Hindawi
Security and Communication Networks
Volume 2022, Article ID 6754744, 13 pages
https://doi.org/10.1155/2022/6754744

Research Article

Privacy-Preserving Task Distribution Mechanism with Cloud-Edge IoT for the Mobile Crowdsensing

Liquan Jiang and Zhiguang Qin

School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Correspondence should be addressed to Liquan Jiang; helenjlq@hotmail.com

Received 21 February 2022; Revised 16 May 2022; Accepted 29 August 2022; Published 26 September 2022

Academic Editor: Jinbo Xiong

Copyright © 2022 Liquan Jiang and Zhiguang Qin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile crowdsensing under big data provides an efficient, win-win, and low-budget data collection solution for IoT applications such as the smart city. However, its open and all access scenarios raise the threat of data security and user privacy during task distribution of mobile crowdsensing. To eliminate the above threat, this paper first designs a privacy-preserving task distribution scheme (Scheme 1), which realizes fine-grained access control and the practical keyword search, as well as protects the access policy. But it incurs expensive computational and communication consumptions for the task performer side. In this regard, we construct Scheme 2 to attain a lightweight trapdoor generation and keyword search mechanism, and it enables the crowdsensing platform to predecrypt a ciphertext without revealing any information about the task and the performer's privacy. Then, the resource-constrained device on the task performer side can recover the task with a few computational and communication overheads. The security of the scheme has been detailedly proved and analyzed, and theoretical comparisons and experiment demonstrate their practicability.

1. Introduction

The Internet of Things (IoT) [1, 2] paradigm realizes timely response to events and real-time collection and processing of huge amounts of data by connecting a large number of intelligent sensing devices with communication, storage, and computing capabilities through wireless sensor networks (WSN) [3]. Benefiting from the distributed network architecture and the potential raised by massive data, IoT is expected to promote innovation and development in many fields, improve user experience, and explore higher management levels. More specifically, as a public service blueprint supported by big data [5], many fields of smart city construction (city governance, smart transportation [4], smart medical care [6, 7], for instance) are expected to benefit greatly from the deployment of IoT. In addition to relying on the widely deployed sensing devices (including sensors, surveillance cameras, and GPS devices) in urban to monitor and collect massive amounts of data in real time, by introducing the mobile crowdsensing schema, residents are encouraged to actively participate in city governance and use their smart mobile devices (such as smartphone) to capture and upload events that are hard to detect, is considered to be a low-cost and emerging trend in the IoT-oriented smart city construction [8].

The mobile crowdsensing systems can be categorized as "participatory" and "opportunistic" according to task allocation strategy [8]. In a typical opportunistic mobile crowdsensing system [9], the system can adaptively assign optimal sensors to collect sensing data based on the operational scenario. This strategy guarantees the efficiency and accuracy of data collection, but at the cost of system flexibility and resident participation. As a flexible crowdsensing strategy emphasizes open participation, participatory mobile crowdsensing enables the urban administrator to publish some "tasks" in the crowdsensing platform, and then, any resident owns a fair opportunity to bid for these tasks. In a more professional view, as illustrated in Figure 1 [10], the

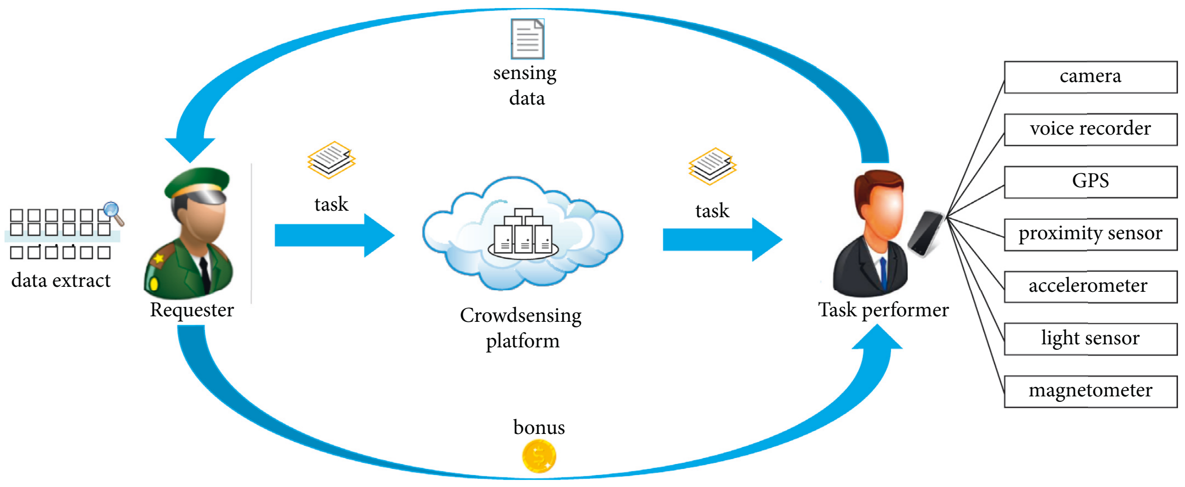

Figure 1: An overview of mobile crowdsensing system.

urban administrator acts as the requester, designs, and releases tasks to the crowdsensing platform. The task performer played by the resident scans the crowdsensing platform, chooses, and then subscribes to an available task. The task performer then executes his/her task and gets a reward by collecting via various embedded sensors and uploading sensing data within a specified time. Finally, the requester aggregates and filters the sensing dataset to obtain the optimal subset [11].

Featured with public access accessibility, low cost, and efficiency, mobile crowdsensing system wins widespread popularity and has been increasingly deployed in the public utilities IoT applications. On the other hand, however, privacy and data security issues in practical scenarios raise as a broad concern. Data circulating in the mobile crowdsensing system including task messages released by requesters and the sensing data collected by task performers, and it is possible for the attacker to conduct an attack by exploiting these two types of data. Specifically speaking, for instance, attackers may extract and analyze the weaknesses of urban facilities from the task message and then further break the vulnerable infrastructure such as the power grid system. On the other side, the attacker can also reveal the privacy information (for example, the home address, commutes route) of task performers from the sensing data [12]. For the sensing data, a public-key cryptosystem [13] is evaluated to be a feasible privacy protection strategy; that is, the task performer encrypts the sensing data with the public key previously released by the requester, and then, the requester can decrypt it to recover the plaintext-form sensing data [14]. However, the same cryptographic measure may not work for the task message, since the requester cannot predict which specific task performer would take over a task, and she/he is restricted to only encrypting and uploading a task message after confirming each task performer, which would reduce the efficiency of the mobile crowdsensing system. Besides, this measure also arouses worries about computational efficiency and the identity privacy of task performers. That is, for a task that requires multiple participants, the requester has to separately encrypt the same task message with the public key of each task performer, which incurs heavy computational overheads. And in this circumstance, the fact that the requester reveals which task performers subscribed to the task (in other words, breaks their identity privacy) is also self-evident.

Fortunately, attribute-based encryption (ABE) [15] provides the fine-grained, one-to-many, and privacy-preserving access control mechanism and is expected to break the above efficiency and privacy dilemma of the crypto-supported mobile crowdsensing system. In a typical ABE scheme, the task performer is labeled with a set of descriptive attributes, while a task message is encrypted with a specified attribute structure. A task performer can recover the task message if and only if his/her attribute set satisfies the access structure. This implies that the requester is just required to assign an [16–20] access structure and encrypt the task message for one time, and then, all task performers whose attribute sets satisfy the access structure are authorized to access the task message. In the process, the requester can reveal nothing about the task performer's identity, while only knowing s/he is an anonymous performer who holds a certain set of attributes.

Inspired by that, the latest research works put forward the solutions for the secure task distribution in mobile crowdsensing. However, there exist some gaps between these theoretically feasible solutions [10, 22, 24, 26, 36] and the practicality of mobile crowdsensing. Specifically, first, there is a practical issue that task performers have to locate their desired task among the stored ciphertext before downloading and decryption. Although there are some ABE-based solutions [26] that support keyword search, they are also hard to be practically deployed on mobile crowdsensing platforms for their cumbersome search procedures. Secondly, existing solutions (such as [10, 22]) are subject to heavy computational and storage overheads of the mobile terminal in the decryption side. This performance issue prevents them from further deployment in the mobile crowdsensing platform. Thirdlyin current ABE-based task distribution solutions for mobile crowdsensing, the sensitive information (such as the occupation and preference) about the task performer may be exposed from the public available access policy. Wang et al. provide a solution to this issue by hiding the access policy, but their solution also suffers from unworkable and cumbersome operations. To date, no work has systematically filled the above gaps by proposing a practical solution for task distribution in the mobile crowdsensing.

1.1. Contribution. In this paper, we put forward an efficient and privacy-preserving task distribution scheme (Scheme 1) and an edge-assisted scheme (Scheme 2) for IoT-oriented mobile crowdsensing, as the complete solution to fill the above gaps between current solutions and the practicality of mobile crowdsensing. Specifically, the contributions in our paper are described as follows.

(i) *Fine-Grained Access Control with Policy Hidden for the Task Distribution.* Inheriting the feature of one-to-many and fine-grained access from ABE, our solution enables the requester to share task messages with multiple task performers with encryption for only one time. During this process, the requester cannot and needs not to confirm the identity of each task performer. We then separate the attribute value from the attribute, thus hiding the specific attributes contained in the access structure, for preventing the sensitive information of the task performer from being leaked.

