Opinions of People: Factoring in Privacy and Trust

Anirban Basu[§], Jaideep Vaidya[†], Juan Camilo Corena[§], Shinsaku Kiyomoto[§],
Stephen Marsh[‡], Guibing Guo[¶], Jie Zhang[¶], Yutaka Miyake[§]

[§]KDDI R&D Laboratories, Inc., 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
{basu, corena, kiyomoto, miyake}@kddilabs.jp
[†]MSIS Department, Rutgers, 1, Washington Park, Newark, NJ 07102, USA
jsvaidya@rutgers.edu
[‡]Faculty of Business and IT, UOIT, 2000 Simcoe St N, Oshawa, ON L1H 7K4, Canada
stephen.marsh@uoit.ca
[¶]School of Computer Engineering, NTU, 50 Nanyang Avenue, Singapore 639798
{gguo1, zhangj}@ntu.edu.sg

ABSTRACT

The growth of online social networks has seen the utilisation of these network graphs for the purpose of providing recommendations. Automated recommendations, however, do not take into account inter-personal trust levels that exist in a social network. In this article, we propose a privacy-preserving trusted social feedback (TSF) scheme where users can obtain feedback on questions from their friends whom they trust. We show that the concept can be extended to the domain of crowdsourcing – the trusted crowdsourcing (TCS) scheme. In crowdsourcing, instead of asking friends, one can solicit opinions from experts in the crowd through a privacy preserving trusted feedback mechanism. Our proposal supports categorical answers as well as single-valued numerical answers. We evaluate our proposals in a number of ways: based on a prototype implementation built atop the Google App Engine, we illustrate the performance of the trusted social feedback. In addition, we present a user study to measure the impact that our trusted social feedback proposal has on users' perception of privacy and on foreground trust. We also present another user study to capture a model for user acceptance testing of the trusted crowdsourcing[1].

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Human Factors, Security

Keywords

privacy, trust, social network, crowd sourcing, recommendation, category

1. INTRODUCTION

Social aware recommendation – a recent phenomenon in recommender systems – has made use of online social networks to provide better, arguably more accurate, recommendations. However, many such automated recommender systems fail to consider the inter-personal context-sensitive trust that exists between individuals in the social network, which can affect the way recommendations are made and interpreted. We see social aware recommendation from a different angle, one that is modelled after the word-of-mouth concept in a human society. We also show that our work can be generalised and is applicable to crowdsourcing. We envisage that inter-personal trust is sensitive information, which the truster may wish to keep private; thus requiring the social aware recommendation mechanisms to be privacy-preserving.

We postulate that the strength of a social relation is often one's asymmetric personal perception of another in a particular context that changes over time. We refer to this as *trust* in this paper[2]. The asymmetric nature of personal perception means that a's trust in b is likely to be different from b's trust in a. This should affect the way one believes a recommendation from a friend. Recommendations based on a community of opinions do not generally consider this interpersonal and contextual trust. In tune with this understanding of trust and modelling after the real society, the user asks for the aggregate feedback from her friends, in her social network, regarding the item of interest. She attaches a certain level of contextual trust to each friend that she asks the question. This non-automated second stage is what we call the *trusted social feedback* (TSF). Assuming that such a recommender system will be deployed on a cloud, the TSF proposal must be privacy preserving. Building on this idea, one can ask a group of domain experts (instead of friends), for feedback on questions. We explore this with what we term as *trusted crowdsourcing* (TCS). The TCS proposal also caters for privacy. Furthermore, we extend both TSF and TCS proposals to support multi-valued categorical answers instead of single-valued numerical answers.

Note that automated personal trust transitivity – an idea closely related with the use of trust in the context of social network graphs – is debatable and subjective. It exists but modelling it is difficult. Jøsang et al. in [21] go as far as

[1] Copyright is held by the authors. This work is based on an earlier work: SAC'14 Proceedings of the 2014 ACM Symposium on Applied Computing, Copyright 2014 ACM 978-1-4503-2469-4/14/03. http://dx.doi.org/10.1145/2554850.2554860.

[2] Apart from this notion of trust, we refer to the concept of foreground trust [8] in section 5.

saying "[. . .] all mathematical operators for trust transitivity proposed in the literature must be considered *ad hoc*; they represent attempts to model a very complex human phenomenon as if it were lendable to analysis by the laws of physics". The authors propose a radically different interpretation of trust transitivity based on subjective logic. The authors observe that in order for transitivity to function, the *advisor* must, in some way, communicate his/her trust on the *trust target* to *the originator relying party*. Thus, in our proposals, we rule out automatic estimation of propagated trust.

The rest of the paper is organised as follows. Before delving into describing our proposal, we present the state-of-the-art in section 2 about recommendations using collaborative filtering as well as question-answer services related to privacy and trust; and privacy aware crowdsourcing. Then, we describe the trusted social feedback (TSF) proposal in sub-section 3.1, the trusted crowdsourcing (TCS) proposal in sub-section 3.2 and the capability of both these proposals to handle multi-valued categorical answers in sub-section 3.3. A security analysis is presented in section 4 and the evaluation results on our model in section 5 before concluding in section 6.

2. THE STATE-OF-THE-ART

Herlocker et al.'s work [16] is one of the older works on automated collaborative filtering algorithms. Golbeck's work [12] on FilmTrust utilised trust in social networks for movie recommendations. Guo's work [15] is the closest to ours in the way they combined opinions of neighbours in a social network, weighted by trust values. Unlike our proposal, the paper used the concept of trust propagation and it does not preserve privacy in the aggregation process. Trust propagation is a hard-to-model subjective concept. Two recent proposals: [21] and [27] describe interesting ways of looking at trust propagation. Jamali and Ester [20] employed matrix factorisation to deduce trust propagation, which was then used in collaborative filtering. TidalTrust [13] and MoleTrust [26] are similar with the latter considering ratings from a maximum depth only, in a breadth first search over a trust network to compute a prediction. In [28], authors suggested that the traditional emphasis on user similarity in recommender systems was overstated, and proposed two trust based recommender system solutions. TrustWalker [19] used a random walk method to combine item-based collaborative filtering with trust-based recommendation.

Privacy preserving collaborative filtering (PPCF) has been studied by many [3, 4, 7, 5, 33, 32, 14]. In the context of medical recommendation systems, authors in [17] propose a privacy-friendly architecture where patients submit their ratings in a protected form and the computation of recommendations is done using secure multiparty computation techniques. Existing work can be classified into using either cryptographic or perturbation techniques to preserve privacy. Very few of these proposals have been tested on real world cloud platforms. Canny's work [6] utilised factor analysis and homomorphic encryption for PPCF; while in [14], the authors computed PPCF from a combination of random perturbation and secure multiparty computation. Polat's works [31, 33] have used randomisations to preserve privacy. Several question-answer services exist, including commercial ones, such as Yahoo! Answers, Aardvark. Fleming [10, 11] proposed a privacy enhanced question-answer system based on stigmergic routing where privacy is provided by plausible deniability in a decentralised network.

In [34], the author focuses on the problem of reliability of the work submitted to anonymous members of paid microtask crowdsourcing platforms (e.g., Amazon Mechanical Turk) as well as the privacy of the input given to them. The solution involves generating perturbations of the data to preserve anonymity and a majority voting rule to determine the correct output. Other works involved in achieving reliability for human computations can be found in [22, 23]. A line of research more akin to ours is presented in [35], where the authors introduce a framework to preserve user privacy in geographic crowdsourcing applications by performing user anonymization and delayed upload of information. The limited existing work in privacy tends to focus on privacy of the questions themselves instead of the privacy of the interpersonal trust and that of responses, which are what we concentrate on. In [18], the authors identify two different concerns for privacy in reputation systems and propose a solution based on electronic cash technology and designated verifier proofs.

The field of social aware recommendation is relatively new in comparison with traditional recommender systems. In [1], we developed the first privacy-preserving solution for trusted social feedback. In this article we expand on that work and apply it to the problem of crowdsourcing. Our prior work was limited to single valued numeric responses, where as the current article extends the work to multivalued responses as well, which is much more realistic. We also expand on the security analysis, and perform a completely new evaluation of privacy and user acceptance. Our proposals are about trust empowerment because we see trust as an idiosyncratic, context sensitive, neither entirely rational nor subjective feeling that changes over time [9]. Our privacy preserving solutions also consider privacy from a user-centric perspective.

3. OPINIONS OF PEOPLE

In this section, we describe two methods of obtaining opinions from people: (a) asking trusted friends and (b) asking domain experts through crowdsourcing.

