, we propose a ϵ-differential privacy aggregation protocol, which can achieve the goal of resisting collusion attacks without incurring extra noise. Remarkably, the proposed collusion-tolerant aggregation protocol don't require a priori estimation on the fraction of those colluded parties; secondly, we give an efficient cryptographic scheme which can support large plaintext spaces for computing the aggregation summation, to resist eavesdropping attacks; thirdly, we also extend our differential private protocol to make it more applicable in reality, such as supporting parties' dynamic joins or leaves.

II. System Model

A. System Setting

Here, we elaborate our aggregation system model in mobile crowdsensing. We consider the distributed scenarios where there are k parties in total and each party P_i holds his locally private sensing data $X_{i,t}$ ($i \in [0, k)$) at time t, while the sensing data aggregator (SDA for short) want to compute the sum of k measurements ($i.e., \sum_{i=0}^{k-1} X_{i,t} = X_t$ in all t).

B. Attack Model

The attack model adopted in our system is the adaptive semi-honest adversary model [17], an appropriate model widely accepted in distributed settings [17]. In this adversary model, an adaptive adversary is capable of choosing which parties to corrupt with during the computation procedure, rather than having a fixed set of corrupted parties, while previous works usually assume a fixed set of compromised parties. In our mobile crowdsensing system, an adversary can either be SDA or a participating party, who manages to learn both the accurate local data (i.e., $X_{i,t}$) of each party and sum of local data (i.e., X_t) from all parties. Besides, all parties honestly follow the proposed protocol, but adversaries obtain all their internal values. Additionally, all communication channels are assumed to be insecure, which means that parties' transmitted messages to SDA are susceptible to eavesdropping attacks.

C. Designing Goals

- **Utility and Privacy Guarantee.** Since the accurate sum statistic over distributed parties can potentially violate parties' data privacy [7], we should let each party add an appropriate noise to his data to protect the data privacy. Meanwhile, we should also compromise two objectives: preserving privacy and ensuring good utility. To achieve this, we using the infinite divisibility of Laplace distribution propose a ϵ-differential private aggregation protocol, allowing parties collectively add only a Laplace noise without any additional error to each aggregation result. On the other hand, we use the secret sharing to make the proposed protocol collusion-tolerant, to protect honest parties' privacy against the collusion attack between colluded parties and SDA.

- **Security Guarantee.** Note that using the infinite divisibility of Laplace distribution, the accumulated error in the sum statistics is only an copy of Laplace noise, but the magnitude of the noise incorporated to parties' data is not large enough to protect parties' data privacy. And communication channels between them are insecure, making their data suffer from eavesdropping attacks. So, only based on the data perturbation mechanisms, the security of parties' data can be violated. We address this security threat by designing an efficient cryptographic scheme.
  Given those, we design a differential privacy mechanism combining with cryptographic construction, to provide both security and privacy guarantee. The main security goal in our distributed aggregation protocol is to ensure that SDA learn only the final noisy summation results, and nothing else about the private data provided by parties.
- **Additional Guarantee.** In the distributed aggregation environment, the problem that participating parties can dynamically leave and join should be considered in the proposed protocol. Additionally, some other situations, such as data pollution, also need to be considered.

III. Basic Protocol Blocks

A. Protocol Sketch

Here, we give a high level description of the proposed aggregation protocol. Firstly, each party independently adds an appropriate noise to his sensing data using the data perturbation scheme mentioned below, then encrypts his noisy data using the corresponding encryption scheme and at last sends the encrypted noisy sensing data to SDA who can decrypt summation statistics results ($X_t = \sum_{i=0}^{k-1} X_{i,t}$). Note that all parties collectively add a Laplace noise (required for providing ϵ-differential privacy) to every summation query count X_t at time t, i.e., $X_t = \sum_{i=0}^{k-1} X_{i,t}$. Next, we describe those two basic blocks, i.e., the Laplace perturbation scheme and encryption scheme.

B. The Laplace Perturbation Scheme

Before presenting the Laplace perturbation scheme, we firstly give the introduction to differential privacy [6]. Differential privacy is proposed to protect the sensitive information that needs to be released, and can also provide information-theoretic guarantees that hold against computationally unbounded adversaries. It balances the tradeoff between privacy protection and utility loss. The definition of differential privacy is as follows.

