BREPubSub: A Secure Publish-Subscribe Model using Blockchain and Re-encryption for IoT Data Sharing Management

Hoang-Anh Pham
Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet St., District 10, Ho Chi Minh City, Vietnam
Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Ho Chi Minh City, Vietnam

***Abstract*—As a result of the incredible growth and diversity of IoT systems and applications over the past several years, an enormous amount of sensing data has been generated, which is critical for developing IoT-based intelligent systems. So far, it has taken a significant amount of time and money to collect sufficient sensing data for these smart systems leading to demands of sharing or exchanging available and valuable data to reduce the time and money spent on the data collection process. However, ensuring the data sharing process's integrity, security, and fairness is fraught with challenges. This paper proposes a Blockchain-based model that supports a secure publish-subscribe protocol for data sharing management by addressing three criteria such as confidentiality, integrity, and availability. In addition, the proposed model adopts a re-encryption technique to optimize shared data storage with multiple users and enhance the security of the data exchange process in a transparent and public environment like Blockchain. We have developed a DApp to demonstrate the feasibility of our design and evaluate its performance.**

Keywords*—Publish-subscribe; blockchain; re-encryption; IoT data sharing

I. Introduction

The Internet of Things (IoT) is a term that refers to the fusion of sensor technologies, big data, artificial intelligence, and network infrastructure. Sensors operate as senses in IoT devices, collecting data about their surrounding environment. These data can be analyzed locally on IoT end-devices or transferred to remote servers for advanced analysis. IoT devices can take actions based on the analytical results via actuators. Currently, IoT is one of the core technologies in the industry 4.0 era and plays a crucial role in many aspects of our lives, from inside to outside of society. IoT is ubiquitous, yet it is not always visible. IoT is renovating physical objects into an ecosystem of information that can be shared between implanted, portable, and even wearable devices. This will enrich our lives with both data and technology. However, building a sufficient database system to manage and storage those IoT data spends a lot of time, money and is inefficient. Poor management might result in the loss of sensitive data such as health status, lifestyle patterns, and device control.

Traditional IoT access management programs are mainly built on popular access management models such as discretionary access control (DAC) [1], attribute-based access control (ABAC) [2], role-based access control (RBAC) [3], or capability-based access control (CapBAC) [4]. It's worth mentioning that in the aforementioned management methods, a centralized entity confirms object access permissions. As a result, it can lead to a single point of failure. To address this issue, distributed CapBAC models have been developed [5][6], in which IoT devices themselves, rather than a centralized entity, authenticate access. However, IoT devices are small, low-power, and have limited computing capacity, thus they can't serve as an access authorization entity.

In the meanwhile, the decentralized architecture enables Blockchain to work efficiently without a central authority. It allows participants to conduct transactions securely, even though the fact that they may not trust one another in a trustless network. Various studies presented systematic investigations and reviews of the potential use cases of Blockchain beyond cryptocurrencies, focusing on how Blockchain may mitigate certain problems in IoT, such as access control [7][8], security and privacy [9][10]. Combining Blockchain and IoT is more applicable since smart contracts can automatically execute agreement terms and conditions after being digitalized, built, and stored on Blockchain. In addition, the execution of smart contracts on Blockchain is precise and transparent to all parties in the ecosystem. Therefore, Blockchain is a viable option for IoT access control management.

Many works have been studied to demonstrate the potential of incorporating Blockchain into IoT under various application scenarios such as shared economy [11], data trading management [12], authorization [13], smart-home [14], healthcare [15], access control [16][17], and IoT cloud [18]. These works inspired us to contribute a study on applying Blockchain to an IoT application scenario, such as access control management. Regarding IoT access management, it can take several forms, such as device ownership management, data disclosure management, and data sharing management. Within the scope of this study, we focus on a data management system that can manage and distribute data between people who own the equipment that generates the data and those who need to access and utilize those IoT data under a publish-subscribe protocol. As above-mentioned, Blockchain combined with IoT will be an efficient solution to solve the problems encountered in centralized architecture when managing IoT data. Additionally, we adopt re-encryption methods to minimize the storage but still secure shared data. All data sharing transactions between system participants are kept transparent and immutable. Consequently, tracking transactions and settling inter-party disputes are simple, and non-repudiation is assured. Some factors like smart contract security and transaction costs are taken into account while evaluating the system. The main contributions of our work are summarized as follows.

- Propose a Blockchain-based model for IoT data sharing management under publish-subscribe fashion.
- Utilize proxy re-encryption to enhance the security of the data exchange process and optimize the storage space for sharing data with multiple users.
- Develop a DApp to demonstrate the feasibility of the proposed model and evaluate the performance in terms of transaction's cost.
- Conduct additional tests to validate security issues related to smart contracts in Blockchain, such as re-entrancy attack.

The rest of this paper is organized as follows. Section 2 summarizes related works to clarify the scope of our study. Then, Section 3 describes the architecture of our proposed system with explanation in detail. The implementation and evaluation will be discussed in Section 4, and Section 5 provides the final concluding remarks and future works.

