DPWeVote: differentially private weighted voting protocol for cloud-based decision-making

Article *in* Enterprise Information Systems · March 2018

DOI: 10.1080/17517575.2018.1442935

3 authors:

Ziqi Yan
Beijing Jiaotong University
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Jiqiang Liu
Beijing Jiaotong University
86 PUBLICATIONS **361** CITATIONS
SEE PROFILE

Shaowu Liu
University of Technology Sydney
16 PUBLICATIONS **103** CITATIONS
SEE PROFILE

DPWeVote: differentially private weighted voting protocol for cloud-based decision-making

Ziqi Yan[a], Jiqiang Liu[b] and Shaowu Liu[c]

[a,b]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China. {zichiyen, jqliu}@bjtu.edu.cn
[c]Advanced Analytics Institute, University of Technology Sydney, Ultimo, Australia. shaowu.liu@uts.edu.au

ARTICLE HISTORY
Compiled April 5, 2018

ABSTRACT
With the advent of Industry 4.0, cloud computing techniques have been increasingly adopted by industry practitioners to achieve better workflows. One important application is cloud-based decision-making, in which multiple enterprise partners need to arrive an agreed decision. Such cooperative decision-making problem is sometimes formed as a *weighted voting game*, in which enterprise partners express 'YES/NO' opinions. Nevertheless, existing cryptographic approaches to *Cloud-Based Weighted Voting Game* have restricted collusion tolerance and heavily rely on trusted servers, which are not always available. In this work, we consider the more realistic scenarios of having semi-honest cloud server/partners and assuming maximal collusion tolerance. To resolve the privacy issues in such scenarios, the `DPWeVote` protocol is proposed which incorporates *Randomized Response* technique and consists the following three phases: the *Randomized Weights Collection* phase, the *Randomized Opinions Collection* phase, and the *Voting Results Release* phase. Experiments on synthetic data have demonstrated that the proposed `DPWeVote` protocol managed to retain an acceptable utility for decision-making while preserving privacy in semi-honest environment.

KEYWORDS
Industry 4.0; Cloud-Based Design and Manufacturing; Cloud-Based Decision-Making; Weighted Voting; Differential Privacy

1. Introduction

As the fourth industrial revolution, *Industry* 4.0 aims to achieve the distributed, collaborative and automated design & manufacturing workflow, by taking advantages of the increasingly sophisticated technologies of Internet of Things, Cloud Computing and Big Data (Thames and Schaefer 2017). In recent years, the *Cloud-Based Design and Manufacturing* (CBDM) (Wu et al. 2012, 2013; Schaefer 2014) has become a fundamental paradigm that may fulfill the basic requirements of Industry 4.0 and has attracted widespread attentions from both academia and industry. On the basis of CBDM, advanced information systems such as the Cloud ERP (Enterprise Resource Planning) system (Symonds 2012; Synergy 2015; Cisco 2015) have been developed

CONTACT Jiqiang Liu. Email: jqliu@bjtu.edu.cn

and deployed to help the enterprise partners to achieve better resource scheduling and collaborative decision-making tasks (Cloud 2016; ERP 2016). In this way, the pattern of *Cloud-Based Decision-Making* would be of great benefit to all kinds of enterprise partners with the trend of Industry 4.0. Meanwhile, the *Cooperative Games* (Branzei, Dimitrov, and Tijs 2008) would also contribute to the cloud-based decision-making as it has done to the traditional decision-making. Among them, the *Weighted Voting Game*, in which the voters have diverse weights, can be regarded as a possible process for cloud-based decision-making in the scenario of Industry 4.0, based on the real-world situations (Steinberger 2007).

However, in a weighted voting game for cloud-based decision-making, there exist potential privacy leakages towards the weight w_i and voting opinion ϕ_i, which are owned by each enterprise partner (voter) i and need to be uploaded to cloud server for voting result computation, especially under the assumption of semi-honest cloud server and partners. In this way, the sensitive information such as the weight of an enterprise partner and his/her attitude towards the given elected candidate are exposed to the cloud server as well as to the other enterprise partners, which harms the long-term interests of industry or business.

There have been some researches on the design of secure weighted voting protocol by adopting cryptographic approach such as homomorphic encryption (Nakanishi et al. 2004; Chen, Lin, and Wang 2013; Zhang et al. 2016), but these works have limits on the controlled assumption about collusion, and/or on the setting of additional trusted servers.

In this paper, we aim to design a private data collection protocol to tackle the privacy issues in the weighted voting game for cloud-based decision-making, as well as with the assumptions that the involved entities are all semi-honest and the maximum collusion may occur among them. Moreover, in order to rigorously measure and prove the degree of privacy protection in the proposed protocol, we consider the *Local Differential Privacy* (LDP) model (Kasiviswanathan et al. 2008; Duchi, Jordan, and Wainwright 2012, 2013), which originates from the *Differential Privacy* (DP) model (Dwork et al. 2006; Dwork 2011) and is particularly suitable for the scenario of private data collection. As inheriting the basic privacy-preserving characteristics from the DP model, LDP ensures that the cloud server cannot confidently (measured by privacy parameter ϵ) infer the present or absent status of a single record in each distributed partner's original database by just observing the uploaded data from each partner.

Although there have been some existing researches on differentially private voting such as Chen et al. (2013); Leung and Lui (2012); Lee (2015); Hay, Elagina, and Miklau (2017), these methods do not consider the scenario of weighted voting, where the weights data are diverse and should also be protected. To our best knowledge, the differentially private weighted voting game is a novel problem. The contributions in this paper include the following:

- We first research and formulate the differentially private weighted voting game, in which the cloud server and the partners are assumed to be semi-honest, and both of weights and opinions should be protected. Furthermore, the partners weights are classified into three groups wI, wII and $wIII$, for directly simulating the high, middle and low level decision-making power in real-world scenario, from which the intuition of double RR mechanism based protocol is driven.
- We adopt the *Randomized Response* (RR) technique to design a differentially private weighted voting protocol `DPWeVote`, which satisfies the local differential

privacy. There are three phases within the proposed protocol, in which we separately apply RR in the phases 1 and 2 for first enabling the partners to perturb their weights w_i and opinions ϕ_i data locally and then enabling the cloud server to estimate useful population statistics of the partners without disclosing their individual data. In the third phase, the cloud server only needs to compute the final voting result by judging whether the estimated summation $\widehat{\sum_{i \in N} w_i \cdot \phi_i}$ is larger than the estimated quota $\widehat{q}$. Finally the voting result regarding the candidate will be released.

- We evaluate the *Accuracy* and *Mean Squared Error* (MSE) of differentially private weighted voting game on synthetic data, showing that the proposed `DPWeVote` protocol always outperforms the baseline algorithm which leverages the laplace mechanism, under varying the parameters ϵ (privacy budget) and n (numbers of partners). Generally, the `DPWeVote` can attain an acceptable data utility in a narrow range of ϵ (from 0.1 to 1.0), and a high data utility in a wider range (larger than 1.0), which is consistent with most of the existing researches on RR-based local differential privacy algorithm.

The rest of this paper is organised as follows. We present the preliminaries and related works in Section 2, and provide the problem statement in Section 3. Section 4 is devoted to describing the `DPWeVote` protocol for achieving the differentially private weighted voting game, followed by its theoretical privacy analysis. Section 5 presents experimental results, and conclusions are drawn in Section 6.

2. Preliminaries and Related Work

This section reviews three fundamental concepts: *Cooperative Games*, *Weighted Voting Game* and *Local Differential Privacy*, and then briefly surveys the related works in *Privacy-preserving Voting Games* and *Local Differential Privacy*.

Table 1 lists the relevant notations used in this paper.

2.1. *Preliminaries*

2.1.1. Cooperative Game

In the game theory, the most difference between the cooperative games and the non-cooperative games is that the former would consider to model the behaviours and associated payoffs of groups (or called coalitions), instead of considering that of the individual partner. In general, the cooperative games are also known as the coalition games, whose definition with transferable utility assumption is given as:

Definition 2.1 (Coalition Game with Transferable Utility (Leyton-Brown and Shoham 2008))**.** A coalition game with transferable utility is a pair (N, v), where

- N is a finite or infinite set of partners, indexed by i; and
- The characteristic function $v : 2^N \to \mathbb{R}$ associates with each coalition $C \subseteq N$ a real-valued payoff (or called worth) $v(C)$ that the coalition's members can distribute among themselves.

In the view of cooperative game theory, the enterprise partners, who may control the distributed manufacturing resources, can be seen as the self-interested players and

Table 1. Summary of Notation

Symbol	Meaning
N	the set of partners $i \in N = \{1, ..., n\}$
v	the characteristic function
C	the coalitions which are formed by partners
K	the set of proposal candidates $l \in K = \{1, ..., k\}$
w_i	the weight adhere to each partner i for $w_i \in \mathbf{w} = \{w_1, ..., w_n\} \in \{1, 2, 3\}^n$ in the game defined in this paper
$\widetilde{w_i}$	the perturbed weight adhere to each partner i
wI	weight group I in which the weights of members are set to 1, $wI = \{w_{j_1} = 1, j_1 \in [1, m_1]\}$
wII	weight group II in which the weights of members are set to 2, $wII = \{w_{j_2} = 2, j_2 \in [1, m_2]\}$
$wIII$	weight group III in which the weights of members are set to 3, $wIII = \{w_{j_3} = 3, j_3 \in [1, m_3]\}$
ϕ_i	the opinion adhere to each partner i, and $\phi_i \in \boldsymbol{\phi} = \{\phi_1, ..., \phi_n\} \in \{0, 1\}^n$
$\widetilde{\phi_i}$	the perturbed opinion adhere to each partner i
ϕY	the 'Yes' opinion group in which the opinions of members are set to 1
ϕN	the 'No' opinion group in which the opinions of members are set to 0
q	the quota of weights that a passed proposal candidate must raise
$\widehat{q}$	the approximate quota which is estimated by the cloud server
$\mathbf{M}_w$	the transformation matrix of weights
$\mathbf{M}_\phi$	the transformation matrix of opinions
ϵ	overall privacy budget

also intend to cooperate with others for better profits. In other words, these partners would make decisions or take actions based on all-win cooperative attitude.