Definition 1: **Differential Privacy.** A randomized function M gives ϵ-differential privacy (ϵ is the privacy parameter) if for all neighborhood dataset D_1 and D_2 differing in at most one record, and all $S \in Range(M)$,

$$Pr[M(D_1) \in S] \leq exp(\epsilon) * Pr[M(D_2) \in S] \qquad (1)$$

The Laplace Perturbation Scheme. Here, we want to publish the output of a function f in a differentially private way. Dwork et al. [6] propose the Laplace perturbation scheme to guarantee the ϵ-differential privacy via adding suitably-chosen noise to the value of f. The noise is generated according to the Laplace distribution. Within this manuscript, we denote $Lap(\lambda)$ a random variable drawn from the Laplace distribution. The Laplace perturbation scheme first exactly computes the accurate value of f, and then perturb the value by adding an independent noise $Lap(\lambda)$. More formally, it computes and outputs $\bar{f} = f + Lap(\lambda)$. Differential privacy is guaranteed if the parameter λ of the Laplace noise is calibrated according to the sensitivity of f, i.e., Δ. The below theorem presented in [6] formalizes this intuition.

Theorem 1: For all $f : \mathbb{D} \rightarrow \mathbb{R}^r$, the following mechanism $\mathcal{A}$ is $\epsilon-$differential privacy: $\mathcal{A}(D) = f(D) + Lap(\frac{\Delta}{\epsilon})$, where $Lap(\frac{\Delta}{\epsilon})$ ($\lambda = \frac{\Delta}{\epsilon}$) is an independently generated random variable following the Laplace distribution and Δ denotes the global sensitivity of f.

In our aggregation system, f is the summation function of all parties' private sensing data, i.e., $X_t = \sum_{i=1}^{k} X_{i,t}$, and the sensitivity Δ is the maximum value that an input can take. According to Theorem 1, the proposed aggregation protocol can be achieved in a differentially private way by perturbing the output of f by simply adding a Laplace noise $Lap(\lambda)$ to the output value of f.

C. The Distributed Laplace Perturbation Scheme

Intuition. Here, our goal is to give a ϵ-differential private aggregation protocol in the distributed scenarios, while resisting collusion atttacks without incurring extra error. To this, the Laplace perturbation algorithm described above gives the intuition that we can let just one designated party add a Laplace noise to the final statistic without incurring any extra error. But one problem coming along with this intuitive solution is that since SDA knows this appointed one, he then can corrupt with this party to violate other parties' data privacy.

The Basic Collusion-tolerant Perturbation Scheme. Given that, we using the secret sharing propose an random select algorithm (i.e., Alg.1) to randomly select a party responsible for adding the Laplace noise, which ensures that only this selected one knows that he is selected and no one else knows about this. The proposed random select algorithm which let all parties jointly select a party securely randomly is presented in Alg.1.

In Alg.1, SDA assigns each party P_i ($i \in [0, k)$) an unique index I_i randomly, and distributes each the index in k secret share form in $\mathbb{Z}_p$ to all parties. Then, all parties using **JRP**^{q} in [13] collectively produce a random number R ($R \in \mathbb{Z}_p$) in secret share form. Via the modulo reduction technique proposed in [13], the secret share of the random number R is converted to modulo k. After that, each party P_i uses the secure equality test method in [13] to judge whether R is equal to I_i or not. Note that, $R \in [0, k-1)$ and the index for each party is different, so R will only match one of I_i ($i \in [0, k)$). Since none of the parties can gain knowledge of the jointly created random number R, SDA will not be able to determine which party is selected unless he happens

Algorithm 1 RandomSelect(SDA, P)

Require: SDA : the sensing data aggregator
Require: P : the distributed party set $P = \{P_0, P_2, ..., P_{k-1}\}$

1: SDA randomly assign an unique index $I_i \in [0, k)$ for each party P_i and splits each I_i using secret sharing scheme into k secret sharing where each P_i holds a secret share of I_i
2: P jointly create a random number R using JRP, and each P_i holds a secret share R_i
3: ModuloReduction(R, k)
4: **for** $i = 0$ to $k - 1$ **do**
5: P_i initiates secureEqualityTest(R and I_i)
6: **if** P_i gets 1 from equality test **then**
7: P_i is selected
8: **end if**
9: **end for**

to corrupt with the randomly selected party. The probability of such a accidental successful collusion is only $\frac{1}{k}$. Using Alg.1 to randomly select a party responsible for adding the Laplace noise (i.e., $Lap(\lambda)$), the Laplace perturbation scheme can provide differential privacy guarantee while resisting collusion attacks with high probability.

