JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Based on Time and Frequency Limitation A Privacy Protection Scheme for IoT Big Data

T. Suneetha, Assistant Professor in Dept of Computer Science, Loyola Academy Degree and PG College, Alwal, Hyderabad, TS.

Abstract:

Various applications of the Internet of Things assisted by deep learning such as autonomous driving and smart furniture have gradually penetrated people's social life. These applications not only provide people with great convenience but also promote theprogress and development of society. However, how to ensure that the important personal privacy information in the big data ofthe Internet of Things will not be leaked when it is stored and shared on the cloud is a challenging issue. The main challenges include (1) the changes in access rights caused by the flow of manufacturers or company personnel while sharing and (2) the lack of limitation on time and frequency. We propose a data privacy protection scheme based on time and decryption frequency limitation that can be applied in the Internet of Things. Legitimate users can obtain the original data, while users without a homomorphic encryption key can perform operation training on the homomorphic ciphertext. On the one hand, this scheme does not affect the training of the neural network model, on the other hand, it improves the confidentiality of data. Besides that, this scheme introduces a secure two-party agreement to improve security while generating keys. While revoking, each attributeis specified for the validity period in advance. Once the validity period expires, the attribute will be revoked. By using storage lists and setting tokens to limit the number of user accesses, it effectively solves the problem of data leakage that may be caused by multiple accesses in a long time. The theoretical analysis demonstrates that the proposed scheme can not only ensure safety but also improve efficiency.

Keywords: Homomorphic ciphertext, Confidentiality, IoT Networks.

1. Introduction

The development of emerging computing technologies (e.g., cloud computing) have brought opportunity for various industries, such as hyperspectral remote sensing image algo- rithms [1, 2], classification algorithms [3], matrix operations under linear systems [4, 5], and data generated by Internet ofThings (IoT) devices. If the data in a solution is stored in the cloud or the calculation is outsourced to the cloud, the local storage and calculation pressure will be greatly reduced. Among them, for IoT big data, because IoT devices generate huge amounts of data, the structure of the traditional machine learning model is relatively simple, which can no longer meet the new needs of IoT applications. Thus, deep learning technology has been widely used in IoT applications[6], e.g., smart home [7], smart city [8, 9], and autonomous driving [10].

In the scenario of applying deep learning technology to big data in the IoT, in order to train a neural network, large amounts of data need to be obtained from the IoT devices. For example, crowdsensing systems collect data that comes from sensors embedded on personally owned mobile devices [11]. These data may contain sensitive information of some users. However, IoT networks are becoming more vulnerable to various web attacks [12]. Obviously, once they "share" these IoT data with the same field, they are likely to lose control ofthis data. If these data containing private information are leaked, and there is a lack of elective protection mechanism

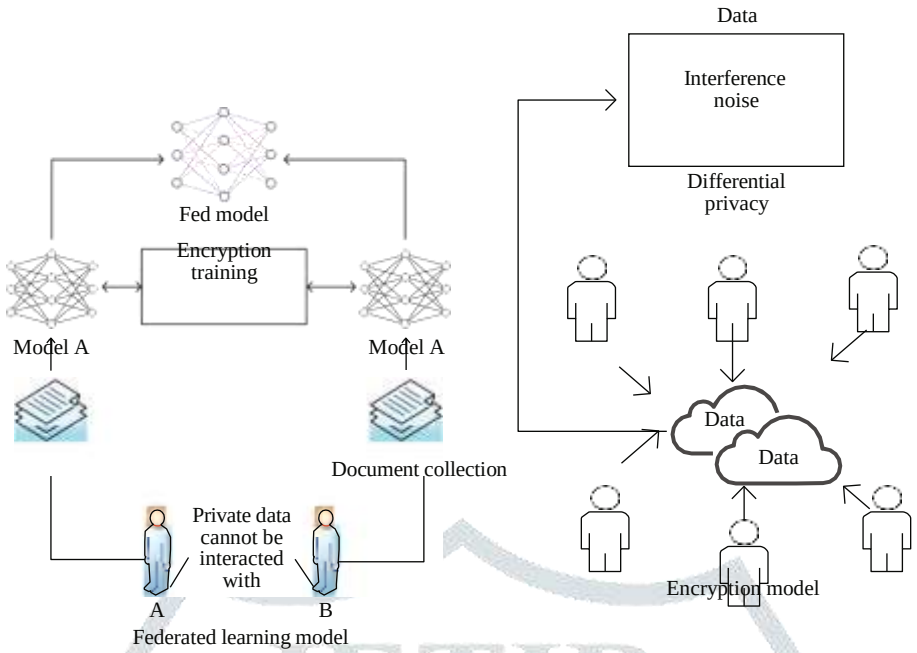

Figure 1: Three different types of working principles.

in the process of IoT search [13], it may cause irreversible harm to the people whose information is leaked. For example, in the field of healthcare, human physiological data collectedby wearable IoT devices are put into deep learning models, which can predict the physical condition of patients [14–17].Once these data are leaked, it will not only cause a patient's economic loss but also endanger life [18]. In the field of auton- omous driving, the prediction system of deep learning may bemaliciously interfered. Once location privacy data is obtained maliciously, it may cause traffic safety problems and bring troubles to society [19]. It can be seen that how to protect users' private data still faces severe challenges for projects that use deep learning to assist IoT applications, and it is a problem that must be solved.

At present, many solutions have been proposed to solve the big data privacy protection problem in machine learning [20] or deep learning. Generally, these schemes are divided into three categories: federated learning [21, 22], encryption-based technologies [23–26], and differential privacy technolo- gies [27, 28], as shown in Figure 1. Figure 1 shows the working principles of three different types of privacy protection. Among them, encryption-based technologies mainly use direct encryption of data, such as using homomorphic encryp- tion algorithms or setting access control on data uploaded to cloud servers. However, in actual situations, data owners notonly want to share training data with others but also want to guarantee data security. Although homomorphic encryption solution realizes the encryption of data, it cannot meet the needs of multiuser data sharing when sharing data in the same field, and it cannot achieve one-to-many fine-grained commu- nication. In attribute-based encryption, only users who meet the access strategy set by the owner can obtain the data, which can achieve more flexible access control. Therefore, to handle the problem of the incompatibility of secure storage and fine-grained sharing of IoT big data in deep learning, an attribute-based encryption solution can be introduced. Among them, the encryption of the ciphertext strategy is more suitable to be used in this scenario than the key-based encryption due to the characteristics of the ciphertext contact access strategyand key contact access structure.

