A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of Things Based on Blockchain

Kwame Opuni-Boachie Obour Agyekum 🅞, Qi Xia 🅞, Emmanuel Boateng Sifah 🅞, Christian Nii Aflah Cobblah 🅞, Hu Xia 🅞, and Jianbin Gao 🅞

Abstract—The evolution of the Internet of Things has seen data sharing as one of its most useful applications in cloud computing. As eye-catching as this technology has been, data security remains one of the obstacles it faces since the wrongful use of data leads to several damages. In this article, we propose a proxy re-encryption approach to secure data sharing in cloud environments. Data owners can outsource their encrypted data to the cloud using identity-based encryption, while proxy re-encryption construction will grant legitimate users access to the data. With the Internet of Things devices being resource-constrained, an edge device acts as a proxy server to handle intensive computations. Also, we make use of the features of information-centric networking to deliver cached content in the proxy effectively, thus improving the quality of service and making good use of the network bandwidth. Further, our system model is based on blockchain, a disruptive technology that enables decentralization in data sharing. It mitigates the bottlenecks in centralized systems and achieves fine-grained access control to data. The security analysis and evaluation of our scheme show the promise of our approach in ensuring data confidentiality, integrity, and security.

Index Terms—Access control, blockchain, data security, identity-based proxy re-encryption, information-centric network (ICN), Internet of Things (IoT).

I. INTRODUCTION

THE Internet of Things (IoT) has emerged as a technology that has great significance to the world nowadays and its utilization has given rise to an expanded growth in network traffic volumes over the years. It is expected that a lot of devices will get connected in the years ahead. Data is a central notion to the IoT paradigm as the data collected serves several purposes in applications such as healthcare, vehicular networks, smart cities, industries, and manufacturing, among others [1]. The sensors measure a host of parameters that are very useful for stakeholders involved. Consequently, as enticing as IoT seems to be, its advancement has introduced new challenges to security and privacy. IoT needs to be secured against attacks that hinder it from providing the required services, in addition to those that pose threats to the confidentiality, integrity, and privacy of data.

A viable solution is to encrypt the data before outsourcing to the cloud servers. Attackers can only see the data in its encrypted form when traditional security measures fail. In data sharing, any information must be encrypted from the source and only decrypted by authorized users in order to preserve its protection. Conventional encryption techniques can be used, where the decryption key is shared among all the data users designated by the data owner. The use of symmetric encryption implies that the same key is shared between the data owner and users, or at least the participants agree on a key. This solution is very inefficient. Furthermore, the data owners do not know in advance who the intended data users are, and, therefore, the encrypted data needs to be decrypted and subsequently encrypted with a key known to both the data owner and the users. This decrypt-and-encrypt solution means the data owner has to be online all the time, which is practically not feasible. The problem becomes increasingly complex when there are multiple pieces of data and diverse data owners and users.

Although simple, the traditional encryption schemes involve complex key management protocols and, hence, are not apt for data sharing. Proxy re-encryption (PRE), a notion first proposed by Blaze *et al.* [2], allows a proxy to transform a file computed under a delegator's public key into an encryption intended for a delegatee. Let the data owner be the delegator and the data user be the delegate. In such a scheme, the data owner can send encrypted messages to the user temporarily without revealing his secret key. The data owner or a trusted third party generates the re-encryption key. A proxy runs the re-encryption algorithm with the key and revamps the ciphertext before sending the new ciphertext to the user. An intrinsic trait of a PRE scheme is that the proxy is not fully trusted (it has no idea of the data owner's secret key). This is seen as a prime candidate for delegating access to encrypted data in a secured manner, which is a crucial component in any data-sharing scenario. In addition, PRE allows for encrypted data in the cloud to be shared to authorized users while maintaining its confidentiality from illegitimate parties.

Data disclosures can be minimized through the use of encryption since only users delegated by the data owner can effectively access the outsourced data.

Motivated by this scenario, this article proposes an improvement in IoT data sharing by combining PRE with identity-based encryption (IBE), information-centric networking (ICN), and blockchain technology. Shamir [3] first presented the notion of IBE, in which a sender encrypts a message to a recipient using the identity (email) as the public key. It is a very powerful primitive used to combat numerous key distribution problems and has consented to the development of several cryptographic protocols, including public-key searchable encryption [4], [5], secret handshakes [6], and chosen ciphertext attack (CCA) secure public-key encryption [7]. IBE is preferred over attribute-based encryption (ABE) because ABE involves heavy computations on data encryption, decryption, and key management, and these processes are not convenient for the resource-constrained IoT devices. The strength of this article is increased by borrowing the idea of ICN to cater for the growth in information sharing.

The appeal for low-latency applications introduced the notion of ICN [8]–[11], where data owners can distribute and assign unique names to their data which can be replicated and saved in network caches [12], [13]. This ensures that there is an efficient data delivery and utilization of network bandwidth, which is a prerequisite for the IoT ecosystem regardless of the enormous growth in network volumes. On issues of trust, a decentralized, distributed system that can smoothen secure and trusted data sharing was introduced by Nakamoto [14]. This is the blockchain technology, and it has gained much attention due to its ability to preserve data privacy. Although there exist optimization issues when storing vast sizes of data, emerging system applications have used the blockchain for access control in database management. Data confidentiality and user revocation can also be achieved using blockchain.

PRE, together with IBE and the features of ICN and blockchain, will enhance security and privacy in data-sharing systems. PRE and IBE will ensure fine-grained data access control, while the concept of ICN promises a sufficient quality of service in data delivery because the in-network caching provides efficient distribution of data. The blockchain is optimized to prevent storage and data-sharing overheads and also to ensure a trusted system among entities on the network. In our article, the data owner propagates an access control list which is stored on the blockchain. Only the authorized users are able to access the data. The contributions of this article are summarized as follows.

1) We propose a secure access control framework to realize data confidentiality, and fine-grained access to data are achieved. This will also guarantee data owners' complete control over their data.
2) We give a detailed description of our PRE scheme and the actualization of a complete protocol that guarantees security and privacy of data.
3) To improve data delivery and effectively utilize the network bandwidth, edge devices serve as proxy nodes and perform re-encryption on the cached data. The edge devices are assumed to have enough computation capabilities than the IoT devices and as such provide high performance networking.

4) The security analysis of our scheme is presented, and we also test and compare its performance with existing schemes.

This article is structured as follows. Section II reviews some literature on PRE, IBE, ICN, and blockchain for data sharing and access control. Security definitions and preliminaries are formally described in Section III. In Section IV, we define a data-sharing problem and present the system model. The implementation of our model is illustrated in Section V and the formal security analysis is outlined in Section VI. Section VII evaluates and discusses our proposed scheme, while Section VIII concludes the article.

II. RELATED WORKS

In this section, we review some of the applications of the technologies used in this article in relation to data sharing and access control in the cloud.

A. PRE Data Sharing

Yu et al. [15] combined key-policy ABE (KP-ABE) and PRE to propose a system for data sharing in the cloud. The data was encrypted using KP-ABE which meant that only an appropriate collection of the attribute secret keys can make decryption possible. Besides the encrypted data, the cloud also managed all attribute secret keys except one special secret key in order to handle revocation of users. When users are revoked, new keys were distributed to the remaining users by the data owner and the encrypted data had to be re-encrypted. Although the scheme was efficient, the re-encryption was performed in a lazy way, and, therefore, the security of the scheme was weakened. Park [16] provided a modification to the scheme in [15], where collusion between the service provider and revoked users is avoided. Their scheme was to basically replace the service provider with a trusted third party, which implies that there should be reliance on stronger trust assumption. Other schemes [17]–[19] have made similar approaches but utilized ciphertext-policy ABE (CP-ABE) rather, in which the access policy is associated with the ciphertext instead of the secret keys. Liu et al. [20] also proposed a time-constrained access control scheme based on PRE and ABE. ABE was used to design time-based access control policies while PRE was used to update the time attributes. Although these schemes have their advantages, they are not suitable in the context of IoT due to the heavy computations on encryption and decryption.