II. Related Works

The industrial revolution 4.0 brings tremendous global changes, particularly in the Internet of Things, machine learning, and big data. The amount of data created by sensors is enormous and sensitive, posing challenges for access management in the IoT that may be handled in various methods. However, as above-mentioned, our study only focuses on the issue of IoT data storage and sharing management. The proposed system, unlike traditional models, eliminates the role of the third-party service during data exchange between a *Data Owner* DO, who has data to share or sell, and a *Data User* DU, who wishes to purchase the data. Our proposed model employs Blockchain, a decentralized architecture to strengthen the security of the data exchange process, in which all actions are executed via smart contracts, like other existing works.

Both DO and DU are identified using the Blockchain's key system. Everyone on the Blockchain network will own a key pair, including a private key (secret key) and a public key generated from the private key. Each user in the system will be identified by the address associated with the public key. Key pairs are used for encryption during data storage and exchange and for creating IDs for network users. The system will use the Blockchain network environment (e.g., Ethereum) to allow the DO and DU to exchange data. Instead of relying on a third-party intermediary for data sharing, the system performs data exchange using sets of rules and management rules built on Smart contracts.

Unlike conventional methods without Blockchain, data sharing is transparent and public via smart contracts. Every share transaction is recorded on the decentralized network, making changing the transaction history incredibly impossible. At the same time, utilizing the Blockchain network platform's public and private keys combination helps secure data integrity and security, avoiding intermediaries stealing, decrypting, and exploiting data (i.e., a man-in-the-middle attack).

Our proposed solution is comparable to the work given in [19], which largely solves the problem of sharing data that the Aggregator wants to share their collected data and does not have to expose the secret key of subscribers to decrypt the data packet. Each data slot, as well as each subscriber, will have a unique re-encryption key, which will prohibit subscribers from reading all of the data in the system. Furthermore, because both the Aggregator and the Publisher interact with Blockchain, there is no single point of failure, and all actions are recorded in Blockchain as verifiable and trustable transactions. When Aggregator publishes data, they must sign it with their private key, which helps to prove who owns that data packet. Furthermore, data is fragmented into chunks and kept across several storage nodes when utilizing distributed hash-tables, making it challenging to assemble data. However, the work in [19] has yet to present detailed implementation and performance evaluation.

Another similar work [20] primarily comprises a gateway that receives and processes data from IoT devices before re-encrypting and sending it to DU. The authors have successfully developed a PoC system to demonstrate the proposed idea. However, its current model still has some security and performance issues.

- First, since the gateway listens to all events from smart contracts, bottleneck concerns may arise when the number of queries is significant enough. For example, if mulitple DUs try to request data simultaneously, the gateway will get overloaded, resulting in a single-point failure. Although, the authors also attempted to tackle the problem by limiting access to a certain number of times. However, it is not practicable and creates an overhead for managing access in the smart contract.
- Second, when receiving a data request event, the gateway will retrieve data from cloud storage, encrypt it with DO's private and DU's public keys, and then transfer data packets to DU accordingly. To do the re-encryption, the gateway must store DO's private key, which can be readily compromised on the gateway. As a result, if hostile parties attack the gateway, they can obtain the DO's private key and data packets, allowing information to leak.
- Third, the work in [20] did not utilize a proxy re-encryption; instead, as demonstrated in Fig. 1, whenever a DU requests data, DO will decrypt the encrypted data with DO' secret key sk_{Pub} (private key), and then encrypt data again with DU' public key pk_{Sub} before uploading data back to the Storage for sharing to DU who will download shared data and decrypt it with DU' secret key sk_{Sub}. This procedure will be repeated for different DUs, which will consume time (due to multiple encryption and decryption) and storage space (due to duplicates of the same data with different encryption). Leveraging proxy re-encryption can overcome the above issues. As demonstrated in Fig. 2, DO does not need decrypt the data, instead, DO will generate and share the re-encryption key rk_{Sub} for each DU. Then, DU can use this re-encryption key to decrypt the shared data.

III. The Proposed Approach

A. System Design

Fig. 3 illustrates the architecture of our proposed model, in which data can be gathered from IoT end-devices and

