sensors

MDPI

Article

# A Secured Proxy-Based Data Sharing Module in IoT Environments Using Blockchain

**Kwame Opuni-Boachie Obour Agyekum [1,2], Qi Xia [1,2,*], Emmanuel Boateng Sifah [1], Jianbin Gao [3], Hu Xia [1], Xiaojiang Du [4] and Moshen Guizani [5]**

1 Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu 611731, China; obour539@yahoo.com (K.O.-B.O.A.); emmanuelsifah@yahoo.com (E.B.S.); xiahu@uestc.edu.cn (H.X.)
2 CETC Big Data Research Institute Co., Ltd., Guiyang 550008, China
3 School of Resources and Environment, Center for Digital Health, University of Electronic Science and Technology of China, Chengdu 611731, China; gaojb@uestc.edu.cn
4 Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA; dxj@ieee.org
5 Department of College of Engineering, Qatar University, Doha, Qatar; mguizani@ieee.org
* Correspondence: xiaqi@uestc.edu.cn; Tel.: +86-139-8006-0339

Received: 30 January 2019; Accepted: 6 March 2019; Published: 11 March 2019


**Abstract:** Access and utilization of data are central to the cloud computing paradigm. With the advent of the Internet of Things (IoT), the tendency of data sharing on the cloud has seen enormous growth. With data sharing comes numerous security and privacy issues. In the process of ensuring data confidentiality and fine-grained access control to data in the cloud, several studies have proposed Attribute-Based Encryption (ABE) schemes, with Key Policy-ABE (KP-ABE) being the prominent one. Recent works have however suggested that the confidentiality of data is violated through collusion attacks between a revoked user and the cloud server. We present a secured and efficient Proxy Re-Encryption (PRE) scheme that incorporates an Inner-Product Encryption (IPE) scheme in which decryption of data is possible if the inner product of the private key, associated with a set of attributes specified by the data owner, and the associated ciphertext is equal to zero $(0)$. We utilize a blockchain network whose processing node acts as the proxy server and performs re-encryption on the data. In ensuring data confidentiality and preventing collusion attacks, the data are divided into two, with one part stored on the blockchain network and the other part stored on the cloud. Our approach also achieves fine-grained access control.

**Keywords:** Attribute-Based Encryption (ABE); blockchain; cyber-security; fine-grained access control; Inner-Product Encryption (IPE); Internet of Things (IoT); proxy re-encryption

## 1. Introduction

It has been estimated that there will be an enormous growth in the number of devices that will be connected to the internet by 2030 [1], and this will diminish the boundary between physical and digital worlds [2]. Human populace is not the main driver for this growth, but rather it is as a result of advances in wireless communication, embedded computing technologies, actuation and sensing that allow devices in a cyber physical world to become connected entities. The Internet of Things (IoT) is expected to fundamentally transform human daily activities, thereby outlining human-to-machine (H2M), machine-to-machine (M2M) and human-to-human (H2H) interactions in the connected world. Services provided by the IoT, which ensure safety, can be thought of as real drivers towards a better world of connectivity, as expressed by the authors of [3]. A complex task is the development of IoT systems and IoT services, which in particular is a crucial activity that requires

an in-depth research effort. In that essence, Casadei et al. [3] presented an "Opportunistic IoT Service" that extends the already existing IoT service models and considers some essential features for service provisioning. The authors of [4] also developed a toolset that synthesizes and validates human motion data aggregated from wearable computing devices, with the aim of enhancing the privacy of data owners. The toolset is developed to alleviate some of the challenges in data collection from IoT devices, and algorithm development. Their platform offers all the capabilities of existing datasets as well as enables the synthesis of data streams for users, scenarios and activities of their preference. Their proposal is cost effective and provides more extensive data for validation and system refinement.

Although these IoT platforms offer numerous opportunities and provide effective solutions for cyber-security, there are several challenges associated with the IoT. Providing a secured data sharing environment and also ensuring privacy, as the massive volumes of data generated by the IoT devices (either single devices or entire systems) are very sensitive, are a few of these challenges. Data owners tend to worry about how their data are used since the control is out of their hands. The security of their data is the most prominent issue in cloud computing, and this affects the performance of this paradigm [5,6]. Encrypting data before outsourcing them has promised to be a good way in mitigating the security concerns [7]. When the data are encrypted, it becomes difficult to share the data with users because the owner has to share the decryption key with those users, thereby granting access to the data. Another problem that arises from the sharing of keys is user revocation, where there should be denial of service for some users. What data owners usually do is invalidate the existing key by performing a re-encryption over the whole set of data with a new key, and in turn re-distributing the key to the (authorized) users. This action also becomes cumbersome and enormously involving when there are huge amounts of data to outsource, and the owner does not keep a copy of the outsourced data locally. Due to the generation and management of keys and ciphertexts, the provision of access control on encrypted data also becomes a challenge.

Attribute-based encryption (ABE), an encryption scheme first proposed by Sahai and Waters [8], achieves both access control and data security by granting different access rights to users based on their attributes. One of its characteristics is the revocation of access rights to users. The use of attribute-based encryption also aims at providing fine-grained access control, as it determines which user has access rights to which kind of data. ABE is an ideal tool to realize complex access control policies: the data to be accessed is associated with a set of attributes, and the privileges of the user are specified by a logical expression over these attributes. Other studies [9,10] also propose methods to attain fine-grained access control. Some fine-grained access control modules, as proposed by Yu et al. [11], however, place a heavy burden on the cloud (although believed to be more powerful than the data owner). The cloud, as a single point of service, is always expected to serve a greater number of users and therefore it becomes imperative to have as minimal overhead as possible to the cloud. Cloud service providers may also charge data owners on the amount of computations they may have to make. Therefore, the lower the computation, the lesser the cost. To ensure effective data sharing and user revocation, a system model employing Key Policy-Attribute Based Encryption (KP-ABE) and Proxy Re-Encryption (PRE) is proposed in [11]. The proposed scheme is, however, vulnerable to collusion attacks using a revoked user and the cloud server.

In this work, we incorporate the decentralized and consensus-driven blockchain technology and its underlying cryptographic primitives, and proxy re-encryption mechanism based on ABE to actualize the confidentiality of data. In this system, the data after re-encryption, which is done by the blockchain's processing node (a trusted proxy), are divided into two with one part stored on the blockchain and the other part on the cloud server. Therefore, it becomes impractical for a revoked user to obtain the data even after colluding with the honest, but curious cloud server. Furthermore, to lessen the burden on the cloud server, computations on the data are performed by the blockchain's processing nodes. We therefore propose a proxy re-encryption scheme based on a well-established ABE scheme proposed by Park [12].

It's true edge computing brings a better satisfaction to IoT devices. However, the storage of the data is done on the cloud and not much processing is done by/on the cloud. All processes are executed by the blockchain processing nodes, which have more computing power than the resource-constrained IoT devices. Moreover, the protocol still works, be it on the cloud or at the edge, since the main focus of this paper is on the security scheme. Due to constraints on resources by the IoT devices, the implementation of the security model, which is part of the computations and processing, is done by the blockchain network because it has enough processing power. Therefore, there are no specific hardware/software requirements for the resource-constrained IoT devices.

To summarize, our proxy re-encryption satisfies fine-grained access control in that users have access right to different sets of data, which is made possible by the ABE scheme. Our scheme is also collusion resistant as the cloud server and/or the proxy and the (revoked) user cannot collude to access data. This is made possible because the blockchain network is a decentralized system and all processes (transactions) are monitored by every participant on the network, and also recorded and stored into blocks. Furthermore, there is an appreciable level of trust between the data owner and the users due to the utilization of blockchain, as it ensures a trustworthy environment among participants involved. The proxy is uni-directional, as it transforms a ciphertext $C$ into a ciphertext $C'$ in only one direction, but not in the reverse transform.

The remainder of this paper is organized as follows. In Section 2, related works on the cryptographic primitives, IoT and blockchain are reviewed. In Section 3, we introduce the notations to be used in this paper, while the system model is formulated in Section 4. Our proposed scheme and its security model are presented in Sections 5 and 6, respectively. Implementation and performance analysis are presented in Section 7, while Section 8 provides a set of discussions. Section 9 concludes the paper.

## 2. Related Works

The secured sharing of data among several users via a cloud service provider is extensively researched in [13–15]. Mambo and Okamoto's [16] novel PRE scheme has been adopted as the technique to achieve this, and it was further extended by Blaze et al. [17] by basing their findings on the El-Gamal cryptosystem [18]. In their work, a proxy can transform a message encrypted under Alice's key into an encryption of the same message under Bob's key because it utilizes a re-encryption key. While effective data sharing can be achieved by these schemes by meeting some security requirements and properties, there is no enforcement of fine-grained access control on the shared data.

Attribute-based proxy encryption techniques [19–22] have therefore been adopted to enforce this. Both the ciphertext and the private key of the user are associated with an attribute set in the ABE scheme, and decryption is possible when there is a match between the set of attributes for both the private key and the ciphertext [8,23,24]. These approaches, nevertheless, help only in the adversary not obtaining any information about the encrypted message. Katz et al. [25] therefore presented an attribute hiding scheme for a class of predicates. This was known as Inner-Product Encryption (IPE), and it preserves the confidentiality of the attributes associated with the ciphertext. Following that, a hierarchical IPE scheme that uses an $n$-dimensional vector space in bilinear maps of prime order was proposed by Okamoto et al. [26], and the full security under the standard model was achieved. Park [12] therefore presented an IPE scheme that supports an attribute hiding property, and also is secure against Decisional Bilinear Diffie–Hellman (D-BDH) and decisional linear assumptions.

Du et al. [27] presented an efficient and scalable key management scheme for heterogeneous sensor networks. Their scheme utilizes the fact that there is a lower communication and computational cost when a sensor only communicates with a small portion of its neighbors. An Elliptic Curve Cryptographic (ECC) scheme is used to further improve key management, as it also reduces sensor storage requirement and energy consumption while achieving better security. Xiao et al. [28] surveyed the various techniques utilized in the key management for Wireless Sensor Networks (WSNs). Their survey paper looks at both the advantages and disadvantages of the various techniques.

It is realized that no key distribution technique is ideal to all the scenarios where the sensor networks are deployed, and therefore the technique being employed should meet the requirements of both the application in question and the resources of the individual sensor networks. The authors of [29] presented an effective key management scheme for heterogeneous sensor networks, which is quite similar to the work in [27]. Their work portrays how efficient the performance of their scheme is, and that it significantly achieves a better security than existing sensor network key management schemes. Du et al. in [30] presented the security issues in WSNs. Quite similar to the aforementioned sensor-related papers, they investigated schemes that achieve better security and also lower computational cost for the sensor networks.

Blockchain technology offers a suitable platform that can be used for numerous applications in medical care. Improving the security in medical data sharing and automating the delivery of health-related notifications are the massive potentials of this technology, and they are compatible with the Health Insurance Portability and Accountability Act (HIPAA) [31]. Several authors have provided blockchain health-related applications [32–35]. The authors of [32] determined the current challenges of Electronic Medical Record (EMR) systems and the potential they have in providing solutions to security challenges and interoperability, with the use of blockchain technology. Focus has been on the application of blockchain to Electronic Health Records (EHRs) to facilitate interoperability. Medrec, a prototype released by MIT, expresses a practical way of sharing healthcare data between EHRs and blockchain [33]. A secure and scalable access control system for confidential information sharing on blockchain was also presented by the authors of [34]. Their results portray the effectiveness of their system in instances where traditional methods of access control failed. Yue et al. designed a concept for an application that presents patients with the opportunity to grant access to information about their health records to designated individuals [35]. The authors of [36] proposed a novel protocol that achieves patient privacy preservation by applying the concept of blockchain in an eHealth platform.

