Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 81-100
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Enabling Privacy Preservation and Decentralization for Attribute-Based Task Assignment in Crowdsourcing

Tianqing Liang

College of Information Science and Technology, Jinan University, Guangzhou, China
Email: aurelia.ltq@gmail.com

How to cite this paper: Liang, T.Q. (2020) Enabling Privacy Preservation and Decentralization for Attribute-Based Task Assignment in Crowdsourcing. *Journal of Computer and Communications*, **8**, 81-100.
https://doi.org/10.4236/jcc.2020.84007

Received: March 7, 2020
Accepted: April 21, 2020
Published: April 24, 2020

Abstract

Crowdsourcing allows people who are endowed with certain skills to accomplish special tasks with incentive. Despite the state-of-art crowdsourcing schemes have guaranteed low overhead and considerable quality, most of them expose task content and user's attribute information to a centralized server. These servers are vulnerable to single points of failure, the leakage of user's privacy information, and lacking of transparency. We therefore explored an alternative design for task assignment based on the emerging decentralized blockchain technology. While enabling the advantages of the public blockchain, changing to open operations requires some additional technology and design to preserve the privacy of user's information. To mitigate this issue, we proposed a secure task assignment scheme, which enables task content preservation and anonymous attribute requirement checking. Specifically, by adopting the cryptographic techniques, the proposed scheme enables task requester to safely place his task in a transparent blockchain. Furthermore, the proposed scheme divides the attribute verification process into public pre-verification and requester verification, so that the requester can check only the identity of the worker, instead of verifying the attributes one by one, thereby preserving the identity of worker while significantly reducing the requester's calculation burden. Additionally, security analysis demonstrated unrelated entities cannot learn about the task content and identity information from all data uploaded by requester and worker. Performance evaluation showed the low computational overhead of our scheme.

Keywords

Crowdsourcing, Task Assignment, Attribute-Based Encryption, Blockchain, Smart Contract

1. Introduction

Crowdsourcing is a powerful method, has emerged in the landscape of problem solving, to outsource the work originally done by the designated party to an unknown group of people in an open manner [1]. It enables tasks to be completed by specific professionals on demand, which significantly reduces costs and improves the quality of the solution. Along with these advantages, many large companies have successfully applied it into the market, such as ImageNet [2], Amazon Mechanical Turk [3] and UBER [4]. These applications mainly cover areas where devices have poor or even no computing capacity, and there still require further improvement.

All the participants on the process of crowdsourcing can be divided into three types of roles: requester, worker and platform. To be specific, the one that publishing tasks is considered as the requester, and the one that working on those tasks is named as the worker. The middleman between the requester and the worker is the platform, who is responsible for storing the tasks and maintaining the correct execution of the whole process. Many crowdsourcing applications share a similar structure: the requester submits the task content along with the reward to the platform, and then the workers accept the task and submit the solution of this task within the fixed time. After that, the requester confirms the quality of the solution and pays for the pre-declared reward to the worker.

Although these crowdsourcing applications have achieved considerable success, some of the key challenges still need to be addressed. One critical aspect is the lack of a credible guarantee on the quality of the work. Workers who have accepted the work may not have the corresponding skills to provide valuable answers [5] [6]. Statistical aggregation algorithms can tolerate some low-quality answers [7] [8] [9], but leaves a waste of resources. A straightforward approach is to customize the credentials based on the background of each worker and make sure that those workers only accept tasks within their capabilities [10] [11]. Currently, these credentials are usually distributed by various agencies. The crowdsourcing system will ask workers to upload these credentials in order to achieve capacity limitations during the task assignment process.

Another aspect is data confidentiality. Traditional centralized platforms typically obtain task content in plaintext. The compromise of the platform will result in the disclosure of information of the user. Therefore, most of existing solutions assume that the platform should be honest during the protocol, which is impractical [12]. Various examples have shown the potential threats of platform compromises, such as UBER, which has been affected by unreliable order issues and users' data leakage [13] [14]. To address this, an alternative design needs to be explored to achieve secure task assignment based on a more open and distributed infrastructure.

The Blockchain is a decentralized and intelligent infrastructure [15] [16]. Compared to the traditional distributed solution, blockchain enables the masses to join as participants, making it an ideal start point. In this paper, we adopt the

design of the consortium chain because it has the best performance. In blockchain, the data will be initially verified by the agencies, then encapsulated into a block and appended to an existing chain. The remaining network participants perform the verification. When a chain is verified by a participant, any changes of this chain can be recognized by the participant. This feature has spawned countless fascinating decentralized applications [17] [18] [19]. Implementing the task assignment on the blockchain alleviates concerns about single points of failure. However, the open setting of the blockchain may pose a more serious threat to data confidentiality.

To solve the security issues in task assignment process, this paper uses skill credential to restrict the access of task content, which is achieved through Attribute-based Encryption (ABE) [20]. Specifically, each skill corresponds to an attribute one by one. Depending on the attributes owned by the worker, the authority will only distribute the credential keys of those attributes to the worker. By applying credential during the encryption process, the requester can ensure that his task content is only visible to those who fully satisfy task's access control settings. Unfortunately, this method only preserves the privacy of the requester and still requires disclosure of the worker's identity. Because a task can accept multiple solutions. If there are no restrictions, workers will be motivated to submit their solutions multiple times, in such way that they can get more rewards than they actually do. Traceable Attribute-Based Signature [21] allows a signer, who own a set of attributes, to sign a message and make the recipient of the signature believe that the signer owns some attributes. It introduces a special tracing authority that has the capable of revealing the identity of the signer, but also brings back the weakness of centralization, so this technology cannot be directly introduced.

Correspondingly, the requester should be responsible for his own task. The requester will be given the ability to reveal and verify the identity of the participants in his task. But the true identity of the worker is not needed for the requester. Therefore, the identity shown to the requester will be replaced by an anonymous account approved by the authority. To fulfill the requirements of verification in above way, we design a novel scheme based on the ABE scheme proposed by Lewko and Waters [22] and bring out corresponding functional expansion. We propose a credential that are constructed by binding the anonymous account and the worker's attributes. Only the owner of the credentials can use it for decryption. Then, the worker can cover his real identity in the credential and form a proof. Anyone can check the validation of the proof and confirm that the prover satisfies certain attributes. But only the person designated by the prover (the requester of the corresponding task) can reveal the identity from the proof.

Our contributions: In this paper, our main contributions are as follows.

1) A secure attribute-based task assignment scheme is proposed, which can preserve information security on a transparent blockchain. Moreover, everyone

can verify the correctness of the process without revealing the identity of the worker.

2) We preserved the privacy of worker with a random anonymous account, so that workers can change their identity at any time, which prevent requester from discovering associations among participants in different tasks.

3) We designed the attribute verification protocol with two aspects: public pre-verification and requester verification. Most verification works are performed by blockchain while only some steps are performed privately by the requester who knows the extra information, which significantly saves the computation cost in the requester's side. Therefore, the requester can prove the misbehaving of the worker by exposing additional information he knows.