3.1 Trusted social feedback: finding out what friends think

Modelled after a likeness of the word-of-mouth in a human society, the user asks people in her virtual social network for a *trusted social feedback* (TSF) on a query. For the sake of simplicity at this point, a *feedback* is a numeric rating in response to a *query*. In a later section, we will illustrate that the feedback could also be a categorical answer. A query is defined as a question for soliciting an opinion on an item or topic of interest. For instance, a query could be "What is your opinion on the Canon 5D Mark III DSLR camera?".

The feedback acts as a trust empowering information aid to the user in making a choice. In the simplest case, the feedback is an average of the feedback inputs from all friends within one degree of separation, each weighted by the directional trust the user has on that friend. This is similar to the model presented in the FilmTrust work by Jennifer Golbeck [12]. The feedback is obtained per query. Because of the dynamic nature of queries as well as the trust levels specified during queries, no feedback can be pre-defined or stored on the cloud platform that hosts the social network.

In order to preserve privacy, TSF must ensure the non-

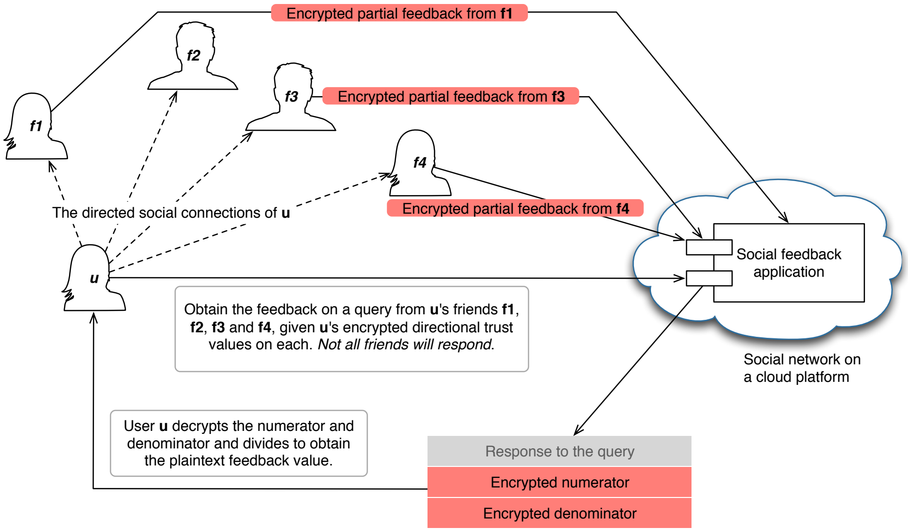

Encrypted partial feedback from **f1**
f1
f2
f3
Encrypted partial feedback from **f3**
f4
Encrypted partial feedback from **f4**
The directed social connections of **u**
u
Social feedback application
Social network on a cloud platform
Obtain the feedback on a query from **u**'s friends **f1**, **f2**, **f3** and **f4**, given **u**'s encrypted directional trust values on each. *Not all friends will respond.*
User **u** decrypts the numerator and denominator and divides to obtain the plaintext feedback value.
Response to the query
Encrypted numerator
Encrypted denominator

Figure 1. Overview of the trusted social feedback mechanism.

disclosure of: (a) the directional trust values in a query to the friends and to the social network; and (b) the feedback from a particular friend of the user to the social network and the user. (c) the aggregated feedback result to the social network or to any friend.

3.1.1 Additively homomorphic cryptosystem – Paillier

Before delving further into the proposal, here we briefly introduce the notion of an additively homomorphic cryptosystem. The Paillier public-key cryptosystem [29] exhibits additively homomorphic properties, which we utilise in our proposals. Denoting encryption and decryption functions by $\mathcal{E}()$ and $\mathcal{D}()$ respectively, the encryption of the sum of two plaintext messages m_1 and m_2 is the modular product of their individual ciphertexts:

$$\mathcal{E}(m_1 + m_2) = \mathcal{E}(m_1) \cdot \mathcal{E}(m_2) \tag{1}$$

while, the encryption of the product of one plaintext messages m_1 and a plaintext integer multiplicand π is the modular exponentiation of the ciphertext of m_1 with π as the exponent:

$$\mathcal{E}(m_1 \cdot \pi) = \mathcal{E}(m_1)^{\pi}. \tag{2}$$

With such an additively homomorphic cryptosystem at our disposal, let us denote the directional trust from user a to friend b as $\mathcal{T}_{a \to b}$, the feedback from a friend i on a query k as $\omega_{i,k}$ and the total number of friends responding to the query as n. The trust value and the individual feedback value are discrete integers. The trusted feedback on query k for user u is given as:

$$\mathcal{F}_{u,k} = \frac{\sum_{i|i \neq u}^{n} \omega_{i,k} \mathcal{T}_{u \to i}}{\sum_{i|i \neq u}^{n} \mathcal{T}_{u \to i}} \tag{3}$$

This computation can be performed over the (additively homomorphic) encrypted domain for user u as:

$$\mathcal{F}_{u,k} = \frac{\mathcal{D}(\prod_{i|i \neq u}^{n} \mathcal{E}(0, r_i) \mathcal{E}(\mathcal{T}_{u \to i})^{\omega_{i,k}})}{\mathcal{D}(\prod_{i|i \neq u}^{n} \mathcal{E}(\mathcal{T}_{u \to i}))} \tag{4}$$

The encryption of zero performed by the friend i, (denoted as $\mathcal{E}(0, r_i)$) ensures[3] that the encrypted partial feedback from friend i, i.e., $\mathcal{E}(0, r_i)\mathcal{E}(\mathcal{T}_{u \to i})^{\omega_{i,k}}$ does not reveal $\omega_{i,k}$ despite the cloud's knowledge of $\mathcal{E}(\mathcal{T}_{u \to i})$, unless the user u and the cloud collaborate. The formal proof is in section 4.2. The trusted social feedback mechanism is illustrated in figure 1 and is described in algorithm 1[4]. While sending the question, the user attaches an encrypted trust value for each friend to the question such that when a friend responds, the response is homomorphically multiplied by the trust value. The cloud aggregates those individual responses from the friends and sends back the aggregate response to the user after a threshold number of friends have responded. The flow of information in TSF is shown in figure 2.

As trust is personal and idiosyncratic [9], our proposed feedback mechanism is only there for trust empowerment, not to enforce a trust decision on the user. What the user does with the feedback is solely her choice. Therefore, a mathematical model for trust transitivity over multiple degrees of separation in the social network graph is often inadequate and meaningless because the model would tend to suggest a particular trust level. Trust is also sensitive to changes over time and context. In our proposal, the trust values can be as short-lived as a single query, which caters for temporal changes. The user can solicit the response to her query from a selected group (based on any particular context) of friends, thereby enabling context sensitivity. Thus, the queries in TSF are short-lived and context sensitive.

Untrust [25], which can be expressed in our proposed feedback mechanism, is also context sensitive. This means that Alice could trust her friend Bob for an opinion on cloud security but at the same time untrust him regarding any opinion on quantum entanglement. Untrust can prove use-

[3] The notations $\mathcal{E}(x, r_u)$ and $\mathcal{E}(x)$ are synonymous, i.e., encryption performed by the user u. The random number notation is used only when the operation is performed by some other user i with u's public key, i.e., $\mathcal{E}(x, r_i)$.

[4] The $\cdot$ is used to denote multiplication for the sake of readability.

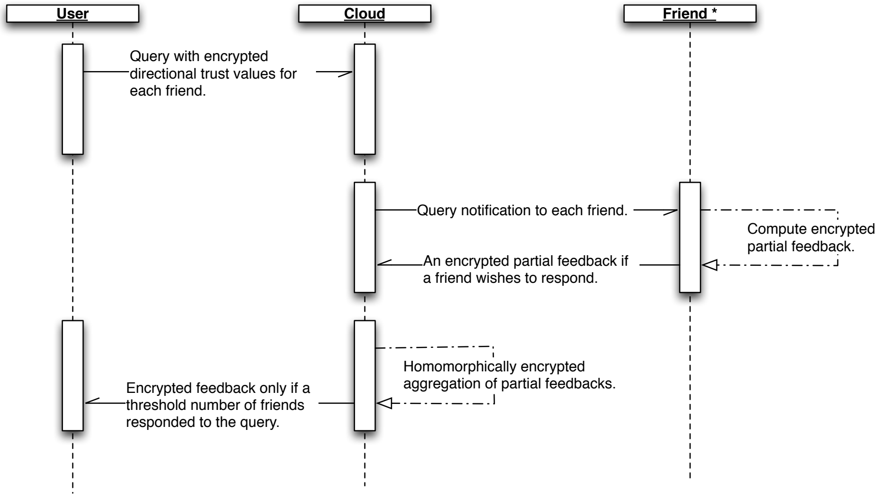

User
Cloud
Friend *
Query with encrypted directional trust values for each friend.
Query notification to each friend.
Compute encrypted partial feedback.
An encrypted partial feedback if a friend wishes to respond.
Homomorphically encrypted aggregation of partial feedbacks.
Encrypted feedback only if a threshold number of friends responded to the query.