(ii) *Lightweight Keyword Search for the Encrypted Tasks.* It is obviously impractical for a task performer to search his/her desired ciphertext by downloading all ciphertext and seriatim decrypting them. To content the requirement that the task performer retrieves desired ciphertexts without decrypting, Scheme 1 designs the ciphertext keyword retrieval mechanism, in which the requester chooses a keyword that is associated with the task and then generates an "index," and the task performer computes a "trapdoor" with his interested keyword. The crowdsensing platform can search those related ciphertexts by using the trapdoor, and during the process, the crowdsensing platform cannot learn any information about the task and the keyword. In Scheme 2, we further improve Scheme 1 by reducing the computational and communication cost of trapdoor generation and delivery for the performer side, as well as the cost of search on ciphertexts for the crowdsensing platform.

(iii) *Lightweight Decryption Operations.* To alleviate the computational overheads of the resource-constrained device on the task performer side, we delegate the decryption operation that should be assumed by the performer to the edge device in Scheme 2. Distinct from [10], in our solution, the edge device is considered to be semitrusted, which implies that we can prevent it from obtaining sensitive information including the user secret key.

1.2. Related Works. A sequence of solutions has been designed for the issues of data security and privacy protection of the task distribution phase in IoT-oriented mobile crowdsensing recently. Tao et al. [16] presented an anonymous bilateral authentication mechanism to guarantee the data authenticity while protecting the task performer's identity privacy. Besides, they designed to solve the problem of large-scale key management in practical scenarios by using the pseudonym set. The requester usually requires the location of task performers to optimize task allocation, but it may reveal the location privacy of task performers. To protect the location privacy, Wang et al. [17] presented to fuzz the accurate location under the differential privacy constraint, thus protecting the location privacy. Karati and Biswas [18] proposed to simultaneously protect the data confidentiality and authenticity by inducing the identity-based encryption and designated verifier signature scheme. They also removed all pairing operations in their solution to improve the performance of cryptographic calculations. Ni et al. [19] proposed the SPOON scheme to attain the privacy protection of mobile users for task allocation. Specifically, it guarantees the confidentiality and authenticity of tasks with proxy reencryption and BBS+ [21] signature and uses an anonymous mechanism to protect the mobile user's identity privacy.

Motivated by the feature of fine-grained and one-to-many access control of ABE, there are some works designed to integrate ABE into the data security and privacy protection strategy in mobile crowdsensing task distribution. Zhang et al. [36] proposed an ABE scheme with direct user revocation, which provides fine-grained access control on the encrypted time-sensitive task message for hierarchical task performers in a mobile crowdsensing system. Xue et al. [22] realized fine-grained and forward secure task access control in mobile crowdsensing by integrating ABE with Bloom filter encryption [23]. Besides, a "puncture" mechanism is imposed to the user secret key to prevent key reuse. Nkenyereye et al. [24] put forward a secure protocol based on ABE for mobile crowdsensing in the fog-based vehicular cloud [25], which supports policy updates for the fine-grained access control. Besides, they simultaneously protect the data authenticity and identity privacy with the pseudo-identity-based signature mechanism. To content the practical requirement that task performers search for some interested ciphertexts without decrypting them, Miao et al. [26] designed an ABE scheme with multikeyword search for mobile crowdsensing, and it realizes the flexible and comparable attribute access control by using the 1-encoding and 0-encoding technology [27]. Miao et al. [28] then presented a universal ABE scheme with ciphertext keyword search under the shared multiple data owners setting, and they also designed to hide the access policy to prevent the privacy information of data users from being revealed from their attributes. However, this scheme suffers from expensive computational consumption of the operations of ciphertext keyword search and decryption. Also aiming at attribute privacy, Zeng et al. [29] proposed a secure data sharing scheme for the medical IoT based on the partially policy-hidden ABE. In addition, it supports scalable flexibility and security: specifically, it is available for large attribute universe and user decryption key trace. Han et al. [30] proposed an ABE scheme with a similar partial policy-hiding mechanism, which also provides the user revocation and user decryption tracing to prevent the maliciously key leakage of a data user. Phuong et al. [44] put forward a fully policy-hidden ABE scheme, and as its name implies, it reveals nothing about the attributes in the access policy. But it is evaluated to be too inefficient to be practically deployed for its cumbersome algorithm structure. On this basis, Zhang et al. [31] alleviated the decryption overheads by applying the secure outsourced computing technology, but the system still cannot escape from the complicated algorithm structure. To address this problem, more recently, Ying et al. [32] constructed a novel fully policy-hidden ABE scheme with significant efficiency improvements by their designed security-enhanced Attribute Cuckoo Filter. This scheme also subtly integrates policy hiding into a policy update system.

Focus on the efficiency improvement, Tang et al. [33] indicated to alleviate the computational overheads of task encryption and decryption phase with online/offline encryption [34] and outsourced decryption [35] technology and designed to recommend the optimal task for task performers with the claimed "win-win" strategy. More recently, Wang et al. [10] presented a fine-grained access control protocol for the mobile crowdsensing platform, which enables the lightweight keyword generation and search and specifies the crowdsensing platform to predecrypt the ciphertext to reduce the computational overheads of the task performer side. However, their proposal is insecure unless they assumed the crowdsensing platform to be fully trusted, since they direct deliver the performer's user secret key to the crowdsensing platform.

1.3. Organizations. The remainder of this paper is organized as follows: Section 2 enumerates the preliminaries of our work, Section 3 presents the first scheme, Section 4 analyzes the security of the first scheme, Section 5 describes the second lightweight scheme, Section 6 evaluates the performance, and Section 7 concludes our work.

2. Preliminaries

2.1. Basic Concepts. **Bilinear Map**. Suppose G and G_T are two cyclic groups with prime order p, and g is the generator of G. A bilinear map e: $G \times G \longrightarrow G_T$ satisfies the following properties:

(i) Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$, where $g \in G$, and $a, b \in Z_p$

(ii) Non-degeneracy: $e(g, g) = 1$

(iii) Computaility: there exists an efficient algorithm to compute $e(g, g)$ for any $g \in G$

Hardness Assumption. Assume $a, b \in Z_p^*$, and $g \in G$ is selected as a generator of group G. The decisional Diffie-Hellman (DDH) assumption is described as follows: given a three tuple (g, g^a, g^b, R), there exists a probabilistic polynomial time (PPT) algorithm to determine whether $R = g^{ab}$ or R is a random element from group G.

2.2. System and Security Model. This paper designs an interactive system that involves four entities: the trusted authority (TA), the requester, the task performer, and the crowdsensing platform. TA is a fully trusted entity, which is responsible for initializing the system and distributing the user secret key according to the attribute set for each task performer. The requester is designed as a fully trusted entity that uploads the ciphertext (encrypted task message) to the crowdsensing platform. The task performer searches for his/her interested encrypted task message and then recovers the task message. The crowdsensing platform is used for storing the ciphertext and retrieving the keyword-related ciphertext for the task performer. It is also powerful enough to assist the resource-constrained task performer to decrypt the ciphertext. It is evaluated as a semitrusted entity; that is, it can honestly execute the cryptographic protocol but is curious about the sensitive information of its stored data and the user's privacy.

Figure 2 illustrates the workflow of the proposed system, where the specially designed algorithms for our Scheme 2 are denoted with blue dotted boxes. In logical order, the TA first runs the **Setup** algorithm to initialize the whole system, and then, it performs the **KeyGen** algorithm to distribute the user secret key for each registered task performer. The requester encrypts the task message with an access policy and ties the keyword index to the ciphertext by, respectively, running the **Encrypt** and the **Index** algorithms. Then, the requester uploads the ciphertext and the index to the crowdsensing platform. If a task performer wants to take on a task, s/he first specifies an interested queried keyword and then generates a trapdoor [42–46] with his/her user secret key and the queried keyword. The task performer forwards the trapdoor to the crowdsensing platform to request all ciphertexts related to the keyword. Subsequently, by running the **Search** algorithm, the crowdsensing platform estimates whether a ciphertext satisfies the keyword requirement of the trapdoor, and then, it returns the satisfied ciphertext to the task performer. Upon receiving the ciphertext from the crowdsensing platform, the task performer recovers the plaintext-form task message from the ciphertext with his/her user secret key by running the **Decrypt** algorithm. Considering the performance constraint on the task performer side, as well as the mass data stored in the crowdsensing platform, we deploy the more efficient Scheme 2 for the resource-constrained task performer device and the crowdsensing platform, and we deploy the more efficient Scheme 2 for the resource-constrained task performer device and the crowdsensing platform. Specifically, the **KeyGen** and **Search** algorithms are reconstructed in a lightweight manner. Besides, to attain efficient decryption for the task performer, we design the **Transform**, the **TranKeyGen** algorithms, and rebuild the **Decrypt** algorithm as described in Figure 2. To be specific, following the **TranKeyGen** algorithm, the task performer first generates a transformation key based on his/her user secret key and then forwards it to