The Improved Collusion-tolerant Perturbation Scheme. And, due to infinite divisibility of Laplace distribution [14], a Laplace distribution random variable can be computed by summing up n other random variables. So, we can use it to further reduce the probability of such collusion attacks. And, we denote those selected parties as $P_0, P_1, ..., P_{n-1}$. The improved Laplace perturbation algorithm (i.e., Alg.2) picks n parties via Alg.1, to let each selected party P_i ($i \in [0, n)$) add an noise $\sigma_{i,t} = \mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda)$ to perturb his data $X_{i,t}$, such that $Lap(\lambda) = \sum_{i=0}^{n-1}(\mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda))$, where $\mathcal{G}_1(n, \lambda)$ and $\mathcal{G}_2(n, \lambda)$ are i.i.d gamma noise. Yet, those reminding unselected parties don't perturb their data.

Algorithm 2 The Improved Data Perturbation Algorithm

Require: SDA : the sensing data aggregator
Require: P : the distributed party set $P = \{P_0, P_1, ..., P_{k-1}\}$ and the dataset $X_t = \{X_{0,t}, X_{1,t}, ..., X_{k-1,t}\}$
Require: ϵ: the privacy parameter
Require: n: the number of parties who are selected to add an noise $\sigma_{i,t}$ to his data $X_{i,t}$
1: **for** $i = 0$ to $n - 1$ **do**
2: RandomSelect(SDA,P): Randomly select a party P_m ($P_m \in P$)
3: This selected party P_m adds a noise $\sigma_{m,t} = \mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda)$ to his data $X_{m,t}$
4: **end for**

D. The Improved Encryption Scheme

The authors in [21] propose an aggregation encryption scheme which don't require all parties must be simultaneously online and interact with each other. In addition, their encryption scheme can provide *Aggregation Security* at lower communication cost and the parties' secret keys can be kept fresh, since each party's secret key $H(t)^{sk_i}$ ($i \in [0, k)$) varies with different time t, and H denotes the hash function.

Yet, to compute the aggregation value X_t, the discrete log needs to be computed, making it only supports polynomial-sized plaintext spaces where decryption can be achieved through a brute-force search. Even using Pollard's lambda method, decryption time is roughly square root in the plaintext space.

Since discrete logarithm is computational expensive, we instead using the modular property $(1 + N)^m = 1 + mN \mod N^2$ propose the improved protocol which has below steps:

- **Setup**(k,λ): A one-time setup algorithm, run by a trusted dealer, takes two input parameters: the number of parties k, and a security parameter λ as inputs. The trusted dealer randomly generates a modulus $N = pq$, which is the product of two equal-size primes p, q. It outputs the following:
$$(param, sk_k, \{sk_i\}_{i\in[0,k)})$$
where *param* are system parameters. Capability sk_k is distributed to SDA, and sk_i ($i \in [0, k)$) is a secret key distributed to party $P_i(i \in [0, k))$, such that $sk_0 + sk_1 + ... + sk_k = 0$. The parties will use their secret keys to encrypt their data, and SDA will use its decryption secret key sk_k to decrypt the sum. The setup step only need to be performed once during the whole learning procedure, which can largely reduce the cost.
- **Encrypt**(*param*,$sk_i, X_{i,t}$): At time t, each P_i first calculates $(1 + X_{i,t} \cdot N) \mod N^2$. Then the party multiplies it by secret parameter $H(t)^{sk_i}$ to get the ciphertext $c_{i,t}$:
$$c_{i,t} = (1 + X_{i,t} \cdot N) \cdot H(t)^{sk_i} \mod N^2.$$
After that, he uploads the ciphertext $c_{i,t}$ to SDA.
- **Decrypt**(*param*,$sk_k, c_{0,t}, ..., c_{k-1,t}$): After receiving the ciphertexts $\{c_{i,t}\}_{i\in[0,k)}$ from all parties, SDA then calculates the following C_t:
$C_t = H(t)^{sk_k} \cdot \prod_{i=0}^{k-1} c_{i,t} = H(t)^{sk_k} \cdot \prod_{i=0}^{k-1}(1 + X_{i,t} \cdot N) \cdot H(t)^{sk_i} = (1 + N\sum_{i=0}^{k-1} X_{i,t}) \mod N^2$
Then, the aggregator only needs to calculate $(C_t - 1)/N \mod N = \sum_{i=0}^{k-1} X_{i,t} \mod N^2$ to decrypt the final sum $\sum_{i=0}^{k-1} X_{i,t}$ ($X_{i,t} \in \mathbb{Z}_N$). Although two different modular operations are used here, it does not affect the decryption.