An IBE PRE scheme suitable for data sharing was presented by Han et al. in [21]. The re-encryption keys were not only bound to the users' identities but also to a specific ciphertext. This implied that the data owner had to create a different re-encryption key for each pair of data user and shared file. A similar idea was proposed by Lin et al. [22] where they used a hierarchical PRE instead of an identity-based PRE. These two schemes tend to be inefficient when multiple and complex data pieces are considered. An identity-based broadcast encryption (IBBE) combined with PRE was proposed by Zhou et al. in [23] for data sharing. Their scheme was a hybrid one that allowed the conversion to be done between the two protocols without leaking any sensitive information. Wang et al. [24] also designed

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

AGYEKUM *et al.*: PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE IOT BASED ON BLOCKCHAIN 3

an identity-based PRE (IBPRE) scheme for accessing health records. The scheme achieved coarse-grained access control. If a proxy receives the re-encryption key from the data owner, either all the ciphertexts can be re-encrypted and accessible to the intended users or none at all. On that note, Shao *et al.* [25] proposed an IBE PRE scheme that is based on conditions. In their proposal, the proxy could transform a subset of ciphertexts under an identity to other ciphertexts under another identity. However, decryption rights to a group of users could not be authorized. In addition to the above, PRE has been used to mitigate security problems in IoT [26].

B. Blockchain-Based Access Control and Data Sharing

Zyskind *et al.* [27] used blockchain to provide distributed personal data management and ensure privacy as well. The blockchain was utilized as an automatic access control manager, and, hence, no third party was required. Only the data address was stored on the blockchain and a distributed hash table was used as the implementation of the data storage. This reduced the risk of data leakage. However, no specific access control model was proposed in their scheme. Maesa *et al.* [28] proposed a blockchain-based access control scheme where the data owner defines policies on the data and stores them on the blockchain. The policies are then assigned to the users as access rights.

Fan *et al.* [29] designed a similar model to [28] where the encrypted data is uploaded to the cloud and access policies on the data are stored on the blockchain as transactions. Although these two schemes achieve tamper-proof systems and easy auditing, there is a leakage of access policies since the blockchains used are public ones and are thus visible to everyone. Singh and Kim [30] presented a blockchain-based model for sharing data in vehicular networks and also enable secure communication among vehicles. However, the use of a public blockchain does not work well in peer-to-peer (P2P) data sharing among vehicles due to the high cost involved in establishing a public blockchain in resource-constrained vehicles.

C. Access Control Schemes for ICN

In order to control content in ICN frameworks, several centralized and decentralized access control mechanisms have been proposed in literature. Silva and Zorzo [31] presented an access control system for named data networking which relied on an ABE scheme and a proxy server. Before a content is published, the data owner encrypts the content and generates an access policy that binds it. The encrypted data is stored in the immediate routers while the access policy is stored on the server. When a user wants to access content, the user retrieves the content from the router, obtains the access policy from the proxy server, and then decrypts the data. Their scheme enables user revocation; however, it suffers from a single point of failure if a proxy server fails to work because the proxy server takes part in each content access. Li *et al.* [32] designed a privacy enhancing scheme using ABE for access control in ICN, and a trusted third party is deployed to manage attributes. A content publisher generates an access policy based on the attributes defined by the third party and uses a random symmetric key to encrypt the data. The publisher then hides the random key and the access policy in the content name and only authorized users can gain access to

TABLE I
NOTATION

Symbol	Meaning
CT	Ciphertext
DO	Data owner
DU	Data user
H	Hash function
id/ID	Identity
m, M	message, Message space
msk	Master secret key
RK	Re-encryption key
$params$	public parameters

the content. The proposed scheme achieves privacy by hiding the access policy in the content name, but user revocation is not guaranteed.

For decentralized access control systems, Misra *et al.* [33] proposed a secure content delivery ICN framework using Shamir's threshold secret sharing scheme and broadcast encryption but without the services of a third party. A symmetric key is used to encrypt the content which is broadcast to the network along with the key generation materials. Only authorized users can use these keying materials and decrypt the encrypted data using their individual keys. The scheme provides user revocation services, but an account of each content access or the history of keying materials' update is not kept. This makes auditing difficult. Abdallah *et al.* [34] made use of the Diffie–Hellman (DH) protocol in the process of content publishing to achieve decentralized access control. The content, its name, and metadata are sent to the ICN, while only the content name is published. After going through the various stages of the DH key exchange protocol, the ICN verifies the metadata and sends the encrypted data together with the shared key. There is no single point of failure in this scheme; however, the cached content in the ICN is in the plaintext form which makes it vulnerable to attacks.

Cloud servers are used to facilitate IoT data sharing and provide seamless, efficient, and robust sharing services in [35]–[37]. However, there are privacy concerns [38], [39]; the cloud is not trusted, and, hence, it is indispensable to enforce data access control over potentially untrusted platforms. Besides these, several schemes [40]–[42] are based on ABE. Although they are efficient, the high computations in key generation and distribution are not opportune for IoT. Inspired by the drawbacks in the applications of the various technologies for access control and data sharing, this article utilizes PRE, IBE, and the features of ICN and blockchain to solve the challenges in data sharing. To the best of our knowledge, this article is the first to combine these mechanisms to establish secure data sharing in the cloud. Ateniese *et al.* [43] proposed a re-encryption scheme that is unidirectional, noninteractive, of multiuse, and nontransitive. These properties are suitable for our proposed architecture, and, hence, the scheme is adopted in this article. A detailed construction of the security proof is also provided.

III. SECURITY DEFINITIONS

In this section, we outline the security settings and computational problems to be used in this article, after which the PRE scheme is defined. For ease of understanding, Table I shows the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

4

IEEE SYSTEMS JOURNAL

significant mathematical symbols and their notations. However, all other symbols are duly explained.

A. Bilinear Maps

Consider G_1 and G_2 to be two groups of order p for some large prime p. Our scheme utilizes a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ among these two groups. The following conditions about the map should be satisfied.

- *Bilinear:* A map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is said to be bilinear if $\hat{e}(uP, vQ) = \hat{e}(P, Q)^{uv}, \forall P, Q \in G_1, \forall u, v \in Z$.
- *Nondegenerate:* The map is nondegenerate (i.e., all pairs in $G_1 \times G_2$ are not sent to the identity in G_2). Observe that because G_1, G_2 are groups of prime order, whenever P is a generator of G_1, $\hat{e}(P, P)$ becomes a generator of G_2.
- *Computable:* There exists an efficient algorithm that computes $\hat{e}(P, P)$ for any $P, Q \in G_1$.

B. Decisional Bilinear Diffie Hellman Assumption (DBDH)

The security of our scheme is based on a variant of the computational Diffie–Hellman assumption known as the decisional bilinear Diffie–Hellman assumption (DBDH) in G_1, G_2. It is defined as follows: with $\hat{e} : G_1 \times G_1 \rightarrow G_2$, let g be a generator of G_1. When given a tuple $(g, g^u, g^v, g^w, J) \in G_1^4 \times G_2$, a decision needs to be made as to whether J is just one random element in G_2 or $J = e(g, g)^{uvw}$.

Define λ to be a security parameter. For all probabilistic polynomial time (p.p.t) algorithms $\mathcal{A}$, there exists the condition shown in (1) where $\psi(\cdot)$ is a negligible function. That is, for all polynomial functions $p(\cdot)$, $\psi(\lambda)p(\lambda) < 1$, the DBDH assumption holds in the groups (G_1, G_2) as can be seen from (1) as shown at the bottom of this page.

C. Identity-Based Encryption

$Setup, KeyGen, Encrypt,$ and $Decrypt$ are four algorithms that characterize an IBE scheme, and they are defined below.