In the actual data sharing scenario, due to the numerous attributes of the visitor, there are many departments in the enterprise engaged in the IoT, so the attribute fluidity is relatively large. Access users obtain the key through their own identity attribute information. If the attribute used to represent the identity does not have a valid period, it means that even if an employee resigns or a department merges, it will not affect the access rights of the resigned employee or the original department staff, and these employees can still obtain data through their own identity attributes. If a resigned employee sells IoT big data in exchange for economic benefits, it will not only endanger the interests of the company but also harm people's personal safety. This showsthat it is necessary to set the validity period for each user attribute. The attribute will be cancelled when it expires. Moreover, many current solutions allow users to access unlimited times within the set time. To prevent the number of visits from being abused, it is necessary to limit the num- ber of visits within the set time. By limiting the user's access period and access frequency, to a certain extent, it is possibleto reduce the occurrence of data leakage caused by the sale of data information by employees or outsiders using decryption attributes to access big data of the Internet of Things.

We consider the data privacy problems of big data generated in the field of IoT for mobile computing and use attribute revocation idea [29, 30], then propose an IoT big data

privacy protection scheme based on time and the number of decryption restrictions. This scheme combines homomorphic encryption and attribute-based encryption. In summary, the main contributions of this paper are as follows:

(1) We propose a scheme that limits attribute usage time and user decryption frequency. By setting the attribute version number for each attribute as a mark, it is compared with the local time to determine whether the time has expired and realize the revocation. Besides, it limits the number of user accesses by establishing a user decryption frequency table and setting access tokens.

(2) We combine homomorphic encryption with ciphertext-based attribute-based encryption technology, which makes this solution more effective in improving data confidentiality without affecting neural network model training.

(3) We analyse the security of the scheme in a real deployment.

The remainder of the paper is organized as follows. After introducing the related work in Section 2, we provide related technologies used in this paper in Section 3. Section 4 describes the design of our scheme. We analyse security and effectiveness of our scheme in Section 5. Finally, Section6 concludes this study.

2. Related Work

Although deep learning has brought great convenience to human life, its application is inseparable from data. If some IoT data involves the user's private information, once it is leaked, it will cause property and life safety issues. More and more solutions [31–34] are proposed to solve data security issues, which are implemented by not directly processing data. In addition, people can also protect their privacy by processing data. Lv et al. [35] proposed a secure transaction framework based on the blockchain, which uses the encryption mechanism of the blockchain to ensure information security, but it does not achieve fine-grained access control. Lindell et al. [36] proposed that two parties can process data sets collaboratively without revealing their privacy. Agrawal et al. [37] proposed a scheme that implements the function of outsourcing data to others for data mining tasks. This scheme is confirmed that it does not reveal the data owner's private information during the outsourcing process. Homomorphic encryption technology is considered to be the most effective and most direct means of protecting user privacy [38]. It can directly perform operations, and the results can be consistent with the results of plaintext operations. In 2007, Orlandi et al. [39] introduced homomorphic encryption technology and multiparty secure computing technology to feed the encrypted data into the neural network model for training, which not only ensured the consistency of the plaintext and ciphertext calculation results but also considered security. In [40], the authors proposed a neural network model that uses encrypted data for training. At the same time, in this scheme, it is also proved that cloud services can be used to put encrypted data into the neural network for prediction operations, and the results are returned from the cloud in the form of ciphertext. In [41], the authors improved the scheme [40] and proved that encrypted data can also train neural networks.

In addition to directly encrypting big data, there are also many solutions for setting access control to the data protection layer. In [42], the author created the first CP-ABE solutions, the access policy and ciphertext are sent to the receiver together. Due to the existence of user or attribute revocation problems, research on revocation of ABE has always received extensive attention. Shi et al. [43] proposed a scheme under a hierarchical cryptosystem. Once the attributes are revoked, the public key, private key, and ciphertext of the scheme need to be updated, so the revoking effciency of this scheme is not high. In [44, 45], the authors pointed out that the private key can be divided into two parts. If the attribute is revoked, the two keys need to be updated, and it is necessary to reencrypt the ciphertext and header files, so the cost of revocation is relatively large. In [46], the authors proposed a user revocation scheme based on a time limit, but it did not achieve fine-grained attribute revocation. In [47], the authors proposed a scheme for using smart contracts to revoke attributes. In addition to these revocation schemes, the purpose of revocation can also be realized by limiting the number of user visits. In [48], the authors proposed a scheme that decryption frequency can be limited. But the function of this scheme is a bit single. While sharing IoT big data that can be used for neural network training, users can adopt a scheme that combines homomorphic encryption and CP-ABE. The solution proposed in [49] has proved that combining the two technologies in such scenarios can not only reduce the risk of data leakage but also reduce the number of key communications. However, in the field of deep learning-assisted IoT applications, there are very few solutions that can combine these technologies to limit user access time and specify the number of user accesses.

3. Preliminaries

Bilinear Maps. Suppose there is a large prime number p and two cyclic groups G_1 and G_2, their orders are both p, and g is a generator of G_1. Then, there is a mapping $e : G_1 \times G_1 \longrightarrow G_2$ from G_1 to G_2, and it has the following properties [50]:

(1) Bilinearity: $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$ for $\forall a, b \in Z_p^*$ and $\forall u, v \in G_1$

(2) Nondegeneracy: there exists $x, y \in G_1$, such that $e(x, y) \neq 1$, where 1 is the identity element of group G_2

(3) Computability: for $\forall u, v \in G_1$, $e(u, v)$ can be calculated by an effective algorithm.

Then, we call the above mapping e a bilinear mapping. In general, the cyclic group G_1 is an additive cyclic group, and the cyclic group G_2 is a multiplicative cyclic group.

