Privacy-preserving Crowd-sensed trust aggregation in the User-centeric Internet of People Networks

MUHAMMAD AJMAL AZAD, School of Computing, The University of Derby, United Kingdom
CHARITH PERERA, School of Computer Science and Informatics, Cardiff University, United Kingdom
SAMIRAN BAG, Department of Computer Science, Warwick University, United Kingdom
MAHMOUD BARHAMGI, Claude Bernard Lyon 1 University, France, France
FENG HAO, Department of Computer Science, Warwick University, United Kingdom

Today we are relying on Internet technologies for numerous services, for example, personal communication, online businesses, recruitment, and entertainment. Over these networks, people usually create content, a skillful worker profile, provide services that are normally watched and used by other users, thus developing a social network among people termed as the Internet of People. Malicious users could also utilize such platforms for spreading unwanted content that could bring catastrophic consequences to a social network provider and the society, if not identified on time. The use of trust management over these networks plays a vital role in the success of these services. Crowd-sensing people or network users for their views about certain content or content creators could be a potential solution to assess the trustworthiness of content creators and their content. However, the human involvement in crowd-sensing would have challenges of privacy-preservation and preventing intentional assignment of the fake high score given to certain user/content. To address these challenges, in this paper, we propose a novel trust model that evaluates the aggregate trustworthiness of the content creator and the content without compromising the privacy of the participating people in a crowdsource group. The proposed system has inherent properties of privacy-protection of participants, performs operations in the decentralized setup and considers the trust weights of participants in a private and secure way. The system ensures privacy of participants under the malicious and honest-but-curious adversarial models. We evaluated the performance of the system by developing a prototype and applying it to different real data from different online social networks.

CCS Concepts: • **Network Security**; • **Cybersecurity**; • **Social Networks**; • **Networks** → Network reliability;

Additional Key Words and Phrases: Content Rating, Trustworthiness, Crowdsourcing, Privacy-preserving system

ACM Reference Format:
Muhammad Ajmal Azad, Charith Perera, Samiran Bag, Mahmoud Barhamgi, and Feng Hao. 2018. Privacy-preserving Crowd-sensed trust aggregation in the User-centeric Internet of People Networks. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 24 pages. https://doi.org/10.1145/1122445.1122456

Authors' addresses: Muhammad Ajmal Azad, School of Computing, The University of Derby, United Kingdom, m.azad@derby.ac.uk; Charith Perera, School of Computer Science and Informatics, Cardiff University, United Kingdom, pererac@cardiff.ac.uk; Samiran Bag, Department of Computer Science, Warwick University, United Kingdom, samiran.bag@warwick.ac.uk; Mahmoud Barhamgi, Claude Bernard Lyon 1 University, France, France, mahmoud.barhamgi@univ-lyon1.fr; Feng Hao, Department of Computer Science, Warwick University, United Kingdom, feng.hao@warwick.ac.uk.

2476-1249/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Online social networks (Facebook, Twitter, Youtube, etc.) provide an opportunity for users to post and disseminate their stories and videos (breaking news, stories, video content, etc.) to a large number of viewers. Today, content providers have different purposes and operate in different settings, e.g. a yelp network allows its customer to review products and businesses on their networks; StackOverflow helps scientific community to share the solutions on technical problems; Wikipedia provides a platform for the collaborative information sharing and editing, and Youtube allows users to create, share and monetize their video content. The dissemination of unwanted, inappropriate and fake content would not only bring a bad image to the content provider but also have serious damage in terms of finance, social and psychology of the users. For example, copying and pasting code from the StackOverflow snippets in commercial products and applications would have serious security and privacy issues [1], or running advertisements on inappropriate content would result in backfire from advertisers or spread of malicious content would have serious social and psychological consequences. In 2017, leading companies across the world have pulled out their advertisements from Youtube after discovering that their advertisements have been shown on the videos that contain hate speech, religiously extreme content or suicide scenes [2–4].

Due to the openness of online platforms, objectionable/inappropriate content is common; however, there have been efforts to identify and block such content in a timely manner. Historically, these approaches analyze content and content metadata including the social network attributes and user comments to identify objectionable content [5–7]. For instance, Aggarwal et al. [6] presented an effort supported by machine learning to identify offensive or objectionable content on YouTube by conducting a manual analysis on the content posted over the content sharing platform. Recent advancements have been focused on video content analysis [8–10]; however, there remain limitations in achieving an effective, objective and timely decision. Another way to fight against is to rely on human intelligence in the form of user feedback, which can be used to protect the users from the hate, fake and inappropriate content.

Crowdsourcing is a distributed mechanism that enables a selected set of users to provide their views about the specific task, product or content. The crowdsourcing process has seen applied in many domains, e.g. conducting the survey and securing the network from the malicious actors [11, 12]. Online social networks including Google, eBay, Amazon, IMDB, etc. are heavily relying on their users to fight against unwanted, fake, hate and inappropriate content. For example, the Youtube network has developed a system that utilizes the automated system and the feedback from a set of trusted users to decide about the permissibility of the content [13] on its network. Specifically, during the period from October to December 2017, Youtube has deleted 8.3 million videos, out of which 1.5 million were deleted based on the feedback from the set of hired users [13]. Figure 1 represents the questioner or query sent by Youtube to its users for the video rating. Similarly, Facebook asked its users to provide their naked photos so they should train their machine learning system to fight against the revenge porn [3]. However, users may feel reluctant to take part in collaboration because they care about their privacy, which could not be maintained if data and feedback ratings are not well protected. In P2P file-sharing systems, crowdsourcing is used to identify whether the content provided by the host is real or fake [14].

The existing works on privacy-preserving crowdsourcing and aggregation are based on two types. 1) protecting the user data by using cryptography [15, 16] and differential privacy techniques [17, 18], 2) using a trusted data collector system [19, 20]. In a trusted data collector setup, the content provider has to get informed consent for collecting and processing the user data for a specific purpose. This collection, however, brings more responsibility on the data collector. Further, users have a fear of privacy while reporting the feedback to a trusted centralized system. The leak

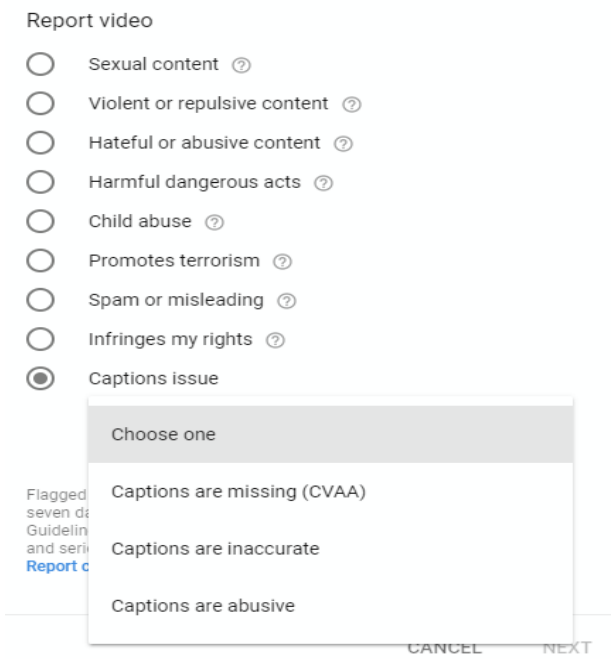


Fig. 1. Youtube Video Reporting System

of private data to malicious parties would not only bring a bad reputation for the providers but also cause a huge monetary fine as proposed in the European GDPR (General Data Protection Regulation) [21]. The addition of noise to the user data by adopting differential privacy can protect privacy to some extent, but it will affect the aggregation accuracy. The system we propose in this paper has two major characteristics. 1) the collection of data does not require any trusted system for collection and management of data, and 2) no noise is added to the data so the system provides an accurate aggregation results with the guaranteed property of privacy.

This paper presents a crowdsource based system called "PRIVCS (**PRIV**ate **C**rowd**S**ourcing)" that enables content providers to compute the weighted ratings of the content creators and the posted content by crowdsourcing the rating task to users of their network. The most important feature of the proposed approach is that it performs all operations in a decentralized way while also preserving the privacy of the participants and content creators. The crowdsource users submit their feedback rating of the query about the particular content or content creator in a secure encrypted form, which is then used to compute the aggregated trust score of the content creator or content, respectively. The entities in the system, either malicious or honest participants, would not be able to infer the submitted feedback of users and the number of users in the crowdsource group. The system also utilizes weights of the users in the crowdsource group based on their previous behavior and these weights remain hidden from the participants. The proposed system also does not allow malicious users to disrupt the functionality of computation by providing scores outside the prescribed range. This is achieved through the use of non-interactive zero-knowledge proof (ZK-Proof). The performance of the system is evaluated by performing the computation over the real social network datasets. This work is different from [22] in the following aspects: 1) it provides a mechanism for computing the aggregate trust from rating feedback; 2) it provides a mechanism for updating aggregate trust over time, and 3) it provides a comprehensive evaluation over the real online datasets. In summary, the paper makes the following major contributions:

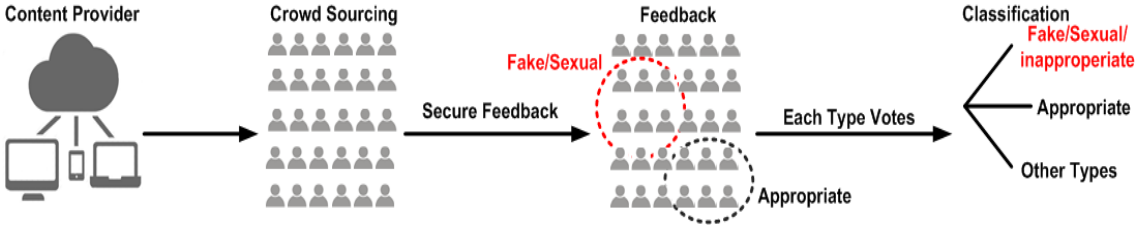


Fig. 2. Ecosystem of Crowdsourcing on Social Networks.