1) *Setup* $[(params, msk) \leftarrow \lambda]$: The setup algorithm takes a security parameter λ, as the input and outputs a set of public parameters $params$ and a master key msk. The public parameters contain a description of the message space M and also a description of the ciphertext CT. The public parameters are known while the master key is kept secret.
2) *KeyGen* $[\alpha \leftarrow (params, msk, ID)]$: The key generation algorithm takes the public parameters, the master key, and an arbitrary $ID \in (0, 1)^*$ as inputs and produces a decryption key α, which corresponds to the *ID*.
3) *Encrypt* $[CT \leftarrow (params, ID, m \in M)]$: The encryption algorithm returns a ciphertext CT after taking the public parameters, an *ID*, and a message m as inputs.
4) *Decrypt* $[m \leftarrow (params, CT, \alpha)]$: The decrypt algorithm takes the public parameters, CT, and the decryption key

as inputs and returns the message m. The constraint in the following equation must be satisfied:

$$\forall m \in M : Decrypt$$
$$(params, Encrypt(params, ID, M), \alpha) = m. \quad (2)$$

D. Identity-Based Proxy Re-Encryption

This scheme is an extended version of the IBE scheme. The difference between IBPRE and IBE schemes is the introduction of two algorithms; a re-encryption key generation algorithm $ReKey$ and a re-encryption algorithm $ReEnc$. The data owner generates the re-encryption key RK and hands it to the proxy. Then the proxy uses RK to transform ciphertexts. $ReKey$ and $ReEnc$ algorithms are defined below.

1) *ReKey:* On inputting the public parameters, corresponding secret key, and IDs $(ID_{DO}, ID_{DU}) \in \{0, 1\}^*$, the algorithm returns the re-encryption key RK which is given as $RK_{ID_{DO} \rightarrow ID_{DU}} \leftarrow (params, \alpha_{ID_{DO}}, ID_{DO}, ID_{DU})$.
2) *ReEnc:* When the inputs are the public parameters, re-encryption key, and the original ciphertext under identity ID_{DO}, the re-encrypted ciphertext is produced. That is $CT_{ID_{DU}} \leftarrow (params, RK_{ID_{DO} \rightarrow ID_{DU}}, CT_{ID_{DO}})$.

Correctness: To ascertain the correctness of the scheme, an IBPRE scheme is correct when the expected outcome of a properly formulated ciphertext is obtained if the $Decrypt$ algorithm is run. More formally, let $\alpha_{ID_{DO}} \leftarrow KeyGen(msk, ID_{DO})$, $\alpha_{ID_{DU}} \leftarrow KeyGen(msk, ID_{DU})$, and $ReKey \leftarrow (params, \alpha_{ID_{DO}}, ID_{DO}, ID_{DU})$, for this constraints (3)–(5) shown at the bottom of the next page must be satisfied.

E. Security Model

For a scheme defined by the tuples as stated earlier, its security is based on the indistinguishability against proxy identity and chosen plaintext attack (CPA), $IND_{\text{PRID/CPA}}$. There are five stages involved in this security game where the adversary $\mathcal{A}$ engages the challenger $\mathbb{C}$ in a series of games.

1) *Select phase:* The attacker selects $\mu \in (0, 1)$ and gives to the challenger.
2) *Setup phase:* The challenger obtains $params, msk$ after running the $Setup$ algorithm and gives $params$ to $\mathcal{A}$.
3) *Find phase:* The adversary makes the following queries. $\mathcal{A}$ selects an $id^* \in (0, 1)^*$ and $(m_0, m_1 \in M^2)$ at the conclusion of this phase.
   a) $\mathbb{C}$ returns $msk = KeyGen(params, msk, id)$ to $\mathcal{A}$ when a query of $(KeyGen, id)$ is made.
   b) For the situation where $id_{DO} \neq id_{DU}$, the re-encryption key IDDU is given to $\mathcal{A}$ when a query of the form $(ReKey, id_{DO}, id_{DU})$ is made.
   c) When $\mathcal{A}$ queries (Dec, id, CT), return $\perp$.

$$\left| \begin{array}{l} Pb\left[u, v, w \xleftarrow{\$} Z_p^*; 1 \leftarrow \mathcal{A}\left(g, g^u, g^v, g^w, e(g, g)^{uvw}\right),\right] - \\ Pb\left[u, v, w \xleftarrow{\$} Z_p^*; J \xleftarrow{\$} G_2; 1 \leftarrow \mathcal{A}\left(g, g^u, g^v, g^w, J\right),\right] \end{array} \right| \leq \psi(\lambda) \quad (1)$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

AGYEKUM *et al.*: PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE IOT BASED ON BLOCKCHAIN 5

d) When $\mathcal{A}$ queries $(ReEnc, id_{DO}, id_{DU}, CT)$, return $\perp$. $\mathcal{A}$ is not authorized to choose id^* in such a way that there is a possibility of a trivial decryption using keys generated during this phase.

4) *Decision and Challenge phase:* $\mathbb{C}$ computes and gives $CT^* = Enc(params, id^*, m_\mu)$ to $\mathcal{A}$, when the adversary presents $(choice, id^*, m_0, m_1)$.

5) *Guess phase:* Just as in the *find phase*, the adversary continues to make queries until at the end of this stage $\mathcal{A}$ yields μ^*, where $\mu^* \in (0, 1)$. The adversary wins the game if $\mu^* = \mu$. With the security in the random oracle, let $(KeyGen, RK, Dec, ReEnc)$ and $(KeyGen', RK', Dec', ReEnc')$ be algorithms in the find and guess phases, respectively. The adversary's advantage in the game is defined in (5). The security of the scheme against the attack is achieved if for all p.p.t algorithms $\mathcal{A}$, $Adv_{\mathcal{A}}^{IND_{\text{PRID/CPA}}} \leq \psi(\lambda)$.

IV. PROBLEM DEFINITION AND SYSTEM OVERVIEW

In this section, we illustrate a simple data-sharing problem and introduce the system model.

A. Problem Definition

IoT data sharing has become prevalent in several applications, ranging from healthcare and vehicular networks to smart homes and energy trading. Whenever an IoT device (sensor, page maker, smart phone, etc.) wants to share its data among other users, the data is usually encrypted and outsourced to cloud repositories. Access rights and privileges are bound to this data to preserve privacy, enable an efficient access mechanism, and prevent malicious activities in the network. Fig. 1 epitomizes a data-sharing scenario.

In such a system, the data producers are the entities that generate the data. They can participate in data protection from the onset by encrypting the data and outsourcing it to the cloud service providers (CSPs) themselves. Generation does not necessarily translate to ownership and, hence, the distinction between data producers and the data owners. The data owners usually center on who owns the data. The data owner generates a random number which is used to encrypt the data before uploading into the cloud and sharing with prospective users. Access rights on the data are initiated. Data owners can be producers themselves; however, this does not rule out the possibility of separate entities getting involved in data production. It is assumed that the data owners communicate with other entities through an agent/server that runs on a trusted computer.

The data user domain consists of legitimate recipients of the information that is shared by the owners/producers. The users
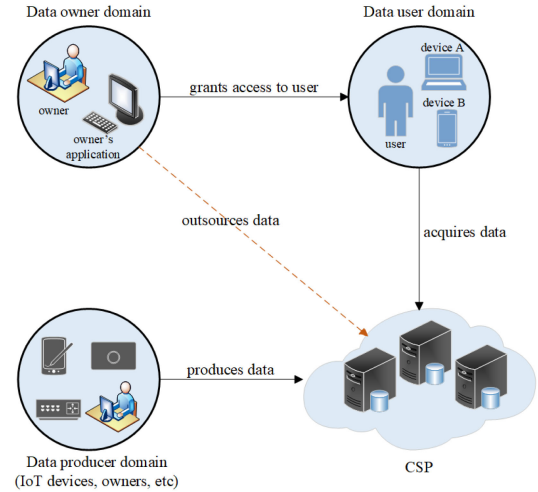

Fig. 1. Simplified data-sharing platform.

not only comprise people but devices as well. These data users must access the shared data from the CSP which is a semitrusted party that offers storage services to the data. It houses the encrypted data from the owner and the data is received through a secure communication channel. They provide data-sharing services without being able to learn anything about the plaintext.

Any information that must be accessed should be encrypted from the source and decrypted by only legitimate users. Nonetheless, due to its semitrust nature, the CSP may have incentives for trying to read the data. With data sharing comes instances where $user_2$ might want to access a particular data which had been previously shared between the data owner and $user_1$. To improve the quality of service in data delivery and have an efficient use of the bandwidth, there is the need for the cached content in edge nodes to be shared with $user_2$ using its identity or credentials, instead of obtaining that same data from the cloud server and performing another encryption. This prevents overhead and increases the network performance.

