sensors

MDPI

Article

Lightweight Fine-Grained Access Control for Wireless Body Area Networks

Mohammad Ali [1], Mohammad-Reza Sadeghi [1] and Ximeng Liu [2,3,4,*]

[1] Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran 159163-4311, Iran; mali71@aut.ac.ir (M.A.); msadeghi@aut.ac.ir (M.-R.S.)
[2] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[3] Guangdong Provincial Key Laboratory of Data Security and Privacy Protection, Guangzhou 510632, China
[4] Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China
* Correspondence: snbnix@gmail.com

Received: 27 December 2019; Accepted: 11 February 2020; Published: 17 February 2020

Abstract: Wireless Body Area Network (WBAN) is a highly promising technology enabling health providers to remotely monitor vital parameters of patients via tiny wearable and implantable sensors. In a WBAN, medical data is collected by several tiny sensors and usually transmitted to a server-side (e.g., a cloud service provider) for long-term storage and online/offline processing. However, as the health data includes several sensitive information, providing confidentiality and fine-grained access control is necessary to preserve the privacy of patients. In this paper, we design an attribute-based encryption (ABE) scheme with lightweight encryption and decryption mechanisms. Our scheme enables tiny sensors to encrypt the collected data under an access control policy by performing very few computational operations. Also, the computational overhead on the users in the decryption phase is lightweight, and most of the operations are performed by the cloud server. In comparison with some excellent ABE schemes, our encryption mechanism is more than 100 times faster, and the communication overhead in our scheme decreases significantly. We provide the security definition for the new primitive and prove its security in the standard model and under the hardness assumption of the decisional bilinear Diffie-Hellman (DBDH) problem.

Keywords: fine-grained access control; wireless body area networks; lightweight computation; attribute-based encryption; cloud computing

1. Introduction

Nowadays, because of several improvements in public health, nourishment, and medicine, the aging population around the world has been quickly increasing. For instance, in the United States, the population of people over the age of 65 is predicted to double by 2040 [1]. Also, in the People's Republic of China, it is predicted that the number of people aged over 60 will be doubled by 2040 [2]. These estimates show that increasing the number of elderly people with various health problems may significantly increase healthcare costs in the near future [3–5]. Therefore, the current healthcare system may not be able to respond to the patients' requests in the coming years [4,6].

With the rapid development of medical sensors and wireless communications [7], wireless body area networks (WBANs) are under rapid development. WBANs have significant potential for improving the current health system. As we have shown in Figure 1, a WBAN consists of several implantable or wearable sensors and a controller. The responsibility of the sensors is to monitor the vital parameters of a patient (e.g., breathing rate, blood pressure, diabetes, and asthma) as well as measuring the environmental parameters such as humidity and temperature. The sensors collect health data files and encrypt them. Then, they transfer the generated ciphertext to the collector. The controller

working as a gateway transfers the gathered health data to a cloud service provider. WBANs can significantly raise the efficiency of healthcare services as individuals do not need to visit the hospital anymore. Thus, WBANs play an important role in affording highly reliable ubiquitous healthcare services. However, as in the cloud-based WBANs the health data are outsourced to a third-party cloud server, some security concerns over fine-grained access control and data confidentiality are raised. Moreover, as tiny sensors in WBANs usually have limited computational and power resources, providing a secure lightweight encryption mechanism is another challenge in this scenario.

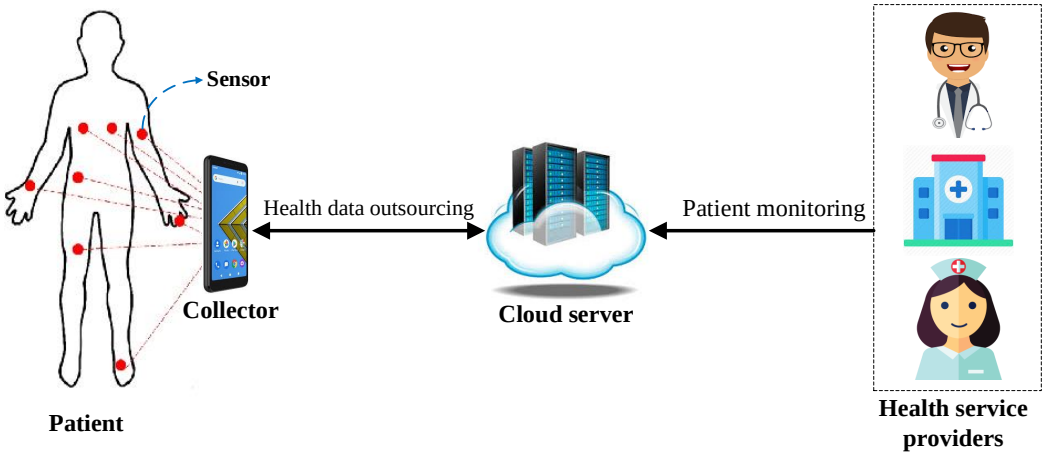

Figure 1. A typical WBAN.

Attribute-based encryption (ABE) [8,9] is a promising tool to afford confidentiality and fine-grained access control simultaneously. Generally, ABE schemes can be divided into three categories key-policy ABE (KP-ABE) [10], ciphertext-policy ABE (CP-ABE) [11], and dual-policy ABE (DP-ABE) [12]. In a KP-ABE, a data user's secret-key is associated with an access control policy which is defined by a central authority, and each ciphertext is labeled by a set of attributes. A data user can decrypt a ciphertext if the access policy associated with its secret-key is satisfied by the attribute set associated with the ciphertext. Also, in a CP-ABE, a data user's secret-key is associated with the data user's attributes, and ciphertexts are associated with an access control policy. The secret-key of a data user can decrypt a ciphertext only if the attribute set of the data user satisfies the access policy associated with the ciphertext. In a DP-ABE scheme, secret-key of a data user corresponds to both an access control policy defined by the central authority and the data user's attributes. Each ciphertext also is associated with both an access control policy defined by a data owner and a set of attributes. A data user can decrypt a ciphertext if and only if the access control policy embedded in the ciphertext is satisfied by attributes of the data user, and attributes of the ciphertext satisfy the data user's access policy. It seems that CP-ABE is more comfortable for both data owners and data users.

However, to the best of the authors' knowledge, current ABE schemes suffer from expensive computational operations in the encryption phase. Therefore, since the sensors have limited computational and power resources, existing ABE schemes are not appropriate for providing fine-grained access control in WBANs. To address this problem, in this paper, we design a lightweight fine-grained access control scheme called LW-FGAC which is able to offer lightweight encryption and decryption mechanisms. Our main contributions are given below:

- Lightweight encryption mechanism: Our proposed encryption mechanism is very efficient. In fact, in contrast with existing schemes, in our encryption scheme, the number of expensive operations performed by data owners (smart devices in the WBAN) does not depend on the number of attributes in the access control policy, and almost all the computational operations are offloaded onto the cloud service provider. As we will see, our encryption approach is more than 100 times faster than some excellent schemes in the literature.

- Lightweight communication overhead: In LW-FGAC, in comparison with the existing work, the communication overhead from a data owner to the cloud server is very few. Indeed, in LW-FGAC, lightweight partial ciphertexts are uploaded to the cloud server instead of ciphertexts with huge size.
- Lightweight decryption mechanism: Similar to the encryption phase, in the decryption phase, heavy computational operations can be outsourced to the CSP such that the CSP learns no partial information about data users' secret-keys and also the underlying data files.
- Security definition and security proof: We formalize the system model and the security definition for the new primitive. Also, we prove the security of the scheme under the hardness assumption of the DBDH problem in the standard model.

2. Related Work

Cao et al. presented a thorough survey on WBANs [13]. Their work surveyed several basic WBAN research projects and enabling technologies. It also explored application scenarios, radio systems, smart devices, and the interconnection of WBANs to afford perspective on the trade-offs between data rate, power consumption, and network coverage. Li et al. [14] introduced an anonymous key agreement and mutual authentication scheme for WBANs. Their work enables the sensor nodes attached to patients' bodies to authenticate with the local server and establish a session key in an unlinkable and anonymous way. Chen et al. presented a detailed review of body area networks and their related issues [15]. They provided a comprehensive investigation of sensor devices, data link layer, physical layer, and radio technology aspects of WBANs. They also introduced some of the design challenges and open problems in this area. Zhang et al. [16] designed an efficient key agreement mechanism for WBANs. Their scheme enables neighboring nodes in WBANs to share a common key established by electrocardiogram (ECG) signals. Their proposed key agreement scheme can secure data communications over WBANs in a plug-n-play manner with no key distribution overhead. He et al. [17] introduced the security and performance challenges related to sensor networks for wireless medical monitoring. They also proposed an attack-resistant and lightweight trust management scheme. Zhou et al. [18] presented several fundamental and sophisticated cyberattacks to wireless sensors networks and introduced some substantial and promising solutions to satisfy the requirements. Ghamari et al. [19] presented a survey on WBANs for health care systems. They compared some current low-power communication technologies supporting the quick advancement and deployment of WBANs. Zhou et al. [20] proposed a privacy-preserving key management system for cloud-based WBANs in m-healthcare social networks. Their proposed scheme protects the patient's identity privacy, location privacy, and sensor deployment privacy by employing a blinding technique and embedding the human body's symmetric structure into the Blom's symmetric-key mechanism with a modified secret sharing technique. Liu et al. [21] designed a medium access control for WBANs. In their work, by employing the Nash Bargaining Solution (NBS), they proposed a cooperative game-theoretic method providing priority-based tuning and maintaining the fairness axioms of game theory. Shen et al. [22] proposed a lightweight multi-layer authentication protocol for WBANs. In their work, using the ECC algorithm, they designed a one-to-many group authentication mechanism and a group key establishment algorithm between personal digital assistants and the other sensor nodes. They also designed a certificateless authentication mechanism without pairing. Whereas, it is known that access control is a major problem in WBANs [23], the mentioned schemes did not consider this problem.