2.1.2. Weighted Voting Game

Being one of the simplest useful cooperative games, the weighted voting game considers the model settings where each partner has certain weight for voting a given proposal candidate. Any coalition whose member's weights summation exceeds a threshold would be the winning coalition and get the proposal candidate passed. The game can be formally defined as the following:

Definition 2.2 (Weighted Voting Game (Chalkiadakis, Elkind, and Wooldridge 2011))**.** A *weighted voting game* G with a set of partners $N = \{1, ..., n\}$ is given by a list of weights $\mathbf{w} = \{w_1, ..., w_n\} \in \mathbb{R}^n$ and a *quota* $q \in \mathbb{R}$; the game will be written $G = [N; \mathbf{w}; q]$. Its characteristic function $v : 2^N \to \{0, 1\}$ of each coalition $C \subseteq N$ is given by

$$v(C) = \begin{cases} 1 & \text{if } \sum_{i \in C} w_i \geq q, \\ 0 & \text{otherwise.} \end{cases}$$

In general, the quota q belongs to $(0, \sum_{i \in N} w_i]$, which ensures that the empty coalition would lose the voting. In this paper, it is convenient and realistic to assume that $q = \frac{1}{2} \sum_{i \in N} w_i$.

2.1.3. Local Differential Privacy

The Local Differential Privacy (LDP) (Kasiviswanathan et al. 2008; Duchi, Jordan, and Wainwright 2012, 2013) is a distributed privacy model for data collection and analysis, which derives from the following traditional definition of Differential Privacy (Dwork et al. 2006; Dwork 2011):

Definition 2.3 (ϵ-Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ gives ϵ-differential privacy if for all neighbour databases D and D' differing in at most one record, and for all $O \subseteq Range(\mathcal{M})$, we have

$$Pr[\mathcal{M}(D) \in O] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(D') \in O].$$

The intuition behind the above definition is that the adversary cannot confidently distinguish the two outputs of a differentially private algorithm $\mathcal{M}$ when inputting database D and its neighbor database D'. That is, the present or absent status of a single record within input database is rigorously protected with the uncertainty of the algorithm's outputs, which is measured by the privacy parameter (also called privacy budget) ϵ. Besides, algorithm $\mathcal{M}$ is associated with the *sensitivity*, which measures the maximum change on the result of query function f when one record from D changes:

Definition 2.4 (Sensitivity)**.** For any function $f : D \to \mathbb{R}^d$, and for all D, D' differing in at most one record, the sensitivity of f is $\triangle f = \max_{D,D'} \|f(D) - f(D')\|_1$.

To satisfy the definition of differential privacy and for those query functions f which have numeric output, the *Laplace* mechanism are usually utilized. It relies on the strategy of adding the *Laplacian* noise $Laplace(\cdot)$ to the query result, and can be formally defined as follows.

Definition 2.5 (Laplace Mechanism)**.** Given a function $f : D \to \mathbb{R}^d$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(D) = f(D) + (Y_1, ..., Y_d),$$

where Y_i are i.i.d random variables drawn from $Laplace(\frac{\triangle f}{\epsilon})$.

To guarantee the overall privacy budget ϵ when it comes to a sequence of differentially private operations, we have the following composition property:

Theorem 2.6 (Sequential Composition (McSherry 2009))**.** *Given n independent randomized algorithms $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n$ where $\mathcal{M}_i (1 \leq i \leq n)$ satisfies ϵ_i-differential privacy, a sequence of $\mathcal{M}_i$ over the dataset S satisfies ϵ-differential privacy, where $\epsilon = \sum_1^n (\epsilon_i)$.*

In LDP model, each distributed partner would first perturb his/her data locally by adopting a randomized mechanism (or called local randomizer $\mathcal{R}$) which is provided by the semi-honest cloud server and satisfies ϵ-differential privacy, and then upload the perturbed data to cloud server, who cannot infer the sensitive information of every single partner but can post-process those data to obtain useful population statistics for further analysis. The LDP can be formally defined as follows:

Definition 2.7 (Local Differential Privacy (Bassily et al. 2017))**.** An algorithm satisfies ϵ-Local Differential Privacy (LDP) if it accesses the database $\mathbf{v} = (v_1, ..., v_n) \in \mathcal{V}^n$ only via invocations of a local randomizer $\mathcal{R}$ and if for all $i \in [n]$, if $\mathcal{R}^{(1)}, ..., \mathcal{R}^{(k)}$

denote the algorithms invocations of $\mathcal{R}$ on the data sample v_i, then the algorithm $\mathcal{A}(\cdot) \triangleq (\mathcal{R}^{(1)}(\cdot), \mathcal{R}^{(2)}(\cdot), ..., \mathcal{R}^{(k)}(\cdot))$ is ϵ-differentially private. That is, if for any pair of data samples v, $v' \in \mathcal{V}$ and $\forall \mathcal{S} \subseteq \mathsf{Range}(\mathcal{A})$, $Pr[\mathcal{A}(v) \in \mathcal{S}] \leq e^{\epsilon} \cdot Pr[\mathcal{A}(v') \in \mathcal{S}]$.

In recent years, the *Randomized Response* (RR) technique (Warner 1965; Chaudhuri 2016) has became the widely used mechanism that achieves LDP in an efficient and effective way. As an original survey technique in statistics, RR allows the partners who take participate in questionnaire survey have the opportunity to answer sensitive questions with *Plausible Deniability* [1]. Specifically, when being asked a question whose answer can be either 'yes' or 'no', the partner is allowed to first flip a biased coin and then gives his/her true answer to the investigator (or cloud server in our scenario) if the coin turns head with a probability p, or otherwise reports the false answer. In this way, the investigator (cloud server) cannot make sure whether the received answer is the true one or the false one from the partner, since the answers have deep relationship with p. Therefore, for a semi-honest partner who intends to follow the above protocol honestly, the RR would provide him/her the possibility to generate randomized answers to the cloud server and then guarantees their privacy protection by granting them this plausible deniability.

Interestingly, it has pointed out that RR for binary attribute survey can be regarded as a specific randomized algorithm that satisfies the ϵ-differential privacy, if the value of coin flipping probability p has the following relationship with the privacy budget ϵ (Erlingsson, Pihur, and Korolova 2014; Bassily and Smith 2015; Wang, Wu, and Hu 2016):

$$p = \frac{e^{\epsilon}}{1 + e^{\epsilon}}. \tag{1}$$

Next, we could use a transformation matrix $\mathbf{M}$ shown below to include the related coin flipping probabilities $p_{i,j}$ which represent the probability that a true answer j is transformed into the provided answer i. In this matrix, the number '0' denotes the answer 'no' while the number '1' denotes the answer 'yes'.

$$\mathbf{M} = \begin{pmatrix} p_{0,0} & p_{0,1} \\ p_{1,0} & p_{1,1} \end{pmatrix} = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}.$$

Then the reconstruction property of RR can be applied in the cloud server's side. Based on the knowledge of the above transformation matrix M, once the semi-honest cloud server obtains y_1 (or y_0), the number of the partners who provide the answer 'yes' (or 'no'), he/she would post-process this number to estimate $\widehat{x_1}$ (or $\widehat{x_0}$), the approximate number of the partners whose true answer are 'yes' (or 'no'). The estimation is an unbiased MLE (Maximum Likelihood Estimate) (Huang and Du 2008) and can be executed by the following equation (Groat et al. 2013; Sei and Ohsuga 2017):

$$\overrightarrow{\widehat{X}} = \mathbf{M}^{-1} \overrightarrow{Y}, \tag{2}$$

where $\overrightarrow{\widehat{X}} = (\widehat{x_0}, \widehat{x_1})^{\tau}$, $\overrightarrow{Y} = (y_0, y_1)^{\tau}$, and $\mathbf{M}^{-1}$ is the inverse matrix of $\mathbf{M}$.

[1] https://en.wikipedia.org/wiki/Plausible˙deniability

It is also notable that RR can prevent the maximum collusion attack in which the cloud server and $N-1$ of N partners collude with each other.

2.2. *Related Work*

2.2.1. Privacy-preserving Voting Games

Voting-based methods are one of the most common techniques for achieving group decision-making. There are many researches on secure voting scheme analysis or design based on cryptographic techniques. For example, Springall et al. (2014) provided a comprehensive security analysis of the Estonian I-voting system and pointed out the potential attacks. Grewal et al. (2015) proposed a remote electronic voting protocol under the assumption of using untrusted computers. The hardware token is needed to distribute the trust between the voter's computer and the election authorities' server. In the work of Will et al. (2015), a cloud-based mobile electronic voting scheme was proposed by leveraging the Homomorphic Encryption technique. Moreover, the work suggested that using a dedicated hardware server for homomorphic tallying and decryption. Park and Rivest (2017) proposed an secure implementation of the *Quadratic Voting* scheme in which both the voting and payments should be considered. In Rivest, Stark, and Perumal (2017) a specific framework BatchVote was proposed, which is based on the finding of a family of social choice functions that can ensure the ease of auditing. Bernhard et al. (2017) showed a comprehensive survey on the secure voting requirements, the existing solutions, and the future research directions. In a latest doctoral thesis Riemann (2017), P2P based online voting protocols were proposed to reduce the dependence on the trusted third parties and solely on cryptographic techniques.