The decryption time in the improved encryption scheme is only $O(1)$, while that in the naive encryption scheme proposed in [21] is at least $O(\sqrt{n\Delta})$, which happens only when the plaintext space is small. As for the large plaintext space, the decryption time for this naive encryption scheme will be inconceivable.

IV. The Proposed Aggregation Protocol

A. Computation of Sensitivity

Note that, the proposed aggregation protocol provide ϵ-differential privacy guarantee by adding a Laplace noise

$Lap(\lambda = \frac{\Delta}{\epsilon})$, where the sensitivity Δ is set as $\Delta = \max(X_{0,t}, X_{1,t}, ..., X_{k-1,t})$.

B. Protocol Description

The proposed protocol consists of following phases:

- **Setup.** Similar to the improved encryption scheme, each party P_i ($i \in [0, k)$) obtains the private key sk_i (sk_i), and SDA obtains the capability sk_k.
- **RandomSelect.** In this phase, n parties are firstly selected by the Alg.2 (i.e., the improved collusion-tolerant perturbation scheme) to perturb their data. For simplicity, we denote those n selected parties as $P^1 = P_0^1, P_1^1, ..., P_{n-1}^1$, and those unselected ones as $P^2 = P_n^2, ..., P_{k-1}^2$, such that $P = P^1 \cup P^2$.
  To ensure their ϵ-differential privacy, each party $P_i^1 \in P^1$ ($i \in [0, n)$) adds an noise $\sigma_{i,t} = \mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda)$ to the original data $X_{i,t}$ before encrypting them. So, the perturbed data of $\mathbf{P}_i^1 \in P^1$ is $\bar{X}_{i,t} = X_{i,t} + \sigma_{i,t} = X_{i,t} + \mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda)$. In contrast, those parties in P^2 don't perturb their data $X_{j,t}$ ($j \in [n, k)$). Here, some parties' data might not be an integer, but can convert to an integer by a specific designed scale, which doesn't affect the final aggregation results as long as all data do this.
- **DataEnc.** Using the improved encryption scheme, each party P_i^1 in P^1($i \in [0, n)$) encrypts his perturbed data $\bar{X}_{i,t}$. Similarly, each party P_j ($j \in [n, k)$) in P^2 also encrypts their original data $X_{j,t}$. Here, we use $c_{i,t}$ ($i \in [0, n)$) and $c_{j,t}$ ($j \in [n, k)$) to respectively represent the ciphertexts of $\bar{X}_{i,t}$ and $X_{j,t}$.
- **ResultDec.** As soon as receiving the ciphertexts $(c_{0,t}, c_{1,t}, ..., c_{k-1,t})$ from all parties, SDA then can obtain the summation plaintexts $C_t = H(t)^{sk_k} \prod_{i=0}^{k-1} c_{i,t}$. That is to say, through the decrypt algorithm in the encryption scheme, SDA can obtain: $C_t = H(t)^{sk_k} \prod_{i=0}^{k-1} c_{i,t} = \prod_{i=0}^{n-1}(1 + \bar{X}_{i,t} \cdot N)) \cdot \prod_{i=n}^{k-1}(1 + X_{i,t} \cdot N) = (1 + (\sum_{i=0}^{n-1} \bar{X}_{i,t} + \sum_{i=n}^{k-1} X_{i,t}) \cdot N)$

Note that, n distributed parties in P^1 selected by the Alg.2 collectively add one copy of Laplace noise $Lap(\lambda)$ to the final summation statistic, and then the aggregator obtain the perturbed summation $\bar{X}_t = \sum_{i=0}^{k-1} X_{i,t} + \sum_{i=0}^{n-1}(\mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda)) = \sum_{i=0}^{k-1} X_{i,t} + Lap(\lambda)$.