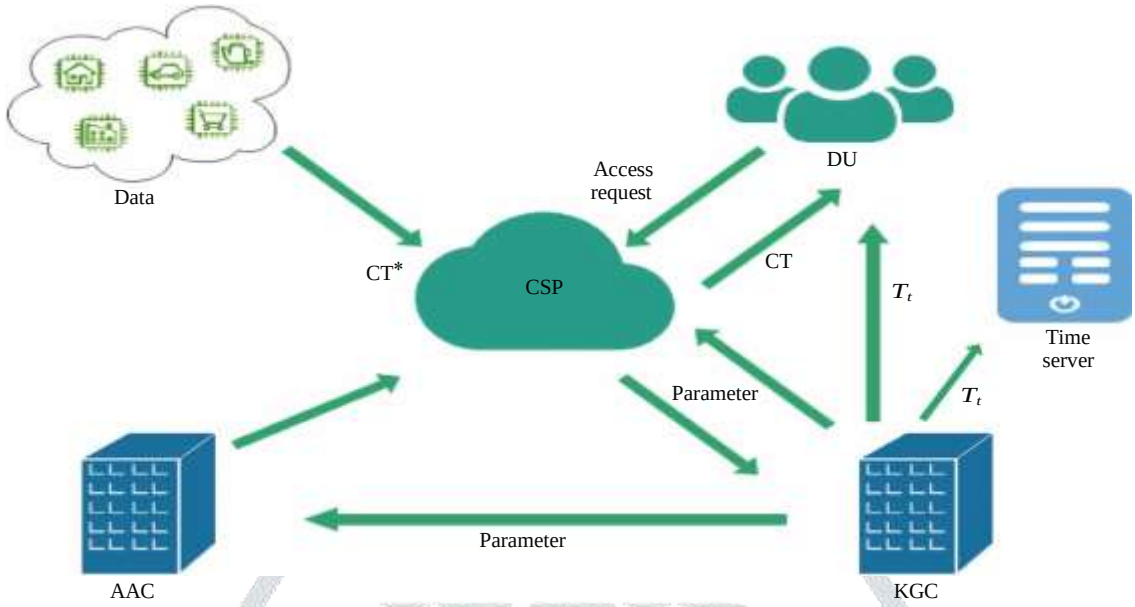

Figure 2: System model.

Diffie-Hellman Problem. For the additive cyclic group G_1 in the above bilinear map e, there are the following difficult problems in cryptography and discrete mathematics, various cryptosystems based on bilinear mapping are built on the basis of these difficult problems.

Definition 1 (discrete logarithm problem (DL)). If there are any two elements g and Y, $g \in G_1$, $Y \in G_1$, and satisfy $Y = g^k$, where $k \in Z_p^*$, it is difficult to calculate the value of k.

Definition 2 (computational Diffie-Hellman problem (CDH)). Given that a triplet is $\langle g, g^a, g^b \rangle$ where g is a generator of group G_1, $a, b \in Z_p^*$, it is difficult to calculate the value of g^{ab}.

Definition 3 (decisional Diffie-Hellman problem (DDH)). If there is a four-tuple $\langle g, g^a, g^b, g^c \rangle$ where g is a generator, $a, b, c \in Z_p^*$, it is difficult to determine whether $c = ab \bmod p$ is true.

Because the above three types of problems are based on group G_1, they are all regarded as group G_1 problems.

DBDH Assumption. Given that a five-tuple is $[g, g^a, g^b, g^c, Z]$, where g is a generator of group G_1, $a, b, c \in_R Z_p$, $Z \in G_2$, it is difficult to determine whether $Z = e(g, g)^{abc}$ is true.

Access Structure. The structure is a set of judgment conditions, usually expressed as Γ, which contains several attribute elements in the attribute set A and threshold logic operators (such as OR and AND). If there is an attribute set that satisfies the judgment condition, this attribute set is called an authorized set, otherwise, we called it an unauthorized set. Let $P = \{P_1, P_2, \cdots, P_n\}$ be the entity set of n participants. For $\forall B, C$, if $B \in C$ and $B \subseteq C$, there is $C \in A$, then, the set $A \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotonous. An access structure is a nonempty subset of $\{P_1, P_2, \cdots, P_n\}$, namely, $A \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \backslash \{\phi\}$. In this proposed solution, the identity information of each user can be described by multiple attri- butes, such as company, department, and position, which are all his attributes.

Secure Two-Party Computing Protocol. A secure two- party computing protocol [51–53] means that in a network environment with a low safety factor, two participants can obtain the value of a function after collaborative calculation. Then, they can also obtain the desired value from each otheraccording to this agreement. However, apart from knowing the value of oneself, other information cannot be derived. Through this agreement, it can be ensured that the privacy of the participants themselves will not be leaked when they do not trust each other, which improves program security.

Homomorphic Encryption. Definition $E_k(a)$ means using an encryption algorithm to encrypt a, the key is k, and F means a certain algorithm of homomorphic encryp- tion, if there is an effective algorithm I, it can be satisfied: $E_k(F(a_1, a_2, \cdots, a_n)) = I(k, F(E_{a_1}, E_{a_2}, \cdots, E_{a_n}))$. It means that E is homomorphic to F.

4. The Proposed System

System Solution. In our proposed solution, there exist six types of entities: IoT device, cloud server, data user, attribute authorization centre, key generation centre, and time server.The scheme model is shown in Figure 2.

From Figure 2, we can know that the data owner can encrypt all kinds of data from IoT devices and upload the data to CSP. The access user makes an access request to the cloud server. Legitimate users can download document set

from the cloud server and decrypt it. CSP and KGC jointly generate keys for users through continuous interaction. The time server is responsible for detecting whether the time sent to it by other entities has expired or has been forged or tampered with.

System Algorithms. We let group G be a bilinear group, let g be a generator in group G. Let $e : G \times G \longrightarrow G_1$ be a bilinear mapping. We choose three hash functions in this scheme: $H : \{0,1\}^* \longrightarrow G$, so that each attribute can be mapped to the group, $H_1 : G_1 \longrightarrow Z_p^*$ and $H_2 : \{0,1\}^* \longrightarrow \{0,1\}^*$. In addition, for any $i \in Z_p^*$, an attribute set A, the Lagrangian coefficient is defined as $\Delta_{i,A}(x) = \prod_{j \in A, j \neq i} (x - j)/(i - j)$.