In terms of privacy protection for the cloud-based weighted voting game, Nakanishi et al. (2004) proposed a weighted voting protocol with secret weights by using Homomorphic Encryption technique. This work is based on the assumption that the number of collusive servers cannot beyond the threshold K. Inspired by Nakanishi et al. (2004), the work of Chen, Lin, and Wang (2013) developed a privacy-preserved joint group time scheduling mechanism with the scenario that each user cannot know his/her weight, and with the assumption of non-collusive users. Zhang et al. (2016) proposed two privacy-friendly weighted-reputation aggregation protocols respectively for the semi-honest adversary setting and malicious adversary setting, by adopting the Homomorphic encryption and zero-knowledge proofs.

For the discussions on privacy notations, Bernhard et al. (2012) adopted the computational conditional entropy to measure the privacy in voting schemes and provided theoretical theorem to better analyse the privacy of both cryptographic voting protocols and non-cryptographic protocols. They also demonstrated the connections of the proposed privacy notation and two existing ones. In (Ashur, Dunkelman, and Talmon 2016), a privacy breaching algorithm was proposed to show the weakness of Israels paper ballot voting system which is based on anonymization techniques. Talmon (2015) studied the k-anonymizing preference orders in protecting privacy in elections.

From the aspects of the recent popular privacy model, differential privacy, McSherry and Talwar (2007) initially built the bridge between mechanism design and differential privacy, and proposed the Exponential mechanism especially for private auction. There are some subsequent researches on private auction and its truthfulness such as (Nissim, Smorodinsky, and Tennenholtz 2012; Xiao 2013). For private voting, in (Chen et al. 2013), they proposed a novel way to incorporate differential privacy directly

into the player's utility functions, a private two-candidate elections mechanism was designed. Leung and Lui (2012) considered the Bayesian setting of the distribution of the players types and proposed the Bayesian differential privacy while achieving persistent approximate truthfulness. Lee (2015) proposed an algorithm that satisfies both ϵ-differential privacy and ϵ-strategyproof for protecting participant privacy in tournament voting rules. Recently, considering the rank aggregation scenario in which the data curator is trusted, Hay, Elagina, and Miklau (2017) extended three non-private rank aggregation algorithms to their differentially private versions.

In this paper, we intend to explore the methods to achieve differential privacy of the *Weighted Voting Game* in a distributed scenario, which has not yet been researched.

2.2.2. Local Differential Privacy

There are some prior theoretical researches on Local Differential Privacy (LDP) model and its related algorithms. Kasiviswanathan et al. (2008) proved that the equivalence of learnability with RR-based LDP learning algorithms and statistical query (SQ) model, and also showed the limitations of these algorithms in terms of the required amount of data (exponential), as well as the less powerful learning ability in some situations. Duchi, Jordan, and Wainwright (2012) studied the statistical convex risk minimization problem under the LDP model and provided the bounds on the convergence rates of the estimation procedure. In their subsequent research (Duchi, Jordan, and Wainwright 2013), they showed a more general results which are based on no mechanism restrictions, and gave minimax-optimal error rates. Kairouz, Oh, and Viswanath (2014) studied the privacy-utility trade-off between LDP and f-divergence utility functions, and proposed the Extremal Mechanisms for maximizing utility. Lately, they studied the secure multi-party computation in LDP model, and considered the interactive setting and non-interactive setting (Kairouz, Oh, and Viswanath 2015, 2016).

Due to its unique reconstruction property that allowing the cloud server to estimate population statistics from the collected noisy data of every single partner, RR has been adopted in many researches on private data collection and analysis. The RAPPOR was proposed by Erlingsson, Pihur, and Korolova (2014) to address the differentially private frequency estimation problem which is considered in Google's Chrome web browser. Bassily and Smith (2015) theoretically proposed a protocol for frequency estimation and finding heavy hitters in LDP model, by using the succinct histogram (SH) to represent the data. Considering the heavy hitters over set-valued data, the LDPMiner was proposed by Qin et al. (2016), which is based on Sampling RAPPOR and sampling SH, and contained two phases to optimize the estimation procedure. Under the data collection scenario of single binary attribute and multiple polychotomous attributes, Wang, Wu, and Hu (2016) studied the relationship between Laplace mechanism and RR by comparing their theoretical utility error and showing the ϵ-differential privacy satisfaction results of RR. Sei and Ohsuga (2017) proposed the S2M and S2Mb schemes, a kind of RR algorithm which require less number of samples in estimation, to achieve privacy-preserving mobile crowdsensing. In the scenario of privately local search the frequent records in a web search log, Avent et al. (2017) proposed a hybrid differential privacy model in which the trusted curator model and the LDP model are both considered. Wang et al. (2017) proposed a framework for better comparing different LDP protocols. And two optimized protocols were proposed on the basis of RAPPOR and SH. Recently, Bassily et al. (2017) proposed two locally private heavy hitters algorithms TreeHist and Bitstogram to achieve better trade-off among the utility, complexity and privacy, compared with RAPPOR.

In this paper, we also intend to leverage the appealing properties of RR to meet the stated privacy and utility requirements in cloud-based weighted voting game.

2.2.3. Discussion

As for solving the privacy issues in *Cloud-based Weighted Voting Game*, the existing works mainly rely on the cryptographic approaches which always need the assumption of a certain level of collusion tolerance, and/or the setting of additional trust server. Moreover, the complexity caused by encryption and decryption also makes the related schemes not so convenient for deployment. Although there are some researches on differentially private voting game, none of them consider the specific privacy issues in weighted voting game. To the best of our knowledge, this is the first attempt to apply the lightweight RR technique to achieve local differential privacy in weighted voting game while assuming that the involved entities are all semi-honest and the maximum collusion may occur among them.

3. Problem Statement

This section first introduces the *system assumptions* considered in this work, and then clearly presents the *differentially private weighted voting game* problem, along with its challenges.

3.1. *Assumptions and Problem Definition*

We consider the scenario of the cloud-based weighted voting game. Because the conventional weighted voting games always have no limits on the setting of the values of w_i, and for the simulating the high, middle and low level decision-making power in real-world scenario, in this paper we initially classify them into three groups wI, wII and $wIII$, whose members' weights are 1, 2 and 3, respectively. That is, we have $wI = \{w_{j_1} = 1, j_1 \in [1, m_1]\}$, $wII = \{w_{j_2} = 2, j_2 \in [1, m_2]\}$ and $wIII = \{w_{j_3} = 3, j_3 \in [1, m_3]\}$.

When the game is being created, the cloud server would first assign the proposal candidates $l \in K = \{1, ..., k\}$ for voting to the partners $i \in N = \{1, ..., n\}$, who are associated with weights w_i that belong to three weight groups $(wI, wII, wIII)$. Secondly, the partners would upload their opinions ϕ_i (value 1 for 'yes' and 0 for 'no') respectively back to the cloud server. And finally, the cloud server would judge whether $\sum_{i \in N} w_i \cdot \phi_i \geq q$ and release the voting results to the partners.

In this system model, the cloud server and partners are all assumed to be semi-honest, which means that they would be honest when providing their information for the whole voting process and be curious about the sensitive information from others. In the meanwhile, although those semi-honest [2] entities would not attempt to cheat in all the interactive process during the voting, they would like to cooperate to gather information out of the established protocol. Based on the above system assumptions, we aim to address the following problem of *differentially private weighted voting game*:

Problem 1 (Differentially Private Weighted Voting Game)**.** Given a weighted voting game $G = [N; \mathbf{w}; q]$ which is developed by a semi-honest cloud server, design a protocol that protects the weight w_i and opinion ϕ_i of each semi-honest partner with

[2] https://en.wikipedia.org/wiki/Secure_multi-party_computation

ϵ-differential privacy. Moreover, the protocol should release an approximate accuracy results that satisfy the minimization of given utility metrics.

3.2. *Research Issues*

In this paper, we aim to solve the differentially private weighted voting game problem by leveraging the *Randomized Response* (RR) technique, which brings two main research issues:

How to tailor the RR to fit the problem? By analysing the process of weighted voting game, we propose the rationale of a double RR mechanism to obtain the frequency estimation of weights and opinions separately. And we further develop a three-phase protocol `DPWeVote` whose details are described in Section 4.

How much the utility the protocol can maintain? We noticed that three intermediate estimations cause the error of the final results in the proposed `DPWeVote` protocol. Currently, we observe the maintained utility by empirical evaluations in terms of the changing parameters and utility metrics in Section 5.

4. Private Weighted Voting Protocol

In this section, we propose a RR-based ***D**ifferentially **P**rivate **We**ighted **Vot**ing* (`DPWeVote`) protocol, which not only ensures ϵ-differential privacy in each partner's data, but also allows the cloud server to be able to extract useful population statistics information from the collected perturbed data.

4.1. *Protocol Overview*

The `DPWeVote` protocol actually consists of a series of interactive phases between the distributed partners and the cloud server, which aims to collect and analysis the information of weights and opinions from partners in a private way. Because it requires that each partner uploads his/her perturbed weight and opinion to the cloud server by adopting RR mechanisms that satisfies local differential privacy, the potential privacy leakage in the scenario described in Section 1 would be avoided. The overview of the protocol is shown in Fig. 1.