ABE is a promising solution to the access control problem. The notion of ABE was first proposed by Sahai and Waters [8]. In their proposed scheme, a data owner can determine the authorized user to access its data by specifying an attribute set and a threshold value d. Each data user that has at least d common attributes with the specified set can access the outsourced data. After proposing ABE schemes, three schemes [12,24,25] divided ABE schemes into three categories key-policy ABE (KP-ABE), ciphertext-policy ABE (CP-ABE), and dual-policy ABE (DP-ABE), respectively. Zhou et al. [26] designed a constant size CP-ABE. In their work, the size of ciphertexts is not sensitive to the number of

attributes in access control policies. This feature significantly reduces the storage and communication overhead of the system. Guo et al. [27] designed a lightweight CP-ABE scheme with a constant secret-key size [28]. In their scheme, the length of a user's secret-key does not depend on the number of the user's attributes. Chen et al. [29] proposed an attribute-based scheme with short ciphertexts and signatures. Their proposed scheme has adaptive security in the standard model. However, none of the schemes presented in [26,28,29] provide a flexible access structure. Indeed, the schemes presented in [26,28] only supports the And-gates access control policy, and [29] only provides the threshold access control policy. Yao et al. [30], designed a KP-ABE scheme for IoT applications. Their work supports access trees as access control policies. Also, in their work, by using the ECC algorithm, the communication and storage overhead is reduced significantly. He et al. [31] proposed an ABE scheme for mobile cloud-assisted cyber-physical systems. In their work, by eliminating pairing operations, they tried to lighten the encryption and decryption overhead. However, several expensive operations still remain. So, it seems that their scheme is not suitable for WBANs. Moreover, none of the mentioned ABE schemes provide lightweight encryption and decryption mechanisms which is not desirable for WBANs. To address this issue, several lightweight ABE schemes have been put forward. Yang et al. [32,33] designed lightweight access control systems for healthcare IoT networks. Their scheme provides a lightweight decryption mechanism and supports access trees as access control policies. Also, their schemes have adaptive security in the standard model. Xu et al. [34] proposed a lightweight DP-ABE for healthcare IoT systems. Their work offers a lightweight decryption system, and it is provably secure in the selective model. Lin et al. [35] proposed CP-ABE with a lightweight decryption mechanism by using an outsourcing technique. Lai et al. [36] put forward a CP-ABE scheme with verifiable outsourced decryption. Their work also provides a lightweight decryption approach and is provable in the adaptive model. However, none of the mentioned ABE schemes provide a lightweight encryption mechanism. Indeed, in these schemes, the computational operations on the user's side in the encryption phase is very expensive. This feature definitely makes such schemes inappropriate for WBANs. Table 1 compares the features of the mentioned ABE schemes with our proposed LW-FGAC. As we see, LW-FGAC is the only one providing a lightweight encryption approach. Also, we see that LW-FGAC is the only scheme that simultaneously meets all the features given in the table. We refer the reader to [37–44], to see more references related to attribute-based systems and wireless sensor networks.

Table 1. Comparison of Properties in Different ABE Schemes.

Schemes	KP/CP/DP-ABE	Lightweight Encryption Mechanism	Flexible Access Control	Lightweight Decryption Mechanism	Security Model
[8]	ABE	No	No	No	Selective
[12]	DP-ABE	No	Yes	No	Selective
[24]	KP-ABE	No	Yes	No	Selective
[25]	CP-ABE	No	Yes	No	Selective
[26]	CP-ABE	No	No	No	Selective
[27]	CP-ABE	No	No	No	Selective
[28]	KP/CP-ABE	No	No	No	Adaptive
[29]	CP-ABE	No	Yes	No	Adaptive
[30]	KP-ABE	No	Yes	No	Selective
[31]	CP-ABE	No	Yes	No	Selective
[32]	CP-ABE	No	Yes	Yes	Adaptive
[33]	CP-ABE	No	Yes	Yes	Adaptive
[34]	DP-ABE	No	Yes	Yes	Selective
[35]	CP-ABE	No	Yes	Yes	Selective
[36]	CP-ABE	No	Yes	Yes	Adaptive
LW-FGAC	CP-ABE	Yes	Yes	Yes	Adaptive

3. System Architecture

In this section, we present the architecture of our proposed health system. We first describe the system model, and then we present the threat model of our system.

3.1. System Model

As we have shown in Figure 2, our proposed system consists of four generic entities Healthcare Authority (HA), the Cloud Service Provider (CSP), several data owners, and several data users. In below, we describe the mentioned four entities:

- HA: This entity is responsible for initializing the health system and also generating secret-keys of data owners and data users according to their attributes.
- CSP: The CSP has almost unlimited computational and storage resources. Its primary responsibility is to provide storage and computational services. When data owners want to encrypt their collected data, they can outsource most of the computational operations of the encryption phase to the CSP. Moreover, data users can also use the CSP's computational services. When a data user retrieves an encrypted health data, the CSP can help it to recover the associated data by performing most of the heavy computations of the decryption phase without learning any partial information about the underlying health data.
- Data owner: Data owners modeling the tiny wireless sensors attached to bodies of patients and employed to monitor the patients' vital physiological parameters such as blood pressure, heart rate, diabetes, asthma, and etc. The health data collected by data owners first is encrypted under an access control policy and then transferred to a smart device. Finally, the health data are outsourced to the CSP for online/offline analyzing and long-term storage.
- Data owner: Data owners modeling smart devices that collect the health data from patients' bodies and transfer the data to the CSP. The smart devices can be categorized into two following groups:

    1. Implanted and wearable sensors: These sensors usually embedded on the surface of a patient's body or implanted in the deep tissue of a human body. Their main responsibility is to monitor the patients' vital physiological parameters such as blood pressure, heart rate, diabetes, asthma, and etc. After collecting the health data, the sensors first partially encrypt the data under a predetermined access control policy. Then, the partially encrypted data are transferred to the data collector. Note that as the sensors usually have limited computational and power resources, the partial encryption process should be adequate sufficient and does not include costly operations.
    2. Data collector: A data collector could be the WBAN's controller or a mobile device like a tablet or a smartphone. Its main responsibility is to transfer the collected partially encrypted health data to the CSP for completing the encryption process, long-term storage, and online/offline analyzing.

- Data user: Data users model health service providers such as hospitals, doctors, medical clinics, etc. They can be specified by a set of descriptive attributes. Each data user should obtain a secret-key corresponding to its attribute set. Its secret-key can decrypt an outsourced encrypted health data only if the attribute set associated with the secret-key satisfies the access control policy associated with the ciphertext.

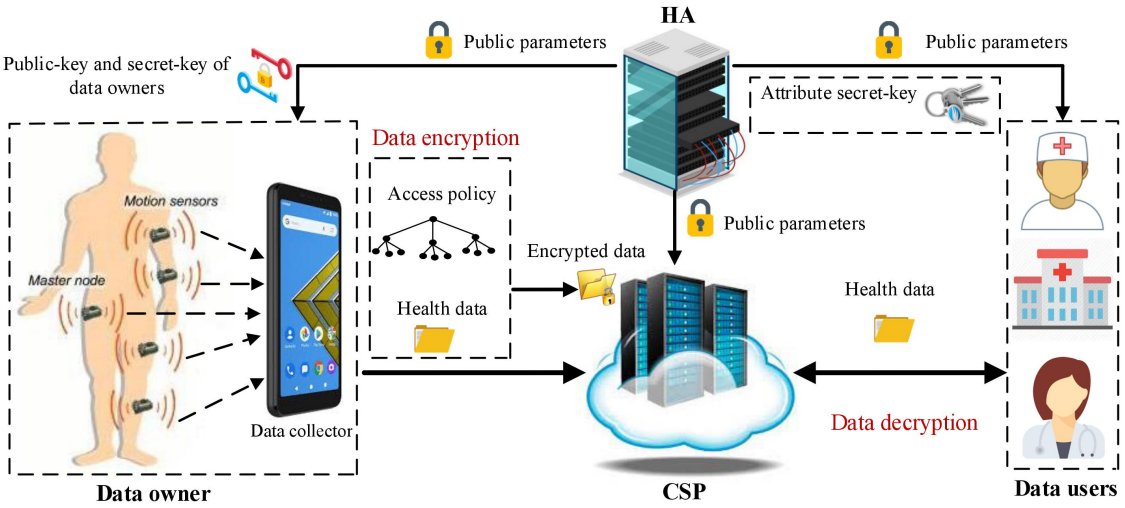

Figure 2. Architecture of our proposed LW-FGAC scheme.

In the following, we give an overview of our proposed LW-FGAC. As shown in Figure 3, our proposed scheme consists of four phases Systeminitialization, Key delegation, Data encryption, and Decryption described below:

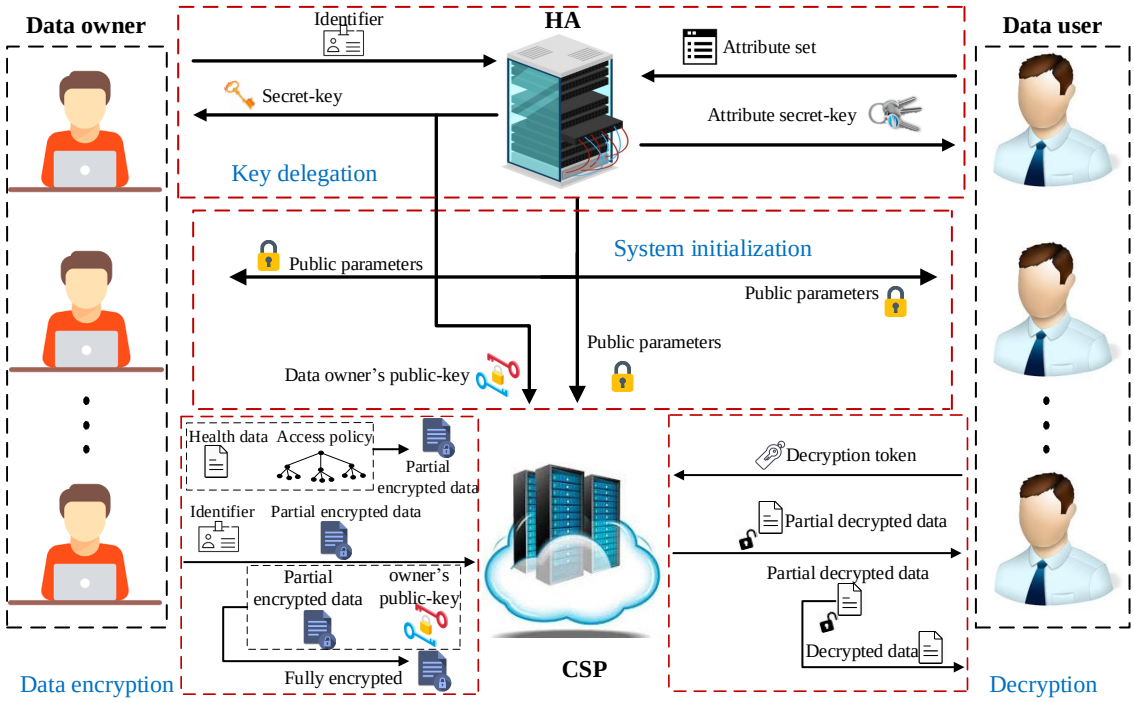

Figure 3. Workflow of our proposed LW-FGAC scheme.