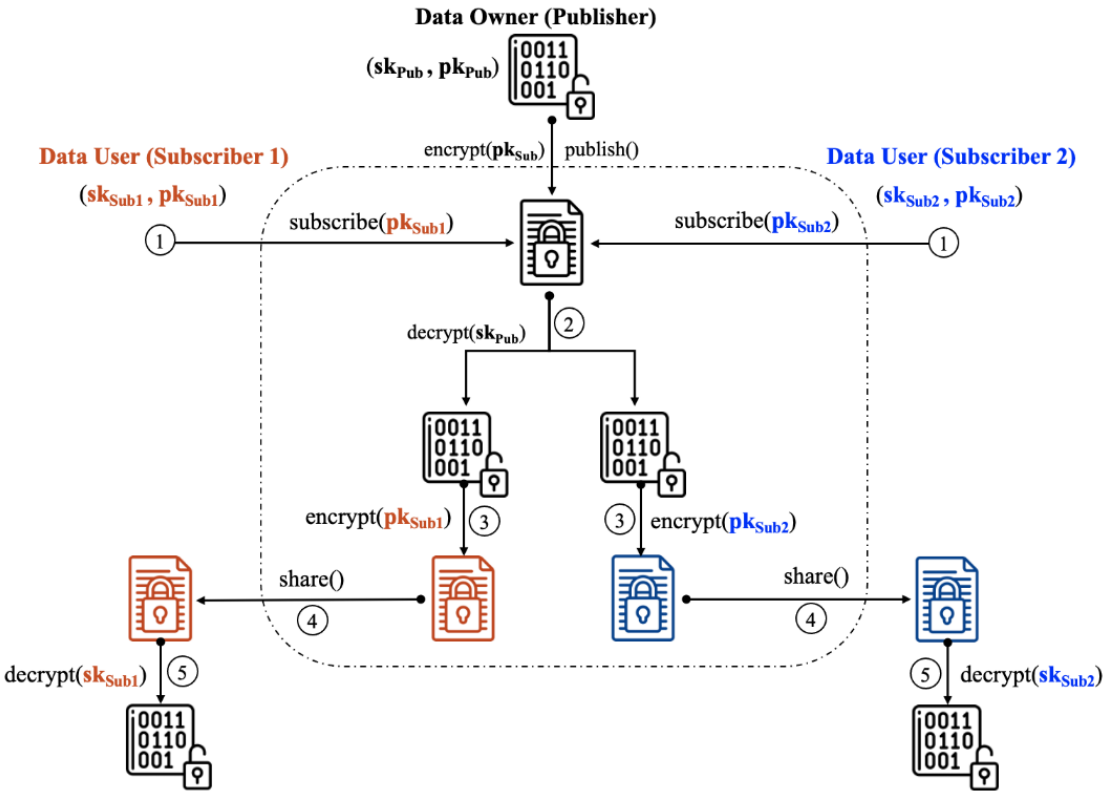

Fig. 1. Repeated encryptions/decryptions and multiple copies of shared data in the sharing process without using re-encryption.

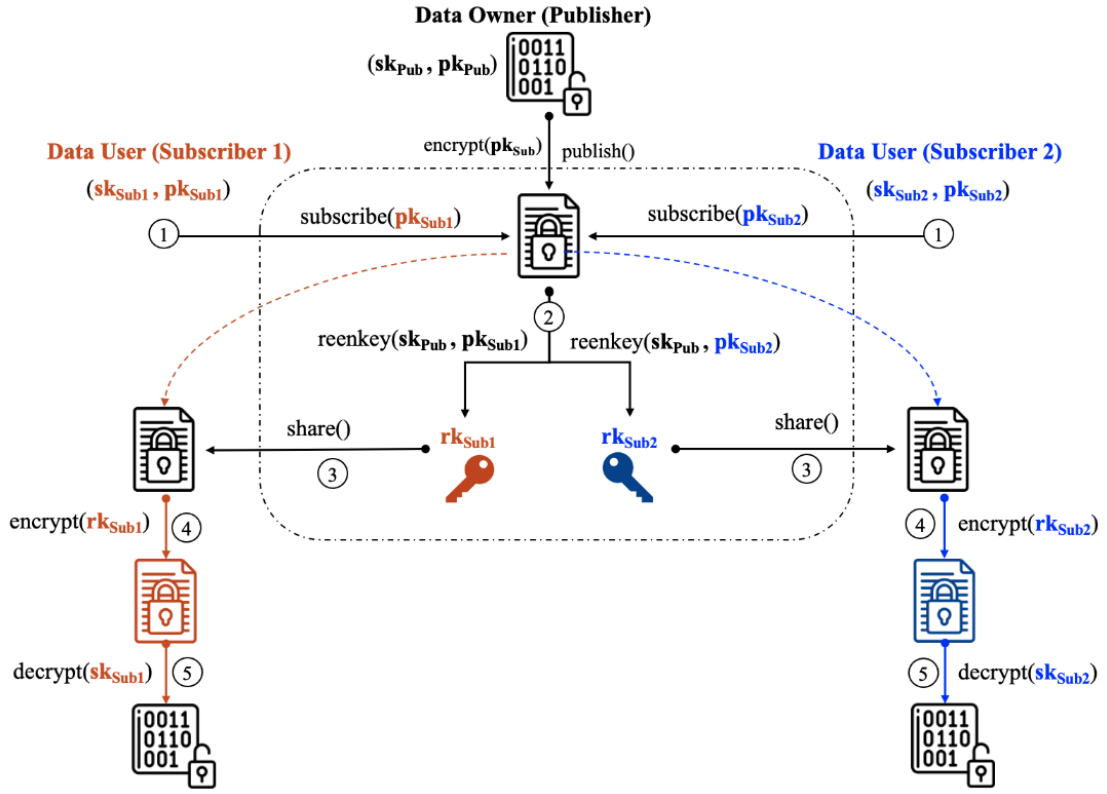

Fig. 2. Optimize shared data storage by using re-encryption.

subsequently transferred to the *Off-chain Storage* via the *Data Management System* (DMS). The proposed model adopts IPFS - a distributed system, as the Off-chain Storage for storing and gaining access to data and files via content identifiers (CID). The detail of the publish-subscribe process between DO and DU can be summarized in eight steps as follows.