B. System Model

Our system model in Fig. 2 introduces a blockchain-based PRE approach to data sharing. The additional entities to the data-sharing model as discussed in Fig. 1 are the edge devices and the blockchain. The edge devices serve as proxy nodes and provide re-encryption services to the authorized user(s). When the data is cached at the edge of the network, the edge devices provide services to users with high availability and performance. They receive the re-encryption key from the data owner, fetch

$$m \leftarrow Decrypt\left(params, CT_{ID_{DO}}, \alpha_{ID_{DO}}\right) \tag{3}$$

$$m \leftarrow Decrypt\left(params, \alpha_{ID_{DU}} ReEnc\left(params, RK_{ID_{DO} \rightarrow ID_{DU}}, CT_{ID_{DO}}\right)\right) \tag{4}$$

$$\left| Pb\left[\mu^* = \mu \middle| \begin{array}{l} \mu \leftarrow (0, 1); Setup(\lambda) \rightarrow (params, msk) \\ A^{[KeyGen(\cdot), RK(\cdot), Dec(\cdot), ReEnc(\cdot)]}(params) \rightarrow (id^*, m_0, m_1, j) \\ CT^* \leftarrow Enc\left(params, id^*, m_\mu\right) \\ A^{[KeyGen'(\cdot), RK'(\cdot), Dec'(\cdot), ReEnc'(\cdot)]}(params, CT^*, j) \rightarrow \mu^* \end{array} \right] - 1/2 \right| \tag{5}$$
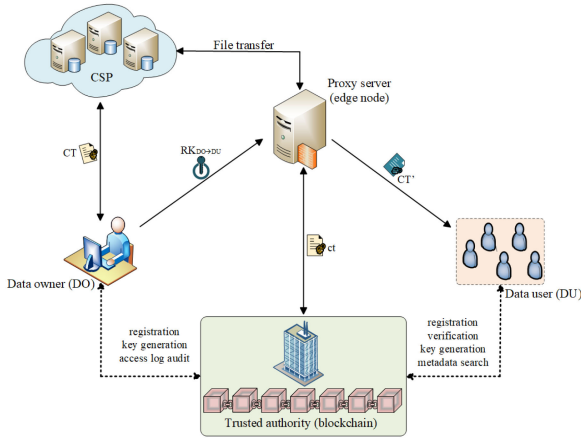
Fig. 2. Data-sharing system model.

the ciphertext from the CSP, and transform the ciphertext in the identity of the data user. It is an honest-but-curious entity.

The blockchain serves as the trusted authority (TA) that initiates the system parameters. The TA also provides secret keys that are bound to the users' identities. By utilizing this distributed ledger, authenticity, transparency, and verifiability are achieved in the network, which enhances the security and privacy of data. Data owners are therefore able to manage their data effectively. The blockchain network registers and issues membership keys to the data owner(s) and user(s). When a user requests data access, the owner generates a re-encryption key by using the identity of the user and sends it to the proxy server. Access rights and policies on the use of the data are instantiated and sent to the blockchain network. A data user is verified before access is granted.

The TA runs the $Setup$ algorithm to generate system parameters and a master key in the system initialization phase. Simultaneously, the $KeyGen$ algorithm is used to create keys for the users. The data owner runs the $Encrypt$ algorithm to create a ciphertext CT. The ciphertext is then outsourced to the CSP and the metadata is stored on the blockchain.

In our model, incorporating data caches in the forwarding process ensures that content delivery is more robust against packet losses, and this improves the availability of the content. Not only does it support content caching but functionality caching (which is re-encryption in this case) as well. Also, the multipoint delivery system of ICN assures an effective utilization of bandwidth and storage. When the number of users increases, the content will not be unicasted and this will reduce the bandwidth usage.

V. SYSTEM IMPLEMENTATION

In this section, we give concrete details of the workflow of the system and how the blockchain works as well. The re-encryption scheme is also described.

A. System Workflow

Data storage and retrieval on the system are detailed as follows. The hash of the data is calculated using the $(SHA-256)$ hashing algorithm to achieve data integrity. The data owner generates a random number which is used to encrypt the data and the resulting ciphertext is uploaded to the CSP. A metadata is created to support search functionality and the data owner

produces a digital signature on the data by using his private key to sign the hash function.

The data owner generates the re-encryption key based on the identity of the user and gives it to the proxy server. The user is included in an access list which is sent to the proxy server. The proxy verifies the owner's signature for authenticity. Having stored CT on the CSP, the proxy retrieves a uniform resource locator (URL) to the ciphertext and generates and assigns an ID (d_{ID}) to the URL. The server appends its signature on d_{ID} which is then cached in the proxy server. Finally, the metadata, access control policy, signatures of both the data owner and the proxy server, hash, and d_{ID} are uploaded to the blockchain.

When a user places a request for data access, the user queries the metadata on the blockchain. The authenticity of the data is verified by checking the signatures of the data owner and the proxy server. A timestamp is appended if authentication is successful, after which the signed data is sent to the proxy server in a request for the actual data. The related information on the data is fetched from the cache, while the associated ciphertext is also retrieved from the CSP. The proxy server performs ciphertext re-encryption and sends the result to the user. The user can now decrypt the ciphertext with his private key. The blockchain beforehand verifies the authenticity of the user by using his signature. The timestamp is verified and the request is stored on the blockchain for auditing purposes.

B. Blockchain

Blockchain technology is seen as a disruptive technology that can play a major role in securing IoT devices. As a decentralized, distributed paradigm, the blockchain uses a cryptographically linked chain of blocks to validate and store processed data. A consensus algorithm is used by the processing nodes in generating the blocks. Smart contracts, which are programmable scripts that are automatically executed, are used to manipulate the data. A generated block consists of a header and a body. Constituents of a block header include a current version number, the address of the previous block, the target hash value of the current block, a Merkle root, a nonce, and a timestamp. A block body typically consists of transactions, and they differ in application areas.

The components of the block header are vital in generating an accurate and reliable header. The previous block's hash is a 32-b long string that effectively secures the chain by being linked to the previous block or the parent block. A 4-b long nonce is a value used by miners to create different permutations and also create a correct hash in the sequence. The timestamp enables everyone to see the encoded record of a particular event. It usually provides the date and time of block creation, and it is 4-b long. The Merkle root is a 32-b long string that contains all the hashed transactions within a hashed transaction. The version number keeps track of changes and updates while the target difficulty is used to adjust how hard it is for miners to solve the block. Their byte length is 4 each. In all, the header is an 80-b long-string. The structure of a block is shown in Fig. 3.

Practical byzantine fault tolerance (PBFT) is a consensus algorithm that is adopted in this article. Processing nodes in the blockchain serve as miners responsible for block creation. Whenever a block is received, the nodes get engaged in a voting process before reaching consensus. The PBFT algorithm verifies the correctness of a block. Each processing node can become a

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

AGYEKUM *et al.*: PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE IOT BASED ON BLOCKCHAIN 7
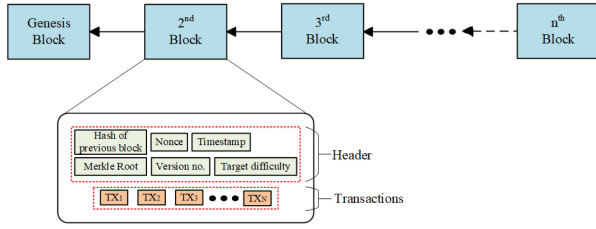

Fig. 3. Block structure.

leader because each has complete access to the transaction. In a consortium blockchain, the leader is chosen until after the consensus process unlike in public blockchains where mining incurs high costs and lengthy delays because there is a cryptographic puzzle to be solved in Proof of Work consensus algorithm. Digital signatures are used to sign the encrypted transactions to guarantee their authenticity. The signed transactions are then cryptographically linked to form a tamper-proof block. Several such blocks are then chronologically linked by hash pointers to form a chain.

In dynamic IoT environments, centralized data services result in high bandwidth use and server load and are, therefore, not scalable to meet the growing demands of IoT systems. A consortium blockchain is adopted due to its suitability to access control and privacy preservation. Only authorized users can have access to the data. Data owners can effectively manage their data and audit logs. Consortium blockchains provide a high level of security. IoT security concerns that are addressed by the blockchain network include verifying the identity of the connected users or devices, their account information, and also preventing cached data from being misused.

