Journal of Information Processing Vol.30 694–705 (Oct. 2022)

[DOI: 10.2197/ipsjjip.30.694]

Regular Paper

An Efficient Anonymous Reputation System for Crowdsensing

Shahidatul Sadiah[1,a)] Toru Nakanishi[2,b)]

Received: December 20, 2021, Accepted: July 6, 2022

Abstract: Crowdsensing is a participatory sensing service where a server analyzes sensing data gathered from multiple users' devices. In crowdsensing, user's anonymity is desired, since the server collects their sensitive data including GPS locations and moving path. However, the anonymous submission may compromise the sensing data trust, because users may submit inappropriate data without being traced. Therefore, ARTSense (Oscar et al., Infocom 2013) has been proposed to achieve both anonymity and trust in crowdsensing. The trust of sensing data is assessed from the sensed environment, similarity check, and user reputation, which is anonymously managed on the feedback of the data trust assessment. However, in ARTSense, the user needs to wait a random time after the submission phase before requesting the reputation update, which causes communication delay. Hence, an efficient anonymous reputation system for crowdsensing is proposed to be integrated with the trust assessment of ARTSense. In the proposed system, the reputation update is anonymously completed because each user manages his/her reputation on the user side instead of the server. The validity of the reputation is ensured by a certificate and anonymously checked by zero-knowledge proofs. As a result, communication rounds are also reduced. Therefore, the proposed system achieves better efficiency without delay.

Keywords: crowdsensing, anonymity, trust assessment, reputation, zero-knowledge proofs, pairings

1. Introduction

1.1 Background

In recent years, *crowdsensing* [1] has been paid attention to and researched, due to the spread of sensor-integrated mobile smart devices such as smartphones, wearable devices, and in-vehicle devices. In crowdsensing (or known as participatory sensing), a server gathers and analyzes sensing data from lots of mobile devices. The example applications include monitoring real-time traffic patterns and pollution at the city level. The flow of a crowdsensing model starts with the user's registration to the server in the service provider. Then, a user voluntarily moves with a mobile device while sensing, and submits the sensing data to the server together with the GPS location. The server gathers the sensing data from lots of users to mine meaningful results for the applications.

In crowdsensing, the user's GPS locations are frequently submitted to the server. This concerns the user's privacy since the user's movement is tracked and recorded by the server. Therefore, in crowdsensing, the anonymity of users is desired to preserve the user's privacy, as in Refs. [2], [3], [4]. However, if the user could submit the sensing data anonymously, the service is vulnerable to a malicious user that gives inappropriate sensing data. Hence, it is needed that both anonymity and trust are satisfied.

1.2 Previous works

As the system to achieve both anonymity and trust for crowd sensing, ARTSense [5] was proposed. The name indicates the objective of the system, which is Anonymity, Reputation, and Trust in a participatory sensing. ARTSense consists of two components: Trust assessment for sensing data and an anonymous reputation system. The former provides the trust of sensing data, and the latter manages the reputation of users. In the trust assessment, the trust of submitted sensing data is evaluated by the server, based on the sensed location, time, and environment together with the user's reputation level and the similarity to the other users' sensed data for the same sensing task. In the reputation system, the reputation value is anonymously managed by the server, and it is given feedback based on the trust of the sensing data.

In the reputation system of ARTSense, to achieve the anonymity and unlinkability (i.e., infeasibility to decide the sameness of users in any two data submissions), a blind signature is used, as follows. Before the data submission, the user obtains a certificate certifying the reputation level (i.e., a rough estimate of the user's reputation value). In the sensing data submission, a blinded certificate without revealing the user's ID is also sent to show the reputation level. The server calculates the feedback value based on the trust assessment and returns the feedback certificate to the user. After that, the user re-sends the server an unblinded reputation certificate with the user's ID and the feedback certificate, and the server updates the user's reputation in the reputation database.

However, we can observe that the reputation system in ART-

[1] School of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

[2] Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima, Hiroshima 739–8527, Japan

[a)] shahidatulsadiah@utm.my

[b)] t-nakanishi@hiroshima-u.ac.jp

Sense has an efficiency problem as follows. After the data submission, the user must wait a random period to re-send the unblinded reputation certificate and the feedback certificate. If the user quickly re-sends them, the server can link the data submission to the same user's re-sending, since the number of submissions are insufficient to keep the anonymity. This implies linking the data submission to the user's ID, which compromises the anonymity. However, the waiting causes a communication delay.

1.3 Our Contributions

In this paper [*1], we propose an efficient anonymous reputation system for crowdsensing, which can be integrated into the trust assessment in ARTSense. The proposed system is based on the anonymous reputation system in Ref. [7] for P2P services such as marketplaces and adjusted to the crowdsensing. In the P2P system, the user's reputations are not kept in the server's database, and thus the user's ID is not needed in the protocols between the server and the user. The user's reputation is signed by the server as a certificate and issued to the user, where an integer range including the reputation value, which corresponds to the reputation level, is anonymously verified through a zero-knowledge proof of knowledge. Then, the reputation certificate can be updated to reflect the feedback value without revealing the reputation value. The P2P system has a complex model and mechanism to address the P2P environment. Thus, in this paper, the model and the construction of the previous P2P system are simplified to adjust the crowdsensing environment. In the proposed system, during the sensing data submission phase, the user's reputation and certificate can be updated. This means that the user does not need to wait to complete the whole process to submit a sensing data. Furthermore, this results in the reduction of the number of communication rounds. Thus, the proposed system achieves the better efficiency with no delay. On the other hand, the proposed system has a limitation that the interactions between the server and users are executed sequentially (i.e., not concurrently), as explained in Section 5.1 and Section 8, although the other functions are the same as ARTSense. One of our future works is to solve this problem.

1.4 Related Works

Here, we show other works related to the anonymous reputation system.

In Ref. [8], an anonymous reputation system is proposed to prevent the misbehaving of peer users, using an anonymous e-cash system. The reputation system model used in this work is similar to Ref. [7], which is a P2P model. The peer users anonymously award reputation points to the other peer user through e-coins which are issued by a trusted bank. Then, the ratee can deposit the e-coins to the bank, where the bank manages the database of each user and the reputations. In the deposit, to achieve the unlinkability of the ratee, the blind signature is used in the similar way to ARTSense. This causes the communication delay which is the same as ARTSense.

Meanwhile, in Refs. [9], [10], [11], anonymous reputation systems are proposed for a product review, where users rate each product or item that was purchased previously. Each user registers with the System Manager, and then the user can send an anonymous signature for rating a purchased product. To prevent an anonymous user from rating a product multiple times, this signature is linkable for rating the same product. However, these systems are not suitable for measuring an anonymous user's reputation in crowdsensing, which is the target of this paper. This is because the systems in Refs. [9], [10], [11] the anonymity (and unlinkability w.r.t. ratings for different products) of raters, but the ratees (products or items in the setting of the product review) are not anonymous and unlinkable. The goal of our system (and ARTSense) measures the ratee' reputation with keeping the unlinkability of ratees.

The anonymity, unlinkability, and unforgeability properties of anonymous reputation systems are similar to group signature schemes. In the proposed system, BBS+ signatures are used as a certificate. The BBS+ signature is introduced by Boneh, Boyen and Shacham, which was originally used in the group signature scheme [12]. The BBS+ signature allows the owner to prove the knowledge of signed messages by the zero-knowledge proofs. In the original group signatures, to a group member, the BBS+ signature is issued as the membership certificate from the group manager. The user's secret is signed in the membership certificate, and the group signature proves the secret and certificate, which ensures the membership to the group. On the other hand, in the proposed system, the BBS+ signature is used for ensuring the accumulated reputation value. Namely, in addition to the user's secret, the accumulated reputation value is signed by the BBS+ signature. In each **Show** protocol, the owner can anonymously prove the knowledge (in fact, the integer range) of the reputation value to the crowdsensing server. In addition, in each **Show** protocol, the certificate for a new reputation value reflecting the user's current sensing activity is re-issued by the server. Thus, how to use BBS+ signatures is different from the original group signature scheme [12] and is novel.

2. Preliminaries

2.1 Bilinear Maps

In this paper, we utilize the bilinear groups with a bilinear map.

(1) $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p. Here, we adopt the asymmetric setting where $\mathbb{G}_1 \neq \mathbb{G}_2$.

(2) $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ are randomly chosen generators.

(3) e is an efficiently computable bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties:

- Bilinearity: for all $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: $e(g, h) \neq 1_{\mathbb{G}_T}$ where $1_{\mathbb{G}_T}$ is an identity element of $\mathbb{G}_T$.

2.2 Assumptions

The security of our system is based on the q-SDH assumption [13] for BB signatures [13] and BBS+ signatures [12].

Definition 1 (q-SDH assumption)**.** *For all PPT algorithm $\mathcal{A}$, the*

[*1] A preliminary version of this paper was presented at 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW) [6].

probability

$$Pr[\mathcal{A}(u, v, v^{a}, \ldots, v^{(a^{q})}) = (b, v^{1/(a+b)}) \wedge b \in \mathbb{Z}_p]$$

is negligible, where $u \in_R \mathbb{G}_1$, $v \in_R \mathbb{G}_2$ and $a \in_R \mathbb{Z}_p$.

2.3 BB signatures

We use the BB signature scheme proposed in Ref. [13]. In this system, a message and the signature can be proved in the zero-knowledge by the following *PoK*.

The algorithms are described as follows.

- **BB-Setup:** Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with a prime order p and a bilinear map e. Then, select $g \overset{R}{\leftarrow} \mathbb{G}_1$ and $h \overset{R}{\leftarrow} \mathbb{G}_2$.
- **BB-KeyGen:** Choose $\gamma \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and let $w = h^{\gamma}$. The public key is $pk = w$ and the secret key is $sk = \gamma$.
- **BB-Sign:** Given a message $m \in \mathbb{Z}_p$, compute $A = g^{1/(m+\gamma)}$.
- **BB-Verify:** Given a message m and a signature A, check if $e(A, wh^{m}) = e(g, h)$.

As the security, the existential unforgeability of BB signatures against the weakly chosen message attack are proved in Ref. [13] under the q-SDH assumption. In the weakly chosen message attack model, the adversary is required to query all signatures for messages chosen by the adversary before seeing the signer's public key. Consider the following $\mathbf{Game_{wCMA}}$ between an adversary and a challenger.

Query: The adversary sends to the challenger a list of messages $m_1, \ldots, m_{q_s} \in \mathbb{Z}_p$.