1) The DO register IoT devices information to smart contract, including device's name, description, and price per day that will be used as a parameter to calculate the amount of money the subscriber has to pay.
2) When the DU subscribes to a device, he must send the subscription fee to the smart contract that corresponds to the device and the time period for which he wishes to get shared data. The fee will be retained in the smart contract until it is confirmed by the DU after successfully receiving data.
3) The DO will receive an event once a DU subscribes to data from the smart contract.
4) As shown in Fig. 3, DO collects the data from IoT devices and sends these data to the DMS, which will encrypt data by DO's public key before uploading it to IPFS to get the CID. Then, DMS will create the re-encryption key from the corresponding DU's public and DO's secret keys. Finally, DO publishes the CID of encrypted data and the list of re-encryption keys to the smart contract for sharing data with DUs.
5) The DU listens and receives the published data or updates key-event in order to get the CID of encrypted data and list of re-encryption key or new list of re-encryption key from the smart contract.
6) The DU goes to the IPFS to get both the encrypted-data and corresponding re-encryption keys.
7) The DU extracts the corresponding re-encryption key. In order to get the raw data, the DU will perform two steps of decryption sequentially. First, the DU will re-encrypt the encrypted-data with re-encryption key, then continue to decrypt re-encrypted data with DU's private key.
8) If the data is valid, the DU will certify on the smart contract that the data was received accurately. Once the DU confirms successful data reception, the subscription fee pre-paid by the DU will be transferred to the DO via the smart contract.

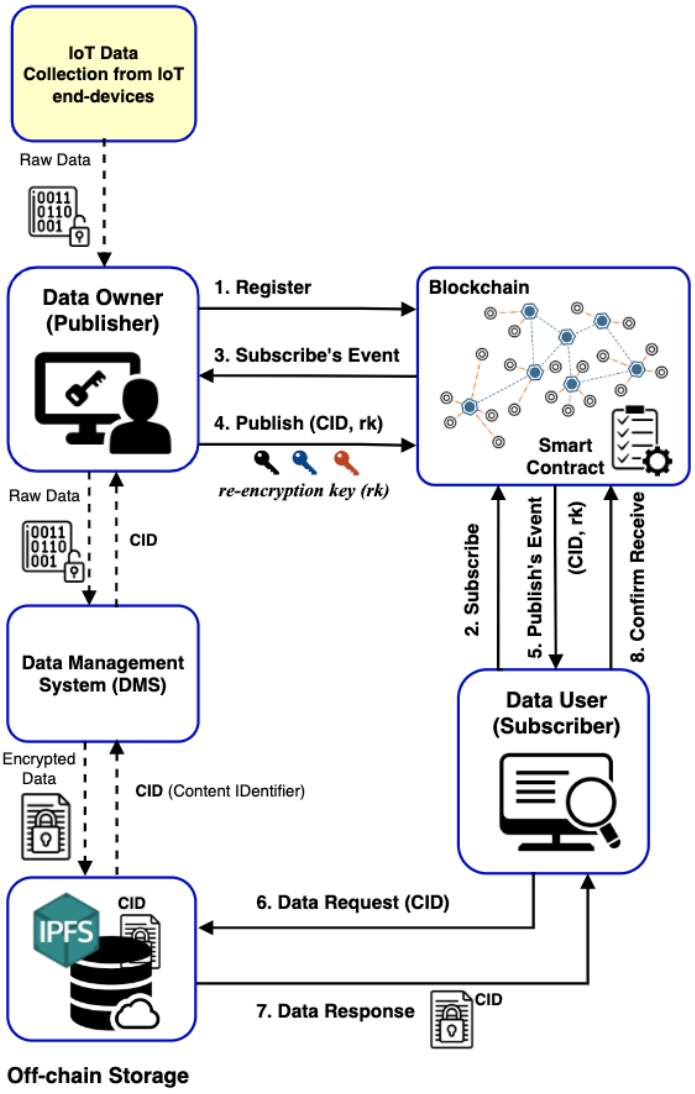

Fig. 3. System architecture of the proposed model.

B. Implementation

We use many technologies, including Blockchain platform, programming languages, development framework, and libraries as summarized in Table I, to develop a prototype to demonstrate and evaluate the proposed model. Meanwhile, Fig. 4 depicts whole sequence diagram of the data sharing management process between DO and DU that correctly executes eight steps described in the proposed model as shown in Fig. 3. In addition, Fig. 5 shows a snapshot of the WebApp that displays all DO's registered devices that are ready to share and be subscribed by users.

Since IoT data collection is not the main objective in our current study, we use virtual devices to generate data for evaluating the sharing management in the proposed model on Blockchain.