Because edge devices have enough computing resources and storage, they act as proxy servers to provide re-encryption services and other computations for the resource-constrained IoT devices. It is, therefore, easy to cache data at these edge nodes. Retrieving data via high-speed networks, the user can make requests for data access, thus providing a smooth user experience. Due to the dynamic nature and mobility of edge networks, it is a requirement that the edge devices and stakeholders in general have unique identities. The ID of all entities on the network is represented by the tuple (id, k_{pu}, k_{pr}, rl). id is the cryptographic hash of the public key k_{pu}, i.e., $id = hash(k_{pu})$. k_{pr} denotes the private key and rl is the role of the entity. Apart from the data owner and users serving their roles as their names suggest, the edge devices themselves could also be data users. Before a transaction can be initiated, all identities need to be known and verified. If the verification fails, the connection is terminated. The S/Kademlia static crypto puzzle [44] is used to create the public and private keys to prevent Sybil attack. The public key is used to sign messages (transactions) in order to verify their authenticity.

C. Scheme Construction

The scheme is formally described in this section.
1) *System Setup:* Let the bilinear map be defined as $\hat{e}: G_1 \times G_1 \to G_2$, where $G_1 = \langle g \rangle$ and the order of G_2 is p. H_1 and H_2 are two hash functions defined by $H_1: G_1 \leftarrow (0, 1)^*$, $H_2: G_1 \leftarrow G_2$. The public parameters are

generated as $params = (G_1, H_1, g, g^\delta)$. δ is the secret key which is selected from the group Z_p^*.
2) *Key Generation:* Given the public parameters, the secret key, and an ID, this algorithm extracts the decryption key for identity $id \in (0, 1)^*$ and returns the secret key of the data owner, $xk_{ID_{DO}} = H_1(id_{DO})^\delta$.
3) *Encryption:* In order to encrypt m using the identity of the data owner, a random number $r \in Z_p^*$ is selected and the output ciphertext is given as $CT_{ID_{DO}} = (CT_1, CT_2)$ where $CT_1 = g^r$, $CT_2 = m \cdot e(g^\delta, H_1(id_{DO}))^r$.
4) *Re-encryption Key Generation:* ϑ is selected from G_2 and the tuple $\langle \Psi_1, \Psi_2 \rangle = Enc(params, id_{DU}, \vartheta)$. The resulting re-encryption key is given as $RK_{ID_{DO} \to ID_{DU}} = \langle \Psi_1, \Psi_2, xk_{ID_{DO}}^{-1} \cdot H_2(\vartheta) \rangle$.
5) *Re-encryption:* In order to re-encrypt CT from the data owner to the data user, $RK_{ID_{DO} \to ID_{DU}} = (\Psi_1, \Psi_2, \Psi_3)$ and the re-encrypted ciphertext is defined as $CT_{ID_{DU}} = \langle CT_1, CT_2 \cdot e(CT_1, \Psi_3), \Psi_1, \Psi_2 \rangle$.
6) *Decryption:* To obtain the message, $m = CT_2/e(CT_1, xk_{ID})$. For the re-encrypted ciphertext, $CT'_{ID} = \langle CT_3, CT_4 \rangle$, compute $\vartheta_2 = Dec(xk_{ID}, CT'_{ID})$ and retrieve the plaintext via $\vartheta = CT_2/e(CT_1, H_2(\vartheta_2))$.

Correctness: For a ciphertext produced from the Enc algorithm, $CT_{ID_{DO}} = (g^r, m \cdot e(g^\delta, H_1(id_{DO}))^r)$ and $xk_{ID_{DO}} = H_1(id_{DO})^\delta$, m can be recovered as follows:

$$m = \frac{CT_2}{e(CT_1, xk_{ID_{DO}})}$$

$$= \frac{m \cdot e(g^\delta, H_1(id_{DO}))^r}{e(g^r, H_1(id_{DO})^\delta)}$$

$$= m.$$

Having $CT_{ID_{DO}} = (g^r, CT_2)$ and $RK_{ID_{DO} \to ID_{DU}} = (\langle \Psi_1, \Psi_2 \rangle = (params, id_{DU}, \vartheta)\Psi_3)$, the re-encrypted ciphertext can be obtained as $CT_{ID_{DU}} = (g^r, CT'_2 = CT_2 \cdot e(g^r, \Psi_3), \Psi_1, \Psi_2)$ where

$$CT'_2 = CT_2 \cdot e(g^r, \Psi_3)$$

$$= m \cdot e(g^r, H_1(id_{DO}))^r \cdot e\left(g^r, H_1(id_{DO})^{-\delta} \cdot H_2(\vartheta)\right)$$

$$= m \cdot e(g, H_2(\vartheta))^r.$$

The resulting ciphertext $CT_{ID_{DU}} = (g^r, CT'_2, \Psi_1, \Psi_2)$, and when given $xk_{ID_{DU}} = H_1(id_{DU})^\delta$, the message can be obtained as follows. Let $CT'_{ID_{DU}} = \langle \Psi_1, \Psi_2 \rangle$. This can be decrypted under $xk_{ID_{DU}}$ to obtain $\vartheta = Dec(params, xk_{ID_{DU}}, CT'_{ID_{DU}})$. The message can then be computed as

$$m = \frac{CT'_2}{e(g^r, H_2(\vartheta))}$$

$$= \frac{m \cdot e(g, H_2(\vartheta))^r}{e(g^r, H_2(\vartheta))}$$

$$= m.$$

In practice, it is observed that the scheme exhibits unidirectionality since $RK_{DO \to DU}$ can be used to transform ciphertexts from the data owner to the data user and not vice versa. Also, the
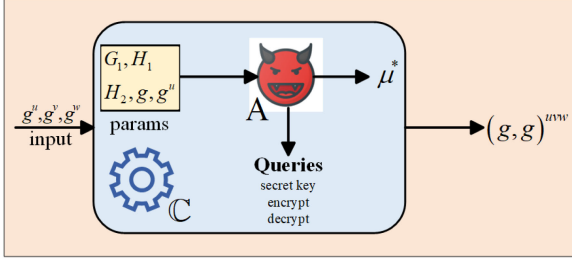
Fig. 4.   Structure of IND-CPA security proof.

data user is not involved in $RK_{DO \to DU}$ generation, and, hence, that makes it noninteractive. Moreover, the proxy is not granted the permission to generate a new $RK_{DO \to DU}$ from existing ones making the scheme nontransitive. Finally, the scheme exhibits a multiuse property. That is, the proxy can perform re-encryption on an already re-encrypted message multiple times. Realizing that CT_3' is just the identity of the data user, the re-encryption protocol can be recursively applied to CT_3' by the proxy to allow another data user recover the original message m.

VI. SECURITY PROOF AND ANALYSIS

The security proof and analysis of our scheme are discussed in this section. Furthermore, we outline the attacks that our system can counter.

A. Security Proof

Theorem 6.1. The system is $IND_{\text{PRID/CPA}}$ secure under the DBDH assumption.

Proof: The interaction between the adversary and the challenger is shown in Fig. 4. Consider $\mathcal{A}$ to be a p.p.t algorithm with non-negligible advantage ϵ in $e^{IND_{\text{PRID/CPA}}}$. In order to formulate another algorithm $\mathbb{C}$ that has non-negligible advantage in solving the DBDH problem in G_1, G_2, $\mathcal{A}$ is engaged. $\mathbb{C}$'s input is the tuple $\langle G_1 = \langle g \rangle, g^u, g^v, g^w, J \rangle \in G_1^4 \times G_2$ for which the output will be 1 if $J = e(g, g)^{uvw}$. The interaction between $\mathcal{A}$ and $\mathbb{C}$ is shown below.

The random oracle $G_1 \leftarrow H_1 : (0, 1)^*$ is simulated by $\mathbb{C}$ as follows: When an ID query is received, a random number $\theta \to Z_p^*$ is selected and a randomly flipped coin $\eta \to 1$ with probability χ is set. Otherwise, $\eta \to 0$. $h \leftarrow (g^w)^\theta$ when $\eta \to 0$, else $h \leftarrow g^\theta$. The tuple (ID, h, θ, η) is recorded. h is returned as the query result, for which it has a random distribution. $\mathbb{C}$ continues to simulate the random oracle $H_2 : G_1 \leftarrow G_2$. It returns random elements in G_1.