Figure 2. The trusted social feedback sequence.

ful to accept negative feedback or reject positive feedback from untrusted friends for specific queries. In our current prototype, we do not model untrust.

ALGORITHM 1: Computing the trusted social feedback for user u on item k.

```
Require: Additively homomorphic encrypted domain for user $u$, i.e., $\mathcal{E}$ and corresponding public key.
Require: Encrypted directional trust $\mathcal{E}(\mathcal{T}_{u\to i})$ from user $u$ to each friend $i$.
for each encrypted directional trust $\mathcal{E}(\mathcal{T}_{u\to i})$ do
    if $i$ wishes to respond then
        $i$ computes encrypted partial feedback,
            $\psi_i \leftarrow \mathcal{E}(0, r_i) \cdot \mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}$
        social network updates encrypted trusted feedback,
            $\Psi \leftarrow \Psi \cdot \psi_i$
        social network updates encrypted response cardinality,
            $\eta \leftarrow \eta \cdot \mathcal{E}(\mathcal{T}_{u\to i})$
    end if
end for
return encrypted trusted feedback, $\Psi$.
return encrypted response cardinality, $\eta$.
user $u$ obtains the trusted social feedback, $\mathcal{F}_{u,k} = \frac{\mathcal{D}(\Psi)}{\mathcal{D}(\eta)}$.
```

3.2 Trusted crowdsourcing: opinions of domain experts

An extension of the *trusted social feedback* is a concept, which we call the *trusted crowdsourcing* (TCS). Through TCS, the user no longer solicits opinions from friends but uses a crowdsourcing platform to pose questions to domain experts. We assume the existence of a reputation system for domain experts through which experts gain or lose reputation based on how their answers are received. A mechanism to infer reputation based on the TCS model is left for future work.

The TCS ensures the non-disclosure of: (a) the directional trust values in a query to the eligible responders and to the crowdsourcing platform; (b) the feedback from a particular responder to the crowdsourcing; and (c) the aggregated feedback result to the crowdsourcing platform or to any responder.

The key difference between TCS and TSF is that in TCS, the user does not know at the time of the query who in the crowd will be eligible and willing to respond. Furthermore, the absence of a social network connection between the user and the potential responder means that it is impossible for the user to specify directional trust values at the time of query as it was done in TSF. However, before letting a domain expert answer a query, the crowdsourcing platform can obtain, from the user, the encrypted trust that the user chooses to assign to the responder. Thus, before receiving an answer, the user knows the identities of the responders – this is in sharp contrast with the TSF model. From the perspective of the user, the trust may be specified based on the reputation of the responder. In keeping with the understanding that trust is personal and idiosyncratic, the trust in TCS is personal; reputation is a trust aid, not a trust enforcement. Apart from the reputation, the response eligibility criteria is similar to that used in most crowdsourcing platforms. For instance, if the user asks the question “I am going to Japan next week for the first time. Which places would you recommend visiting?” then a likely response eligibility criteria is that the responder must have visited Japan and/or have lived/currently living in Japan. The eligibility criteria is user-specified and may not exist if the user so chooses.

Responders express their wish to the crowdsourcing platform to respond to a particular question. If the eligibility criteria is met, the crowdsourcing platform is responsible for obtaining and sending the encrypted trust values to each el-

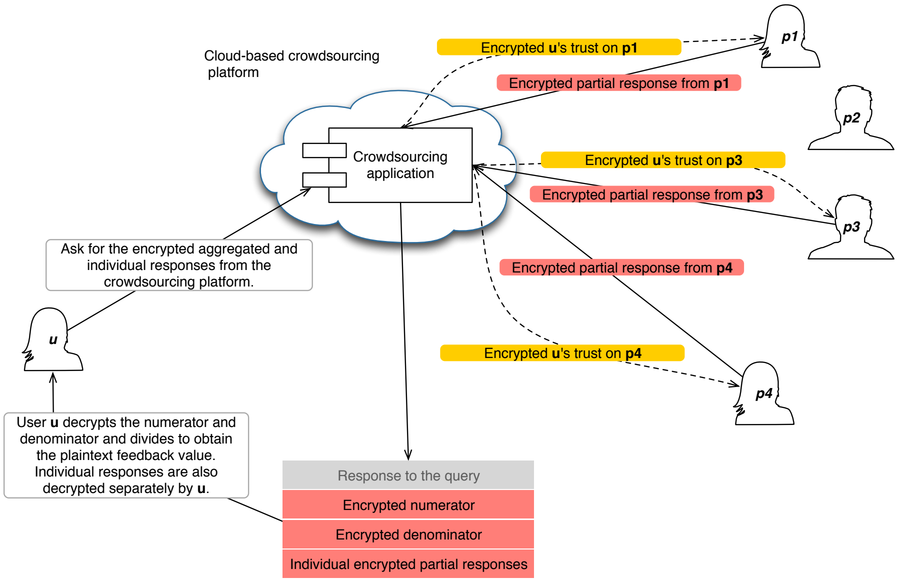

Cloud-based crowdsourcing platform
Crowdsourcing application
Encrypted u's trust on p1
Encrypted partial response from p1
p1
p2
Encrypted u's trust on p3
Encrypted partial response from p3
p3
Encrypted partial response from p4
Encrypted u's trust on p4
p4
Ask for the encrypted aggregated and individual responses from the crowdsourcing platform.
u
User u decrypts the numerator and denominator and divides to obtain the plaintext feedback value. Individual responses are also decrypted separately by u.
Response to the query
Encrypted numerator
Encrypted denominator
Individual encrypted partial responses

Figure 3. Overview of the trusted crowdsourcing mechanism.

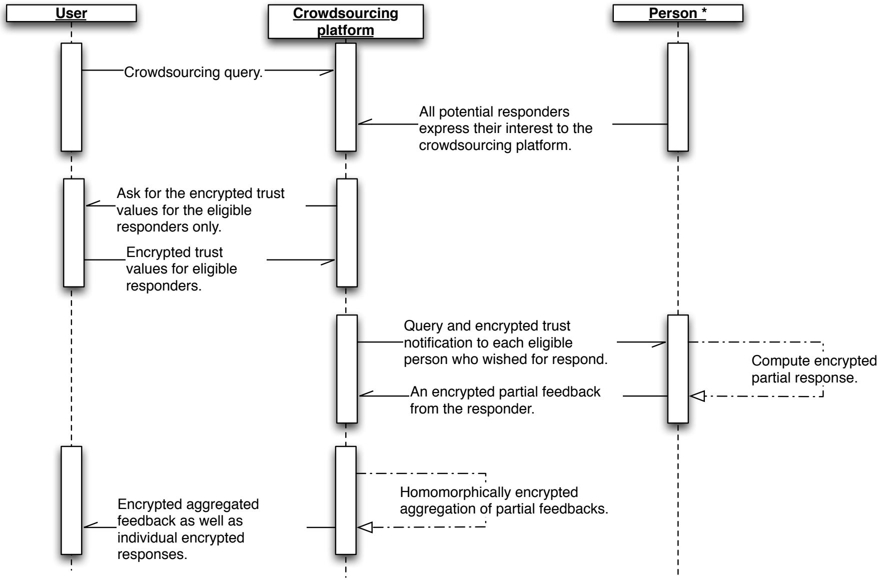

User
Crowdsourcing platform
Person *
Crowdsourcing query.
All potential responders express their interest to the crowdsourcing platform.
Ask for the encrypted trust values for the eligible responders only.
Encrypted trust values for eligible responders.
Query and encrypted trust notification to each eligible person who wished for respond.
Compute encrypted partial response.
An encrypted partial feedback from the responder.
Homomorphically encrypted aggregation of partial feedbacks.
Encrypted aggregated feedback as well as individual encrypted responses.

Figure 4. The trusted crowdsourcing sequence.

igible responder. From that point forward, TCS works like TSF. An overview diagram of TCS is shown in figure 3 while the operating steps are described in algorithm 2, which is only slightly modified in comparison with algorithm 1. The flow of information in TCS is shown in figure 4. Notice that the crowdsourcing platform sends back to the user, both the aggregated response as well as the individual responses.