A possible efficient data sharing platform among interested parties and the preservation of privacy are a just few of the opportunities blockchain technology offers. For blockchain to reach its maximum potential, it is essential to tackle one of the most important problems facing this technology: data access control. This work therefore places more emphasis on providing a secured data access control in a data sharing environment. A blockchain processing node acts as a proxy and performs re-encryption on data that are given to a secondary user. Our system preserves data confidentiality and integrity, and avoids collusion attacks. Fine-grained access control is also achieved.

## 3. Preliminaries

We introduce some of the notations that will be utilized throughout this paper in this section.

### *3.1. Bilinear Maps*

Our protocol is based on bilinear maps [37]. Let $G$ and $G_T$ be two multiplicative cyclic groups that have a prime order $p$, and $g$ be a generator of $G$. A bilinear map $e : G \times G \rightarrow G_T$ has the following properties:

1. Bilinear: For all $a, b \in Z_p$, $g, h \in G$, then $e(g^a, h^b) = e(g, h)^{ab}$ can be computed efficiently.
2. The map is non-degenerate. That is, if $g$ generates $G$ and $h$ also generates $G$, then $e(g, h)$ generates $G_T$. In addition, $e(g, h) \neq 1$. The map does not send all pairs in $G \times G$ to the identity in $G_T$.
3. It is computable; there exists an efficient algorithm to compute the map $e(g, h)$ for any $g, h \in G$.

Note that $e(,)$ is symmetric since $e(g^a, h^b) = e(g, h)^{ab} = e(g^b, h^a)$.

### *3.2. Inner-Product Encryption (IPE)*

The Inner Product Encryption (IPE) scheme, as proposed in [12], is an attribute-based encryption technique in which both ciphertext(s) and private (secret) key(s) are associated with vectors. Access to and decryption of an encrypted data can only be possible if and only if the inner product of the private

key, which is related to vector $\overrightarrow{v}$, and the ciphertext, also related to vector $\overrightarrow{x}$, is 0. That is, for the two vectors, $(\overrightarrow{v} \cdot \overrightarrow{x}) = \sum_{i=1}^{n} x_i \cdot v_i \mod p = 0$. Let $\sum$ be a set of attributes peculiar to particular encrypted data that involves vector $\overrightarrow{v}$ and has a dimension of $n$. Denote $F$ as representing a predicate class that involves an inner product over vectors, i.e., $F = f_{\overrightarrow{x}} | \overrightarrow{x} \in \sum$ such that $f_{\overrightarrow{x}}(\overrightarrow{v}) = 1$ iff $(\overrightarrow{x} \cdot \overrightarrow{v}) = 0$. Two $n$-dimensional vectors, $\overrightarrow{x} = (x_1, ..., x_n)$ and $\overrightarrow{v} = (v_1, ..., v_n)$, all belonging to the set of attributes, $\sum$, are, respectively, utilized in the encryption and key decryption phases.

We incorporate the rationale behind a proxy's re-encryption key (RE key) into this work by using the IPE scheme to transform a ciphertext associated with a vector into a new ciphertext associated with another vector but encrypts the same message ($m \in M$). We ensure that there is no revealing of the information about the encrypted data.

### *3.3. Attribute Based Encryption (ABE)*

There are two main classifications of ABE schemes, namely Ciphertext Policy-Attribute Based Encryption (CP-ABE) [23] and Key Policy-Attribute Based Encryption (KP-ABE) [38]. In this paper, we make use of KP-ABE, as the data are encrypted by a set of attributes and the private keys of the users are associated with the access structure of KP-ABE. Thus, if the attribute of the encrypted data satisfies the access structure of the user's private key, decryption of the ciphertext can occur.

### *3.4. Proxy Re-Encryption (PRE)*

The notion of "atomic proxy cryptography" is the basis for proxy re-encryption, which was first introduced by Mambo and Okamoto [16]. This scheme basically makes use of a semi-trusted proxy that transforms the ciphertext for Alice into a ciphertext for Bob, without actually knowing or gaining access to the plaintext. Popular, well-known proxy re-encryption schemes are the Blaze, Bleumer and Strauss (BBS) [17] and the Ateniese, Fu, Green and Hohenberger (AFGH) [39] schemes, which are based on El Gamal and Bilinear maps cryptographic algorithms, respectively. In this work, the blockchain processing node (a trusted entity) serves as the proxy, and performs re-encryption on the data.

### *3.5. Blockchain Network*

Blockchain technology, originally proposed by Satoshi Nakamoto [40], acts as a shared, decentralized ledger to record transactions. Public, private and consortium blockchains are the three main types of blockchain. For decentralized networks and offering transparency, public blockchain is predominantly used. Private and consortium blockchains are, however, preferred when more control and privacy are of the essence. Consensus and decentralization, key features of blockchain, are the reasons for using blockchain technology in our system. Moreover, our blockchain's processing node serves as the trusted proxy that performs the re-encryption on the data before they are given to the secondary user. Proof-of-work (PoW) and Practical Byzantine Fault Tolerance (PBFT) provide security offered by the use of this technology. They utilize the agreement of nodes in the addition of a block to the chain, which acts as a ledger for all transactions.

Blockchain has helped in the effectiveness and advancement of many industries. It is also capable of implementing smart contracts, which are programmable scripts that automatically execute actions based on pre-defined triggers. The smart contracts are called upon when a data user requests access to data. Prior to the data being sent to the cloud, the owner specifies how its data are to be used and gives the details to the blockchain network. A processing node then embeds the contract into the data being given to the requestor. Our blockchain keeps logs of the transactions to achieve effective auditing.

Due to privacy concerns, our system utilizes the distributed ledger property of the blockchain, namely immutability, for authenticity and verifiability, and also the use of the consortium blockchain. Only authorized users can gain access to data. This enhances transparency for data owners, and allows them to effectively manage their data.

A block consists of a single event, with the event spanning from the time a request is made to when the block is broadcast onto the blockchain. Consensus nodes are responsible for mining and reporting all activities. A block is made up of a format that distinctively describes the block. This is followed by a block size, and then a block header, which is hashed with $sha256(sha256())$ as implemented in Bitcoin headers [40]. The block size contains the size of the block and the header ensures immutability. Changing a block header, in order to falsify a piece of information, requires a change to all headers starting from the genesis (parent) block.

A block header also contains the version number which indicates the validation rules to follow. The previous block's hash is also contained in the header. A timestamp is also included in the header and it indicates when the block was created. A target difficulty, which is a value that indicates how processing is achieved by the consensus nodes, is also found in the header. This makes processing difficult for malicious nodes but solvable by verified consensus nodes. There is also an arbitrary number generated by the consensus nodes, which modifies the header hash in order to produce a hash below the target difficulty. This is called a nonce. A transaction counter is found in the block, whose function is to record the total number of transactions in the entire block. The transaction is made up of the consensus transaction and the user transaction. Each type comprises a timestamp and the data. A block locktime defines the structure for the entire block. This is a timestamp that records the last entry of a transaction as well as the closure of a block. When all conditions are met, the block is then broadcast onto the blockchain network.

For scalability concerns, our blockchain stores hashes of transactions. Transactions on this blockchain typically include data requests, data processing (encryption and/or re-encryption), and data access.

## 4. System Model

### 4.1. Problem Statement

We demonstrate a simple IoT file/data sharing scenario in a healthcare environment for the sake of clarity, where we consider a patient whose data can only be accessed by his/her physician, pharmacist or relatives. Patients' data are normally collected and collated by health sensors that are usually bound to them, and uploaded onto a cloud server after recording. Before a patient's medical data are outsourced to a cloud server, the patient encrypts their own data under a set of attributes, which indicates the access privilege on the data. The patient then gives the details of all authorized users to the blockchain's processing node. Thus, access to a patient's data can be possible only if the user satisfies the attribute set and also uses the private key related to that attribute set.

However, there may be an instance where a physician might share the patient's data, depending on the kind of ailment they are treating, with other healthcare professionals who are not in the same hospital and therefore have a different access policy on the data. It now lies of the proxy (blockchain processing node) to re-encrypt the patient's data under the patient's attribute set to the new attribute set in a way that does not reveal any information about the data and its corresponding attributes. This must also be done in an efficient and secured way. The model of our system is presented in Figure 1.

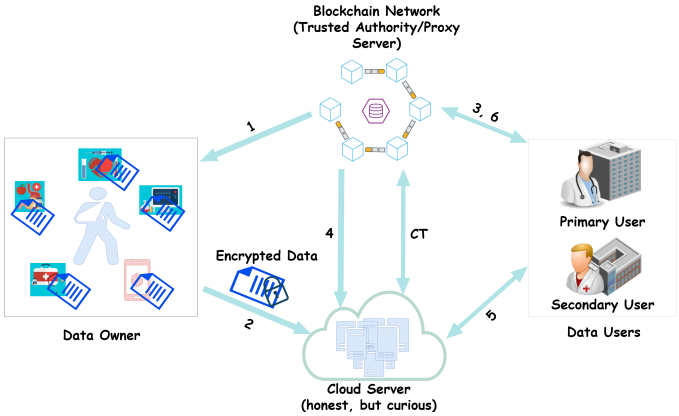


**Figure 1.** System model.

### *4.2. System Overview*

1. **Data Owner**: This is the entity (the patient in this case) whose data are to be accessed. Access is possible if and only if the private key of the data user corresponds to the attribute set specified by the data owner.
2. **Data User**: This is the entity who wants to make use of the data from the owner. Both the data owner and user(s) should be registered on the blockchain.
3. **Cloud Server**: This is the repository for the data from the owner. All encrypted files are sent to the cloud server (honest, but curious) through a secured communication channel.
4. **Blockchain Network**: This primarily consists of the following entities:

    - Issuer: This entity registers the participants (data owner and users) on the blockchain network. It gives out membership keys to them and that serves as their identity (ID).
    - Verifier: The verifier, which also serves as an authentication unit, checks whether a user who makes an access request or a data owner who uploads its data onto the cloud, are actually members of the blockchain network.
    - Processing node: This is the heartbeat of the blockchain network. All processes (transactions) that ever occur on the network are performed by this entity. In this work, however, it serves as the (trusted) proxy that oversees the re-encryption process.
    - Smart contract center: This unit prepares the contract that binds how data are to be used.

The various processes that happen in the system model are described below:

1. The proxy generates a secret key, $SK$, and a public key $P_{pub}$, and hands the public key and access policy to the data owner. That is, the data owner is given $\{P_{pub}, H_{access}\}$.
2. The patient encrypts the data with the attribute set and sends the encrypted data to the cloud through a secured channel. The encrypted data are $CT = \{Enc\,(M, \overrightarrow{x})\}$.
3. The data user makes a request for the data.
4. The proxy accesses the permission rights of the data users from the cloud server. After accessing it, the blockchain network, which also serves as a trusted authority, gives the private key to the user according to the user's attributes.
5. Users can now access data from the cloud server.
6. The primary user is given $PK_{\overrightarrow{v}}$ while the secondary user is given $PK_{\overrightarrow{v'}}$. The proxy generates a re-encryption key $REKey$ and transforms the policy set $H \rightarrow H'$ for the secondary user who wants the shared data from the primary user but holds a different access policy, $H'$.

## 5. The Scheme

As in several security algorithms, our proposed scheme consists of the following algorithms: *Setup*, *KGen*, *Encrypt*, *RKGen*, *ReEncrypt*, and *Decrypt*. The IPE scheme, as presented in [12], is adopted

in this work and therefore most of the algorithms will be the same. *Setup*, *KGen*, *Enrypt* and *Decrypt* have been previously presented in [12].

The assumption is made here that $\sum = (Z_p)^n$ is the set of attributes bound to data, where $n$ is the dimension of the vectors, $\overrightarrow{x}$ and $\overrightarrow{v}$, and $p$ is the prime order of the group, $Z$. For any vector $\overrightarrow{v} = (v_1, ..., v_n) \in \sum$, each element $v_i$ belongs to the set $Z_p$. The algorithms are as follows.