- We design a novel privacy-preserving crowdsourcing system that enables social networks and content providers to assess the trustworthiness of the content and content creators by crowdsourcing its users in a privacy-preserving way. To this extent, the system uses a homomorphic cryptosystem in a decentralized way, so it does not require a trusted third party. The members of the crowdsource have different weights in their feedback, and these weights and feedback remain unlikable during the entire computation process.
- We analyze the privacy and security of the system from the perspective of malicious and honest participants.
- A prototype is implemented, and the performance of the system is evaluated based on using real social network data.

The rest of the paper is organized as follows. In Section 2, we define crowdsourcing and how it can be used for privacy-preserving ratings. Section 3 provides discussion on the related work. Section 4 provides an overview of the proposed system followed by the discussion on the protocol operations in Section 5. Section 6 analyzes the security and privacy properties of the system. Section 7 analyses performance of the system. Section 8 concludes the paper.

2 BACKGROUND

In this section, we first define the mechanism of crowdsourcing and then present our problem statement.

2.1 Crowdsourcing

Crowdsourcing is the process that allows a selected set of users to provide their opinions for characterizing the behavior of objects, actors or other users in the network. Crowdsourcing normally makes use of human intelligence about a certain task whereas computers or machines are not good at providing a meaningful opinion. The ecosystem of crowdsourcing for computing reputation of content is shown in Figure 2. The system consists of the following main components: 1) the registered users who provide their opinions on the questions asked by the crowdsourcer, 2) the crowdsourcing platform which provides a platform for conducting the survey, 3) the content provider which provides content and requires feedback from their users to build their analytic, and 4) a response collector that can provide aggregate analytic to content provider and is owned by the crowd-sourcing platform. In Figure 2, the content provider provides a set of questions to the crowdsourcing platform which in turn distributes the questions to the registered users. The registered users who wish to respond present their feedback to the feedback or response collector. The response collector finally performs two operations, aggregation of scores and classification of content based on the aggregated feedback. Finally, the results are sent back to the content provider which further blocks or allows content on its network.

There are several crowdsourcing platforms (for example, Amazon Mechanical MTurk1 , Crowd-Flower2, and Witmart3) available that provide an opportunity to collect opinions from the users. Crowdsourcing has also seen applications in securing the network from the malicious actors, e.g.

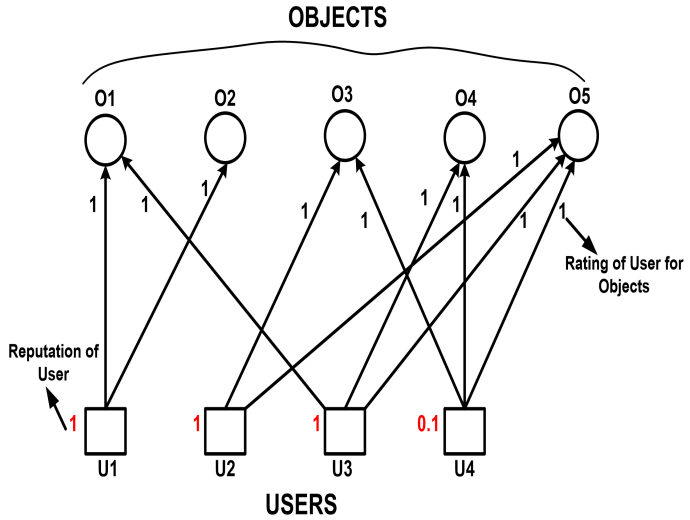


Fig. 3. Example Network of Weighted Reputation System.

securing the network from the unwanted communication e.g. email spam [11, 12] and unwanted calls [23] – where the user provides his opinion on questions from the service providers. The rapid increase of inappropriate content (fake news, sexual, hate content, etc.) over the social networks (Facebook, Twitter, Youtube, etc.) has highlighted the importance that human intelligence and machine learning should be used together for blocking inappropriate content. The challenge in crowdsourcing is to ensure the privacy of participants so that a large group of users agrees to participate in answering crowdsource queries.

2.2 Problem Definition

Normally, the content providers have a large user base but only a small percentage of users are content providers. Similarly, a very small percentage of users normally provide feedback about certain content and content creators. The challenge in the design of the crowdsourcing system is two-fold: 1) privacy-preservation and 2) decentralization. With the privacy-preservation crowdsourcing, we mean that the feedback values or ratings of the crowdsourced users are not known to either crowdsourced platforms or the content provider. Furthermore, the entities in the system would not be able to infer the information of another entity. With the decentralization, we mean no single entity is responsible for holding the data. We need to have a system that relies on a subset of their users for assessing the trustworthiness of subjects under observation i.e. a content creator or the content itself. Assume that out of N registered users, the content provider selects a subset of users in the crowdsource group i.e. U. The content providers asked these crowdsource users to provide their feedback ratings about the O objects (content creator or content). Let the rating value submitted by the member of the crowdsource group is S and $S \in 0, 1$. We assume that the crowdsource members submit these scores through a WebClient or special mobile application. In our case, the user i ($i \in 1 \ldots, u$) would like to provide rating for objects j ($j \in 1 \ldots, m$), where m is the number of content providers or contents. This rating interaction between users and objects is represented as the weighted bipartite graph G = (U;O;S;W). Here, U is a group of users included in the crowdsourced group, O is an object for which content provider wants to assess trustworthiness and are rated by the user, S is the trust rating designated by the user i for the particular object O_j under observation, and W is the trust weight of the user in the crowdsource group.

An example of a weighted user-objects network is shown in Figure 3. There are four users U = u1; u2; u3; u4 selected from the crowdsource group and O = o1; o2; o3; o4; o5 are the five objects for which content providers wants to access trustworthiness. In example, U_1 rates objects O_1 and O_2 with the trust score of 1, and U_4 rates objects O_3, O_4, and O_5 with the score of 1. Given such a scenario, if we take rating sum as the reputation of an object then all the objects are considered as appropriate because all the objects have a rating greater than 1. However, such a ranking system does not account user's trust weight into consideration. When the trust weights of the users are considered then the rating of an object O_5 is reduced to 0.666, whereas all other objects have a rating of 1. The procedure used to access the trustworthiness of objects must include the trust weights of users included in the crowdsource group. This is because of the fact that different users have a different level of trust within the social networks. In this paper, we consider these phenomena of weighted trust aggregation. To this extent, the content providers assign different trust weights to members in the crowdsource group.

We define the problem of assessing trustworthiness as following. Let there be a content provider (CP) who wish to assess the trustworthiness of its content and content creator by asking its users included in the crowdsource group. Let CP has selected users $U_1, U_2, \ldots, U_n$ for the crowdsource group. The user rates the object O_i under the observation over the scale of $s_i \in \{0, 1\}, \forall i \in [1, n]$. The returned feedback scores are then aggregated as the weighted average sum. The weights and individual ratings of crowdsourcing remain hidden throughout the computation process.

To protect the private information of users in the crowdsourced group, we propose a privacy-preserving trust assessment method based on the homomorphic cryptosystem, which allows the content provider to assess the trustworthiness of the content provider or content by asking crowd users for their opinion. In this way, the private information of users remains private to themselves yet participating in estimating the trustworthiness of objects (content or content provider).

3 RELATED WORKS

Several approaches have been proposed to guarantee the privacy of users while computing aggregate statistics over their shared values. Yang et.al [24] identify many security and privacy challenges that are essential for the design of a privacy-preserving crowdsourcing system. Rashidi et al. [25] proposed a DroidNet, a framework that assists mobile users to have feedback from other users about privacy-related permissions of applications. The objective is to identify malicious apps. However, the DroidNet framework itself can easily learn about the user's apps usage. Jin and Zhang [26, 27] proposed a novel framework to select spectrum-sensing participants in a privacy-preserving way. The framework is based on the semantics of differential privacy [28] and ensures the privacy of location privacy and truthfulness. Zhang et al. [29] also adopted differential privacy under the non-trusted server setup to ensure the privacy of participants in the crowdsourcing system. However, adding noise to data where the accurate result is necessary is not a desirable choice.

Erlingsson et al. [17] presented RAPPOR (Privacy-Preserving Aggregatable Randomized Response) for collecting statistics from clients while providing strong semantics of privacy-preservation using randomize response generation. RAPPOR collects a user's feedback or values about the set of strings using Bloom filters [30] with strong differential privacy guarantees. Polat et al. [31] proposed a collaborative filtering solution that randomized the user's responses using Randomized Perturbation techniques with the inclusion of the noise. Erkin et al. [32] proposed the system for generating the recommendation by encrypting the user's responses (rating for certain products or objects) in the homomorphism-based cryptographic system. Azad et al. [33] proposed a collaborative system that considers the encrypted feedback and weights of providers for computing the reputation of users in the respective content provider. However, the system is not completely decentralized as it depends on the trusted setup for the protecting assigned weights of raters. Wang

et al. [34] proposed a distributed agent-based privacy-preserving framework, called DADP, which consists of multiple agents that handle the user responses before relaying them to the untrusted server. In [35] a decentralized system is proposed for enabling users to participate in providing the feedback for different applications. Gibbs and Boneh [15] proposed a Prio system that consists of clients who hold the private data value and a small set of servers for computing the statistical function over the values reported by the clients.

The privacy of the client is purely dependent on the honesty of the servers. Primault et al. [36] proposed a Private Data Donor (PDD) platform for aggregating the web query results in a decentralized and privacy-preserving way. Bonawitz et al. [37] proposed a scheme for aggregating the values represented as a vector. The scheme ensures the privacy and security of participants under the honest-but-curious and malicious adversaries. Halevi et al. [38] proposed an aggregation scheme based on the homomorphic cryptosystem that evaluates the mathematical function securely and privately. However, the scheme requires PKI. Miao et al. [39] proposed a framework that performs a weighted aggregation over the user's encrypted data. The framework employs a homomorphic cryptosystem that has high accuracy in aggregation as well as protects the privacy of users. However, weights in this scheme are sent directly to participants. Luca et al. [40] proposed efficient cryptographic methods for the private aggregation of the large data stream. The data aggregation is performed in a privacy-preserving way using data sketches, instead of the raw data inputs. Dongxiao et al. [41] proposed an anonymous reputation system for the retail market that ensures privacy of consumers by using blockchain technology. The system protects the real identity of the user and his review using the anonymization approach; however, the private information of users can be deanonymized using some background information e.g. the buying history of the users. Rupeng et al. [42] proposed a blockchain-based decentralized anonymous credential system that exchanges the list of users blacklisted by the particular user in a privacy-preserving way. The system utilizes the tally like system for the sharing of the blacklist. Wang and Singh [43] proposed a trust and reputation model for the multiagent systems that use how agents in the system would produce the trust score from the evidence of their direct interactions. The system does not provide any discussion on how the privacy of participating agents is protected. In our work, we estimated the trustworthiness of the nodes (objects, content creators) while also protecting the privacy of the participant's feedback.