1) *Setup phase:* $\mathcal{A}$ is given $params = (G_1, H_1, H_2, g, g^u)$ as generated by $\mathbb{C}$.
2) *Find phase:* $\mathbb{C}$ evaluates $H(ID)$ after $\mathcal{A}$ has submitted $(KeyGen, ID)$ to obtain (ID, h, θ, η). A secret key $msk_{ID} = (g^u)^\theta$ belonging to the queried ID is given to $\mathcal{A}$. When $\mathcal{A}$ sends the query $(ReKey, ID_{DO}, ID_{DU})$, $\mathbb{C}$ selects random numbers $r \xleftarrow{\$} Z_p^*$, $x \xleftarrow{\$} G_1$ and $\vartheta \xleftarrow{\$} G_2$ and evaluates $(\eta_1, \theta_1) \leftarrow H_1$ and $(\eta_2, \theta_2) \leftarrow H_2$ for ID_{DO} and ID_{DU}, respectively.
   a) When $\eta_1 = 0$, $\mathcal{A}$ receives $RK_{ID_{DO} \to ID_{DU}} = ((g^v)^r, J^{r\theta_2} \cdot \vartheta, x)$ from $\mathbb{C}$.
   b) When $\eta_1 = 1$, $\mathcal{A}$ receives $RK_{ID_{DO} \to ID_{DU}} = (g^r, e(g^u, H_1(ID_{DU})^r) \cdot \vartheta, (g^u)^{-\theta_1} \cdot H_2(\vartheta))$ from $\mathbb{C}$.
3) *Challenge phase:* $\mathcal{A}$ outputs ID^*, m_0, m_1 at the end of the find phase but such that the choice of ID^* is not trivial. $\mathbb{C}$ selects $\mu \leftarrow (0, 1)$ and then recovers ID^*, h, θ, η by evaluating $H_1(ID^*)$. The ciphertext $CT^* = \{g^v, J^\theta \cdot m_\mu\}$ is given to $\mathcal{A}$.
4) *Guess phase:* $(KeyGen, \ldots)$ and $(ReKey, \ldots)$ queries are made by $\mathcal{A}$ as in the find phase, except with a restriction on making queries that result in trivial solutions. $\mathcal{A}$ outputs its guess $\mu^* \in (0, 1)$. If any of the following conditions turns out to be false, $\mathbb{C}$ terminates the simulation. Else, it outputs 1 if $\mu^* = \mu$, or 0 otherwise.
   a) The corresponding value of ID^*, $\eta = 0$.
   b) $n_i = 1$, for each $(KeyGen, ID_i)$ query made by $\mathcal{A}$.

For a correctly formed DBDH tuple $\langle g, g^u, g^v, g^w, J \rangle$, the view given by the adversary is identical to the real attack if $\mathbb{C}$ does not terminate the simulation. $\mathcal{A}$, therefore, cannot distinguish the simulation since it cannot notice the improperly formed re-encryption keys. The definition of $\mathcal{A}$ holds that $|Pb[\mu = \mu^*] - \frac{1}{2}| = \epsilon$ if CT^* is a correctly formed ciphertext for the encryption of m_μ under ID^* when the DBDH tuple is the input to $\mathbb{C}$. $\mathbb{C}$ thus outputs 1 with probability, $|Pb[\mu = \mu^*]| = \epsilon + \frac{1}{2}$. With a random input to $\mathbb{C}$, CT^* is the ciphertext formed for a random element in G_2, regardless of $\mathbb{C}$'s choice of μ. The probability becomes $|Pb[\mu = \mu^*]| = \frac{1}{2}$. Hence, $\mathbb{C}$ has a non-negligible advantage in distinguishing the DBDH tuples.

B. System Security Analysis

In this subsection, we analyze the attacks that our proposed system mitigates.

1) Man-in-the-Middle Attack: Our system is secure against man-in-the-middle (MITM) attacks. MITM attacks get to the certificate authority (CA) to provide the user with forged public keys. This often leads to the decryption of sensitive information. In our system, the blockchain acts as the CA. The public keys of the users are put in published blocks, and the data is distributed over the participating nodes with links to both the previous and following blocks. This makes the public key immutable and it becomes harder for attackers to publish fake keys. Also, there is no single point of failure due to the distribution.

2) Data Tampering: When hackers compromise a system, they inject their own versions of the data into the system. There is no definite way to make sure that the data has not been tampered with if the hash can be compromised and changed. In contrast, our blockchain-based model permits every user to publish a hash associated with a particular data which needs to be protected from tampering. While an attacker might be able to compromise the storage location and tamper with the data, he will not be able to change the hash stored on the blockchain. This will make it known to everyone that the data has been manipulated.

3) Anomaly Attacks: In blockchain-based systems and applications, forks become important with every chance of the evolution of a malicious purpose. Although attacks may happen once within a device, their repetition over time against other devices almost behaves in the same way. In our model, information on previous attacks is collected and blacklisted in order to prevent the attacks on entities that have not been attacked yet.

TABLE II
FUNCTIONAL COMPARISON

Functionality	ZDWQ [23]	WMXZL [24]	SWLX [25]	Our Scheme
Confidentiality of data encryption	IBBE	IBE	IBE	IBE
Re-encryption condition	-	-	keyword	Access policy
Decentralization	✕	✕	✕	✓
Security notion	IND-ID-CCA	IND-ID-CCA	IND-ID-CPA & IND-ID-CCA	IND-ID-CPA
Assumption	DBDH	DBDH	DBDH	DBDH

Information collected on forks include the start time of the fork, detection time of the fork, and the number and type of malicious transactions. These details are propagated in the network to all the peers.

VII. PERFORMANCE EVALUATION

Our performance evaluation is classified into two categories, functional comparison and performance analysis, and they are described in different sections. Our scheme is compared with the schemes in [23]–[25].

In [23], the authors presented a hybrid IBPRE scheme that allowed data that has been encrypted to multiple users to be re-encrypted to one user. It involved two separate techniques: IBBE and IBE. These schemes had different parameters and algorithms but maintained a seamless connection. IBBE was employed for users with powerful computing abilities while IBE was deployed for users with limited computing resources. Nonetheless, both schemes were used to achieve an efficient access control over outsourced data. The authors in [24] discussed the possibility of integrating IBE and IBPRE techniques and a signature scheme into an electronic-health cloud system for efficient data sharing. Basically, the work focused on proposing schemes that would be cost-effective for E-health cloud systems. The novelty of their work was the manner in which they embedded the master secret key in the private key. They analyzed the security of their approach and also showed the performance of their scheme. An ID-based conditional PRE scheme for secure and fine-grained forwarding of encrypted email was proposed by the authors in [25]. In their work, they combined several schemes to achieve chosen ciphertext and identity attack and constructed and proved their model's security.

A. Functional Comparison

Here, we compare our scheme with the ones in literature in terms of the encrypted data confidentiality, the condition(s) for re-encryption, the achieved security notion and its assumption, and whether the scheme supports decentralization. The results are shown in Table II.

From the table, it is realized that all the schemes use IBE to share encrypted data with (a set of) recipients except [23], which uses IBBE. For the re-encryption technique used, our scheme and the scheme of Shao *et al.* [25] can achieve re-encryption via a proxy using an access policy and keyword, respectively. However, the schemes presented by Zhou *et al.* [23] and Wang *et al.* [24] allow the authorized data user to re-encrypt all the data belonging to the data owner. Our scheme is decentralized in nature due to the use of blockchain, while the other schemes are centralized and rely on only CSPs for data storage and access control. They have the tendency to experience a single point of failure should the computations increase exponentially. From a

TABLE III
COMPUTATION COST COMPARISON

Scheme	Enc	Re-Enc	Dec-1	Dec-2
ZDWQ [23]	$T_E(N+5)$	$T_E(3N+3)$	$NT_E + 2T_P$	$T_E + 3T_P$
WMXZL [24]	$2(T_E + T_M)$	$T_E + T_P$	$2T_P$	$5T_P$
SWLX [25]	$4T_E + T_P$	$2T_p$	T_P	$2T_P$
Our scheme	$T_E + T_G$	T_P	T_G	$2T_G$

TABLE IV
EXPERIMENTAL PERFORMANCE IN ms

Scheme	Enc	Re-Enc	Dec-1	Dec-2
ZDWQ [23]	174.25	188.88	55.58	46.35
WMXZL [24]	21.66	20.83	24.81	12.45
SWLX [25]	20.28	19.98	23.12	9.19
Our scheme	19.97	18.86	20.99	7.03

security point of view, schemes in [23]–[25] are secure against IND-ID-CCA attack, while our scheme is secure against IND-ID-CPA attacks. This is also achieved in [25]. Furthermore, all schemes are based on the DBDH assumption.