(1) Setup$(\lambda) \longrightarrow (PK_{KGC}, MK_{KGC}, PK_{CSP}, MK_{CSP}, PK_{sign}, MK_{sign})$. First, the security parameter λ is used to generate three pairs of public and private keys, which are the key generation centre's key pair PK_{KGC}, MK_{KGC}, the cloud server's key pair PK_{CSP}, MK_{CSP}, and the public and private key pair for digital signature PK_{sign}, MK_{sign}. KGC randomly selects $\beta \in_R Z_p^*$ and sets $h = g^{\beta}$, so $(PK_{KGC} = h, MK_{KGC} = \beta)$. At the same time, KGC also selects a random number $\gamma \in_R Z_p^*$, so that the public and private key pair used for digital signature is $(PK_{sign} = g^{\gamma}, MK_{sign} = \gamma)$. CSP randomly selects $\alpha \in_R Z_p^*$, it sets $(PK_{CSP} = e(g, g)^{\alpha}, MK_{CSP} = g^{\alpha})$. Second, CSP allocates initialization information other than public and private keys for users accessing IoT data, including setting the unique identity of the ith user as u_i, where $u_i \in Z_p^*$. A list L is stored in the cloud server, which contains the user's unique mark u_i, the number of user visits σ, and the state-related mark K_c. Third, KGC selects a random secret value $r_j \in Z_p^*$ for the user, and AAC selects a mark $v_i \in Z_p^*$ for each attribute. Therefore, the system public key is $PK = \{G, g, h, f = g^{1/\beta}, e(g, g)^{\alpha}, g\}$, and the master key is $MK = \{\alpha, \beta\}$. The initial value of σ is set to 0.

(2) KeyGen$(PK, MK, MK_{sign}, A, U_i, st) \longrightarrow (SK_{u_t})$. In this part, the digital signature private key MK_{sign}, the user's attribute set A, and the attribute version key U_i, and outputs the user's decryption key. The following four parts are included:

(a) Generate attribute version key. This part is executed by AAC. AAC randomly selects any value $t_i \in Z_p^*$ for each attribute, and t_i is used as a parameter for subsequent use, so the attribute version key is set to $U_i = v_i t_i$, and the attribute version key is generated and sent to CSP.

(b) Generate partial user keys. This part is formed by the simultaneous operation of KGC and CSP via introducing a secure two-party computing proto-col. First, KGC takes the parameters (r_j, β) as input, and CSP takes the parameter α as input.

Through calculation, $x = (\alpha + r_j)/\beta$ is obtained, and the result is output to CSP. CSP selects a random number $\delta \in Z_p^*$, calculates $A = g^{x\delta} = g^{(\alpha + r_j)\beta\delta}$, and sends the calculation result to KGC. When KGC receives the result, calculate $B = A^{1/\beta^2} = (g^{(\alpha + r_j)\beta\delta})^{1/\beta^2} = g^{(\alpha + r_j)\delta/\beta}$, and finally, the result B is sent to CSP. CSP calculates $SK_C = B^{1/\delta} = g^{(\alpha + r_j)/\beta}$ from the received result B. KGC inputs the set attribute version key and outputs partial user's private key $(SK_k = (D = g^{(\alpha + r_j)/\beta}, \forall \lambda \in A, D_{\lambda} = g^{r_j} H(\lambda)^{U_i}, D_{\lambda}^* = g^{U_i}))$. The partial user's decryption key is composed of a combination of the private key generated by CSP and KGC: $SK = \{SK_C, SK_k\} = \{D = g^{(\alpha + r_j)/\beta}, \forall \lambda \in A, D_{\lambda} = g^{r_j} \cdot H(\lambda)^{U_i}, D_{\lambda}^* = g^{U_i}\}$

(c) In this part of the algorithm, $K_p = g^{1/(H_2(st) + u_i)}$, $K_c = E^{1/(H_2(st) + u_i)}$, K_c is the output value of the algorithm VRF [54], K_p, K_c refer to the calculation and detection scheme of the algorithm VRF. Therefore, the final decryption key is $SK_{u_t} = \{SK, st, K_c, K_p\} = \{D = g^{(\alpha + r_j)/\beta}, \forall \lambda \in A, D_{\lambda} = g^{r_j} \cdot H(\lambda)^{U_i}, D_{\lambda}^* = g^{U_i}, st, K_c, K_p\}$, the generated decryption key is sent to the user.

(d) Set the expiration time T for each attribute and digitally sign $T_t \xi = g^{1/(H_2(T_t) + \gamma)}$.

(3) HKeyGen (). This algorithm generates the key of a homomorphic encryption algorithm. This scheme uses the DGHV encryption algorithm. In this algorithm, the key is selected as follows: we choose a ran- domly generated positive prime number as the key p, where $p \in [2^{\eta^2 - 1}, 2^{\eta^2})$.

(4) Encryption$(PK, \Gamma, M, p) \longrightarrow (CT^*)$. This algorithm first inputs the system public key and access policy tree Γ, homomorphic encryption key p, and plaintext message M. Then, this algorithm outputs encrypted ciphertext CT^*. First, the data owner uses the homomorphic encryption key p to encrypt the plaintext M. The specific operation is as follows: they choose two random numbers q, r, where $p \in [2^{\eta^s - 1}, 2^{\eta^s})$, $r \in [2^{\eta - 1}, 2^{\eta})$, $p/2 > 2r$, $q \gg p$. The ciphertext of the document set is calculated by formula $pq + 2r + M$, M is expressed in binary, and the generated ciphertext is uploaded. Second, the data owner encrypts the homomorphic key and uploads the key with attribute access control to the cloud. The data owner regards the attributes as leaf nodes, the root node of the tree is R, and the other nodes are threshold logic opera- tors. The encryption operation performs from the root node and, from top to bottom, produces a linked order for each node, which is d_x polynomial q_x. If n_x is the threshold of nonleaf nodes, then there is a rela-tion $d_x = n_x - 1$. Then, they select a random value s