Response: The challenger uses **BB-KeyGen** to generate (pk, sk) and uses **BB-Sign** to generate the signature A_i for every $m_1, \ldots, m_{q_s}$. The challenger then runs the adversary with pk and signatures $(A_1, \ldots, A_{q_s})$.

Output: Eventually, the adversary outputs a pair (m^*, A^*) and wins if a signature on m^* is not from the list returned by the challenger, and **BB-Verify** on pk, m^*, A^* is valid.

i The digital signature is existentially unforgeable against *weakly* chosen message attacks if, for all PPT adversaries, the probability that the adversary wins $\mathbf{Game_{wCMA}}$ is negligible.

2.4 BBS+ signatures

The BBS+ signature is an extension from the BB signature to sign a block of multiple messages, which is informally introduced in Ref. [12]. Moreover, the concrete construction is shown in Refs. [15], [16].

The algorithms of the BBS+ signature on a block of L messages are as follows.

- **BBS+-Setup:** Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with a prime order p and a bilinear map e. Then, select $g, g_1, \ldots, g_{L+1} \overset{R}{\leftarrow} \mathbb{G}_1$ and $h \overset{R}{\leftarrow} \mathbb{G}_2$.
- **BBS+-KeyGen:** Choose $\gamma \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and let $w = h^{\gamma}$. The public key is $pk = w$ and the secret key is $sk = \gamma$.
- **BBS+-Sign:** Given a vector $\mathcal{M}$ of messages $(m_1, \ldots, m_L) \in \mathbb{Z}_p^L$, choose $\eta, \zeta, \overset{R}{\leftarrow} \mathbb{Z}_p$, and compute $A = (g g_1^{\zeta} g_2^{m_1} \ldots g_{L+1}^{m_L})^{1/(\eta+\gamma)}$. Let the signature $\sigma = (A, \eta, \zeta)$.
- **BBS+-Verify:** For the signature $\sigma = (A, \eta, \zeta)$ and $(m_1, \ldots, m_L)$, check if $e(A, wh^{\eta}) = e(g g_1^{\zeta} g_2^{m_1} \cdots g_L^{m_{L+1}}, h)$.

As the security, the existential unforgeability of BBS+ signatures against an adaptively chosen message attack is proved in Ref. [16] under the q-SDH assumption. The attack model is stronger than the above weakly chosen message attack. Consider the following $\mathbf{Game_{CMA}}$ between an adversary and a challenger.

Setup: The challenger generates keypair (pk, sk) using **BBS+-KeyGen** and runs the adversary with pk.

Queries: The adversary can adaptively choose q_s messages $\mathbf{m}_i = (m_1, \ldots, m_L) \in \mathbb{Z}_p^L$, and request the signatures. Then, the challenger responds with the signature σ_i using **BBS+-Sign**.

Output: Finally, the adversary outputs a pair $(\mathbf{m}^*, \sigma^*)$ and wins if $(\mathbf{m}^*, \sigma^*)$ is not any of $(\mathbf{m}_1, \sigma_1), \ldots, (\mathbf{m}_{q_s}, \sigma_{q_s})$, and **BBS+-Verify** on $pk, \mathbf{m}^*, \sigma^*$ is valid.

The digital signature is existentially unforgeable against adaptively chosen message attacks if, for all PPT adversaries, the probability that the adversary wins in $\mathbf{Game_{CMA}}$ is negligible.

2.5 Commitments

We utilize a variant of Pedersen commitment to commit multiple messages. In advance, public parameters $g_1, \ldots, g_{k+1} \overset{R}{\leftarrow} \mathbb{G}_1$ are set up. The commitment c to $\mathbf{m} = (m_1, \ldots, m_k) \in \mathbb{Z}_p$ with a randomness $r \overset{R}{\leftarrow} \mathbb{Z}_p$ is computed as $c = g_1^{r} g_2^{m_1} \cdots g_{k+1}^{m_k}$. To open the commitment, $\mathbf{m} = (m_1, \ldots, m_k)$ and r are revealed to verify that c is indeed the commitment. The Pedersen commitment is (information-theoretically) perfect hiding, and is computationally binding under the discrete logarithm assumption, which is implied by the q-SDH assumption. Perfect hiding means that for all adversaries, the adversary is given a commitment of either m_1 or m_2 and cannot guess which is given. Computationally binding means that for all PPT adversaries, the adversary cannot compute $(m_1, \ldots, m_k, r)$ and $(m'_1, \ldots, m'_k, r')$ s.t. $g_1^{r} g_2^{m_1} \ldots g_{k+1}^{m_k} = g_1^{r'} g_2^{m'_1} \ldots g_{k+1}^{m'_k}$ and $(m_1, \ldots, m_k) \neq (m'_1, \ldots, m'_k)$ with a non-negligible probability.

2.6 Proof of Knowledge (*PoK*)

We use the Proof of Knowledge (*PoK*) on reputations, which is known as Σ-protocol [18]. In this paper, the following *PoK* on $\mathbb{G}_1$ and $\mathbb{G}_T$ are used, which are shown in Refs. [16], [17].

- ***PoK* of representation:** A *PoK* proving the knowledge of a representation of $C \in \mathbb{G}_1$ to the bases $g_1, g_2, \ldots, g_t \in \mathbb{G}_1$ is denoted as:

  $$PoK\{(x_1, \ldots, x_t) : C = g_1^{x_1} \cdots g_t^{x_t}\}.$$

  This can also be constructed on group $\mathbb{G}_T$.
- ***PoK* of representation with equal parts:** A *PoK* proving the knowledge of representations of $C, C' \in \mathbb{G}_1$ to the bases $g_1, g_2, \ldots, g_t \in \mathbb{G}_1$, where the representations include equal values as parts, is denoted as:

  $$PoK\{(x_1, \ldots, x_u) : C = g_{i_1}^{x_{j_1}} \cdots g_{i_u}^{x_{j_u}} \wedge C' = g_{i'_1}^{x_{j'_1}} \cdots g_{i'_{u'}}^{x_{j'_{u'}}}\},$$

  where indices $i_1, \ldots, i_u, i'_1, \ldots, i'_{u'} \in \{1, \ldots, u\}$ refer to the bases $g_1, \ldots, g_u$ and indices $j_1, \ldots, j_u, j'_1, \ldots, j'_{u'} \in \{1, \ldots, u\}$ refer to the secrets $x_1, \ldots, x_u$ (the indices are known by the verifier). By this *PoK*, the sameness of secrets such as $x_{j_1} = x_{j'_1}$ can be proved. This *PoK* can be extended for different groups $\mathbb{G}_1$ and $\mathbb{G}_T$ with the same order p, such as:

$$PoK\{(x_1, \ldots, x_u) : C = g_{i_1}^{x_{j_1}} \cdots g_{i_u}^{x_{j_u}} \wedge C' = h_{i'_1}^{x_{j'_1}} \cdots h_{i'_{u'}}^{x_{j'_{u'}}}\},$$

where $C, g_1, \ldots, g_u \in \mathbb{G}_1$ and $C', h_1, \ldots, h_u \in \mathbb{G}_T$.

The underlying interactive *PoK* consists of three moves between the prover and the verifier. At first, the prover sends the verifier an initial message. Then, the verifier returns a random challenge, and the prover sends the verifier the response message, where the verifier checks the correctness of the response. Using Fiat-Shamir heuristic, the protocol is transformed into non-interactive, where the challenge is generated by a hash function on the initial message. The non-interactive *PoK* satisfies the simulatability and the extractability in the random oracle model.

- **Simulatability**: Given the public parameters, it is able to simulate a transcript of the protocol without the secret knowledge of the prover.
- **Extractability**: Given two accepting transcripts for the same proved relation where the initial messages are the same but the challenges are distinct, we can compute the proved secret knowledge. Thus, by rewinding the prover, we can extract the knowledge (As explained in Section 5.1, the rewinding needs the sequential executions of protocols).

3. Previous System

This section reviews the previous system, ARTSense [5].

3.1 Overview of ARTSense

In the crowdsensing, users with mobile devices and a server participate, where each mobile user submits sensing data to the server. To make the crowdsensing service reliable, ARTSense mainly consists of two components: *Trust assessment* for sensing data and *anonymous reputation system*. The former evaluates the trust of sensing data, and the latter manages the reputation of users. The flow of ARTSense is shown in **Fig. 1**. When a user's device senses a data, the user sends it to the server anonymously. The data includes the user's reputation certificates showing the user's reputation level, the blinded user ID, and the context of sensing data. Upon receiving the data and the reputation level, the server assesses the trustworthiness of data in the trust assessment. Additionally, a feedback value is generated and the feedback is returned to the user to update the user's reputation.

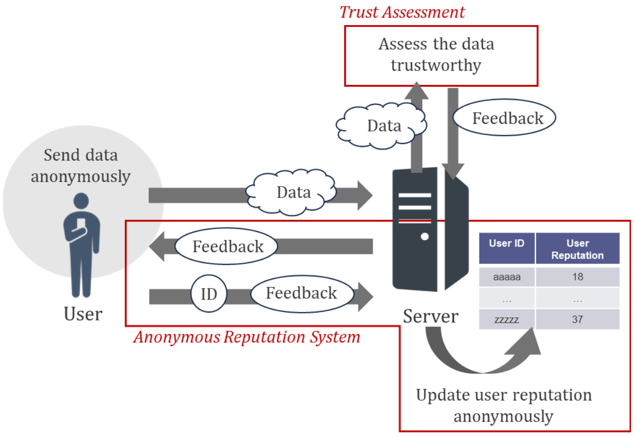

Fig. 1 Overview of ARTSense.

The reputation (and the reputation level) in ARTSense are as follows.

Definition 2. *Trust of sensing data* is a value which is the probability of the sensing data being correct, which is evaluated by the server.

Definition 3. *Reputation of user* is an integer value accumulated from feedback values. The feedback is calculated by the server simultaneously when the trust of sensing data is evaluated.

Definition 4. *Reputation level of user* is a discrete approximation of the user's concrete reputation value. It is used by the user to demonstrate his/her reputation to the server without revealing the accurate reputation value.

Aside from the ARTSense, the idea of reputation level has been used in anonymous reputation systems, for example in [8], where users with a slightly difference in reputation values are grouped in a reputation level. This is why the server can not associate a concrete reputation value to any specific user, while the server can only grasp the approximation value of the user reputation. In our work, the reputation values are split into integer ranges and the ranges are numbered by $\ell \in [1, L]$. The ℓ-th range of the reputation values shows reputation level ℓ. The user shows reputation level ℓ instead of using the reputation value directly.