4) We implemented the proof-of-prototype and the experimental results have shown the validation and feasibility of our proposed scheme.

The rest of this paper is organized as follows. Section 2 reviews the related work on task assignment for crowdsourcing system. We present models and goals in Section 3. Next, our scheme detail is presented in Section 4. The privacy discussions and performance evaluation are presented in Sections 5 and 6 respectively, followed by a conclusion in Sections 7.

2. Related Work

2.1. Attribute-Based Encryption

ABE was first proposed by Sahai and Waters [23]. In an ABE system, each user has a unique ID and a set of attributes. In general, ABE can be divided into two categories: Key-Policy Attribute-Based Encryption (KP-ABE) [24] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [25]. In KP-ABE, ciphertext is associated with a set of attributes, and the user's private key is associated with an access structure. The user can decrypt the ciphertext if and only if the attributes in the ciphertext satisfy the access structure of the user's private key. However, the encryptor cannot completely control over the encryption policy in KP-ABE. In CP-ABE, ciphertext is created with an access structure, and the user's private key is generated based on the user's attributes. The user can decrypt the ciphertext if and only if the attribute of the user's private key satisfies the ciphertext access policy. In doing so, the encryptor is enabled to determine the access control of the ciphertext.

These schemes use a centralized approach with only one key distribution center (KDC), so they inherit all the centralized weaknesses such as single point of failure. The multi-authority ABE protocol is proposed and addressed to this problem. In multi-authority ABE, the entire attribute set is divided into N disjoint sets and managed by N authorities. Under this setting, each authority only knows part of user's attribute, and user is required to get the private key from all KDCs. Based on this model, many attribute-based encryption schemes with multiple authorities have been proposed, but they still rely on a semi-honest central authority [26] [27] [28], or cannot resist the user's collusion

attack [29]. The work proposed by Jung *et al.* [20] can tolerant up to $N-2$ authority compromise, and do not require a trusted server. However, their work is difficult to modify the number of authority after setup,. On the other hand, the work by Lewko *et al.* [22] cannot prevent the authority from being aware of the user's key during the key generation phase.

2.2. Crowdsourcing System

2.2.1. Centralized Crowdsourcing Systems

Many crowdsourcing systems are built in a centralized manner [3] [30] [31]. In order to understand the capabilities of workers and the tasks they are interested in, the platform requires the worker to complete their profile before joining. Correspondingly, the platform needs to learn the plain text of the task content so that the task content can be sent to the worker. During this process, requesters and workers submit their private information in exchange for platform services. This type of information is known and stored by a single party and is therefore vulnerable to a variety of attacks and privacy leakage. In a system with limited task content, such as Mturk [3], workers only need to complete some human intelligence tasks. Worker only needs to pass a non-robot test to become a qualified worker. This convenience allows worker to change their account at low cost. Dynamo [32] specifically designed a wrapper for it, using pseudo IDs to provide unlinkability, but it is difficult to extend to multi-attribute task content and greatly limits its scope of application.

2.2.2. Distributed Crowdsourcing Systems

In spatial crowdsourcing (SC), the geographic location of workers and requesters is considered private information and should not be known to the platform and unrelated people. Liu *et al.* [33] proposes a model that divides the server into SC server and crypto service provider (CSP). The users encrypt their locations using the public key provided by CSP and hands it over to the SC server. The SC server then operates calculation on the ciphertexts and passes the results to the CSP. The CSP then decrypts and publishes the results, but only the eligible workers can restore the location. This model requires that both the SC and CSP are semi-honest, and do not consider the case of collusion, so the degree of decentralized is very limited. In addition, the requester's geographic location is still known to the SC server, which is a privacy leakage of the requester.

2.2.3. Decentralized Crowdsourcing Systems

Li *et al.* [34] uses a reputation system to regulate workers' behaviors. Although workers use pseudonyms as their identity, the linkability between different tasks expose the interest of workers. And changing identity will lose its existing reputation, which brings great damages to workers. Lu *et al.* [35] proposes a private and anonymous crowdsourcing system based on common-prefix-linkable anonymous authentication. Each task has a unique prefix. Unless a worker proves his identity twice in a prefix, he stays anonymous and unlinkable across tasks.

However, these systems still treat the task content as open access data, which cause privacy leakage of the requester. In addition, in order to make users identifiable, these systems use registry authority to identify users, which makes the decentralized effect of the system questionable.

3. Preliminaries

3.1. System Model

Our system model is shown in Figure 1. It contains four entities as follows:

Authority: The authority has the right to endorse certain abilities in specific areas and provide qualified workers with keys that correspond to their anonymous accounts and capabilities. Note that each ability is treated as a single attribute. In addition, the authorities act as proposers of the blockchain block, that is, they are responsible for packing the information sent to the contract into blocks and appending them to the existing chain. Other entities can get the chain and verify it.

Smart Contract: The contract receives and stores the task content ciphertext posted by requester and the attribute proofs submitted by the worker. It validates the legitimacy of proof to detect misconduct, thereby ensuring a fair judgment in the dispute between the requester and the worker.

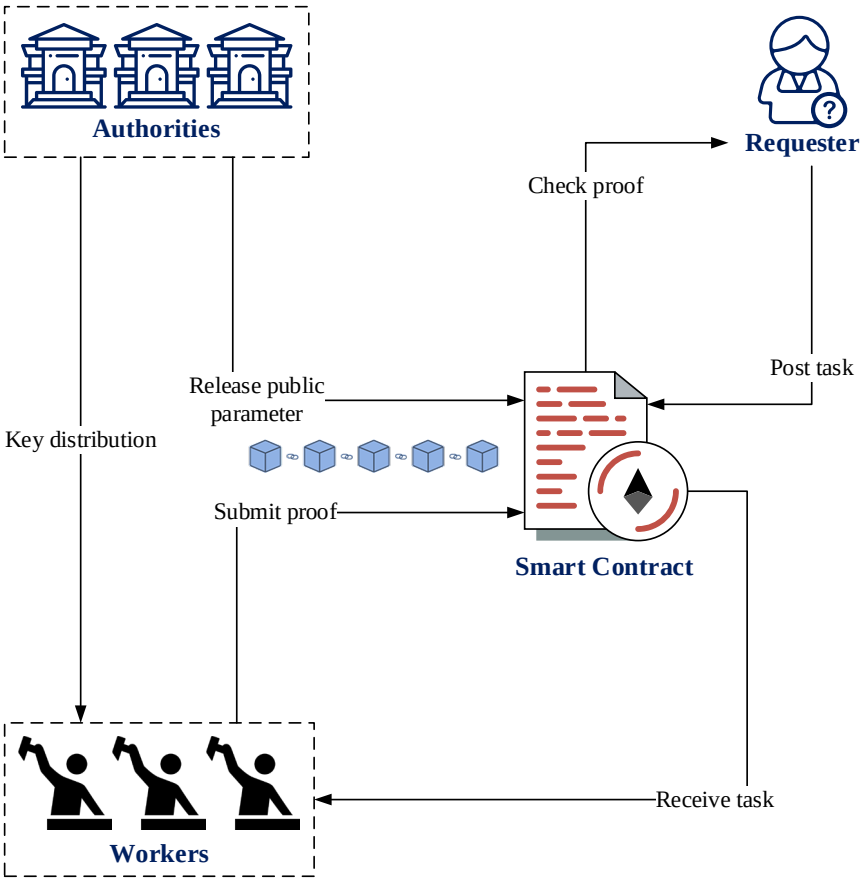

Figure 1. System model.