TABLE I. IMPLEMENTATION DESCRIPTION

Modules	Implementation descriptions
Blokchain platform	Ethereum
Smart Contract	Solididy
Data Management System (DMS)	Nodejs, Express
Communication between DMS and IPFS	ipfs-http-client
Proxy Re-encryption	recryptjs
DApp testing	Truffle, Ganache

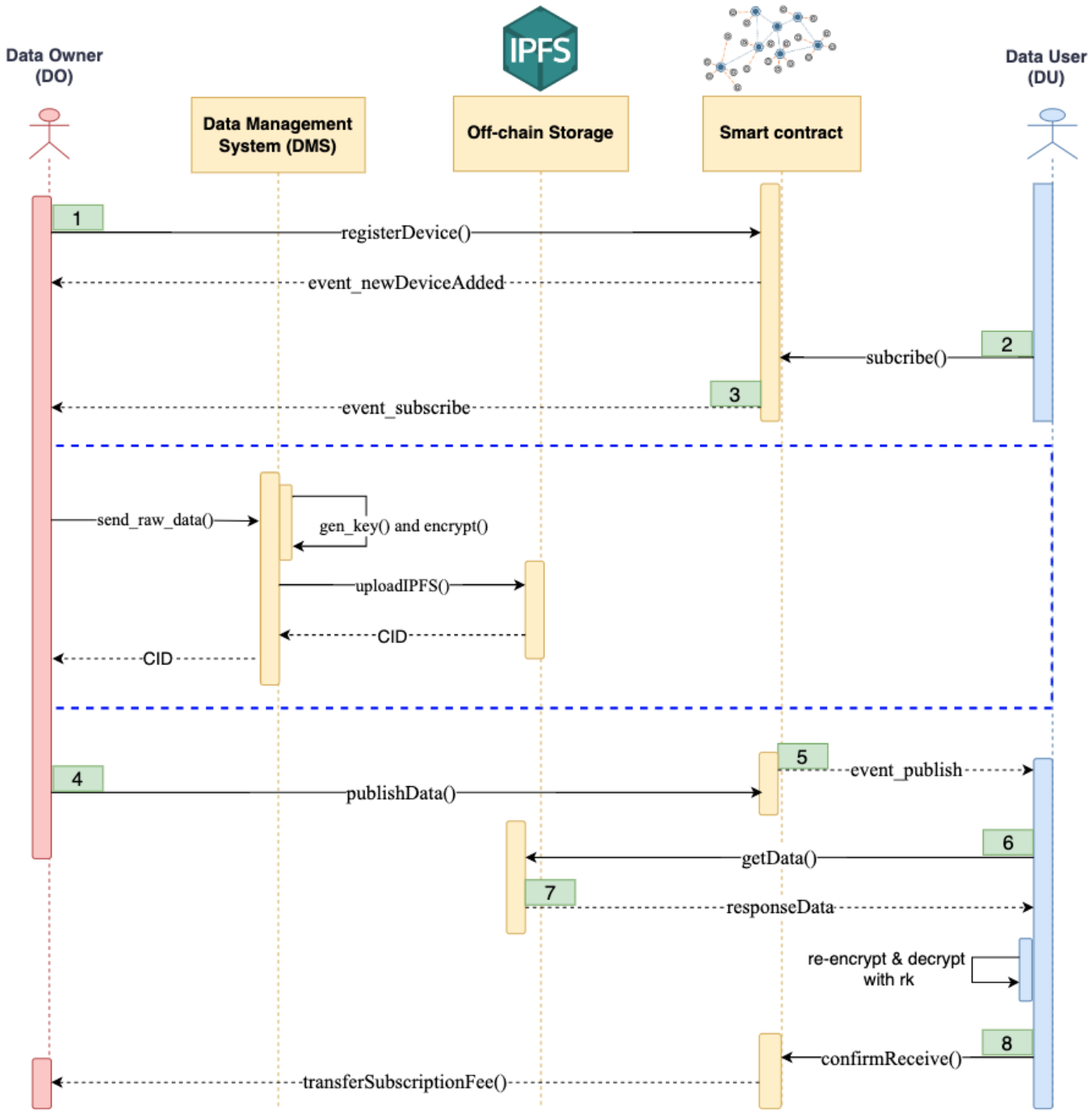

Fig. 4. Sequence diagram of the publish-subscribe procedure in the proposed model.