We examine the involved computation operations and consider to solve the differentially private weighted voting game in the following independent phases:

Randomized Weights Collection For permitting the cloud server to create a new weighted voting game, the partners would first upload their perturbed weights $\widetilde{w_i}$, which respectively belong to one of the three weight groups wI, wII and $wIII$. Then based on the reconstruction property of RR, the cloud server would estimate the unbiased number $\widehat{x_{wI}}$, $\widehat{x_{wII}}$ and $\widehat{x_{wIII}}$, and further computes the approximate quota $\widehat{q}$.

Randomized Opinions Collection In this phase, the cloud server would first require the partners in weight group wI, wII and $wIII$ to upload their perturbed opinions $\widetilde{\phi}$ regarding whether they support the proposal candidates l. Then the cloud server would estimate the unbiased number such as $\widehat{x_{wI,\phi Y}}$ and $\widehat{x_{wI,\phi N}}$, and further computes the approximate summation $\widehat{\sum_{i \in N} w_i \cdot \phi_i}$.

Voting Results Release Based on the previous phases, the cloud server would have

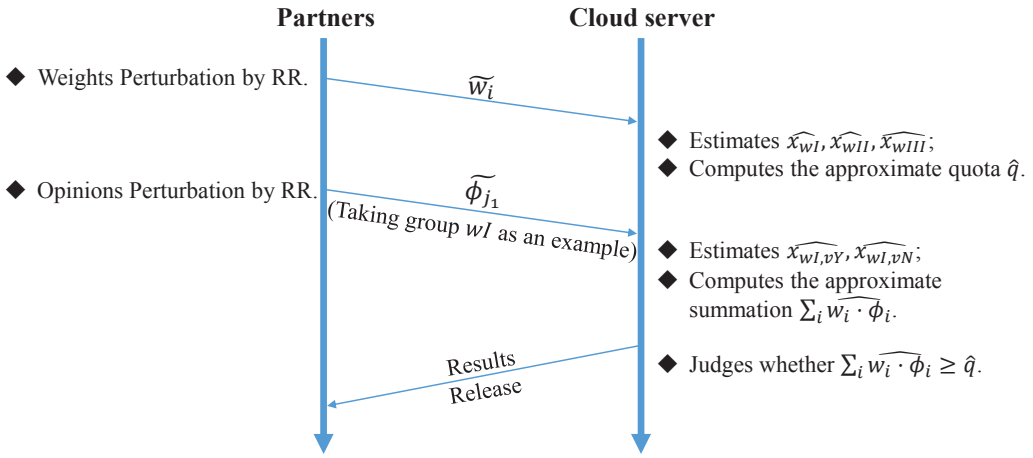


Figure 1. Protocol Overview

the ability to judge whether the approximate summation $\sum_{i \in N} \widehat{w_i \cdot \phi_i}$ is larger than the approximate quota $\widehat{q}$. For each proposal candidates $l \in K = \{1, ..., k\}$, the above phases would be executed sequentially. The cloud server would finally release the winner candidates list to the partners.

Details for the *Randomized Weights Collection* is presented in Section 4.2, followed by the *Randomized Opinions Collection* in Section 4.3 and the *Voting Results Release* in Section 4.4. The related privacy analysis is provided in Section 4.5.

4.2. *Randomized Weights Collection*

In the *Randomized Weights Collection* phase, it requires that on the one hand the partners have the ability to deny their uploaded weights are the actual ones that belong to certain group, and on the other hand the cloud server has the ability to compute the approximate quota $\widehat{q} \approx \frac{1}{2} \sum_{i \in N} \widehat{w_i}$.

Next, we adopt the *Randomized Response* (RR) technique to ensure the above requirements and formulate the problem as categorical data collection. For the sake of simplicity, we use the extended version of the transformation matrix $\mathbf{M}$ proposed in Warner scheme (Warner 1965) to represent the probability of reporting each partner's weight w_i (belong to one weight group) in its original value or a modified version. The transformation matrix of weights is formally described in the following:

$$\mathbf{M}_w = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} = \begin{pmatrix} p_w & \frac{1-p_w}{2} & \frac{1-p_w}{2} \\ \frac{1-p_w}{2} & p_w & \frac{1-p_w}{2} \\ \frac{1-p_w}{2} & \frac{1-p_w}{2} & p_w \end{pmatrix},$$

where all the diagonal elements are assigned to p_w and the rest of elements to $\frac{1-p_w}{2}$.

Based on the above settings, the two protocols that involves interactions of the partners and the cloud server can be further developed. The basic processes of these protocols are shown in Fig. 2. Each partner i generates his/her perturbed weight $\widetilde{w_i}$

locally by RR mechanism and uploads it to the cloud server. In the side of cloud server, it observes that the noisy numbers y_{wI}, y_{wII} and y_{wIII}, and attempt to estimate the approximate numbers $\widehat{x_{wI}}$, $\widehat{x_{wII}}$ and $\widehat{x_{wIII}}$ for further computing the approximate quota $\widehat{q}$.

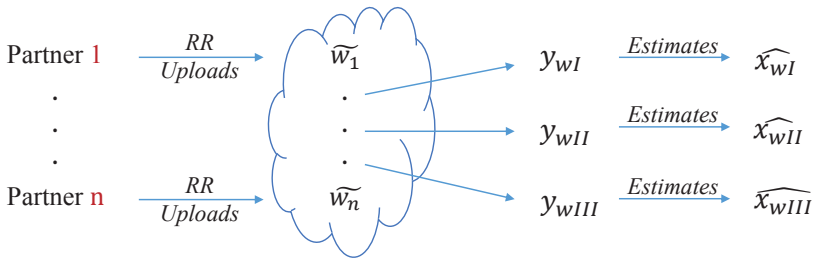


Figure 2. Randomized Weights Collection

Perturbation protocol We assume that each partner would upload his/her weight value by a local RR application that provided by the cloud server. The application contains the transformation matrix $\mathbf{M}_w$ and automatically generate and upload the perturbed weight once the partner decides to join the voting game.

Reconstruction protocol When having collected the perturbed weights $\widetilde{w_i}$ from partners, the cloud server intends to compute the approximate quota $\widehat{q}$ by leveraging the obtained y_{wI}, y_{wII} and y_{wIII}, which are the observed numbers of partners who respectively belong to the weight group wI, wII and $wIII$. It is important to reconstruct the numbers of partners who belong to certain weight group by the following equation, which originates from Eq. 2:

$$\overrightarrow{\widehat{X}_w} = (\mathbf{M}_w)^{-1}\overrightarrow{Y_w}, \tag{3}$$

where $\overrightarrow{\widehat{X}_w} = (\widehat{x_{wI}}, \widehat{x_{wII}}, \widehat{x_{wIII}})^\tau$, $\overrightarrow{Y_w} = (y_{wI}, y_{wII}, y_{wIII})^\tau$, and $(\mathbf{M}_w)^{-1}$ is the inverse matrix of $\mathbf{M}_w$.

Then the cloud server would be able to represent the quota q and compute the approximate quota $\widehat{q}$ by the following equations:

$$\begin{aligned} q &= \frac{1}{2}\sum_{i\in N} w_i \\ &= \frac{1}{2}[wI \cdot x_{wI} + wII \cdot x_{wII} + wIII \cdot x_{wIII}] \end{aligned} \tag{4a}$$

and

$$\widehat{q} = \frac{1}{2}[wI \cdot \widehat{x_{wI}} + wII \cdot \widehat{x_{wII}} + wIII \cdot \widehat{x_{wIII}}], \tag{4b}$$

where $\widehat{x_{wI}}$, $\widehat{x_{wII}}$ and $\widehat{x_{wIII}}$ are refer to the estimated numbers of the partners who belong to group wI, wII and $wIII$, respectively.

4.3. *Randomized Opinions Collection*

In the *Randomized Opinions Collection* phase, it requires that on the one hand the partners have the ability to deny their uploaded opinions actually support or do not support for the given proposal candidate, and on the other hand the cloud server has the ability to estimate the summation $\sum_{i \in N} w_i \cdot \phi_i$ by computing the approximate summation $\sum_{i \in N} \widehat{w_i \cdot \phi_i}$.

Herein we also adopt RR technique to ensure the above requirements and we formulate the problem as binary data collection this time. Firstly, we denote the opinions of every partners are either $\phi Y = 1$ or $\phi N = 0$. Then we use the naive transformation matrix $\mathbf{M}$ proposed in Warner scheme (Warner 1965) to represent the probability of reporting each partner's opinion ϕ_i in its original value or flipped version. The transformation matrix of opinions is formally described in the following:

$$\mathbf{M}_\phi = \begin{pmatrix} p_{0,0} & p_{0,1} \\ p_{1,0} & p_{1,1} \end{pmatrix} = \begin{pmatrix} p_\phi & 1 - p_\phi \\ 1 - p_\phi & p_\phi \end{pmatrix},$$

where all the diagonal elements are assigned to p_ϕ and the rest of elements to $1 - p_\phi$.

Based on the above settings, similar to the last phase, the two protocols that involves interactions of the partners and the cloud server can be further developed. Taking the weight group wI as an example, the basic processes of these protocols are shown in Fig. 3. Each partner of weight group wI generates his/her perturbed opinion $\widetilde{\phi_i}$ locally by RR mechanism and uploads it to the cloud server. In the side of cloud server, it observes that the noisy numbers $y_{wI,\phi Y}$ and $y_{wI,\phi N}$, and attempt to estimate the approximate numbers $\widehat{x_{wI,\phi Y}}$ and $\widehat{x_{wI,\phi N}}$. After the opinions from all the weight groups are collected by the cloud server, it would have the ability to compute the approximate summation $\sum_{i \in N} \widehat{w_i \cdot \phi_i}$.