$\in Z_p^*$, set the polynomial on the root node to $q_R(0) = s$, and use the homomorphic encryption key p to encrypt the plaintext, and use the encryption result to calculate $C = \text{Enc}(p) \cdot e(g, g)^{\alpha s}$, $\hat{C} = h^s$. Let the polynomial of other nodes be $q_x(0) = q_{p(x)}(\text{index}(x))$, where index (x) represents the number associated with any node x. The order of nodes is indicated from left to right. In the entire access policy tree, the information carried by each leaf node must be calculated, $C_\lambda = g^{q_\lambda(0)}$, $C'_\lambda = H(\lambda)^{q_\lambda(0)}$. Then, the final CT* is

$$\text{CT}^* = \{\Gamma, C = \text{Enc}(p) \cdot e(g, g)^{\alpha s}, \hat{C} = h^s, \forall \lambda \in \gamma : C_\lambda = g^{q_\lambda(0)}, C'_\lambda = H(\lambda)^{q_\lambda(0)}\}$$

(5) $\text{TimeCheck}(\text{SK}_{u_t}, S, T, \xi, \text{PK}_{\text{sign}})$. In this part of the algorithm, after the time server receives the validity period of the attribute, it first needs to verify it with digital signature technology to check whether it has been forged or tampered with and verify it with the following calculation method:

$$\begin{aligned} e\left(g^{H_2(T_t)} \cdot \text{PK}_{\text{sign}}, \xi\right) &= e\left(g^{H_2(T_t)} \cdot g^{\gamma}, g^{1/H_2(T_t)+\gamma}\right) \\ &= e\left(g^{H_2(T)+\gamma}, g^{1/H_2(T)+\gamma}\right) \quad (1) \\ &= e(g, g). \end{aligned}$$

If the verification is successful, it means that the attribute has not been forged or tampered with. The time server compares the validity period T_t with the present time to determine whether the attribute has exceeded the validity period. If it has not expired, you need to continue to execute step 6. If it expires, the attribute needs to be revoked. On the contrary, if the verification fails, it means that the validity period T_t has been maliciously modified, then return $\perp$.

(6) $\text{GenToken}(u_i, \text{st}, \text{ctr}_{\max}) \longrightarrow (B_T)$ after verifying the attribute validity period T_t, it also needs to verify the user's access times, but the difference is that even if certain attribute fails, the user still has the possibility of access rights, but if the access times exceed the setthreshold, then the user does not have the right to access IoT resource data. This algorithm makes the user's unique identity u_i, the user's current state st, and the maximum allowed number of decryption $\text{ctr}_{\max}$ into a token and sends the token to the cloud server.

(7) $\text{Predecryption}(\text{SK}_{u_t}, B_T) \longrightarrow (\text{timeindex})$. In this part, the cloud server first detects $e(g^{H_2(st)} * g^{u_i}, K_p) = E$ and $K_c = e(g, K_p)$ after receiving the token with information. If it meets the verification conditions, CSP will detect the number of decryption $\sigma + 1 \le \text{ctr}_{\max}$ in the list L, if it is satisfied, let $\sigma = \sigma + 1$, timeindex $= \perp$, it means accessing users can no longer access IoT big data even if they have access rights.

(8) $\text{Decryption}(\text{SK}_{u_t}, \text{CT}^*) \longrightarrow (p)$. This part of the algorithm is executed by the decryption user and is divided into the following four parts:

(a) When the node x in the access policy tree belongsto the leaf node in the access policy tree, let $i = \text{att}(x)$, it means that the attribute corresponding to the node x computes

$$\text{DecryptNode}(\text{SK}_{u_t}, \text{CT}^*, x) = \frac{e(D_i, C_x)}{e(D^*, C^*)} = \frac{e(g, g)^{q_x(0)(r_j+U_i)}}{e(g, g)^{q_x(0)U_i}} = e(g, g)^{r_j q_x(0)}. \quad (2)$$

If the attribute is not in the user's attribute set, return $\perp$.

(b) When λ belongs to a nonleaf node in the structure tree, we let S_x be the set of child nodes each node z of k_x. When F_z exists and the user's current decryption frequency meet the requirements, then compute

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(\text{timeindex})} = \prod_{z \in S_x} e(g, g)^{r_j q_z(0) \Delta_{i,S'_x}} \\ &= \prod_{z \in S_x} e(g, g)^{r_j q_{\text{parent}(z)}(\text{index}(z)) \Delta_{i,S'_x}} \\ &= e(g, g)^{r_j q_\lambda(0)}. \end{aligned} \quad (3)$$

If the root node R in this structure tree is replaced by the x node in the above formula, it can be computed as $A = \text{DecryptNode}(\text{SK}_u, \text{CT}^*, R) = e(g, g)^{r_j s}$.

(c) When the user's attribute set meets the requirements, decryption is performed:

$$\begin{aligned} \text{Dec}\left(\frac{C}{\hat{C}/A}\right) &= \text{Dec}\left(\frac{e(g, g)^{\alpha s} \cdot \text{Enc}(p)}{\dfrac{e(h^s, g^{(\alpha + r_j)/\beta})}{e(g, g)^{r_j s}}}\right) \\ &= \text{Dec}\left(\frac{e(g, g)^{\alpha s} \cdot \text{Enc}(M)}{\dfrac{e(h^s, g^{(\alpha + r_j)/\beta})}{e(g, g)^{r_j s}}}\right) \quad (4) \\ &= \text{Dec}\left(\frac{e(g, g)^{\alpha s} \cdot \text{Enc}(M)}{e(g, g)^{\alpha s}}\right) = M = p \end{aligned}$$

update K_c at this time and store it in the list L, and

then, the user and CSP continue to perform step 8. Then, let timeindex = 1, otherwise, timeindex = ⊥. If

(d) After the data visitor obtains the homomorphic key p, users can obtain the document set by using the homomorphic key p.

(9) Revocation. When the attribute is revoked, this algorithm is executed. In the algorithm, it consists of three parts:

(a) First, KGC randomly selects a reencryption parameter ψ, which is assigned to AAC, CSP, and users whose attributes have been revoked, so that they can update relevant component information in time. Receiving the update information, AAC updates the attribute ver- sion keys U_i' of the revoked attributes that itmanages, $U_i' = v_i t_i'$.