- System initialization: This phase is managed by the HA. In this phase, the HA generates the public parameters and the master secret-key of the system. It publishes the public parameters to the other parties and keeps the master secret-key confidential by itself.
- Key delegation: This phase is operated by the HA. In this phase, public-key and secret-key of data owners as well as secret-keys of data users associated with their attributes are issued. Each data owner should ask the HA to generate its public-key and secret-key. The generated secret-key is given to the data owner, and the public-key is outsourced to the CSP. Also, in this phase, each data user possessing an attribute set can request its secret-key corresponding to the attribute set from

the HA. The HA first checks if the data user has the attributes or not. If so, it provides the data user with an attribute secret-key.

- Data encryption: This phase is executed by data owners and the CSP. When a data owner wants to outsource its collected health data to the CSP, to provide confidentiality and access control, it should define an access control policy and encrypt the health data under it. However, as the computational power of the data owner (implanted and wearable sensors) is assumed to be limited, the heavy computational operations should be offloaded onto the CSP. Using its secret-key, the data owner (implanted and wearable sensors) first performs some lightweight computations and generates a partial ciphertext. Then, the data owner (data collector) gives the partially encrypted data to the CSP, and the CSP completes the encryption procedure. In this phase, the CSP cannot learn any partial information about the underlying health data.
- Decryption: This phase is managed by the CSP and data users. When a data user is authorized for accessing an outsourced health data, using its secret-key obtained in the key delegation phase, it can make a decryption query to the CSP. The CSP performs heavy operations associated with the decryption phase without obtaining any information about the data user's secret-key and also the associated health data. Afterward, the data user can recover the health data by performing some lightweight computational operations.

3.2. Threat Model

The HA is assumed to be trustworthy. It does not collude with data users and does not gives unauthorized secret-keys to them. Data owners also are assumed to be trusted. They do not reveal the contents of their data to the other parties and do not grant access rights to unauthorized data users. The CSP is assumed to be honest but curious entity. It always executes the given protocols correctly, but it is curious to learn some unauthorized information about the outsourced health data. To gain some information about the outsourced data files, it may collude with unauthorized data users. Data users are assumed to be malicious. Although they do not reveal the contents of health data files if they are authorized to access them, they may try to learn some unauthorized information about the other outsourced health data through colluding with the CSP and the other data users.

4. Preliminaries

For an arbitrary set S, let $x \leftarrow S$ denote the random selection of an element $x \in S$. Also, for algorithm $\mathcal{A}$, let $O \leftarrow \mathcal{A}(I)$ denote executing $\mathcal{A}$ on input I and outputting O. In the following, we present some related cryptographic notions.

4.1. Cryptographic Background

Bilinear map: Consider two cyclic groups G_1 and G_2 of a prime order q. A function $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is said to be a bilinear map if the following conditions hold:

- Bilinearity: $\hat{e}(g^a, g^b) = \hat{e}(g^b, g^a) = \hat{e}(g, g)^{ab}$, For each $a, b \in \mathbb{Z}_q$ and $g \in G_1$,
- Non-degeneracy: There is a $g \in G_1$ such that $\hat{e}(g, g) \neq 1$.
- Computability: There exists an efficient algorithm computing $\hat{e}(g, h)$, for any $g, h \in G_1$.

Assume that $\mathcal{G}$ is a probabilistic polynomial-time (PPT) algorithm that $(\lambda, q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$, where λ is the security parameter of the system and $(q, G_1, G_2, \hat{e})$ is the same as before. In this work, we consider the following assumption called decisional bilinear Diffie Hellman (DBDH) on $\mathcal{G}$:

Decisional Bilinear Diffie Hellman assumption (DBDH): Consider $(\lambda, q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$, $g \leftarrow G_1$ and $\alpha, \beta, \gamma \leftarrow \mathbb{Z}_q$. The DBDH assumption states that for all PPT adversaries $\mathcal{A}$ there is a negligible function *negl* such that

$$|\Pr(\mathcal{A}(\lambda, q, g, g^\alpha, g^\beta, g^\gamma, g^{\alpha\beta\gamma}, G_1, G_2, \hat{e}) = 1) - \Pr(\mathcal{A}(\lambda, q, g, g^\alpha, g^\beta, g^\gamma, g^z, G_1, G_2, \hat{e}) = 1)| \leq negl(\lambda), \quad (1)$$

where the above probabilities are taken over the random selection of $g \in G$ and $\alpha, \beta, \gamma, z \in \mathbb{Z}_q$, and also the randomness employed in $\mathcal{G}$ and $\mathcal{A}$.

4.2. Access Trees

In an access tree, each leaf is associated with a unique attribute, and each inner node represents a threshold value. Also, the threshold value of each leaf node is assumed to be 1. Suppose that $\mathcal{T}$ is an access tree, v_a is the leaf associated with an attribute a, k_v is the threshold value associated with a node v in $\mathcal{T}$, $R_{\mathcal{T}}$ is the root node of $\mathcal{T}$, $L_{\mathcal{T}}$ is the leaf node set of $\mathcal{T}$, and $\mathcal{T}_v$ is a subtree of $\mathcal{T}$ rooted at a node v.

Let $\mathbb{U}$ be the universal attribute set, and $\mathcal{T}$ be an access tree on $\mathbb{U}$. For a given attribute set $Att \subseteq \mathbb{U}$ and a node v in $\mathcal{T}$, let $F_{\mathcal{T}_v}$ be a function mapping Att to $\{0,1\}$ and performing as follows:

- When v is a leaf node corresponding to an attribute a, $F_{\mathcal{T}_v}(Att) = 1$ if $a \in Att$, and 0 otherwise.
- When v is an inner node, $F_{\mathcal{T}_v}(Att) = 1$ if and only if v has at least k_v children $c_1, \ldots, c_{k_v}$ that $F_{\mathcal{T}_{c_i}}(Att) = 1$, for any $i = 1, \ldots k_v$.

We say that an attribute set Att satisfies an access tree $\mathcal{T}$ if $F_{\mathcal{T}_{R_{\mathcal{T}}}}(Att) = 1$.

Suppose that q is a prime number, and $\mathcal{T}$ is an access tree. Consider an algorithm $\{q_v(0)\}_{v \in L_{\mathcal{T}}} \leftarrow Share_q(\mathcal{T}, r)$ which shares a secret $r \in \mathbb{Z}_q$ according to $\mathcal{T}$ and q and performs as below:

- It generates a $(k_{R_{\mathcal{T}}} - 1)$-degree polynomial $q_{R_{\mathcal{T}}}$ for $R_{\mathcal{T}}$ such that $q_{R_{\mathcal{T}}}(0) = r$, and its other coefficients are chosen uniformly at random from $\mathbb{Z}_q$.
- For each node v having a polynomial q_v, it generates a polynomial q_{c_i} for the i-th child of v such that $q_{c_i}(0) = q_v(i)$, and the other its coefficients are uniform elements of $\mathbb{Z}_q$.

When this algorithm stops, it assigns a value $q_v(0)$ to each leaf node v in the tree.

5. System Definition and Security Model

In this section we present the system definition and the secrity model. Table 2 presents the notations used in this section.

Table 2. Notations Employed in The System Definition And Our Proposed Construction.

Notation	Description
λ	Security parameter of the system
$\mathbb{U}$	Universal attribute set of the system
$params$	Public parameters of the system
MSK	Master secret-key of the HA
Att_u	Attribute set of a data user
id_u	Identifier of a data user
SK_O	Secret-key of a data owner
PK_O	Public-key of a data owner
M	A data file
$\mathcal{T}$	An access tree
$PCT_{\mathcal{T}}$	Partial ciphertext associated with an access tree $\mathcal{T}$
$CT_{\mathcal{T}}$	Ciphertext associated with an access tree $\mathcal{T}$
SK_u	Attribute secret-key of a data user
TK_u	Decryption token generated by a data user in the decryption phase
k	Private-key generated by a data user in the decryption phase
M'	Partial decrypted ciphertext

5.1. Definition of LW-FGAC

LW-FGAC scheme is a tuple of PPT algorithms $(Setup, User.KeyGen, Owner.KeyGen, Part.Enc, Full.Enc, TokenGen, Part.Dec, Full.Dec)$ defined as below:

- $Setup(\lambda, \mathbb{U})$: This algorithm is operated by the HA. It takes as input the security parameter λ and the universal attribute set $\mathbb{U}$. It outputs public parameters *params* and the master secret-key MSK.
- $User.KeyGen(params, MSK, id_u, Att_u)$: This algorithm is executed by the CSP. On input the public parameters *params*, the master secret-key MSK, a data user's identifier id_u, and an attribute set Att_u, this algorithm outputs a secret-key SK_u associated with id_u and Att_u.
- $Owner.KeyGen(params)$: This algorithm can be run by a data owner or the HA. It inputs the public parameters *params* and outputs a pair of secret-key and public-key (SK_O, PK_O).
- $Part.Enc(params, \mathcal{T}, SK_O, M)$: A data owner executes this algorithm. The public parameters of the system, an access tree $\mathcal{T}$, the data owner's secret-key, and a message M are the input of the algorithm. This algorithm outputs a partial ciphertext $PCT_{\mathcal{T}}$ associated with the message M and the access tree $\mathcal{T}$.
- $Full.Enc(params, PCT_{\mathcal{T}}, PK_O)$: The CSP runs this algorithm. This algorithm takes the public parameters *params*, a partial ciphertext $PCT_{\mathcal{T}}$, and a data owner's public-key PK_O. It outputs a ciphertext $CT_{\mathcal{T}}$.
- $TokenGen(params, id_u, SK_u, CT_{\mathcal{T}})$: This algorithm is executed by a data user. On input the pubic parameters *params*, a data user's identifier id_u, a secret-key SK_u, and a ciphertext $CT_{\mathcal{T}}$, this algorithm returns a private-key k and a decryption token TK_u, or it outputs an error message $\perp$.
- $Part.Dec(params, CT_{\mathcal{T}}, TK_u)$: The CSP runs this algorithm. It takes as input the public parameters *params*, a ciphertext $CT_{\mathcal{T}}$, and a decryption token TK_u. This algorithm outputs a partial decrypted ciphertext M'.
- $Full.Dec(params, M', k)$: A data user operates this algorithm. On input the public parameters *params*, the partial decrypted ciphertext M', and its associated private-key k, this algorithm returns the message associated with M'.