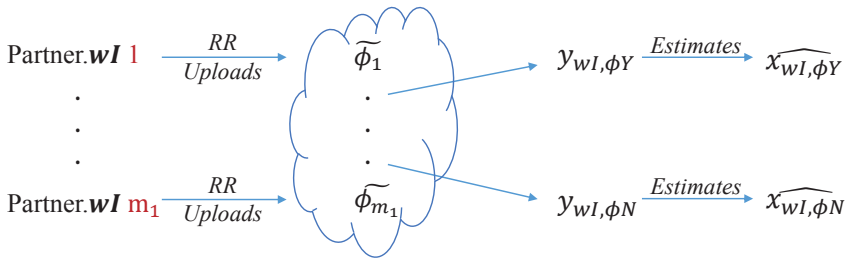


Figure 3. Randomized Opinions Collection

Perturbation protocol Similar to the last phase, a local RR application provided by the cloud server would help the partners automatically generate and upload their perturbed opinions, based on the built-in transformation matrix $\mathbf{M}_\phi$.

Reconstruction protocol Taking the weight group wI as an example, when having collected the perturbed opinions $\widetilde{\phi_{j_1}}$ from partners of wI, the cloud server intends to compute the approximate numbers $\widehat{x_{wI,\phi Y}}$ and $\widehat{x_{wI,\phi N}}$ by leveraging the obtained $x_{wI,\phi Y}$ and $x_{wI,\phi N}$, which are the observed numbers of partners who support and do not support the proposal candidate. It is important to reconstruct the actual numbers of partners' bipartite opinions by the following

equation, which originates from Eq. 2:

$$\overrightarrow{\widehat{X}_{\phi_1}} = (\mathbf{M}_\phi)^{-1}\overrightarrow{Y_{\phi_1}}, \tag{5}$$

where $\overrightarrow{\widehat{X}_{\phi_1}} = (\widehat{x_{wI,\phi Y}}, \widehat{x_{wI,\phi N}})^\tau$, $\overrightarrow{Y_{\phi_1}} = (y_{wI,\phi Y}, y_{wI,\phi N})^\tau$, and $(\mathbf{M}_\phi)^{-1}$ is the inverse matrix of $\mathbf{M}_\phi$.

In this way, the cloud server would also obtain $\overrightarrow{\widehat{X}_{\phi_2}} = (\widehat{x_{wII,\phi Y}}, \widehat{x_{wII,\phi N}})^\tau$ and $\overrightarrow{\widehat{X}_{\phi_3}} = (\widehat{x_{wIII,\phi Y}}, \widehat{x_{wIII,\phi N}})^\tau$. Then it would be able to approximately compute the summation $\sum_{i\in N} w_i \cdot \phi_i$ by the following process.

Firstly, based on the collected weights, the expression of summation $\sum_{i\in N} w_i \cdot \phi_i$ can be written as the addition of each weight group's opinions as shown below.

$$\begin{aligned}\sum_{i\in N} w_i \cdot \phi_i &= w_1 \cdot \phi_1 + \ldots + w_n \cdot \phi_n \\ &= w_1 \sum \phi_{j_1} + w_2 \sum \phi_{j_2} + w_3 \sum \phi_{j_3} \\ &= wI \cdot \phi_{wI} + wII \cdot \phi_{wII} + wIII \cdot \phi_{wIII},\end{aligned} \tag{6}$$

where ϕ_{wI}, ϕ_{wII} and ϕ_{wIII} refer to the summation of the opinions from each weight group respectively.

Then the first term of the above expression can be computed as

$$\begin{aligned}wI \cdot \phi_{wI} &= wI \cdot [x_{wI,\phi Y} \cdot \phi Y + x_{wI,\phi N} \cdot \phi N] \\ &\approx wI \cdot [\widehat{x_{wI,\phi Y}} \cdot \phi Y + \widehat{x_{wI,\phi N}} \cdot \phi N] \\ &= wI \cdot \widehat{x_{wI,\phi Y}},\end{aligned} \tag{7}$$

where $\widehat{x_{wI,\phi Y}}$ is the estimated number of the partners who belong to the weight group wI and support the proposal candidate.

Hence, the summation $\sum_{i\in N} w_i \cdot \phi_i$ can be finally represented and approximately computed by the following equations:

$$\sum_{i\in N} w_i \cdot \phi_i = wI \cdot x_{wI,\phi Y} + wII \cdot x_{wII,\phi Y} + wIII \cdot x_{wIII,\phi Y} \tag{8a}$$

and

$$\sum_{i\in N} \widehat{w_i \cdot \phi_i} = wI \cdot \widehat{x_{wI,\phi Y}} + wII \cdot \widehat{x_{wII,\phi Y}} + wIII \cdot \widehat{x_{wIII,\phi Y}}. \tag{8b}$$

4.4. *Voting Results Release*

Finally, in order to judge whether the summation $\sum_{i\in N} w_i \cdot \phi_i$ is larger than the quota q, which indicates the proposal candidate's winning or losing, the cloud server only need to judge whether the approximate summation $\sum_{i\in N} \widehat{w_i \cdot \phi_i}$ is larger than the approximate quota $\widehat{q}$.

4.5. *Privacy Analysis*

Herein we will prove that the proposed `DPWeVote` protocol satisfies LDP.

As introduced in Section 2.1.3, the elements value of transformation matrix have relationship with ϵ-differential privacy. We adopt the associated value in our proposed protocol according to the following theorems proposed in Wang, Wu, and Hu (2016):

Theorem 4.1 (Binary Attribute Collection & DP)**.** *For a given differential privacy parameter ϵ, the transformation matrix of randomized response scheme for binary attribute collection should have the following pattern,*

$$\boldsymbol{M} = \begin{pmatrix} \frac{e^{\epsilon}}{1+e^{\epsilon}} & \frac{1}{1+e^{\epsilon}} \\ \frac{1}{1+e^{\epsilon}} & \frac{e^{\epsilon}}{1+e^{\epsilon}} \end{pmatrix}.$$

Theorem 4.2 (Polychotomous Attribute Collection & DP)**.** *For a given differential privacy parameter ϵ, the transformation matrix $\boldsymbol{M} = \{p_{j,i}\}$ of randomized response scheme for polychotomous attribute collection should have the following form,*

$$p_{j,i} = \begin{cases} \frac{e^{\epsilon}}{t-1+e^{\epsilon}}; & if\ j = i, \\ \frac{1}{t-1+e^{\epsilon}}; & if\ j \neq i. \end{cases}$$

Based on the above theorems, we set

$$p_w = \frac{e^{\epsilon_1}}{2 + e^{\epsilon_1}} \tag{9a}$$

and

$$p_\phi = \frac{e^{\epsilon_2}}{1 + e^{\epsilon_2}}, \tag{9b}$$

where $\epsilon = \epsilon_1 + \epsilon_2$. Then we have the following theorem on the privacy guarantee of the proposed protocol.

Theorem 4.3. *The `DPWeVote` protocol satisfies ϵ-LDP.*

Proof. Two independent phases regarding privately data collection of the `DPWeVote` protocol can respectively satisfy relevant level of differential privacy as follows:

(1) For the *Randomized Weights Collection* phase, since we adopt a local randomizer $\mathcal{R}_w$ to perturb and collect polychotomous attribute of weights and use the following transformation matrix:

$$\mathbf{M}_w = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} = \begin{pmatrix} p_w & \frac{1-p_w}{2} & \frac{1-p_w}{2} \\ \frac{1-p_w}{2} & p_w & \frac{1-p_w}{2} \\ \frac{1-p_w}{2} & \frac{1-p_w}{2} & p_w \end{pmatrix},$$

we then list the ratios between any pair of $p_{u,u}$ and $p_{u,v}$ as follows:

$$\frac{p_{1,1}}{p_{1,2}} = \frac{p_{1,1}}{p_{1,3}} = \frac{p_{2,2}}{p_{2,1}} = \frac{p_{2,2}}{p_{2,3}} = \frac{p_{3,3}}{p_{3,1}} = \frac{p_{3,3}}{p_{3,2}} = \frac{2p_w}{1-p_w}.$$

Based on the setting under Theorem 4.2, we can obtain

$$\frac{Pr[\mathcal{R}_w(w_i = u) = u]}{Pr[\mathcal{R}_w(w_i = v) = u]} = \frac{2p_w}{1 - p_w} = e^{\epsilon_1}.$$

In the meanwhile, all the ratios between any pair from p_{u,v_i} (where $v_i = 2$ or 3) equal to 1, which can be seen as e^0. Based on Definition 2.7, we can conclude that the *Randomized Weights Collection* phase satisfies ϵ_1-LDP where $\epsilon_1 = ln(\frac{2p_w}{1-p_w})$.

(2) For the *Randomized Opinions Collection* phase, since we adopt a local randomizer $\mathcal{R}_\phi$ to perturb and collect binary attribute of opinions and use the following transformation matrix:

$$\mathbf{M}_\phi = \begin{pmatrix} p_{0,0} & p_{0,1} \\ p_{1,0} & p_{1,1} \end{pmatrix} = \begin{pmatrix} p_\phi & 1 - p_\phi \\ 1 - p_\phi & p_\phi \end{pmatrix},$$

we then list the ratios between any pair of $p_{u,u}$ and $p_{u,v}$ as follows:

$$\frac{p_{0,0}}{p_{0,1}} = \frac{p_{1,1}}{p_{1,0}} = \frac{p_\phi}{1 - p_\phi}.$$

Based on the setting under Theorem 4.1, we can obtain

$$\frac{Pr[\mathcal{R}_\phi(\phi_i = u) = u]}{Pr[\mathcal{R}_\phi(\phi_i = v) = u]} = \frac{p_\phi}{1 - p_\phi} = e^{\epsilon_2}.$$

Then based on Definition 2.7, we can conclude that the *Randomized Opinions Collection* phase satisfies ϵ_2-LDP where $\epsilon_2 = ln(\frac{p_\phi}{1-p_\phi})$.