$(P_{pub}, SK) \leftarrow$ **Setup** $(\lambda, n)$: With any security parameter $\lambda \in Z^+$, the setup algorithm runs $\sigma(\lambda)$ after which a tuple $(p, G, G_2, e)$ is obtained. A random generator $g \in G$, along with random exponents $\delta_1, \delta_2, \theta_1, \theta_2, \{w_{1,i}\}_{i=1}^n, \{t_{1,i}\}_{i=1}^n, \{f_{1,i}, f_{2,i}\}_{i=1}^n$ and $\{h_{1,i}, h_{2,i}\}_{i=1}^n$, found in $Z_p$ are all selected. A random element, $g_2 \in G$, is also selected. Furthermore, it selects a random number, $\Psi \in Z_p$ and obtains the set of elements $\{w_{2,i}\}_{i=1}^n, \{t_{2,i}\}_{i=1}^n$ in $Z_p$ with constraints such that

$$\Psi = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \Psi = \theta_1 t_{2,i} - \theta_2 t_{1,i}$$

The setup algorithm then computes

$$W_{1,i} = g^{w_{1,i}}, W_{2,i} = g^{w_{2,i}}, T_{1,i} = g^{t_{1,i}}, T_{2,i} = g^{t_{2,i}}, F_{1,i} = g^{f_{1,i}}, F_{2,i} = g^{f_{2,i}}, H_{1,i} = g^{h_{1,i}}, H_{2,i} = g^{h_{2,i}}$$

Now, the following notations are also given:

$$Q_1 = g^{\delta_1}, Q_2 = g^{\delta_2}, R_1 = g^{\theta_1}, R_2 = g^{\theta_2}, g_1 = g^{\Psi}, \mathrm{Y} = e(g, g_2)$$

The public $P_{pub}$ and secret $SK$ keys are then, respectively, computed as:

$$P_{pub} = \left(g, g^{\Psi}, \{W_{1,i}, W_{2,i}, F_{1,i}, F_{2,i}\}_{i=1}^n, \{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^n, \{Q_i, R_i\}_{i=1}^2, \mathrm{Y}\right) \in G^{8n+6} \times G_T$$

$$SK = \left(\{w_{1,i}, w_{2,i}, t_{1,i}, t_{2,i}, f_{1,i}, f_{2,i}, h_{1,i}, h_{2,i}\}_{i=1}^n, \{\delta_i, \theta_i\}_{i=1}^2, g_2\right) \in Z_p^{8n+4} \times G$$

$PK_{\overrightarrow{v}} \leftarrow$ **KGen** $\left(SK, \overrightarrow{v}\right)$: For a vector $\overrightarrow{v} = (v_1, ..., v_n)$, the algorithm selects random exponents $\lambda_1$, $\lambda_2$, $\{r_i, \phi_i\}_{i=1}^n$ in $Z_p$, and creates a private key $PK_{\overrightarrow{v}} = \left(K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^n, \{K_{3,i}, K_{4,i}\}_{i=1}^n\right) \in G^{4n+2}$. The composition of the various elements in the $PK_{\overrightarrow{v}}$ is defined as follows:

$$\left\{K_{1,i} = g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}, K_{2,i} = g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\right\}_{i=1}^n, \left\{K_{3,i} = g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}, K_{4,i} = g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right\}_{i=1}^n,$$

$$K_A = g_2 \prod_{i=1}^n K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-h_{1,i}} K_{4,i}^{-h_{2,i}}, K_B = \prod_{i=1}^n g^{-(r_i + \phi_i)}$$

$CT \longleftarrow$ **Encrypt** $(P_{pub}, \overrightarrow{x}, M)$: To encrypt a message $M \in G_T$ and a vector $\overrightarrow{x} = (x_1, ..., x_n) \in (Z_p)$ under the public key $P_{pub}$, the algorithm selects random elements $\{s_i\}_{i=1}^4 \in Z_p$ and uses them to compute the ciphertext $CT$ as follows:

$$CT = (g^{s_2}, g_1^{s_1}, \left\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{x_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{x_i s_3}\right\}_{i=1}^n, \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{x_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{x_i s_4}\right\}_{i=1}^n,$$

$$\left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{x_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{x_i s_4}\right\}_{i=1}^n, \mathrm{Y}^{-s_2} M) \in G^{4n+2} \times G_T$$

where $s_2 = \Psi s_1$.

$REKey_{\overrightarrow{v}} \leftarrow$ **RKGen** $(SK, \overrightarrow{v}, \overrightarrow{x'})$: **KGen** algorithm is first called and a random element, $l \in Z_p$, is selected. It then computes $\alpha$, $\alpha^{\delta_2}$, $\alpha^{-\delta_1}$, $\alpha^{\theta_2}$, and $\alpha^{-\theta_1}$, where $\alpha = g_2^l$. The **Encrypt** algorithm is then called to encrypt $\alpha$ under the vector $\overrightarrow{x'}$ by utilizing $Encrypt(P_{pub}, \overrightarrow{x'}, \alpha)$. The output is a ciphertext

$CT_A$. The *RKGen* algorithm then selects random exponents $\{\lambda_i'\}_{i=1}^{2}$, $\{r_i', \phi_i'\}_{i=1}^{n} \in Z_p$ and uses them to compute $REKey_{\overrightarrow{v}}$ as follows:

$$\left\{K_{1,i}' = g^{-\delta_2 r_i'} g^{\lambda_1' v_i w_{2,i}} \alpha^{\delta_2}, K_{2,i}' = g^{\delta_1 r_i'} g^{-\lambda_1' v_i w_{1,i}} \alpha^{-\delta_1}\right\}_{i=1}^{n}, \left\{K_{3,i}' = g^{-\theta_2 \phi_i'} g^{\lambda_2' v_i t_{2,i}} \alpha^{\theta_2}, K_{4,i}' = g^{\theta_1 \phi_i'} g^{-\lambda_2' v_i t_{1,i}} \alpha^{-\theta_1}\right\}_{i=1}^{n},$$

$$K_A' = g_2 \prod_{i=1}^{n} K_{1,i}^{'-f_{1,i}} K_{2,i}^{'-f_{2,i}} K_{3,i}^{'-h_{1,i}} K_{4,i}^{'-h_{2,i}}, K_B' = \prod_{i=1}^{n} g^{-(r_i' + \phi_i')}$$

$CT' \longleftarrow$ **ReEncrypt** $(REKey_{\overrightarrow{v}}, CT)$: On input of the ciphertext $CT$ and the re-encryption key $REKey_{\overrightarrow{v}}$, this algorithm first checks whether the attributes list of the user in $REKey_{\overrightarrow{v}}$ satisfies the attribute set of the $CT$. If that is not the case, it returns $\perp$; else, $\forall i = \{1, ..., n\}$, the algorithm first computes the following:

$$\prod_{i=1}^{n} e\left(C_{1,i}, K_{1,i}'\right) \cdot e\left(C_{2,i}, K_{2,i}'\right) \cdot e\left(C_{3,i}, K_{3,i}'\right) \cdot e\left(C_{4,i}, K_{4,i}'\right)$$

$$= \prod_{i=1}^{n} e\left(g^{w_{1,i} s_1} g^{f_{1,i} s_2} g^{\delta_1 x_i s_3}, g^{-\delta_2 r_i'} g^{\lambda_1' v_i w_{2,i}} \alpha^{\delta_2}\right) \cdot e\left(g^{w_{2,i} s_1} g^{f_{2,i} s_2} g^{\delta_2 x_i s_3}, g^{\delta_1 r_i'} g^{-\lambda_1' v_i w_{1,i}} \alpha^{-\delta_1}\right)$$

$$\cdot e\left(g^{t_{1,i} s_1} g^{h_{1,i} s_2} g^{\theta_1 x_i s_4}, g^{-\theta_2 \phi_i'} g^{\lambda_2' v_i t_{2,i}} \alpha^{\theta_2}\right) \cdot e\left(g^{t_{2,i} s_1} g^{h_{2,i} s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi_i'} g^{-\lambda_2' v_i t_{1,i}} \alpha^{-\theta_1}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{w_{1,i} s_1}, g^{-\delta_2 r_i'}\right) \cdot e\left(g^{f_{1,i} s_2}, g^{-\delta_2 r_i'} g^{\lambda_1' v_i w_{2,i}} \alpha^{\delta_2}\right) \cdot e\left(g^{\delta_1 x_i s_3}, g^{\lambda_1' v_i w_{2,i}}\right) \cdot e\left(g^{w_{1,i} s_1}, \alpha^{s_2}\right) \cdot e\left(g^{w_{2,i} s_1}, g^{\delta_1 r_i'}\right)$$

$$\cdot e\left(g^{f_{2,i} s_2}, g^{\delta_1 r_i'} g^{-\lambda_1' v_i w_{1,i}} \alpha^{-\delta_1}\right) \cdot e\left(g^{\delta_2 x_i s_3}, g^{-\lambda_1' v_i w_{1,i}}\right) \cdot e\left(g^{w_{2,i} s_1}, \alpha^{-\delta_1}\right) \cdot e\left(g^{t_{1,i} s_1}, g^{-\theta_2 \phi_i'}\right)$$

$$\cdot e\left(g^{h_{1,i} s_2}, g^{-\theta_2 \phi_i'} g^{\lambda_2' v_i t_{2,i}} \alpha^{\theta_2}\right) \cdot e\left(g^{\theta_1 x_i s_4}, g^{\lambda_2' v_i t_{2,i}}\right) \cdot e\left(g^{t_{1,i} s_1}, \alpha^{\theta_2}\right) \cdot e\left(g^{t_{2,i} s_1}, g^{\theta_1 \phi_i'}\right)$$

$$\cdot e\left(g^{h_{2,i} s_2}, g^{\theta_1 \phi_i'} g^{-\lambda_2' v_i t_{1,i}} \alpha^{-\theta_1}\right) \cdot e\left(g^{\theta_2 x_i s_4}, g^{-\lambda_2' v_i t_{1,i}}\right) \cdot e\left(g^{t_{2,i} s_1}, \alpha^{-\theta_1}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{-\delta_2 w_{1,i}}, g^{r_i' s_1}\right) \cdot e\left(g^{s_2}, \left(g^{-\delta_2 r_i'} g^{\lambda_1' v_i w_{2,i}} \alpha^{\delta_2}\right)^{f_{1,i}}\right) \cdot e\left(g, g\right)^{\lambda_1' \delta_1 w_{2,i} x_i v_i s_3} \cdot e\left(g^{w_{1,i} s_1}, \alpha^{\delta_2}\right) \cdot e\left(g^{\delta_1 w_{2,i}}, g^{r_i' s_1}\right)$$

$$\cdot e\left(g^{s_2}, \left(g^{\delta_1 r_i'} g^{-\lambda_1' v_i w_{1,i}} \alpha^{-\delta_1}\right)^{f_{2,i}}\right) \cdot e\left(g, g\right)^{-\lambda_1' \delta_2 w_{1,i} x_i v_i s_3} \cdot e\left(g^{w_{2,i} s_1}, \alpha^{-\delta_1}\right) \cdot e\left(g^{-\theta_2 t_{1,i}}, g^{\phi_i' s_1}\right)$$

$$\cdot e\left(g^{s_2}, \left(g^{-\theta_2 \phi_i'} g^{\lambda_2' v_i t_{2,i}} \alpha^{\theta_2}\right)^{h_{1,i}}\right) \cdot e\left(g, g\right)^{\lambda_2' \theta_1 t_{2,i} x_i v_i s_4} \cdot e\left(g^{t_{1,i} s_1}, \alpha^{\theta_2}\right) \cdot e\left(g^{\theta_1 t_{2,i}}, g^{\phi_i' s_1}\right)$$