C. Privacy and Security Analysis

1) **Privacy of each party.** For the insecure communication channels in our aggregation system, we design an efficient encryption scheme to ensure each party's data privacy. The efficient encryption scheme used in the proposed protocol is an improved version of the encryption scheme proposed in [21]. According to the security proof proved in [21], our designed efficient protocol meets the aggregation obvious security notion.
2) **Privacy of the Aggregate Statistic.** In our aggregation protocol, those selected parties are responsible for jointly adding a Laplace noise to final aggregation result. And, according Theorem 1, our aggregation protocol can provide ϵ-differential privacy assurance. Meanwhile, no extra error is incurred excepted the Laplace noise required for providing ϵ-differential privacy guarantee. Next, we via a example show that the proposed aggregation protocol can resist the collusion attack with high probability.
   Here, we assume that Alg.2 randomly selects two parties (i.e., $n = 2$) to collectively add the calibrated Laplace noise to provide the ϵ-differential privacy guarantee. When the adversary is SDA and he is able to collude with the two selected parties, he then can gain access to the true aggregation results. However, the random select algorithm we propose can to a large degree bound the probability of such a successful collusion attack. Since neither the parties nor SDA can know which two parties are selected, SDA can only randomly pick a party to corrupt. Therefore assuming there are k parties, and SDA can only corrupt C of them, then the probability for SDA gaining access to accurate aggregation results is $\frac{C(C-1)}{k(k-1)}$. More specifically, if SDA can control 100 out of 1000 parties, he only has around 1% chance to learn the true results. When the adversary is a party, he have to firstly corrupt with SDA, since only SDA has the noisy aggregation results. Once SDA is corrupted, the similar analysis for SDA being an adversary applies, however if the party cannot corrupt with SDA, he will have to corrupt with all other parties in order to get accurate aggregation results.
   Therefore, under our attack model, the proposed aggregation protocol is most effective when the number of parties is large, since the relative probability for adversaries to gain accurate aggregation results in such a setting is extremely small.

To sum up, the proposed protocol can provide both security and ϵ-differential privacy guarantee, and can also resist collusion attacks with very high probability, while incurring no additional error except the Laplace noise needed for ensuring ϵ-differential privacy.

D. Discussion

In reality, distributed parties' dynamic joins and leaves should be well considered, which we solve via the interleaved grouping technique. Those two problem, malicious modification and data pollution, also need to be considered. Due to the limited space, we will leave their detailed introduction to the full paper.

V. Performance Analysis

A. Theoretical Analysis

The proposed aggregation protocol via Alg.2 allows n randomly selected parties jointly add exactly a Laplace noise $Lap(\lambda = \frac{\Delta}{\epsilon})$ to every period aggregation summation (i.e., X_t). More specifically, each selected party incorporates a noise $(\mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda))$ into his data $X_{i,t}$, such that $Lap(\lambda) = \sum_{i=0}^{n-1}(\mathcal{G}_1(n, \lambda) - \mathcal{G}_2(n, \lambda))$. According to Theorem 1, the proposed aggregation protocol is ϵ-differential private. Additionally, the added Laplace noise remains independent of the number of distributed parties.

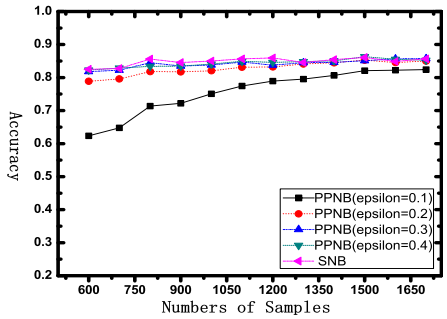

Fig. 1: Classification accuracy comparison between PPNB and SNB

B. Practical Performance

Naive Bayes, one of the widely used classification algorithms (e.g., Adaboost [23], Support Vector Learning [9]), is of great use in performing the knowledge discovery, especially in the mobile crowdsensing scenarios [19]. Here, we use the proposed aggregation protocol to perform the privacy-preserving Naive Bayes (PPNB for short) learning over the horizontally partitioned dataset [22].