Although the mechanism to build reputations of responders is left for future work, a potential way to infer reputations could be to compare individual answers with the collective answer. This might help the user form a better understanding of where each individual answer stands. In addition, the availability of individual answers also helps with forming a better view of the distribution of answers than just relying on the mean.

3.3 Numeric answers to categorical answers

Although, in an example in the previous section, we mentioned a question “I am going to Japan next week for the first time. Which places would you recommend visiting?”

ALGORITHM 2: Computing the trusted crowdsourcing response for user u on item k.

```
Require: Additively homomorphic encrypted domain for
    user $u$, i.e., $\mathcal{E}$ and corresponding public key.
Require: Encrypted directional trust $\mathcal{E}(\mathcal{T}_{u \to i})$ from user $u$
    based on the reputation level (or any other deciding
    factor) of each person $i$ who is eligible to respond.
for each encrypted directional trust $\mathcal{E}(\mathcal{T}_{u \to i})$ do
    if $i$ wishes to respond then
        $i$ computes encrypted partial response,
            $\psi_i \leftarrow \mathcal{E}(0, r_i) \cdot \mathcal{E}(\mathcal{T}_{u \to i})^{\omega_{i,k}}$
        crowdsourcing platform updates the encrypted
        trusted response,
            $\Psi \leftarrow \Psi \cdot \psi_i$
        crowdsourcing platform updates encrypted
        response cardinality,
            $\eta \leftarrow \eta \cdot \mathcal{E}(\mathcal{T}_{u \to i})$
    end if
end for
return encrypted trusted response, $\Psi$.
return encrypted response cardinality, $\eta$.
return each encrypted trusted response, $\psi_i$.
user $u$ obtains the trusted crowdsourcing response,
    $\mathcal{F}_{u,k} = \frac{\mathcal{D}(\Psi)}{\mathcal{D}(\eta)}$.
```

sent out to the crowd, the TSF or TCS models described so far cannot support the answer to such a question. The reason is that the both the TSF and the TCS models only support single-valued numeric answers to questions such as "What is your opinion on the Canon 5D Mark III DSLR camera?".

However, this can be easily extended to support a multi-valued categorical answer. Let us re-consider the question that requires a multi-valued answer: "I am going to Japan next week for the first time. Which places would you recommend visiting?". Let us, now, assume that the potential places are to be chosen from a list of classes: *Tokyo*, *Kyoto*, *Hakone* and *Sapporo*. For simplicity, we will assume that each responder will have to include all these classes in the response. A system that allows for ranking only some classes by each user is to be investigated in the future, and may bear some similarity to our work on rank aggregation of partially ranked lists [2].

With each response including all classes, a responder p_1 may respond with the following class values $Tokyo = 0.4$, $Kyoto = 0.3$, $Hakone = 0.2$, $Sapporo = 0.1$, thus specifying a probability distribution with the class values summing up to 1. Someone else p_2 can specify: $Tokyo = 0.2$, $Kyoto = 0.3$, $Hakone = 0.3$, $Sapporo = 0.2$. We can apply the TSF or the TCS algorithm on each class independently of the others. Thus, the end result computed by the user, $\mathcal{F}_{u,k}$ will be per class, which can be normalised further to obtain the correct probability distribution. Table 1 is an example, in the plaintext domain, of a multi-valued answer. Even if the final weighted averages (*Weighted mean* in table 1) do not add up to unity, we do not need to normalise in order to find a ranking from the answer. In the aforementioned example, *Tokyo* is the most preferred place to visit followed, in order, by *Kyoto*, *Hakone* and *Sapporo*. Note that the implementation will have to scale the class values in the probability distribution to integer-only domain so that those values can be used by a homomorphic cryptosystem that works only with integers.

4. SECURITY ANALYSIS

In this section, we present the description of various attacks by honest-but-curious adversaries and discuss how our model copes. Furthermore, we also provide a proof of obfuscation by the encryption of zero. We also present a specialised partial response disclosure attack.

4.1 Honest but curious adversary model

In this discussion, the word *cloud* will be considered synonymous with either the *social network* or the *crowdsourcing platform* in terms of threats because both will usually utilise a cloud environment. Thus, the internal privacy threats to either can arise from the cloud infrastructure. We assume that the parties involved in this process are ***honest but curious***. Therefore, attacks involving collaborations between the cloud and the attacker are not considered as realistic threats although we have described some such possible attacks. For a malicious user, a specialised attack for partial response disclosure is also described in section 4.3.

4.1.1 Curious user, multi-query and sybil attacks

The user can run multiple queries requesting the feedback on the same question from the same set of friends or domain experts. In doing so, and by varying the user's directional trust on each responder, the user can acquire the information necessary to reveal the feedback provided by each responder. However, the feedback response is slow and some responders may choose not to respond. Furthermore, the feedback from the same person may vary over time. In addition, with crowdsourcing, each query may cost money making this an expensive attack. Therefore, using a multi-query attack is not guaranteed to succeed. To further enhance the privacy of the feedback, the responder can perturb his/her feedback input in bounded integral ranges – an avenue we have left open for future work.

However, in a *sybil attack* the user asks a question to one real person and a number of sybil identities. Upon receiving the responses, the asker can find out the exact response from the real person given the knowledge of those from the sybil identities. Our model is not resistant against this type of sybil attacks.

4.1.2 Curious cloud, man-in-the-middle attack

Despite the query itself being sent in clear text, the directional trust values from the user and the partial feedback from each responder are both in the encrypted domain of the user. Even though the cloud knows the encrypted directional trust value, it cannot decipher the actual feedback from any responder since encrypted zero, i.e., $\mathcal{E}(0, r_i)$, is homomorphically added by each responder thus making the encrypted trusted feedback component probabilistic. The cloud, however, can tell who responded to the query.

4.1.3 Curious responder

Any particular responder cannot determine the directional trust value because it is encrypted by the user's public key.

Table 1. An example of multi-valued answer.

	Tokyo	**Kyoto**	**Hakone**	**Sapporo**	*Trust*
Person 1 (p_1)	0.4	0.3	0.2	0.1	0.9
Person 2 (p_2)	0.2	0.3	0.3	0.2	0.5
Person 3 (p_3)	0.1	0.4	0.3	0.2	0.4
Person 4 (p_4)	0.5	0.1	0.2	0.2	0.8
Weighted mean	0.34615	0.25385	0.25	0.16538	–
Class rank	1	2	3	4	–

4.1.4 *Collaborative attacks*

If the user and the cloud collaborate then all the partial feedbacks can be deciphered since the cloud will be able to decrypt partial feedback values with the help of the user. If a responder and the cloud collaborate, the responder can learn how many other people responded to the query but it cannot decipher the actual individual feedback values. If the user and a responder collaborates, they can only learn about each others' secrets – the directional trust value and the feedback.

4.1.5 *Out-of-the-range attacks*

Both the responder and the cloud can encrypt arbitrary numbers and send them to the user in the response. Homomorphic range check protocols [30] may be applicable to protect those scenarios but this falls within the remits of future work.

4.2 Proof of obfuscation by encryption of zero

Since the numeric feedback on item k from a responder, i, is in a fixed discrete integral range, the cloud can attempt to learn it by pre-computing all possible values[5] of $\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}$ using a trial-and-error method of dividing what the responder sends by the pre-computed value to eliminate the obfuscating encryption of zero. Let us assume that the correct value of $\omega_{i,k}$ in question is ω_1 and a wrong value is ω_2. This is what happens.

4.2.1 *Case A: correct pre-computed value*

If the cloud used the correct pre-computed value: $\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_1}$, we have:

$$\begin{aligned}\frac{\mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}}{\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_1}} &= \mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}-\omega_1}\\ &= \mathcal{E}(0,r_i)\end{aligned}$$

Now, the cloud computes:

$$\begin{aligned}\frac{\mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}}{\mathcal{E}(0,r_i)} &= \mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}\\ &= \mathcal{E}(\mathcal{T}_{u\to i})^{\omega_1}\end{aligned}$$

Thus, the cloud obtains the same value as the one it pre-computed.

4.2.2 *Case B: wrong pre-computed value*

If the cloud used a wrong pre-computed value: $\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_2}$, we have:

$$\frac{\mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}}{\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_2}} = \mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}-\omega_2}$$

Now, the cloud computes:

$$\begin{aligned}\frac{\mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}}{\mathcal{E}(0,r_i)\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}-\omega_2}} &= \mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}-\omega_{i,k}+\omega_2}\\ &= \mathcal{E}(\mathcal{T}_{u\to i})^{\omega_2}\end{aligned}$$

Here again, the cloud obtains the same value as the one it pre-computed.