$$\cdot e\left(g^{s_2}, \left(g^{\theta_1 \phi_i'} g^{-\lambda_2' v_i t_{1,i}} \alpha^{-\theta_1}\right)^{h_{2,i}}\right) \cdot e\left(g, g\right)^{-\lambda_2' \theta_2 t_{1,i} x_i v_i s_4} \cdot e\left(g^{t_{2,i} s_1}, \alpha^{-\theta_1}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{\delta_1 w_{2,i} - \delta_2 w_{1,i}}, g^{r_i' s_1}\right) \cdot e\left(g^{\theta_1 t_{2,i} - \theta_2 t_{1,i}}, g^{\phi_i' s_1}\right) \cdot e\left(g^{s_2}, K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}} K_{3,i}^{'h_{1,i}} K_{4,i}^{'h_{2,i}}\right) \cdot e\left(g^{-\delta_1 w_{2,i} + \delta_2 w_{1,i}}, \alpha^{s_1}\right)$$

$$\cdot e\left(g, g\right)^{\left[\lambda_1'\left(\delta_1 w_{2,i} - \delta_2 w_{1,i}\right) s_3 + \lambda_2'\left(\theta_1 t_{2,i} - \theta_2 t_{1,i}\right) s_4\right] x_i v_i} \cdot e\left(g^{-\theta_1 t_{2,i} - \theta_2 t_{1,i}}, \alpha^{s_1}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{\Psi}, g^{r_i' s_1}\right) \cdot e\left(g^{\Psi}, g^{\phi_i' s_1}\right) \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right) \cdot e\left(g, g\right)^{\Psi\left[\lambda_1' s_3 + \lambda_2' s_4\right] \overrightarrow{x} \cdot \overrightarrow{v}}$$

$$\cdot e\left(g^{s_2}, K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}} K_{3,i}^{'h_{1,i}} K_{4,i}^{'h_{2,i}}\right) \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right)$$

$$= e\left(g^{\Psi s_1}, \prod_{i=1}^{n} g^{\left(r_i' + \phi_i'\right)}\right) \cdot e\left(g, g\right)^{\Psi\left[\lambda_1' s_3 + \lambda_2' s_4\right] \overrightarrow{x} \cdot \overrightarrow{v}} \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}} K_{3,i}^{'h_{1,i}} K_{4,i}^{'h_{2,i}}\right)$$

After completing this computation, the algorithm then computes $CT_B$ as:

$$CT_B = e\left(A, K_A'\right) \cdot e\left(B, K_B'\right) \cdot \prod_{i=1}^{n} e\left(C_{1,i}, K_{1,i}'\right) \cdot e\left(C_{2,i}, K_{2,i}'\right) \cdot e\left(C_{3,i}, K_{3,i}'\right) \cdot e\left(C_{4,i}, K_{4,i}'\right)$$

$$= e\left(g^{s_2}, g_2 \prod_{i=1}^{n} K_{1,i}^{'-f_{1,i}} K_{2,i}^{'-f_{2,i}} K_{3,i}^{'-h_{1,i}} K_{4,i}^{'-h_{2,i}}\right) \cdot e\left(g^{\Psi s_1}, \prod_{i=1}^{n} g^{-\left(r_i' + \phi_i'\right)}\right) \cdot e\left(g^{\Psi s_1}, \prod_{i=1}^{n} g^{\left(r_i' + \phi_i'\right)}\right) \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right)$$

$$\cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}} K_{3,i}^{'h_{1,i}} K_{4,i}^{'h_{2,i}}\right) \cdot e\left(g,g\right)^{\Psi\left[\lambda_1' s_3 + \lambda_2' s_4\right]\overrightarrow{x}\cdot\overrightarrow{v}}$$

$$= e\left(g^{s_2}, g_2\right) \cdot e\left(g,g\right)^{\Psi\left[\lambda_1' s_3 + \lambda_2' s_4\right]\overrightarrow{x}\cdot\overrightarrow{v}} \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right)$$

recalling that $A = g^{s_2}$, $B = g^{\Psi s_1}$, with $s_2 = \Psi s_1$.

The re-encrypted ciphertext $CT'$ therefore becomes the tuple $\left(A, B, CT_A, CT_B, D = e\left(g,g_2\right)^{-s_2} M\right)$.

$M \leftarrow$ **Decrypt** $\left(CT, PK_{\overrightarrow{v}}\right)$: On the input of the ciphertext $CT$ and a private key $PK_{\overrightarrow{v}}$, the algorithm begins to decrypt the ciphertext, but based on two conditions.

**Case I**: For a well-formed ciphertext, the algorithm decrypts $CT = \left(A, B, \{C_{1,i}, C_{2,i}\}_{i=1}^{n}, \{C_{3,i}, C_{4,i}\}_{i=1}^{n}, D = e\left(g,g_2\right)^{-s_2} M\right)$ using the private key $PK_{\overrightarrow{v}} = \left(K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n}\right)$ in order to output a message $M$, given by

$$M \leftarrow D \cdot e\left(A, K_A\right) \cdot e\left(B, K_B\right) \cdot \prod_{i=1}^{n} e\left(C_{1,i}, K_{1,i}'\right) \cdot e\left(C_{2,i}, K_{2,i}'\right) \cdot e\left(C_{3,i}, K_{3,i}'\right) \cdot e\left(C_{4,i}, K_{4,i}'\right)$$

**Correctness**: Assume the actual vector $\overrightarrow{x} = (x_1, ..., x_n)$ is used for the formation of the ciphertext $CT$. The message can be recovered as follows: Let $\beta = D \cdot e\left(A, K_A\right) \cdot e\left(B, K_B\right)$ and $\gamma = \prod_{i=1}^{n} e\left(C_{1,i}, K_{1,i}\right) \cdot e\left(C_{2,i}, K_{2,i}\right) \cdot e\left(C_{3,i}, K_{3,i}\right) \cdot e\left(C_{4,i}, K_{4,i}\right)$

Solving for $\gamma$, we have

$$\gamma = \prod_{i=1}^{n} e\left(g^{w_{1,i}s_1} g^{f_{1,i}s_2} g^{\delta_1 x_i s_3}, g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}\right) \cdot e\left(g^{w_{2,i}s_1} g^{f_{2,i}s_2} g^{\delta_2 x_i s_3}, g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\right)$$

$$\cdot e\left(g^{t_{1,i}s_1} g^{h_{1,i}s_2} g^{\theta_1 x_i s_4}, g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}\right) \cdot e\left(g^{t_{2,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{w_{1,i}s_1}, g^{-\delta_2 r_i}\right) \cdot e\left(g^{f_{1,i}s_2}, g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}\right) \cdot e\left(g^{\delta_1 x_i s_3}, g^{\lambda_1 v_i w_{2,i}}\right) \cdot e\left(g^{w_{2,i}s_1}, g^{\delta_1 r_i}\right) \cdot e\left(g^{f_{2,i}s_2}, g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\right)$$

$$\cdot e\left(g^{\delta_2 x_i s_3}, g^{-\lambda_1 v_i w_{1,i}}\right) \cdot e\left(g^{t_{1,i}s_1}, g^{-\theta_2 \phi_i}\right) \cdot e\left(g^{h_{1,i}s_2}, g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}\right) \cdot e\left(g^{\theta_1 x_i s_4}, g^{\lambda_2 v_i t_{2,i}}\right) \cdot e\left(g^{t_{2,i}s_1}, g^{\theta_1 \phi_i}\right)$$

$$\cdot e\left(g^{h_{2,i}s_2}, g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right) \cdot e\left(g^{\theta_2 x_i s_4}, g^{-\lambda_2 v_i t_{1,i}}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{-\delta_2 w_{1,i}}, g^{s_1 r_i}\right) \cdot e\left(g^{s_2}, \left(g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}\right)^{f_{1,i}}\right) \cdot e\left(g,g\right)^{\lambda_1 \delta_1 w_{2,1} \cdot x_i \cdot v_i \cdot s_3} \cdot e\left(g^{\delta_1 w_{2,i}}, g^{s_1 r_i}\right)$$

$$\cdot e\left(g^{s_2}, \left(g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\right)^{f_{2,i}}\right) \cdot e\left(g,g\right)^{-\lambda_1 \delta_2 w_{1,1} \cdot x_i \cdot v_i \cdot s_3} \cdot e\left(g^{-\theta_2 t_{1,i}}, g^{s_1 \phi_i}\right) \cdot e\left(g^{s_2}, \left(g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}\right)^{h_{1,i}}\right) \cdot$$

$$e\left(g,g\right)^{\lambda_2 \theta_1 t_{2,1} \cdot x_i \cdot v_i \cdot s_4} \cdot e\left(g^{\theta_1 t_{2,i}}, g^{s_1 \phi_i}\right) \cdot e\left(g^{s_2}, \left(g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right)^{h_{2,i}}\right) \cdot e\left(g,g\right)^{-\lambda_2 \theta_2 t_{1,1} \cdot x_i \cdot v_i \cdot s_4} \cdot$$

$$= \prod_{i=1}^{n} e\left(g^{\delta_1 w_{2,i} - \delta_2 w_{1,i}}, g^{r_i s_1}\right) \cdot \left(g^{\theta_1 t_{2,i} - \theta_2 t_{1,i}}, g^{\phi_i s_1}\right) \cdot e\left(g^{s_2}, K_{1,i}^{f_{1,i}} K_{2,i}^{f_{2,i}} K_{3,i}^{h_{1,i}} K_{4,i}^{h_{2,i}}\right)$$

$$\cdot e\left(g,g\right)^{\left[\lambda_1\left(\delta_1 w_{2,i} - \delta_2 w_{1,i}\right) s_3 + \lambda_2\left(\theta_1 t_{2,i} - \theta_2 t_{1,i}\right) s_4\right] x_i \cdot v_i}$$

$$= \prod_{i=1}^{n} e\left(g^{\Psi}, g^{r_i s_1}\right) \cdot e\left(g^{\Psi}, g^{\phi_i s_1}\right) \cdot e\left(g^{s_2}, K_{1,i}^{f_{1,i}} K_{2,i}^{f_{2,i}} K_{3,i}^{h_{1,i}} K_{4,i}^{h_{2,i}}\right) \cdot e\left(g,g\right)^{\Psi\left[\lambda_1 s_3 + \lambda_2 s_4\right]\overrightarrow{x}\cdot\overrightarrow{v}}$$

$$= e\left(g^{\Psi s_1}, \prod_{i=1}^{n} g^{(r_i + \phi_i)}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{f_{1,i}} K_{2,i}^{f_{2,i}} K_{3,i}^{h_{1,i}} K_{4,i}^{h_{2,i}}\right) \cdot e\left(g,g\right)^{\Psi\left[\lambda_1 s_3 + \lambda_2 s_4\right]\overrightarrow{x}\cdot\overrightarrow{v}}$$

$$= e\left(g^{s_2}, \prod_{i=1}^{n} g^{(r_i + \phi_i)}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{f_{1,i}} K_{2,i}^{f_{2,i}} K_{3,i}^{h_{1,i}} K_{4,i}^{h_{2,i}}\right) \cdot e\left(g,g\right)^{\Psi\left[\lambda_1 s_3 + \lambda_2 s_4\right]\overrightarrow{x}\cdot\overrightarrow{v}}$$

The message $M$ can then be recovered as,

$$M \leftarrow D \cdot e(A, K_A) \cdot e(B, K_B) \cdot \gamma$$

$$= e\left(g,g_2\right)^{-s_2} M \cdot e\left(g^{s_2}, g_2 \prod_{i=1}^{n} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-h_{1,i}} K_{4,i}^{-h_{2,i}}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} g^{-(r_i + \phi_i)}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} g^{(r_i + \phi_i)}\right)$$

$$\cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{f_{1,i}} K_{2,i}^{f_{2,i}} K_{3,i}^{h_{1,i}} K_{4,i}^{h_{2,i}}\right) \cdot e\left(g,g\right)^{\Psi[\lambda_1 s_3+\lambda_2 s_4]\overrightarrow{x_i}\cdot\overrightarrow{v_i}}$$

$$= e\left(g,g_2\right)^{-s_2} M \cdot e\left(g,g_2\right)^{s_2} \cdot e\left(g,g\right)^{\Psi[\lambda_1 s_3+\lambda_2 s_4]\overrightarrow{x_i}\cdot\overrightarrow{v_i}}$$

$$= M \cdot e\left(g,g\right)^{\Psi[\lambda_1 s_3+\lambda_2 s_4]\overrightarrow{x_i}\cdot\overrightarrow{v_i}}$$

The above result outputs 1 iff $\left(\overrightarrow{x}, \overrightarrow{v}\right) = 0$ in $Z_p$. If it happens that $\left(\overrightarrow{x}, \overrightarrow{v}\right) \neq 0$, then $\lambda_1 s_3 + \lambda_2 s_4 = 0$. The probability of being the identity then becomes $1/p$ since the exponents $\lambda_1$, $\lambda_2$, $s_3$, and $s_4$ are all randomly chosen from $Z_p$.