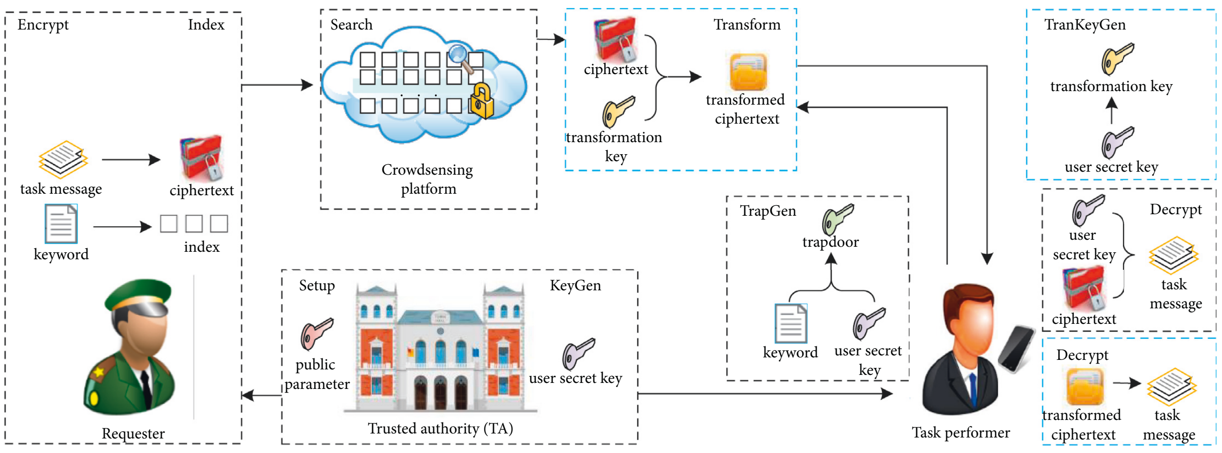

Figure 2: The system workflow.

the crowdsensing platform. By utilizing the transformation key, the crowdsensing platform runs the **Transform** algorithm to predecrypt the ciphertext and then returns the transformed ciphertext to the task performer. Finally, the task performer can execute the blue-marked **Decrypt** algorithm to recover the task message with lightweight operations.

2.3. Security Model. The basic scheme (Scheme 1) is indistinguishable under the chosen plaintext attack (IND-CPA) secure. The security model is parsed as an interactive game $\mathrm{Game}_{\mathrm{IND-CPA}}$ between a probabilistic polynomial time (PPT) adversary $\mathscr{A}$ and a challenger $\mathscr{C}$ as follows.

(i) **Initialize.** The adversary $\mathscr{A}$ specifies a challenged access structure $\mathbb{A}^*$

(ii) **Setup.** The challenger $\mathscr{C}$ runs the Setup algorithm to generate the public parameter MPK for $\mathscr{A}$

(iii) **Phase 1**. The adversary $\mathscr{A}$ queries on the user secret key of an attribute set, and then, the challenger $\mathscr{C}$ runs the **KeyGen** algorithm to generate a valid user secret key usk_W for $\mathscr{A}$

(iv) **Challenge**. The adversary $\mathscr{A}$ designates two equal-length task messages M_0 and M_1, then forwards them to $\mathscr{C}$, and $\mathscr{C}$ executes the Encrypt algorithm; that is, it randomly picks $b \in \{0,1\}$ and encrypts M_b with an access policy $\mathbb{A}$ and then returns the ciphertext CT to $\mathscr{A}$

(v) **Phase 2.** This phase is the same as Phase 1

(vi) **Guess**. $\mathscr{A}$ outputs its guess $b' \in \{0,1\}$ on b, and if $b' = b$, then we say $\mathscr{A}$ wins the game

Definition 1. If the basic scheme (Scheme 1) is indistinguishable against the chosen plaintext attack, then the probability for any PPT adversary $\mathscr{A}$ to win the above game $\mathrm{Game}_{\mathrm{IND-CPA}}$ is negligible.

Similarly, the security model of indistinguishability under the chosen keyword attack is parsed as an interactive game $\mathrm{Game}_{\mathrm{IND-CKA}}$ between a PPT adversary $\mathscr{A}$ and a challenger $\mathscr{C}$ as follows.

(i) **Initialize.** The adversary $\mathscr{A}$ specifies a challenged access structure $\mathbb{A}^*$

(ii) **Setup.** The challenger $\mathscr{C}$ runs the Setup algorithm to generate the public parameter MPK for $\mathscr{A}$

(iii) **Phase 1.** The adversary $\mathscr{A}$ answers the queries issued by $\mathscr{C}$

(iv) User secret key query. $\mathscr{A}$ queries on the user secret key of an attribute set W, and then, the challenger $\mathscr{C}$ runs the **KeyGen** algorithm to generate a valid user secret key usk_W for $\mathscr{A}$

(v) Trapdoor query. $\mathscr{A}$ queries on the trapdoor of a queried keyword q, and then, the challenger $\mathscr{C}$ runs the TrapGen algorithm to generate a valid trapdoor TD for $\mathscr{A}$

(vi) **Challenge**. The adversary $\mathscr{A}$ designates two equal-length task messages M_0 and M_1, then forwards them to $\mathscr{C}$, and $\mathscr{C}$ executes the Encrypt and Index algorithm; that is, it randomly picks $b \in \{0,1\}$ and encrypts M_b with an access policy $\mathbb{A}$ and then returns the ciphertext CT to $\mathscr{A}$

(vii) **Phase 2**. This phase is the same as Phase 1

(viii) **Guess**. A outputs its guess $b' \in \{0,1\}$ on b, and if $b' = b$, then we say $\mathscr{A}$ wins the game

Definition 2. If Scheme 2 is indistinguishable against the chosen keyword attack, then the probability for any PPT adversary $\mathscr{A}$ to win the above game $\mathrm{Game}_{\mathrm{IND-CKA}}$ is negligible.

3. Basic Scheme (Scheme 1)

This section detailedly describes the four interactive phases among those four kinds of entities in the basic scheme (Scheme 1).

3.1. System Initialization. TA runs the following Setup algorithm to establish the system and generate requisite system parameters.

Setup (λ): Taking the security parameter λ as input, TA selects two multiplicative cyclic groups $\mathbf{G}$, $\mathbf{G_T}$ with prime order $\mathbf{p}$, and $\mathbf{g}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}$ are five generators in group $\mathbf{G}$ and then define the bilinear pairing $\mathbf{e}$: $\mathbf{G} \times \mathbf{G} \longrightarrow \mathbf{G_T}$. Besides, we define a collision-resistant hash function $\mathbf{H}$: $\{0,1\}^* \longrightarrow \mathbf{Z_p}$. We set the attribute universe as $\mathbf{U} = \{\mathbf{A}_1, \ldots \mathbf{A_n}\}$. It also randomly selects $\alpha, \beta \in \mathbf{Z_p}$, then keeps secret the master secret key $\mathbf{MSK} = \alpha, \beta$, and makes the public parameter $\mathbf{MPK} = (\mathbf{p}, \mathbf{g}, \mathbf{G}, \mathbf{G_T}, \mathbf{H}, \mathbf{e}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}, \mathbf{Z} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha}, \mathbf{Z\prime} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha\beta}, \mathbf{U})$ to be publicly available.

3.2. User Registration. A newly added task performer with identity ID issues a registration request to TA. In response, by running the **KeyGen** algorithm, TA distributes the user private key usk_W for each registered task performer according to the identity ID and attribute set W.

KeyGen (MSK, MPK, W): this algorithm takes the public parameter MPK, the master secret key MSK, and the user attribute value set $W = \{W_1, \ldots, W_n\}$, where $\{W_i\} \in \{0,1\}$. For each attribute $A_i \in W$, TA picks $\{r_i\} \in Z_p$, where $i \in [1,n]$ randomly, and it also samples $r \in Z_p$. Under the above settings, TA computes $K_0 = g^{\alpha} w^r$, $K_0' = g^{\alpha\beta} w^r$, $K_1 = g^r$, $\{K_{i,2} = g^{r_i}\}_{i \in [1,n]}$, $\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\}_{i \in [1,n]}$, and TA then assembles the user secret key $usk_W = (W, K_0, K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i \in [1,n]})$ and delivers usk_W to the task performer via secure channel.

3.3. Task Encryption and Distribution. To attain secure task distribution, the requester encrypts his/her tasks and uploads the encrypted task to the crowdsensing platform by running the following described Encrypt and Index algorithm. Notice that for each task message$M$, we specify a keywordδ to enable the encrypted task can be retrieved by any task performers without revealing to irrelevant entities the detailed information about the keyword.

(i) **Encrypt** $(MPK, M, \mathbb{A})$: the requester takes the public parameter MPK, the plaintext-form task message M, the revocation list RL, and the AND-Gate access policy $\mathbb{A}$, and then, s/he randomly chooses $s, s_1, \ldots, s_{n-1} \in Z_p$ and calculates $s_n = s - \sum_{i=1}^{n-1} s_i$. The access structure $\mathbb{A}$ is instantiated to $S = \{S_1, \ldots, S_n\}$, where $\{S_i\} \in \{0,1\}$. On this basis, the requester picks $t_1, \ldots, t_n \in Z_p$ and then calculates $C_0 = M \cdot e(g,g)^{\alpha s}, C_1 = g^s$, $\{C_{i,2} = w^{s_i} v^{t_i}, C_{i,3} = g^{t_i}\}_{i \in [1,n]}$. Besides, this algorithm requires the requester to compute $C_{i,4} = (u^{A_i} h)^{-t_i}$ for each attribute $A_i \in W \cap S$, while randomly selects $C_{i,4} \in G$ for each attribute $A_i \notin W \cap S$.