Since the results from both the right and the wrong guesses are indistinguishable, the cloud cannot guess which one is the true value of $\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}$ and hence $\omega_{i,k}$.

4.3 A specialised partial response disclosure attack

Our construction is not inherently secure against a malicious user wishing to know the responses of her responders from the aggregate encrypted values. This attack consists of creating a vector with several coordinates inside a single encrypted value. These coordinates can be read independently by the malicious user. Consider an x-bit number and treat it as a vector of dimension y, where each coordinate is represented using $\frac{x}{y}$ bits. If operations are performed on this vector with no individual coordinate exceeding $2^{\frac{x}{y}}-1$; then there is no loss of information for that coordinate. The following example illustrates this idea.

1. Assume $x = 16$ and $y = 4$, then each coordinate can represent values in the range $[0 \quad 15]$.
2. The user asks four responders, i.e., $f_1 \ldots f_4$ a question using the following trust values (spaces introduced for readability), represented as bit sequences.

$$\begin{aligned}\mathcal{T}_{u\to f_1} &= 0000\,0000\,0000\,0001\ [decimal: 1]\\ \mathcal{T}_{u\to f_2} &= 0000\,0000\,0001\,0000\ [decimal: 32]\\ \mathcal{T}_{u\to f_3} &= 0000\,0001\,0000\,0000\ [decimal: 512]\\ \mathcal{T}_{u\to f_4} &= 0001\,0000\,0000\,0000\ [decimal: 8192]\end{aligned}$$

3. Each responder provides his/her response in the range $[1 \quad 15]$ weighted by the ingress trust value, i.e.,

$$\mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}.$$

The encryption of zero is left out for simplicity because it does not stop this attack, which happens in the plaintext domain.

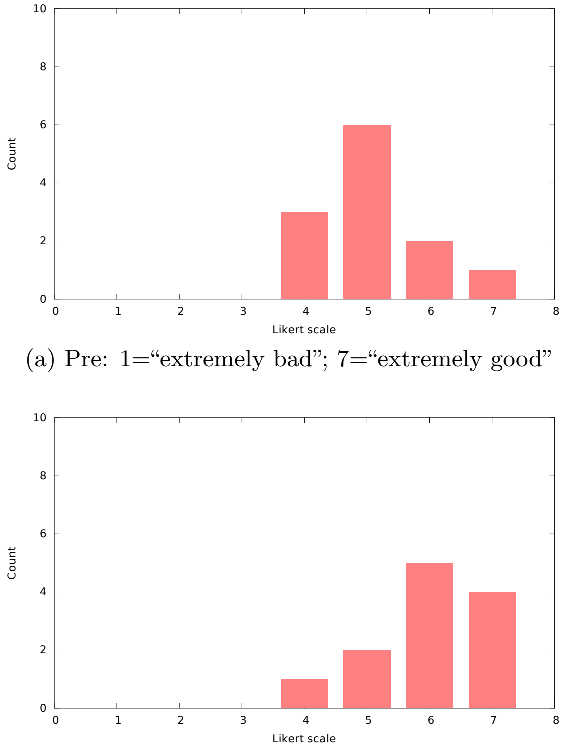
4. The cloud aggregates the resultant numerator as:

$$\prod_{i|i\neq u}^{n} \mathcal{E}(\mathcal{T}_{u\to i})^{\omega_{i,k}}$$

[5] Note that this homomorphic multiplication has deterministic values.

Since none of the coordinates in the numerator has a value greater than 15, the malicious user can extract the answer from each responder by reading the decrypted numerator, 4-bits at a time. The technique also works for trust values where the particular non-zero nibble is greater than 0001, for example 0010 0000 0000 0000 [$decimal$: 16384]. In that case, the malicious user simply needs to adapt the coordinate length accordingly and once extracted, divide it by the original trust value assigned to that particular responder. To prevent this attack, a proof stating that the trust values are in a given range is necessary. Alternatively, if the number of responders asked is large enough in comparison with the bit space of the plaintext trust values then the bit manipulations will overlap, thus making it impossible for the attacker to identify individual ratings.

5. IMPLEMENTATION AND EVALUATION



(a) Pre: 1="extremely bad"; 7="extremely good"

(b) Post: 1="extremely bad"; 7="extremely good"

Figure 5. Responder's change in the understanding of the TSF prototype (Q1), pre- and post- experiment.

In this section we present the results from: 1. the user studies for the trusted social feedback prototype; 2. the performance evaluation of the speed of cryptographic primitives on the web front-end; 3. the speed of the essential functions of the prototype at the back-end of the trusted social feedback prototype; and 4. the user studies for the trusted crowdsourcing model.

The experimental trusted social feedback prototype runs on the Google App Engine for Java. The application uses Facebook to perform user login and to determine social connections.

Table 2. Change in uncertainties associated with 4 questions and 12 users.

(a) **Change in response uncertainties for each question per user.**

	Q1	Q2	Q3	Q4
U1	Reduced	No change	Reduced	Reduced
U2	Reduced	Reduced	Reduced	No change
U3	No change	Reduced	Reduced	Increased
U4	Reduced	Reduced	Increased	No change
U5	Reduced	Reduced	Reduced	Reduced
U6	No change	Reduced	No change	Reduced
U7	Reduced	No change	Reduced	Reduced
U8	No change	Reduced	Reduced	No change
U9	Reduced	Reduced	Reduced	No change
U10	No change	Reduced	Reduced	Reduced
U11	Reduced	Reduced	Reduced	Reduced
U12	Reduced	Reduced	No change	Reduced

(b) **Change in response uncertainties per question.**

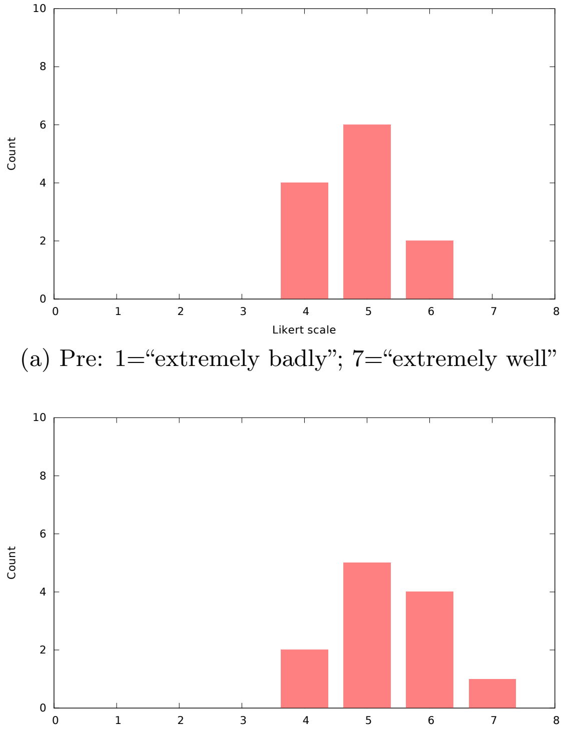
	Increased	No change	Reduced
Q1	0 (0%)	4 (33%)	8 (67%)
Q2	0 (0%)	2 (17%)	10 (83%)
Q3	1 (8%)	2 (17%)	9 (75%)
Q4	1 (8%)	4 (33%)	7 (58%)

Table 3. Performances of Paillier in Javascript. Times are in milliseconds. KG: key generation, E: encryption, HA: homomorphic add; HM: homomorphic multiplication; D: decryption. The number suffixed to these abbreviations indicate cryptosystem bit size.

	Chrome	Firefox	IE	Safari
KG-512	69	57	441	1032
E-512	23	16	195	367
HA-512	2	2	9	27
HM-512	11	8	104	214
D-512	23	15	195	371

Table 4. The average times taken for various servlet function calls. The `profile` servlet is responsible for user profile specific functions while `qaserv` deals with questions and their responses.