Requester: The requester encrypts the task content according to the attribute requirements of his task and submits the task ciphertext to the contract. When the worker accepts the task and submits the attribute proof, he needs to verify the legality of the worker's anonymous identity and require the contract to make a judgment when the verification fails.

Worker: The worker creates an anonymous account in advance and obtains the credential keys from the authorities based on his or her attributes. Using these keys, he can decrypt the task ciphertext which satisfies the required policy and submit the appropriate attribute proof when deciding to accept the task.

3.2. Security Model

The authorities are semi-honest which means they follow our proposed scheme in general. We assume authorities are interested in which worker is using the key they distributed to participate in the work, but they will not collude with users or other authorities. Note that our system inherits the weakness of the blockchain. Although the authorities are semi-honest, the blockchain can resist 51% attacks. However, such attacks against blockchain infrastructure are considered out of scope.

The smart contract runs on the blockchain, which guarantees its availability and integrity, but does not include confidentiality. Other entities can directly read its data through the blockchain, but have no ability to tamper it.

The requester also assumed to besemi-honest. His request can only be assigned to a valid worker when the task assignment process is properly executed, so requester will follow the scheme in general. In particular, we assume that he is interested in the identity information of the workers involved in his task.

Workers are untrusted since they are random users. They may collude with other workers to accept a task which they are not allowed to or attempt to accept a task more than one times.

In our scenario, we define the security of worker and requester information as follows:

Task content security: The task content ciphertext should only be decrypted by workers who fully satisfy the task attribute requirements.

Worker identity security: When the worker decides to accept a task, he will upload a proof to the contract. These can be divided into three main cases:

1) Given a proof. No entity can restore worker's global identity from the proof.

2) Given two workers who have accepted the attribute key distributed by the authority, and a proof constructed by one of them. The authority cannot distinguish which worker constructed the proof.

3) Given two anonymous account and a proof constructed by one of them, other workers who can decrypt the task cannot distinguish which account construct the proof.

3.3. Epoch

Tasks generally involve time-related restrictions such as deadlines. Therefore,

the workers should check the consistency of time with the blockchain when accepting tasks. We introduce the epoch to the process of task acceptance. There is a stamp in each epoch. The worker's request is legal only when the worker uses the stamp of the current epoch in his message. Figure 2 is an example of every three blocks as an epoch. The stamp of the epoch is the hash of the last block of the previous epoch. Since the hash value of a block has only a negligible probability of collision, if the worker's message is not packed in a block of a certain epoch, the worker can ensure that his message has expired. This prevents messages from being packed into blocks after a long time.

In the proposed scenario, the consortium chain does not need to propose the block through the proof-of-work, so the block time is stable. Therefore, the duration of each epoch does not have a large deviation.

4. Proposed Secure Task Assignment Scheme

In this section, the proposed secure attribute-based task assignment scheme will be described in detail. To give a better understanding, the main notations will be listed in Table 1.

4.1. Scheme Overview

As shown in Figure 3, the proposed scheme consists of five steps. In step 1, the

Table 1. Notations.

Notation	Definition
W_{id}	Worker's global identity
APK_i	The public key of attribute i
ASK_i	The secret key of attribute i
UPK_z	The public key of anonymous account Z
USK_z	The secret key of anonymous account Z
$WSK_{i,id,z}$	The credential key of attribute i for W_{id}'s anonymous account Z
$PK_{\text{requester}}$	The public key used by the requester when publishing the task
$Enc(key, content)$	A function to encrypt content with a public key encryption, for instance, RSA
$Rset_{\text{account}}$	Revoked anonymous account collection
CT_{task}	Task content ciphertext
$Proof_{\text{task}}$	Proof of attribute for the task
$stamp$	The stamp of current blockchain epoch

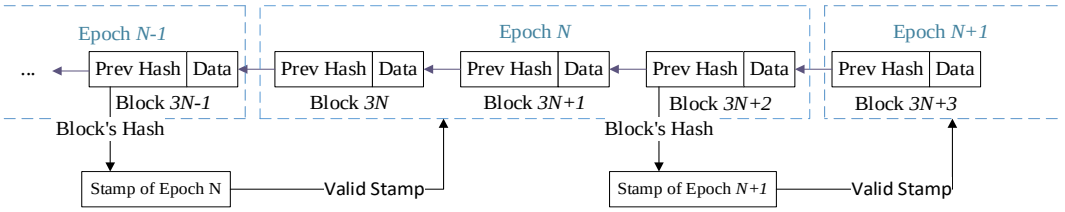

Figure 2. Description of the epoch.