Definition 1. *We say that an* $LW-FGAC$ *scheme* Π *is correct if for any security parameter* λ*, universal attribute set* $\mathbb{U}$*, public parameters and master secret-key* $(params, MSK) \leftarrow Setup(\lambda, \mathbb{U})$*, attribute set* Att_u*, identifier* id_u*, access tree* $\mathcal{T}$ *satisfied by* Att_u*, secret-key* $SK_u \leftarrow User.KeyGen(params, MSK, id_u, Att_u)$*, public-key and secret-key* $(SK_O, PK_O) \leftarrow Owner.KeyGen(params)$*, message* M*, partial ciphertext* $PCT_{\mathcal{T}} \leftarrow Part.Enc(params, \mathcal{T}, SK_O, M)$*, and ciphertext* $CT_{\mathcal{T}} \leftarrow Full.Enc(params, PCT_{\mathcal{T}}, PK_O)$*, we have:*

$$Full.Dec(params, M', k) = M, \tag{2}$$

where $M' \leftarrow Part.Dec(params, CT_{\mathcal{T}}, TK_u)$ *and* $TK_u \leftarrow TokenGen(params, id_u, SK_u, CT_{\mathcal{T}})$.

5.2. Security Definition

Security of LW-FGAC requires that for any PPT adversary modeling the CSP colluding with unauthorized data users, the advantage of the adversary in learning partial information about encrypted data files is a negligible function in the security parameter of the system. In other words, the adversary is unable to distinguish the encryption of two data files of its choice. We formalize the security requirement by using the following indistinguishability experiment.

Indistinguishability experiment $LW-FGAC_{\mathcal{A},\Pi}(\lambda)$:

Let $\Pi = (Setup, User.KeyGen, Owner.KeyGen, Part.Enc, Full.Enc, TokenGen, Part.Dec, Full.Dec)$ be an LW-FGAC scheme and $\mathcal{A}$ be a PPT adversary. Consider the following experiment:

1. Setup: A challenger chooses a security parameter λ and a universal attribute set $\mathbb{U}$. It executes $(params, MSK) \leftarrow$Setup$(\lambda, \mathbb{U})$. *params* is given to $\mathcal{A}$ and MSK is maintained by the challenger.
2. Phase 1: For polynomially many times, $\mathcal{A}$ makes some queries to the following oracle, and for each data user with identifier id_u, the challenger maintains a list L_{id_u} which is initially empty.

   $\mathcal{O}_{User.KeyGen}(Att, id_u)$: The challenger runs $SK_u \leftarrow UKeyGen(PK, MSK, Att, id_u)$ and returns SK_u to the adversary. It also substitutes $L_{id_u} \cup Att$ with L_{id_u}.

3. Challenge: $\mathcal{A}$ declares an access tree $\mathcal{T}^*$ and two equal-length messages M_0 and M_1. The challenger checks if there is an identifier id_u such that L_{id_u} satisfies $\mathcal{T}^*$ or not. If so, the challenger stops and returns 0. Otherwise, it first selects $b \leftarrow \{0,1\}$ and an identifier id_O. Then, it runs $(SK_O, PK_O) \leftarrow Owner.KeyGen(params)$ and $PCT_{\mathcal{T}}^b \leftarrow Part.Enc(params, \mathcal{T}, SK_O, M_b)$. PK_O and $PCT_{\mathcal{T}}^b$ are given to $\mathcal{A}$.
4. Phase 2: $\mathcal{A}$ makes more queries to the oracle $\mathcal{O}_{\textbf{User.KeyGen}}(Att, id_u)$ and the challenger answers it provided $Att \cup L_{id_u}$ does not satisfy $\mathcal{T}^*$.
5. Guess: $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

The output of the experiment is defined to be 1 if $b = b'$, and 0 otherwise. We say that the adversary $\mathcal{A}$ wins the game, and we write $LW - FGAC_{\mathcal{A},\Pi}(\lambda) = 1$ if the experiment's output is equal to 1.

Definition 2. *An $LW - FGAC$ scheme Π is said to be secure if for all PPT adversaries $\mathcal{A}$ there exists a negligible function $negl$ such that*

$$Pr(LW - FGAC_{\mathcal{A},\Pi}(\lambda) = 1) \leq \frac{1}{2} + negl(\lambda). \tag{3}$$

6. Our Construction

In this section, we present our proposed LW-FGAC scheme. As mentioned in Section 3.1, our proposed scheme consists of four phases System initialization, Key delegation, Data encryption, and Decryption. In the following, the mentioned four phases are described in detail. The notations employed in our construction are given in Table 2.

6.1. System Initialization

In this phase, the HA selects a security parameter λ and a universal attribute set $\mathbb{U}$. Then, it executes $(params, MSK) \leftarrow Setup(\lambda, \mathbb{U})$ as follows and publishes $params$ to the other entities.

$Setup(\lambda, \mathbb{U})$: This algorithm runs $(\lambda, q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$ and selects $P_0, P_1, P_2, X_1 \leftarrow G_1$, and $x_0 \leftarrow \mathbb{Z}_q$. Then, for each $i \in \mathbb{U}$, it chooses $sk_i \leftarrow \mathbb{Z}_q$ and computes $PK_i = sk_i P_0$. It sets

$$MSK = (x_0, P_1, X_1, \{sk_i\}_{i=1}^m) \tag{4}$$

and

$$params = (\lambda, G_1, G_2, \hat{e}, P_0, P_2, E_1, E_2, \{PK_i\}_{i=1}^m), \tag{5}$$

as the master secret-key and the global public parameters of the system, respectively, where $E_1 = \hat{e}(x_0 P_0, P_1)$ and $E_2 = \hat{e}(P_0, X_1)$.

6.2. Key Delegation

As shown in Figure 4, in this phase, the HA provides data users with some secret-keys according to their attributes and also provides each data owner with a pair of public-key and secret-key. Each data user possessing an attribute set Att_u should first select a unique identifier id_u and ask the HA to generate its secret-key. The HA runs $SK_u \leftarrow User.KeyGen(params, MSK, id_u, Att_u)$ and returns SK_u to the data user. Also, each data owner with identifier id_O can request its public-key and secret-key from the HA. The HA runs $(SK_O, PK_O) \leftarrow Owner.KeyGen(params)$ and returns SK_O to the data owner. (id_O, PK_O) is also outsourced to the CSP. Note that secret-key and public-key of a data owner can be generated by itself. However, as its computational power is assumed to be limited, this task usually is outsourced to the HA. In the following, we describe the mentioned two algorithms:

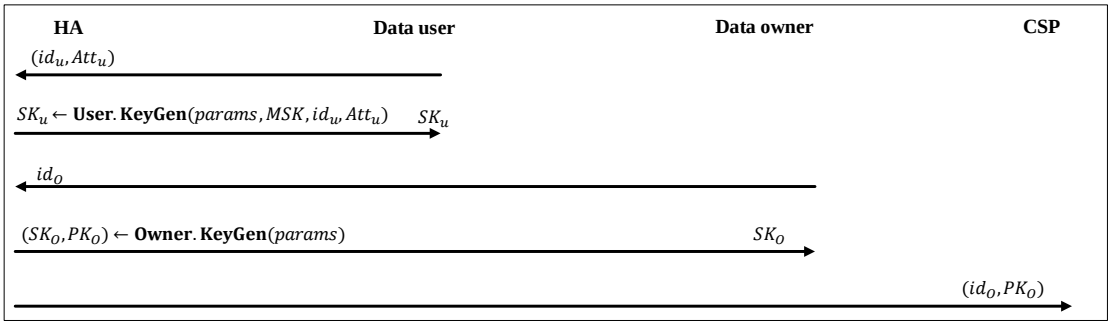

Figure 4. Key delegation phase.

$User.KeyGen(params, MSK, id_u, Att_u)$: It calculates:

$$SK_{i,u} = x_0 P_1 + X_1 + sk_i id_u, \tag{6}$$

for each $i \in Att_u$, and outputs $SK_u = \{SK_{i,u}\}_{i \in Att_u}$.

$Owner.KeyGen(params)$: It selects $d_O \leftarrow \mathbb{Z}_q$ and calculates $PK_O^{(1)} = E_2^{d_O}$, $PK_O^{(2)} = d_O P_0$, $PK_O^{(3)} = d_O P_2$ and $PK_{i,O} = d_O(PK_i - P_2)$, for each $i \in \mathbb{U}$. It returns (SK_O, PK_O), where $SK_O = d_O$ and $PK_O = (PK_O^{(1)}, PK_O^{(2)}, PK_O^{(3)}, \{PK_{i,O}\}_{i \in \mathbb{U}})$.

6.3. Data Encryption

As shown in Figure 5, in this phase, data owners encrypt their data by outsourcing most of the computational operations to the CSP. A data owner with identifier id_O and public-key and secret-key (SK_O, PK_O) that wants to encrypt a message M defines an access tree $\mathcal{T}$ and runs $PCT_{\mathcal{T}} \leftarrow Part.Enc(params, \mathcal{T}, SK_O, M)$ to generate a partial ciphertext $PCT_{\mathcal{T}}$. The data owner makes a request $(id_O, PCT_{\mathcal{T}})$ to the CSP to complete the encryption procedure. Then, the CSP executes $CT_{\mathcal{T}} \leftarrow Full.Enc(params, PSK_{\mathcal{T}}, PK_O)$ and generates a ciphertext associated with the message M and the access tree $\mathcal{T}$. The mentioned two algorithms are presented below:

$Part.Enc(params, \mathcal{T}, SK_O, M)$: It selects $r \leftarrow \mathbb{Z}_q$ and runs $\{q_{v_i}(0)\}_{v_i \in L_{\mathcal{T}}} \leftarrow Share_q(r + SK_O, \mathcal{T})$. Then, it calculates $C_1 = E_1^{-r} M$, $\tilde{r} = r + SK_O$ and returns partial ciphertext $PCT_{\mathcal{T}} = (\mathcal{T}, C_1, \tilde{r}, \{q_{v_i}(0)\}_{v_i \in L_{\mathcal{T}}})$.