(ii) **Index**(MPK, usk_W, δ): the requester assigns the most appropriate keyword by referring to the keyword dictionary for a task message M. Specifically, s/he takes as input the public key MPK, the user secret key usk_W, and the keyword δ and then invokes the collision-resistant hash function $H(\cdot)$ and generates the index as $I = Z' H(\delta)^s$. Finally, the requester assembles the ciphertext $CT = (C_0, C_1, \{C_{i,2}, C_{i,3}, C_{i,4}\}_{i \in [1,n]}, I)$ and uploads CT to the crowdsensing platform.

3.4. Task Encryption and Distribution. This phase describes the workflow on the task performer side. Specifically, s/he first generates a trapdoor about his/her skilled fields with a keyword query by running the **TrapGen** algorithm. By using the trapdoor, the crowdsensing platform locates the target encrypted task message with the **Search** algorithm and forwards it to the task performer. Finally, the task performer recovers the plaintext-form task message by running the **Decryption** algorithm.

(i) **TrapGen** (MPK, q, usk_W): the task performer inputs the public parameter MPK, the queried keyword q, and his/her user secret key usk_W and then calculates $T_0 = K' \cdot H(q)$, $T_1 = K \cdot H(q)$, and $\{T_{i,2} = K_{i,2}^{H(q)}, T_{i,2} = K_{i,3}^{H(q)}\}_{i \in [1,n]}$. The trapdoor is assembled as $TD = (T_0, T_1, \{T_{i,2}, T_{i,3}\}_{i \in [1,n]})$ and is forwarded to the crowdsensing platform via secure channel when the task performer requests a task

(ii) **Search**(MPK, TD, CT): The crowdsensing platform inputs the public parameter MPK, the trapdoor TD, and the ciphertext CT and then checks whether the following equation holds:

$$I = \frac{e(C_1, T_0)}{\prod_{i=1}^{n} \left(e(C_{i,2}, T_1) e(C_{i,4}, T_{i,2}) e(C_{i,3}, T_{i,3}) \right).} \quad (1)$$

If it holds, the crowdsensing platform returns the ciphertext CT to the task performer via the public channel, and otherwise, it aborts and feedbacks $\perp$

(iii) **Decrypt** (CT, usk_W): upon obtaining the desired ciphertext CT, the task performer takes as input his/her user secret key usk_W and recovers the plaintext-form task message M by figuring up the following equation:

$$M = \frac{\prod_{i=1}^{n} \left(e(C_{i,2}, K_1) e(C_{i,4}, K_{i,2}) e(C_{i,3}, K_{i,3}) \right)}{e(C_1, K_0)}. \quad (2)$$

4. Security Analysis

4.1. System Initialization

Theorem 1. *(IND-CPA): if a probabilistic polynomial time (PPT) adversary $\mathscr{A}$ can breach the proposed system with nonnegligible probability under the chosen plaintext attack, then a challenger algorithm$\mathscr{C}$ can be constructed to solve the DDH problem with a nonnegligible advantage*

Proof. The proof is constructed on the basis of the proof of the ciphertext policy-hidden ABE scheme in [37]. Given the four-tuple (g, g^a, g^b, R) as the input of the DDH assumption, the challenger $\mathscr{C}$ aims to determine whether $R = g^{ab}$ or a random value in the group G_T.

(i) **Initialize**: the adversary $\mathscr{A}$ claims its target AND-Gate access structure $\mathbb{A}^*$ (it can be instantiated as the value set $S^* = (S_1^*, \ldots, S_n^*)$ to be challenged).

(ii) Setup: $\mathscr{C}$ defines two multiplicative cyclic groups G, G_T with prime order p and regulates the bilinear map $e\colon G \times G \longrightarrow G_T$, where g is selected as a generator of group G, $\mathscr{C}$ then samples $x, y, z, \alpha \in Z_p$, and computes $u = g^x$, $h = g^y$, $w = g^z, Z = e(g,g)^{\alpha}$, and sets $v = g^a$. Define the attribute universe $U = \{A_1, \ldots, A_n\}$. Finally, $\mathscr{C}$ returns to $\mathscr{A}$ the public parameter $MPK = (p, g, G, G_T, e, u, h, w, v, Z, U)$ and keeps secret the master secret key $MSK = \alpha$

(iii) Phase 1: The adversary $\mathscr{A}$ issues a sequence of queries to the challenger $\mathscr{C}$ as follows. Specifically, $\mathscr{A}$ forwards to $\mathscr{C}$ an attribute set W on the premise of that $W = \{W_1, \ldots, W_n\}$ does not content the challenged AND-Gate access structure S^*. As response, $\mathscr{C}$ randomly $r, \{r_i\} \in Z_{n+1}$ and then computes $K_0 = g^{\alpha} w^r$, $K_0' = g^{\alpha\beta} w^r$, $K_1 = g^r$, $\{K_{i,2} = g^{r_i}\}_{i\in[1,n]}$, and $\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\}_{i\in[1,n]}$, and C returns $usk_W = (W, K_0, K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,n]})$ to $\mathscr{A}$.

(iv) Challenge: the adversary $\mathscr{A}$ forwards two equal-length task messages M_0^* and M_1^*. As response, the challenger $\mathscr{C}$ randomly selects $\mu \in \{0,1\}$ and implicitly sets $s = ab$ by regulating $C_0 = M \cdot e(g,g)^{\alpha s} = e(g,g)^{\alpha ab}, C_1 = R$. Assume that $A_j \notin S$, for each attribute A_i, where $i \in [1,n], i = j$, the challenger $\mathscr{C}$ randomly chooses $s_i, t_i \in Z_p$. Besides, $\mathscr{C}$ calculates $s_j = ab - \sum_{i=1, i\neq j}^{n} s_i$ and $t_j = -zb$ for the circumstance $i = j$, and $\mathscr{C}$ randomly selects $\mu \in \{0,1\}$ and implicitly sets $s = ab$ by regulating $C_0 = M \cdot e(g,g)^{\alpha s} = e(g,g)^{\alpha ab} = e(g^a, g^b)^{\alpha}$ and $C_1 = R$. If $i = j$, then $\mathscr{C}$ calculates $C_{j,2} = w^{s_j} v^{t_j} = g^{z(ab - \sum_{i=1,i\neq j}^{n} s_i)} g^{-azb} = g^{-z\sum_{i=1,i\neq j}^{n} s_i}$ and $C_{j,3} = g^{t_j} = g^{-zb} = (g^b)^{-z}$ and randomly chooses $C_{j,4}$ from group G. If $i \neq j$, $\mathscr{C}$ directly calculates $\{C_{i,2} = w^{s_i} v^{t_i}, C_{i,3} = g^{t_i}\}_{i\in[1,n]}$ and $C_{i,4} = (u^{A_i} h)^{-t_i}$.

(v) Phase 2: This phase is the same as Phase 1.

(vi) Guess: The adversary A outputs its guess $\mu' \in \{0,1\}$ on μ'. If $\mathscr{A}$ outputs $\mu' = \mu$, $\mathscr{C}$ returns 1 to guess $R = g^{ab}$. Otherwise, if $\mathscr{A}$ outputs $\mu' \neq \mu$, $\mathscr{C}$ returns 0 to guess R is a random element in group G. □

Theorem 2. *(IND-CKA): If a probabilistic polynomial time (PPT) adversary $\mathscr{A}$ can breach the proposed system with nonnegligible probability under the chosen keyword attack, then a challenger algorithm $\mathscr{C}$ can be constructed to solve the DDH problem with a nonnegligible advantage.*

Proof. Given the four-tuple (g, g^a, g^b, R) as the input of DDH assumption, the challenger $\mathscr{C}$ aims to determine whether $R = g^{ab}$ or a random value in the group G_T ,

(i) **Initialize**: The adversary $\mathscr{A}$ claims its target AND-Gate access structure $\mathbb{A}^*$ (it can be instantiated as the value set $S^* = (S_1^*, \ldots, S_n^*)$ to be challenged)

(ii) **Setup**: $\mathscr{C}$ defines two multiplicative cyclic groups G, G_T with prime order p and regulates the bilinear map $e\colon G \times G \longrightarrow G_T$, where g is selected as a generator of group G, $\mathscr{C}$ also regulates a collision-resistant hash function $H\colon \{0,1\}^* \longrightarrow Z_p$, and $\mathscr{C}$ then samples $x, y, z, \alpha \in Z_p$, computes $u = g^x$, $h = g^y, w = g^z, Z = e(g,g)^{\alpha\beta}$, and sets $v = g^a$. Define the attribute universe $U = \{A_1, \ldots, A_n\}$. Finally, $\mathscr{C}$ returns to $\mathscr{A}$ the public parameter $MPK = (p, g, G, G_T, H, e, u, h, w, v, Z', U)$ and keeps secret the master secret key $MSK = \alpha, \beta$.

(iii) **Phase 1**: The adversary $\mathscr{A}$ issues a sequence of queries to the challenger $\mathscr{C}$ as follows.