A privacy-preserving solution is proposed for the spatial crowdsourcing [44]. The scheme ensures privacy in two aspects: firstly, protection on the location of users in the crowdsource group, and secondly, the content of tasks is protected against the server and other users in the crowdsource group. To protect the location privacy the authors divide the location into grids and encrypt the grids as the code. For this purpose, the authors use attribute-based encryption and symmetric-key encryption. Wu et al.[45] proposed a data aggregation scheme using the bilinear pairing and homomorphic encryption. However, the scheme requires a third party system i.e. a Fog computing server to ensure the privacy of workers in the network. Fredrikson et al. [46] proposed to protect the privacy of patients and analyze the risk to the health of the patient using differential privacy with different privacy budgets. Kim et al. [47] presented an effort to address the challenge of protecting the privacy of health data streams emerging from smart devices. However, the aggregator or collector is a central component usually hosted by the healthcare service provider. Yifeng et al. [48] proposed crowdsensing methods that utilize the design choice of trust discovery. The design has inherent properties of privacy-protection of participants and also have reasonable improved bandwidth and computation requirements for the participating users. However, the proposed systems require the trusted server for the handling of data and computation of results.

The existing research considers mostly the honest-but-curious adversarial models but this model can be easily bypassed by the malicious participants to disrupt the operations of the protocol.

Furthermore, the existing systems either rely on trusted systems or the anonymization techniques to ensure the privacy of participants. Using the centralized system is not the realistic approach and anonymized data can be de-anonymized using some background information of participants. In this paper, we present a decentralized system without relying on any trusted system and trusted setup for the cryptographic operations and management of collected data. Further, the tally server does not need to be a trusted authority, thus the system provides correct operations even in the presence of malicious raters or participants.

4 SYSTEM FRAMEWORK AND PRELIMINARIES

In this section, we outline the system design of the proposed approach and define protocol preliminaries and assumptions.

4.1 System Design

The system design of PRIVCS is presented in Figure 4. The proposed crowdsourcing system has two major parts: firstly, an initialization phase in which a content provider initiates the query for assessing the trustworthiness of objects for specific features, and secondly a rating submission phase from the crowdsource group. In the user-rating phase, the user encrypts the rating score and submits it to the tally system along with information proving the correctness of feedback. Besides crowdsource group and content providers, the system also has the database system that holds cryptograms of every user's ratings, their ZK proofs, and the essential cryptographic parameters. The database is accessible to all participants but only users in the crowdsource group can write to this database. We assume that all users in the crowdsource group must respond to query within the designated time window. We consider a system setup consisting of a single centralized content provider having users with two attributes: 1) users who can create the content and view the content created by other users e.g. Youtube setup where users can create and post content as well as watch the content of other users, 2) users who only view the content. Here, we are only concerned with knowing the reputation of users who create content. We assume the content provider has selected users in the crowdsource group. The content provider can have this group by including users randomly or including users who are well-reputed. The users in the crowdsource group can provide their initial system settings (cryptographic parameters) to the tally system or the database owned by the content provider.

4.2 Threat Model

In this model, our objectives are three-fold. 1) We wish to assess the trustworthiness of content creators or content itself while preventing malicious clients, content provider and participants from learning feedback values of clients. 2) The second objective is to rate the content without learning who rated it and how. 3) The third and most important objective is to compute the trustworthiness while excluding the non-formed feedback from the final tally process. We achieve the following two properties for the honest-but-curious and the malicious participants:

Integrity of Scores A content provider cannot infer the feedback rating submitted by the users in the crowdsource group, however, it can assess the trustworthiness of users and content as an aggregate. Similarly, other users in the crowdsource group would not be to infer the trust weight of the users and their submitted feedback.

Well-Formedness of Feedback The users in the crowdsource group can behave in two threat models: semi-honest model, where users follow the protocol specification but try to learn private information of others, and the malicious model, where users have intentions of disrupting the

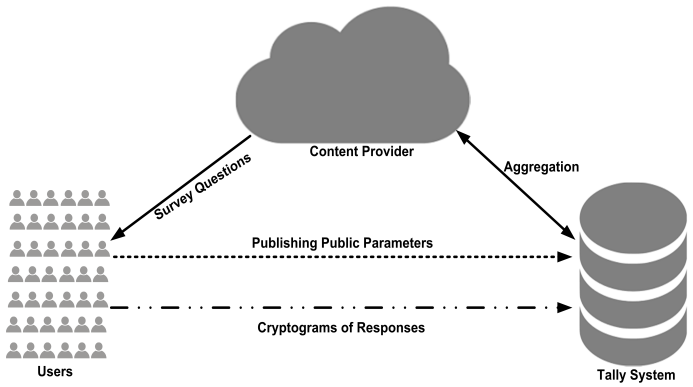


Fig. 4. Building Block of PRIVCS System.

Notations	Description
G	a cyclic Group in which DDH problem is hard
$U_1, U_2, \ldots, U_u$	Set of Users registered in a Crowdsourcing
(x_{1i}, x_{2i})	Secret key of U_i
$(g^{y_{1i}}, g^{y_{2i}})$	restructured key of U_i
$(g^{x_{1i}}, g^{x_{2i}})$	public key of U_i
s_i	feedback commitment of crowdsourced userU_i in ecnrypted form
w_i	trust weight of $U_i, 1 \leq w_i \leq a$
α_i	value generated user U_i for encrypted feedback

Table 1. Notations used the in PRIVCS System.

protocol operations by providing malformed feedback. The privacy under attack from the semi-honest users is protected using encrypted users whereas under the malicious model the correctness is achieved through the use of efficient ZK proof.

4.3 Cryptographic Approach

In our proposed privacy-preserving system, we adopte a homomorphic cryptosystem to assess the trustworthiness. The homomorphic system allows parties to compute the mathematical function in such a way that parties would not be able to see the input values of others yet can have the result of a particular mathematical function. Generally, the operations of the homomorphic cryptosystem consist of three steps: generation of public pk and secret key sk, the encryption phase that uses the public key to encrypt the plaintext, and the decryption phase that uses the sk to decrypt the final result. In our settings, we use the additive homomorphic cryptosystem that satisfies the following equations.

$$E_{pk}(m_1 + m_2) = E_{pk}(m_1) \oplus E_{pk}(m_2) \tag{1}$$

$$E_{pk}(W \cdot m_1) = W \otimes E_{pk}(m_1) \tag{2}$$

where m_1, m_2 are the feedback ratings (plaintext) that need to be encrypted and W is a weight.

There are many additive homomorphic cryptographic approaches, we adopt the feedback randomization scheme proposed by Hao et al. [49, 50] in assessing the trust of entities in the ecosystem. The randomization allows performing the secure summation on the encrypted rating scores in the decentralized settings without involving any trusted third party. Let there be a set of users denoted as $U = \{1, 2, \ldots, n\}$ selected by the content provider in the crowdsource group. These users present their encrypted commitments about the behavior of objects on the scale 0 and 1. Let there are big

primes q and p in the multiplicate cyclic group such that $q \mid p-1$. Let g be a generator which is in subgroup $\mathbb{Z}_q^*$ of order q. To present the commitment of feedback, the user in the crowdsource group first computes the preliminary constants used for encrypting the feedback values. In this step, the user first computes the random public key ($sk \in \mathbb{Z}_q$) and the private key (pk). The user keeps the sk to himself and distributes pk to other users via the decentralized tally system. The public key (pk) is generated from a value of sk as follows.

$$pk_i = g^{sk_i} \tag{3}$$

Once all the crowdsourced users have sent their pk to the tally system, the user in the crowdsource group then computes his own restructured key. This key is specifically used for generating the cryptogram of ratings and ZK proof. This key (Y_i) is computed as follows.

$$Y_i = \prod_{j \in N, j<i} pk_j \Big/ \prod_{j \in N, j>i} pk_j \tag{4}$$

The equation 4 ensures the following property.

$$\prod_{i \in N} Y_i^{sk_i} = 1. \tag{5}$$

Equation 4 is fundamental in our randomization approach and allows computation of aggregated average trust value of objects without employing a specialized system for managing and distributing crypto parameters.

4.4 Assumptions

We consider some of the properties that are relevant to use in cryptography. We assume the multiplicative group modulo p with the set of elements of big primes p and q under the group operation multiplication modulo p. The primes p and q satisfy $q \mid p-1$. We assume the following additional assumptions in our design.

Assumption 1. ***Decisional Diffie-Hellman assumption (DDH):*** *The DDH assumptions is based on the assumption that a discrete logorithmic in cycle group is hard to solve computationally [51]. The DDH problem in G, is to distinguish the distributions (g, g^a, g^b) and (g^a, g^b, g^c) We assume that the following two probability distributions are computationally indistinguishable. Given g, g^a, g^b and a challenge $\Omega \in \{g^{ab}, R\}$, where $R \xleftarrow{\$} G$, it is computationally hard to find whether $\Omega = g^{ab}$ or $\Omega = R$. where a and b are randomly and independently chosen from Z_q.*

Assumption 2. *Given g, g^a, g^b and a challenge $\Omega \in \{g^{ab}, g^{ab}g^a\}$, it is computationally in-feasible to compute whether $\Omega = g^{ab}$ or $\Omega = g^{ab}g^a$.*

Assumption 3. ***DDH assumption:*** *Given $g, g^a, g^b \in G, t \in \mathbb{Z}_p$ and a challenge $\Omega \in \{g^{ab}, g^{ab}g^t\}$, it is computationally infeasible to find whether $\Omega = g^{ab}$ or $\Omega = g^{ab}g^t$.*

Assumption 4. ***Tally System:*** *We assume that the content provider has deployed a tally system, the address of which is revealed to its users. The tally system is accessible to all users for reading the data from the tally system, however, write access is only provided to users in the crowdsource group. We assume that content providers could have multiple tally systems and each tally system is only handling a limited number of users in the crowdsource group.*

Assumption 5. ***Weight Assignment:*** *We assume that the content provider has inherent mechanism to assign trust weights to its crowdsource group.*

Assumption 6. ***Feedback Submission:*** *We assume that the users in the crowdsource group must provide their feedback once they agreed to provide feedback in the crowdsource selection process. The system operation is disrupted if any or some crowdsource users do not provide feedback ratings after publishing their initial system parameters. This limitation can be overcome by utilizing the approach mentioned in [40].*

5 PRIVCS THE FINAL PROTOCOL

The system operates in two steps. In the first step, the members of the crowdsource group are selected, and in the second step, these members provide their opinion about the objects. In the member selection step, the content provider determines the crowdsource group by choosing the subset of users registered in its systems. In the voting phase, the voter provides feedback on the credibility of the content presented to it.