Consequently, according to the *Sequential Composition* which is shown in Theorem 2.6, we can conclude that the `DPWeVote` protocol satisfies ϵ-LDP where $\epsilon = \epsilon_1 + \epsilon_2$. □

5. Experiment and Analysis

In this section, we design experiments to evaluate the performance of the proposed `DPWeVote` protocol by answering the following questions:

How does the `DPWeVote` protocol retain the utility? The `DPWeVote` protocol aims to release voting results with acceptable utility. In Section 5.2, we will examine its performance in terms of *Accuracy* and *Mean Squared Error* (MSE) respectively on the intermediate estimations and the released results. The comparisons with a Laplace mechanism based baseline algorithm are provided.

How do the parameters affect the protocol performance? There are two parameters ϵ and n involved in the `DPWeVote` protocol: privacy budget ϵ controls the overall privacy protection level of protocol; and n determines the total numbers of the partners who participate in the weighted voting game. In Section 5.2.1 and 5.2.2, we will present and analyse their impacts on the protocol performance.

5.1. *Experiment Setting*

5.1.1. Data and Configuration

We evaluate the `DPWeVote` protocol and the baseline algorithm on synthetic data, in which the data of weights and opinions of each partner are i.i.d. generated from uniform distribution. The generated weight data will be assigned to the three weight groups wI, wII and $wIII$, while the generated opinions data to the two opinion groups ϕY and ϕN. Besides, we set the size of each weight group or opinion group are same.

The `DPWeVote` protocol and the baseline algorithm are implemented in Python 2.7 codes and all the experiments are conducted on an Intel Core i5-3210M 2.50GHz PC with 6GB memory. In each experiment, the protocol is executed 2000 times, and its average score is reported. As to the setting of ϵ_1 and ϵ_2, for simplicity we make $\epsilon_1 = \epsilon_2 = \epsilon/2$.

5.1.2. Experiment Parameters

We consider two parameters ϵ and n since the performance of protocol could be affected by them:

The privacy budget ϵ. Although the relationship between the privacy budget ϵ and the utility under the single RR mechanism has been investigated in previous studies, here we also expect to explore the situation in which the double RR mechanism are adopted. Besides, we will observe the performance of the `DPWeVote` protocol in two scales of observation range: one is for $\epsilon \in [0.1, 1.0]$, which is a general range for evaluating differentially private algorithms; and the another is for $\epsilon \in [0.1, 5.0]$, which is a wider range for evaluating RR based LDP algorithms as the previous studies (Wang, Wu, and Hu 2016; Qin et al. 2016; Wang et al. 2017; Bassily et al. 2017) adopted.

The number of partners n. The number of data contributors in single RR mechanism always affects the final utility, as investigated in previous studies. However, there is no guarantee for the double RR mechanism. Here, we aim to explore the utility of final results and intermediate estimations of the `DPWeVote` protocol, respectively. And we expect that the protocol could perform well even when n is relatively small. Specifically, we set $n = 10, 100, 500$ when observing the results of `DPWeVote` with $\epsilon \in [0.1, 5.0]$, and set $n = 10, 50, 100$ when comparing `DPWeVote` with baseline algorithm at small ϵ, which is in $[0.1, 1.0]$.

In our experiments, we will vary the above parameters to investigate their impacts on the proposed protocol, in terms of the utility metrics as mentioned in Section 5.1.4.

5.1.3. Compared Algorithm

When varying ϵ in range $[0.1, 1.0]$, we consider a *Baseline* algorithm as the competitor of the `DPWeVote` protocol. The baseline algorithm is based on the *Output Perturbation* approach which introduces differential privacy by directly adding *Laplacian* noise to the local original values of weight w_i and opinion ϕ_i on each partner's side.

For the perturbation of the weight of partner i, we use $f_w(\cdot)$ to denote the weight query function on i's weight data whose size is one.

$$\widetilde{f_w(i)} = f_w(i) + Laplace(\frac{\Delta f_w}{\epsilon_1}).$$

The added Lapacian noise is calibrated to Δf_w (the sensitivity of $f_w(\cdot)$), which is equal to 2 since the maximum change of w_i is resulted from the weight group changing between wI and $wIII$.

Similarly, the perturbation of the opinion of partner i can be formulated as follows:

$$\widetilde{f_\phi(i)} = f_\phi(i) + Laplace(\frac{\Delta f_\phi}{\epsilon_2}).$$

We use $f_\phi(\cdot)$ to denote the opinion query function on i's opinion data whose size is also one. The sensitivity of $f_\phi(\cdot)$ is equal to 1 since the maximum change of ϕ_i is resulted from the opinion group changing between ϕY and ϕN.

In this way, the partners would first use baseline algorithm perturb their local data and then upload. Based on these collected perturbed data, the cloud server would calculate and release the final result. Since the baseline algorithm intuitively adds coarse-grained noise to achieve differential privacy, we expect that it will underperform the `DPWeVote` protocol.

5.1.4. Utility Metrics

We adopt the *Accuracy* and *Mean Squared Error* (MSE) to measure the utility performance regarding the final results and intermediate estimations of the proposed `DPWeVote` protocol and the baseline algorithm.

Accuracy. In general, *Accuracy* measures the systematic errors, which cause the differences between the observed results and the true values. We aim to investigate the systematic errors of `DPWeVote` by observing the statistical differences between the estimated voting results and the actual ones in several tests. The accuracy can be calculated as

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{10}$$

where TP is the true positive, FP is the false positive, TN is the true negative and FN is false negative. Table 2 shows the semantics of these four variables.

Table 2. Semantics of TP, FP, TN and FN

TP	The number of test in which **both** the estimated voting result and the actual one for a given proposal candidate are PASS.
FP	The number of test in which **only** the estimated voting result is PASS the given proposal candidate and the actual one is NOT.
TN	The number of test in which **both** the estimated voting result and the actual one for a given proposal candidate are NOT PASS.
FN	The number of test in which **only** the estimated voting result is NOT PASS the given proposal candidate and the actual one is PASS.

Mean Squared Error (MSE). The *Mean Squared Error* (MSE) is a measure to quantify the difference between an estimator and what is estimated. In general, a lower MSE means a better utility the protocol can achieve. Herein we aim to investigate the MSE of three intermediate estimations within the `DPWeVote`

protocol as shown below:

$$
\begin{aligned}
MSE_w &= \mathbb{E}[(\frac{\widehat{\overrightarrow{X_w}}}{n} - \frac{\overrightarrow{X_w}}{n})^2] \\
&= \frac{1}{3}[(\frac{\widehat{x_{wI}}}{n} - \frac{x_{wI}}{n})^2 + (\frac{\widehat{x_{wII}}}{n} - \frac{x_{wII}}{n})^2 + (\frac{\widehat{x_{wIII}}}{n} - \frac{x_{wIII}}{n})^2],
\end{aligned} \tag{11}
$$

$$
MSE_q = \mathbb{E}[(\frac{\widehat{q}}{\sum_{i \in N} w_i} - \frac{q}{\sum_{i \in N} w_i})^2] \tag{12}
$$

and

$$
\begin{aligned}
MSE_\phi &= \mathbb{E}[(\frac{\widehat{\overrightarrow{X_{\phi Y}}}}{n} - \frac{\overrightarrow{X_{\phi Y}}}{n})^2] \\
&= \frac{1}{3}[(\frac{\widehat{x_{wI,\phi Y}}}{n} - \frac{x_{wI,\phi Y}}{n})^2 + (\frac{\widehat{x_{wII,\phi Y}}}{n} - \frac{x_{wII,\phi Y}}{n})^2 \\
&\qquad + (\frac{\widehat{x_{wIII,\phi Y}}}{n} - \frac{x_{wIII,\phi Y}}{n})^2],
\end{aligned} \tag{13}
$$

where n is the number of partners in N.

For the baseline algorithm, since it does not produce the intermediate estimations $\frac{\widehat{\overrightarrow{X_w}}}{n}$ and $\frac{\widehat{\overrightarrow{X_{\phi Y}}}}{n}$, we will only focus on its MSE_q (the MSE in terms of $\frac{\widehat{q}}{\sum_{i \in N} w_i}$).