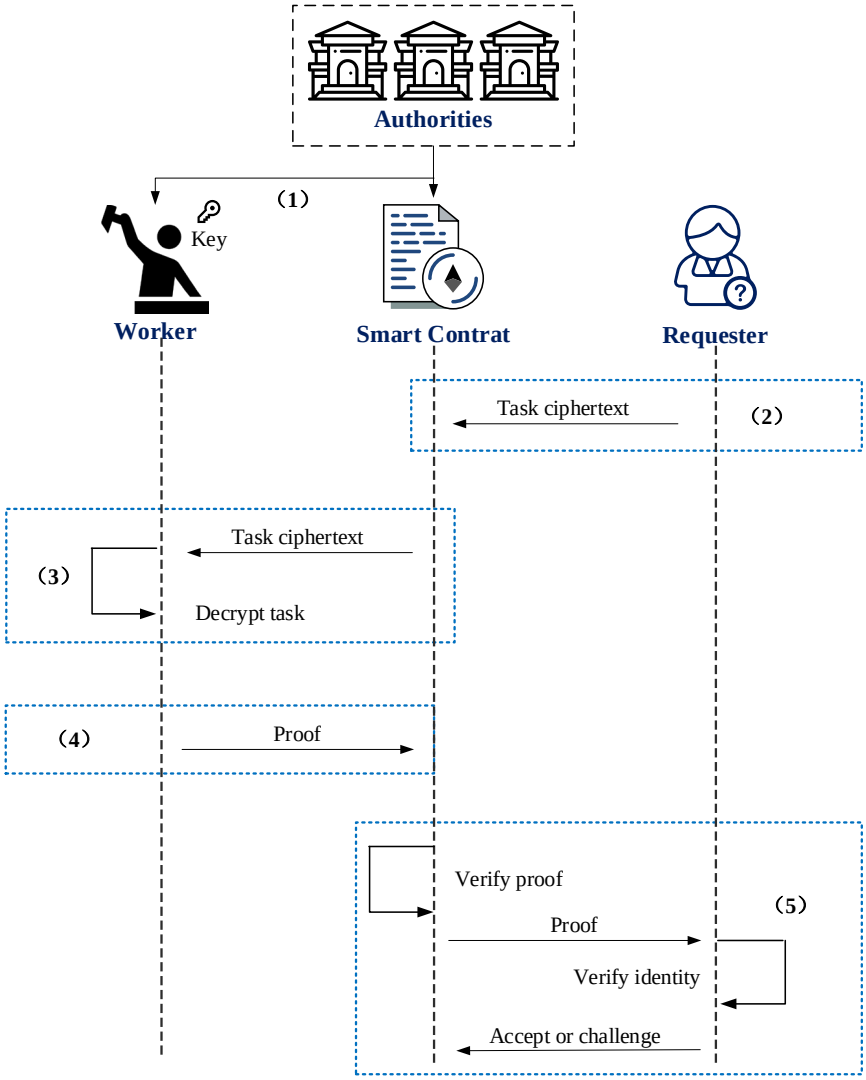

Figure 3. Overview of task assignment process.

authority creates key pair APK_i, ASK_i for each of the attribute i managed by him, and publishes the APK_i on the contract. The worker creates his own anonymous account z and requests the authority to distribute the credential key $WSK_{i,id,z}$ bound to his account. In step 2, the requester encrypts the task content according to the requirement policy, and publishes the task ciphertext CT_{task} on the contract. In step 3, the worker obtains the task ciphertext that satisfies the requirement policy and decrypts it with his secret key $USK_z, WSK_{i,id,z}$ to learn the task content. In step 4, when the worker intends to accept the task, he can use his secret key to build a proof $Proof_{\text{task}}$ corresponding to the task and then send it to the contract. In step 5, the contract first verifies the correctness of the proof and then asks the requester to verify whether the proof comes from a legitimate anonymous account. Otherwise, the requester can initiate the challenge by providing additional information and submit it to the contract for final judgment.

4.2. Scheme Construction

The proposed scheme is based on ABE with multiple authorities proposed by Lewko and Waters [22]. In the task publication and task decryption phase, their ABE scheme will serve as an encryption method to preserve the security of the task content. The concrete construction is shown as follows.

System Initialization In the global settings, select a prime p, groups $\mathbb{G}_1$ and $\mathbb{G}_T$ of order p, a map $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_T$, and two hash functions $\mathrm{H}:\{0,1\}^*\to\mathbb{G}_1$, $\mathrm{F}:\{0,1\}^*\to\mathbb{Z}_p$ the former maps the user's identity W_{id} to $\mathbb{G}_1$ and the latter maps a stamp to integers. Also, two random primitives $g,u\in\mathbb{G}_1$ are picked. Finally, the public parameter is set as $\{p,\mathbb{G}_1,\mathbb{G}_T,e,\mathrm{H},\mathrm{F},g,u\}$.

Authority Setup For each attribute i managed by the authority, choose two random values $a_i,y_i\in\mathbb{Z}_p$. Note that the attribute indices i for each authority are distinct, *i.e.* each attribute corresponds to only one authority. The authority will keep these two values as the secret key to attribute i:

$$ASK_i=\{a_i,y_i\} \tag{1}$$

The public key of the attribute i will be posted on the contract as public knowledge.

$$APK_i=\left\{e(g,g)^{a_i},g^{y_i},g^{\frac{1}{a_i}},u^{a_i},u^{y_i}\right\} \tag{2}$$

Worker registration Workers can create new anonymous accounts at any time and request the appropriate credential keys from the authority, Workers are not allowed to have multiple qualified anonymous accounts at the same time, so when a worker applies for a new anonymous account through his global identity W_{id}, since the authority knows the association between the worker's global identity and the anonymous account, the authority can announce that the worker's previous anonymous account has been revoked. Because the authorities are semi-honest, the revoke process for anonymous accounts can be done reliably. Authorities are assumed to jointly maintain and disclose a set of revoked accounts $Rset_{\text{account}}$.

To create an anonymous account, the worker picks a random number $z\in\mathbb{Z}_p$ as his secret key of anonymous account USK_z and then sends the corresponding public key $UPK_z=g^z$ and his identity W_{id} to the authority. The authority will distribute a key for each attribute i that the worker has

$$WSK_{i,id,z}=g^{a_i z}\mathrm{H}(W_{id})^{y_i} \tag{3}$$

Task publication According to the attribute requirements of the task, the requester constructs the linear secret-sharing schemes (*LSSS*) matrix R with ρ mapping its rows to attributes, and then the requester the task content as follows:

1) Selecting two random values $s,k\in\mathbb{Z}_p$ and generate an asymmetric key pair, where $PK_{\text{requester}}$ is the public key. The task description is combined with k

as the message M.

2) Choose a random vector $v \in \mathbb{Z}_p^h$ with s as its first entry, and a random vector $w \in \mathbb{Z}_p^h$ with 0 as its first entry, where h is the number of rows in R.

3) For each row R_x of R, calculate $\lambda_x = R_x \cdot v$ and $\omega_x = R_x \cdot w$, and choose a random value $r_x \in \mathbb{Z}_p$.

4) Calculate the following parameters

$$\begin{aligned} &C_0 = M \cdot e(g,g)^s, C_1 = u^k, \\ &\text{for all } x: \\ &C_{x,1} = e(g,g)^{\lambda_x + a_{\rho(x)} r_x}, C_{x,2} = g^{y_{\rho(x)} r_x + \omega_x}, C_{x,3} = g^{r_x}, C_{x,4} = u^{a_{\rho(x)} k} \end{aligned} \tag{4}$$

Finally, Requester sends the task ciphertext CT_{task} to the contract.

$$CT_{\text{task}} = \left\langle R, \rho, C_0, C_1, PK_{\text{requester}}, \left\{ C_{x,1}, C_{x,2}, C_{x,3}, C_{x,4} \right\}_{\forall x} \right\rangle \tag{5}$$

The contract checks the correctness of $C_{x,4}$ by Equation (6) because $C_{x,4}$ will be used in the subsequent proof verification phase. The task will be retained if the check is passed.

$$e(C_{x,4}, u) = e\left(u^{a_{\rho(x)} k}, u\right) = e\left(u^{a_{\rho(x)}}, u^k\right) = e\left(u^{a_{\rho(x)}}, C_1\right) \tag{6}$$

Task decryption The worker obtains the task ciphertext CT_{task} from the contract and decrypts it as follows:

1) Calculate $\mathrm{H}(W_{id})$ with its identity W_{id}.

2) Choose a subset of rows R_x from R such that the worker has the credential key $WSK_{\rho(x),id,z}$ and constants $c_x \in \mathbb{Z}_N$ which satisfies $\sum_x c_x R_x = (1, 0, \cdots, 0)$.