3.2 Trust Assessment

The trust assessment of ARTSense is as follows. When the user submits the sensing data, the server also obtains the reputation level ℓ, which is an integer that approximately describes the user's latest reputation rep_{t-1}. Then, from the location, time, and environment information included in the sensing data submission, and the reputation level, the server calculates the base trust T_b. Besides, based on the similarity between the submitted data and the other users' sensing reports in the same sensing task, the server calculates the similarity factor *sim*. Thus, the server can calculate the final trust of the submitted data as $T_f = T_b(1 + sim)$. Furthermore, from trust T_f and the reputation level ℓ, the server calculates the feedback Δrep_t to the reputation rep_{t-1}, where the feedback is used in the following reputation system.

3.3 Anonymous Reputation System

The functionality of the anonymous reputation system which can be integrated to the trust assessment in ARTSense is as follows. In each user's data submission, the server is given the reputation level for the user's reputation from the user. The reputation level is input to the trust assessment, which generates the feedback value. Then, in the anonymous reputation system, the server updates the user's reputation by the feedback value. For the anonymous reputation system, the paper [5] considers the privacy and soundness as security. The privacy means that the sensing report does not contain any information on the user's ID, and multiple sensing reports from the same user are not linkable. The soundness means that the user cannot control the reputation of the user (only the server can determine the reputation based on the past behaviors), and the user cannot lie on the reputation level of a rough reputation value.

The construction of the anonymous reputation system in ARTSense is as follows. In this system, the server maintains the reputation database of each user's reputation which is linked to the user ID.

(1) **Issue of Reputation Certificate**: This phase is executed before submitting the sensing data. In this phase, the user sends his/her ID U_i and the task ID TID for this sensing task to the server. Using U_i, the server obtains the reputation level ℓ of the user's reputation rep_{t-1} from the reputation database. Then, the server generates two certificates $Sig(U_i|\ell|TID)$ and $Sig(\ell|TID)$, where Sig is the digital signature function by the server's secret key. The server sends the certificates to the user.

(2) **Construction of Blind ID**: In this phase, for $Sig(U_i|\ell|TID)$, the user executes blinding in a blind signature scheme (In [5], the blind RSA signature is used) to obtain the blind ID BID. After this phase, the user submits the sensing data together with BID and $Sig(\ell|TID)$ to the server.

(3) **Generation of Reputation Feedback Coupon**: After the trust assessment for the submitted data, the server generates the feedback Δrep_t to the reputation. Then, the server generates the reputation feedback coupon as $Sig(BID)|Sig(Enc(\Delta rep_t)|Sig(\ell|TID))$, where Enc is the encryption with the server's public key, and sends the coupon to the user.

(4) **Unblinding Coupon**: The user removes the blinding factor from the sent $Sig(BID)$ to obtain $Sig(Sig(U_i|\ell|TID))$ by the unblinding process of the blind signature. After the user waits a random period, the user sends $Sig(Sig(U_i|\ell|TID))|Sig(Enc(\Delta rep_t)|Sig(\ell|TID))$ to the server.

(5) **Redemption of Coupon**: The server checks the validity of signatures and the encryption. If these are valid, in the entry of U_i in the reputation database, the server updates the user's reputation rep_{t-1} to rep_t based on feedback Δrep_t.

3.4 Problems in Anonymous Reputation System

In the previous anonymous reputation system of ARTSense [5], the server manages the reputation of each user in the server's database. In this system, the anonymity of the sensing data submission is realized using a blind signature, as follows. The user sends a blinded signature BID which does not reveal U_i and TID. Then, the server sends the reputation feedback coupon to ensure the correspondence between blinded $Sig(U_i|\ell|TID)$ and Δrep_t. Finally, the server can correctly update the reputation of U_i by Δrep_t. Due to the blinding process, the communication round of the data submission and feedback coupon response is unlinkable to the communication round of sending an unblinded coupon (with the user ID) and the redemption, which leads the anonymity.

To achieve sufficient unlinkability between the rounds, the user must wait a random period to send the unblinded coupon. If the user quickly sends the unblinded coupon, the server can link the two rounds by the same user, which weakens the anonymity, since the number of submissions are insufficient. However, the waiting causes the communication delay. Another problem is that the server may link the two rounds of the same user by the value of the feedback Δrep_t. In the ARTSense paper [5], the authors suggest the variation of the feedback values is very small such as 5 to avoid this linking. Nevertheless, this may reduce the flexibility of the feedback.

4. Our Approach to Efficient Anonymous Reputation System

In this paper, we propose an efficient anonymous reputation system for crowdsensing, to which the trust assessment of ARTSense is combined. Our approach is to extend the model of the anonymous reputation system in Ref. [7] for P2P services such as market places and adjust it to the crowdsensing. In the P2P anonymous reputation system, a user (ratee) is rated by another user (rater), and additionally, a semi-honest server participates. In this system, using **Register** protocol, a user who will be a ratee registers with the server in advance, and the user is issued a certificate. The certificate ensures the user's reputation that is accumulated from past ratings. Using **Show** protocol, a user can anonymously prove his/her reputation to other users, where only the integer range including the reputation value is revealed to show the trust of the user. After a P2P interaction between the ratee user and a rater user, the server is given a rating from the rater. Finally, using **Update** protocol, the server issues the ratee an updated certificate of the reputation summed up by the new rating. The characteristic of this system is that the server does not manage the database of the reputation of each user. Instead, the reputation is managed in each user side. This is why the server does not need the user's ID. To prevent the ratee from maliciously modify the reputation, the reputation is certified by the certificate issued from the server. Furthermore, to achieve the anonymity, the update process of the certificate becomes blind, i.e., the reputation value is kept secret for the server in the certificate generation. The advantage of this system is that after the ratee is rated, the certificate is updated with no delay. By bringing this approach to crowdsensing, we can achieve an efficient reputation-update process with no delay and less number of communication rounds.

However, this previous anonymous reputation system targets P2P services. In such services, before the P2P interaction, a ratee shows his reputation (range). Then, after the P2P interaction, the ratee is rated, and the user's certificate is updated based on the new rating. Commonly, before the rating, the ratee wants to show his/her reputation for another interaction. On the other hand, after the rating, the certificate should be updated to reflect the new rating even if it is a negative rating, but a malicious user may try to show the previous reputation to discard the current negative rating. Thus, this reputation system has a mechanism to prevent the user from discarding the negative rating, as follows. **Show** protocol correspondent to a P2P interaction is indexed by integer i, which is included in the certificate. In **Show** protocol, it is checked whether the interactions for all indexes i are not rated in an anonymous way. After the i-th interaction is rated, the index i is removed from the certificate. This is why the ratee cannot discard any negative rating.

We adapt this previous P2P system to the crowdsensing environment. The crowdsensing is a simple client-server model, i.e., a central crowd sensing server communicates to each mobile user. In addition, in the model of ARTSense, the server can decide the rating (feedback) during the phase where the sensing data is submitted. This is why we can combine **Update** and **Show** protocol into a single protocol called **Show**. In **Show** protocol of our system, the user shows the reputation range (level), and the certificate is updated by the feedback based on the trust assessment in ARTSense.

In this model, since the certificate is compulsorily updated by the server, we do not need to counter the user's discarding the negative rating. Thus, the mechanism to counter it can be removed, and thus the reputation system can be simplified and become more efficient.

5. Model of Proposed System

ARTSense [5] informally shows the security requirements (privacy and soundness as shown in Section 3.3) of the anonymous reputation system, and does not define the formal security model. Therefore, we formally define the security model of an anonymous reputation system in this section.

5.1 Syntax

The proposed anonymous reputation system consists of the following algorithm and protocols. The participants of this system are the server and the users with mobile devices for the crowd-sensing.

- **Setup:** This is an algorithm for the server. The inputs are the security parameter λ and number of reputation levels L. The algorithm generates the server's public key spk and secret key ssk, and initializes the set SSet that keeps the session tags for used one-time reputation certificates.
- **Register:** This is an interactive protocol between a user and the server, where the user is registered with the server. The common input is spk and the server's input is ssk. The user's output of this protocol is $cert_0$ that is the user's initial one-time reputation certificate certifying the initial reputation $rep_0 = 0$.
- **Show:** This is an interactive protocol between the user and the server, where the user convinces the server of his/her reputation level (the integer range in which the reputation is included) and the reputation is updated. The common input are spk, the reputation level ℓ, and feedback Δrep_t, which is derived from the assessment for the sensing report and the reputation level (cf. Section 3.3). The user's input is his latest $cert_{t-1}$ certifying the reputation rep_{t-1}. The server's input is SSet. If the server judges that rep_{t-1} is not included in the integer range of ℓ, the user is rejected. Otherwise, the user's output is one-time fresh reputation certificate $cert_t$ certifying the updated reputation rep_t added by the feedback Δrep_t. The server's output is the updated SSet. Set SSet consists of session tags included in the past used certificates to detect the double use of a certificate. If the double use is detected, this protocol is aborted.

We assume the interactions between the server and users are executed sequentially (i.e., not concurrently). Namely, while the server executes a **Register** protocol or a **Show** protocol with a user, the server does not execute another **Register** protocol or another **Show** protocol with a user concurrently. After a **Register**/**Show** protocol finishes, the server can start the next **Register**/**Show** protocol. This limitation is caused by the used PoK which needs rewinding for extracting the secrets, and is also introduced in the group signature setting [20]. As a result, the overall processing time depends on the number of active users. But, as mentioned in Ref. [20], this limitation may be solved using an extractable commitment. The application of the extractable commitment to our approach is our future work.

In this model, for each sensing data submission, only **Show** protocol is executed, where any delay is not needed.

5.2 Security Model

At first, we informally describe the security requirement, which consists of the *misauthentication resistance* and the *anonymity*, which are the same as in an anonymous reputation system in ARTSense. The misauthentication resistance means that a user cannot maliciously accumulate the reputation value and cannot prove an inappropriate range. The anonymity means that any information to guess who is the communicating user is not leaked beyond the sequence of transactions with the inputs (i.e., the reputation level ℓ and the feedback Δrep_t in **Show**).