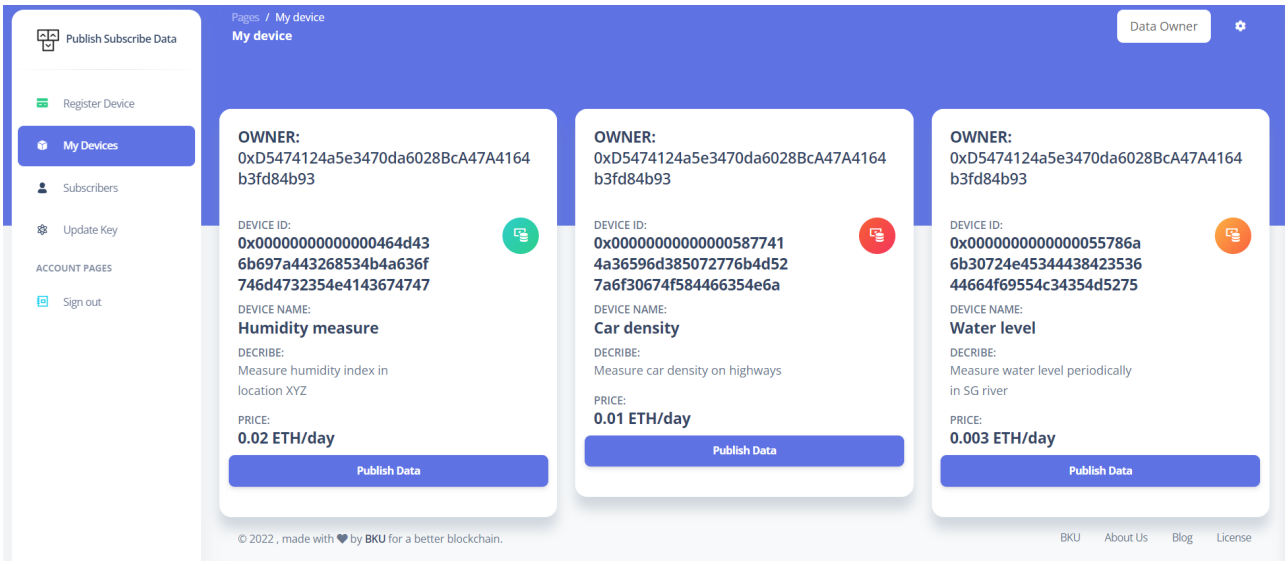

Fig. 5. A snapshot of our WebApp displaying all registered devices that are ready to publish data for sharing.

IV. Experiments and Evaluation

A. Smart Contract Testing

Since all functions related to publish-subscribe process are executed via smart contracts, it is critical to validate the correction of smart contact's execution. We use Truffle, which is a framework for developing dApps on Ethereum to perform testing on smart contracts written in Javascript. In addition, we utilize Ganache, toolkit in Truffle framework, to build a virtual Ethereum environment on a personal workstation for testing smart contracts before real deployment. We develop numerous testing scripts written in JavaScript that might occur in our proposed model and ensure that all scenarios are passed. Table II summarizes descriptions of 10 testing scenarios conducted in our study.

In addition, smart contracts can hold a lot of values, so they can become the subjects for hackers, such as Re-entrancy attack, that happens when a function in a smart contract calling to a function in an external malicious smart contract. Then, the malicious smart contract can call back to the original one. By that way, attackers can drain the fund of the smart contract. As illustrated in Fig. 6, when we call transfer function, smart contract will check balance of the caller. If the condition is satisfied, it begins to transfer the cryptocurrency to receiver before updating the balance. However, the victim smart contract will call the malicious smart control in order to transfer cryptocurrency. Then, the malicious smart contract will call transfer again since the balance is still not updated. Therefore, attackers can drain all the money inside the victim's smart contract. In our implementation, we adopt checks-effects-interactions pattern in smart contracts to prevent this attack.

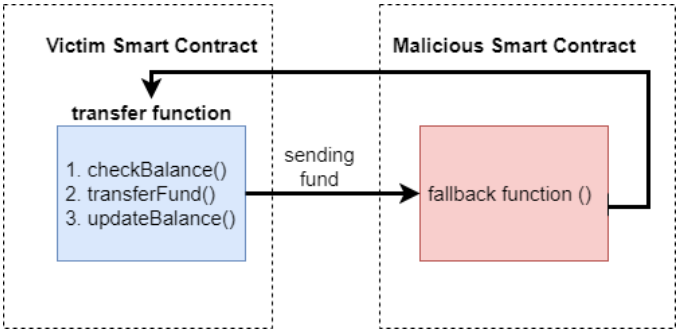

Fig. 6. An illustration of re-entrancy attack.

B. Security Analysis

Data security of the proposed model is discussed according to four main criteria as follows:

- **Confidentiality**: Combining asymmetric encryption and re-encryption based on public and private key pairs allows only the DO and authorized DU to access and read the shared data. All the stored secret keys in the database are encrypted with a secret key phrase. Therefore, all the key is confidential.
- **Integrity**: Since IPFS is a distributed content-addressing database, which means that the CID of a file is like the hash of a file. Each content will generate a unique CID; therefore, with a given CID, the integrity of the data is guaranteed. Furthermore, Blockchain assures that transactions are not tampered with and that the entire transaction process is transparent and traceable.
- **Availability**: The decentralized architecture of Blockchain networks and file storage eliminates single points of failure and protects smart contract services from DDoS attacks, hence enhancing service availability.
- **Immutability:** DU can trust that once DU received the data and re-key CID. No one can revoke the privilege including DO thanks to the combination of SC on blockchain and decentralized database (IPFS). Data once publish will be record permanently and DU

TABLE II. SCENARIOS FOR TESTING SMART CONTRACTS

No.	Scenario Name	Description	Expected Result
1	SC-Deploy	Check the deployment of a smart contract.	Successful deployment
2	Dev-Register	DO registers a new device with non-existing ID.	Be able to retrieve device information via device ID.
3	Dev-Register-E	Check if a device is registered with existing ID.	Raise an error message "duplicate device ID".
4	Subscribe	Check if a DU is able to subscribe a registered device.	Subscribe device successfully.
5	Subscribe-E1	DU can not subscribe a registered device due to payment fee.	Raise an error message "not enough money to subscribe".
6	Subscribe-E2	DU can not subscribe a unregistered device	Raise an error message "non-existing device".
7	Publish	Check if a DO is able to publish data belonging to his registered device.	Data can be published.
8	Publish-E1	DO can not publish data because of data duplication.	Raise an error message "data duplication".
9	Publish-E2	DO can not publish data belonging to not-owned devices.	Raise an error message "non-self-owned device".
10	Confirmation	Check if money is transferred to DO after DU confirmed successful data reception.	Payment is successful.

always can retrieve the data.