In experiments, we compare the classification performance between: standard Naive Bayes (SNB for short) and PPNB. Fig.1 shows simulation results on the dataset Car Evaluation [16]. We vary the number of samples n, and compare their practical utility (i.e., classification performance) under fixed privacy parameters ($\epsilon = 0.1, 0.2, 0.3, 0.4$ respectively). For each n, we average and record the ten-fold cross-validation accuracy over 2000 runs, since it is a randomized algorithm. The simulation shows that PPNB has comparable or even batter classification performance when compared with SNB, especially when the number of samples increases. From Fig.1, we can clearly see that the larger ε is, the better classification performance PPNB have.

VI. Conclusion

In this paper, we propose a ϵ-differential privacy aggregation protocol, which can resist collusion attacks without incurring extra error and requiring the priori estimation on those colluded parties, and along with give an efficient encrypt scheme to make it secure under insecure communication channels, and lastly make some practical extensions. The experimental results show that the proposed aggregation protocol are effective to be applicable in practice.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61303206, 61232016, U1405254, U1301256, 61170058, and 61202028, and the PAPD foundation No. KJR1506.

References

[1] Gergely Ács and Claude Castelluccia. I have a dream!(differentially private smart metering). In *Information Hiding*, pages 118–132. Springer, 2011.

[2] Gilles Barthe, George Danezis, Benjamin Grégoire, Cesar Kunz, and Santiago Zanella-Beguelin. Verified computational differential privacy with applications to smart metering. In *CSF*, pages 287–301. IEEE, 2013.

[3] T-H Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography and Data Security*, pages 200–214. Springer, 2012.

[4] Jianwei Chen and Huadong Ma. Privacy-preserving aggregation for participatory sensing with efficient group management. In *GLOBECOM, 2014 IEEE*, pages 2757–2762.

[5] Delphine Christin, Andreas Reinhardt, Salil S Kanhere, and Matthias Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.

[6] Cynthia Dwork. Differential privacy. In *Encyclopedia of Cryptography and Security*, pages 338–340. Springer, 2011.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, pages 265–284. Springer, 2006.

[8] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39, 2011.

[9] Bin Gu, Victor S Sheng, Keng Yeow Tay, Walter Romano, and Sinan Li. Incremental support vector learning for ordinal regression. *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[10] Wenbo He, Xue Liu, Hoang Nguyen, Klara Nahrstedt, and Tarek Abdelzaher. Pda: Privacy-preserving data aggregation in wireless sensor networks. In *INFOCOM*, pages 2045–2053. IEEE, 2007.

[11] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138. ACM, 2006.

[12] Taeho Jung, XuFei Mao, Xiang-Yang Li, Shao-Jie Tang, Wei Gong, and Lan Zhang. Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation. In *INFOCOM*, pages 2634–2642. IEEE, 2013.

[13] Eike Kiltz. Unconditionally secure constant round multi-party computation for equality, comparison, bits and exponentiation. *IACR Cryptology ePrint Archive*, 2005:66, 2005.

[14] Samuel Kotz, Tomasz Kozubowski, and Krzysztof Podgorski. *The Laplace distribution and generalizations: a revisit with applications to communications, exonomics, engineering, and finance*. Number 183. Springer Science & Business Media, 2001.

[15] Qinghua Li and Guohong Cao. Efficient and privacy-preserving data aggregation in mobile sensing. In *ICNP*, pages 1–10. IEEE, 2012.

[16] M. Lichman. UCI machine learning repository, 2013.

[17] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):5, 2009.

[18] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In *Advances in Cryptology-CRYPTO 2009*, pages 126–142. Springer, 2009.

[19] Valentin Radu, Lito Kriara, and Mahesh K Marina. Pazl: A mobile crowdsensing based indoor wifi monitoring system. In *CNSM*, pages 75–83, 2013.

[20] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD*, pages 735–746.

[21] Elaine Shi, T-H Hubert Chan, Eleanor G Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *NDSS*, volume 2, page 3, 2011.

[22] Jaideep Vaidya, Murat Kantarciouglu, and Chris Clifton. Privacy-preserving naive bayes classification. *The VLDB Journal*, 17(4):879–898, 2008.

[23] Xuezhi Wen, Ling Shao, Yu Xue, and Wei Fang. A rapid learning algorithm for vehicle classification. *Information Sciences*, 295:395–406, 2015.