Formally, we adopt the simulation-based definition of security, which is derived from Ref. [21] for the blacklistable anonymous credential system PEREA (This model is also adopted in Refs. [22], [23]). The model of Ref. [21] is based on *real world* and *ideal world*. In the real world, multiple players (users and a server in our setting) participate, and communicate via cryptographic protocols. Furthermore, there is an adversary $\mathcal{A}$ that controls dishonest players. Furthermore, there is an environment $\mathcal{E}$ that provides inputs to players, instructs them to do each transaction in the system, and receives their outputs. $\mathcal{E}$ also interacts with $\mathcal{A}$. In the ideal world, the same players participate and there is $\mathcal{A}$ controling dishonest players, but the communication is not direct, but via a trusted party $\mathcal{T}$ handling the communication between players, where $\mathcal{T}$ achieves the functionalities instead of the cryptographic protocols. The environment $\mathcal{E}$ also provides inputs to players, instructs them to do each transaction in the system, and receives their outputs. $\mathcal{E}$ also interacts with $\mathcal{A}$.

The real world and ideal world for the proposed system support the following transactions. In this model, honest and dishonest players are fixed in advance. All communications with $\mathcal{T}$ are not anonymous. It is also assumed that communication between honest parties is not viewed by the real-world adversary and that when the real-world adversary receives a message, it does not know the origin of the message. In each world, all transactions are scheduled according to $\mathcal{E}$'s wishes, where Setup transaction is initially invoked once.

- Setup:
  - *Real world*: The server runs **Setup** algorithm to generate spk, ssk. spk is available to all players in the system.
  - *Ideal world*: The trusted party $\mathcal{T}$ initializes database $\mathcal{D}$ with entries (i, sum_i) of a user's ID and the user's reputation sum.
- $\mathsf{Register}(\mathsf{tid}, i)$: On a unique transaction ID tid, $\mathcal{E}$ instructs

user i to register with the server. This transaction is not anonymous.

- *Real world*: User i sends Register request to the server, who responds accept/reject, as follows. If user i has executed Register transaction in the past, the honest server rejects this request by sending reject. Otherwise, the server sends accept to the user, and they execute **Register** protocol. User i and the server individually output the outcome of this transaction (tid, success/failure), indicating whether this transaction is correctly finished, to $\mathcal{E}$.
- *Ideal world*: User i sends Register request to $\mathcal{T}$. $\mathcal{T}$ forwards the request to the server. $\mathcal{T}$ also checks if user i has executed Register transaction in the past, and the result is notified to the server. The server returns accept/reject to $\mathcal{T}$, which is forwarded to the user. $\mathcal{T}$ initializes the entry (i, sum_i) of $\mathcal{D}$, where $\mathsf{sum}_i = 0$. User i and the server individually output the outcome of this transaction (tid, success/failure) to $\mathcal{E}$.

- Show(tid, i): On a unique transaction ID tid, $\mathcal{E}$ instructs user i to the following process of **Show** to the server.
  - *Real world*: User i sends Show request together with the reputation level ℓ and feedback Δrep_t to the server. Then, the user conducts **Show** protocol on $\ell, \Delta rep_t$ with the server. User i and the server individually output the outcome of this transaction (tid, success/failure) to $\mathcal{E}$.
  - *Ideal world*: User i sends Show request together with the reputation level ℓ and feedback Δrep_t to $\mathcal{T}$, which are forwarded to the server. Then, $\mathcal{T}$ finds the entry (i, sum_i) of $\mathcal{D}$, and checks if sum_i is included in the integer range of the reputation level ℓ. The result is sent to the server. The server returns accept/reject to $\mathcal{T}$, which is forwarded to user i. If the server returns accept, $\mathcal{T}$ updates the entry (i, sum_i) of $\mathcal{D}$, where $\mathsf{sum}_i = \mathsf{sum}_i + \Delta rep_t$. User i and the server individually output the outcome of this transaction (tid, success/failure) to $\mathcal{E}$.

Then, as well as Ref. [21], the security is defined as follows. The proposed system is secure, if for every real-world PPT adversary $\mathcal{A}$, every PPT environment $\mathcal{E}$, there is an ideal-world adversary $\mathcal{S}$ which is a PPT simulator with black-box access to $\mathcal{A}$, $\mathcal{E}$ cannot tell whether it is running in the real world with $\mathcal{A}$ and it is running in the ideal world with $\mathcal{S}$.

Definition 5. *Let λ be a security parameter. Let $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ (resp., $\mathbf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)$) be the probability that $\mathcal{E}$ outputs 1 when in the real world (resp., ideal world) with adversary $\mathcal{A}$ (resp., adversary $\mathcal{S}$ with black-box access to $\mathcal{A}$). The proposed system is secure if for all PPT algorithms $\mathcal{E}, \mathcal{A}$, there exists a PPT algorithm $\mathcal{S}$ s.t. $|\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathbf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)|$ is negligible.*

As mentioned in Ref. [21], this definition captures both the misauthentication resistance and the anonymity. In the case that an adversary controls a subset of users but does not control the server, for the honest server, the misauthentication, i.e., a user maliciously accumulates the reputation value and proves an inappropriate range, should be protected. In this definition, $\mathcal{T}$ correctly calculates the user's reputation value and checks the range in the ideal world, and thus the indistinguishability between the ideal world and the real world implies that the adversary-controlled user cannot succeed the misauthentication in the real world. On the other hand, in the case that an adversary controls a subset of users and the server, the anonymity of honest users to the adversary-controlled server should be protected. In this definition, due to $\mathcal{T}$'s forwarding, the adversary cannot know any information beyond the sequence of transactions with the inputs in the ideal world, and thus the indistinguishability between the ideal world and the real world implies the anonymity in the real world.

6. Proposed Reputation System for Crowdsensing

6.1 Outline of Proposed System

Before describing the construction of the proposed system, we show the outline, and mention the difference from the underlying system.

- **Setup:** In this algorithm, the server generates key pairs of BB signatures and BBS+ signatures. Then, the server computes the BB signature on every value in the integer range of reputation level $1 \leq \ell \leq L$ as the reputation level certificate.
- **Register:** The server issues a registering user an initial reputation certificate $cert_0$, which is a BBS+ signature on the user's secret x, a tag S_0, and the initial reputation $rep_0 = 0$.
- **Show:** The user's input is his/her latest certificate $cert_{t-1}$ which contains secret x, last used session tag S_{t-1}, and latest cumulative reputation value rep_{t-1}, where t indicates the session number. At first, the user proves the reputation level ℓ. This is performed by the PoK proving the BBS+ signature in $cert_{t-1}$ for rep_{t-1} and proving the BB signature for the reputation level ℓ and the value rep_{t-1}. In addition, the server checks if the certificate $cert_{t-1}$ with the session tag S_{t-1} has been used in the past. Thus, the user needs to send the session tag S_{t-1} to the server. If the user is being honest, this session tag S_{t-1} is not found in the set SSet. Otherwise, the user is trying to update the reputation value rep_{t-1} twice, which then the server must abort the show protocol. Next, for the feedback Δrep_t, the server blindly updates the user reputation as $rep_t = rep_{t-1} + \Delta rep_t$ via the commitment of rep_{t-1}. Finally, the server generates a new BBS+ signature as the updated certificate $cert_t$ for rep_t to send to the user, where the tag S_t is committed and signed, which will be revealed in the next **Show**.

The difference from the P2P anonymous reputation system [7] is as follows. As mentioned in Section 4, since **Update** is integrated to **Show**, the mechanism to avoid the user's discarding negative feedbacks is removed and simplified. As the mechanism, an accumulator was used, and a structure-preserving signature was used to sign the accumulator of a group element in the certificate and to prove the knowledge. However, in the proposed system, only $\mathbb{Z}_p$ elements are signed, and thus the more efficient BBS+ signature is used. Because of this, commitments used to blindly sign messages are modified and simplified to a vector-type commitment used in BBS+ signatures. In addition, in Ref. [7], the PoK for the BB signature needs three proved relations. On the other hand, in Ref. [24], the PoK using only one relation is shown. Thus, in this paper, using this technique in

Ref. [24], the *PoK* is optimized.

6.2 Proposed Construction

Setup: In this algorithm, the server generates key pairs of public and private keys for BB signatures and BBS+ signatures, and issues the certificates (BB signatures) for all integer ranges of reputation level ℓ for $1 \leq \ell \leq L$, where L is the maximum number of the reputation levels.

(1) The server selects bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, and a bilinear map e with a prime order $p > 2^{\lambda}$, where λ is the given security parameter. Then, the server selects $g_0, g_1, g_2, g_3, g_4, f_0, f_1 \overset{R}{\leftarrow} \mathbb{G}_1$, $h_0 \overset{R}{\leftarrow} \mathbb{G}_2$. For all $1 \leq \ell \leq L$, the server chooses $\gamma_{0,\ell} \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and computes $w_{0,\ell} = h_0^{\gamma_{0,\ell}}$, where $\gamma_{0,\ell}$ is the secret key for the BB signature proving the reputation level ℓ. Then, as the key pairs of BBS+ signatures, the server chooses $\gamma_1 \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and computes $w_1 = h_0^{\gamma_1}$, where γ_1 is the secret key for the user's reputation certificate.

(2) For all $1 \leq \ell \leq L$, the server generates the reputation level certificate $A_{\ell,R_{\ell,k}} = f_0^{1/(\gamma_{0,\ell}+R_{\ell,k})}$ (BB signature) for every value $R_{\ell,k}$ in the ℓ-th integer range indicating reputation level ℓ (cf. Section 3.1), where K_ℓ is the number of the values in the ℓ-th integer range.

(3) The server initializes set SSet as empty, and outputs the public key

$$spk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \{w_{0,\ell}\}_{\ell=1}^{L}, w_1, g_0, g_1, g_2, g_3, g_4, f_0, f_1, h_0, \{\{A_{\ell,k}\}_{k=1}^{K_\ell}\}_{\ell=1}^{L}),$$

and the secret key $ssk = \gamma_1$.

Register: This is a protocol between the user **U** and the server **S**. In this protocol, the server issues an initial reputation certificate $cert_0$ for the user. The common input is spk, and the server's input is ssk.

(1) **[U]:** Select secret $x \overset{R}{\leftarrow} \mathbb{Z}_p^*$, a reputation certificate's tag $S_0 \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and a random factor $\zeta'_0 \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and compute the commitment to the vector of messages (x, S_0) to be signed by $C'_{m,0} = g_1^{\zeta'_0} g_2^x g_3^{S_0}$. Then, prove to the server that $C'_{m,0}$ is correctly formed by the following *PoK*.

$$PoK\{(\zeta'_0, x, S_0) : C'_{m,0} = g_1^{\zeta'_0} g_2^x g_3^{S_0}\} \qquad (1)$$

(2) **[S]:** Set the initial reputation as $rep_0 = 0$, and choose random factors $\zeta''_0, \eta_0 \overset{R}{\leftarrow} \mathbb{Z}_p^*$. Then, using the secret key γ_1 of BBS+ signatures, sign the vector of messages (x, S_0, rep_0) as $B_0 = (g_0 g_1^{\zeta''_0} C'_{m,0} g_4^{rep_0})^{1/\gamma_1+\eta_0}$, and send back $\tilde{\sigma}'_0 = (B_0, \eta_0, \zeta''_0)$ to the user.