**Case II**: However, for the re-encrypted version of the ciphertext, $CT = \left(A,\ B,\ CT_A, CT_B, D = e\left(g,g_2\right)^{-s_2} M\right)$, the algorithm first decrypts $CT_A$ by utilizing $PK_{\overrightarrow{v}}$, as shown below to obtain $\alpha$. That is, $\alpha \longleftarrow Decrypt\left(PK_{\overrightarrow{v}}, CT_A\right)$. The deduction and correctness are shown below. We first compute $CT_A$ as follows:

$$CT_A = \prod_{i=1}^{n} \left\{g^{w_{1,i}s_1} g^{f_{1,i}s_2} g^{\delta_1 x_i s_3}, g^{w_{2,i}s_2} g^{f_{2,i}s_2} g^{\delta_2 x_i s_3}\right\} \cdot \left\{g^{-\delta_2 r'_i} g^{\lambda'_i v_i w_{2,i}} \alpha^{\delta_2}, g^{\delta_1 r'_i} g^{-\lambda'_i v_i w_{1,i}} \alpha^{-\delta_1}\right\}$$

$$= \prod_{i=1}^{n} e\left(g^{w_{1,i}s_1}, g^{-\delta_2 r'_i}\right) \cdot e\left(g^{f_{1,i}s_2}, g^{-\delta_2, r'_i} g^{\lambda'_1 v_i w_{2,i}} \alpha^{\delta_2}\right) \cdot e\left(g^{\delta_1 x_i s_3}, g^{\lambda'_1 v_i w_{2,i}}\right) \cdot e\left(g^{w_{1,i}s_1}, \alpha^{s_2}\right)$$

$$\cdot e\left(g^{w_{2,i}s_1}, g^{\delta_1 r'_i}\right) \cdot e\left(g^{f_{2,i}s_2}, g^{\delta_1, r'_i} g^{-\lambda'_1 v_i w_{1,i}} \alpha^{-\delta_1}\right) \cdot e\left(g^{\delta_2 x_i s_3}, g^{-\lambda'_1 v_i w_{1,i}}\right) \cdot e\left(g^{w_{2,i}s_2}, \alpha^{-\delta_1}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{-\delta_2 w_{1,i}}, g^{r'_i s_1}\right) \cdot e\left(g^{s_2}, \left(g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} \alpha^{\delta_2}\right)^{f_{1,i}}\right) \cdot e\left(g,g\right)^{\lambda'_1 \delta_1 w_{2,i} x_i v_i s_3} \cdot e\left(g^{w_{1,i}s_1}, \alpha^{\delta_2}\right)$$

$$\cdot e\left(g^{\delta_1 w_{2,i}}, g^{r'_i s_1}\right) \cdot e\left(g^{s_2}, \left(g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} \alpha^{-\delta_1}\right)^{f_{2,i}}\right) \cdot e\left(g,g\right)^{-\lambda'_1 \delta_2 w_{1,i} x_i v_i s_3} \cdot e\left(g^{w_{2,i}s_2}, \alpha^{-\delta_1}\right)$$

$$= \prod_{i=1}^{n} e\left(g^{\delta_1 w_{2,i} - \delta_2 w_{1.i}}, g^{r'_1 s_1}\right) \cdot e\left(g^{s_2}, K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}}\right) \cdot e\left(g^{-\delta_1 w_{2,i} + \delta_2 w_{1,i}}, \alpha^{s_1}\right) \cdot e\left(g,g\right)^{\lambda'_1 [\delta_1 w_{2,i} - \delta_2 w_{1,i}] x_i \cdot v_i \cdot s_3}$$

$$= \prod_{i=1}^{n} e\left(g^{\Psi}, g^{r'_i s_1}\right) \cdot e\left(g^{s_2}, K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}}\right) \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right) \cdot e\left(g,g\right)^{\Psi \lambda'_1 x_i \cdot v_i \cdot s_3}$$

$$= e\left(g^{\Psi s_1}, \prod_{i=1}^{n} g^{r'_i s_1}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}}\right) \cdot e\left(g^{-\Psi s_1}, \alpha\right) \cdot e\left(g,g\right)^{\Psi \lambda'_1 s_3 \overrightarrow{x}\cdot\overrightarrow{v}}$$

$$= e\left(g^{s_2}, \prod_{i=1}^{n} g^{r'_i}\right) \cdot e\left(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}}\right) \cdot e\left(g^{-s_2}, \alpha\right) \cdot e\left(g,g\right)^{\Psi \lambda'_1 s_3 \overrightarrow{x}\cdot\overrightarrow{v}}$$

Multiplying the result with $PK_{\overrightarrow{v}}$, we have

$$\alpha = \alpha \cdot e\left(g,g_2\right)^{\Psi \lambda'_1 s_3 \left(\overrightarrow{x}\cdot\overrightarrow{v}\right)}$$

Thus, the output of the above is $\alpha$ iff $\overrightarrow{x} \cdot \overrightarrow{v} = 0$. After completing the computation for $\alpha$, we compute the message $M$ as $M \longleftarrow D \cdot CT_B \cdot e\left(g^{\Psi s_1}, \alpha\right)$, and it is shown below.

$$= e\left(g,g_2\right)^{-s_2} M \cdot e\left(g^{s_2}, g_2\right) \cdot e\left(g,g\right)^{\Psi[\lambda'_1 s_3+\lambda'_2 s_4]\left(\overrightarrow{x}\cdot\overrightarrow{v}\right)} \cdot e\left(g^{-\Psi}, \alpha^{s_1}\right) \cdot e\left(g^{\Psi s_1}, \alpha\right)$$

Recalling that $\alpha = g_2^l$, we have

$$= e\left(g,g_2\right)^{-s_2} \cdot M \cdot e\left(g,g_2\right)^{s_2} \cdot e\left(g,g\right)^{\Psi[\lambda'_1 s_3+\lambda'_2 s_4]\left(\overrightarrow{x}\cdot\overrightarrow{v}\right)} \cdot e\left(g,g_2\right)^{-\Psi l s_1} \cdot e\left(g,g_2\right)^{\Psi l s_1}$$

$$= M \cdot e(g,g)^{\Psi[\lambda_1' s_3 + \lambda_2' s_4](\overrightarrow{x} \cdot \overrightarrow{v})}$$

The above result outputs 1 iff $(\overrightarrow{x}, \overrightarrow{v}) = 0$ in $Z_p$. If it happens that $(\overrightarrow{x}, \overrightarrow{v}) \neq 0$, then $\lambda_1' s_3 + \lambda_2' s_4 = 0$. The probability of being the identity then becomes $1/p$ since the exponents are all randomly chosen from $Z_p$.

## 6. Security Model

Following the approach in [25], we prove that our scheme exhibits attribute-hiding property. The adversary, $\mathcal{A}$, and the challenger, $\mathcal{C}$, are engaged in a series of games in our security model. Both $\mathcal{A}$ and $\mathcal{C}$ are, by assumption, given the attribute set $\sum$, and the predicate class $\mathcal{F}$ beforehand. The security game is played over the vectors of the re-encryption process.

**Initialize**: The adversary, $\mathcal{A}$, outputs two vectors $\overrightarrow{x'}, \overrightarrow{y'} \in \sum$.

**Setup**: The challenger, $\mathcal{C}$, runs *Setup* to obtain the public key $P_{pub}$ and the secret key $SK$, after which $\mathcal{A}$ is given $P_{pub}$.

**Query Phase 1**: $\mathcal{A}$ adaptively issues private key queries for the vector $\overrightarrow{v} = \{v_i, ..., v_n\} \in \sum$ subject to the restriction that, $\forall i$, $\langle \overrightarrow{v_i}, \overrightarrow{x'} \rangle = 0$ iff $\langle \overrightarrow{v_i}, \overrightarrow{y'} \rangle = 0$. $\mathcal{C}$ responds with $PK_{\overrightarrow{v}} \leftarrow \mathbf{KGen}(SK, \overrightarrow{v_i})$

**Challenge**: Two messages $M_0, M_1 \in \mathcal{M}$ are output by $\mathcal{A}$. If $M_0 \neq M_1$, it is a requirement that $\langle \overrightarrow{v_i}, \overrightarrow{x'} \rangle \neq \langle \overrightarrow{v_i}, \overrightarrow{y'} \rangle \neq 0$ for all queries made on the vector $\overrightarrow{v}$. $\mathcal{C}$ picks a random bit, $b \in \{0,1\}$. If $b = 0$, $\mathcal{C}$ gives $CT' \leftarrow \mathbf{Encrypt}(P_{pub}, \overrightarrow{x'}, M_0)$ to $\mathcal{A}$, and if $b = 1$, $CT' \leftarrow \mathbf{Encrypt}(P_{pub}, \overrightarrow{y'}, M_1)$ is given to $\mathcal{A}$.

**Query Phase 2**: Additional private key queries are made by $\mathcal{A}$ for additional vectors, subject to the same restrictions as stated above.

**Guess**: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$, and wins the game if $b = b'$.

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}(\mathcal{A}) = |Pr[b = b'] - \frac{1}{2}|$. For the scenario where the two messages are not the same, i.e. $M_0 \neq M_1$, $\mathcal{A}$ is not permitted to issue private key queries for vectors $\overrightarrow{v_i}$ such that $\langle \overrightarrow{v_i}, \overrightarrow{x'} \rangle = \langle \overrightarrow{v_i}, \overrightarrow{y'} \rangle = 0$. This is done throughout all the query phases. If that is not the case, for a vector $\overrightarrow{v_i}$, the adversary can obtain a private key $PK_{\overrightarrow{v_i}}$ and decrypt the challenge ciphertext using the private key corresponding to that vector. The restriction is however not required for the case where $M_0 = M_1$.

*Security Proof*

In proving the security of our scheme, we introduce a series of security games between the adversary and the challenger as stated above. We also consider the case where there is a distinction between the two messages. As stated in the security model, the adversary is not in any way permitted to make private key queries for the vector $\overrightarrow{v}$ such that $\langle \overrightarrow{v_i}, \overrightarrow{x'} \rangle = \langle \overrightarrow{v_i}, \overrightarrow{y'} \rangle = 0$.