5.1 Selecting members of Crowdsource Group

In this phase, the content provider (CP) selects the member of the crowdsource group from whom CP is seeking their opinion. The CP first assigns a unique identity to each user and then selects a crowdsource group from all registered users. This selection can be random or manual. The identity of the user is the same as the identity of the user that he has chosen while registering with the content provider. The CP can select the crowdsource group based on the content and interests of the user. We assume that the CP is honest in choosing the group because it is necessary to his business model and is deliberately not colluding with the members of crowdsourcing to maliciously increase or decrease the trust of certain content creators or objects. The CP can also select members from registered users of the network and the professionals hired by the CP for the specific tasks. The reason behind using the sample set for crowdsourcing is that over the social network content is normally seen by a large number of people but is liked or disliked by only a fraction of people [52]. The voting process can be enhanced further by using information automatically, for example, defining the duration of view or spending time on the post. In this case, a positive vote is considered if the user has seen a particular video or post for the fixed time and consider as negative vote alternatively.

For the protocol operations and privacy preservation, The content provider CP randomly generates two integers $\omega_1, \omega_2 \in \mathbb{Z}_p$ and calculates two variables $\sigma_1 = g^{\omega_1}$ and $\sigma_2 = g^{\omega_2}$. The content provider distributes these values on the decentralized tally system. The CP also publishes ZK-proof as $PW[\omega_1 : \sigma_1]$ and $PW[\omega_2 : \sigma_2]$ at the tally system. The ZK-proof $PW[\omega_i : \sigma_i]$ ensures that values of ω_i, such that $\sigma_i = g^{\omega_i}$, for all $i \in \{1, 2\}$ is indeed generated by CP and is known to CP to him. Finally, the CP creates the ranking question and advertise it to members of the crowdsource group.

The members of the crowdsource group $U_i; i \in [1, n]$ also creates two random integers $a_{1i}, b_{1i} \in \mathbb{Z}_p$. Using these integers the members then computes variables $\theta_{1i} = g^{a_{1i}}$ and $\delta_{1i} = g^{b_{1i}}$. The member U_i then distributes θ_{1i} and δ_{1i} and its associate ZK-proof (knowledge of θ_{1i} and δ_{1i}. Finally, the CP generates another variables $\theta_{2i} = (g^{w_i}/(\theta_{1i})^{\omega_1})^{1/\omega_2}$ and $\delta_{2i} = 1/(\delta_{1i}^{\omega_1})^{1/\omega_2}$ for all $i \in [1, n]$ along with their ZK-proof ($PW[\theta_{1i}, \theta_{2i}, \sigma_1, \sigma_2, g]$ and $PW[\delta_{1i}, \delta_{2i}, \sigma_1, \sigma_2, g]$ for all $i \in \{1, 2, \ldots, n\}$). The proof of first variable reveals the statement that $\theta_{1i}^{\omega_1}\theta_{2i}^{\omega_2} \in \{1, g, g^2, \ldots, g^a\}$ without disclosing ω_1 or ω_2. The second ZK proof shows that the following statement holds $\theta_{1i}^{\omega_1}\theta_{2i}^{\omega_2} = 1$. The systematization of these ZK-proof can be found in [22]. The CP places these variables and proofs $\theta_{2i}, \delta_{2i}, PW[\theta_{1i}, \theta_{2i}, \sigma_1, \sigma_2, g]$ and $PW[\delta_{1i}, \delta_{2i}, \sigma_1, \sigma_2, g]$ on the tally system. These calculations are done for each member of the crowdsource group i.e. for all $i \in \{1, 2, \ldots, U\}$.

5.2 Voting Phase

This is the main phase. It collects the opinions from crowdsource group. In this phase, the members of crowdsource group are responsible for two major operations. The generation of credentials i.e. the public and secret keys, and secondly the computation of cryptogram of feedback values. The member of crowdsource group $U_i, i \in [1, n]$, generates a big prime $x_{1i}, x_{2i} \in \mathbb{Z}_p$ for the secret key and computes the public key $pk_i = (X_{1i}, X_{2i}) = (g^{x_{1i}}, g^{x_{2i}})$. The member keeps sk to itself and distributes pk to other members of crowdsource group via the tally system. The member U_i then creates a ZK-proof for the x_{1i} and x_{2i}. These ZK-proofs are denoted as $PW_i[x_{ji} : X_{ji}]$ for $j = 1, 2$. The members of the crowdsource group finally computes the restructured key (Y_{1i}, Y_{2i}) as following:

$$Y_{ji} = g^{y_{ji}} = g^{\sum_{k=1}^{i-1} x^{jk} - \sum_{k=i+1}^{n} x^{jk}} = \frac{\prod_{k=1}^{i-1} g^{x_{jk}}}{\prod_{k=i+1}^{n} g^{x_{jk}}}, \forall j = 1, 2 \tag{6}$$

The member $U_i, i \in [1, n]$ finally chose a random value $\alpha_i \in \mathbb{Z}_p$ for computing the cryptogram of his feedback value $\mathbf{c}_i = (B_{1i}, B_{2i}, A_i)$ as following:

$$B_{1i} = Y_{1i}^{x_{1i}} (\theta_{1i})^{s_i} (\delta_{1i})^{\alpha_i} \tag{7}$$

$$A_i = g^{\alpha_i} \tag{8}$$

$$B_{2i} = Y_{2i}^{x_{2i}} (\theta_{2i})^{s_i} (\delta_{2i})^{\alpha_i} \tag{9}$$

In equations, $s_i \in \{0, 1\}$ is the value of the feedback a member U_i gives to the particular object. Along with the cryptogram of feedback, the member $U_i, i \in [1, n]$ also systemize ZK-proof

$$PW_i[B_{1i}, B_{2i} : X_{1i}, X_{2i}, Y_{1i}, Y_{2i}, \theta_{1i}, \theta_{2i}, \delta_{1i}, \delta_{2i}, A_i]$$

. This proof is essential in our design as it excludes malicious members from the computation process. The encrypted feedback values and associate ZK-proof are then published on the tally system

Each member of crowdsource $U_i, i \in [1, n]$ who is providing encrypted feedback, systemize ZK-proof

$$PW_i[B_{1i}, B_{2i} : X_{1i}, X_{2i}, Y_{1i}, Y_{2i}, \theta_{1i}, \theta_{2i}, \delta_{1i}, \delta_{2i}, A_i]$$

The ZK-proof establishes the truth that the B_{ji} for $j = 1, 2$ given $X_{ji} = g^{x_{ji}}, Y_{ji} = g^{y_{ji}}, \theta_{ji}, \delta_{ji}, A_i = g^{\alpha_i}$ and $s_i \in \{0, 1\}$ is within the defined range of values. This ZK-proof prove that following statement σ holds: $\sigma \equiv ((B_{1i} = Y_{1i}^{x_{1i}} \delta_{1i}^{\alpha_i}) \wedge (B_{2i} = Y_{2i}^{x_{2i}} \delta_{2i}^{\alpha_i})) \vee ((B_{1i} = Y_{1i}^{x_{1i}} \theta_{1i} \delta_{1i}^{\alpha_i}) \wedge (B_{2i} = Y_{2i}^{x_{2i}} \theta_{2i} \delta_{2i}^{\alpha_i}))$. Here, the secret inputs of the user U_i are x_{1i}, x_{2i}, α_i, and the publicly known variables are $B_{1i}, B_{2i}, Y_{1i}, Y_{2i}, \theta_{1i}, \theta_{2i}, g^{\alpha_i}$. In our approach, these proofs are established the properties of correctness in the non-interactive way. To make non-interactive we utilize widely used Σ protocol by making it non-interactive using the Fiat-Shamir heuristic. The details how such proofs are made non-interactive are shown in the Appendix section.

5.3 Computing Final Trust

Once the feedback scores and ZK-proof are published on the tally system, the CP can then assess the trustworthiness of objects by utilizing encrypted information from the tally system in secure and privacy-preserving way. Using the published encrypted feedback values $C = (C_1, C_2)$, the CP assess trustworthiness of objects as following:

$$C_j = \prod_{i=1}^{n} B_{ji} \tag{10}$$

$$= \prod_{i=1}^{n} Y_{ji}^{x_{ji}} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i} \tag{11}$$

$$= \prod_{i=1}^{n} g^{x_{ji} y_{ji}} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i} \tag{12}$$

$$= g^{\sum_{i=1}^{n} x_{ji} y_{ji}} \prod_{i=1}^{n} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i} \tag{13}$$