B. Performance Analysis

The functional analysis is complimented with an experimental evaluation. Our execution environment was a Windows operating system desktop computer with 3.0 GHz, Intel i7, 16 GB RAM, 1600 MHz DDR3 specifications. We implemented the pairing-based schemes using the jPBC library [45], which is a pairing-based cryptography library for Java. A super-singular curve of the form $y^2 = x^3 + 3$ with 3072 b of field size and a group order of 256 b was used. This achieves 128 b of security and is secure against the discrete logarithm problem in G_1 and G_2. Group-based schemes were also implemented using elliptic curve cryptography over a field of prime order, and the NIST P-256 curve which also provides 128 b of security [46]. We made use of exponentiation and pairing operations for efficiency satisfaction. These are the main operations for which computational costs are based on. The results of this analysis are shown in Table III.

Let T_P be the cost of a single pairing operation, T_E be the exponent operation cost, N be the number of users, T_G be the operation in group G_2, and T_M be a multiple exponentiation operation cost. Simple multiplication, symmetric encryption and decryption, and hash costs are ignored. Interestingly, there is a great difference in performances of the various schemes. For instance, few exponentiations are needed in our scheme as opposed to the others, which require as much as 4 in [25], 2 in [24], and an infinite number in [23] due to the number of users, N. The increase in the exponentiation is due to the fact that there are additional costs incurred in achieving CCA-security.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

10   IEEE SYSTEMS JOURNAL
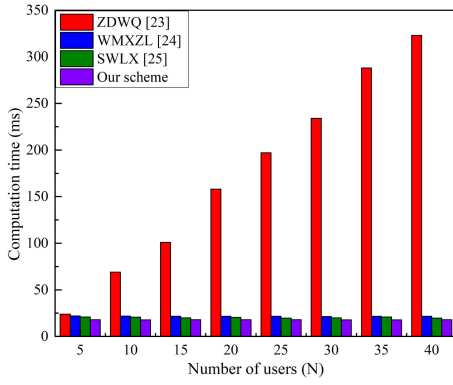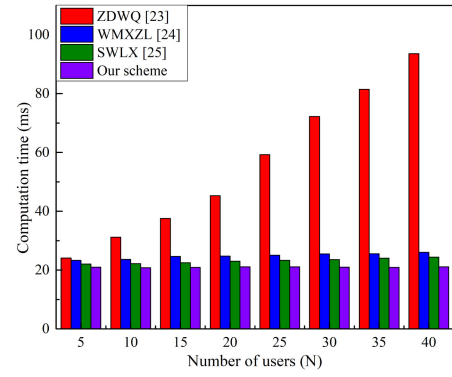

Fig. 5.   Data encryption computation time.

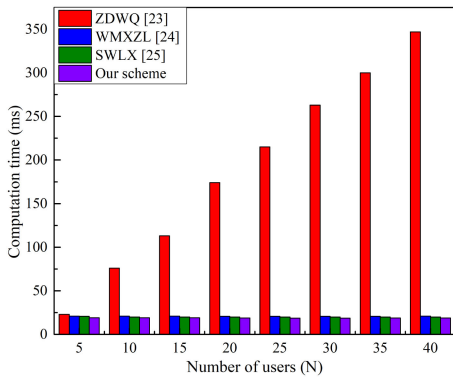Fig. 7.   Decryption-1 computation time.

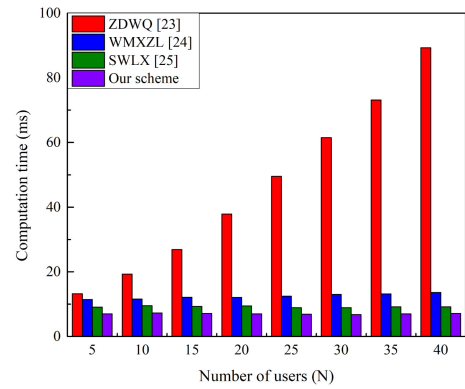Fig. 6.   Data re-encryption computation time.

Fig. 8.   Decryption-2 computation time.

Table IV shows the cost in ms of the operations of the schemes. The figures measured were as a result of the average CPU time of 50 executions for each type of operation. Fig. 5 shows the computation time for data encryption in the various schemes. It can be realized that [23] exhibits a linear growth in its encryption algorithm because it is executed for a group of users using the broadcast encryption method. In contrast, our scheme and those of [24] and [25] show a constant growth because the encryption is meant for an individual user. A similar analysis is given for the re-encryption execution time in Fig. 6. It is worth noting that the high performance of [23] is due to the increasing number of users.

Figs. 7 and 8 reveal the computation times on the user side to decrypt the first- and second-level ciphertexts, respectively. It can be realized that the computation time for the other schemes grows at a faster pace than our scheme. This is reasonable because extra pairing operations are required in both decryption phases for those schemes. Also, CPA schemes have less-sized ciphertexts compared to CCA schemes since the latter involves additional elements such as signature, for the validation of ciphertexts.

In the blockchain simulation, the feasibility of our work was tested on a Hyperledger Fabric blockchain, using Ubuntu 16.04.1 operating system. We utilized Java-based application web3.js in generating transactions in JSON payload to peers. Transaction latency, which is the amount of time it takes for a transaction to be completed and recorded, was simulated.
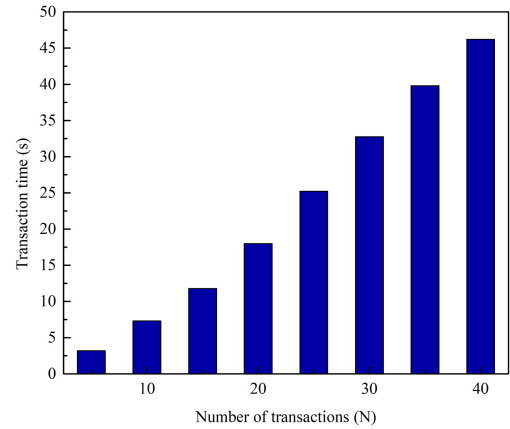

Fig. 9.   Transaction latency.

Latency in blockchain networks is as a result of the overheads on the flow of messages. In our simulation, the average time it takes for a transaction to be processed by the nodes is the time it takes for the node to receive an order and propagate the transaction through the system components. It was evident that there was a steady, linear increase in the latency as the number of transactions increases. This is shown in Fig. 9. The simulation result provides an insight to system optimizations that can be made to improve the efficiency.

VIII. CONCLUSION

The emergence of the IoT has made data sharing one of its most prominent applications. To guarantee data confidentiality, integrity, and privacy, we propose a secure identity-based PRE data-sharing scheme in a cloud computing environment. Secure data sharing is realized with IBPRE technique, which allows the data owners to store their encrypted data in the cloud and share them with legitimate users efficiently. Due to resource constraints, an edge device serves as the proxy to handle the intensive computations. The scheme also incorporates the features of ICN to proficiently deliver cached content, thereby improving the quality of service and making great use of the network bandwidth. Then, we present a blockchain-based system model that allows for flexible authorization on encrypted data. Fine-grained access control is achieved, and it can help data owners achieve privacy preservation in an adequate way. The analysis and results of the proposed model show how efficient our scheme is, compared to existing schemes.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tut.*, vol. 17, no. 4, pp. 2347–2376, Oct./Dec. 2015.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, May 1998, pp. 127–144.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptographic Techn.*, Springer, Aug. 1984, pp. 47–53.

[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, May 2004, pp. 506–522.

[5] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, "Building an encrypted and searchable audit log," in *NDSS*, vol. 4. Citeseer, Feb. 2004, pp. 5–6.