(3) **[U]:** Set $C_{m,0} = C'_{m,0} g_4^{rep_0}$ for $rep_0 = 0$, compute $\zeta_0 = \zeta'_0 + \zeta''_0$, and set the BBS+ signature on the messages (x, S_0, rep_0) as $\tilde{\sigma}_0 = (B_0, \eta_0, \zeta_0)$, where $B_0 = (g_0 g_1^{\zeta_0} g_2^x g_3^{S_0} g_4^{rep_0})^{1/\gamma_1+\eta_0}$. Output $cert_0 = (x, rep_0, \tilde{\sigma}_0, S_0, C_{m,0})$.

Show: In this protocol, the user's reputation level ℓ is proved on the certificate $cert_{t-1}$, the certificate is updated by adding the feedback Δrep_t to the previous reputation rep_{t-1}, and then the updated reputation certificate $cert_t$ is issued. The user's inputs are $cert_{t-1} = (x, rep_{t-1}, \tilde{\sigma}_{t-1}, S_{t-1}, C_{m,t-1})$, where $\tilde{\sigma}_{t-1} = (B_{t-1}, \eta_{t-1}, \zeta_{t-1})$. Here, t indicates the session number.

(1) **[U]:** From spk, retrieve a reputation level certificate $A_{\ell,rep_{t-1}}$ such that his current reputation rep_{t-1} is in ℓ-th range. Choose $r_{A_\ell} \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute the commitment $C_{A_\ell} = A_{\ell,rep_{t-1}} f_1^{r_{A_\ell}}$ and $\rho = r_{A_\ell} \cdot rep_{t-1}$. Then, choose $\hat{\zeta} \overset{R}{\leftarrow} \mathbb{Z}_p$, compute the commitment $C_{B_{t-1}} = B_{t-1} g_1^{\hat{\zeta}}$, and set $\theta = \zeta_{t-1} + \hat{\zeta}\eta_{t-1}$. Choose $\zeta'_t \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and $S_t \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and compute $C'_{m,t} = g_1^{\zeta'_t} g_2^x g_3^{S_t} g_4^{rep_{t-1}}$ as the commitment to the vector of (x, S_t, rep_{t-1}). Send $C_{A_\ell}, C_{B_{t-1}}, C'_{m,t}, S_{t-1}$ to the server, and using the following *PoK*, prove that the reputation rep_{t-1} is in the ℓ-th range, $cert_{t-1}$ is valid, and $C'_{m,t}$ is correct.

$$\begin{aligned}
&PoK\{(r_{A_\ell}, rep_{t-1}, \rho, \theta, x, \hat{\zeta}, \eta_{t-1}, \zeta'_t, S_t) : \\
&\quad e(C_{A_\ell}, w_{0,\ell}) \cdot e(f_0, h_0)^{-1} = e(f_1, w_{0,\ell})^{r_{A_\ell}} \\
&\quad \cdot e(C_{A_\ell}, h_0)^{-rep_{t-1}} \cdot e(f_1, h_0)^{\rho} \\
&\quad \wedge e(C_{B_{t-1}}, w_1) \cdot e(g_0, h_0)^{-1} \cdot e(g_3, h_0)^{-S_{t-1}} \\
&\quad = e(g_1, h_0)^{\theta} \cdot e(g_2, h_0)^{x} \cdot e(g_4, h_0)^{rep_{t-1}} \\
&\quad \cdot e(g_1, w_1)^{\hat{\zeta}} \cdot e(C_{B_{t-1}}, h_0)^{-\eta_{t-1}} \\
&\quad \wedge C'_{m,t} = g_1^{\zeta'_t} g_2^x g_3^{S_t} g_4^{rep_{t-1}}\}.
\end{aligned}$$

(2) **[S]:** To check the freshness of the proved certificate, check if $S_{t-1} \in$ SSet. If it is true, abort. Otherwise, add tag S_{t-1} in set SSet. Verify the *PoK*. If it is invalid, abort. Otherwise, update the user's reputation certificate to $cert_t$, where Δrep_t is added to commitment as $C_{m,t} = C'_{m,t} g_4^{\Delta rep_t}$ and it is signed as $B_t = (g_0 g_1^{\zeta''_t} C_{m,t})^{1/\gamma_1+\eta_t} = (g_0 g_1^{\zeta''_t} g_1^{\zeta'_t} g_2^x g_3^{S_t} g_4^{rep_{t-1}} g_4^{\Delta rep_t})^{1/\gamma_1+\eta_t}$ for $\zeta''_t, \eta_t \overset{R}{\leftarrow} \mathbb{Z}_p^*$. Then, send back $\tilde{\sigma}'_t = (B_t, \eta_t, \zeta''_t)$ to the user. Output the updated SSet.

(3) **[U]:** Compute $\zeta_t = \zeta'_t + \zeta''_t$, $rep_t = rep_{t-1} + \Delta rep_t$ and set the signature on the vector of messages (x, S_t, rep_t) as $\tilde{\sigma}_t = (B_t, \eta_t, \zeta_t)$, where $B_t = (g_0 g_1^{\zeta_t} g_2^x g_3^{S_t} g_4^{rep_t})^{1/\gamma_1+\eta_t}$. Output $cert_t = (x, rep_t, \tilde{\sigma}_t, S_t, C_{m,t})$.

7. Security

Before considering the security of the proposed system, we show the following lemma.

Lemma 1. *The PoK in* **Show** *proves the knowledge of* $A'_{\ell,rep_{t-1}}$, ξ, $rep_{t-1}, B_{t-1}, \zeta_{t-1}, \eta_{t-1}, x$ *such that*

$$\begin{aligned}
&A'_{\ell,rep_{t-1}} = (f_0 f_1^{\xi})^{1/(\gamma_{0,\ell}+rep_{t-1})}, \\
&B_{t-1} = (g_0 g_1^{\zeta_{t-1}} g_2^x g_3^{S_{t-1}} g_4^{rep_{t-1}})^{1/\gamma_1+\eta_{t-1}}.
\end{aligned}$$

Proof. From the *PoK*, we can extract r_{A_ℓ}, rep_{t-1}, ρ, θ, x, $\hat{\zeta}$, and η_{t-1} such that

$$\begin{aligned}
e(C_{A_\ell}, w_{0,\ell}) \cdot e(f_0, h_0)^{-1} &= e(f_1, w_{0,\ell})^{r_{A_\ell}} \\
&\quad \cdot e(C_{A_\ell}, h_0)^{-rep_{t-1}} \cdot e(f_1, h_0)^{\rho}, \qquad (2)
\end{aligned}$$

$$\begin{aligned}
&e(C_{B_{t-1}}, w_1) \cdot e(g_0, h_0)^{-1} \cdot e(g_3, h_0)^{-S_{t-1}} \\
&\quad = e(g_1, h_0)^{\theta} \cdot e(g_2, h_0)^{x} \cdot e(g_4, h_0)^{rep_{t-1}} \\
&\qquad \cdot e(g_1, w_1)^{\hat{\zeta}} \cdot e(C_{B_{t-1}}, h_0)^{-\eta_{t-1}}. \qquad (3)
\end{aligned}$$

Then, from Eq. (2), we have the following transformations.

$$\begin{aligned}
&e(C_{A_\ell}, w_{0,\ell}) \cdot e(C_{A_\ell}, h_0)^{rep_{t-1}} \cdot e(f_1, w_{0,\ell})^{-r_{A_\ell}} = e(f_0, h_0) e(f_1, h_0)^{\rho} \\
&e(C_{A_\ell}, w_{0,\ell} h_0^{rep_{t-1}}) \cdot e(f_1, w_{0,\ell})^{-r_{A_\ell}} = e(f_0 f_1^{\rho}, h_0)
\end{aligned}$$

$$
\begin{aligned}
&e(C_{A_\ell}, w_{0,\ell}h_0^{rep_{t-1}}) \cdot e(f_1, w_{0,\ell})^{-r_{A_\ell}} e(f_1, h_0)^{-r_{A_\ell} rep_{t-1}}\\
&\quad = e(f_0 f_1^{\rho}, h_0) e(f_1, h_0)^{-r_{A_\ell} rep_{t-1}}\\
&e(C_{A_\ell}, w_{0,\ell}h_0^{rep_{t-1}}) \cdot e(f_1^{-r_{A_\ell}}, w_{0,\ell}h_0^{rep_{t-1}})\\
&\quad = e(f_0 f_1^{\rho}, h_0) e(f_1^{-r_{A_\ell} rep_{t-1}}, h_0)\\
&e(C_{A_\ell} f_1^{-r_{A_\ell}}, w_{0,\ell}h_0^{rep_{t-1}}) = e(f_0 f_1^{\rho - r_{A_\ell} rep_{t-1}}, h_0)
\end{aligned}
$$

Thus, by setting $A'_{\ell,rep_{t-1}} = C_{A_\ell} f_1^{-r_{A_\ell}}$ and $\xi = \rho - r_{A_\ell} rep_{t-1}$, we obtain $e(A'_{\ell,rep_{t-1}}, w_{0,\ell}h_0^{rep_{t-1}}) = e(f_0 f_1^{\xi}, h_0)$, which implies $A'_{\ell,rep_{t-1}} = (f_0 f_1^{\xi})^{1/(\gamma_{0,\ell}+rep_{t-1})}$.