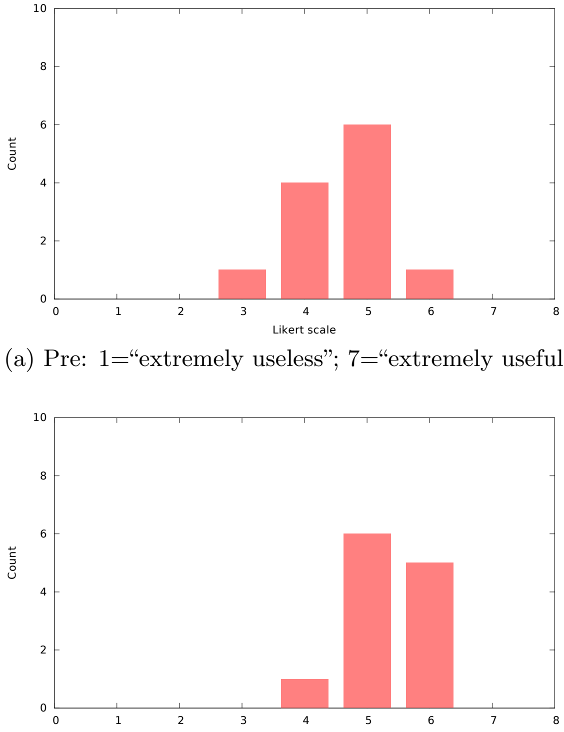
Servlet:Action	Call count	Time (ms)
`profile:getProfile`	121	3128
`profile:getTopUsers`	207	1816
`profile:savePublicKey`	36	2012
`qaserv:answerQuestion`	195	233
`qaserv:askQuestion`	81	1826
`qaserv:myNotifications`	552	783
`qaserv:myQuestions`	262	1779



(a) Pre: 1="extremely badly"; 7="extremely well"

(b) Post: 1="extremely badly"; 7="extremely well"

Figure 6. Responder's change in the perception of how well privacy is preserved by the TSF prototype (Q2), pre- and post- experiment.



(a) Pre: 1="extremely useless"; 7="extremely useful"

(b) Post: 1="extremely useless"; 7="extremely useful"

Figure 7. Responder's change in the perception of usefulness of the TSF prototype (Q3), pre- and post- experiment.

5.1 TSF: measuring perception of privacy and foreground trust

The following questions were used to record user responses and measure changes in uncertainty in these responses. Each question was presented once before and once after each user had a chance to play with the experimental prototype.

Q1 How is your understanding about what you can do with this application?

Q2 How well do (did) you feel that the application will preserve (preserved) the privacy of the personal trust levels that you have on your friends, and the privacy of the responses from your friends?

Q3 How useful do you think is this application?

Q4 How likely are you to use such an application, if available publicly?

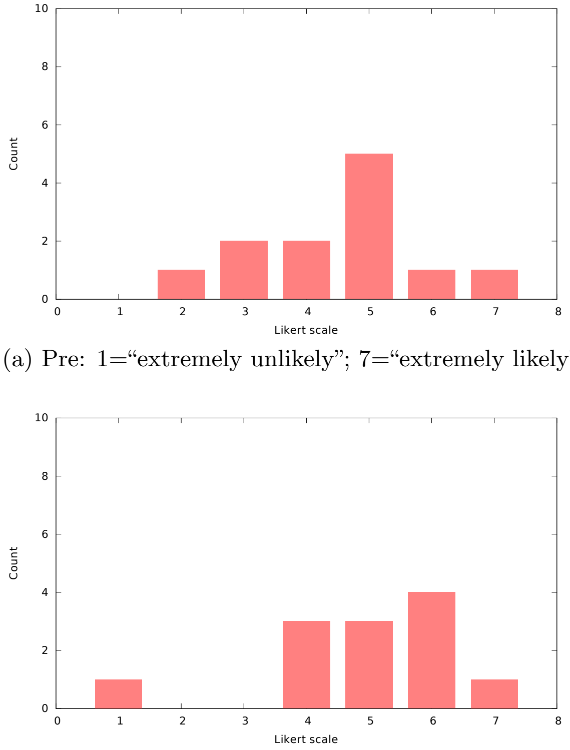
Each question was followed by a question to measure uncertainty: *How certain are you about your previous response?*. Responses to each question was recorded on a 7-point Likert scale [24].

Figures 5 through 8 show the user responses before and after the experiment in terms of understanding, privacy-preservation, usefulness and likelihood of use of the prototype. Together, the figures show the change in the user perceptions before and after the experiment. In figure 5, the results suggest that use of the prototype may have helped the users to understand the application better. Similarly, figure 6 shows that the the usersâĂŹ perception of how well privacy is preserved increased after using the prototype. The users also perceived the application to be more useful after having used the prototype, which is illustrated by figure 7. According to figure 8, the likelihood of users using such a system in the future also slightly improved overall after having used the prototype. Thus, this suggests that our prototype was perceived by the users to be an effective tool for preserving privacy while obtaining feedback.

Closely related to the sense of privacy, this user study also inferred users' trust in the application, which relates to the concept of foreground trust [8]. It is different from the trust between friends that we have discussed so far. Dwyer et al. in [9], suggested that a reduction of uncertainty is positively correlated with the increase of trust. Thus, a measure of uncertainty is used to infer trust. In our user study with 12 participating users, we have employed pre-use and post-use questionnaires to determine the changes in uncertainty. The users are highly technically competent and were aware of this research work before using the prototype. Table 2 shows that the uncertainty in the users' responses usually declined, thus suggesting a likely increase in foreground trust.

5.2 TSF: measuring performance

The speed at which a feedback can be obtained depends almost entirely on the speed at which friends respond to the question; and to some extent on the speed of cryptographic operations and that too on the client-side because the speed of the limited cryptographic operations on the cloud-side is usually negligible compared to delays caused by network la-



(a) Pre: 1="extremely unlikely"; 7="extremely likely"

(b) Post: 1="extremely unlikely"; 7="extremely likely"

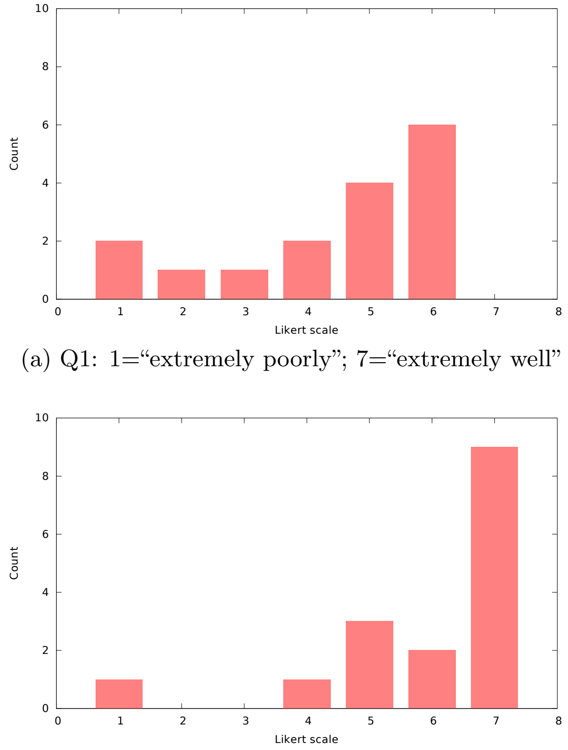
Figure 8. Responder's change in the likeliness of the future use of the TSF prototype (Q4), pre- and post-experiment.

tencies, cloud instance initialisations, and datastore access. For every partial response submitted by a friend, the cloud is responsible for exactly two homomorphic additions, see line 4 in algorithm 1. We present a comparison of performances of cryptographic primitives on the client side. We have built a Google Web Toolkit wrapper for an optimised Javascript implementation of the Paillier cryptosystem using the Stanford Javascript BigInteger library. The result of each test, in table 3, is a rounded-off average from 50 runs. The tests were carried out on Windows 8, running on a 64-bit 3.4GHz Intel i7-3770 dual quad-core processor with 16GB RAM. The versions of the browsers are: Chrome 28.0.150072m, Firefox 22.0, Internet Explorer (IE) 10.0.9200.16599 and Safari 5.1.7. IE and Safari failed to finish the tests when the cryptosystem was set to 1024 bits, so we used a 512 bits cryptosystem for our tests.

Using the F1 (600MHz) instance class and the high-replication datastore of the Google App Engine, the averages of the times taken for the different servlet calls are shown in table 4. The time taken for a particular function call also includes the time taken to execute any intermediate servlet filters, for instance the filter that verifies the logged-in users.

5.3 TCS: measuring user acceptance

Further to measuring foreground trust and sense of privacy for the trusted social feedback implementation, we also measured user acceptance of the trusted crowdsourcing model.



(a) Q1: 1="extremely poorly"; 7="extremely well"

(b) Q2: 1="extremely unimportant"; 7="extremely important"

Figure 9. The perception and importance of privacy from Alice's viewpoint.

The public questionnaire[6] was preceded by a brief description of the TCS model. The users were asked the following questions after reading the description of the model.