(b) The next step is to update the user key. CSP obtains the reencryption parameters allocated in the previous step and regenerates the user's latest version key together with KGC. The updated user key is $SK_u = \{D = g^{(\alpha+r_j)/\beta}, D_{\lambda'}$

$= g^{r_j} \cdot H(\psi\lambda)^{U_i}, D^*_{\lambda'} = g^{U_i}, \forall \lambda \in A \setminus \{\lambda'\}: D_\lambda$

$= g^{r_j} \cdot H(\lambda)^{U_i}, D^*_\lambda = g^{U_i}$, st, K_c, K_p}.

(c) The third step is to update the ciphertext. In this part, CSP first selects a random cipher value $s' \in Z^*_p$ to ensure forward security and then updates the relevant components of the ciphertext after receiving the reencryption parameters. The updated ciphertext is

$$CT^* = \{\Gamma, C = e(g, g)^{\alpha s+s'} \cdot \mathrm{Enc}_k(M), \hat{C}$$

$$= h^{s+s'}, \forall \lambda \in I : C_\lambda = g^{q_\lambda(\alpha\lambda+s')}, C^*_\lambda \quad (5)$$

$$= H(\psi\lambda)^{q_\lambda(\alpha\lambda+s')} \quad \lambda = \lambda', C^*_\lambda$$

$$= H(\lambda)^{q_\lambda(\alpha\lambda+s')} \quad \lambda \neq \lambda'\}$$

5. Safety and Efficiency Analysis

Solution Security Analysis

Confidentiality. The confidentiality of this scheme is achieved through two aspects. On the one hand, the attri- butes of the user must be able to meet the policy set by data owner. If the access policy is not met, then the attributes can-not be used to calculate $e(g, g)^{rs}$, so it can prevent unautho-rized users from stealing sensitive data. On the other hand, while generating the user's key, to reduce the conditionimpact of low safety factor and untrustworthy, a secure two-party computing protocol is used to protect the related information of the private key from being obtained by anyone other than itself.

Forward Security. Since each user is set to limit decryption frequency, when users access data, if they meet the requirements of the access policy, they also need to send a token carrying the number of times of decryption to the cloud server. If the number of accesses exceeds the limit, then the user can no longer be decrypted, which ensures forward security.

Collusion Resistance. Users need to use their own attributes to calculate $e(g, g)^{rs}$. If users with different permissions want to create a conspiracy attack, then KGC and CSP will generate partial decryption keys through a secure two-party calculation protocol $D = g^{(\alpha+r_j)/\beta}$, $D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}$, $D^* = g^{U_i}$, where u_i is a unique random value for each user, so even if the attackers collude, they cannot calculate the value of $e(g, g)^{rs}$.

Chosen-Plaintext Attack

Proof. We consider that there exists a polynomial adversary A that is able to break this solution and algorithm B that can overcome the DBDH problem with the advantage of ε.

Initialization: adversary A selects an access structure tree T and sends this access strategy tree to challenger B, and challenger B executes the Setup () initialization algorithm. This part of the process is as follows:

Randomly select four values to calculate $e(g, g)^{ab} e(g, g)^x = e(g, g)^a$, where $a, b, c, x \in Z^*_p$.

For each attribute $\lambda \in A$, select a random value $\ell_i \in Z^*_p$, when the attribute does not exist in the access structure tree T, we set $Y_i = H^{1/\ell_i}$, $y_i = b/\ell_i$, if the attribute exists in the access structure tree T, we let $Y_i = g^{\ell_i}$ and $y_i = \ell_i$.

The public key $PK = \{G, h, g, f = g^{1/\beta}, e(g, g)^\alpha\}$ is published, and challenger B keeps the private key $MK = \{g^\alpha, \beta\}$.

Phase 1: after challenger B obtains the public key, adver-sary A can issue a query request. Adversary A selects an attribute set $s = \{\lambda_i \mid \lambda_i \in T\}$ and u_i and submits the informationto challenger B to apply for a private key. Challenger B randomly selects r_i, generates the corresponding private key. The calculation process is as follows:

$$SK = \{D = g^{(\alpha+x+r_j)/\beta}, D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}, D^*_\lambda = g^{U_i} : \quad (6)$$

If the number of decryptions meets the requirements, st, K_c, K_p will not affect the final decryption effect.

Challenge: adversary A has obtained the access control tree T at this time and then submits two plaintexts of the same length to challenger B. By comparing the attribute sets, if the attribute set sent in the previous step does not meet the structure tree T, then the two plaintexts are set to m_0, m_1, andthe two plaintexts are sent to challenger B along with the access strategy tree. Then, B randomly selects $\beta \in \{a, b\}$, cal- culate:

$$C_0 = \mathrm{Enc}\{M_\beta \cdot e(g, g)^{\alpha s} = \mathrm{Enc}\{M_\beta \cdot e(g, g)^{(\alpha ab+x)\beta s}$$

$$= \mathrm{Enc}\{M_\beta \cdot e(g, g)^{abs} e(g, g)^{xs} = \mathrm{Enc}\{M_\beta$$

$$\cdot e(g, g)^{abs} \cdot e(g^x, g^s)\}, \quad (7)$$

$$C_0 = h^s, \forall \lambda \in I : C_\lambda = g^{q_\lambda(\alpha\beta)}, C^*_\lambda = H(\lambda)^{q_\lambda(\alpha\beta)}:$$

Challenger B sends this information to A.

Table 1: Functional comparison.

Schemes	Revocability	Time	Number	Collusion	Ciphertext operability
[55]	User and attribute	×	×	✓	×
[46]	User	✓	×	✓	×
[48]	User	×	✓	✓	×
[47]	Attribute	✓	×	✓	×
[49]	None	×	×	✓	✓
[56]	Attribute	×	×	✓	×
Our scheme	User and attribute	✓	✓	✓	✓

Table 2: Cost comparison.

Schemes	Secret key cost	Decryption cost		Revocation	User-cost
		User	CSP	Attribute-cost	
[55]	3*e*	3p	×	3ð*n* + 1Þ*p*	*np*
[56]	4*e*	×	*e*	*e*	×
Our scheme	*O*ð5*n*Þ	*O*ð*n*Þ	*O*ð*n*Þ	*O*ð5*n*Þ	*O*ð1Þ

Phase 2: *A* can always ask *B* for private key-related information, and then, *A* guesses the ciphertext and needs to give his own guess value p'.