Next, from Eq. (3),

$$
\begin{aligned}
&e(C_{B_{t-1}}, w_1) \cdot e(g_1, w_1)^{-\hat{\zeta}} \cdot e(C_{B_{t-1}}, h_0)^{\eta_{t-1}}\\
&\quad = e(g_0, h_0) \cdot e(g_1, h_0)^{\theta} \cdot e(g_2, h_0)^{x} \cdot e(g_3, h_0)^{S_{t-1}} \cdot e(g_4, h_0)^{rep_{t-1}}\\
&e(C_{B_{t-1}}, w_1 h_0^{\eta_{t-1}}) \cdot e(g_1, w_1)^{-\hat{\zeta}} = e(g_0 g_1^{\theta} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}}, h_0)\\
&e(C_{B_{t-1}}, w_1 h_0^{\eta_{t-1}}) \cdot e(g_1, w_1)^{-\hat{\zeta}} \cdot e(g_1, h_0)^{-\hat{\zeta}\eta_{t-1}}\\
&\quad = e(g_0 g_1^{\theta} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}}, h_0) \cdot e(g_1, h_0)^{-\hat{\zeta}\eta_{t-1}}\\
&e(C_{B_{t-1}}, w_1 h_0^{\eta_{t-1}}) \cdot e(g_1^{-\hat{\zeta}}, w_1 h_0^{\eta_{t-1}})\\
&\quad = e(g_0 g_1^{\theta} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}}, h_0) \cdot e(g_1^{-\hat{\zeta}\eta_{t-1}}, h_0)\\
&e(C_{B_{t-1}} g_1^{-\hat{\zeta}}, w_1 h_0^{\eta_{t-1}}) = e(g_0 g_1^{\theta - \hat{\zeta}\eta_{t-1}} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}}, h_0)
\end{aligned}
$$

Thus, by setting $B_{t-1} = C_{B_{t-1}} g_1^{-\hat{\zeta}}$ and $\zeta_{t-1} = \theta - \hat{\zeta}\eta_{t-1}$, we obtain $e(B_{t-1}, w_1 h_0^{\eta_{t-1}}) = e(g_0 g_1^{\zeta_{t-1}} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}}, h_0)$, which implies $B_{t-1} = (g_0 g_1^{\zeta_{t-1}} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}})^{1/\gamma_1+\eta_{t-1}}$. □

This lemma shows that the user proves the knowledge of $A'_{\ell,rep_{t-1}}$ s.t. $A'_{\ell,rep_{t-1}} = (f_0 f_1^{\xi})^{1/(\gamma_{0,\ell}+rep_{t-1})}$. This $A'_{\ell,rep_{t-1}}$ is a variant of BB signature, and not the same as a BB signature on rep_{t-1}, due to the part f_1^{ξ}. However, as proved in Ref. [24], forging the variant can be reduced to forging the BB signature, which is shown in the proof of **Theorem 1**.

Based on **Definition 5**, we will show the following theorem.

Theorem 1. *The proposed system is secure under the simulatability and the extractability of the PoK, the existential unforgeability of BB signatures against the weakly chosen message attack, the existential unforgeability of BBS+ signatures against the adaptively chosen message attack, and the perfect hiding and the computationally binding of the commitments in the random oracle model.*

Similarly to the original proofs in Ref. [21], for any real-world adversary $\mathcal{A}$ and environment $\mathcal{E}$, we will construct an ideal-world adversary $\mathcal{S}$ as a simulator with black-box access to $\mathcal{A}$ s.t. $\mathcal{E}$ cannot determine whether it is interacting with $\mathcal{A}$ in the real world or $\mathcal{S}$ in the ideal world. We construct different simulators, in case that $\mathcal{A}$ controls only a subset of users, and in case that $\mathcal{A}$ controls a subset of users and the server, as in Ref. [21]. Thus, we will show the following two lemmas corresponding to the cases, and by combining the lemmas, we can conclude the above theorem.

Lemma 2. *For any PPT environment $\mathcal{E}$ and any PPT real-world adversary $\mathcal{A}$ controlling a subset of users, there exists a PPT ideal-world adversary (simulator) $\mathcal{S}$ s.t. $|\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathbf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)|$ is negligible in the random oracle model.*

Proof. In this case, we construct $\mathcal{S}$ which represents dishonest users controlled by $\mathcal{A}$ to the trusted party $\mathcal{T}$ and $\mathcal{E}$, and on the other hand, $\mathcal{S}$ represents the honest users and server to $\mathcal{A}$ in each transaction in the ideal world. In the simulation with $\mathcal{A}$, $\mathcal{S}$ can extract the secret in each *PoK*, by rewinding $\mathcal{A}$. This extraction on rewinding is used in the security proof of group signatures (e.g., [20]) and so on. This rewinding requires the sequential executions of proofs of knowledge (In the setting of group signatures, join protocols are sequentially executed), and concurrent executions are not permitted. In this paper, we also assume that executions of **Register** and **Show** protocols between users and the server are sequential, and the concurrent executions are not permitted. As discussed in Ref. [20], using an extractable commitment, rewinding can be excluded. The application of this to our approach is our future work.

The simulation is as follows.

- Setup:
  - *Representing honest server to $\mathcal{A}$*: $\mathcal{S}$ runs **Setup** algorithm to generate spk, ssk. spk is sent to $\mathcal{A}$. $\mathcal{S}$ initializes a database $\tilde{\mathcal{D}}$ with entries (i, x_i) of a user's ID and the user's secret key.
- Register:
  - *Representing dishonest user i controlled by $\mathcal{A}$ to $\mathcal{T}$ / honest server to $\mathcal{A}$*: On the request on tid from $\mathcal{A}$ on behalf of user i, if user i has executed Register transaction in the past, $\mathcal{S}$ sends reject to $\mathcal{A}$. Otherwise, $\mathcal{S}$ as the server executes **Register** protocol with $\mathcal{A}$ as the user, where $\mathcal{S}$ uses ssk. In the protocol, using the extractor for the PoK in **Register** protocol, $\mathcal{S}$ extracts x (denoted as x_i). If the extraction fails, abort. Otherwise, $\mathcal{S}$ sends the request to $\mathcal{T}$ on behalf of user i. $\mathcal{S}$ forwards the protocol outcome from $\mathcal{A}$ on behalf of user i to $\mathcal{E}$. $\mathcal{S}$ stores entry (i, x_i) in $\tilde{\mathcal{D}}$.
- Show:
  - *Representing dishonest user i controlled by $\mathcal{A}$ to $\mathcal{T}$ / honest server to $\mathcal{A}$*: On the request on tid, ℓ, and Δrep_t from $\mathcal{A}$ on behalf of user i, $\mathcal{S}$ as the server executes **Show** protocol with $\mathcal{A}$ as the user, where $\mathcal{S}$ uses ssk. However, $\mathcal{A}$ may use $cert_t$ of another controlled user $\hat{\imath}$. To locate $\hat{\imath}$, $\mathcal{S}$ extracts x (denoted as $x_{\hat{\imath}}$), together with other secrets ($A'_{\ell,rep_{t-1}}$, ξ, $rep_{t-1}, B_{t-1}, \zeta_{t-1}, \eta_{t-1}$) from the PoK of **Show** protocol, using the extractor of the PoK. If the extraction fails, abort. Next, $\mathcal{S}$ finds entry $(\hat{\imath}, x_{\hat{\imath}})$ in $\tilde{\mathcal{D}}$. $\mathcal{S}$ on behalf of user $\hat{\imath}$ sends **Show** request on $\ell, \Delta rep_t$ to $\mathcal{T}$. Finally, the outcome of this transaction (tid, success/failure) from $\mathcal{A}$ on behalf of user i is forwarded to $\mathcal{E}$.

In this simulation, consider the following *bad* events.

E1: In a Register transaction, $\mathcal{S}$ fails to extract x from the PoK.

E2: In a successful Show transaction (successful Show transaction means that the dishonest user and the honest server outputs success), $\mathcal{S}$ fails to extract x from the PoK.

E3: In a successful Show transaction, the extracted BBS+ signature $\tilde{\sigma}_{t-1} = (B_{t-1}, \eta_{t-1}, \zeta_{t-1})$ has never been issued from $\mathcal{S}$. This case is when $\mathcal{A}$ forges the BBS+ signature.

E4: In a successful Show transaction, for the extracted variant of BB signature $A'_{\ell,rep_{t-1}}$ on rep_{t-1}, any BB signature on rep_{t-1} has never been issued by $\mathcal{S}$. This case is when $\mathcal{A}$ forges the BB signature.

E5: In successful Register/Show transactions, the committed values extracted from the PoKs compromise the binding property in the commitment scheme.

However, the probability that **E1** event or **E2** event happens is negligible, due to the extractor in the PoK, as well as Ref. [21]. If one of **E3**–**E5** events occurs, the corresponding BB signature, BBS+ signature, or the commitment is compromised by simulating the game with $\mathcal{A}$, which contradicts the security of the primitive. The reduction to each attack is as follows.

$\mathcal{A}_{BB}$**:** We construct $\mathcal{A}_{BB}$ which corresponds to the case that $\mathcal{A}$ forges a BB signature $A_{\tilde{\ell},rep_{t-1}}$, where $\mathcal{A}_{BB}$ conducts the weakly chosen message attack, as follows. At first, $\mathcal{A}_{BB}$ selects $\tilde{\ell} \in_R [1, L]$ and requests $\{R_{\tilde{\ell},k}\}_{1\leq k\leq K_{\tilde{\ell}}}$ as messages to be signed, where $K_{\tilde{\ell}}$ is the number of values in the $\tilde{\ell}$-th range. Then, $\mathcal{A}_{BB}$ is given the public key of BB signature, $pk_{BB} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, w, g, h)$, and the BB signatures $A_{\tilde{\ell},k}$ for the requested messages. Using the given values, $\mathcal{A}_{BB}$ prepares spk in the proposed system, as follows. Set $w_{0,\tilde{\ell}} = w$, $f_0 = g$, $h_0 = h$. Then, choose $\varpi \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and compute $f_1 = f_0^{\varpi}$. Choose and compute other parameters in spk as in the real **Setup**. Then, start the simulation with $\mathcal{A}$, where spk is given to $\mathcal{A}$ in Setup. Note that keys of BB signatures for level ℓ s.t. $\ell \neq \tilde{\ell}$ and the signatures $A_{\ell,k}$ are generated as in the real **Setup**. In Register and Show transactions, the response to $\mathcal{A}$ does not require the secret key for BB signatures, and thus these are responded as in the real **Register** and **Show** protocols. Then, $\mathcal{A}$ in this case outputs a PoK in a successful Show transaction, where a variant of BB signature $A_{\tilde{\ell},rep_{t-1}} = (f_0 f_1^{\xi})^{1/(\gamma_{0,\tilde{\ell}}+rep_{t-1})}$ on message rep_{t-1} in the ℓ-th level is extracted as shown in **Lemma 1**, but any BB signature on rep_{t-1} has never been issued. Then, if $\ell \neq \tilde{\ell}$, abort. For a random $\tilde{\ell}$, the probability that it is not aborted is at least $1/L$, i.e., non-negligible. Otherwise, if $\varpi\xi + 1 = 0 \pmod p$, abort. If $\varpi\xi + 1 = 0 \pmod p$, we can construct an adversary to compute the discrete log ϖ of f_1 to f_0 using $\mathcal{A}$ with $\varpi\xi + 1 = 0 \pmod p$. Thus, the probability of aborting is negligible. Then, obtain the original BB signature $A_{\tilde{\ell},rep_{t-1}}$ from $A'_{\tilde{\ell},rep_{t-1}}$, by

$$\begin{aligned} A_{\tilde{\ell},rep_{t-1}} &= A'^{\,1/(\varpi\xi+1)}_{\tilde{\ell},rep_{t-1}} && (4)\\ &= (f_1^{\xi} f_0)^{1/((\gamma_{0,\tilde{\ell}}+rep_{t-1})(\varpi\xi+1))} && (5)\\ &= (f_0^{\varpi\xi+1})^{1/((\gamma_{0,\tilde{\ell}}+rep_{t-1})(\varpi\xi+1))} && (6)\\ &= f_0^{1/(\gamma_{0,\tilde{\ell}}+rep_{t-1})}, && (7) \end{aligned}$$

and output $(rep_{t-1}, A_{\tilde{\ell},rep_{t-1}})$. This means the forgery of a BB signature.