(iv) User secret key query: $\mathscr{A}$ forwards to $\mathscr{C}$ an attribute set W on the premise of that $W = \{W_1, \ldots, W_n\}$ does not content the challenged AND-Gate access structure S^*. As response, $\mathscr{C}$ randomly $r, \{r_i\} \in Z_{n+1}$ and then computes $K_0 = g^{\alpha} w^r$, $K_0' = g^{\alpha\beta} w^r$, $K_1 = g^r$, $\{K_{i,2} = g^{r_i}\}_{i\in[1,n]}$, and $\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\}_{i\in[1,n]}$, and $\mathscr{C}$ returns $usk_W = (W, K_0, K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,n]})$ to $\mathscr{A}$.

(v) Trapdoor query: $\mathscr{A}$ issues a query on the transformation key of $usk_W = (W, K_0,\ K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,n]})$ to $\mathscr{C}$, and $\mathscr{C}$ assigns a desired keyword q, calculates $T_0 = K' H(q)$, $T_1 = KH(q)$, $\{\ T_{i,2} = K_{i,2} H(q)$, $\{T_{i,3} = K_{i,3} H(q)\}_{i\in[1,n]}$, and returns the trapdoor $TD = (T_0, T_1, \{T_{i,2}, T_{i,3}\}_{i\in[1,n]})$ to $\mathscr{A}$.

(vi) Challenge: The adversary $\mathscr{A}$ forwards two equal-length task messages M_0^* and M_1^*. As response, the challenger $\mathscr{C}$ randomly selects $\mu \in \{0,1\}$ and implicitly sets $s = ab$ by regulating $I = Z' H(\delta)^s = e(g,g)^{\alpha\beta} H(\delta)^s = e(g,g)^{\alpha\beta} H(\delta)^{ab}$, $C_1 = R$. Assume that $A_j \notin S$, for each attribute A_i, where $i \in [1,n], i = j$, the challenger $\mathscr{C}$ randomly chooses $s_i, t_i \in Z_p$. Besides, $\mathscr{C}$ calculates $s_j = ab - \sum_{i=1,i\neq j}^{n} s_i$ and $t_j = -zb$ for the circumstance $i = j$. If $i = j$, the $C_{j,2} = w^{s_j} v^{t_j} = g^{z(ab - \sum_{i=1,i\neq j}^{n} s_i)} g^{-azb} = g^{-z\sum_{i=1,i\neq j}^{n} s_i}$, $C_{j,3} = g^{t_j} = g^{-zb} = (g^b)^{-z}$, and randomly chooses $C_{j,4}$ from group G. If $i \neq j$, $\mathscr{C}$ directly calculates $\{C_{i,2} = w^{s_i} v^{t_i}, C_{i,3} = g^{t_i}\}_{i\in[1,n]}$ and $C_{i,4} = (u^{A_i} h)^{-t_i}$.

(vii) Phase 2: This phase is the same as Phase 1.

(viii) Guess: The adversary $\mathscr{A}$ outputs its guess $\mu' \in \{0,1\}$ on μ. If $\mathscr{A}$ outputs $\mu' = \mu$, $\mathscr{C}$ returns 1 to guess $R = g^{ab}$. Otherwise, if $\mathscr{A}$ outputs $\mu' \neq \mu$, $\mathscr{C}$ returns 0 to guess R is a random element in group G. □

4.2. Collusion Attack Resistance

4.2. Collusion Attack Resistance. Collusion attack indicates that multiple task performers whose attribute set does not satisfy the access structure may cheat the access authorization by combining their attributes-associated user secret keys. However, collusion attack is unavailing to our proposed scheme. Notice that the user secret key is parsed as $\mathbf{K_0 = g^{\alpha} w^{r}}$, $\mathbf{K_0' = g^{\alpha\beta} w^{r}}$, $\mathbf{K_1 = g^{r}}$, $\{\mathbf{K_{i,2} = g^{r_i}}\}_{\mathbf{i}\in[1,\mathbf{n}]}$, $\{\mathbf{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}}\}_{\mathbf{i}\in[1,\mathbf{n}]}$ for the component $\mathbf{K_{i,3}}$ that corresponds to the attribute $\mathbf{A_i}$, and it is masked by the randomly selected $\mathbf{r} \in \mathbf{Z_p}$, which is various for different task performers. Thus, multiple task performers cannot obtain a valid user secret key by just combining their individual user secret keys.

4.3. Attribute Privacy Protection and Policy Hidden. We instantiate the access structure with the mechanism in [38] to attain policy hidden. Specifically, the attribute universe $\mathbf{U} = \{\mathbf{A}_1, \ldots, \mathbf{A_n}\}$ is available for each entity in the proposed system. The task performer issued the attribute value set $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W_n}\}$, while the plaintext-form task message is encrypted with another attribute value set (access policy) $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S_n}\}$. What is remarkable is that elements $\mathbf{W_i}$ and $\mathbf{S_i}$ are Boolean value or the wildcard $*$, and they just indicate whether the i-th attribute in the attribute universe U is contented for $\mathbf{W}$ or $\mathbf{S}$, or say "do not care" for the i-th attribute in $\mathbf{U}$ [38]. Therefore, (policy) attributes privacy cannot be revealed from the task performer's attribute set $\mathbf{W}$ and access policy $\mathbf{S}$.

4.4. Keyword Privacy and Unlinkability. The keyword and the queried keyword are, respectively, embedded in the ciphertext and the trapdoor in the form of $I = Z' \cdot H(\delta)^s$ and $T_0 = K' H(q)$, $T_1 = KH(q)$, $\{T_{i,2} = K_{i,2} H(q) T_{i,3} = K_{i,3} H(q)\}_{\mathrm{i}\in[1,\mathrm{n}]} \{T_{i,3} = K_{i,3} H(q)\}_{\mathrm{i}\in[1,\mathrm{n}]}$. The crowdsensing platform is unable to reveal $H(\delta)$ from I since it is masked by the secret s. Similarly, it also cannot extract $H(q)$ from those trapdoor components for its unknown of the user secret key. Besides, we assert that nobody can reveal the equality of two trapdoors from different two task performers, despite they correspond to the same queried keyword q, since each task performer secretly holds his/her unique user secret key usk_W.

5. An Improved Scheme (Scheme 2)

Motivated by [40, 41], we design a more efficient scheme for the task performer and the crowdsensing platform. This scheme provides a lightweight trapdoor generation and search mechanism and delegates most decryption operations of the task performer to the edge device [39]. In comparison to Scheme 1, on the task performer side, we alleviate the computational and communication cost of the trapdoor generation and transmission and also significantly reduce the decryption cost while, in the edge side, we eliminate similar (or repeated) computations to lower the computational cost of ciphertext keyword search.

5.1. System Initialization. Setup (λ): Taking the security parameter λ as input, TA selects two multiplicative cyclic groups $\mathbf{G}$, $\mathbf{G_T}$ with prime order $\mathbf{p}$, and $\mathbf{g}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}$ are five generators in group $\mathbf{G}$, then defines the bilinear pairing $\mathbf{e}$: $\mathbf{G} \times \mathbf{G} \longrightarrow \mathbf{G_T}$. Besides, we define a collision-resistant hash function $\mathbf{H}$: $\{0,1\}^* \longrightarrow \mathbf{Z_p}$. We set the attribute universe as $\mathbf{U} = \{\mathbf{A}_1, \ldots, \mathbf{A_n}\}$. It also randomly selects $\alpha, \beta \in \mathbf{Z_p}$, then keeps secret the master secret key $\mathbf{MSK} = (\alpha, \beta)$, and makes the public parameter $\mathbf{MPK} = (\mathbf{p}, \mathbf{g}, \mathbf{G}, \mathbf{G_T}, \mathbf{H}, \mathbf{e}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}, \mathbf{Z} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha}, \mathbf{Z'} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha\beta}, \mathbf{U})$ to be publicly available.

5.2. User Registration. **KeyGen** (MSK, MPK, W): This algorithm takes the public parameter MPK, the master secret key MSK, and the user attribute value set $W = \{W_1, \ldots, W_n\}$, where $\{W_i\} \in \{0,1\}$. For each attribute $A_i \in W$, TA picks $\{r_i\} \in Z_p$, where $i \in [1,n]$ randomly, it also samples $r \in Z_p$. Under the above settings, TA computes $K_0 = g^{\alpha} w^r$, $K_0' = g^{\alpha\beta} w^r$, $K_1 = g^r$, $\{K_{i,2} = g^{r_i}\}_{i\in[1,n]}$, $\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\}_{i\in[1,n]}$, and $K_4 = g^{\alpha\beta}$. TA then assembles the user secret key $usk_W = (W, K_0, K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,n]}, K_4)$ and delivers usk_W to the task performer via secure channel.