We can see that $\sum_{i=1}^{n} x_{ji} y_{ji} = 0$. Thus, $C_j = \prod_{i=1}^{n} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i}$ for $j = 1, 2$. The CP then computes $L = C_1^{\omega_1} C_2^{\omega_2}$ and place the result over the tally system along with the its ZK proof $PW_L[L : C_1, C_2, \sigma_1, \sigma_2]$. This ZK proof proves value L shown on the tally is indeed $C_1^{\omega_1} C_2^{\omega_2}$ given C_1, C_2, σ_1 and σ_2. Note that $L = \prod_{i=1}^{n} (\theta_{1i}^{s_i} \delta_{1i}^{\alpha_i})^{\omega_1} (\theta_{2i}^{s_i} \delta_{2i}^{\alpha_i})^{\omega_2} = \prod_{i=1}^{n} (\theta_{1i}^{\omega_1} \theta_{2i}^{\omega_2})^{s_i} (\delta_{1i}^{\omega_1} \delta_{2i}^{\omega_2})^{\alpha_i}$. Now, $\theta_{1i}^{\omega_1} \theta_{2i}^{\omega_2} = g^{w_i}$ and $\delta_{1i}^{\omega_1} \delta_{2i}^{\omega_2} = 1$ for all $i \in [1, n]$. Hence, $L = \prod_{i=1}^{n} g^{w_i s_i} = g^{S}$. Finally, a brute force search is performed on L to get the sum of votes in the favour of objects (positive) and sum of votes against (negative) the object. The weighted aggregated positive R is then computed as following:

$$R = S / \sum_{i=1}^{n} w_i \tag{14}$$

Equation 14 represents the positive trust about the object. The negative trust value (NT) of the object can be easily computed by subtracting the R from number of users in crowdsource group i.e. ($N_T = U - R$). Let R and N_T represents the collective value of positive and negative trust score of the objects provided by the users in the crowdsource group, the final reputation (F_R) function of the object can be computed using beta reputation [53] model as:

$$F_R = \frac{R - N_T}{R + N_T + 2} \tag{15}$$

We can also compute the final trust as the average of positive and negative ratings. The behavior of the users within the social networks changes over time. Due to this reason, the old feedback scores of the users may not always reflect the actual trustworthiness of the objects and users in the crowdsource group. We need to assign some high weights to recent feedback and reputation values of objects than the old feedback value. The CP incorporates the time factor while computing the trustworthiness of the objects as following:

$$A_R = \beta * F_{Rt} + (1 - \beta) * F_{Rt-1} \tag{16}$$

Where β defines the importance of the voting cycle t, A_R is the aggregated trustworthiness of the object.F_R^t represents the reputation of object at the current time cycle t and F_R^{t-1} defines the reputation of an object at previous aggregation cycle $t - 1$. We assigned more weight to the current voting cycle than the previous voting cycle. The CP would also increase or decrease the weight of users in the crowdsource group.

5.4 Final Classification

Once the aggregated score of the content is computed, the CP then asks the experts for the manual analysis if the aggregate score of the certain content or content creator is less than the predefined threshold. The CP then also increases or decreases the overall credibility score of the content creators depending on the classification result. Additionally, the CP can also utilize other features for example duration of video seen, overall interaction over the video and post, demographic distribution along with the aggregate score to better classify the credibility of the content creator. The final trust of the content creator is then computed based on the policies of the content provider.

The content provider either warns the content creator or block content creator straight away based on his trust score. This decision can be either based on machine learning approaches or can use the fixed threshold below which content creators are blocked.

6 SECURITY ANALYSIS

In this section, we provide a discussion on how the proposed scheme ensures the security and privacy of members of the crowdsource group under the threat model mentioned earlier. Specifically, we discuss these properties in the presence of semi-honest, malicious members of the crowdsource group and malicious content providers. The semi-honest members follow protocol specification whereas the malicious members do not care about. On the other hand, malicious content providers have the motivation of learning the feedback scores or behavior of members towards certain content which might reveal private information of members e.g. social or emotional behavior content which reveals personality attributes. Additionally, we also consider the scenario where some members of the crowdsource group collaborate with others to learn the private information of other members of the group. Our objective is to compute the trustworthiness $S = \sum_{i=1}^{n} w_i s_i$ without affecting the privacy of members of the group.

6.1 Correctness Property

Here, we prove that the proposed scheme correctly compute the trustworthiness of objects even in the presence of malicious members in the crowdsource group. The members of the group provide their opinion about the object in the encrypted form to the tally system in the form of $(B_{1i}, B_{2i}, g^{\alpha_i})$, where $B_{ji} = Y_{ji}^{x_{ji}} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i} = g^{x_{ji} y_{ji}} \theta_{ji}^{s_i} \delta^{\alpha_i} : j = 1, 2$. The CP utilizes the posted encrypted feedback from the tally system and computes $C = (C_1, C_2)$. $C_j = \prod_{i=1}^{n} B_{ji} = g^{x_{ji} y_{ji}} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i} : j \in \{1, 2\}$. This implies that $\sum_{i=1}^{n} x_{ji} y_{ji} = 0$ for $j = 1, 2$ and result in a $C_j = \prod_{i=1}^{n} \theta_{ji}^{s_i} \delta_{ji}^{\alpha_i}$. As $\theta_{ji} = g^{a_{ji}}$ and $\delta_{ji} = g^{b_{ji}}$ for $j = 1, 2$. Therefore, $C_j = \prod_{i=1}^{n} g^{a_{ji} s_i} g^{b_{ji} \alpha_i}$ which results in a $L = C_1^{\omega_1} C_2^{\omega_2} = \prod_{i=1}^{n} g^{(\omega_1 a_{1i} + \omega_2 a_{2i}) s_i} * \prod_{i=1}^{n} g^{(\omega_1 b_{1i} + \omega_2 b_{2i}) \alpha_i}$. a_{2i} and b_{2i}s satisfy the following equations holds i.e. $\omega_1 a_{1i} + \omega_2 a_{2i} = w_i$ and $\omega_1 b_{1i} + \omega_2 b_{2i} = 0$. Hence the final trustworthiness of object is, $L = \prod_{i=1}^{n} g^{w_i s_i} = g^{\sum_{i=1}^{n} w_i s_i} = g^S$. Therefore, the trustworthiness is correctly computed from the encrypted feedback even in the presence of malicious members.

6.2 Integrity of Member's Trust Weight

The *CP* likes that the trust weight it assigns to the members of the crowdsource group should remain private to itself only and should not be linkable by using other background information from the tally system. It is also desirable that these weights should not be revealed even if the subset of members collaborate with each to infer trust weight of other target member. The trust weight is used to compute $g^{a_{2i}} = (g^{w_i} / g^{a_{1i} \omega_1})^{1/\omega_2}$. Here, we need to prove the statement that the computation of $g^{a_{2i}}$ would not reveal the trust weight (w_i. $g^{a_{2i} \omega_2} = (g^{w_i} / g^{a_{1i} \omega_1})$) of members in any condition. Assume that there is malicious member $\mathcal{A}$ who has ability to differentiate the following two statements $w_i = w$ and $w_i = w'$, where $w' > w$. We show that the malicious member $\mathcal{A}$ can use this to break the assumption 3. Let the DDH adversary has input value $t, g^{\omega_2}, g^{a_{2i}}$ and a challenge $\Omega \in \{g^{a_{2i} \omega_2}, g^{a_{2i} \omega_2} g^t\}$. The value of t is $(w' - w)$. Malicious member $\mathcal{A}$ then selects random $a_{1i} \in \mathbb{Z}_p$ and computes $g^{\omega_1} = (g^{w'} / \Omega)^{1/a_{1i}}$. Therefore, if $\Omega = g^{a_{2i} \omega_2}$, then $g^{a_{1i} \omega_1 + a_{2i} \omega_2} = g^{w'}$ satisfies. Otherwise, if $\Omega = g^{a_{2i} \omega_2} g^{w' - w}$, then $g^{a_{1i} \omega_1 + a_{2i} \omega_2} = g^{w}$ holds. Therefore, if $\mathcal{A}$ can differentiate these two cases then it would have ability to differentiate between possible values of Ω viz. $g^{a_{2i} \omega_2}$ and $g^{a_{2i} \omega_2} g^{w' - w}$. From this we can establish the truth that the weights assigned to the members of the group remain secret throughout the computation process.

6.3 Integrity of Member's Trust Scores

The members of the crowdsource group provide their trust score in the encrypted form. The plaintext feedback is encrypted by utilizing the encryption key created from the public keys of the members of the group. This encrypted feedback is available over the tally system and anyone either CP, its users in the crowd group and other users can see these encrypted values. However, individually these published encrypted scores would not reveal any meaningful information about the members of the group. These values can only be used in an aggregate way to assess the trustworthiness of objects. This holds even if some members of the crowdsource group collude with each other or the malicious CP collaborates with some members of the group.

6.4 Feedback Stuffing

Feedback stuffing or ballot stuffing is the method by which the users submit ratings more than the allowed limits. This would result in an unfair rating (positive or negative) towards the particular object. This is very challenging in many online reputation systems are normally controlled by imposing some cost on the number of submitted votes, for example in eBay, users are only allowed to submit their feedback after the transaction i.e. buying a product. However, in other online social networks like Facebook or YouTube, liking or disliking a particular content does not incur the cast. However, over these networks, the most recent feedback value is always considered as the final feedback of the users. In our proposed system, as the crowdsource group is managed by the CP, thus could have an inherent mechanism of only considering the last vote as the final vote from the users. In this way, users are not able to vote single content more than once.

6.5 Colluding

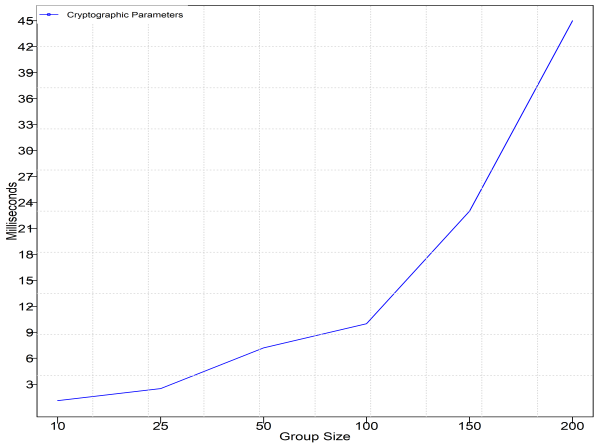
One of the major features of PRIVCS is that it protects the privacy of participants even if the number of participants colludes with each to compromise the privacy of the target participant. The colluding attack in this setup is only successful if all participants reveal their ratings to compromise or learn the ratings of the target. Another way where participants collaborate is to artificially increase or decrease the aggregate trust of content or content creator. This attack is only feasible if there exists a substantially large number of colluding participants exist in the crowdsource group. As this crowdsource group is normally generated by the content provider then might not be feasible for the group of attackers to circumvent the aggregation process.

7 PERFORMANCE ANALYSIS

In this section, we evaluate the computation and communication overhead over the synthetic and real datasets.