$Full.Enc(params, PCT_{\mathcal{T}}, PK_O)$: Given a partial ciphertext $PCT_{\mathcal{T}} = (\mathcal{T}, C_1, \tilde{r}, \{q_{v_i}(0)\}_{v_i \in L_{\mathcal{T}}})$ and a data owner's public-key $PK_O = (PK_O^{(1)}, PK_O^{(2)}, PK_O^{(3)}, \{PK_{i,O}\}_{i \in \mathbb{U}})$, it calculates

$$C_2 = \tilde{r} P_2 - PK_O^{(3)} = r P_2, \tag{7}$$

$$C_3 = E_2^{-\tilde{r}}(PK_O^{(1)}) = E_2^{-r}, \tag{8}$$

and for any leaf node v_i in $\mathcal{T}$, it sets

$$C_{v_i}^{(1)} = q_{v_i}(0) P_0 - PK_O^{(2)} = (q_{v_i}(0) - SK_O) P_0, \tag{9}$$

$$C_{v_i}^{(2)} = q_{v_i}(0)(PK_i - P_2) - PK_{i,O} = (q_{v_i}(0) - SK_O)(PK_i - P_2). \tag{10}$$

Finally, this algorithm outputs a ciphertext

$$CT_{\mathcal{T}} = (\mathcal{T}, C_1, C_2, C_3, \{C_{v_i}^{(1)}\}_{v_i \in L_{\mathcal{T}}}, \{C_{v_i}^{(2)}\}_{v_i \in L_{\mathcal{T}}}). \tag{11}$$

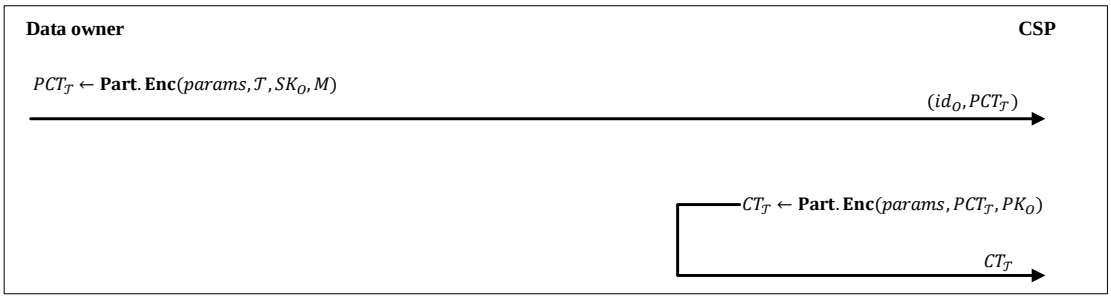

Figure 5. Data encryption phase.

6.4. Decryption

As we have shown in Figure 6, in this phase, by outsourcing the heavy computational operations to the CSP, a data user can recover its desired data. Assume that $CT_{\mathcal{T}}$ has been retrieved from the CSP. To decrypt the ciphertext, a data user with secret-key SK_u and identifier id_u first executes $TK_u \leftarrow TokenGen(params, id_u, SK_u, CT_{\mathcal{T}})$ and generates a decryption token TK_u. It sends a decryption request $(CT_{\mathcal{T}}, TK_u)$ to the CSP. Then, the CSP runs $M' \leftarrow Part.Dec(params, CT_{\mathcal{T}}, TK_u)$ and returns the partial decrypted ciphertext M' to the data user. The data user can run the lightweight algorithm $M \leftarrow Full.Dec(parms, M', k)$ and recover the associated message M. Detail of the mentioned three algorithms are given below:

$TokenGen(params, id_u, SK_u, CT_{\mathcal{T}})$: Given a data user's secret-key $SK_u = \{SK_{i,u}\}_{i \in Att_u}$ associated with an attribute set Att_u, a ciphertext $CT_{\mathcal{T}}$ associated with an access tree $\mathcal{T}$, and an identifier id_u, this algorithm checks if there is an attribute set $S \subseteq Att_u$ satisfying $\mathcal{T}$ or not. If not, it returns $\perp$. Otherwise, it selects $k \leftarrow \mathbb{Z}_q$ and calculates $K = k\, id_u$ and $K_i = k\, SK_{i,u}$, for each $i \in S$. It outputs a private-key k and a token $TK_u = (K, \{K_i\}_{i \in S})$.

$Part.Dec(params, CT_{\mathcal{T}}, TK_u)$: Given a ciphertext $CT_{\mathcal{T}} = (\mathcal{T}, \{C_i\}_{i=1}^{4}, \{C_{v_i}^{(1)}\}_{v_i \in L_{\mathcal{T}}}, \{C_{v_i}^{(2)}\}_{v_i \in L_{\mathcal{T}}})$ and a token $TK_u = (K, \{K_i\}_{i \in S})$, it first computes

$$\begin{aligned} L_i &= \frac{\hat{e}(K_i, C_{v_i}^{(1)})}{\hat{e}(K, -C_{v_i}^{(2)})} \\ &= E_1^{kq_{v_i}(0)} E_2^{kq_{v_i}(0)} \hat{e}(id_u, P_2)^{kq_{v_i}(0)}, \end{aligned} \tag{12}$$

for each $i \in S$. Then, by using the polynomial interpolation method, it computes

$$L = E_1^{kr} E_2^{kr} \hat{e}(id_u, P_2)^{kr}. \tag{13}$$

Finally, it returns $M' = (C', C_1)$, where

$$C' = \frac{L}{\hat{e}(K, C_2)} = E_1^{kr} E_2^{kr}. \tag{14}$$

$Full.Dec(parms, M', k)$: On input a partial decrypted ciphertext M' and its associated private-key k, this algorithm outputs a message

$$M = C'^{k^{-1}} C_1 C_3. \tag{15}$$

Figure 6. Decryption phase.

7. Correctness and Security Analysis

In this section, we first show that our proposed scheme is correct. Then, we prove its security in the standard model.

7.1. Correctness Proof

Theorem 1. *Our proposed LW-FGAC scheme is correct.*

Proof. We prove that LW-FGAC fulfills Definition 1. Given $(params, MSK) \leftarrow Setup(\lambda, \mathbb{U})$, an attribute set Att_u, an identifier id_u, an access tree $\mathcal{T}$ satisfied by Att_u, a message M, $SK_u \leftarrow User.KeyGen(params, MSK, id_u, Att_u)$, $(SK_O, PK_O) \leftarrow Owner.KeyGen(params)$, $PCT_{\mathcal{T}} \leftarrow Part.Enc(params, \mathcal{T}, SK_O, M)$, $CT_{\mathcal{T}} \leftarrow Full.Enc(params, PCT_{\mathcal{T}}, PK_O)$, we show that the output of the decryption phase is equal to M. Let $CT_{\mathcal{T}} = (\mathcal{T}, \{C_i\}_{i=1}^4, \{C_{v_i}^{(1)}\}_{v_i \in L_{\mathcal{T}}}, \{C_{v_i}^{(2)}\}_{v_i \in L_{\mathcal{T}}})$, and $TK_u = (K, \{K_i\}_{i \in S})$ be a decryption token generated by $TokenGen(params, id_u, SK_u, CT_{\mathcal{T}})$, where $S \subset Att_u$ satisfies $\mathcal{T}$. We first prove the correctness of Equation (12). We have:

$$\begin{aligned}
L_i &= \frac{\hat{e}(K_i, C_{v_i}^{(1)})}{\hat{e}(K, -C_{v_i}^{(2)})} \\
&= \frac{\hat{e}(kSK_{i,u}, q_{v_i}(0)P_0)}{\hat{e}(kid_u, q_{v_i}(0)(-P_2 + PK_i))} \\
&= \frac{\hat{e}(kx_0P_1 + kX_1 + ksk_iid_u, q_{v_i}(0)P_0)}{\hat{e}(kid_u, P_2)^{-q_{v_i}(0)}\hat{e}(kid_u, PK_i)^{q_{v_i}(0)}} \\
&= \frac{\hat{e}(kx_0P_1, q_{v_i}(0)P_0)\hat{e}(kX_1, q_{v_i}(0)P_0)\hat{e}(kid_u, PK_i)^{q_{v_i}(0)}}{\hat{e}(kid_u, P_2)^{-q_{v_i}(0)}\hat{e}(kid_u, PK_i)^{q_{v_i}(0)}} \\
&= \hat{e}(kx_0P_1, q_{v_i}(0)P_0)\hat{e}(kX_1, q_{v_i}(0)P_0)\hat{e}(kid_u, P_2)^{q_{v_i}(0)} \\
&= \hat{e}(x_0P_1, P_0)^{kq_{v_i}(0)}\hat{e}(X_1, P_0)^{kq_{v_i}(0)}\hat{e}(id_u, P_2)^{kq_{v_i}(0)} \\
&= E_1^{kq_{v_i}(0)}E_2^{kq_{v_i}(0)}\hat{e}(id_u, P_2)^{kq_{v_i}(0)}.
\end{aligned} \tag{16}$$

So, Equation (12) is correct. Also, the correctness of Equations (13) and (14) is clear. Moreover, we see that

$$\begin{aligned}
C'^{k^{-1}}C_1C_3 &= (E_1^{kr}E_2^{kr})^{k^{-1}}ME_1^{-r}E_2^{-r} \\
&= (E_1^rE_2^r)ME_1^{-r}E_2^{-r} \\
&= M.
\end{aligned} \tag{17}$$

It proves the theorem. □

7.2. Security Proof

Theorem 2. *If the DBDH problem is hard relative to $\mathcal{G}$, then LW-FGAC construction is secure in the standard model.*

Proof. Let Π be our proposed LW-FGAC scheme, and $\mathcal{A}$ is a PPT adversary in the experiment $LW - FGAC_{\mathcal{A},\Pi}(n) = 1$ introduced in Section 6. In the following, we show that there exists a negligible function *negl* such that:

$$\Pr(LW - FGAC_{\mathcal{A},\Pi}(\lambda) = 1) \leq \frac{1}{2} + negl(\lambda), \tag{18}$$

where λ is the security parameter of the system. Suppose that $\mathcal{A}'$ is another PPT adversary that attempts to solve the DBDH problem. Recall that the adversary $\mathcal{A}'$ receives $(\lambda, q, G_1, G_2, \hat{e}, P, \alpha P, \beta P, \gamma P, \hat{e}(P,P)^z)$, where $P \leftarrow G_1$, $\alpha, \beta, \gamma \leftarrow \mathbb{Z}_q$, and z is equal to $\alpha\beta\gamma$ or is a uniform element of $\mathbb{Z}_q$. The aim of $\mathcal{A}'$ is to determine the case of z. $\mathcal{A}'$ runs $\mathcal{A}$ as a subroutine as follows:

1. Setup: At first, $\mathcal{A}'$ considers a universal attribute set $\mathbb{U}$, and for each $i \in \mathbb{U}$, chooses a uniform element $sk_i \in \mathbb{Z}_q$. Then, it selects $t \leftarrow \mathbb{Z}_q$ and $X \leftarrow G_1$ and sets

$$P_0 = P, \tag{19}$$
$$P_1 = \alpha P, \tag{20}$$
$$P_2 = tP, \tag{21}$$
$$x_0 P_0 = \beta P, \tag{22}$$
$$E_1 = \hat{e}(\beta P, P_1), \tag{23}$$
$$E_2 = \hat{e}(P, X).\hat{e}(\alpha P, \beta P)^{-1}, \tag{24}$$

and

$$PK_i = sk_i P, \tag{25}$$

for any attribute $i \in \mathbb{U}$. $\mathcal{A}'$ gives $params = (\lambda, q, G_1, G_2, \hat{e}, P_0, P_1, P_2, E_1, E_2, , \{PK_i\}_{i\in\mathbb{U}})$ to $\mathcal{A}$ as the global public parameters of the system. Note that, if we assume that the master secret-key $MSK = (x_0, P_1, X_1, \{sk_{a_i}\}_{i=1}^m)$ is chosen such that the following equations

$$x_0 = \beta, \tag{26}$$
$$X = \beta P_1 + X_1 = \alpha\beta P + X_1, \tag{27}$$

hold, then one can see that

$$\begin{aligned} E_2 &= \hat{e}(P, X).\hat{e}(\alpha P, \beta P)^{-1} \\ &= \hat{e}(P, \alpha\beta P + X_1).\hat{e}(\alpha P, \beta P)^{-1} \\ &= \hat{e}(P, \alpha\beta P).\hat{e}(P, X_1).\hat{e}(\alpha P, \beta P)^{-1} \\ &= \hat{e}(\alpha P, \beta P).\hat{e}(P, X_1).\hat{e}(\alpha P, \beta P)^{-1} \\ &= \hat{e}(P, X_1) \\ &\overset{(19)}{=} \hat{e}(P_0, X_1). \end{aligned} \tag{28}$$

So, E_2 is chosen correctly. The correctness of the other components of *params* can be easily checked.