5.2. *The Performance of* `DPWeVote`

5.2.1. Impact of Privacy Budget

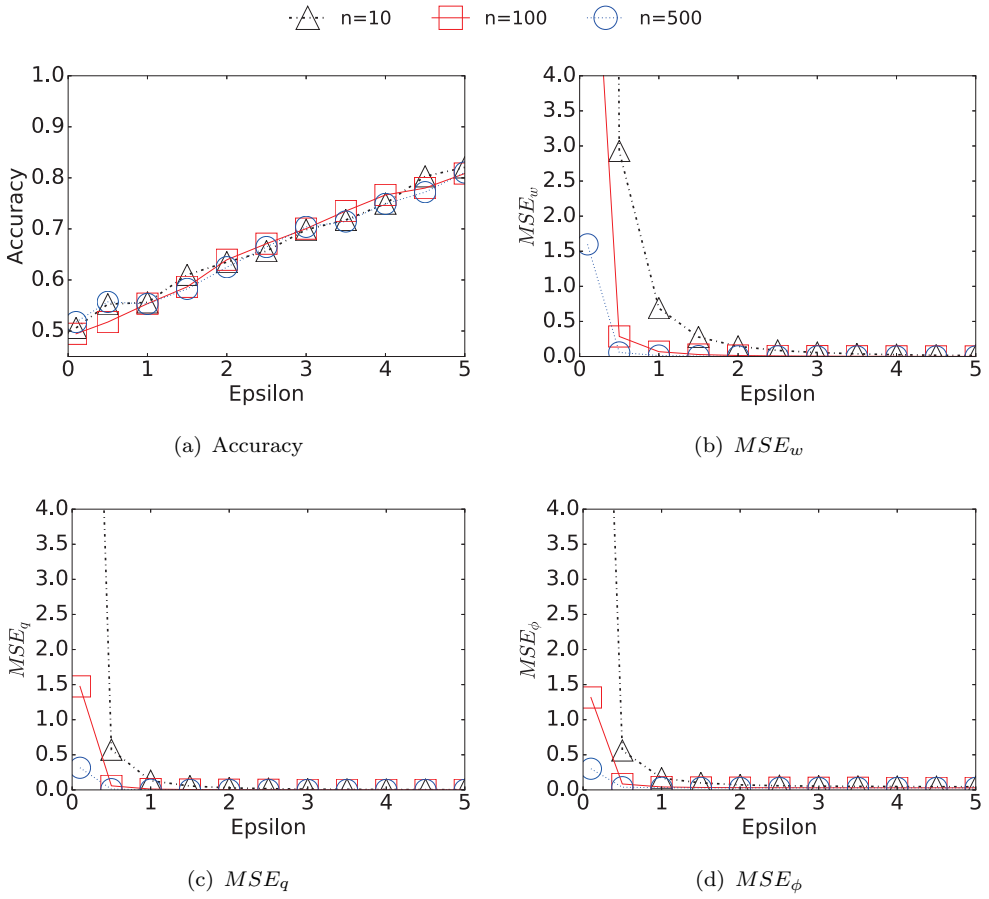
In this experiment, we set $n = 10$ and present the four kinds of utility measures of the `DPWeVote` protocol when varying ϵ from 0.1 to 5.0. Fig. 4(a) shows the accuracy over synthetic datasets along with the change of privacy budget ϵ. Overall, we observe that the `DPWeVote` protocol has higher accuracy when increasing ϵ, which is consistent with the general law in differentially private algorithm design. Specifically, there is basically a linear relationship between accuracy and ϵ.

For the MSE measurement, Fig. 4(a)(b) and (c) show the estimated proportions of $\widehat{\overrightarrow{X_w}}$, $\widehat{q}$ and $\widehat{\overrightarrow{X_{\phi Y}}}$ in the related total quantity. We observe that the magnitudes of MSE decreases sharply with a modest small ϵ such as from the range 0.1 to 1.0. Interestingly, although the MSE of the three intermediate estimations can be changed significantly in exponential with varying ϵ, there have no such effect on the current linear relationship between the final results' accuracy and ϵ.

5.2.2. Impact of Partner Number

In this experiment, we present the performances of the `DPWeVote` protocol when $n = 10, 100, 500$. Somewhat surprisingly, as shown in Fig. 4(a), the impact of partner number on the final results' accuracy is unclear. We suspect that this is due to the possible internal offset effect within double RR mechanism, which should be further studied in future.

In contrast, for the MSE measurement, Fig. 4(a)(b) and (c) show that the `DPWeVote` protocol with a larger number of partners generally outperforms that with a smaller



(a) Accuracy (b) MSE_w

(c) MSE_q (d) MSE_ϕ

Figure 4. (a) shows the Accuracy of the final results over synthetic data. (b)(c) and (d) show the MSE of three intermediate estimations over the same synthetic data.

number, although the differences become less obvious when the ϵ is larger than 1. Another observation is that the MSE_q is very close to MSE_ϕ from the aspect of variation trends.

5.2.3. Comparison with Baseline Algorithm

For evaluating the differentially private algorithms in general range where ϵ is usually changed from 0.1 to 1.0, here we compare the performance of `DPWeVote` protocol with that of baseline algorithm in a narrow range of ϵ, and set $n = 10, 50, 100$. Table 3 and Table 4 present the comparison results in terms of accuracy and MSE_q, respectively.

As to the results of accuracy, we see the trend that both of `DPWeVote` protocol and baseline algorithm increased slowly when tuning ϵ from 0.1 to 1.0. But obviously, the results show that the `DPWeVote` protocol outperforms the baseline algorithm with a larger growth rate. Besides, we also cannot find a clear effect by changing n.

Table 3. Comparison of the `DPWeVote` protocol and Baseline algorithm in terms of Accuracy

Accuracy	ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$n = 10$	Baseline	0.49440	0.50120	0.50125	0.50128	0.50140	0.51145	0.51185	0.51995	0.52025	0.52695
	DPWeVote	0.50680	0.51265	0.51665	0.52675	0.53345	0.53700	0.54660	0.55505	0.55540	0.56840
$n = 50$	Baseline	0.50060	0.50043	0.50105	0.50415	0.50500	0.50985	0.51103	0.51120	0.51790	0.52135
	DPWeVote	0.50795	0.50920	0.51725	0.52370	0.52550	0.53820	0.54690	0.55195	0.56060	0.56265
$n = 100$	Baseline	0.49840	0.50053	0.50150	0.50260	0.50450	0.50890	0.51195	0.51225	0.51300	0.52010
	DPWeVote	0.50800	0.51340	0.51070	0.52335	0.53110	0.53630	0.54345	0.54510	0.55700	0.56310

The results of MSE_q show that the baseline algorithm significantly underperforms the `DPWeVote` protocol. And for a larger n, both of them get lower MSE_q as expected.

Table 4. Comparison of the `DPWeVote` protocol and Baseline algorithm in terms of MSE_q

MSE_q	ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$n = 10$	Baseline	20.80675	5.18172	2.34181	1.31362	0.82597	0.59720	0.42769	0.33390	0.26256	0.20914
	DPWeVote	15.82780	3.79594	1.68442	0.92401	0.59020	0.39621	0.28239	0.21623	0.16892	0.13490
$n = 50$	Baseline	4.00614	1.00797	0.44805	0.25437	0.16142	0.11203	0.08213	0.06390	0.04941	0.04070
	DPWeVote	3.01404	0.74125	0.31822	0.17802	0.11303	0.07640	0.05671	0.04168	0.03253	0.02548
$n = 100$	Baseline	1.97664	0.50439	0.22056	0.12592	0.08012	0.05566	0.04160	0.03130	0.02509	0.01985
	DPWeVote	1.48116	0.36118	0.16328	0.08678	0.05549	0.03759	0.02717	0.02070	0.01608	0.01292

5.3. *Summary and Recommendations*

The above experimental results basically verify that the proposed `DPWeVote` protocol can preserve an acceptable utility while maintaining a relatively strong ϵ-differential privacy. Due to the inner limitations of RR technique, which have been shown in the previous research such as Wang, Wu, and Hu (2016) and Qin et al. (2016), the `DPWeVote` protocol requires the ϵ to be larger than 1 for getting better final results' accuracy. Nonetheless, the `DPWeVote` protocol can always outperform Laplace mechanism including the situations where the ϵ is from a narrower range. This paper aims to show a possible way to achieve differentially private weighted voting game by adopting

naive RR techniques. In principle, it can be further optimized by carefully selecting of RR transformation matrices, as well as by assigning the overall privacy budget ϵ to ϵ_1 and ϵ_2.

6. Conclusions

In the light of Industry 4.0, *Weighted Voting Game* as a common form of decision-making problem is being deployed into the could environment. However, the potential privacy leakage within the cloud-based weighted voting game is an emerging issue that must be addressed. Current research based on cryptographic approaches are limited to some of the strong assumptions in their security model. This paper studies the possibility of achieving local differential privacy in a cloud-based weighted voting game by using *Randomized Response* (RR) technique. Specifically, based on the semi-honest cloud server/partners and the maximal collusion tolerance assumptions, we design a three-phase protocol `DPWeVote`, in which the privacy of weights and opinions from each individual can be protected separately in a double RR mechanism. In the meanwhile, the empirical results show that the proposed `DPWeVote` protocol can maintain an acceptable utility under the recommendation for parameter setting. Compared with a underperforming Laplace mechanism based baseline algorithm, the `DPWeVote` protocol can be a potential solution to the differentially private weighted voting game.

We are eager to continue this work by exploring the related theoretical bounds for double RR mechanism and the selection of parameters and RR transformation matrices for optimization. And we also plan to consider the assumption that a certain weight group may contain different weight values.

References

Ashur, Tomer, Orr Dunkelman, and Nimrod Talmon. 2016. "Breaching the Privacy of Israel's Paper Ballot Voting System." In *E-VOTE-ID*, Vol. 10141 of *Lecture Notes in Computer Science*, 108–124. Springer.

Avent, Brendan, Aleksandra Korolova, David Zeber, Torgeir Hovden, and Benjamin Livshits. 2017. "BLENDER: Enabling Local Search with a Hybrid Differential Privacy Model." In *USENIX Security Symposium*, 747–764. USENIX Association.

Bassily, Raef, Kobbi Nissim, Uri Stemmer, and Abhradeep Guha Thakurta. 2017. "Practical Locally Private Heavy Hitters." In *NIPS*, 2285–2293.

Bassily, Raef, and Adam D. Smith. 2015. "Local, Private, Efficient Protocols for Succinct Histograms." In *STOC*, 127–135. ACM.

Bernhard, David, Véronique Cortier, Olivier Pereira, and Bogdan Warinschi. 2012. "Measuring vote privacy, revisited." In *ACM Conference on Computer and Communications Security*, 941–952. ACM.