7.1 Experimental Implementation Real System Setup

The operation overheads of the designed system are evaluated by developing the prototype for two major components of the system i.e. the user agent– used for creating and providing the encrypted feedback, and the tally system used to aggregate the encrypted feedback scores provided by the users. We coded the prototype in Java. In the evaluation setup, we implemented the client and tally system as the Java application, however in the real setup the client can be implemented as the browser extension in Javascript to provide a real-time facility to users to provide the feedback. To deploy it in the real system scenario we can directly embed the functionality of the client within the HTML page. As soon as the client watched some content or have interaction with some profile, the content provider sends a query to the client to provide his feedback about the interaction. The



(a) Cryptography Parameters

Fig. 5. Computation Time Required for varying group size

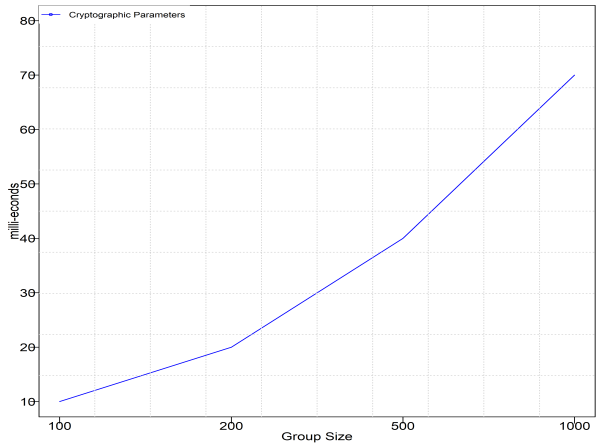
query to the client can be either sent periodically or at the end of the interaction and the client has to answer the query within the specified time.

For the cryptographic implementation, we utilized standard NIST Curve P-256 for 128-bit security. We analyzed the performance overhead over the system with i7-core (CPU 3.4Gh) with 8GB of RAM and Windows operating system. We simulated the feedback and aggregation phase for a single user, however, the number of users in the crowdsource group varies from 10 to 1000. The performance has been analyzed for the client-side and computation of aggregated scores. For the client-side, the performance has been analyzed for two metrics: 1) time required for generating cryptography parameters and 2) the time required for generating the cryptogram. At the tally side, the evaluation is performed for the time required to compute the whole tally and updating of trust scores.

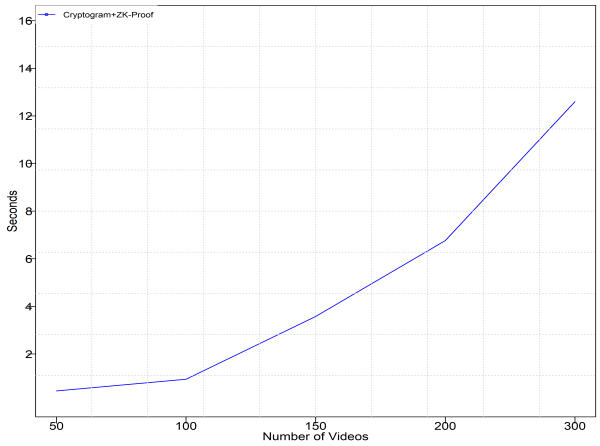
In real setup, the tally system can be a standalone third party system or system belonging to the content provider. In both cases, the data from the tally system could be used to infer the private information of the users. In content provider setup, the query is directly generated by the content provider whereas in case of a standalone third party system the content provider has to trust party for his query data and list of users provided to the trusted third party which is not a quite realistic choice. The design system operates in the decentralized system and can be easily implemented as the smart contract over the private or the public blockchain. In this setting, the participants are provided with the unique token that they used to post or submit their feedback to the blockchain. The transaction data from the block is then used by any party (participants or content provider) to estimate the aggregate reputation of content or users who created that content.

7.2 Performance Measures

We analyzed the time complexity using the setup YouTube is using to collect the report feedback about the video from the client. YouTube collects the user's feedback about the particular video for the 9 features. We simulated the same setup for collecting the feedback from a crowd member. The crowd member specifically performs two major operations, creating the cryptographic parameters (secret, public and restructured keys) and cryptogram of feedback (encrypted score and ZK-proof). The creation of the public and secret key is seamless and is done within a millisecond, however, the



(a) Cryptography Parameters

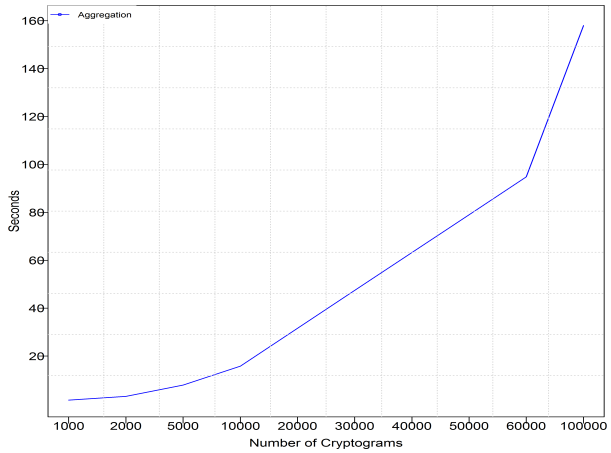


(b) Feedback Response

Fig. 6. Computation Time Required for Cryptography parameters and Encrypted feedback.

computation of restructured key depends on the number of users in the crowd-group. The larger the group size, the higher the time required for generating the restructured key. Figure 5 computation time by varying the number of users in the crowd-group. The group size is an important feature for providing privacy-preservation and correctness of protocol operations. The size of the group can vary from a few users to a few hundred users. The smaller size could have a threat to privacy but it is easy to handle and would provide efficient results. On the other hand, the larger group size could provide absolute privacy but at the cost of utility as it might be possible that a single group member may restrain to provide the feedback cryptogram after publishing his cryptographic parameters. Our approach ensures privacy-preservation even for the small group size as it requires at least K-1 (K is the size of group) members to collude with each other to learn information about the target user.

Figure 6.A shows that time for generating the cryptography parameters, which increases linearly with several users in the crowd-group, however, this time is not very much high and is acceptable.



(a) Computing Trustworthiness

Fig. 7. Computation Time Required for aggregating the Feedback

We prefer to have a small group size as it not only decreases the computation time but is also feasible.

Figure 6.B shows the time required by the group member to respond to the content providers for 9 different options. We varied the number of videos from 5 to 100 for which the group member submit the feedback. The time is not very high and can be decreased further used parallel computation over multiple cores.

Finally, Figure 7 shows the time CP consumes while computing the trustworthiness of video in all 9 dimensions. The CP computes the final score from the 100K cryptograms in around 8 seconds. This does not constitute the time required for verifying the correctness of feedback.

Dataset	# of User	# of Objects	Rating Scale
Epnions-1 (Product)	131,828	139,738	(1-5)
Epnions-2 (User Ratings)	49,290	49,290	(0,1)
Slashdot	82,168	82,168	(0,1)
Dating	194,4399	220,970	(0,1)
Jastor	135,359	150	(-10,10)
Digg	279,630	279,630	(0,1)

Table 2. Data set and their Associated Computation and Communication overhead

7.3 Evaluation on Real Dataset

We used five major datasets to evaluate the performance of our system. The dataset are downloaded from different sources. The networks we use have a rating scale of (0-1) and (1-5). The detail on each of the dataset that is being used for the evaluation is as follow.

Epinions Epinions.com is a web site where users can write reviews or provides ratings for the products (such as cars, books, movies, music, software, etc.). The rating range is 1(min) to 5 (mac). Users are also allowed to rate other users of the system who have provided the rating for the products, developing a network of a trusted group. We obtained two variants of the Epinions

dataset from the website[1], 1) user to product ratings, 2) users to users rating. The dimensions of the dataset are shown in Table 2.

Slashdot Slashdot is the technology news site where users can rate each other as a foe (0) or friend (1). We consider these ratings as an indicator of whether the user has shown trust in other fellow users is not. We obtained from Stanford [2] data repository and it consists of 82,168 users with around 948,464 links.

Dating Dataset The dataset we used consists of data from a real online dating service âĂŞ Libimseti. This data set was collected by Lukas et al. [54]. In the dataset, users rate the profile photograph of other users. Overall the dataset contains 194,439 users, who provided 11,767,448 ratings.

Digg Digg is a social news aggregator website that allows users to submit and manage the news stories. Digg allows users to rate other users as friends forming a trusted friendship network among users of the network. The dataset [3] we obtained consists of 279,630 users and 1,731,653 total votes.

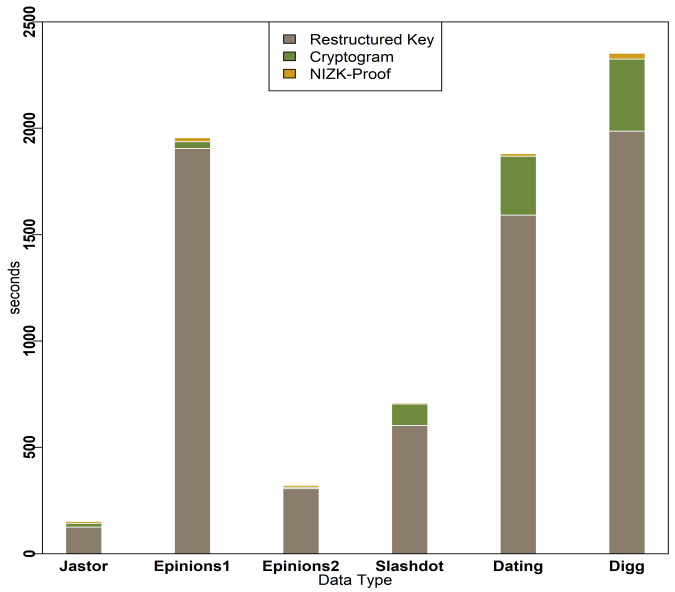


Fig. 8. Client's Computation overhead for the real dataset. The number of objects are fixed to 20% of total objects in the data, and the number of users are same as in the dataset.