$\mathcal{A}_{BBS+}$**:** We construct $\mathcal{A}_{BBS+}$ which is corresponds to the case that $\mathcal{A}$ forges a BBS+ signature B_{t-1}. $\mathcal{A}_{BBS+}$ is given the public key $pk_{BBS+} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_0, g_1, g_2, g_3, g_4, h_0, w_1)$ of BBS+ signature. $\mathcal{A}_{BBS+}$ conducts the chosen message attack, as follows. For spk, select other parameters except pk_{BBS+} as in the real **Setup**. Then, start the simulation with $\mathcal{A}$, where spk is given to $\mathcal{A}$ in Setup. For Register transaction from a user controlled by $\mathcal{A}$, extract the secrets ζ'_0, x, S_0 from the PoK in the executed **Register** protocol and query the BBS+ signature on (x, S_0, rep_0) to the signing oracle of the BBS+ signatures. Then, the signature $\tilde{\sigma}_0 = (B_0, \eta_0, \zeta_0)$ can be obtained. After that, return $\tilde{\sigma}'_0 = (B_0, \eta_0, \zeta''_0 = \zeta_0 - \zeta'_0)$ to $\mathcal{A}$. For Show transaction from a user controlled by $\mathcal{A}$, similarly, using the extractor of PoK and the BBS+ signing oracle, return $\tilde{\sigma}'_t$. Then, in a successful Show transaction, a BBS+ signature is extracted, but the BBS+ signature on some messages has never been issued. Thus, as $\mathcal{A}_{BBS+}$, output the messages and the forged BBS+ signature.

$\mathcal{A}_{com}$**:** We construct $\mathcal{A}_{com}$ which corresponds to the case that $\mathcal{A}$ compromises the binding property of a commitment. $\mathcal{A}_{com}$ is given the public parameters of commitment $pp = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, g_3, g_4)$. For spk, select other parameters except pp as in the real **Setup**. Then, start the simulation with $\mathcal{A}$, where spk is given to $\mathcal{A}$ in Setup. In this case, using the secret key of BBS+ signatures, respond to each Register and Show transaction with a user controlled by $\mathcal{A}$, as in the real **Register** and **Show** protocols, where by extracting the secrets from PoK, check the binding property. Then, obtain $(\zeta'_t, x, S_t, rep_t)$ s.t. $C_{m,t} = g_1^{\zeta'_t} g_2^{x} g_3^{S_t} g_4^{rep_t}$ from PoK of a successful transaction, and then $(\tilde{\zeta}'_t, \tilde{x}, \tilde{S}_t, r\tilde{e}p_t)$ s.t. $C_{m,t} = g_1^{\tilde{\zeta}'_t} g_2^{\tilde{x}} g_3^{\tilde{S}_t} g_4^{r\tilde{e}p_t}$ from PoK of a later successful transaction (the secrets are signed as $B_t = (g_0 g_1^{\zeta''_t} C_{m,t})^{1/\gamma_t+\eta_t}$, and the knowledge of the compromised secrets $(\tilde{\zeta}'_t, \tilde{x}, \tilde{S}_t, r\tilde{e}p_t)$ is proved in the PoK for B_t). Output the collision $(\zeta'_t, x, S_t, rep_t)$ and $(\tilde{\zeta}'_t, \tilde{x}, \tilde{S}_t, r\tilde{e}p_t)$.

From the above discussions, we can conclude that the probability that the above bad **E1**–**E5** events happen is negligible.

In the remainder of this proof, we will show that players' outputs to $\mathcal{E}$ in the above simulation are the same as those in the real world. In the proposed system, the user's reputation value cannot be modified by anyone except the server. This is because the reputation value rep_{t-1} is certified by the BBS+ signature $\tilde{\sigma}_{t-1} = (B_{t-1}, \eta_{t-1}, \zeta_{t-1})$ issued by the server, where anyone except the server cannot compute $\tilde{\sigma}_{t-1}$. In **Show**, the user has to conduct the PoK, where, as shown in **Lemma 1**, it proves the knowledge $(B_{t-1}, \eta_{t-1}, \zeta_{t-1})$ satisfying $B_{t-1} = (g_0 g_1^{\zeta_t} g_2^{x} g_3^{S_{t-1}} g_4^{rep_{t-1}})^{1/\gamma_1+\eta_{t-1}}$ that is, the BBS+ signature on (x, S_{t-1}, rep_{t-1}). Furthermore, for the proved rep_{t-1}, the user also proves the knowledge of a BB signature on rep_{t-1} (As above-mentioned, strictly a variant of BB signature) from **Lemma 1**. Thus, the correct range of rep_{t-1} is ensured. In addition, the PoK shows $C'_{m,t} = g_1^{\zeta'_t} g_2^{x} g_3^{S_t} g_4^{rep_{t-1}}$ for the rep_{t-1}, and $C'_{m,t} g_4^{\Delta rep_t} = g_1^{\zeta'_t} g_2^{x} g_3^{S_t} g_4^{rep_{t-1}+\Delta rep_t}$ for feedback Δrep_t is signed by the BBS+ signature as the next certificate.

In each **Show** protocol, $\mathcal{S}$ as the honest server correctly renews the signed commitments, and the commitment is not compromised. The following holds in the commitments signed at the timing $t-1$ and t: The value x is always the same and $rep_t = rep_{t-1} + \Delta rep$. Since the reuse of S_{t-1} is checked in each **Show** protocols, the certificate on tag S_{t-1} can be available only once. Thus, we can consider a history sequence of protocol transcript: The first **Register** protocol using $x_i = x$, and then **Show** protocols using the same x_i follow. Due to the sameness of x in B_{t-1} and $C'_{m,t}$, the use of certificate based on another registration using x' is not mixed with the history sequence on $x_i = x$. Therefore, $rep_{t-1} = \mathsf{sum}_i$ should hold.

Thus, since $rep_{t-1} = \mathsf{sum}_i$ and the range proof is correct, there is no successful Show transaction from a dishonest user controlled by $\mathcal{A}$ such that $\mathcal{S}$ on behalf of the honest server accepts the user and outputs successful, but $\mathcal{T}$ indicates that the user does not satisfy the condition that sum_i is included in the range of ℓ. This implies that the outcomes of honest users and server, and dishonest users represented by $\mathcal{S}$ in the ideal world are the same as those of honest users and server, and dishonest users represented by $\mathcal{A}$ in the real world, except some negligible probability due to events **E1**–**E5**. Thus, $|\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathbf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)|$ is negligible. □

Lemma 3. *For any PPT environment $\mathcal{E}$ and any PPT real-world adversary $\mathcal{A}$ controlling a subset of users and the server, there exists a PPT ideal-world adversary (simulator) $\mathcal{S}$ s.t. $|\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathbf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)|$ is negligible in the random oracle model.*

Proof. In this case, we construct $\mathcal{S}$ which represents the dishonest server and users controlled by $\mathcal{A}$ to the trusted party $\mathcal{T}$ and $\mathcal{E}$, and on the other hand, $\mathcal{S}$ represents honest users to $\mathcal{A}$ in each transaction in the ideal world, as follows.

- Setup:
  - *Representing honest users to $\mathcal{A}$ as the dishonest server*: $\mathcal{S}$ receives spk from $\mathcal{A}$.
- Register:
  - *Representing a dishonest user and the dishonest server controlled by $\mathcal{A}$ to $\mathcal{T}$*: In this case, where dishonest user i instructed by $\mathcal{E}$ executes this transaction with the dishonest server, $\mathcal{S}$ acts on behalf of both user i and the server to $\mathcal{T}$. Finally, $\mathcal{S}$ outputs the outcomes from user i and the server to $\mathcal{E}$.
  - *Representing the dishonest server controlled by $\mathcal{A}$ to $\mathcal{T}$ / honest user to $\mathcal{A}$*: On the request on tid from $\mathcal{T}$ on behalf of user i, $\mathcal{S}$ as user i executes **Register** protocol with $\mathcal{A}$ as the server. In the protocol, $\mathcal{S}$ executes the zero-knowledge simulator instead of the *PoK* and uses a random $\mathbb{G}_1$-element instead of the commitment $C'_{m,0}$. $\mathcal{S}$ forwards $\mathcal{A}$'s response accept/reject to $\mathcal{T}$, and finally outputs the outcome (tid, success/failure) of $\mathcal{A}$ as the server to $\mathcal{E}$.
- Show:
  - *Representing a dishonest user and the dishonest server controlled by $\mathcal{A}$ to $\mathcal{T}$*: In this case, similarly to Register, $\mathcal{S}$ acts on behalf of both user i and the server to $\mathcal{T}$. Finally, $\mathcal{S}$ outputs the outcomes from user i and the server to $\mathcal{E}$.
  - *Representing the dishonest server controlled by $\mathcal{A}$ to $\mathcal{T}$ / honest user to $\mathcal{A}$*: On the request on tid, ℓ, and Δrep_t from $\mathcal{T}$ on behalf of an honest anonymous user, $\mathcal{S}$ as the user executes **Show** protocol with $\mathcal{A}$ as the server. In the protocol, $\mathcal{S}$ executes the zero-knowledge simulator instead of the *PoK* and uses random $\mathbb{G}_1$-elements and $\mathbb{Z}_p^*$-element instead of $C_{A_\ell}, C_{B_{t-1}}, C'_{m,t}, S_{t-1}$. $\mathcal{S}$ outputs the outcome (tid, success/failure) of $\mathcal{A}$ as the server to $\mathcal{E}$.