Bernhard, Matthew, Josh Benaloh, J. Alex Halderman, Ronald L. Rivest, Peter Y. A. Ryan, Philip B. Stark, Vanessa Teague, Poorvi L. Vora, and Dan S. Wallach. 2017. "Public Evidence from Secret Ballots." In *E-VOTE-ID*, Vol. 10615 of *Lecture Notes in Computer Science*, 84–109. Springer.

Branzei, Rodica, Dinko Dimitrov, and Stef Tijs. 2008. *Models in cooperative game theory*. Vol. 556. Springer.

Chalkiadakis, Georgios, Edith Elkind, and Michael Wooldridge. 2011. *Computational Aspects of Cooperative Game Theory*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Chaudhuri, Arijit. 2016. *Randomized response and indirect questioning techniques in surveys.* CRC Press.

Chen, Yiling, Stephen Chong, Ian A. Kash, Tal Moran, and Salil P. Vadhan. 2013. "Truthful mechanisms for agents that value privacy." In *EC*, 215–232. ACM.

Chen, Yu-Jia, Chia-Yu Lin, and Li-Chun Wang. 2013. "A Privacy-Preserved Joint Group Time Scheduling Mechanism for Mobile Social Applications." In *TrustCom/ISPA/IUCC*, 150–155. IEEE Computer Society.

Cisco. 2015. "Cloud ERP for Asset Intensive Industries." Http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/sap-applications-on-ucs/cloud-erp-asset-intensive-industries.pdf.

Cloud, 360. 2016. "How Does Cloud ERP Software Improve Business Decisions." Http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/sap-applications-on-ucs/cloud-erp-asset-intensive-industries.pdf.

Duchi, John C., Michael I. Jordan, and Martin J. Wainwright. 2012. "Privacy Aware Learning." In *NIPS*, 1439–1447.

Duchi, John C., Michael I. Jordan, and Martin J. Wainwright. 2013. "Local Privacy and Statistical Minimax Rates." In *FOCS*, 429–438. IEEE Computer Society.

Dwork, Cynthia. 2011. "Differential Privacy." In *Encyclopedia of Cryptography and Security (2nd Ed.)*, 338–340. Springer.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. "Calibrating Noise to Sensitivity in Private Data Analysis." In *TCC*, Vol. 3876 of *Lecture Notes in Computer Science*, 265–284. Springer.

Erlingsson, Úlfar, Vasyl Pihur, and Aleksandra Korolova. 2014. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response." In *ACM Conference on Computer and Communications Security*, 1054–1067. ACM.

ERP, Inside. 2016. "Making the Cloud Decision in ERP." Http://it.toolbox.com/blogs/inside-erp/making-the-cloud-decision-in-erp-74266.

Grewal, Gurchetan S., Mark Dermot Ryan, Liqun Chen, and Michael R. Clarkson. 2015. "Du-Vote: Remote Electronic Voting with Untrusted Computers." In *CSF*, 155–169. IEEE Computer Society.

Groat, Michael M., Benjamin Edwards, James Horey, Wenbo He, and Stephanie Forrest. 2013. "Application and analysis of multidimensional negative surveys in participatory sensing applications." *Pervasive and Mobile Computing* 9 (3): 372–391.

Hay, Michael, Liudmila Elagina, and Gerome Miklau. 2017. "Differentially Private Rank Aggregation." In *SDM*, 669–677. SIAM.

Huang, Zhengli, and Wenliang Du. 2008. "OptRR: Optimizing Randomized Response Schemes for Privacy-Preserving Data Mining." In *ICDE*, 705–714. IEEE Computer Society.

Kairouz, Peter, Sewoong Oh, and Pramod Viswanath. 2014. "Extremal Mechanisms for Local Differential Privacy." In *NIPS*, 2879–2887.

Kairouz, Peter, Sewoong Oh, and Pramod Viswanath. 2015. "Secure Multi-party Differential Privacy." In *NIPS*, 2008–2016.

Kairouz, Peter, Sewoong Oh, and Pramod Viswanath. 2016. "Differentially private multi-party computation." In *CISS*, 128–132. IEEE.

Kasiviswanathan, Shiva Prasad, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2008. "What Can We Learn Privately?" In *FOCS*, 531–540. IEEE Computer Society.

Lee, David Timothy. 2015. "Efficient, Private, and eps-Strategyproof Elicitation of Tournament Voting Rules." In *IJCAI*, 2026–2032. AAAI Press.

Leung, Samantha, and Edward Lui. 2012. "Bayesian Mechanism Design with Efficiency, Privacy, and Approximate Truthfulness." In *WINE*, Vol. 7695 of *Lecture Notes in Computer Science*, 58–71. Springer.

Leyton-Brown, Kevin, and Yoav Shoham. 2008. *Essentials of Game Theory: A Concise Multidisciplinary Introduction.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

McSherry, Frank. 2009. "Privacy integrated queries: an extensible platform for privacy-preserving data analysis." In *SIGMOD Conference*, 19–30. ACM.

McSherry, Frank, and Kunal Talwar. 2007. "Mechanism Design via Differential Privacy." In *FOCS*, 94–103. IEEE Computer Society.

Nakanishi, Toru, Shinji Nakatake, Nobuo Funabiki, and Yuji Sugiyama. 2004. "An efficient weighted voting protocol with secret weights." In *International Symposium on Information Theory and Its Applications (ISITA)*, .

Nissim, Kobbi, Rann Smorodinsky, and Moshe Tennenholtz. 2012. "Approximately optimal mechanism design via differential privacy." In *ITCS*, 203–213. ACM.

Park, Sunoo, and Ronald L Rivest. 2017. "Towards secure quadratic voting." *Public Choice* 172 (1-2): 151–175.

Qin, Zhan, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2016. "Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy." In *ACM Conference on Computer and Communications Security*, 192–203. ACM.

Riemann, Robert. 2017. "Towards Trustworthy Online Voting: Distributed Aggregation of Confidential Data. (Confiance dans le vote en ligne : Agrégation distribuée de données confidentielles)." PhD diss., École normale supérieure de Lyon, France.

Rivest, Ronald L., Philip B. Stark, and Zara Perumal. 2017. "BatchVote: Voting Rules Designed for Auditability." In *Financial Cryptography Workshops*, Vol. 10323 of *Lecture Notes in Computer Science*, 317–333. Springer.

Schaefer, Dirk. 2014. *Cloud-Based Design and Manufacturing (CBDM): A Service-Oriented Product Development Paradigm for the 21st Century*. Springer.

Sei, Yuichi, and Akihiko Ohsuga. 2017. "Differential Private Data Collection and Analysis Based on Randomized Multiple Dummies for Untrusted Mobile Crowdsensing." *IEEE Trans. Information Forensics and Security* 12 (4): 926–939.

Springall, Drew, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. 2014. "Security Analysis of the Estonian Internet Voting System." In *ACM Conference on Computer and Communications Security*, 703–715. ACM.

Steinberger, Jeffrey. 2007. "Control in a General Partnership." Https://www.entrepreneur.com/article/186122.

Symonds, Mark. 2012. "Software & Analysis: Cloud ERP Meets Manufacturing." Http://www.qualitymag.com/articles/88479-software—analysis–cloud-erp-meets-manufacturing.

Synergy, World. 2015. "Cloud ERP System for EnterprisesCKey Considerations." Https://www.worldsynergy.com/wp-content/uploads/2017/05/Evaluation-criteria-what-you-should-be-looking-for-in-a-cloud-ERP-vendor.pdf.

Talmon, Nimrod. 2015. "Privacy in Elections: k-Anonymizing Preference Orders." In *FCT*, Vol. 9210 of *Lecture Notes in Computer Science*, 299–310. Springer.

Thames, Lane, and Dirk Schaefer. 2017. "Industry 4.0: An Overview of Key Benefits, Technologies, and Challenges." In *Cybersecurity for Industry 4.0*, 1–33. Springer.

Wang, Tianhao, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. "Locally Differentially Private Protocols for Frequency Estimation." In *USENIX Security Symposium*, 729–745. USENIX Association.

Wang, Yue, Xintao Wu, and Donghui Hu. 2016. "Using Randomized Response for Differential Privacy Preserving Data Collection." In *EDBT/ICDT Workshops*, Vol. 1558 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Warner, Stanley L. 1965. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias." *Journal of the American Statistical Association* 60 (309): 63–69.

Will, Mark A., Brandon Nicholson, Marc Tiehuis, and Ryan K. L. Ko. 2015. "Secure Voting in the Cloud Using Homomorphic Encryption and Mobile Agents." In *ICCCRI*, 173–184. IEEE Computer Society.

Wu, Dazhong, J Lane Thames, David W Rosen, and Dirk Schaefer. 2012. "Towards a Cloud-Based Design and Manufacturing Paradigm: Looking Backward, Looking Forward." In *Proceedings of the ASME 2012 International Design Engineering Technical Conference & Com-*

puters and Information in Engineering Conference (IDETC/CIE12), Vol. 17.

Wu, Dazhong, J Lane Thames, David W Rosen, and Dirk Schaefer. 2013. "Enhancing the product realization process with cloud-based design and manufacturing systems." *Journal of Computing and Information Science in Engineering* 13 (4): 041004.

Xiao, David. 2013. "Is privacy compatible with truthfulness?" In *ITCS*, 67–86. ACM.

Zhang, Mingwu, Yong Xia, Ou Yuan, and Kirill Morozov. 2016. "Privacy-friendly weighted-reputation aggregation protocols against malicious adversaries in cloud services." *Int. J. Communication Systems* 29 (12): 1863–1872.