Q1 If you were Alice, how well do you feel that the privacy of your personal trust value in Bob is preserved by CrowdTrust?

Q2 If you were Alice, how important would this privacy be to you?

Q3 If you were Bob (the responder) how well do you feel that the privacy of your answer to Alice is hidden from the crowdsourcing platform by CrowdTrust?

Q4 If you were Bob, how important would this privacy be to you?

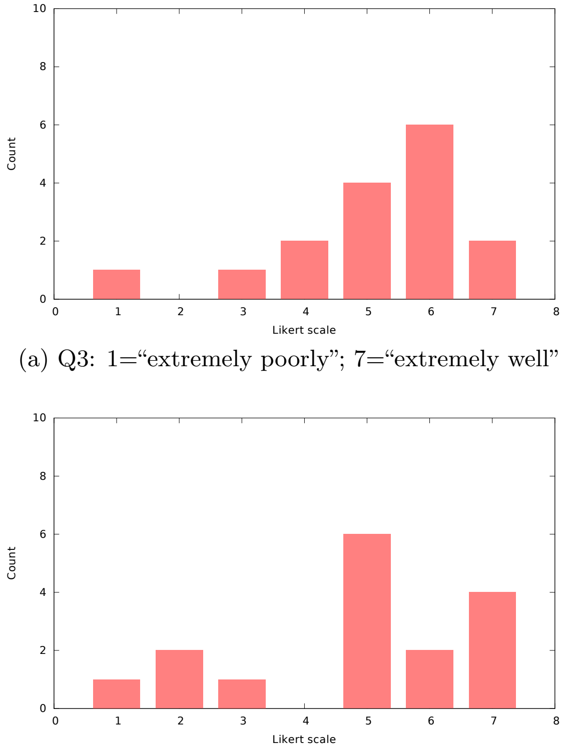
Q5 How likely would you be to use such a service to ask for opinions from the crowd?

Q6 How likely would you be to use such a service to answer questions, get remunerated and develop reputation?

Q7 Would you prefer CrowdTrust to be implemented as an end-to-end user communication where the crowdsourcing platform is oblivious to the privacy preserving layer? Or, do you prefer an existing crowdsourcing platform to integrate and provide CrowdTrust as a separate add-on service?

Responses to each question was recorded on a 7-point Likert scale. There were two other questions that were used to collect comments and concerns from the responders to the survey.

[6] Online questionnaire available at: `http://goo.gl/eIyDx4`.



(a) Q3: 1="extremely poorly"; 7="extremely well"

(b) Q4: 1="extremely important"; 7="extremely important"

Figure 10. The perception and importance of privacy from Bob's viewpoint.

Q8 If you were Alice, would you have any concerns or suggestions?

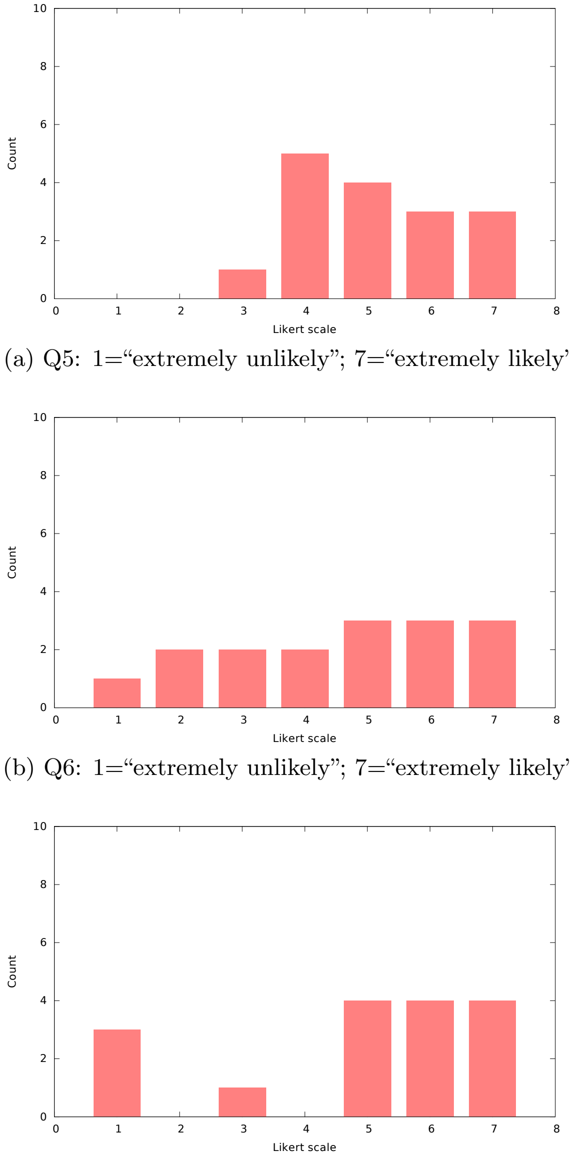
Q9 If you were Bob, would you have any concerns or suggestions?

The responses to Q8 and Q9 were used to refine the TCS model, while the analysis of the results of responses to Q1 through Q7 are presented below. At the time of this writing, the survey had 16 responses. Figure 9 shows to what extent Alice (i.e., the person who uses the crowdsourcing platform to send her query) was convinced that her privacy was preserved in the model, and how important that privacy is to her. Similarly, figure 10 correspondingly shows to what extent Bob (i.e., the responder's) was convinced that her privacy was preserved in the model, and how important that privacy is to him. Figure 11 illustrates to what extent a user would use such a system if they were playing the role of Alice/Bob, and also shows their preferences regarding the form of implementation (end-to-end/integrated) of the TCS framework.

6. CONCLUSIONS

In this paper, we have presented a working prototype for obtaining feedback on queries from one's trusted friends in a privacy preserving manner. We have implemented and tested the prototype on the Google App Engine with Facebook, and have run a user study to evaluate the perception of privacy as well as foreground trust in the prototype. The evaluation shows that the prototype was considered by the users to be an effective tool for preserving privacy while obtaining feedback; and that it can be inferred that the foreground trust in the prototype was high. A novel contribution of the paper is the observation that the technique for privacy-preserving social feedback can also be used in the domain of crowdsourcing. Based on this, we develop a framework for trusted crowdsourcing to enable users to obtain feedback from domain experts who may not necessarily be friends. We have evaluated users' perception of privacy preservation and the acceptance of our trusted crowdsourcing model through a user survey. The results of the survey demonstrate that the users, in general, felt that the model was good at preserving the privacy.

In the future, we plan to expand this work by developing:



(a) Q5: 1="extremely unlikely"; 7="extremely likely"

(b) Q6: 1="extremely unlikely"; 7="extremely likely"

(c) Q7: 1="definitely end-to-end"; 7="definitely integrated"

Figure 11. Alice's and Bob's preferences on the implementation of privacy-preserving trusted crowdsourcing and the likelihood of use.

(a) a mechanism to infer reputations of responders in the trusted crowdsourcing model; (b) means to support incompleteness in multi-valued categorical answers; and (c) practical homomorphic range checks to protect out-of-range attacks in the encrypted domain. This should enable wider acceptance of crowdsourcing even in sensitive applications.

7. REFERENCES

[1] A. Basu, J. C. Corena, S. Kiyomoto, S. Marsh, J. Vaidya, G. Guo, J. Zhang, and Y. Miyake. Privacy preserving trusted social feedback. In *Proceedings of the 29th ACM Symposium on Applied Computing (SAC) TRECK track, Gyeongju, Korea*, 2014.

[2] A. Basu, J. C. Corena, S. Kiyomoto, J. Vaidya, S. Marsh, and Y. Miyake. PrefRank: fair aggregation of subjective user preferences. In *Proceedings of the 29th ACM Symposium on Applied Computing (SAC) RS track, Gyeongju, Korea*, 2014.

[3] A. Basu, J. Vaidya, and H. Kikuchi. Perturbation based privacy preserving Slope One predictors for collaborative filtering. In *Proceedings of the 6th IFIP WG 11.11 International Conference on Trust Management (IFIPTM), Surat, India*, 2012.

[4] A. Basu, J. Vaidya, H. Kikuchi, T. Dimitrakos, and S. K. Nair. Privacy preserving collaborative filtering for SaaS enabling PaaS clouds. *Journal of Cloud Computing: Advances, Systems and Applications*, 1(8), 2012.

[5] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci. Privacy-enhanced collaborative filtering. In *Proc. User Modeling Workshop on Privacy-Enhanced Personalization*, 2005.