3) For each row in R_x, compute

$$\begin{aligned} TK_x &= \frac{\left(C_{x,1}\right)^z \cdot e\left(C_{x,2}, \mathrm{H}(W_{id})\right)}{e\left(C_{x,3}, WSK_{\rho(x),id,z}\right)} \\ &= \frac{e(g,g)^{\lambda_x z + a_{\rho(x)} r_x z} \cdot e\left(g, \mathrm{H}(W_{id})\right)^{y_{\rho(x)} r_x + \omega_x}}{e(g,g)^{a_{\rho(x)} r_x z} \cdot e\left(g, \mathrm{H}(W_{id})\right)^{y_{\rho(x)} r_x}} \\ &= e(g,g)^{\lambda_x z} \cdot e\left(g, \mathrm{H}(W_{id})\right)^{\omega_x} \end{aligned} \tag{7}$$

4) Restore the task content using the anonymous account private key USK_z

$$M = \frac{C_0}{\left(\prod_x (TK_x)\right)^{\frac{1}{USK_z}}} = \frac{M \cdot e(g,g)^s}{\left(e(g,g)^{sz}\right)^{\frac{1}{z}}} \tag{8}$$

Finally, the worker can determine the correctness of his decryption through k in task content M and C_1 in CT_{task}.

Proof publication After learning the content of the task, if the worker decides to accept the task, he needs to construct a proof to prove that his attributes can meet the requirements and he has not accepted this task before. This proof will be published on the contract for verification by other entities.

Proving that the worker can meet the requirements is equivalent to proving

that the worker has a unrevoked anonymous account with credential key corresponding to the R_x used in the decryption, so the worker constructs the proof as follows:

1) Select a random value $d \in \mathbb{Z}_p$ blind $\mathrm{H}(W_{id})$, in addition, get the latest epoch stamp of the blockchain *stamp*.

2) For each row in R_x, pick a random number $t_x \in \mathbb{Z}_p$.

3) Calculate the following parameters

$$P_0 = \mathrm{H}(W_{id})^k \cdot g^d, P_1 = stamp, P_{\text{account}} = Enc\left[PK_{\text{requester}}, UPK_z\right],$$

for all x:

$$P_{x,2} = u^{t_x}, P_{x,3} = u^{y_{\rho(x)}t_x}, \tag{9}$$

$$P_{x,4} = \left(WSK_{\rho(x),id,z}\right)^k \cdot g^{\frac{z}{t_x}+y_{\rho(x)}d} = g^{a_{\rho(x)}kz+\frac{z}{t_x}+y_{\rho(x)}d}\mathrm{H}(W_{id})^{ky_{\rho(x)}},$$

$$P_{x,5} = g^{zt_x+\frac{z}{a_{\rho(x)}k}}, P_{x,6} = u^{\frac{1}{t_x+\mathrm{F}(stamp)}}, P_{x,7} = Enc\left(PK_{\text{requester}}, u^{a_{\rho(x)}t_xk}\right)$$

At last, The worker sends the proof $Proof_{\text{task}}$ to the contract as a request to accept the task.

$$Proof_{\text{task}} = \left\langle R_x, P_0.P_1, P_{\text{account}}, \left\{P_{x,2}, P_{x,3}, P_{x,4}, P_{x,5}, P_{x,6}, P_{x,7}\right\}_{\forall x}\right\rangle \tag{10}$$

Proof verification The verification of the proof is divided into two parts: the contract verifies the correctness of all parameters except $P_{x,5}, P_{x,7}, P_{\text{account}}$, and the requester verifies the correctness of the above three parameters and the legitimacy of the account.

For the contract, it first checks whether P_1 is the stamp of current epoch, and then tests the following equation:

$$e\left(P_{x,6}, P_{x,2} \cdot u^{\mathrm{F}(P_1)}\right) = e\left(u^{\frac{1}{t_x+\mathrm{F}(stamp)}}, u^{t_x} \cdot u^{\mathrm{F}(stamp)}\right) = e(u,u) \tag{11}$$

If the Equation (11) is true, it demonstrates the worker has knowledge of t_x, so $P_{x,2}$ is not obtained by the worker based on any APK, and it does not contain any information about $a_{\rho(x)}$.

$$\frac{e\left(u^{y_{\rho(x)}}, P_{x,2}\right)}{e\left(P_{x,3}, u\right)} = \frac{e\left(u^{y_{\rho(x)}}, u^{t_x}\right)}{e\left(u^{y_{\rho(x)}t_x}, u\right)} = 1 \tag{12}$$

$$\begin{aligned}\frac{e\left(P_{x,4}, P_{x,2}\right)}{e\left(P_0, P_{x,3}\right)} &= \frac{e\left(g^{a_{\rho(x)}kz+\frac{z}{t_x}+y_{\rho(x)}d}\mathrm{H}(W_{id})^{ky_{\rho(x)}}, u^{t_x}\right)}{e\left(\mathrm{H}(W_{id})^k \cdot g^d, u^{y_{\rho(x)}t_x}\right)} \\ &= \frac{e(g,u)^{a_{\rho(x)}kzt_x+z} \cdot e(g,u)^{y_{\rho(x)}dt_x} \cdot e\left(\mathrm{H}(W_{id}),u\right)^{y_{\rho(x)}t_xk}}{e(g,u)^{y_{\rho(x)}dt_x} \cdot e\left(\mathrm{H}(W_{id}),u\right)^{y_{\rho(x)}t_xk}} \\ &= e(g,u)^{a_{\rho(x)}kzt_x+z} = e\left(g^{zt_x+\frac{z}{a_{\rho(x)}k}}, u^{a_{\rho(x)}k}\right) = e\left(P_{x,5}, C_{x,4}\right)\end{aligned} \tag{13}$$

when Equation (12) and Equation (13) are true, it indicates that if $P_{x,5}$ is correctly constructed, then $P_{x,4}$ must contains $g^{a_{\rho(x)}}$, which can only be assigned by the authority. This means that the worker does have the corresponding $WSK_{\rho(x),id,z}$.

If the public verification succeeds, the next part will be verified by the requester. Since the requester owns the private key corresponding to the $PK_{\text{requester}}$, he can recover the plaintext of the $P_{x,7}, P_{\text{account}}$. First the requester checks whether the anonymous account claimed in the P_{account} is not in the revoked account set $Rset_{\text{account}}$ and does not equal any other account that accepts this task, then he checks the following equation:

$$\frac{e\left(P_{x,7},u\right)}{e\left(P_{x,2},C_{x,4}\right)}=\frac{e\left(u^{a_{\rho(x)}t_x k},u\right)}{e\left(u^{t_x},u^{a_{\rho(x)}k}\right)}=1 \tag{14}$$

$$\frac{e\left(P_{x,5},C_{x,4}\right)}{e\left(P_{\text{account}},P_{x,7}\right)\cdot e\left(P_{\text{account}},u\right)}=\frac{e\left(g^{zt_x+\frac{z}{a_{\rho(x)}k}},u^{a_{\rho(x)}k}\right)}{e\left(g^{z},u^{a_{\rho(x)}t_x k}\right)\cdot e\left(g^{z},u\right)}=1 \tag{15}$$

Equation (14) and Equation (15) can prove the correct construction of $P_{x,5}$ and its relevance with P_{account}, which means those attributes in the above public verifications does belong to this anonymous account. If the above check fails, the requester can reveal the plaintext of $P_{x,7}, P_{\text{account}}$, then the contract can repeat this process to determine which entity is misbehaving. Otherwise, the requester accepts the worker's participation and the task assignment process ends.