$Game_1$ : The challenge ciphertext is generated under $(\overrightarrow{x'}, \overrightarrow{x'})$ and $M_0$, and it is computed as

$$CT_1 = \left( g^{s_2}, g_1^{s_1}, \left\{ W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{x_i' s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{x_i' s_3} \right\}_{i=1}^{n}, \left\{ T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{x_i' s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{x_i' s_4} \right\}_{i=1}^{n}, \Upsilon^{-s_2} M_0 \right)$$

$Game_2$ : $(\overrightarrow{x'}, \overrightarrow{x'})$ and a random message $R_x \in G_T$ are used to generate the challenge ciphertext, and it is computed as

$$CT_2 = \left( g^{s_2}, g_1^{s_1}, \left\{ W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{x_i' s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{x_i' s_3} \right\}_{i=1}^{n}, \left\{ T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{x_i' s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{x_i' s_4} \right\}_{i=1}^{n}, R_x \right)$$

$Game_3$ : The challenge ciphertext is generated under $(\overrightarrow{x'}, \overrightarrow{0})$ and a random message $R_x \in G_T$, and it is computed as

$$CT_3 = \left(g^{s_2}, g_1^{s_1}, \left\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{x'_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{x'_i s_3}\right\}_{i=1}^{n}, \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2}\right\}_{i=1}^{n}, R_x\right)$$

$Game_4$ : The challenge ciphertext is generated under $(\overrightarrow{x'}, \overrightarrow{y'})$ and a random message $R_x \in G_T$, and it is computed as

$$CT_4 = \left(g^{s_2}, g_1^{s_1}, \left\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{x'_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{x'_i s_3}\right\}_{i=1}^{n}, \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{y'_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{y'_i s_4}\right\}_{i=1}^{n}, R_x\right)$$

$Game_5$ : The challenge ciphertext is generated under $(\overrightarrow{0}, \overrightarrow{y'})$ and a random message $R_x \in G_T$, and it is computed as

$$CT_5 = \left(g^{s_2}, g_1^{s_1}, \left\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2}\right\}_{i=1}^{n}, \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{y'_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{y'_i s_4}\right\}_{i=1}^{n}, R_x\right)$$

$Game_6$ : The challenge ciphertext is generated under $(\overrightarrow{y'}, \overrightarrow{y'})$ and a random message $R_x \in G_T$, and it is computed as

$$CT_6 = \left(g^{s_2}, g_1^{s_1}, \left\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{y'_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{y'_i s_3}\right\}_{i=1}^{n}, \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{y'_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{y'_i s_4}\right\}_{i=1}^{n}, R_x\right)$$

$Game_7$ : $(\overrightarrow{y'}, \overrightarrow{y'})$ and $M_1$ is used to generate the challenge ciphertext, and it is computed as

$$CT_7 = \left(g^{s_2}, g_1^{s_1}, \left\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot Q_1^{y'_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot Q_2^{y'_i s_3}\right\}_{i=1}^{n}, \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot R_1^{y'_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot R_2^{y'_i s_4}\right\}_{i=1}^{n}, \mathrm{Y}^{-s_2} M_1\right)$$

We prove that $Game_1$ and $Game_7$ are indistinguishable to an adversary with polynomial time. This is achieved by proving the computational indistinguishability of the transitions between the games. This is because the indistinguishability between $Game_1$ and $Game_2$ also indicates that $Game_6$ and $Game_7$ are also indistinguishable, by the property of symmetry of the hybrid games [25].

Under the $(t, \epsilon)$ Decision Bilinear Diffie–Hellman assumption, $Game_1$ and $Game_2$ cannot be distinguished by an adversary running in polynomial time $t$ with an advantage greater than $\epsilon$, assuming there is an adversary $\mathcal{A}$ with non-negligible advantage $\epsilon$ that can attack the scheme. We describe the game between the challenger and the adversary as follows. On input $(g, g^a, g^b, g^c, Z) \in G^4 \times G_T$, the goal of the challenger is to output 1 if $Z = g^{abc}$, and 0 otherwise. The challenger and the adversary engage in the following interaction:

**Public parameters:** The challenger chooses random exponents $\{\delta_i, \theta_i\}_{i=1}^{2}$, $\{w_{1,i}, t_{1,i}\}_{i=1}^{n}$, $\{f_{1,i}, f_{2,i}\}_{i=1}^{n}$, $\{h_{1,i}, h_{2,i}\}_{i=1}^{n}$, $and \omega \in Z_p$. A random $\Psi \in Z_p$ is also selected to obtain $\{w_{2,i}, t_{2,i}\}_{i=1}^{n}$ under the constraints

$$\Psi = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \Psi = \theta_1 t_{2,i} - \theta_2 t_{1,i}$$

If $\Psi = 0$, the challenger selects a new set of random exponents. It then sets the following conditions

$$W_{1,i} = g^{w_{1,i}}, W_{2,i} = g^{w_{2,i}}, T_{1,i} = g^{t_{1,i}}, F_{1,i} = g^{\hat{f_{1,i}}}, F_{2,i} = g^{\hat{f_{2,i}}}, H_{1,i} = g^{\hat{h_{1,i}}}, H_{2,i} = g^{\hat{h_{2,i}}} T_{2,i} = g^{t_{2,i}}$$

where

$$\hat{f_{1,i}} = x'_i \delta_1 b + f_{1,i}, \hat{f_{2,i}} = x'_i \delta_2 b + f_{2,i}, \hat{h_{1,i}} = x'_i \theta_1 b + h_{1,i}, \hat{h_{2,i}} = x'_i \theta_2 b + h_{2,i}$$

$\forall i = 1, ..., n$ and $g_2 = g^{-\Psi ab} g^{\omega}$.

The challenger then initiates the following notations:

$$Q_1 = g^{\delta_1}, Q_2 = g^{\delta_2}, R_1 = g^{\theta_1}, R_2 = g^{\theta_2}, g_1 = g^{\Psi}, \Upsilon = e(g^a, g^b)^{-\Psi} \cdot e(g,g)^{\omega}$$

**Key Derivation:** $\mathcal{A}$ issues private key queries for the vectors. Considering making queries for the vector $\overrightarrow{v} = (v_1, ..., v_n) \in Z_p$, $\mathcal{A}$ can request for private key queries as long as $\langle \overrightarrow{v}, \overrightarrow{x'} \rangle = \varrho \neq 0$. The challenger selects random exponents $\lambda'_1, \lambda'_2, \{r'_i, \phi'_i\}_{i=1}^{n} \in Z_p$ in generating the re-encrypted key $REKey_{\overrightarrow{v}}$, and sets

$$\hat{\lambda'_1} = \mu a + \lambda'_1, \hat{\lambda'_2} = \mu a + \lambda'_2$$

where $\mu = \frac{1}{2\varrho}$. The re-encrypted keys $K'_{1,i}, K'_{2,i}, K'_{3,i}, K'_{4,i}$ are then generated as follows:

$$K'_{1,i} = (g^a)^{v_i w_{2,i} \mu} g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} \alpha^{\delta_2}, K'_{2,i} = (g^a)^{-v_i w_{1,i} \mu} g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} \alpha^{-\delta_1},$$

$$K'_{3,i} = (g^a)^{v_i t_{2,i} \mu} g^{-\theta_1 \phi'_i} g^{\lambda'_2 v_i t_{2,i}} \alpha^{\theta_2}, K'_{4,i} = (g^a)^{-v_i t_{1,i} \mu} g^{\theta_1 \phi'_i} g^{-\lambda'_2 v_i t_{1,i}} \alpha^{-\theta_1},$$

$\forall i = 1, ..., n$. The $K'_A$ and $K'_B$ elements are, respectively, computed as $K'_A = g_2 \prod_{i=1}^{n} K_{1,i}^{'-\hat{f_{1,i}}} K_{2,i}^{'-\hat{f}_{2,i}} K_{3,i}^{'-\hat{h_{1,i}}} K_{4,i}^{'-\hat{h}_{2,i}}$ and $K'_B = \prod_{i=1}^{n} g^{-(r'_i + \phi'_i)}$.

Let $\mathcal{X} = K_{1,i}^{'-\hat{f_{1,i}}} K_{2,i}^{'-\hat{f}_{2,i}}$ and $\mathcal{Y} = K_{3,i}^{'-\hat{h_{1,i}}} K_{4,i}^{'-\hat{h}_{2,i}}$. Computing for both $\mathcal{X}$ and $\mathcal{Y}$ yields

$$\mathcal{X} = \left[ (g^a)^{v_i w_{2,i} \mu} g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} \alpha^{\delta_2} \right]^{-\left(x'_i \delta_1 b + f_{1,i}\right)} \cdot \left[ (g^a)^{-v_i w_{1,i} \mu} g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} \alpha^{-\delta_1} \right]^{-\left(x'_i \delta_2 b + f_{2,i}\right)}$$

$$= \left(g^{ab}\right)^{-v_i x'_i \delta_1 w_{2,i} \mu} (g^a)^{-v_i w_{2,i} f_{1,i} \mu} \left(g^b\right)^{x'_i \delta_1 \delta_2 r'_i} g^{r'_i \delta_2 f_{1,i}} \left(g^b\right)^{-\lambda'_1 v_i x'_i \delta_1 w_{2,i}} g^{-\lambda'_1 v_i w_{2,i} f_{1,i}} \left(\alpha^b\right)^{-x'_i \delta_1 \delta_2} \alpha^{-\delta_2 f_{1,i}}$$

$$\cdot \left(g^{ab}\right)^{v_i x'_i \delta_2 w_{1,i} \mu} (g^a)^{v_i w_{1,i} f_{2,i} \mu} \left(g^b\right)^{-x'_i \delta_1 \delta_2 r'_i} g^{-r'_i \delta_1 f_{2,i}} \left(g^b\right)^{\lambda'_1 v_i x'_i \delta_2 w_{1,i}} g^{\lambda'_1 v_i w_{1,i} f_{2,i}} \left(\alpha^b\right)^{x'_i \delta_1 \delta_2} \alpha^{\delta_1 f_{2,i}}$$

$$= \left(g^{ab}\right)^{v_i x'_i \left[\delta_2 w_{1,i} - \delta_1 w_{2,i}\right] \mu} (g^a)^{v_i \left[w_{1,i} f_{2,i} - w_{2,i} f_{1,i}\right] \mu} g^{r'_i \left[\delta_2 f_{1,i} - \delta_1 f_{2,i}\right]} \left(g^b\right)^{\lambda'_1 v_i x'_i \left[\delta_2 w_{1,i} - \delta_1 w_{2,i}\right]} g^{\lambda'_1 v_i \left[w_{1,i} f_{2,i} - w_{2,i} f_{1,i}\right]}$$

$$\alpha^{\left[\delta_1 f_{2,i} - \delta_2 f_{1,i}\right]}$$

$$= \left(g^{ab}\right)^{-\Psi v_i x'_i \mu} (g^a)^{\chi v_i \mu} g^{r'_i \vartheta} \left(g^b\right)^{-\Psi \lambda'_1 v_i x'_i} g^{\chi \lambda'_1 v_i} \alpha^{-\vartheta}$$

where $\Psi = \delta_1 w_{2,i} - \delta_2 w_{1,i}$, $\chi = w_{1,i} f_{2,i} - w_{2,i} f_{1,i}$ and $\vartheta = \delta_2 f_{1,i} - \delta_1 f_{2,i}$.

$$\mathcal{Y} = \left[ (g^a)^{v_i t_{2,i} \mu} g^{-\theta_2 \phi'_i} g^{\lambda'_2 v_i t_{2,i}} \alpha^{\theta_2} \right]^{-\left(x'_i \theta_1 b + h_{1,i}\right)} \cdot \left[ (g^a)^{-v_i t_{1,i} \mu} g^{\theta_1 \phi'_i} g^{-\lambda'_2 v_i t_{1,i}} \alpha^{-\theta_1} \right]^{-\left(x'_i \theta_2 b + h_{2,i}\right)}$$

$$= \left(g^{ab}\right)^{-v_i x'_i \theta_1 t_{2,i} \mu} (g^a)^{-v_i t_{2,i} h_{1,i} \mu} \left(g^b\right)^{x'_i \theta_1 \theta_2 \phi'_i} g^{\phi'_i \theta_2 h_{1,i}} \left(g^b\right)^{-\lambda'_2 v_i x'_i \theta_1 t_{2,i}} g^{-\lambda'_2 v_i t_{2,i} h_{1,i}} \left(\alpha^b\right)^{-x'_i \theta_1 \theta_2} \alpha^{-\theta_2 h_{1,i}}$$

$$\cdot \left(g^{ab}\right)^{v_i x'_i \theta_2 t_{1,i} \mu} (g^a)^{v_i t_{1,i} h_{2,i} \mu} \left(g^b\right)^{-x'_i \theta_1 \theta_2 \phi'_i} g^{-\phi'_i \theta_1 h_{2,i}} \left(g^b\right)^{\lambda'_2 v_i x'_i \theta_2 t_{1,i}} g^{\lambda'_2 v_i t_{1,i} h_{2,i}} \left(\alpha^b\right)^{x'_i \theta_1 \theta_2} \alpha^{\theta_1 h_{2,i}}$$

$$= \left(g^{ab}\right)^{v_i x'_i \left[\theta_2 t_{1,i} - \theta_1 t_{2,i}\right] \mu} (g^a)^{v_i \left[t_{1,i} h_{2,i} - t_{2,i} h_{1,i}\right] \mu} g^{\phi'_i \left[\theta_2 h_{1,i} - \theta_1 h_{2,i}\right]} \left(g^b\right)^{\lambda'_2 v_i x'_i \left[\theta_2 t_{1,i} - \theta_1 t_{2,i}\right]} g^{\lambda'_2 v_i \left[t_{1,i} h_{2,i} - t_{2,i} h_{1,i}\right]}$$

$$\alpha^{\left[\theta_1 h_{2,i} - \theta_2 h_{1,i}\right]}$$

$$= \left(g^{ab}\right)^{-\Psi v_i x'_i \mu} (g^a)^{\zeta v_i \mu} g^{\phi'_i \xi} \left(g^b\right)^{-\Psi \lambda'_2 v_i x'_i} g^{\zeta \lambda'_2 v_i} \alpha^{-\xi}$$

where $\Psi = \theta_1 t_{2,i} - \theta_2 t_{1,i}$, $\zeta = t_{1,i} h_{2,i} - t_{2,i} h_{1,i}$ and $\xi = \theta_2 h_{1,i} - \theta_1 h_{2,i}$.