[6] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 238–245, New York, NY, USA, 2002. ACM.

[7] R. Cissée and S. Albayrak. An agent-based approach for privacy-preserving recommender systems. In *Proceedings of the 6th International Joint Conference on Autonomous agents and Multiagent Systems*, pages 1–8. ACM, 2007.

[8] N. Dwyer. *Traces of digital trust: an interactive design perspective*. PhD thesis, Victoria University, 2011.

[9] N. Dwyer, A. Basu, and S. Marsh. Reflections on measuring the trust empowerment potential of a digital environment. In *Proceedings of the IFIP WG11.11 International Conference on Trust Management (IFIPTM), Malaga, Spain*, 2013.

[10] S. Fleming. *An ant-inspired, deniable routing approach in ad hoc question & answer networks*. PhD thesis, University of Sussex, 2012.

[11] S. Fleming, D. Chalmers, and I. Wakeman. A deniable and efficient question and answer service over ad hoc social networks. *Information retrieval*, 15(3-4):296–331, 2012.

[12] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *Proceedings of the 4th International Conference on Trust Management (iTrust), Pisa, Italy*. Springer, 2006.

[13] J. A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland, College Park, MD, USA, 2005. AAI3178583.

[14] S. Gong. Privacy-preserving collaborative filtering based on randomized perturbation techniques and secure multiparty computation. *IJACT: International Journal of Advancements in Computing Technology*, 3(4), 2011.

[15] G. Guo, J. Zhang, and D. Thalmann. A simple but effective method to incorporate trusted neighbors in recommender systems. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization (UMAP)*, pages 114–125. Springer, 2012.

[16] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.

[17] T. R. Hoens, M. Blanton, and N. V. Chawla. Reliable medical recommendation systems with patient privacy. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 173–182. ACM, 2010.

[18] R. Ismail, C. Boyd, A. Jøsang, and S. Russel. Strong privacy in reputation systems. In *Proceedings of the 4th International Workshop on Information Security Applications (WISA)*, 2003.

[19] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.

[20] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM.

[21] A. Jøsang, T. Aždерska, and S. Marsh. Trust transitivity and conditional belief reasoning. In *Trust Management VI*, pages 68–83. Springer, 2012.

[22] D. R. Karger, S. Oh, and D. Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 284–291. IEEE, 2011.

[23] R. Kern, H. Thies, and G. Satzger. Statistical quality control for human-based electronic services. In *Service-Oriented Computing*, pages 243–257. Springer, 2010.

[24] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[25] S. Marsh and M. R. Dibben. Trust, untrust, distrust and mistrust – an exploration of the dark(er) side. In *Trust Management*, pages 17–33. Springer, 2005.

[26] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 17–24, New York, NY, USA, 2007. ACM.

[27] T. Muller and P. Schweitzer. On beta models with trust chains. In *Trust Management VII*, pages 49–65.

Springer, 2013.

[28] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, 2005.

[29] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT'99*, volume 1592, pages 223–238. Springer, 1999.

[30] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electronic Commerce Research and Applications*, 7(3):294–312, 2008.

[31] H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 625–628. IEEE, 2003.

[32] H. Polat and W. Du. Privacy-preserving collaborative filtering on vertically partitioned data. *Knowledge Discovery in Databases: PKDD 2005*, pages 651–658, 2005.

[33] H. Polat and W. Du. Achieving private recommendations using randomized response techniques. *Advances in Knowledge Discovery and Data Mining*, pages 637–646, 2006.

[34] L. R. Varshney. Privacy and reliability in crowdsourcing service delivery. In *SRII Global Conference (SRII), 2012 Annual*, pages 55–60. IEEE, 2012.

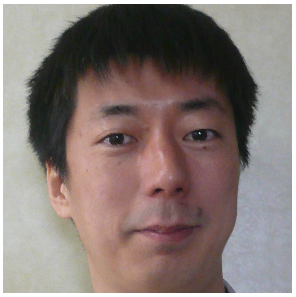
[35] Y. Wang, Y. Huang, and C. Louis. Respecting user privacy in mobile crowdsourcing. *SCIENCE*, 2(2):pp–50, 2013.

ABOUT THE AUTHORS:

Dr. Anirban Basu is a Senior Researcher at KDDI R&D Laboratories in Japan. Prior to that, he worked as a Post-doctoral Researcher at Tokai University. He is also a Visiting Research Fellow at the University of Sussex. He holds a Ph.D. in Computer Science and a Bachelor of Engineering (Hons.) in Computer Systems Engineering from the University of Sussex. His research interests are in computational trust, privacy and security and peer-to-peer networks. He is particularly active within the IFIPTM computational trust management community.

Jaideep Vaidya is an Associate Professor in the MSIS Department at Rutgers University. He received the B.E. degree in Computer Engineering from the University of Mumbai, the M.S. and Ph.D. degree in Computer Science from Purdue University. His general area of research is in data mining, data management, security, and privacy. He has published over 100 technical papers in peer-reviewed journals and conference proceedings, and has received several best paper awards from the premier conferences in data mining, databases, digital government, and informatics.

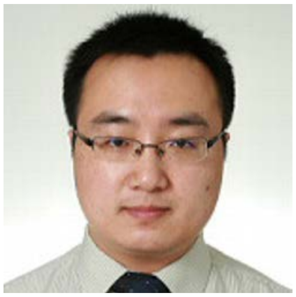
Juan Camilo Corena is a PhD candidate at Keio University in Japan. In the past, he has been an information security researcher at KDDI R&D Laboratories in Japan and an instructor and researcher at Politécnico Grancolombiano in Colombia. He holds a master and a bachelor degree in systems and computer engineering from University of the Andes (Colombia). His research interests include cryptography, erasure codes, algorithm design and network security.



Shinsaku Kiyomoto received his B.E. in engineering sciences and his M.E. in Materials Science from Tsukuba University, Japan, in 1998 and 2000, respectively. He joined KDD (now KDDI) and has been engaged in research on stream ciphers, cryptographic protocols, and mobile security. He is currently a senior researcher at the Information Security Laboratory of KDDI R&D Laboratories Inc. He was a visiting researcher of the Information Security Group, Royal Holloway University of London from 2008 to 2009. He received his doctorate in engineering from Kyushu University in 2006. He received the IEICE Young Engineer Award in 2004. He is a member of JPS and IEICE.

Steve Marsh (www.stephenmarsh.ca) is a trust scientist who works with the phenomenon of trust for computational systems. He is an Assistant Professor of Information Systems in the Faculty of Business and Information Technology, University of Ontario Institute of Technology. His PhD (University of Stirling, 1994) is a seminal work that introduced the first formalisation of the phenomenon of trust (the concept of 'Computational Trust'), and applied it to Multi Agent Systems. As a milestone in trust research, it brought together disparate disciplines and attempted to make sense of a vital phenomenon in human and artificial societies, and is still widely referenced today. Steve's current research builds extensively on this model, applying it to context-adaptive mobile devices, device comfort, protection of the user, and mobile device security and privacy.

Mr. Guibing Guo is a Ph.D student from Nanyang Technological University, Singapore. He received his bachelor degree in Computer Science and Technology and master degree in Computer Software and Theory in 2008 and 2011 respectively at Yanshan University, China. He also received another master degree in Computer Control and Automation in 2010 at Nanyang Technological University, Singapore. His current research is focused on building an effective recommender system to resolve the data sparsity and cold start problems for traditional recommender systems.



Jie Zhang is an Assistant Professor of the School of Computer Engineering, Nanyang Technological University, Singapore. He is also an Academic Fellow of the Institute of Asian Consumer Insight and Associate of the Singapore Institute of Manufacturing Technology (SIMTech). He obtained Ph.D. in Cheriton School of Computer Science from University of Waterloo, Canada, in 2009. During PhD study, he held the prestigious NSERC Alexander Graham Bell Canada Graduate Scholarship rewarded for top PhD students across Canada. He was also the recipient of the Alumni Gold Medal at the 2009 Convocation Ceremony. His research interests are in trust management and multi-agent systems.

Yutaka Miyake received the B.E. and M.E. degrees of Electrical Engineering from Keio University, Japan, in 1988 and 1990, respectively. He joined KDD (now KDDI) in 1990, and has been engaged in the research on high-speed communication protocol and secure communication system. He received the Dr. degree in engineering from the University of Electro-Communications, Japan, in 2009. He is currently a senior manager of Information Security Laboratory in KDDI R&D Laboratories Inc. He received IPSJ Convention Award in 1995 and the Meritorious Award on Radio of ARIB in 2003.