5. Security Analysis

In this section, we will analyze our protocol can preserve the security of worker and requester information.

5.1. Task Content Security

First we discussed that the task content can only be decrypted by the workers who fully satisfy the access policy, since our scheme is based on ABE with multiple authorities [22]. This part of our scheme is under the same security level. We first analyzed the case where the worker cannot satisfy the task's access policy, that is, for any combination of attributes that satisfy the access policy, the worker does not own the credential keys corresponding to the all attributes in the combination. In this case, the worker cannot find any subset of R_x that can satisfy $\sum_x c_x R_x = (1,0,\cdots,0)$, then there is negligible probability to compute $e(g,g)^s$.

Next we discussed that multiple workers cannot collude to access task content that they cannot access individually. Suppose that there is a group of workers, for any combination of attributes that satisfy the access policy, there do not exist worker who has the credential keys corresponding to all attributes in the com-

bination. But there is at least one combination, the credential keys owned by multiple workers can satisfy all the attributes in the combination. However, as shown in Equation (7), the intermediate result TK_x calculated using the credential key contains $e\left(g, \mathrm{H}\left(W_{id}\right)\right)$. Since different workers have different W_{id}, different workers have different TK_x for a *x*. Therefore, when a worker lacks a credential key, it cannot be replaced by another worker's. This shows that even through collusion, workers cannot decrypt tasks that they cannot decrypt individually.

In our scheme, the credential key $WSK_{i,id,z}$ is used in the construction of the proof. We analyze an extreme case where an entity knows all the information of a qualified worker except the secret key of the anonymous account USK_z. This scenario is reasonable, because the key is the only secret that the worker will not share with others. In this case, the entity can perform the first three steps of the task decryption process, but in step four, the entity cannot calculate the task content M according to Equation (8) due to the lack of USK_z. So the entity cannot decrypt the task content.

5.2. Worker Identity Security

The identity information of the workers is anonymous account UPK_z and global identity W_{id}. So we discuss whether other entities can get information about these two parameters from the worker's proof. As to W_{id}, it exists in the form of $H\left(W_{id}\right)^k \cdot g^d$ in the proof, where g^d is a one-time pad and is only known to the worker himself, so worker's global identity cannot be obtained by any other entity from the proof.

Next we discuss that the authorities and other workers cannot obtain information about the anonymous account from the worker's proof. The anonymous account information of the worker only exists in $P_{x,4}$ and $P_{x,5}$. In $P_{x,4}$, due to the use of different primitives g, u and the decisional diffie-hellman inversion problem (DDHI), $g^{\frac{1}{t_x}}$ cannot be calculated with u^{t_x}. So $P_{x,4}$ can only perform pairing operation with $P_{x,2}$ to remove the $g^{\frac{1}{t_x}}$ and get the value containing $e\left(g,u\right)^{a_{\rho(x)}kt_xz}$. Note that the authority does not collude with other entities and cannot decrypt the task, so it does not know the value of *k*. Although authority knows the secret key of attribute a_i, based on the decisional bilinear diffie-hellman problem (DBDH), the authority cannot distinguish $e\left(g,u\right)^{a_{\rho(x)}kt_xz}$ from a random value with the knowledge of u^{t_x}, u^k, g^z. Similarly, as to other workers, although they can decrypt the task content to know the value of *k*, but they do not know the secret key of the attribute. So they cannot distinguish $e\left(g,u\right)^{a_{\rho(x)}kt_xz}$ from a random value with the knowledge of $u^{t_x}, u^{a_{\rho(x)}}, g^z$.

In $P_{x,5}$, based on the DDHI problem, $g^{\frac{1}{k}}$ cannot be calculated by authorities. So authority can only perform pairing operation with $C_{x,4}$ to remove the $g^{\frac{1}{k}}$ and get the value containing $e\left(g,u\right)^{a_{\rho(x)}kt_xz}$, then fall into the same case as

$P_{x,4}$ above. For other workers, $P_{x,5}$ can be viewed as $g^{z\cdot\left(t_x+\frac{1}{a_{\rho(x)}k}\right)}$. If they want to distinguish which anonymous account g^z is used in $P_{x,5}$ with the paring operation, he needs to know the value $g^{t_x+\frac{1}{a_{\rho(x)}k}}$ or $u^{t_x+\frac{1}{a_{\rho(x)}k}}$. However, neither of these values can be calculated using $u^{t_x}, g^{\frac{1}{a_{\rho(x)}}}, k$. In addition, $P_{x,5}$ only has a meaningful pairing operation with $C_{x,4}$, but the result will become the same case as $P_{x,4}$ described above.

6. Performance Evaluation

In this section, we used computational cost as a metric to analyze the performance of our scheme. We used the JPBC library [36] Ver. 2.0.0 as an implementation of cryptographic operations. The implementation used 160-bit elliptic curve group on the curve $y^2 = x^3 + x$ over a 512-bit finite field. All processes were evaluated using a single thread of AMD ryzen CPU.

6.1. Requester's Computational Cost

In our scheme, the requester's calculation was mainly divided into two parts: task publication and proof verification (partial). In **Figure 4(a)**, we illustrated the computational overhead of the requester in a task with only one worker. We used the calculation time as the *y*-axis and the number of attributes included in the task as the *x*-axis. Note that the number of attributes that appear in the proof is related to the access policy. Here we took the worst case that requires all attributes. It can be seen from **Figure 4(a)** that the task encryption time and the proof verification time increase linearly according to the number of attributes. Although single verification time is within a reasonable range, this may become a major burden on the requester as the number of workers increases.

6.2. Worker's Computational Cost

The computational overhead of workers was also divided into two parts: task decryption and proof publication. In **Figure 4(b)**, we described the computational overhead of the worker in two processes, the *y*-axis represents the computation time, and the *x*-axis represents the attribute number used in the process. As shown in **Figure 4(b)**, both task decryption and proof publication increase linearly with the number of attributes. The process of task decryption results in cost saving in computation, which is consistent with the fact that worker needs to decrypt a large number of tasks for selection. In contrast, the cost of proof publication is high, but it is still reasonable compared to the time required for workers to complete their tasks.