$\mathcal{X} \cdot \mathcal{Y}$ results in

$$\mathcal{X} \cdot \mathcal{Y} = \left(g^{ab}\right)^{-2\Psi v_i x'_i \mu} (g^a)^{v_i [\chi + \zeta] \mu} g^{r'_i \vartheta + \phi'_i \xi} \left(g^b\right)^{-\Psi v_i x'_i \left[\lambda'_1 + \lambda'_2\right]} g^{v_i \left[\chi \lambda'_1 + \zeta \lambda'_2\right]} \alpha^{-(\vartheta + \xi)}$$

The challenger can then compute $K'_A$ as

$$K'_A = g_2 \prod_{i=1}^{n} \left(g^{ab}\right)^{-2\Psi v_i x'_i \mu} \left(g^a\right)^{v_i[\chi+\zeta]\mu} \left(g^b\right)^{-\Psi v_i x'_i\left[\lambda'_1+\lambda'_2\right]} \cdot g^{r'_i\vartheta+\phi'_i\xi+v_i\left[\chi\lambda'_1+\zeta\lambda'_2\right]}\alpha^{-(\vartheta+\xi)}$$

$$= g^{\omega} \prod_{i=1}^{n} \left(g^a\right)^{v_i[\chi+\zeta]\mu} \cdot \prod_{i=1}^{n} \left(g^b\right)^{-\Psi v_i x'_i\left[\lambda'_1+\lambda'_2\right]} \cdot \prod_{i=1}^{n} g^{r'_i\vartheta+\phi'_i\xi+v_i\left[\chi\lambda'_1+\zeta\lambda'_2\right]} \cdot \alpha^{-(\vartheta+\xi)}$$

The challenger issues the private key $PK_{\overrightarrow{v'}} = \left(K'_A, K'_B, \left\{K'_{1,i}, K'_{2,i}\right\}_{i=1}^{n}, \left\{K'_{3,i}, K'_{4,i}\right\}_{i=1}^{n}\right)$ for the queried vector.

**Challenge Ciphertext:** In generating the challenge ciphertext, the challenger selects random elements $s_1, s_3, s_4 \in Z_p$, and sets

$$\hat{s_1} = s_1, \hat{s_2} = c, \hat{s_3} = s_3 - bc, \hat{s_4} = s_4 - bc.$$

The challenger then computes $A = g^{s_2} = g^c$ and $B = g^{\Psi s_1} = \left(g^{\Psi}\right)^{s_1} = g_1^{s_1}$, and $\forall i = 1, ..., n$, the ciphertexts $C_{1,i}, C_{2,i}, C_{3,i}, C_{4,i}$ are computed as follows

$$C_{1,i} = g^{w_{1,i}s_1} \cdot \left(g^c\right)^{\hat{f_{1,i}}} \cdot g^{\delta_1 x'_i s_3} = \left(g^{w_{1,i}}\right)^{s_1} \cdot \left(\left(g^b\right)^{x'_i\delta_1} g^{f_{1,i}}\right)^{c} \cdot \left(g^{\delta_1}\right)^{x'_i(s_3-bc)} = W_{1,i}^{\hat{s_1}} \cdot F_{1,i}^{\hat{s_2}} \cdot Q_1^{x'_i\hat{s_3}}$$

$$C_{2,i} = g^{w_{2,i}s_1} \cdot \left(g^c\right)^{\hat{f_{2,i}}} \cdot g^{\delta_2 x'_i s_3} = \left(g^{w_{2,i}}\right)^{s_1} \cdot \left(\left(g^b\right)^{x'_i\delta_2} g^{f_{2,i}}\right)^{c} \cdot \left(g^{\delta_2}\right)^{x'_i(s_3-bc)} = W_{2,i}^{\hat{s_1}} \cdot F_{2,i}^{\hat{s_2}} \cdot Q_2^{x'_i\hat{s_3}}$$

$$C_{3,i} = g^{t_{1,i}s_1} \cdot \left(g^c\right)^{\hat{h_{1,i}}} \cdot g^{\theta_1 x'_i s_4} = \left(g^{t_{1,i}}\right)^{s_1} \cdot \left(\left(g^b\right)^{x'_i\theta_1} g^{h_{1,i}}\right)^{c} \cdot \left(g^{\theta_1}\right)^{x'_i(s_4-bc)} = T_{1,i}^{\hat{s_1}} \cdot H_{1,i}^{\hat{s_2}} \cdot R_1^{x'_i\hat{s_4}}$$

$$C_{4,i} = g^{t_{2,i}s_1} \cdot \left(g^c\right)^{\hat{h_{2,i}}} \cdot g^{\theta_2 x'_i s_4} = \left(g^{t_{2,i}}\right)^{s_1} \cdot \left(\left(g^b\right)^{x'_i\theta_2} g^{h_{2,i}}\right)^{c} \cdot \left(g^{\theta_2}\right)^{x'_i(s_4-bc)} = T_{2,i}^{\hat{s_1}} \cdot H_{2,i}^{\hat{s_2}} \cdot R_2^{x'_i\hat{s_4}}$$

The challenger then computes $D = Z^{-\Psi} \cdot e\left(g, g^c\right)^{\omega} \cdot M_0$.

Under the Decisional BDH assumption, $Game_1$ and $Game_2$ are indistinguishable since, if $Z = e\left(g, g\right)^{abc}$, the challenge ciphertext is as given in $Game_1$, while, if $Z$ is a randomly chosen element in $G_T$, then the challenge ciphertext is as shown in $Game_2$.

## 7. Implementation and Performance Analysis

In this section, we provide details of the implementation of our system and also evaluate the performance of our system. Experiments were designed and some useful parameters were measured. In our system, users (data owners inclusive) are registered on the blockchain network and this involves aggregating information pertaining to a specific user. Users are categorized as specified by the data owner. Each user is then given a public and private key pair, which are associated with their details, and to be used in requesting and accessing data.

We implemented the blockchain system on a private Ethereum blockchain network. Ethereum is a programmable blockchain platform that utilizes the robust nature of Solidity (a state-based scripting language). An application was designed in Python that connects each data owner and performs the proxy re-encryption scheme on the data. This application synchronizes with the blockchain using the JSON-RPC (JavaScript Object Notation—Remote Procedure Calls) library. With the blockchain notified about data request, queries are sent to the cloud server and data are filtered and sent to the blockchain. Re-encryption is either performed or not, based on the user type.

### *7.1. Experiment 1*

In this first experiment, we measured the time it takes to register a user (both data owner and data user) on the blockchain network. To register, the user sends its details to the blockchain and

membership keys are given to the user. We measured the delay it takes in mining this transaction. Variations over 40 runs of this scenario were simulated and the average registration delay was obtained. Experiment results indicate an average delay of 13.94 s, which is not far off the 13 s for a block generation in Ethereum networks. The experiment result is shown in Figure 2.

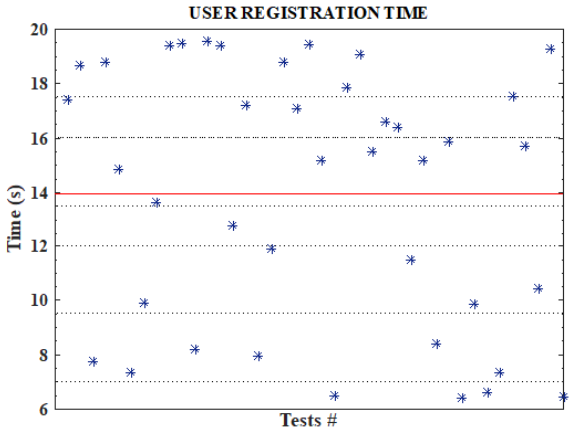


**Figure 2.** User registration delay.

### *7.2. Experiment 2*

In this second experiment, the impact of proxy re-encryption was measured. A flow chart, as shown in Figure 3, was designed that describes data processing as the data are requested by a user. As soon as data request is made, the blockchain network checks if the user is a legitimate member of the network. If successful, it sends a notification to the cloud server, which then filters and retrieves the data before sending them back to the blockchain network.

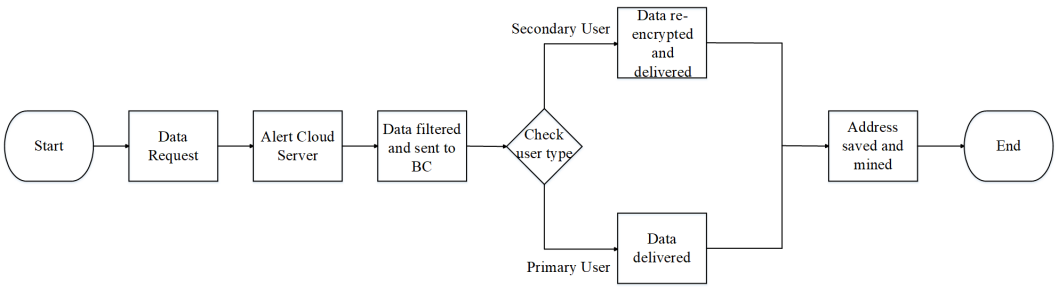


**Figure 3.** Flow chart.

After receiving the data, the blockchain checks for the user type. For a primary user, the blockchain delivers the data and proceeds to mine the address and this becomes a transaction. For a secondary user, the proxy is called upon and it re-encrypts the data before giving it out, after which it is also mined. Experiment results are shown in Figure 4.

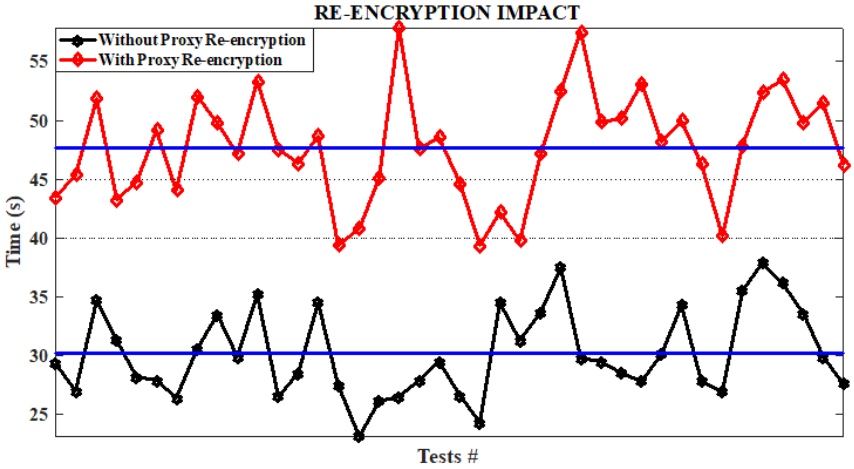


**Figure 4.** Impact of proxy re-encryption.

The tests were run for a variation of 40 times and it was realized that it takes an average of 30.18 s for an end-to-end data processing without re-encryption (as described in the flow chart) to be completed. Similarly, an average of 47.73 s was recorded for a process that involves re-encryption. Consequently, we realized the addition of re-encryption to the scheme increased the delay by 58.15%.