2. Phase 1: For any data user with identifier id_u, $\mathcal{A}'$ makes a list L_{id_u} which is initially empty. When $\mathcal{A}$ submits a query $\mathcal{O}_{\textbf{User.KeyGen}}(Att, id_u)$, it sets $L_{id_u} = L_{id_u} \cup Att$ and computes

$$SK_{i,u} = X + sk_i id_u. \tag{29}$$

Combining Equations (20) and (22), we have:

$$\begin{aligned} SK_{i,u} &= X + sk_i id_u \\ &= \alpha\beta P + X_1 + sk_i id_u \\ &= \beta(\alpha P) + X_1 + sk_i id_u \\ &= \beta P_1 + X_1 + sk_i id_u \\ &= x_0 P_1 + X_1 + sk_i id_u. \end{aligned} \tag{30}$$

Also, by Equations (6) and (30), we see that $SK_{i,u}$ in Equation (29) is a valid secret-key.

3. Challenge: $\mathcal{A}$ declares an access tree $\mathcal{T}^*$ and two equal-length messages M_0 and M_1 such that there is no data user with identifier id_u such that L_{id_u} satisfies $\mathcal{T}^*$. $\mathcal{A}'$ selects $b \leftarrow \{0,1\}$ and $r' \leftarrow \mathbb{Z}_q$ and assumes that for an unknown $SK_O \in \mathbb{Z}_q$, $r' = \gamma + SK_O$. It sets

$$PK_O^{(1)} = E_2^{r'} \hat{e}(-\gamma P, X) \hat{e}(P, P)^z, \tag{31}$$

$$PK_O^{(2)} = r'P - \gamma P, \tag{32}$$

$$PK_O^{(3)} = r'P_2 - t\gamma P, \tag{33}$$

and for each $i \in \mathbb{U}$, it calculates

$$PK_{i,O} = r'(PK_i - P_2) - (sk_i \gamma P - t\gamma P). \tag{34}$$

Then, it runs $\{q_{v_i}(0)\}_{v_i \in L_{\mathcal{T}^*}} \leftarrow Share(r', q, \mathcal{T}^*)$ and calculates

$$C_1 = \hat{e}(P, P)^{-z} M_b. \tag{35}$$

Afterward, it sets $PCT^b_{\mathcal{T}^*} = (\mathcal{T}^*, C_1, \{q_{v_i}(0)\}_{v_i \in L_{\mathcal{T}^*}})$. Finally, it returns $PCT^b_{\mathcal{T}^*}$ and $PK_O = (PK_O^{(1)}, PK_O^{(2)}, PK_O^{(3)}, \{PK_{i,O}\}_{i \in \mathbb{U}})$ to $\mathcal{A}$. We see that

$$\begin{aligned} PK_O^{(2)} &= r'P - \gamma P \\ &= (\gamma + SK_O)P - \gamma P \\ &= SK_O P \\ &= SK_O P_0, \end{aligned}$$

$$\begin{aligned} PK_O^{(2)} &= r'P_2 - t\gamma P \\ &= (\gamma + SK_O)tP - t\gamma P \\ &= SK_O tP \\ &= SK_O P_2, \end{aligned}$$

and

$$\begin{aligned} PK_{i,O} &= r'(PK_i - P_2) - (sk_i\gamma P - t\gamma P) \\ &= (SK_O + \gamma)(PK_i - P_2) - \gamma(PK_i - P_2) \\ &= SK_O(PK_i - P_2). \end{aligned} \tag{36}$$

Therefore, $PK_O^{(2)}$, $PK_O^{(3)}$, and $PK_{i,O}$, for each $i \in \mathbb{U}$, are chosen correctly. Also, when $z = \alpha\beta\gamma$,

$$\begin{aligned} PK_O^{(1)} &= E_2^{r'}\hat{e}(-\gamma P, X)\hat{e}(P,P)^z \\ &= E_2^{r'}\hat{e}(-\gamma P, X)\hat{e}(P,P)^{\alpha\beta\gamma} \\ &= E_2^{r'}\hat{e}(P,X)^{-\gamma}\hat{e}(\alpha P, \beta P)^{\gamma} \\ &\overset{(28)}{=} E_2^{r'}E_2^{-\gamma} \\ &= E_2^{SK_O}, \end{aligned} \tag{37}$$

and

$$\begin{aligned} C_1 &= \hat{e}(P,P)^{-z}M_b \\ &= \hat{e}(\alpha P, \beta P)^{-\gamma}M_b \\ &= E_1^{-\gamma}M_b. \end{aligned} \tag{38}$$

Thus, assuming $z = \alpha\beta\gamma$ and the random element r in Part.Enc algorithm described in Section 6.3 is equal to γ, one can see that PK_O and $PCT_{\mathcal{T}^*}^b$ are chosen correctly.

4. Phase 2: $\mathcal{A}$ makes more queries for data users' secret-keys with the same restriction mentioned in the experiment presented in Section 5.2, and the adversary $\mathcal{A}'$ responds to the queries similar to Phase 1.
5. The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

Once the adversary $\mathcal{A}'$ receives b', it checks whether $b = b'$ or not. If so, it outputs 1. Otherwise, it returns 0.

As we have seen, if $z = \alpha\beta\gamma$, then PK_O and $PCT_{\mathcal{T}^*}^b$ are valid and therefore

$$\Pr(\mathcal{A}'(\lambda, q, P, G_1, G_2, \hat{e}, \alpha P, \beta P, \gamma P, \hat{e}(P,P)^{\alpha\beta\gamma}) = 1) = \Pr(LW - FGAC_{\mathcal{A},\Pi}(\lambda) = 1). \tag{39}$$

Also, it is clear that, if $z \in \mathbb{Z}_q$ is a uniform element, then the adversary $\mathcal{A}$ cannot get any partial information about M_b. Thus,

$$\Pr(\mathcal{A}'(\lambda, q, PG_1, G_2, \hat{e}, \alpha P, \beta P, \gamma P, \hat{e}(P,P)^z) = 1) = \frac{1}{2}. \tag{40}$$

On the other hand, by the hardness assumption of the DBDH problem, we have

$$\begin{aligned} &|\Pr(\mathcal{A}'(\lambda, q, P, G_1, G_2, \hat{e}, \alpha P, \beta P, \gamma P, D = \hat{e}(P,P)^{\alpha\beta\gamma}) = 1) - \\ &\Pr(\mathcal{A}'(\lambda, q, PG_1, G_2, \hat{e}, \alpha P, \beta P, \gamma P, D = \hat{e}(P,P)^z) = 1)| \leq negl(\lambda), \end{aligned} \tag{41}$$

for a negligible function $negl$. Combining Equations (39), (40), and (41), we get

$$\Pr(LW - FGAC_{\mathcal{A},\Pi}(\lambda) = 1) \leq \frac{1}{2} + negl(\lambda). \tag{42}$$

This proves the theorem. □

Corollary 1. *Our proposed system provides a secure lightweight encryption mechanism.*

Proof. As we have seen in Theorem 1, the ciphertext generated by the lightweight encryption process is valid and can be decrypted by the algorithms presented in Section 6.4. Also, considering the security game presented in Section 5.2, the threat model presented in Section 3.2, and Theorem 2, one can see that the encryption mechanism leaks no information about the underlying health data to any PPT adversary modeling a group of unauthorized data users that colludes with the CSP. Therefore, our encryption mechanism is lightweight and secure. □

8. Performance Analysis

In this section, we analyze the performance of our LW-FGAC scheme by comparing its execution time, storage cost, and communication overhead with some existing ABE schemes in terms of both actual execution time and asymptotic complexity. The employed notations in the asymptotic analysis are given in Table 3.

Table 3. Notations Employed in Our Asymptotic Analysis.

Notation	Description
$\|Att_u\|$	Carnality of a data user's attribute set
$\|\mathbb{U}\|$	Carnality of the universal attribute set
$\|L_{\mathcal{T}}\|$	Number of leaf nodes in an access tree $\mathcal{T}$
S	Carnality of a data user's attribute set satisfying a given access tree
T_{e_1}	Exponential operation time in G_1
T_{e_2}	Exponential operation time in G_2
T_p	Pairing operation time
l_{G_1}	Size of an element in G_1
l_{G_2}	Size of an element in G_2

In the asymptotic analysis, we considered three computational operations: exponential operation in G_1, exponential operation in G_2, and paring operation. As the other computational operations are significantly more efficient than the mentioned three operations, we ignore them in our analysis. Also, in measuring storage cost and communication complexity, we consider the size of elements in the groups G_1, G_2, and $\mathbb{Z}_q$.

We implement our scheme by using an Ubuntu 18.04 laptop with an Intel Core i5-2410M Processor 2.3 GHz, 6 GB RAM using python Pairing-Based Cryptography (pyPBC) and hashlib libraries [45,46]. Also, we use the Type A pairings and SHA-1 algorithm. Moreover, in this section, we use And-gates access structure $(a_1 \wedge \ldots \wedge a_n)$ as the access control policy.

In the following, we describe our asymptotic and actual execution results. In our implementation, we assume that the number of leaf nodes in the access tree and the number of data users' attributes are ranged between 10 to 100.