6.3. Authority's Computational Cost

In **Figure 4(c)**, we compared the proportion of computing overhead between the

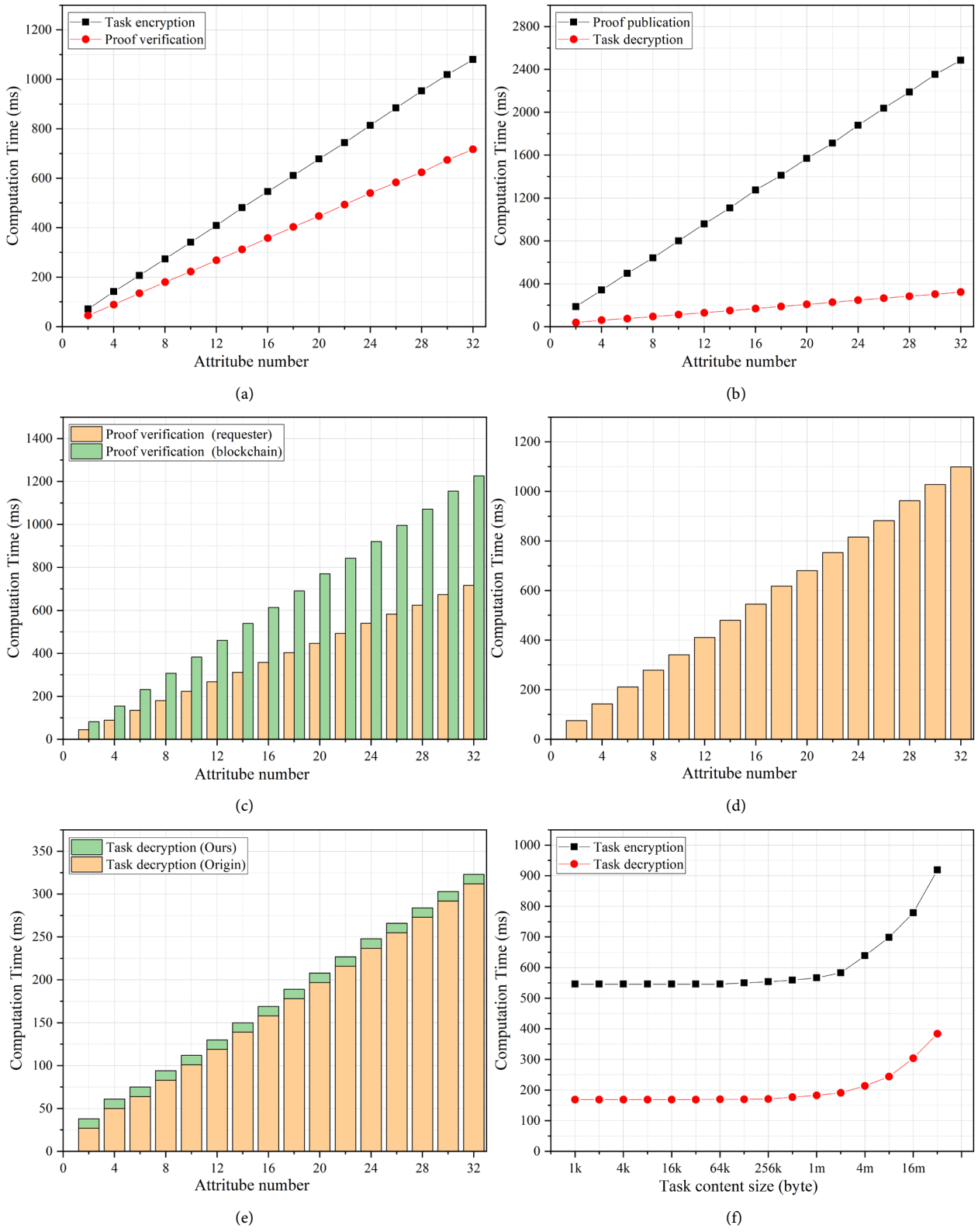

Figure 4. Calculation cost. (a) requester; (b) worker; (c) contract; (d) authority; (e) anonymous account cost; (f) effectiveness.

blockchain and the requester in the verification work, *y*-axis is the calculation time and the *x*-axis is the amount of attribute used in the proof. It can be seen

that the overhead of the verification work increases linearly according to the number of attributes, and the proportion of both remains constant at around 5:3. That is to say, although we had shifted more than half of the burden to the blockchain verifier and greatly reduced the computing time on the requester, the ratio is not large enough for the requester to easily deal with its work.

Next, we described the overhead of the authority to distribute the new key to the worker in Figure 4(d), the *y*-axis is the calculation time and the *x*-axis is the amount of attribute the worker has. It can be seen that although the calculation time increases linearly with the number of attributes, the computation operation is quite fast. This means that the worker can change the anonymous account after each task is completed, which does not incur too much cost to the authority.

6.4. Effectiveness

In Figure 4(e), we studied the performance impact on introducing anonymous accounts into ABE. Since our modification only affects the decryption process, so we compared the computational overhead of decryption with the origin scheme. The *y*-axis is the calculation time and the *x*-axis is the amount of attribute used in the decryption. It can be seen that our scheme introduces only a constant cost and is negligible relative to the overall decryption overhead.

Finally, we discussed the effect of the size of the task content on calculation overhead in Figure 4(f), the *y*-axis is the calculation time and the *x*-axis is the size of the task content. The number of attributes is set to 16. As shown in the Figure 4(f), when the size of the task content is less than 2 m, the impact of the size on the calculation time is less than 10%, which is not a key factor affecting the overhead. Obviously, this size is too small for files such as pictures and videos. Note that the contract will not process this data, it is sufficient to store only a description of how to access the actual data. In this case, 2 m is more than enough.

7. Conclusion

In this paper, we proposed a secure attribute-based task assignment scheme which can preserve information security on a transparent blockchain. First of all, the proposed scheme preserves the privacy of requesters and workers through anonymous accounts and attribute-based encryption. Second, the proposed scheme is compatible to blockchain, so as to get rid of the weakness from centralization and provide transparency. In addition, we divided the verification process into public pre-verification and requester verification, the computing burden of the requester can be greatly reduced. Finally, we analyzed the privacy and performance of the proposed protocol to show the satisfied features in both security and efficiency. In the future work, we will consider the attribute value as part of the requester's privacy for better security requirement and make a further improvement on the performance of the task assignment scheme.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Howe, J. Crowdsourcing: A Definition. https://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing/_a.html

[2] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009) Imagenet: A Large-Scale Hierarchical Image Database. 2009 *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 20-25 June 2009, 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[3] Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome

[4] Uber. https://www.uber.com/

[5] Minder, P. and Bernstein, A. (2012) Crowdlang: A Programming Language for the Systematic Exploration of Human Computation Systems. In: *Proceedings of International Conference on Social Informatics*, Springer, New York, 124-137. https://doi.org/10.1007/978-3-642-35386-4_10

[6] Geiger, D. and Schader, M. (2014) Personalized Task Recommendation in Crowdsourcing Information Systems—Current State of the Art. *Decision Support Systems*, **65**, 3-16. https://doi.org/10.1016/j.dss.2014.05.007

[7] Zhou, D., Liu, Q., Platt, J.C., Meek, C. and Shah, N.B. (2015) Regularized Minimax Conditional Entropy for Crowdsourcing. arXiv preprintarXiv:1503.07240.