C. Gas Fee Estimation

Besides security threats, gas fee optimization is another parameter that should be considered. The gas fee will affect a lot on the topic's practicality. One of the ways that we can reduce the cost fee on each function call without changing the structure of the code is to put the appropriate data location of the variable and function argument. There are three locations to store our variables, including $storage$, $memory$, and $calldata$. The fees for those three locations are different, and we can decide where to store them depending on specific purposes.

- The $storage$ is the most expensive one since the lifetime of the variable is limited to the contract's lifetime.
- The less expensive one is $memory$, which is mostly used for function arguments since it only lasts in the function call, you can read and modify the variable in memory.
- The least one is $calldata$, which behaves mostly like memory but is not able to modify the variable. In our smart contracts, we mostly chose $calldata$ for our complex variable location which will reduce the gas fee for each function call.

Regardless of the content, the length of each data CID is fixed at 46 characters (base58), which means that the CID size is the same for all data. Therefore, the gas price of a smart contract in our model will not depend on the data packet length because we only store the link of data (IPFS CID) on the Blockchain. After performing multiple tests, gas fees of main actions in the proposed model via smart contracts are approximated in Table III.

The cost for each transaction corresponding to each above action via smart contracts will be computed by a product of gas fee and gas price. While the gas fee will not change for the same workload, the gas price will depend on the specific Blockchain network used. Each user can offer their own gas price; the higher the gas price, the more likely their transaction will get picked. The transaction will never be picked if the gas price is too low. Based on the gas fee estimated in Table III, we estimate the transaction's costs in two popular Blockchain networks such as Ethereum and Binance Smart Chain (BSC), as presented in Tables IV and V, respectively.

TABLE III. GAS FEE ESTIMATION FOR EACH ACTION VIA SMART CONTRACTS IN OUR PROPOSED MODEL

Action	Gas
Deploy smart contract	2606585
Register devices (data collectors)	174668
Publish data	216969
Subscribe devices (data)	70763
Confirm (two data packets)	58359

TABLE IV. ESTIMATED COSTS IN ETHEREUM

Action	Gas	Cost (ETH)	Cost (USD)
Deploy smart contract	2606585	0.0417	69.89
Register devices (data collectors)	174668	0.0028	4.69
Publish data	216969	0.0034	5.70
Subscribe devices (data)	70763	0.0011	1.84
Confirm (two data packets)	58359	0.0009	1.51

**The gas price is 16Gwei and ETH price is 1,676$ collected on 14-March-2023 from https://etherscan.io/gastracker*

TABLE V. ESTIMATED COSTS IN BINANCE SMART CHAIN (BSC)

Action	Gas	Cost (BSC)	Cost (USD)
Deploy smart contract	2606585	0.0013	3.98
Register devices (data collectors)	174668	0.0009	0.27
Publish data	216969	0.0011	0.33
Subscribe devices (data)	70763	0.0004	0.11
Confirm (two data packets)	58359	0.0003	0.09

**The gas price is 5Gwei and BSC price is 305$ collected on 14-March-2023 from https://bscscan.com/gastracker*

V. CONCLUSION

We presented a Blockchain-based method to resolve a critical issue in today's IoT, data sharing management. The

proposed model allows Data Owners to securely share their data with Data Users in a publish-subscribe fashion via smart contracts on Blockchain. Like other existing works, adopting smart contracts on a decentralized architecture enables a trustless data sharing mechanism without a third party, increasing service availability. Moreover, the proposed method also inherits the characteristics of Blockchain in enhancing data security in terms of confidentiality, integrity, and availability. However, Blockchain and smart contracts still have security threats themselves. In our implementation, we utilized the checks-effects-interactions pattern to prevent the re-entrancy attack on smart contracts. Besides, we also conducted testing scenarios to guarantee the correct execution of smart contracts before actual deployment.

Compared to previous works, the proposed model employs a proxy re-encryption to minimize shared data storage and enhance data security and privacy in a transparent environment like Blockchain. In addition, we successfully developed a DApp to demonstrate and evaluate how the proposed model works.

ACKNOWLEDGMENT

The authors acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study. The authors also sincerely thank Mr. Long P. Duong and Mr. Phuc M. Nguyen for their assistance in implementing and conducting experiments.

REFERENCES

[1] Moffett, J., Sloman, M., and Twidle, K. *Specifying discretionary access control policy for distributed systems*. Computer Communications, vol. 13, no. 9, pp. 571-580, 1990. DOI:10.1016/0140-3664(90)90008-5.