The actual execution times incurred by data owners and data users in the encryption and decryption phases are shown in Figure 7. As we see in part (a) of the figure, our encryption algorithm is significantly more efficient than the schemes presented in [27,35,36]. The mentioned fact is confirmed by the results given in Table 4. According to the figure, our scheme is more than 100 times faster than the schemes [27,35,36]. Also, as shown in Table 4, in [27], execution time is a function of the universal attribute set's carnality, $|\mathbb{U}|$. We measure its execution time when $\mathbb{U} \in \{100, 200\}$. One can see that this scheme is inefficient for large universal attribute sets, and data owners and data users have to perform a considerable amount of heavy computational operations. Also, Figure 8 and Table 5 compare the execution time of the encryption and decryption phases in LW-FGAC with the schemes presented in [27,35,36]. We see that the performance of our proposed scheme is acceptable in comparison with the other schemes.

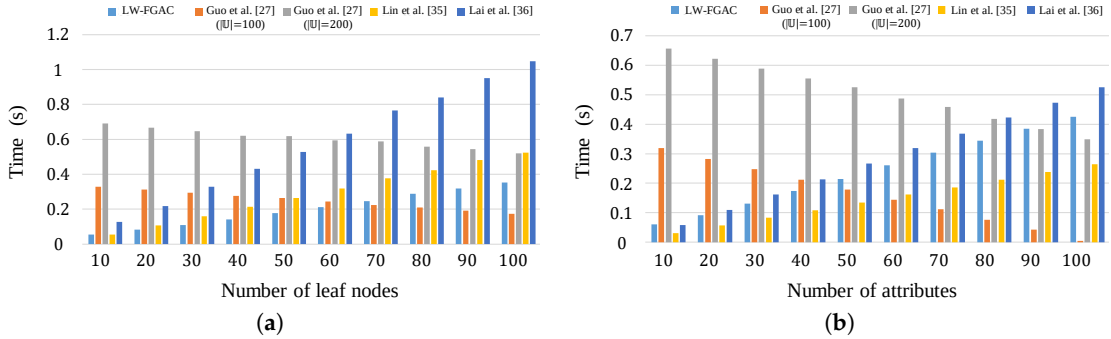

Figure 7. (**a**) Execution time of the encryption phase; (**b**) Execution time of the decryption phase.

Table 4. Comparison of Computational Overhead on Data Owners and Data Users.

Schemes	Encryption	Decryption
Guo et al. [27]	$(2\|\mathbb{U}\| - \|L_{\mathcal{T}}\| + 3)T_{e_1}$	$(2\|\mathbb{U}\| - 2\|S\| + 3)T_{e_1} + 3T_p + T_{e_2}$
Lin et al. [35]	$(2\|L_{\mathcal{T}}\| + 1)T_{e_1}$	$(\|S\| + 2)T_{e_1} + T_{e_2}$
Lai et al. [36]	$(6\|L_{\mathcal{T}}\| + 4)T_{e_1} + 2T_{e_2}$	$(\|S\| + 4)T_{e_1} + T_{e_2}$
LW-ABKS	T_{e_2}	$(\|S\| + 1)T_{e_1} + T_{e_2}$

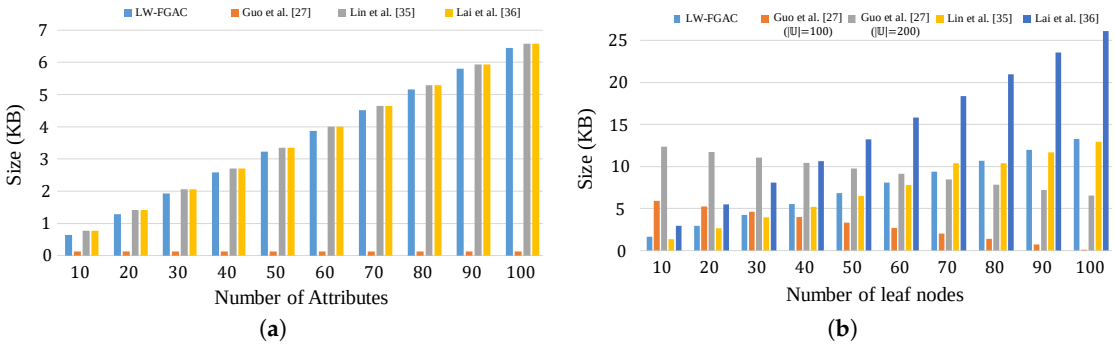

Figure 8. (**a**) Size of data users' attribute secret-key; (**b**) Length of a ciphertext.

Table 5. Computational Complexity in The Encryption And Decryption Phases.

Schemes	Encryption	Decryption
Guo et al. [27]	$(2\|\mathbb{U}\| - \|L_{\mathcal{T}}\| + 3)T_{e_1}$	$(2\|\mathbb{U}\| - 2\|S\| + 3)T_{e_1} + 3T_p + T_{e_2}$
Lin et al. [35]	$(2\|L_{\mathcal{T}}\| + 1)T_{e_1}$	$(2\|S\| + 1)T_p + \|S\|T_{e_2} + T_{e_1}$
Lai et al. [36]	$(6\|L_{\mathcal{T}}\| + 4)T_{e_1} + 2T_{e_2}$	$(4\|S\| + 2)T_p + 2\|S\|T_{e_2} + 2T_{e_1}$
LW-ABKS	$(2L_{\mathcal{T}} + 1)T_{e_1} + 2T_{e_2}$	$(\|S\| + 1)T_{e_1} + T_p(2\|S\| + 1) + T_{e_2}$

The storage overhead in our scheme and the schemes presented in [27,35,36] are given in Table 6 and Figure 9. Comparing the storage overhead in LW-FGAC with the others, one can see that the performance of LW-FGAC is acceptable. Also, we see that the data users' secret-key size in [27] is significantly shorter than the others. However, the length of a ciphertext in [27] grows linearly with $|\mathbb{U}| - |L_{\mathcal{T}}|$, where $|\mathbb{U}|$ is the number of attributes in the system, and $|L_{\mathcal{T}}|$ is the number of leaf nodes in the access tree associated with the ciphertext.

Also, Figure 10 and Table 7 present the communication overhead from data owners to the cloud server. We see that our proposed scheme significantly reduces the overhead as in our scheme data owners just transmit lightweight partially encrypted data to the cloud server. However, in the other scheme, a complete ciphertext should be given to the cloud, which consumes more communication resources.

Table 6. Storage Overhead.

Schemes	Key Size	Ciphertext Size
Guo et al. [27]	$2l_{G1}$	$(\|\mathbb{U}\| - \|L_{\mathcal{T}}\| + 2)l_{G_1}$
Lin et al. [35]	$(\|Att_u\| + 2)l_{G_1}$	$(\|L_{\mathcal{T}}\| + 1)l_{G_1}$
Lia et al. [36]	$(\|Att_u\| + 2)l_{G_1}$	$(4\|L_{\mathcal{T}}\| + 3)l_{G_1} + 2l_{G_2}$
LW-ABKS	$\|Att_u\|l_{G_1}$	$(2\|L_{\mathcal{T}} + 1\|l_{G_1} + 2l_{G_2})$

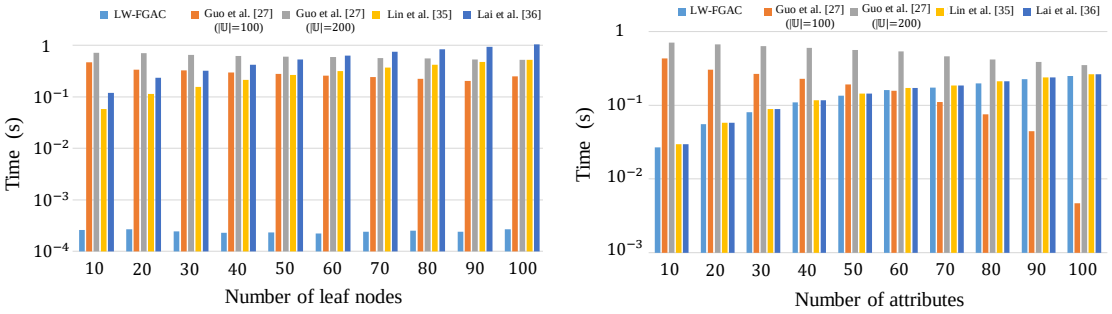

Figure 9. (**a**) Execution-time overhead on data owners in the encryption phase; (**b**) Execution-time overhead on data users in the decryption phase.

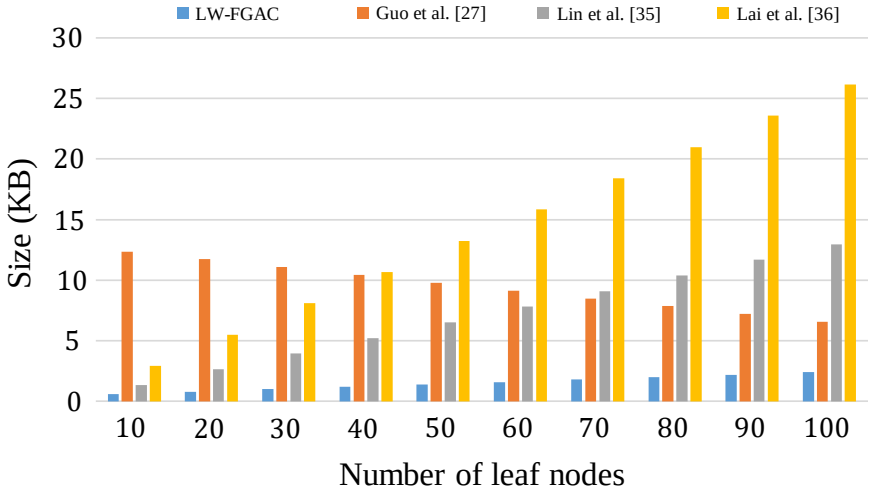

Figure 10. Communication overhead from data owners to the cloud.

Table 7. Communication Overhead from Data Owners to the Cloud.

Schemes	Size of The Transmitted Data
Guo et al. [27]	$(\|\mathbb{U}\| - \|L_{\mathcal{T}}\| + 2)l_{G_1}$
Lin et al. [35]	$(\|L_{\mathcal{T}}\| + 1)l_{G_1}$
Lia et al. [36]	$(4\|L_{\mathcal{T}}\| + 3)l_{G_1} + 2l_{G_2}$
LW-ABKS	$(\|L_{\mathcal{T}} + 1\|l_{\mathbb{Z}_q} + l_{G_2})$

9. Conclusions

We designed a novel attribute-based cryptographic scheme called lightweight fine-grained access control (LW-FGAC) for cloud-based wireless body area networks (WBANs). In our proposed scheme, by performing very lightweight computational operations, a data owner can encrypt its data under an access tree defined by itself. Any data user that its attributes satisfy the access policy can decrypt

the ciphertext. Also, in our designed system, the computational overhead on the data user side is very efficient, and most of the computations in the decryption phase are performed by the cloud service provider. We also provided the security definition for the new primitive, and we proved its security in the standard model under the hardness assumption of decisional bilinear Diffie-Hellman (DBDH) problem.