The simulation to $\mathcal{A}$ is perfect, due to the perfect zero-knowledge-ness of the *PoK*, and the perfect hiding of the commitment. In this simulation, $\mathcal{A}$ obtains no information from the **Register** and **Show** protocols executed with $\mathcal{S}$ on behalf of the honest users, since the values sent to $\mathcal{A}$ are only the zero-knowledge simulator and one-time random values. Thus, the information that $\mathcal{A}$ obtains in this simulation is the same as the information that $\mathcal{A}$ obtains in the ideal world (i.e., occurrences of **Register** protocols and **Show** protocols with the inputs $\ell, \Delta rep_t$). These imply that the information that $\mathcal{A}$ obtains in the real world is the same as the information $\mathcal{A}$ obtains (via the simulation) in the ideal world. Therefore, $\mathcal{A}$ cannot notice the difference between the real world and the simulation in the ideal world, and thus the outcomes of honest users, and dishonest users and server controlled by $\mathcal{A}$ via $\mathcal{S}$ in the ideal world are the same as the outcomes of honest users, and dishonest users and server controlled by $\mathcal{A}$ in the real world. Namely, $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) = \mathbf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)$. □

8. Efficiency Considerations

In this section, we compare the efficiency of our proposed system to the previous reputation system in ARTSense [5]. The number of communication rounds in one cycle of anonymous sensing data submission is shown in **Table 1**. The previous system [5] needs five rounds to complete, while our proposed system only needs two. The five rounds are interactive communications between the user and the server, as explained in Section 3.3. The phase of issuing reputation certificate is included as two rounds in Ref. [5]. Meanwhile, the proposed system needs only two rounds, which are (1) the user submits the sensing data and proves his current reputation level, (2) the server issues the certificate of the reputation updated with the feedback value.

Furthermore, in Ref. [5], "the redemption of coupon" phase is needed to update the user's reputation value in the database of the server side, where the user has to wait a random period for the request. If the period is short, the server can link the user's ID to the data submission. Thus, a relatively long delay is needed in a single cycle of a user's data submission and reputation management. Instead, in the proposed system, since the user's reputation is managed in each user side, the reputation management (i.e., **Show** protocol) completes within the data submission phase, which means that any delay is not needed. On the other hand, the proposed system has a limitation that each protocol must be executed sequentially, although the previous system allows the protocols to be executed concurrently. Thus, in case that lots of users access the server, the communication efficiency of our system may become worse. The extension to the concurrent system is our future work.

Next, we discuss the computation cost of the proposed system that are the overheads. In the cycle of anonymous sensing data submission, Ref. [5] only requires two exponentiation on RSA modules during the blinding and unblinding using blind RSA signatures, and any ordinary digital signature and encryption. Meanwhile, as shown in **Table 2**, the proposed system requires multi-exponentiation (ME) on $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$, and pairings on the user side and the server side. The ME costs on $\mathbb{G}_1, \mathbb{G}_2$ are considered comparable to the RSA or ordinary signature and encryption in

Table 1 Comparison of the number of rounds.

	Proposed system	[5]
Rounds	2	5

Table 2 Computation cost of the proposed system.

		User	Server
ME	$\mathbb{G}_1, \mathbb{G}_2$	4	2
	$\mathbb{G}_T$	2	4
Pairing		1	5

Ref. [5], but the ME on $\mathbb{G}_T$ and pairing computations are relatively heavy. However, the computations in the user side can be pre-computed before the phase of sensing data submission (the on-line computations are only response computations in the *PoK*, which are only light multiplications).

9. Conclusions

In this paper, an efficient anonymous reputation system for crowd sensing is proposed. The proposed system achieves the reputation update within the data submission, by adapting a P2P anonymous reputation system from Ref. [7] to crowdsensing. As a result, we solved the efficiency problem of communication delay occurred in ARTSense. Furthermore, the communication rounds are also reduced. Meanwhile, the implementation-based evaluations to clarify the practicality in crowdsensing is one of the future works. In addition, to achieve concurrent executions of protocols is our future work.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 19K11964 and UTM GUP Grant Number 16J83.

References

[1] Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N.Y., Huang, R. and Zhou, X.: Mobile crowdsensing and Computing: The Review of an Emerging Human-powered Sensing Paradigm, *ACM Comput. Surv.*, Vol.48, No.1, pp.7:1–7:31 (2015).

[2] Shin, M., Cornelius, C., Peebles, D. Kapadia, A., Kotz, D. and Triandopoulos, N.: AnonySense: A System for Anonymous Opportunistic Sensing, *Pervasive and Mobile Computing*, Vol.7, No.1, pp.16–30 (2011).

[3] Cristofaro, E.D. and Soriente, C.: Extended Capabilities for a Privacy-Enhanced Participatory Sensing Infrastructure (PEPSI), *IEEE Trans. Information Forensics and Security*, Vol.8, No.12, pp.2021–2033 (2013).

[4] Rahaman, S., Cheng, L., Yao, D.D., Li, H. and Park, J.J.: Provably Secure Anonymous-yet-Accountable Crowdsensing with Scalable Sublinear Revocation, *PoPETs*, Vol.2017, No.4, pp.384–403 (2017).

[5] Oscar, X., Cheng, W., Mohapatra, P. and Abdelzaher, T.: ARTSense: Anonymous Reputation and Trust in Participatory Sensing, *2013 Proc. IEEE INFOCOM*, pp.2517–2525 (2013).

[6] Sadiah, S. and Nakanishi, T.: An Efficient Anonymous Reputation System for Crowd Sensing, *2019 Seventh International Symposium on Computing and Networking Workshops* (*CANDARW*), pp.374–380 (2019).

[7] Nakanishi, T. and Funabiki, N.: An Anonymous Reputation System with Reputation Secrecy for Manager, *IEICE Trans. Fundamentals*, Vol.E97-A, No.12, pp.2325–2335 (2014).

[8] Androulaki, E., Choi, S.G. Bellovin, S.M. and Malkin, T.: Reputation Systems for Anonymous Networks, *Proc. PET 2008*, LNCS 5134, pp.202–218, Springer-Verlag (2008).

[9] Blömer, J., Juhnke, J. and Kolb, C.: Anonymous and Publicly Linkable Reputation Systems, *Proc. FC 2015*, LNCS 8975, pp.478–488, Springer-Verlag (2015).

[10] Blömer, J., Eidens, F. and Juhnke, J.: Practical, Anonymous, and Publicly Linkable Universally-Composable Reputation Systems, *Proc. CT-RSA 2018*, LNCS 10808, pp.470–490, Springer-Verlag (2018).

[11] Kaafarani, A.E., Katsumata, S. and Solomon, R.: Anonymous Reputation Systems Achieving Full Dynamicity from Lattices, *Proc. FC 2018*, LNCS 10957, pp.388–406, Springer-Verlag (2018).

[12] Boneh, D., Boyen, X. and Shacham, H.: Short Group Signatures, *CRYPTO 2004*, LNCS 3152, pp.41–55, Springer-Verlag (2004).

[13] Boneh, D. and Boyen, X.: Short Signatures Without Random Oracles, *EUROCRYPT 2004*, LNCS 3072, pp.56–73, Springer-Verlag (2004).

[14] Blenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A. and Sacham, H.: Randomizable proofs and delegatable anonymous credentials, *CRYPTO 2009*, LNCS 5677, pp.108–125, Springer-Verlag (2009).

[15] Au, M.H., Susilo, W. and Mu, Y.: Constant-size dynamic k-TAA, *SCN 2006*, LNCS 4116, pp.111–125, Springer-Verlag (2006).

[16] Camenisch, J., Drijvers, M. and Lehmann, A.: Anonymous attestation using the strong Diffie Hellman assumption revisited, *Proc. 9th International Conference on Trust and Trustworthy Computing* (*TRUST 2016*), pp.1–20, Springer (2016).

[17] Camenisch, J.: Group signature schemes and payment systems based on the discrete logarithm problem, PhD thesis, pp.1–174, ETH Zurich, ISBN 978-3-89649-286-9 (1998).

[18] Damgård, I.: On Σ-Protocols, available from ⟨http://www.daimi.au.dk/˜ivan/Sigma.pdf⟩.

[19] Schnorr, C.P.: Efficient signature generation for smart cards, *Journal of Cryptology*, Vol.4, No.3, pp.239–252 (1991).

[20] Libert, B., Mouhartem, F., Peters, T. Yung, M.: Practical "Signatures with Efficient Protocols" from Simple Assumptions, *Proc. AsiaCCS 2016*, pp.511–522 (2016).

[21] Au, M.H., Tsang, P.P. and Kapadia, A.: PEREA: Practical TTP-free revocation of repeatedly misbehaving anonymous users, *ACM Trans. Information and System Security* (*TISSEC*), Vol.14, No.4, pp.29:1–29:34 (2011).

[22] Au, M.H., Kapadia, A. and Susilo, W.: BLACR: TTP-free blacklistable anonymous credentials with reputation, *Proc. 19th Annual Network and Distributed System Security Symposium* (*NDSS 2012*) (2012).

[23] Au, M.H. and Kapadia, A.: PERM: Practical reputation-based blacklisting without TTPs, *Proc. 2012 ACM Conference on Computer and Communications Security* (*ACM-CCS 2012*), pp.929–940 (2012).

[24] Nakanishi, T., Fujii, H., Hira, Y. and Funabiki, N.: Revocable Group Signature Schemes with Constant Costs for Signing and Verifying, *IEICE Trans. Fundamentals*, Vol.E93-A, No.1, pp.50–62 (2010).

Shahidatul Sadiah received a M.Eng. in Electronic and Information System Engineering from Okayama University, Japan, in 2015, and a Ph.D. in Information Engineering from Hiroshima University, Japan, in 2018. She joined the department of Electronics and Computer Engineering, Universiti Teknologi Malaysia as a Senior Lecturer in 2019. Her research interests include cryptography, information security, digital system design, and computer architecture.

Toru Nakanishi received his M.S. and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1995 and 2000 respectively. He joined the Department of Information Technology at Okayama University, Japan, as a research associate in 1998, and moved to the Department of Communication Network Engineering in 2000, where he became an assistant professor and an associate professor in 2003 and 2006 respectively. In 2014, he moved to the Department of Information Engineering (currently, the Graduate School of Advanced Science and Engineering (Informatics and Data Science Program)) at Hiroshima University as a professor. His research interests include cryptography and information security.