[6] D. Balfanz *et al.*, "Secret handshakes from pairing-based key agreements," in *Proc. IEEE, Symp. Secur. Privacy*, 2003, pp. 180–196.

[7] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, 2004, pp. 207–222.

[8] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," in *Proc. Conf. Appl., Techn., Architectures, Protoc. Comput. Commun.*, Aug. 2007, pp. 181–192.

[9] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing information networking further: From PSIRP to pursuit," in *Proc. Int. Conf. Broadband Commun., Netw. Syst.*, Springer, Oct. 2010, pp. 1–13.

[10] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information," in *Proc. INFOCOM IEEE Conf. Comput. Commun. Workshops*, 2010, pp. 1–6.

[11] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *Proc. IEEE INFOCOM 2004*, vol. 2, 2004, pp. 918–928.

[12] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. 2nd ed. ICN Workshop Inform.-Centric Netw.*, Aug. 2012, pp. 55–60.

[13] Y. Sun *et al.*, "Trace-driven analysis of ICN caching algorithms on video-on-demand workloads," in *Proc. 10th ACM Int. Conf. Emerging Netw. Exp. Technol.*, Dec. 2014, pp. 363–376.

[14] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, vol. 4. Bitcoin.org, 2008. Available: https://bitcoin. org/bitcoin. pdf

[15] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[16] N. Park, "Secure data access control scheme using type-based re-encryption in cloud environment," in *Semantic Methods Knowledge Management and Communications*. Berlin, Germany: Springer, 2011, pp. 319–327.

[17] G. Wang, Q. Liu, J. Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," *Comput. Secur.*, vol. 30, no. 5, pp. 320–331, Jul. 2011.

[18] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Apr. 2011.

[19] P. K. Tysowski and M. A. Hasan, "Hybrid attribute-and re-encryption-based key management for secure and scalable mobile applications in clouds," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 172–186, Nov. 2013.

[20] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Inform. Sci.*, vol. 258, pp. 355–370, Feb. 2014.

[21] J. Han, W. Susilo, and Y. Mu, "Identity-based data storage in cloud computing," *Future Gener. Comput. Syst.*, vol. 29, no. 3, pp. 673–681, Mar. 2013.

[22] H.-Y. Lin, J. Kubiatowicz, and W.-G. Tzeng, "A secure fine-grained access control mechanism for networked storage systems," in *Proc. IEEE 6th Int. Conf. Softw. Secur. Rel.*, Jun. 2012, pp. 225–234.

[23] Y. Zhou *et al.*, "Identity-based proxy re-encryption version 2: Making mobile access easy in cloud," *Future Gener. Comput. Syst.*, vol. 62, pp. 128–139, Sep. 2016.

[24] X. A. Wang, J. Ma, F. Xhafa, M. Zhang, and X. Luo, "Cost-effective secure e-health cloud system using identity based cryptographic techniques," *Future Gener. Comput. Syst.*, vol. 67, pp. 242–254, Feb. 2017.

[25] J. Shao, G. Wei, Y. Ling, and M. Xie, "Identity-based conditional proxy re-encryption," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2011, pp. 1–5.

[26] K. O. B. Obour Agyekum *et al.*, "A secured proxy-based data sharing module in IoT environments using blockchain," *Sensors*, vol. 19, no. 5, Jan. 2019, Art. no. 1235.

[27] G. Zyskind *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 180–184.

[28] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. IFIP Int. Conf. Distributed Appl. Interoperable Syst.*, Springer, Jun. 2017, pp. 206–220.

[29] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G," *IET Commun.*, vol. 12, no. 5, pp. 527–532, Mar. 2018.

[30] M. Singh and S. Kim, "Branch based blockchain technology in intelligent vehicle," *Comput. Netw.*, vol. 145, pp. 219–231, Nov. 2018.

[31] R. S. Da Silva and S. D. Zorzo, "An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf.*, Jan. 2015, pp. 128–133.

[32] B. Li, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for ICN naming scheme," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 2, pp. 194–206, Apr. 2016.

[33] S. Misra *et al.*, "Accconf: An access control framework for leveraging in-network cached data in the ICN-enabled wireless edge," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 5–17, Feb. 2017.

[34] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DACPI: A decentralized access control protocol for information centric networking," in *Proc. IEEE Int. Conf. Commun.*, May 2016, pp. 1–6.

[35] Y. Zhang, R. H. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 15, no. 9, pp. 5099–5108, Jan. 2019.

[36] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, "Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing," *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4519–4528, Jan. 2018.

[37] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 2, pp. 759–767, May 2017.

[38] Z. Wei, J. Li, X. Wang, and C.-Z. Gao, "A lightweight privacy-preserving protocol for VANETs based on secure outsourcing computing," *IEEE Access*, vol. 7, pp. 62785–62793, 2019.

[39] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Apr. 2018.

[40] L. Zhang, Q. Wu, Y. Mu, and J. Zhang, "Privacy-preserving and secure sharing of PHR in the cloud," *J. Med. Syst.*, vol. 40, no. 12, pp. 1–13, Dec. 2016.

[41] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[42] S. Niu, L. Chen, J. Wang, and F. Yu, "Electronic health record sharing scheme with searchable attribute-based encryption on blockchain," *IEEE Access*, vol. 8, pp. 7195–7204, 2019.

[43] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inform. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, Feb. 2006.

[44] R. Pecori, "S-kademlia: A trust and reputation method to mitigate a sybil attack in Kademlia," *Comput. Netw.*, vol. 94, pp. 205–218, Jan. 2016.

[45] A. De Caro and V. Iovino, " JPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun.*, Jun. 2011, pp. 850–855.

[46] E. Barker, L. Chen, S. Keller, A. Roginsky, A. Vassilev, and R. Davis, "Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep. 800-56Ar3, Aug. 2017.
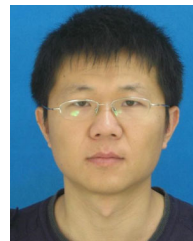
Emmanuel Boateng Sifah received the B.Sc. degree in telecommunications engineering from Ghana Technology University College, Accra, Ghana, in 2014 and the M.Eng. degree in computer science and technology in 2017, from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, where he is currently working toward the Ph.D. degree in computer Science and technology.

His current research interests include blockchain technology and its application and big data security and privacy.

Kwame Opuni-Boachie Obour Agyekum received the B.Sc. degree in telecommunications engineering from Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 2014, and the M.Eng. degree in communication and information engineering in 2017, from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, where he is currently working toward the Ph.D. degree in computer science and technology.

His research interests include blockchain technology and its application, data and network security and privacy, and wireless communication.

Christian Nii Aflah Cobblah received the B.Sc. degree in information science from the University of Ghana, Accra, Ghana, in 2014, and the M.Eng. degree in computer science and technology, in 2019 from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, where he is currently working toward the Ph.D. degree in computer Science.

His current research includes blockchain technology and applications, named data networking, and IoT security and privacy.

Qi Xia received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University Electronic Science and Technology of China (UESTC), Chengdu, China, in 2002, 2006, and 2010, respectively.

She is a Professor with the UESTC. She is currently the Deputy Director of the Cyberspace Security Research Centre, the Executive Director of the Blockchain Research Institute, the Executive Director of the Big Data Sharing and Security Engineering Laboratory of Sichuan province, and a Chief Scientist with YoueData Company Limited. She serves as the Principal Investigator of the National Key Research and Development Program of China in Cyber Security and has overseen the completion of more than 30 high profile projects. She was a Visiting Scholar with the University of Pennsylvania (UPenn), Philadelphia, PA, USA, from 2013 to 2014. She has authored or coauthored more than 40 academic papers. Her research interests include network security technology and its application, big data security, and blockchain technology and its application.

Dr. Xia has won the second place at the National Scientific and Technological Progress Awards in 2012. She a member of the CCF blockchain committee.

Hu Xia received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012.

He was a Visiting Scholar with the University of Minnesota, Twin Cities, MN, USA, from 2010 to 2011. He is currently an Associate Research Fellow with University of Electronic Science and Technology of China.

Jianbin Gao received the Ph.D. degree in computer science from the University Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012.

He was a Visiting Scholar with the University of Pennsylvania, Philadelphia, PA, USA, from 2009 to 2011. He is currently an Associate Professor with UESTC.