Guess: if $\mathrm{p}' = \mathrm{p}'$, then DBDH is established, the advantage is $p_r[\mathrm{p}' = \mathrm{p}' \mid e(g, g)^{abc}] = \varepsilon + 1/2$, if $\mathrm{p}' \neq \mathrm{p}'$, the ciphertext cannot be judged, and the advantage is $p_r[\mathrm{p}' \neq \mathrm{p}' \mid e(g, g)^{\vartheta}] = 1/2$. In summary, $p_r[(g, g^a, g^b, g^c, e(g, g)^{abc}] = 1 - p_r[(g, g^a, g^b, g^c, e(g, g)^{\vartheta}] = 1 \geq \varepsilon$. It shows that this scheme can realize that no adversary can break the scheme with a nonnegligible advantage in polynomial time.

Theoretical Comparison. Our scheme is compared with other schemes in terms of revocation mechanism, time limit, number of decryption limits, and anticollusion. The comparison results are shown in Table 1.

From Table 1, it can be seen that in [46–49, 56], the revocation schemes proposed by the authors do not fully meet the revocation needs. Although in [55] the authors proposed a scheme that can support user revocation and attribute revocation, in the scenario we mentioned, it is also a requirement that the ciphertext can be operated. This scheme in [49] realizes that users can operate on ciphertext, but it is not suitable for scenarios where attributes need to be revoked. Our scheme realizes two revocation functions, solves the basic system security problem, and achieves the ciphertext operable function. What is more, we also consider two factors: time and frequency of decryption.

Our scheme is compared with other schemes in terms of key generation effciency, decryption efficiency, and revocation efficiency. *e* is the exponential calculation cost, and *p* is the bilinear pair calculation cost. The comparison results are shown in Table 2.

It can be seen that in [55] only the user performs the decryption operation and in [56] only CSP performs the decryption operation, which will cause one-side pressure. Our scheme can effectively reduce the amount of user tasks by placing part of the decryption task on the cloud server. Also, in [55], while realizing user revocation, the cost is *np*. However, in our scheme, if the user is revoked after judgment, the user only needs to be removed from the list *L*, thus, its computational complexity is better than the schemes [55, 56]. Although the cost of generating the key is relatively high due to the use of a two-party security protocol, the security of the key is guaranteed through this multiparty cooperation method.

6. Conclusions

Since important personal privacy may be leaked while storing and sharing IoT big data on the cloud, we have proposed an IoT big data privacy protection scheme based on time and decryption frequency limitation, the solution realizes the revocation within the time range and the revocation within the range of decryption times. The access control is set by the combination of homomorphic encryption and attribute-based encryption. In our scheme, legitimate users with a homomorphic encryption key can obtain the original data, and users without a homomorphic encryption key can perform operation training on the homomorphic ciphertext. Our scheme does not only affect the training of the neural network model but also improves the confidentiality of the data. At the same time, the security of the system is improved by introducing a secure two-party agreement. Through theoretical analysis, we found that our scheme realizes two revocation functions, solves the basic system security problem, and achieves the ciphertext operable function. While realizing user revocation, the computational complexity is preferable to other schemes. Besides, our scheme can effectively reduce the amount of user tasks by placing part of the decryption task on the cloud server. Therefore, our scheme can not only ensure safety but also improve efficiency. In the next step, we plan to combine the advantages of decentralization and anonymity of blockchain to protect big data in the Internet of Things in a distributed storage environment.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

There is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China Project (Nos. 61701170 and U1704122), the Key Scientific and Technological Project of Henan Province (Nos. 202102310340 and 202102210352), the Young Elite Scientist Sponsorship Program by Henan Association for Science and Technology (No. 2020HYTP008), the Foundation of University Young Key Teacher of Henan Province (Nos. 2019GGJS040 and 2020GGJS027), and the Key Scientific Research Project of Colleges and Universities in Henan Province (No. 21A110005).

References

[1] W. Huang, Y. Xu, X. Hu, and Z. Wei, "Compressive hyperspectral image reconstruction based on spatial–spectral residual dense network," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 884–888, 2019.

[2] W. Huang, Y. Huang, H. Wang, Y. Liu, and H. J. Shim, "Local binary patterns and superpixel-based multiple kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4550–4563, 2020.

[3] L. Peng, H. Zhang, H. Hassan, Y. Chen, and B. Yang, "Acceler- ating data gravitation-based classification using GPU," *Journal of Supercomputing*, vol. 75, no. 6, pp. 2930–2949, 2019.

[4] X. Zhang, F. Ding, and E. Yang, "State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 7, pp. 1157–1173, 2019.

[5] X. Zhang, L. Xu, F. Ding, and T. Hayat, "Combined state and parameter estimation for a bilinear state space system with moving average noise," *Journal of the Franklin Institute*, vol. 355, no. 6, pp. 3079–3103, 2018.

[6] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in Inter- net of Things with privacy preserving: challenges, solutions and opportunities," *IEEE Network*, vol. 32, no. 6, pp. 144– 151, 2018.

[7] E. Park, Y. Cho, J. Han, and S. J. Kwon, "Comprehensive approaches to user acceptance of Internet of Things in a smart home environment," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2342–2350, 2017.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[9] P. Gope, R. Amin, S. K. Hafizul Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentica-tion scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Gen- eration Computer Systems*, vol. 83, pp. 629–637, 2017.

[10] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: automated test- ing of deep-neural-network-driven autonomous cars," in *Pro- ceedings of the 40th International Conference on Software Engineering*, pp. 303–314, Gothenburg, Sweden, May 2018.

[11] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2020.

[12] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for Internet of Things via ensem- ble classification," *IEEE Transactions on Industrial Informat- ics*, vol. 17, no. 8, pp. 5810–5818, 2020.

[13] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[14] R. Poplin, A. V. Varadarajan, K. Blumer et al., "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning," *Nature Biomedical Engineering*, vol. 2, no. 3, pp. 158–164, 2018.

[15] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting clas- sified sub-dictionaries over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2016.

[16] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans-actions on Geoscience and Remote Sensing*, vol. 56, no. 5,pp. 2811–2821, 2018.

[17] A. Rachedi and A. Benslimane, "Multi-objective optimization for security and QoS adaptation in wireless sensor networks," in *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016.