5.3. Task Encryption and Distribution

(i) Encrypt $(MPK, M, \mathbb{A})$: The requester takes the public parameter MPK, the plaintext-form task message M, the revocation list RL, and the AND-Gate access policy $\mathbb{A}$, and then, s/he randomly chooses $s, s_1, \ldots, s_{n-1} \in Z_p$ and calculates $s_n = s - \sum_{i=1}^{n-1} s_i$. The access structure $\mathbb{A}$ is instantiated to $S = \{S_1, \ldots, S_n\}$, where $\{S_i\} \in \{0,1\}$. On this basis, the requester picks $t_1, \ldots, t_n \in Z_p$ and then calculates $C_0 = M \cdot\ e(g,g)^{\alpha s}, C_1 = g^s$, $\{C_{i,2} = w^{s_i} v^{t_i}$, $C_{i,3} = g^{t_i}\}_{i\in[1,n]}$. Besides, this algorithm requires the requester to compute $C_{i,4} = (u^{A_i} h)^{-t_i}$ for each attribute $A_i \in W \cap S$, while randomly selects $C_{i,4} \in G$ for each attribute $A_i \notin W \cap S$.

(ii) **Index** (MPK, usk_W, δ): The requester assigns the most appropriate keyword by referring to the keyword dictionary for a task message M. Specifically, s/he takes as input the public key MPK, the user secret key usk_W, and the keyword δ and then invokes the collision-resistant hash function $H(\cdot)$ and generates the index as $I = Z' \cdot e(C_1, H(\delta))$. Finally, the requester assembles the ciphertext $CT = (C_0, C_1, \{C_{i,2}, C_{i,3}, C_{i,4}\}_{i\in[1,n]}, I)$ and uploads CT to the crowdsensing platform.

5.4. Task Search

(i) TrapGen (MPK, q, usk_W): The task performer inputs the public parameter MPK, the queried keyword q, and his/her user secret key usk_W and then calculates.

The task performer delivers TD to the crowdsensing platform via secure channel when the task performer requests a task.

(ii) **Search** (MPK, TD, CT): The crowdsensing platform inputs the public parameter MPK, the trapdoor TD, and the ciphertext CT and then checks whether the equation $I = e(TD, C_1)$ holds. If it holds, the crowdsensing platform returns the ciphertext CT to the task performer via public channel; otherwise, it aborts and feedbacks $\perp$.

5.5. Task Reveal. In this phase, we design to delegate the decryption operation to the edge device without directly handing over the user secret key. By following this idea, we blind the user secret key with a randomly selected $t \in Z_p$, and then, the edge device can transform (predecrypt) the ciphertext with the "blinded" key. Specifically, this phase performs by running the following algorithms.

(i) **TranKeyGen** (MPK, usk_W): The task performer inputs the public parameter MPK and his/her user secret key usk_W, and then, s/he picks $t \in Z_p$ and calculates $tk_0 = K_0^t$, $tk_1 = K_1^t$, $\{$ $tk_{i,2} = K_{i,2}^t$, $\{tk_{i,3} = K_{i,3}^t\}_{i\in[1,n]}$, and s/he assembles the transformation key $TK = (tk_0, tk_1, \{tk_{i,2}, tk_{i,3}\}_{i\in[1,n]})$ and forwards TK to the edge device via a public channel. Notice that the task performer is required to keep secret the parameter t.

(ii) Transform (CT, TK): Upon receiving the transformation key TK, the edge device takes as input the desired ciphertext CT and generates the transformed ciphertext by figuring up the following equation:

$$CT' = \frac{\prod_{i=1}^{n}\left(e\left(C_{i,2}, tk_1\right)e\left(C_{i,4}, tk_{i,2}\right)e\left(C_{i,3}, tk_{i,3}\right)\right)}{e\left(C_1, tk_0\right).} \quad (3)$$

(iii) **Decrypt** (CT, CT'): The task performer takes as input the ciphertext CT and the transformed ciphertext CT', and then, s/he recovers the plaintext-form task message M by computing $M = C_0 \cdot CT'^{1/t}$.

Lemma 1. *(IND-CPA): The Scheme 2 is indistinguishable against the chosen plaintext attack if the Scheme 1 is IND-CPA secure.*

Proof. We omit the detail proof since it is similar with the proof of Theorem 1. What is different is that the "transformation key query" phase should be supplemented, which enables the challenger $\mathscr{C}$ to answer a sequence of queries on the transformation key from the adversary $\mathscr{A}$. □

Lemma 2. *(IND-CKA): The Scheme 2 is indistinguishable against the chosen keyword attack if the Scheme 1 is IND-CKA secure.*

Proof. We omit the detail proof since it is similar to the proof of Theorem 2. □

6. Performance Evaluation

6.1. Functionality and Complexity. Table 1 shows the comparisons on functionality among related schemes, including ABKS-SM [28], FGTAC [10] as well as Scheme 1 and Scheme 2 proposed in this paper. As illustrated, all of these schemes provide the security proof of IND-CPA and IND-CKA. In comparison with ABKS-SM [28] and Scheme 1, FGTAC [10] and our Scheme 2 enable the AND-Gate access control, fast ciphertext keyword search, lightweight decryption, and policy hidden. However, lightweight decryption in FGTAC [10] relies on a fully trusted crowdsensing platform, which impairs its practicality. Our Scheme 2 is proposed to attain lightweight decryption for task performers under the semitrusted crowdsensing platform assumption.

Table 2 describes the comparison of the above-mentioned schemes in terms of computational and storage complexity. In addition to the functional and practical advantages, our Scheme 2 is superior to ABKS-SM [28] and FGTAC [10] in storage cost. Our Scheme 2 is also well-performed in other indicators (including user secret key generation, trapdoor generation, search, and decryption) of computational cost except for encryption cost. Of course, we need not worry about the encryption cost since it is executed by the powerful task requester.

6.2. Experiment Results. We have experimented our proposed Scheme 1, Scheme 2 as well as related schemes such as ABKS-SM [28] and FGTAC [10] to evaluate and compare their practical performance. This experiment is conducted on a personal computer with an Intel (*R*) Core(TM) i7-7500U, 2.9 GHZ CPU, and 64 bit Windows 10 OS, and it is supported by the JPBC-2.0.0 library. To attain the 80 bit security, the elliptic curve is instantiated by a supersingular curve $y^2 = x^3 + x$ on the finite field F_p with the embedding degree of 2, where the prime degree of the field F_p is $p = 12qr - 1$, and the order of group G is the 160 bit Solinas prime $q = 2^{159} + 2^{17} + 1$, and then, there exists $|G| = |G_T| =$ 128 bytes and $|Z_p^*| = 20$ bytes. Besides, we designate SHA-256 to be the hash function in the experiment. We implement our proposed Scheme 1, Scheme 2 as well as ABKS-SM [28] and FGTAC [10] on the Enron e-mail Dataset [45], which is a widely used dataset that consists of 1,227,255 emails with 493,384 attachments covering 151 custodians.

The experimental results are pictorially described in Figure 3. When evaluating computing performance, we set the number of attributes to increase from 10 to 100 at the interval of 10, and the number of attributes is set to increase from 10 to 50 with the interval of 10 while evaluating storage performance. It is worth noting that since each attribute contains multiple "attribute values" in ABKS-SM [28], for a fair comparison, we only consider the number of attribute values. Figure 3(a) illustrates the time consumption for task encryption of these four schemes, and their computational time costs grow linearly with the size of involved attributes,

Table 1: Properties comparisons.

Schemes	F1	F2	F3	F4	F5	F6
ABKS-SM [28]	LSSS, AND-gate	CPA, CKA	Semi	×	×	√
FGTAC [10]	AND-gate	CPA, CKA	Fully	√	√	√
Scheme 1	AND-gate	CPA, CKA	Semi	×	×	√
Scheme 2	AND-gate	CPA, CKA	Semi	√	√	√

Notations: F1: access structure; F2: security level; F3: security requirement of the crowdsensing platform; F4: fast search; F5: lightweight decryption; F6: policy hidden.

Table 2: Complexity comparisons.

Schemes	ABKS-SM [28]	FGTAC [10]	Scheme 1	Scheme 2
F1	$(2n + d + 3)\vert G\vert$	$(4n + 1)\vert G\vert$	$(2n + 3)\vert G\vert$	$(2n + 3)\vert G\vert$
F2	$(2n + d + 4)e_G + e_{G_T}$	$(5n + 2)e_G$	$(3n + 5)e_G$	$(3n + 5)e_G$
F3	$(\sum_{i=1}^{n} n_i + d + n + 2)\vert G\vert + 3\vert G\vert_T$	$(3n + 2)\vert G\vert + \vert G_T\vert$	$(2n + m + 1)\vert G\vert + \vert G_T\vert$	$(2n + m + 1)\vert G\vert + \vert G_T\vert$
F4	$(\sum_{i=1}^{n} n_i + 2\ d + n + 2)e_G + 3e_{G_T}$	$(3n + 2)e_G + e_{G_T}$	$(3n + 2m + 1)e_G + e_{G_T}$	$P +\ (3n + 2m + 1)e_G + 2e_{G_T}$
F5	$(2n + 1)\vert G\vert$	$2\vert G\vert$	$(2n + 2)\vert G\vert$	$\vert G_T\vert$
F6	$(2n + 1)e_G$	$2\ e_G$	$(2n + 2)e_G$	$P + e_{G_T}$
F7	$(2n + 1)P + e_{G_T}$	$2P$	$(3n + 1)P$	P
F8	$3P + de_G + de_{G_T}$	e_{G_T}	$(3n + 1)P$	e_{G_T}