[8] Shah. N. and Zhou, D. (2016) No Oops, You Won't Do It again: Mechanisms for Self-Correction in Crowdsourcing. 2016 *International Conference on Machine Learning*, New York, 19-24 June 2016, 1-10.

[9] Ouyang, R.W., Kaplan, L.M., Toniolo, A., Srivastava, M. and Norman, T.J. (2016) Parallel and Streaming Truth Discovery in Large-Scale Quantitative Crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems*, **27**, 2984-2997. https://doi.org/10.1109/TPDS.2016.2515092

[10] Zhang, Y. and Van der Schaar, M. (2012) Reputation-Based Incentive Protocolsin Crowdsourcing Applications. 2012 *Proceedings IEEE INFOCOM*, Orlando, FL, 25-30 March 2012, 2140-2148. https://doi.org/10.1109/INFCOM.2012.6195597

[11] Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B. and Allahbakhsh, M. (2018) Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques and Assurance Actions. *ACM Computing Surveys* (*CSUR*), **51**, 7. https://doi.org/10.1145/3148148

[12] Tang, W., Zhang, K., Ren, J., Zhang, Y. and Shen, X.S. (2019) Privacy-Preserving Task Recommendation with Win-Win Incentives for Mobile Crowdsourcing. *Information Sciences*, 1-7.

[13] Uber China Statement on Service Outage. http://shanghaiist.com/2015/04/18/uber/chinese/operations/recently/hacked.php/

[14] Frenkel, S. Uber Hid 2016 Breach, Paying Hackers to Delete Stolen Data. https://www.nytimes.com/2017/11/21/technology/uber-hack.html

[15] Wood, G., *et al.* (2014) Ethereum: A Secure Decentralized Generalised Transaction Ledger. *Ethereum Project Yellow Paper*, **151**, 1-32.

[16] Cachin, C. (2016) Architecture of the Hyperledger Blockchain Fabric. *Workshop on*

Distributed Cryptocurrencies and Consensus Ledgers, **310**, 4.

[17] Li, M., Weng, J., Yang, A., Liu, J.-N. and Lin, X. (2019) Towards Blockchain Based Fair and Anonymous Ad Dissemination in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, **68**, 11248-11259. https://doi.org/10.1109/TVT.2019.2940148

[18] Weng, J.-S., Weng, J., Li, M., Zhang, Y. and Luo, W. (2018) Deepchain: Auditable and Privacy-Preserving Deep Learning with Blockchain-Based Incentive. *IACR Cryptology ePrint Archive*, **2018**, 679. https://doi.org/10.1109/TDSC.2019.2952332

[19] Kang, J., Yu, R., Huang, X., Wu, M., Maharjan, S., Xie, S. and Zhang, Y. (2018) Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. *IEEE Internet of Things Journal*, **6**, 4660-4670. https://doi.org/10.1109/JIOT.2018.2875542

[20] Jung, T., Li, X.-Y., Wan, Z. and Wan, M. (2014) Control Cloud Data Access Privilege and Anonymity with Fully Anonymous Attribute-Based Encryption. *IEEE Transactions on Information Forensics and Security*, **10**, 190-199. https://doi.org/10.1109/TIFS.2014.2368352

[21] El Kaafarani, A., Ghadafi, E. and Khader, D. (2014) Decentralized Traceable Attribute-Based Signatures. In: *Cryptographers' Track at the RSA Conference*, Springer, New York, 327-348. https://doi.org/10.1007/978-3-319-04852-9_17

[22] Lewko, A. and Waters, B. (2011) Decentralizing Attribute-Based Encryption. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, New York, 568-588. https://doi.org/10.1007/978-3-642-20465-4_31

[23] Sahai, A. and Waters, B. (2005) Fuzzy Identity-Based Encryption. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, New York, 457-473. https://doi.org/10.1007/11426639_27

[24] Goyal, V., Pandey, O., Sahai, A. and Waters, B. (2006) Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In: *Proceedings of the* 13*th ACM Conference on Computer and Communications Security*, ACM, New York, 89-98. https://doi.org/10.1145/1180405.1180418

[25] Bethencourt, J., Sahai, A. and Waters, B. (2007) Ciphertext-Policy Attribute-Based Encryption. 2007 *IEEE Symposium on Security and Privacy* (*SP*'07), Oakland, CA, 20-23 May 2007, 321-334. https://doi.org/10.1109/SP.2007.11

[26] Božovic, V., Socek, D., Steinwandt, R. and Villányi, V.I. (2012) Multi-Authority Attribute-Based Encryption with Honest-But-Curious Central Authority. *International Journal of Computer Mathematics*, **89**, 268-283. https://doi.org/10.1080/00207160.2011.555642

[27] Chase, M. (2007) Multi-Authority Attribute Based Encryption. In: *Theory of Cryptography Conference*, Springer, Heidelberg, 515-534. https://doi.org/10.1007/978-3-540-70936-7_28

[28] Müller, S., Katzenbeisser, S. and Eckert, C. (2008) Distributed Attribute-Based Encryption. In: *International Conference on Information Security and Cryptology*, Springer, Heidelberg, 20-36. https://doi.org/10.1007/978-3-642-00730-9_2

[29] Lin, H., Cao, Z., Liang, X. and Shao, J. (2008) Secure Threshold Multi Authority Attribute Based Encryption without a Central Authority. In: *International Conference on Cryptology in India*, Springer, Heidelberg, 426-436. https://doi.org/10.1007/978-3-540-89754-5_33

[30] Freelancer. https://www.freelancer.com/

[31] Upwork. https://www.upwork.com/

[32] Salehi, N., Irani, L.C., Bernstein, M.S., Alkhatib, A., Ogbe, E., Milland, K., *et al.* (2015) We Are Dynamo: Overcoming Stalling and Friction in Collective Action for Crowd Workers. In: *Proceedings of the* 33*rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, New York, 1621-1630. https://doi.org/10.1145/2702123.2702508

[33] Liu, A., Wang, W., Shang, S., Li, Q. and Zhang, X. (2018) Efficient Task Assignment in Spatial Crowdsourcing with Worker and Task Privacy Protection. *GeoInformatica*, **22**, 335-362. https://doi.org/10.1007/s10707-017-0305-2

[34] Li, M., Weng, J., Yang, A., Lu, W., Zhang, Y., Hou, L., Liu, J.-N., Xiang, Y. and Deng, R.H. (2018) Crowdbc: A Blockchain-Based Decentralized Framework for Crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems*, **30**, 1251-1266. https://doi.org/10.1109/TPDS.2018.2881735

[35] Lu, Y., Tang, Q. and Wang, G. (2018) Zebralancer: Private and Anonymous Crowdsourcing System Atop Open Blockchain. 2018 *IEEE* 38*th International Conference on Distributed Computing Systems* (*ICDCS*), Vienna, Austria, 2-6 July 2018, 853-865. https://doi.org/10.1109/ICDCS.2018.00087

[36] Java Pairing-Based Cryptography Library. http://gas.dia.unisa.it/projects/jpbc/