We evaluated the system and take measurements on a 3.4 GHz Intel i-7 processor with 8GB memory on Windows 10 operating system. We used the following simulation methodology for the evaluation. First, we created the election query for the number of data points in the respective dataset, then we propagated the query to the nodes in the network; which returns the encrypted score to the tally system. In the evaluation, we used a similar rating scale which has been used in the respective dataset. 2 presents the details of the dataset we considered for the evaluation. From the dataset, we have observed that the dataset has a large sparsity that users normally voted only a few objects in the networks. In our system settings, this would create a loud on the client because

[1]Data-Source: http://www.trustlet.org/
[2]http://snap.stanford.edu/data/
[3]http://konect.uni-koblenz.de/networks/

of providing the zero cryptograms for a large number of nodes. Figure 8 presents the computation time required when clients in each dataset report feedbacks for 20% of objects. Though this figure is very high, normally on rating networks client only provides ratings to maximum few hundred objects. It can be observed that the more the number of options for the rating scale, the higher would be the computation cost because the user has to generate cryptograms, NIZK proof, and an additional NIZK proof to prove that he selected only one value from the available options.

7.4 Effect of Weights

In this section, we show how a participant's weight affects the aggregate trust of the object. For the first experiment, we varied weights of the participants from 0.1 to 1 whereas the number of participants in the group varies from 5 to 40. We consider the simple scenario where all participants voted a content with the same feedback score. Figure 9 represents the effect of weights on the aggregate trust of the object. The higher the weight of the participants the greater would be the effect on the aggregate trust. We can also implement the system by assigning different weights to the crowdsource groups. In this case, if we have a higher number of trusted groups then the aggregate trust of the object is more inclined towards the trust revealed by the trusted group. The small number of non-trusted groups in this setup would have a negligible effect on the aggregation process.

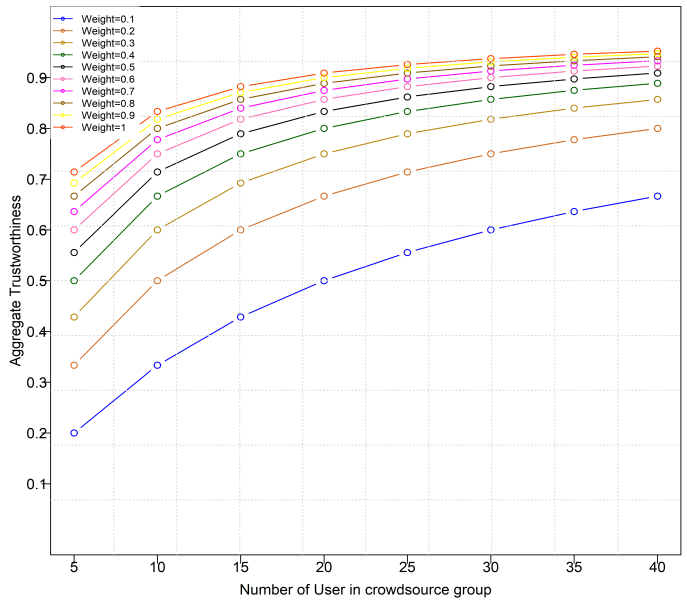


Fig. 9. Variation in Aggregated Trust for varying size of group and weights

8 CONCLUSION

The crowdsourcing process outsources the tasks to a group of human users to provide their intelligent feedback about certain issues. Existing crowdsourcing solutions do not give much importance to the privacy of users and are also dependent on the centralized system for aggregation and processing of feedback. Furthermore, these systems also give equal importance to its participants and do not consider participants having different trust weights while aggregating the feedback scores. In this paper, we have presented a decentralized crowdsource system that guarantees the

privacy of each participant's private feedback without using a trusted system or including any noisy data to the feedback. Furthermore, the system provides verification ability to both participants as well as the content. We have also presented the use case for object ranking over the social network i.e. video ranking over YouTube for blocking the unwanted content. The efficacy of the system has been demonstrated by providing a prototype implementation and performance measures are provided based on real data sets.

REFERENCES

[1] F. Fischer, K. BÃűttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl, "Stack overflow considered harmful? the impact of copy paste on android application security," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 121–136.

[2] "Google's bad week: Youtube loses millions as advertising row reaches us," 2018. [Online]. Available: https://www.theguardian.com/technology/2017/mar/25/google-youtube-advertising-extremist-content-att-verizon

[3] "Google loses more advertisers over hate videos," 2018. [Online]. Available: http://money.cnn.com/2017/06/07/technology/google-youtube-hate-videos-isis/index.html

[4] "Youtuber logan paul apologizes for filming suicide victim, says i did not do it for views," 2018. [Online]. Available: https://www.theverge.com/2018/1/2/16840176/logan-paul-suicide-video-apology-aokigahara-forest

[5] T. Fu, C. Huang, and H. Chen, "Identification of extremist videos in online video sharing sites," *2009 IEEE International Conference on Intelligence and Security Informatics*, pp. 179–181, 2009.

[6] N. Aggarwal, S. Agarwal, and A. Sureka, "Mining youtube metadata for detecting privacy invading harassment and misdemeanor videos," *2014 Twelfth Annual International Conference on Privacy, Security and Trust*, pp. 84–93, 2014.

[7] A. Sureka, P. Kumaraguru, A. Goyal, and S. Chhabra, "Mining youtube to discover extremist videos, users and hidden communities," in *Information Retrieval Technology*, P.-J. Cheng, M.-Y. Kan, W. Lam, and P. Nakov, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 13–24.

[8] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis, "A multimodal approach to violence detection in video sharing sites," in *2010 20th International Conference on Pattern Recognition*, Aug 2010, pp. 3244–3247.

[9] O. Deniz, I. Serrano, G. Bueno, and T. Kim, "Fast violence detection in video," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2, Jan 2014, pp. 478–485.

[10] S. Alghowinem, "A safer youtube kids: An extra layer of content filtering using automated multimodal analysis," in *Intelligent Systems and Applications*, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2019, pp. 294–308.

[11] M. Sirivianos, K. Kim, and X. Yang, "Socialfilter: Introducing social trust to collaborative spam mitigation," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 2300–2308.

[12] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "P2p-based collaborative spam detection and filtering," in *Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004. Proceedings.*, Aug 2004, pp. 176–183.

[13] (2018) Youtube publishes deleted videos report. [Online]. Available: http://www.bbc.co.uk/news/technology-43868633

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th International Conference on World Wide Web*, 2003, pp. 640–651.

[15] H. Corrigan-Gibbs and D. Boneh, "Prio: Private, robust, and scalable computation of aggregate statistics," in *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'17. Berkeley, CA, USA: USENIX Association, 2017, pp. 259–282. [Online]. Available: http://dl.acm.org/citation.cfm?id=3154630.3154652

[16] S. Hohenberger, S. Myers, R. Pass, and a. shelat, "Anonize: A large-scale anonymous survey system," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 375–389.

[17] U. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 1054–1067. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660348

[18] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, ser. STOC '10. New York, NY, USA: ACM, 2010, pp. 715–724. [Online]. Available: http://doi.acm.org/10.1145/1806689.1806787

[19] K. Ligett and A. Roth, "Take it or leave it: Running a survey when privacy comes at a cost," in *Proceedings of the 8th International Conference on Internet and Network Economics*, ser. WINE'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 378–391. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35311-6_28

[20] A. Roth and G. Schoenebeck, "Conducting truthful surveys, cheaply," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, ser. EC '12. New York, NY, USA: ACM, 2012, pp. 826–843. [Online]. Available: http://doi.acm.org/10.1145/2229012.2229076

[21] "Gdpr portal: Site overview," 2018. [Online]. Available: https://www.eugdpr.org/

[22] M. A. Azad, S. Bag, S. Parkinson, and F. Hao, "Trustvote: Privacy-preserving node ranking in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5878–5891, Aug 2019.
[23] Y. S. Wu, S. Bagchi, N. Singh, and R. Wita, "Spam detection in voice-over-ip calls through semi-supervised clustering," in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, June 2009, pp. 307–316.
[24] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 75–81, August 2015.
[25] B. Rashidi, C. Fung, A. Nguyen, T. Vu, and E. Bertino, "Android user privacy preserving through crowdsourcing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 773–787, March 2018.
[26] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.
[27] ——, "Privacy-preserving crowdsourced spectrum sensing," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2018.
[28] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.
[29] Z. Zhang, S. He, J. Chen, and J. Zhang, "Reap: An efficient incentive mechanism for reconciling aggregation accuracy and individual privacy in crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2995–3007, Dec 2018.
[30] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, 2004. [Online]. Available: https://doi.org/10.1080/15427951.2004.10129096
[31] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Third IEEE International Conference on Data Mining*, Nov 2003, pp. 625–628.
[32] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1053–1066, June 2012.
[33] M. Azad, S. Bag, S. Tabassum, and F. Hao, "privy: Privacy preserving collaboration across multiple service providers to combat telecoms spam," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2017.
[34] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, "Privacy-preserving crowd-sourced statistical data publishing with an untrusted server," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.
[35] L. Melis, G. Danezis, and E. D. Cristofaro, "Efficient private statistics with succinct sketches," *In NDSS, 2016.*
[36] V. Primault, V. Lampos, I. Cox, and E. De Cristofaro, "Privacy-preserving crowd-sourcing of web searches with private data donor," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: ACM, 2019, pp. 1487–1497. [Online]. Available: http://doi.acm.org/10.1145/3308558.3313474
[37] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 1175–1191. [Online]. Available: http://doi.acm.org/10.1145/3133956.3133982
[38] S. Halevi, Y. Lindell, and B. Pinkas, "Secure computation on the web: Computing without simultaneous interaction," in *Advances in Cryptology – CRYPTO 2011*, P. Rogaway, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 132–150.
[39] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Privacy-preserving truth discovery in crowd sensing systems," *ACM Trans. Sen. Netw.*, vol. 15, no. 1, pp. 9:1–9:32, Jan. 2019. [Online]. Available: http://doi.acm.org/10.1145/3277505
[40] L. Melis, G. Danezis, and E. D. Cristofaro, "Efficient private statistics with succinct sketches," *CoRR*, vol. abs/1508.06110, 2015. [Online]. Available: http://arxiv.org/abs/1508.06110
[41] D. Liu, A. Alahmadi, J. Ni, X. Lin, and X. Shen, "Anonymous reputation system for iiot-enabled retail marketing atop pos blockchain," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3527–3537, June 2019.
[42] R. Yang, M. H. Au, Q. Xu, and Z. Yu, "Decentralized blacklistable anonymous credentials with reputation," *Computers Security*, vol. 85, pp. 353 – 371, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404818313282
[43] Y. Wang and M. P. Singh, "Evidence-based trust: A mathematical model geared for multiagent systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 5, no. 4, Nov. 2010. [Online]. Available: https://doi.org/10.1145/1867713.1867715
[44] D. Yuan, Q. Li, G. Lia, Q. Wang, and K. Ren, "Priradar: A privacy-preserving framework for spatial crowdsourcing," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019.
[45] H. Wu, L. Wang, and G. Xue, "Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2019.
[46] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, ser. SEC'14. Berkeley, CA, USA: USENIX Association, 2014, pp. 17–32. [Online]. Available:

http://dl.acm.org/citation.cfm?id=2671225.2671227
[47] J. W. Kim, D. Kim, and B. Jang, "Application of local differential privacy to collection of indoor positioning data," *IEEE Access*, vol. 6, pp. 4276–4286, 2018.
[48] Y. Zheng, H. Duan, X. Yuan, and C. Wang, "Privacy-aware and efficient mobile crowdsensing with truth discovery," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2017.
[49] F. Hao, P. Y. A. Ryan, and P. Zielinski, "Anonymous voting by two-round public discussion," *IET Information Security*, vol. 4, no. 2, pp. 62–67, 2010.
[50] F. Hao, M. N. Kreeger, B. Randell, D. Clarke, S. F. Shahandashti, and P. H.-J. Lee, "Every vote counts: Ensuring integrity in large-scale electronic voting," in *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14)*. San Diego, CA: USENIX Association, 2014. [Online]. Available: https://www.usenix.org/conference/evtwote14/workshop-program/presentation/hao
[51] K. Kurosawa, R. Nojima, and L. T. Phong, "Efficiency-improved fully simulatable adaptive ot under the ddh assumption," in *Proceedings of the 7th International Conference on Security and Cryptography for Networks*, ser. SCN'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 172–181. [Online]. Available: http://dl.acm.org/citation.cfm?id=1885535.1885554
[52] A. Chakraborty, G. K. Patro, N. Ganguly, K. P. Gummadi, and P. Loiseau, "Equality of voice: Towards fair representation in crowdsourced top-k recommendations," 2018.
[53] B. E. Commerce, A. JÃÿsang, and R. Ismail, "The beta reputation system," in *In Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.
[54] L. Brozovsky and V. Petricek, "Recommender system for online dating service," in *Proceedings of Conference Znalosti 2007*. Ostrava: VSB, 2007. [Online]. Available: http://www.occamslab.com/petricek/papers/dating/brozovsky07recommender.pdf

APPENDIX

Proof: Well-formedness of Feedback $PW_i[B_{1i}, B_{2i} : X_{1i}, X_{2i}, Y_{1i}, Y_{2i}, \theta_{1i}, \theta_{2i}, \delta_{1i}, \delta_{2i}, A_i]$: This NIZK proof proves the well-formedness of B_{ji} for $j = 1, 2$ given $X_{ji} = g^{x_{ji}}, Y_{ji} = g^{y_{ji}}, \theta_{ji}, \delta_{ji}, A_i = g^{\alpha_i}$ and $s_i \in \{0, 1\}$. In other words, it proves that the following statement σ is correct.
$\sigma \equiv ((B_{1i} = Y_{1i}^{x_{1i}} \delta_{1i}^{\alpha_i}) \wedge (B_{2i} = Y_{2i}^{x_{2i}} \delta_{2i}^{\alpha_i})) \vee ((B_{1i} = Y_{1i}^{x_{1i}} \theta_{1i} \delta_{1i}^{\alpha_i}) \wedge (B_{2i} = Y_{2i}^{x_{2i}} \theta_{2i} \delta_{2i}^{\alpha_i}))$
Since, $Y_{ji} = g^{x_{ji}}, \theta_{ji} = g^{a_{ji}}, \delta_{ji} = g^{b_{ji}}$ for $j = 1, 2$, we can rewrite σ as below:
$\sigma \equiv ((B_{1i} = g^{x_{1i} y_{1i}} g^{b_{1i} \alpha_i}) \wedge (B_{2i} = g^{x_{2i} y_{2i}} g^{b_{2i} \alpha_i})) \vee ((B_{1i} = g^{x_{1i} y_{1i}} g^{a_{1i}} g^{b_{1i} \alpha_i}) \wedge (B_{2i} = g^{x_{2i} y_{2i}} g^{a_{2i}} g^{b_{2i} \alpha_i}))$
The above statement is a one-out-of-two NIZK statement.

Here, we show how the participants construct a NIZK proof for the above statement. The feedback response $B_i = \langle B_{1i}, B_{2i}, g^{\alpha} \rangle$ where either of the two statements holds:
1) $B_{1i} = g^{x_{1i} y_{1i}} g^{b_{1i} \alpha_i} \wedge B_{2i} = g^{x_{2i} y_{2i}} g^{b_{2i} \alpha_i}$
2) $B_{1i} = g^{x_{1i} y_{1i}} g^{b_{1i} \alpha_i} g^{a_{1i}} \wedge B_{2i} = g^{x_{2i} y_{2i}} g^{b_{2i} \alpha_i} g^{a_{2i}}$
That is the voter has to prove that either of the two statements stated above is true. For sake of clarity we write B_{ji} as B_j, x_{ji} as x_j, y_{ji} as y_j, b_{ji} as b_j, a_{ji} as a_j, $\forall j \in \{1, 2\}$ and α_i as α. We need to construct a proof for the statement;
$\sigma \equiv (B_1 = g^{x_1 y_1} g^{b_1 \alpha} \wedge B_2 = g^{x_2 y_2} g^{b_2 \alpha}) \vee (B_1 = g^{x_1 y_1} g^{b_1 \alpha} g^{a_1} \wedge B_2 = g^{x_2 y_2} g^{b_2 \alpha} g^{a_2})$.
The given inputs are these: $g^{x_1}, g^{y_1}, g^{x_2}, g^{y_2}, g^{\alpha}, g^{a_1}, g^{a_2}, g^{b_1}$ and g^{b_2}. Only one of the two statements above is true. Let us assume that the first statement is true, that is $(B_1 = g^{x_1 y_1} g^{b_1 \alpha} \wedge B_2 = g^{x_2 y_2} g^{b_2 \alpha})$. Hence, the prover will have to provide a real proof for the first statement and a simulated proof for the second statement. The prover selects random r_1, r_2 and computes 3 commitments $com_{11} = g^{r_1}, com_{12} = g^{r_2}, com_{13} = (g^{y_1})^{r_1} (g^{b_1})^{r_2}, com'_{11} = g^{r'_1}, com'_{12} = g^{r'_2}, com'_{13} = (g^{y_2})^{r'_1} (g^{b_2})^{r'_2}$. Then the prover selects random $ch_2, res_{21}, res_{22}, res'_{21}, res'_{22} \in_R \mathbb{Z}_p$ and computes these commitments:
$com_{21} = g^{res_{21}} (g^{x_1})^{ch_2}, com_{22} = g^{res_{22}} (g^{\alpha})^{ch_2}, com_{23} = (g^{y_1})^{res_{21}} (g^{b_1})^{res_{22}} (B_1 / g^{a_1})^{ch_2}$
and $com'_{21} = g^{res'_{21}} (g^{x_2})^{ch_2}, com'_{22} = g^{res'_{22}} (g^{\alpha})^{ch_2}, com'_{23} = (g^{y_2})^{res'_{21}} (g^{b_2})^{res'_{22}} (B_2 / g^{a_2})^{ch_2}$. Now let ch be the grand challenge of the NIZK proof, obtained by feeding all the above parameters into a hash function. Let, $ch_1 = ch - ch_2$. The prover computes $res_{11} = r_1 - x_1 * ch_1, res_{12} = r_2 - \alpha * ch_1, res'_{11} = r'_1 - x_2 * ch_1, res'_{12} = r'_2 - \alpha * ch_1$. The verification equations are as follows: $g^{res_{i1}} \stackrel{?}{=} \frac{com_{i1}}{(g^{x_1})^{ch_i}}, i = 1, 2$

$g^{res_{i2}} \stackrel{?}{=} \frac{com_{i2}}{(g^{\alpha})^{ch_i}}, i = 1, 2$

$g^{res'_{i1}} \stackrel{?}{=} \frac{com'_{i1}}{(g^{x_2})^{ch_i}}, i = 1, 2$

$g^{res'_{i2}} \stackrel{?}{=} \frac{com'_{i2}}{(g^{\alpha})^{ch_i}}, i = 1, 2$

$(g^{y_1})^{res_{11}}(g^{b_1})^{res_{12}} \stackrel{?}{=} \frac{com_{13}}{B_1^{ch_1}}$

$(g^{y_1})^{res_{21}}(g^{b_1})^{res_{22}} \stackrel{?}{=} \frac{com_{23}}{(B_1/g^{a_1})^{ch_2}}$

$(g^{y_2})^{res'_{11}}(g^{b_2})^{res'_{12}} \stackrel{?}{=} \frac{com'_{13}}{B_2^{ch_1}}$

$(g^{y_2})^{res'_{21}}(g^{b_2})^{res'_{22}} \stackrel{?}{=} \frac{com'_{23}}{(B_2/g^{a_2})^{ch_2}}$

If all the above 12 equations satisfy, the NIZK statement is true. Similarly, the prover can generate a NIZK proof statement if the second statement is true, that is: $(B_1 = g^{x_1 y_1} g^{b_1 \alpha} g^{a_1} \wedge B_2 = g^{x_2 y_2} g^{b_2 \alpha} g^{a_2})$. Here, we omit this due to space restriction.

The above NIZK proof requires 12 commitments, 8 responses, and two challenges. Hence, the space required to store them is equal to 22. Since there are n peers, the total size of all such proofs is $22n$. The total number of exponentiations required to compute the NIZK proof is 22. Since there are n peers, the total number of exponentiations required by all n peers to compute all the proofs is $22n$. Again, the verification of each such proof requires 28 exponentiations. Hence, in order to verify all n proofs, a public verifier needs to do $28n$ exponentiations in total.