Notations: F1: size of the user secret key; F2: computational cost for user secret key generation; F3: size of the ciphertext; F4: computational cost for encryption; F5: size of the trapdoor; F6: computational cost for trapdoor generation; F7: computational cost for keyword search; F8: computational cost for decryption; **n**: number of attributes; $\mathbf{n_i}$: number of possible values for an attribute $\mathbf{A_i}$; **d**: number of data owners; **m**: number of user's attributes that satisfy the access policy; $|\mathbf{G}|$: an element in group **G**; $|\mathbf{G_T}|$: an element in group $\mathbf{G_T}$; **P**: a pairing operation; $\mathbf{e_G}$: an exponential operation over the group **G**; $\mathbf{e_{G_T}}$: an exponential operation over the group $\mathbf{G_T}$.

where our Scheme 1 and Scheme 2 show slight inferiority. However, they are acceptable since the requester is regarded as a powerful device, and even in our experiment platform, they generate a ciphertext within 5 seconds while the number of attributes reaches 100. This is because our schemes are constructed over the large-universe ABE scheme for attaining the scalable of attributes size in the mobile crowdsensing application; that is, it improves the usability at a few cost of efficiency. Figure 3(b) shows that for ciphertext keyword search, our Scheme 2 outperforms Scheme 1 and ABKS-SM [28] and is similar to FGTAC [10]; that is, the computational overhead is slight and constant. The time costs of FGTAC [10] and Scheme 2 are stable with the number of attributes, and those of the above four schemes, respectively, reach 1938.543 ms, 37.266 ms, 5045.127 ms, and 20.114 ms when the attributes number reaches 100. The excellent search performance of Scheme 2 is owed to our proposed lightweight search mechanism. For each ciphertext, we require the crowdsourcing platform to perform only one pairing operation involving the trapdoor, the index, and the key ciphertext component. We can observe from Figure 3(c) that the decryption time costs for the task performer in ABKS-SM [28], FGTAC [10], and Scheme 2 are constant even if the growth of the attributes number, but that of Scheme 2 is significantly less than ABKS-SM [28]. Specifically, the decryption time cost of Scheme 1 grows with the number of attributes (it attains 4972.268 ms when 100 attributes) while the remainders keep stable, which are within 70 ms and around 10 ms. This is due to the secure outsourcing and edge computing mechanism we implemented in Scheme 2 for the ciphertext decryption. In Figure 3(d), the trapdoor generation time consumption of Scheme 2 is slight and remains stable despite the attributes number increases, which is similar to that of FGTAC [10], and is far superior to ABKS-SM [28]. Specifically, the trapdoor generation time costs of both ABKS-SM [28] and Scheme 1 grow with the attributes number, and they are 2296.451 ms and 2207.195 ms, respectively, for 100 attributes setting. The time costs of FGTAC [10] and Scheme 2 are slight and nearly constant, and both of them are within 20 ms. This phenomenon also benefits from our lightweight keyword search mechanism that only requires a short and accessible trapdoor in Scheme 2 instead of embedding the queried trapdoor to each key component in Scheme 1. Figure 3(e) illustrates the comparison among these four schemes, and their ciphertext storage cost increases with the number of attributes. However, in fact, in Scheme 2, the task performer only needs to receive and store a constant size transformed ciphertext, which reduces the storage overhead of resource-constrained devices on the task performer side. In Figure 3(f), the trapdoor storage costs of FGTAC [10] and Scheme 2 are slight and constant size, which are friendly to the resource-constrained task performer side devices, despite that is growing with the number of attributes in ABKS-SM [28] and Scheme 1. This also benefits from our designed efficient ciphertext keyword search mechanism.

In a nutshell, our Scheme 1 uses ABE as the core to achieve task confidentiality and performer's identity privacy protection. Functionally, compared with other related works on mobile crowdsourcing security task distribution, Scheme 1 hides the access policy, thus preventing the performer's privacy leakage. And it allows the performer to flexibly

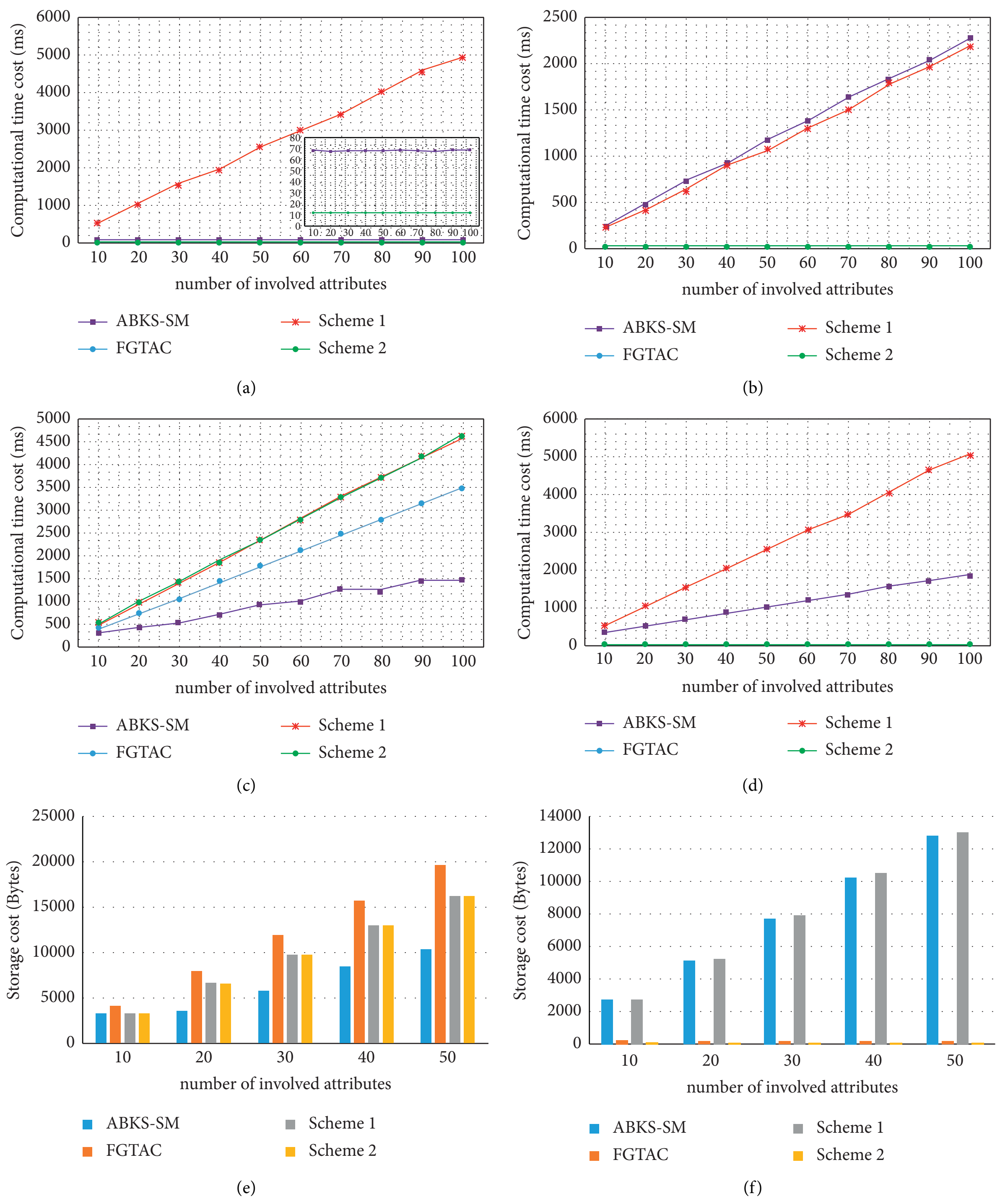

FIGURE 3: Comparisons of time consumption in the publisher/subscriber side. (a) Encryption time cost. (b) Search time cost. (c) Decryption time cost. (d) Trapdoor generation time cost. (e) Ciphertext storage cost. (f) Trapdoor storage cost.

search the encrypted tasks it is interested in without revealing any preferences by designing the ciphertext keyword retrieval mechanism. In terms of performance, on the basis of Scheme 1, Scheme 2 implements an efficient ciphertext search mechanism, which allows the performer and the crowdsensing platform to generate a trapdoor and search ciphertexts with a small and fixed computational and storage overhead, respectively. On this basis, a large number of decryption operations that originally belonged to the task performer were transferred to the edge device. Compared with other related works on mobile crowdsourcing security task distribution, it improves the computational and storage performance on the performer side and crowdsensing platform side as shown in the experiment.

7. Conclusion

This paper designed the efficient and privacy-preserving task distribution mechanism for IoT-oriented mobile crowdsensing. We show our results by two practical cryptographic schemes. Scheme 1 realizes the fine-grained access control and access policy hidden by dividing the attribute into an attribute label and an attribute value, where the attribute value is publicly available, and the attribute label is hidden. We also design a keyword search mechanism over task ciphertexts that enables the task performer to conveniently generate the trapdoor. On this basis, Scheme 2 further improves the efficiency under the semitrusted crowdsensing platform assumption by delegating most operations to the crowdsensing platform and constructing a lightweight trapdoor. We then analyzed their security properties, provided the formalized security proof, and demonstrated their practicability and feasibility.