[2] Vincent C. Hu, D. Richard Kuhn, David F. Ferraiolo, and Jeffrey Voas, *Attribute-Based Access Control*, IEEE Computer, vol. 48, no. 2, pp. 85–88, Feb. 2015, DOI:10.1109/MC.2015.33.

[3] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, *Role-Based Access Control Models*, IEEE Computer, vol. 29, no. 2, pp. 38-47, Feb. 1996, DOI:10.1109/2.485845.

[4] Domenico Rotondi and Salvatore Piccione, *Managing Access Control for Things: a Capability Based Approach*, in: Proceedings of Workshop on Security Tools and Techniques for Internet of Things (SeTTIT), 2012, DOI:10.4108/icst.bodynets.2012.250234.

[5] Ramos, J.L., Jara, A.J., Marín, L., and Gómez-Skarmeta, A.F., *Distributed Capability-based Access Control for the Internet of Things*, J. Internet Serv. Inf. Secur., vol. 3, no. 34, pp. 1-16, 1993. DOI:10.22667/JISIS.2013.11.31.001.

[6] Shorouq Alansari, Federica Paci, and Vladimiro Sassone, *A Distributed Access Control System for Cloud Federations*, in: Proceedings of IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2131–2136, 2017. DOI:10.1109/ICDCS.2017.241.

[7] M. Conoscenti, A. Vetr'o, and J. C. De Martin, *Blockchain for the Internet of Things: A Systematic Literature Review*, in: Proceedings of the IEEE/ACS 13th International Conference of Computer Systems and Applications, pp. 1–6, 2016. DOI:10.1109/AICCSA.2016.7945805.

[8] S. Pal, A. Dorri, and R. Jurdak, *Blockchain for IoT access control: Recent trends and future research directions*. Journal of Network and Computer Applications, Volume 203, 103371, 2022. DOI: 10.1016/j.jnca.2022.103371.

[9] Khan, M.A., and Salah, K. *IoT security: Review, Blockchain Solutions, and Open Challenges*. Future Gener. Comput. Syst., vol. 82, pp. 395–411, 2018. DOI:10.1016/j.future.2017.11.022.

[10] P. Patil, M. Sangeetha, and V. Bhaskar, *Blockchain for IoT Access Control, Security and Privacy: A Review*. Wireless Personal Communications, vol. 117, pp. 1815–1834, 2021. DOI:10.1007/s11277-020-07947-2.

[11] Huckle, S., Bhattacharya, R., White, M., Beloff, N., *Internet of Things, Blockchain and Shared Economy Applications*, Procedia Comput. Sci. 98(C), pp. 461–466, 2016. DOI:10.1016/J.PROCS.2016.09.074.

[12] D. Worner and T. von Bomhard, *When your sensor earns money: Exchanging data for cash with bitcoin*, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 295–298, 2014. DOI:10.1145/2638728.2638786.

[13] Otto Julio Ahlert Pinno, Andre Ricardo Abed Gregio, and Luis CE De Bona. *Controlchain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT*, in: Proceedings of IEEE Global Communications Conference, pp. 1–6, 2017. DOI:10.1109/GLOCOM.2017.8254521.

[14] M. J. Baucas, S. A. Gadsden and P. Spachos, *IoT-Based Smart Home Device Monitor Using Private Blockchain Technology and Localization*, IEEE Networking Letters, vol. 3, no. 2, pp. 52-55, June 2021, DOI:10.1109/LNET.2021.3070270.

[15] M. R. Bataineh, W. Mardini, Y. M. Khamayseh and M. M. B. Yassein, *Novel and Secure Blockchain Framework for Health Applications in IoT*, IEEE Access, vol. 10, pp. 14914-14926, 2022, DOI:10.1109/ACCESS.2022.3147795.

[16] Sara Rouhani and Ralph Deters, *Blockchain based access control systems: State of the art and challenges*. in: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, pp. 423–428, 2019.

[17] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao and K. Yu, *AuthPrivacyChain: A Blockchain-Based Access Control Framework With Privacy Protection in Cloud*, IEEE Access, vol. 8, pp. 70604-70615, 2020. DOI:10.1109/ACCESS.2020.2985762.

[18] Lihua Song et al., *A Novel Access Control for Internet of Things Based on Blockchain Smart Contract*, in: Proceedings of IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Vol. 5, pp. 111–117, 2021. DOI:10.1109/IAEAC50856.2021.9390662.

[19] Nguyen, T.D.T., Pham, HA., and Thai, M.T., *Leveraging Blockchain to Enhance Data Privacy in IoT-Based Applications*, in: Proceedings of International Conference on Computational Social Networks (CSoNet), LNTCS 11280, pp. 211-221, 2018. DOI:10.1007/978-3-030-04648-4_18.

[20] Pham, HA, Le, T.-K, Pham, T.N.M, Nguyen, H.Q.T, and Le, T.V. *Enhanced Security of IoT Data Sharing Management by Smart Contracts and Blockchain*, in: Proceeding of 19th International Symposium on Communications and Information Technologies (ISCIT), pp. 398-403, 2019. DOI:10.1109/ISCIT.2019.8905219.