[18] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: new threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2018.

[19] M. Amoozadeh, A. Raghuramu, C.-n. Chuah et al., "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.

[20] M. Shafiq, Z. Tian, A. K. Bashir, X. du, and M. Guizani, "Cor- rAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[21] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: secure and verifiable federated learning," *IEEE Transactions on Informa-tion Forensics and Security*, vol. 15, pp. 911–926, 2019.

[22] L. Jiang, X. Lou, R. Tan, and J. Zhao, "Differentially private collaborative learning for the IoT edge," *In Proceedings of the 2019 International Conference on Embedded Wireless Sys- tems and Networks (EWSN '19)*, Junction Publishing, USA, 2019.

[23] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artifi- cial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[24] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Transactions on Information Forensics andSecurity*, vol. 14, no. 4, pp. 870–885, 2019.

[25] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 97–109, 2018.

[26] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Pri- vacy preserving data aggregation scheme for mobile edge com- puting assisted IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[27] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson, *Scalable Private Learning with PATE*, 2018.

[28] C. Xu, J. Ren, L. She, Y. Zhang, Z. Qin, and K. Ren, "EdgeSani- tizer: locally differentially private deep inference at the edge for mobile data analytics," *IEEE Internet of Things Journal*, vol. 6,no. 3, pp. 5140–5151, 2019.

[29] J. Hur, "Improving security and efficiency in attribute-based data sharing," *Transactions On Knowledge And Data Engineering*, vol. 25, no. 10, pp. 2271–2282, 2013.

[30] J. Y. Wang and X. J. Zhou, "An attribute-based encryption scheme for ciphertext policy that supports attribute revocation," *Computer Engineering*, pp. 1–7, 2020.

[31] Y. Sun, Z. Tian, M. Li, S. Su, X. Du, and M. Guizani, "Honey- pot identification in softwarized industrial cyber-physical sys- tems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5542–5551, 2020.

[32] Y. Pang, L. Peng, Z. Chen, B. Yang, and H. Zhang, "Imbalanced learning based on adaptive weighting and Gaussian function synthesizing with an application on android malware detection," *Information Sciences*, vol. 484, pp. 95–112, 2019.

[33] V. Ravindranath, S. Ramasamy, R. Somula, K. S. Sahoo, and A. H. Gandomi, "Swarm intelligence based feature selection for intrusion and detection system in cloud infrastructure," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–6, Glasgow, UK, July 2020.

[34] D. Xu, J. Pan, X. Du, B. Wang, M. Liu, and Q. Kang, "Massive fishing website URL parallel filtering method," *IEEE Access*, vol. 6, pp. 2378–2388, 2018.

[35] L. Lv, Z. Yang, L. Zhang, Q. Huang, and Z. Tian, "Multi-party transaction framework for drone services based on alliance blockchain in smart cities," *Journal of Information Security and Applications*, vol. 58, no. 4, p. 102792, 2021.

[36] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Proceedings of the 20th Annual International Cryptology Con- ference on Advances in Cryptology*, Springer, Berlin, Heidel- berg, 2000.

[37] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, Dallas, Texas, USA, 2000.

[38] T. Plantard, W. Susilo, and Z. Zhang, "Fully homomorphic encryption using hidden ideal lattice," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2127–2137, 2013.

[39] C. Orlandi, A. Piva, and M. Barni, "Oblivious neural network computing via homomorphic encryption," *EURASIP Journal on Information Security*, vol. 2007, no. 1, Article ID 037343, 2007.

[40] N. Dowlin, G. B. Ran, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, pp. 201–210, NY, USA, 2016.

[41] E. Hesamifard, H. Takabi, and M. Ghasemi, "Privacy-Preserv-ing Machine Learning in Cloud," in *Proceedings of the 2017 on cloud computing security workshop*, pp. 39–43, 2017.

[42] M. Chase, "Multi-authority attribute based encryption," in *Conference on Theory of Cryptography*, Springer-Verlag, 2007.

[43] J. Shi, C. Huang, K. He, and X. Shen, "ACS-HCA: an access control scheme under hierarchical cryptography architecture," *Chinese Journal of Electronics*, vol. 28, no. 1, pp. 52–61, 2019.

[44] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2017.

[45] S. Wang, K. Guo, and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *PLoS One*, vol. 13, no. 9, articlee0203225, 2018.

[46] G. Dilxat, S. Y. Han, A. Gulmira, and H. Nurmamat, "Time-based user revocation CP-ABE scheme," *Journal of Xinjiang University(Natural Science Edition)*, vol. 36, no. 3, pp. 324–329, 2019.

[47] X. Qin, Y. Huang, Z. Yang, and X. Li, "An access control scheme with fine-grained time constrained attributes based on smart contract and trapdoor," in *2019 26th International Conference on Telecommunications (ICT)*, Hanoi, Vietnam, April 2019.

[48] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Audit-able σ-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Informa- tion Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[49] Y. Tan, L. Lu, and J. Y. Wang, "Ciphertext-policy attribute encryption scheme based on homomorphic encryption," *Computer Engineering and Applications*, vol. 55, no. 19, pp. 115– 120, 2019.

[50] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for out- sourced decryption in attribute based encryption," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 478–487, 2020.

[51] Y. Tang and D. Y. Xu, "A secure two-party computation prob- lem based on the convolution," *Journal of Guizhou Universi- ty(Natural Ences)*, vol. 33, no. 1, pp. 52–57, 2016.

[52] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09*, pp. 121–130, Chicago, Illinois, USA, 2009.

[53] S. S. M. Chow, "Removing escrow from identity-based encryp- tion," in *International Workshop on Public Key Cryptography*,Springer, Berlin, Heidelberg, 2009.

[54] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Public Key Cryptography - PKC 2005*, Springer, 2005.

[55] X. Li, S. Tang, L. Xu, H. Wang, and J. Chen, "Two-factor data access control with efficient revocation for multi-authority cloud storage systems," *IEEE Access*, vol. 5, pp. 393–405, 2017.

[56] Z. T. Jiang, J. Huang, S. Hu, and Z. Xu, "Fully-outsourcing CP- ABE scheme with revocation in cloud computing," *ComputerScience*, vol. 46, no. 7, pp. 114–119, 2019.