Author Contributions: X.L. and M.A. conceived the scheme. M.A. designed the scheme, proved the schemes security, analyzed the data, performed the experiments, and wrote the paper. X.L. and M.-R.S. reviewed and edited the manuscript.

Funding: The work was supported by the National Natural Science Foundation of China (No. U1804263 and 61702105), and the Opening Project of Guangdong Provincial Key Laboratory of Data Security and Privacy Protection (No. 2017B030301004-12).

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper. The authors also declare that they do not have any commercial or associative interest that represents a conflict of interest associated with the submitted paper.

References

1. Kevin, K.; Wan, H. *Unprecedented Global Aging Examined in New Census Bureau Report Commissioned by the National Institute on Aging*; National Institutes of Health: Bethesda, MD, USA, 20 July 2009. Available online: https://www.nih.gov/news-events/news-releases/unprecedented-global-aging-examined-new-ens\us-bureau-report-commissioned-national-institute-aging (accessed on 15 February 2020).
2. Chen, W.; Liu, J.J. *Future Population Trends in China: 2005-2050*; Centre of Policy Studies (CoPS), Victoria University: Melbourne, Australia, 2009.
3. Bodenheimer, T.; Chen, E.; Bennett, H.D. Confronting the growing burden of chronic disease: Can the US health care workforce do the job? *Health Aff.* **2009**, *28*, 64–74. [CrossRef] [PubMed]
4. Anderson, G.; Horvath, J. The growing burden of chronic disease in America. *Publ. Health Rep.* **2004**, *119*, 263–270. [CrossRef] [PubMed]
5. Lehnert, T.; Heider, D.; Leicht, H.; Heinrich, S.; Corrieri, S.; Luppa, M.; Riedel-Heller, S.; König, H.H. Health care utilization and costs of elderly persons with multiple chronic conditions. *Med. Care Res. Rev.* **2011**, *68*, 387–420. [CrossRef] [PubMed]
6. Yach, D.; Hawkes, C.; Gould, C.L.; Hofman, K.J. The global burden of chronic diseases: overcoming impediments to prevention and control. *JAMA* **2004**, *291*, 2616–2622. [CrossRef]
7. Movassaghi, S.; Abolhasan, M.; Lipman, J.; Smith, D.; Jamalipour, A. Wireless body area networks: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1658–1686. [CrossRef]
8. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: New York, NY, USA, 2005; pp. 457–473.
9. Lewko, A.; Waters, B. Decentralizing attribute-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: New York, NY, USA, 2011; pp. 568–588.
10. Li, J.; Yu, Q.; Zhang, Y.; Shen, J. Key-policy attribute-based encryption against continual auxiliary input leakage. *Inf. Sci.* **2019**, *470*, 175–188. [CrossRef]
11. Cui, Y.; Huang, Q.; Huang, J.; Li, H.; Yang, G. Ciphertext-policy attribute-based encrypted data equality test and classification. *Comput. J.* **2019**, *62*, 1166–1177. [CrossRef]
12. Attrapadung, N.; Imai, H. Dual-policy attribute based encryption. *International Conference on Applied Cryptography and Network Security*; Springer: New York, NY, USA, 2009; pp. 168–185.
13. Cao, H.; Leung, V.; Chow, C.; Chan, H. Enabling technologies for wireless body area networks: A survey and outlook. *IEEE Commun. Mag.* **2009**, *47*, 84–93. [CrossRef]
14. Li, X.; Ibrahim, M.H.; Kumari, S.; Sangaiah, A.K.; Gupta, V.; Choo, K.K.R. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. *Comput. Netw.* **2017**, *129*, 429–443. [CrossRef]
15. Chen, M.; Gonzalez, S.; Vasilakos, A.; Cao, H.; Leung, V.C. Body area networks: A survey. *Mob. Netw. Appl.* **2011**, *16*, 171–193. [CrossRef]
16. Zhang, Z.; Wang, H.; Vasilakos, A.V.; Fang, H. ECG-cryptography and authentication in body area networks. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 1070–1078. [CrossRef] [PubMed]

17. He, D.; Chen, C.; Chan, S.; Bu, J.; Vasilakos, A.V. ReTrust: Attack-resistant and lightweight trust management for medical sensor networks. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 623–632. [CrossRef] [PubMed]
18. Zhou, J.; Cao, Z.; Dong, X.; Lin, X.; Vasilakos, A.V. Securing m-healthcare social networks: challenges, countermeasures and future directions. *IEEE Wirel. Commun.* **2013**, *20*, 12–21. [CrossRef]
19. Ghamari, M.; Janko, B.; Sherratt, R.S.; Harwin, W.; Piechockic, R.; Soltanpur, C. A survey on wireless body area networks for ehealthcare systems in residential environments. *Sensors* **2016**, *16*, 831. [CrossRef] [PubMed]
20. Zhou, J.; Cao, Z.; Dong, X.; Xiong, N.; Vasilakos, A.V. 4S: A secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks. *Inf. Sci.* **2015**, *314*, 255–276. [CrossRef]
21. Liu, B.; Yan, Z.; Chen, C.W. Medium access control for wireless body area networks with QoS provisioning and energy efficient design. *IEEE Trans. Mob. Comput.* **2016**, *16*, 422–434. [CrossRef]
22. Shen, J.; Chang, S.; Shen, J.; Liu, Q.; Sun, X. A lightweight multi-layer authentication protocol for wireless body area networks. *Future Gener. Comput. Syst.* **2018**, *78*, 956–963. [CrossRef]
23. Li, M.; Lou, W.; Ren, K. Data security and privacy in wireless body area networks. *IEEE Wirel. Commun.* **2010**, *17*, 51–58. [CrossRef]
24. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.
25. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE symposium on security and privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
26. Zhou, Z.; Huang, D. On efficient ciphertext-policy attribute based encryption and broadcast encryption. In Proceedings of the 17th ACM conference on Computer and communications security, Chicago, IL, USA, 4–8 October 2010; pp. 753–755.
27. Guo, F.; Mu, Y.; Susilo, W.; Wong, D.S.; Varadharajan, V. CP-ABE with constant-size keys for lightweight devices. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 763–771.
28. Chen, C.; Chen, J.; Lim, H.W.; Zhang, Z.; Feng, D.; Ling, S.; Wang, H. Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In *Cryptographers' Track at the RSA Conference*; Springer: New York, NY, USA, 2013; pp. 50–67.
29. Lewko, A.; Okamoto, T.; Sahai, A.; Takashima, K.; Waters, B. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: New York, NY, USA, 2010; pp. 62–91.
30. Yao, X.; Chen, Z.; Tian, Y. A lightweight attribute-based encryption scheme for the Internet of Things. *Future Gener. Comput. Syst.* **2015**, *49*, 104–112. [CrossRef]
31. He, Q.; Zhang, N.; Wei, Y.; Zhang, Y. Lightweight attribute based encryption scheme for mobile cloud assisted cyber-physical systems. *Comput. Net.* **2018**, *140*, 163–173. [CrossRef]
32. Yang, Y.; Liu, X.; Deng, R.H. Lightweight break-glass access control system for healthcare internet-of-things. *IEEE Trans. Ind. Inf.* **2017**, *14*, 3610–3617. [CrossRef]
33. Yang, Y.; Zheng, X.; Guo, W.; Liu, X.; Chang, V. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Inf. Sci.* **2019**, *479*, 567–592. [CrossRef]
34. Xu, S.; Li, Y.; Deng, R.; Zhang, Y.; Luo, X.; Liu, X. Lightweight and Expressive Fine-grained Access Control for Healthcare Internet-of-Things. *IEEE Trans. Cloud Comput.* **2019**. [CrossRef]
35. Lin, S.; Zhang, R.; Ma, H.; Wang, M. Revisiting attribute-based encryption with verifiable outsourced decryption. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2119–2130. [CrossRef]
36. Lai, J.; Deng, R.H.; Guan, C.; Weng, J. Attribute-based encryption with verifiable outsourced decryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1343–1354.
37. Guo, W.; Li, J.; Chen, G.; Niu, Y.; Chen, C. A PSO-optimized real-time fault-tolerant task allocation algorithm in wireless sensor networks. *IEEE Trans. Parallel. Distrib. Syst.* **2014**, *26*, 3236–3249. [CrossRef]
38. Cheng, H.; Xiong, N.; Yang, L.T.; Jeong, Y.S. Distributed scheduling algorithms for channel access in TDMA wireless mesh networks. *J. Supercomput.* **2013**, *63*, 407–430. [CrossRef]
39. Yang, L.H.; Wang, Y.M.; Su, Q.; Fu, Y.G.; Chin, K.S. Multi-attribute search framework for optimizing extended belief rule-based systems. *Inf. Sci.* **2016**, *370*, 159–183. [CrossRef]

40. Mohd, B.J.; Hayajneh, T.; Vasilakos, A.V. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *J. Netw. Comput. Appl.* **2015**, *58*, 73–93. [CrossRef]
41. Cheng, H.; Su, Z.; Xiong, N.; Xiao, Y. Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model. *Inf. Sci.* **2016**, *329*, 461–477. [CrossRef]
42. Guo, W.Z.; Chen, J.Y.; Chen, G.L.; Zheng, H.F. Trust dynamic task allocation algorithm with Nash equilibrium for heterogeneous wireless sensor network. *Secur. Commun. Netw.* **2015**, *8*, 1865–1877. [CrossRef]
43. Ali, M.; Sadeghi, M.R.; Liu, X. Lightweight Revocable Hierarchical Attribute-Based Encryption for Internet of Things. *IEEE Access* **2020**, *8*, 23951–23964. [CrossRef]
44. Yang, Y.; Liu, X.; Deng, R.H.; Li, Y. Lightweight sharable and traceable secure mobile health system. *IEEE Trans. Dependable Secur. Comput.* **2017**, *17*, 78–91. [CrossRef]
45. The python pairing based cryptography library. November 2017. [online]. Available online: https://github.com/debatem1/pypbc (accessed on 10 December 2019).
46. The hashlib python library. [online]. Available online: https://docs.python.org/3/library/hashlib.html#module-hashlib (accessed on 10 December 2019).