We note that although our work prevents the sensitive information of task performers from exposure, it still falls under the category of "partial policy hiding." The authors of [44] pointed out that some ABE schemes with partial policy hiding may still reveal the performer's attribute privacy. We notice that the latest representative work has transformed the primitive of full policy-hidden ABE from the cumbersome theoretical scheme to an efficient practical solution by optimizing the algorithm structure and extending the usability [32]. Therefore, in future work, we intend to further explore the more efficient and flexible ABE schemes with full policy hiding. In addition, although we profoundly reduce the trapdoor generation overhead on the performer side and the search burden of the crowdsensing platform, it may still suffer from the performance bottleneck in the crowdsensing platform with massive storage. In future work, we would like to explore an efficient ciphertext keyword search mechanism for the above practical setting.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Major Scientific Instrument and Equipment Development Project of National Natural Science Foundation of China (62027827).

References

[1] T. Li, Y. Tian, J. Xiong, and M. Z. Bhuiyan, "FVP-EOC: fair, verifiable and privacy-preserving edge outsourcing computing in 5G-enabled IIoT," *IEEE Transactions on Industrial Informatics*, p. 1, 2022.

[2] J. Sun, Y. Yuan, M. Tang, X. Cheng, X. Nie, and M. U. Aftab, "Privacy-preserving bilateral fine-grained access control for cloud-enabled industrial IoT healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6483–6493, 2022.

[3] D. Bd and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks," *Ad Hoc Networks*, vol. 97, Article ID 102022, 2020.

[4] Y. Bao, W. Qiu, X. Cheng, and J. Sun, "Fine-grained data sharing with enhanced privacy protection and dynamic users group service for the IoV," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2022.

[5] Y. D. Zhang, S. C. Satapathy, D. S. Guttery, J. M. Gorriz, and S. H. Wang, "Improved breast cancer classification through combining graph convolutional network and convolutional neural network," *Information Processing & Management*, vol. 58, no. 2, Article ID 102439, 2021.

[6] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie, and R. H. Deng, "Lightweight and privacy-aware fine-grained access control for IoT-oriented smart health," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6566–6575, 2020.

[7] Y. Bao, W. Qiu, and X. Cheng, "Secure and lightweight fine-grained searchable data sharing for IoT-oriented and cloud-assisted smart healthcare system," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2513–2526, 2022.

[8] C. Fiandrino, B. Kantarci, and F. Anjomshoa, "Sociability-driven user recruitment in mobile crowdsensing internet of things platforms," in *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Washington, DC, USA, December 2016.

[9] Q. Liang, X. Cheng, S. C. H. Huang, and D. Chen, "Opportunistic sensing in wireless sensor networks: theory and application," *IEEE Transactions on Computers*, vol. 63, no. 8, pp. 2002–2010, 2014.

[10] J. Wang, X. Yin, and J. Ning, "Fine-grained task access control system for mobile crowdsensing," *Security and Communication Networks*, vol. 2021, Article ID 6682456, 12 pages, 2021.

[11] B. Guo, Q. Han, H. Chen, L. Shangguan, Z. Zhou, and Z. Yu, "The emergence of visual crowdsensing: challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2526–2543, 2017.

[12] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[13] Y. Sun, P. Chatterjee, Y. Chen, and Y. Zhang, "Efficient identity-based encryption with revocation for data privacy in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2734–2743, 2022.

[14] D. Wu, Z. Yang, B. Yang, R. Wang, and P. Zhang, "From centralized management to edge collaboration: a privacy-preserving task assignment framework for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4579–4589, 2021.

[15] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE symposium on security and privacy (SP'07)*, pp. 321–334, IEEE, Berkeley, CA, USA, May 2007.

[16] D. Tao, P. Ma, and M. S. Obaidat, "Anonymous identity authentication mechanism for hybrid architecture in mobile crowd sensing networks," *International Journal of Communication Systems*, vol. 32, no. 14, Article ID e4099, 2019.

[17] L. Wang, D. Yang, and X. Han, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 627–636, Geneva, Switzerland, April 2017.

[18] A. Karati and G. P. Biswas, “Provably secure and authenticated data sharing protocol for IoT-based crowdsensing network,” *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, Article ID e3315, 2019.

[19] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, “Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2020.

[20] O. A. Khashan, “Hybrid lightweight proxy re-encryption scheme for secure Fog-to-Things environment,” *IEEE Access*, vol. 8, pp. 66878–66887, 2020.

[21] M. H. Au, W. Susilo, and Y. Mu, “Constant-size dynamic k-TAA,” *International conference on security and cryptography for networks*, vol. 4116, pp. 111–125, 2006.

[22] L. Xue, J. Ni, and C. Huang, “Forward secure and fine-grained data sharing for mobile crowdsensing,” in *Proceedings of the 2019 17th International Conference on Privacy, Security and Trust (PST)*, pp. 1–9, IEEE, Fredericton, Canada, August 2019.

[23] L. L. Gremillion, “Designing a Bloom filter for differential file access,” *Communications of the ACM*, vol. 25, no. 9, pp. 600–604, 1982.

[24] L. Nkenyereye, S. R. Islam, M. Bilal, M. Abdullah-Al-Wadud, A. Alamri, and A. Nayyar, “Secure crowd-sensing protocol for fog-based vehicular cloud,” *Future Generation Computer Systems*, vol. 120, pp. 61–75, 2021.

[25] J. Sun, G. Xu, T. Zhang, H. Xiong, H. Li, and R. Deng, “Share your data carefree: an efficient, scalable and privacy-preserving data sharing service in cloud computing,” *IEEE Transactions on Cloud Computing*, p. 1, 2021.

[26] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, “Practical attribute-based multi-keyword search scheme in mobile crowdsourcing,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018.

[27] H. Y. Lin and W. G. Tzeng, “An efficient solution to the millionairesąŕ problem based on homomorphic encryption,” *International Conference on Applied Cryptography and Network Security*, vol. 3531, pp. 456–466, 2005.

[28] Y. Miao, X. Liu, K. K. R. Choo et al., “Privacy-preserving attribute-based keyword search in shared multi-owner setting,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1080–1094, 2021.

[29] P. Zeng, Z. Zhang, R. Lu, and K. K. R. Choo, “Efficient policy-hiding and large universe attribute-based encryption with public traceability for internet of medical things,” *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10963–10972, 2021.

[30] D. Han, N. Pan, and K. C. Li, “A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 316–327, 2022.

[31] L. Zhang, W. You, and Y. Mu, “Secure outsourced attribute-based sharing framework for lightweight devices in smart health systems,” *IEEE Transactions on Services Computing*, p. 1, 2021.

[32] Z. Ying, W. Jiang, X. Liu, S. Xu, and R. Deng, “Reliable policy updating under efficient policy hidden fine-grained access control framework for cloud data sharing,” *IEEE Transactions on Services Computing*, p. 1, 2021.

[33] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. Sherman Shen, “Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing,” *Information Sciences*, vol. 527, pp. 477–492, 2020.

[34] S. Hohenberger and B. Waters, “Online/offline attribute-based encryption,”vol. 8383, pp. 293–310, in *Proceedings of the International Workshop on Public Key Cryptography*, vol. 8383, pp. 293–310, Springer, Berlin, Germany, March 2014.

[35] B. Qin, R. H. Deng, and S. Liu, “Attribute-based encryption with efficient verifiable outsourced decryption,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[36] J. Zhang, J. Ma, and T. Li, “Efficient hierarchical and time-sensitive data sharing with user revocation in mobile crowdsensing,” *Security and Communication Networks*, vol. 57, pp. 34–56, 2021.

[37] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, “Attribute based encryption with privacy protection and accountability for CloudIoT,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 762–773, 2022.

[38] T. Nishide, K. Yoneyama, and K. Ohta, “Attribute-based encryption with partially hidden encryptor-specified access structures,”vol. 5037, pp. 111–129, in *Proccedings of the International Conference on Applied Cryptography and Network Security*, vol. 5037, pp. 111–129, Springer, Berlin, Germany, 2008.

[39] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, “A blockchain-based machine learning framework for edge services in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[40] J. Cui, H. Zhou, Y. Xu, and H. Zhong, “OOABKS: online/offline attribute-based encryption for keyword search in mobile cloud,” *Information Sciences*, vol. 489, pp. 63–77, 2019.

[41] Y. Bao, W. Qiu, P. Tang, and X. Cheng, “Efficient, revocable and privacy-preserving fine-grained data sharing with keyword search for the cloud-assisted medical IoT system,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2041–2051, 2022.

[42] J. Hao, C. Huang, and G. Chen, “Privacy-preserving interest-ability based task allocation in crowdsourcing,” in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Shanghai, China, May 2019.

[43] A. De Caro and V. Iovino, “jPBC: java pairing based cryptography,” in *Proceedings of the 2011 IEEE symposium on computers and communications (ISCC)*, pp. 850–855, IEEE, Kerkyra, Greece, July 2011.

[44] T. V. X. Phuong, G. Yang, and W. Susilo, “Hidden ciphertext policy attribute-based encryption under standard assumptions,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 35–45, 2016.

[45] Enron Email Data, “Enron Email Data,” 2016, https://aws.amazon.com/de/datasets/enron-email-data.

[46] Y. Rouselakis and B. Waters, “Practical constructions and new proof methods for large universe attribute-based encryption,” in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 463–474, Berlin, Germany, November 2013.