## 8. Discussion

1. **Collusion Resistance:** Our proposed scheme prevents collusion attack in the sense that the re-encrypted data are divided into two parts with one part stored on the blockchain network, and the other part stored on the cloud. Because the blockchain network and the cloud server work in tandem, a data user has to first obtain the bit-part data stored on the blockchain before obtaining the other half from the cloud. As a first level security check (usually performed before decryption), a data user must prove to the blockchain networks' verification unit its membership before gaining access to the data. A revoked user is deprived of this right because its membership keys have been completely removed from the network and therefore the user becomes unknown to the network. However, for a revoked user who still colludes with the cloud server for access to data, the cloud server still has to provide the user's details to the blockchain processing node for the necessary checks to be made. With collusion attack prevented, the confidentiality of the data is preserved/guaranteed.
2. **Fine-grained access control:** There is an effective management of user access by the implementation of the ABE scheme. The utilization of the inner product encryption scheme enables a fine-grained access control to data. The data owner specifies which attribute set or right a data user enjoys and therefore, to access data, there should be a match-up between the attribute set and the private key set. There is also the possibility of selective delegation due to the weight (information type) set by the data owner. Furthermore, depending on the level of trust between the data owner and the user(s), decryption of either all or some data can be delegated selectively to the user(s).

## 9. Conclusions

In this paper, an inner-product proxy re-encryption scheme that ensures an efficient and secured data access to IoT data is presented. The encryption of IoT data is done according to a given access policy and shared with the various data users, and therefore the problem of data sharing has been addressed. We incorporated a blockchain network, whose processing node acts as the proxy server. A user can access data when it is a registered member of the network, with the verification performed by the blockchain network. The proxy also re-encrypts the data by transforming the policy set in the process of sharing the data. The blockchain network works in tandem with the cloud server

to ensure a collusion-resistant scheme. Our approach also achieves a fine-grained access control to data. Experiment results show that proxy re-encryption increased the delay, but the utilization of a blockchain kept a record of all interactions between entities and eliminated the need of a trusted third party. Making improvements to our scheme, in terms of its efficiency, is the focus of our future work. We also plan to include a detailed smart contract algorithm and more experimental results in the next work.

**Author Contributions:** Conceptualization, K.O.-B.O.A. and Q.X.; software design, E.B.S. and J.G.; security analysis, H.X. and X.D.; and writing—review and editing, M.G.

**Funding:** This work was supported in part by the programs of International Science and Technology Cooperation and Exchange of Sichuan Province under Grant 2017HH0028, Grant 2018HH0102 and Grant 2019YFH0014.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviationsr are used in this manuscript:

| | |
|---|---|
| ABE | Attribute Based Encryption |
| IoT | Internet of Things |
| IPE | Inner Product Encryption |
| PRE | Proxy Re-Encryption |
| ECC | Elliptic Curve Cryptography |
| WSN | Wireless Sensor Network |
| HIPAA | Health Insurance Portability and Accountability Act |
| EMR | Electronic Medical Record |
| EHR | Electronic Health Records |

## References

1. Zheng, J.; Simplot-Ryl, D.; Bisdikian, C.; Mouftah, H. The internet of things. *IEEE Commun. Mag.* **2011**, *49*, 30–31. [CrossRef]
2. Miorandi, D.; Sicari, S.; De Pellegrini, F.; Chlamtac, I. Internet of things: Vision applications and research challenges. *Ad Hoc Netw.* **2012**, *10*, 1497–1516. [CrossRef]
3. Casadei, R.; Fortino, G.; Pianini, D.; Russo, W.; Savaglio, C.; Viroli, M. Modelling and simulation of opportunistic IoT services with aggregate computing. *Future Gener. Comput. Syst.* **2019**, *91*, 252–262. [CrossRef]
4. Bennett, T.R.; Savaglio, C.; Lu, D.; Massey, H.; Wang, X.; Wu, J.; Jafari, R. MotionSynthesis toolset (MoST): A toolset for human motion data synthesis and validation. In Proceedings of the 4th ACM MobiHoc Workshop on Pervasive Wireless Healthcare, Philadelphia, PA, USA, 11–14 August 2014. [CrossRef]
5. Leavitt, N. Is cloud computing really ready for prime time? *Computer* **2009**, *42*, 15–20. [CrossRef]
6. Brodkin, J. Gartner: Seven cloud-computing security risks. *Netw. World* **2008**, *2008*, 1–3.
7. Chow, R.; Golle, P.; Jakobsson, M.; Shi, E.; Staddon, J.; Masuoka, R.; Molina, J. Controlling data in the cloud: outsourcing computation without outsourcing control. In Proceedings of the IEEE 3rd International Conference on Cloud Computing, Chicago, IL, USA, 13–13 November 2009; pp. 85–90.
8. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany; Volume 3494, pp. 457–473.
9. Wang, G.; Liu, Q.; Wu, J. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In Proceedings of the ACM conference on Computer and Communications Security CCS'10, Chicago, IL, USA, 4–8 October 2010.
10. Zhao, G.; Rong, C.; Li, J.; Zhang, F.; Tang, Y. Trusted data sharing over untrusted cloud storage providers. In Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Sciene, Indianapolis, IN, USA, 30 Novemebr–3 December 2011; pp. 96–103.

11. Yu, S.; Wang, C.; Ren, K.; Lou, W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the IEEE International Conference on Computer Communications INFOCOM'10, San Diego, CA, USA, 14–19 March 2010.
12. Park, J.H. Inner-product encryption under standard assumptions. *Des. Codes Cryptogr.* **2011**, *58*, 235–257. [CrossRef]
13. Qin, Z.; Xiong, H.; Wu, S.; Batamuliza, J. A survey of proxy re-encryption for secure data sharing in cloud computing. *IEEE Trans. Serv. Comput.* **2017**, *99*. [CrossRef]
14. Sepehri, M.; Cimato, S.; Damiani, E. Efficient implementation of a proxy-based protocol for data sharing on the cloud. In Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing, SCC@AsiaCCS 2017, Abu Dhabi, UAE, 2 April 2017; pp. 67–74.
15. Sepehri, M.; Cimato, S.; Damiani, E.; Yeuny, C.Y. Data Ssharing on the cloud: A scalable proxy-based protocol for privacy-preserving queries. In Proceedings of the 7th IEEE International Symposium on Ubisafe Computing in Conjunction with 14th IEEE Conference on Trust, Security and Privacy in Computing and Communications, TrustCom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; pp. 357–1362.
16. Mambo, M.; Okamoto, E. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. *IICE Trans. Fundam. Electr. Commun. Comput. Sci.* **1997**, *80A*, 54–63.
17. Blaze, M.; Bleumer, G.; Strauss, M. *Divertible Protocols and Atomic Proxy Cryptography*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 127–144.
18. El Gamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. In Proceedings of the CRYPTO 84 on Advances in Cryptology, Santa Barbara, CA, USA, 18–22 August 1985; Springer: New York, NY, USA, 1985; pp. 10–18.
19. Do, J.-M.; Song, Y.-J.; Park, N. Attribute based proxy re-encryption for data confidentiality in cloud computing environments. In Proceedings of the 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering (CNSI '11), Jeju Island, Korea, 23–25 May 2001; IEEE Computer Society: Washington, DC, USA, 2001; pp. 248–251.
20. Guo, S.; Zeng, Y.; Wei, J.; Xu, Q. Attribute-based re-encryption scheme in the standard model. *Wuhan Univ. J. Nat. Sci.* **2008**, *13*, 621–625. [CrossRef]
21. Liang, X.; Cao, Z.; Lin, H.; Shao, J. Attribute based proxy re-encryption with delegating capabilities. In Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS'09), Sydney, Australia, 10–12 March 2009; ACM: New York, NY, USA, 2009; pp. 276–286.
22. Luo, S.; Hu, J.; Chen, Z. Ciphertext policy attribute-based proxy re-encryption. In Proceedings of the 12th International Conference on Information and Communications Security (ICICS'10), Barcelona, Spain, 15–17 December 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 401–415.
23. 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; IEEE Computer Society: Washington, DC, USA, 2007; pp. 321–334.
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 (CCS'06), Alexandria, VA, USA, 30 October–3 November 2006; ACM: New York, NY, USA, 2006; pp. 89–98.
25. Katz, J.; Sahai, A.; Waters, B. *Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 146–162.
26. Okamoto, T.; Takashima, K. *Hierarchical Predicate Encryption for Inner-Products*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 214–231.
27. Du, X.; Guizani, M.; Xiao, Y.; Chen, H.H. A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1223–1229. [CrossRef]
28. Xiao, Y.; Rayib, V.K.; Sunc, B.; Du, X.; Hue, F.; Gallowaya, M. A survey of key management schemes in wireless sensor networks. *J. Comput. Commun.* **2007**, *30*, 2314–2341. [CrossRef]
29. Du, X.; Xiao, Y.; Guizani, M.; Chen, H.H. An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Netw.* **2007**, *5*, 24–34. [CrossRef]
30. Du, X.; Chen, H.H. Security in wireless sensor networks. *IEEE Wirel. Commun. Mag.* **2008**, *15*, 60–66.

31. Griggs, K.N.; Ossipova, O.; Kohlios, C.P.; Baccarini, A.N.; Howson, E.A.; Hayajneh, T. Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring. *J. Med. Syst.* **2018**, *42*, 130. [CrossRef] [PubMed]
32. Kamau, G.; Boore, C.; Maina, E.; Njenga, S. Blockchain technology: Is this the solution to EMR interoperability and security issues in developing countries? In Proceedings of the 2018 IST-Africa Week Conference (IST-Africa), Gaborone, Botswana, 9–11 May 2018; pp. 1–8.
33. Ekblaw, A.; Azaria, A.; Halamka, J.D.; Lippman, A. A case study for blockchain in healthcare: "Medrec" prototype for electronic health records and medical research data. In Proceedings of the IEEE Open and Big Data Conference, Vienna, Austria, 22–24 August 2016; p. 13.
34. Xia, Q.; Sifah, E.B.; Smahi, A.; Amofa, S.; Zhang, X. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information* **2017**, *8*, 44. [CrossRef]
35. Yue, X.; Wang, H.; Jin, D.; Li, M.; Jiang, W. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *J. Med. Syst.* **2016**, *40*, 218. [CrossRef] [PubMed]
36. Badr, S.; Gomaa, I.; Abd-Elrahman, E. Multi-tier blockchain framework for IoT-EHRs systems. *Procedia Comput. Sci.* **2018**, *141*, 159–166. [CrossRef]
37. Boneh, D.; Franklin, M. Identity-based encryption from the Weil Pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. [CrossRef]
38. Park, N.; Kwak, J.; Kim, H.; Kim, S.; Won, D. WIPI mobile platform with secure service for mobile RFID network environment. In Proceedings of the ICSE 2006 International Workshop on Web-based Internet Computing for Science and Engineering (In Conjunction with APWeb 2006), Harbin, China, 16–18 Janury 2006; Springer-Verlag: Berlin/Heidelberg, Germany, 2006; pp. 741–748.
39. Ateniese, G.; Fu, K.; Green, M.; Hohenberger, S. Improved proxy re-encryption schemes with applications to secure distributed storage. In Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2–4 February 2005.
40. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 11 March 2019).