A Light and Secure Healthcare Blockchain for IoT Medical Devices

Gautam Srivastava*†, Jorge Crichigno‡, and Shalini Dhar§

* Department of Mathematics and Computer Science, Brandon University, Brandon, Canada
† Research Center for Interneural Computing, China Medical University, Taichung, Taiwan, Republic of China
‡ Integrated Information Technology, University of South Carolina, Columbia, U.S.A.
§ Department of Electronics and Communication, University of Allahabad, Allahabad, India

***Abstract*—This paper deals with the incorporation of Blockchain technology in the security of Internet of Things(IoT) based on Remote Patient monitoring systems. The paper presents the benefits and also practical obstacles of blockchain-based security approaches in remote patient monitoring using IoT devices. Furthermore, the paper evaluates various potentially suitable cryptographic technologies for deployment in IoT.**

***Keywords* —Blockchain, Internet of Things, Smart Contract, Information and Network Security, Privacy, Key Management, Authentication**

I. Introduction

Many countries are suffering from a dramatic increase in the number of patients, and it is becoming more difficult for patients to access primary doctors or caregivers. In recent years, the rise of IoT and wearable devices has improved the patient quality of care by remote patient monitoring [1]. It also allows physicians to treat more patients. Remote patient monitoring (RPM) provides monitoring and care of patients outside of the conventional clinical setting (in the home as an example). The main component of a RPM system could be, a specially designed monitoring device to monitor and transmit health data to smart contracts, a smartphone with internet connectivity and a RPM application [2]. Wearable devices and IoT play an important role in RPM and in the current push to develop Smart Cities. Wearable devices collect patient health data and transfer them to hospitals or medical institutions to facilitate health monitoring, disease diagnosis, and treatment. In doing so, we see a Big Data situation develop through all the patient data being analyzed and transferred [3]. To handle such patient data with other institutions, such infrastructure demands secure data sharing. The solution for data privacy and security in IoT scenarios may very well be hidden in blockchain technology [4]. Initially proposed by Satoshi Nakamoto in [5], blockchain technology provides the robustness against failure and data exposure. The miners (responsible for creating blocks) constantly try to solve cryptographic puzzles (named Proof of Work (PoW)) in the form of a hash computation. The process of adding a new block to the blockchain is called *mining*. However, adopting blockchain in the context of IoT is not straightforward. There are several problems currently noticeable in blockchain includingbut not limited to:

- high computational power to solve PoW
- low scalability
- long latency for transaction confirmation [6], [7].

We propose a novel model of blockchain and eliminate the concept of PoW to make it suitable for IoT devices. Our model relies on the distributed nature and other additional security properties to the network. In the next sections, we will discuss in detail the drawbacks in current blockchain models for IoT and our proposed solutions and implementation.

II. Drawbacks and security issues

The main concern in RPM systems is the secure and efficient transmission of medical data. The inability to delete or change information from blocks makes blockchain technology the best technology for healthcare systems and could prevent these issues [8]. But Blockchain technology in its original form is not a long-term solution. It has limitations that when connected to IoT scenarios become very evident. In this section, we discuss the challenges for applying blockchain to IoT and explain how to solve these problems in our model.

A. System Requirements and Solutions

1) **Decentralization:** To ensure robustness and scalability and to eliminate many-to-one traffic flows we need a decentralized system. Using such decentralized systems information delay problems that are seen on occasion with Blockchain. In our model, we are using an *overlay* decentralized network [9].
2) **Authentication and Security of data:** User's computers or cloud services store data that could be modified or lost while transmitting itc̃iteal2017medibchain. The preservation of such incorrect tampered data increases the burden to the system and can also cause issues to the patient using RPM. Therefore to ensure that data is not modified we use a *digital signature* scheme. On the receiver side, data is verified with the user's digital signature and if received correctly, it sends a receipt of data to the patient.
3) **Scalability:** Solving "Proof of Work" (PoW) is computationally intensive, however, IoT devices are very resource restricted [10]. The IoT network in most cases

978-1-7281-0319-8/19/$31.00 ©2019 IEEE

contains a large number of nodes and Blockchain Technology scales poorly to large networks as the number of nodes in the network increases. We eliminate the concept of PoW in our overlay network and also divide our overlay network into several clusters instead of a single chain of blocks. Therefore a single blockchain is not responsible for all nodes in the network, instead nodes are split into several clusters.

4) **Data Storage:** Storing IoT big data over Blockchain technology is not practical and therefore we propose the use of cloud servers to store encrypted data blocks. The data can be interpreted as secure over the cloud due to additional cryptographic security present including digital signatures and high standard encryption [11].
5) **Anonymity of users:** Medical data of a patient may contain sensitive information, and therefore data must be anonymized over the network. For anonymity, we are using a Ring structure along with digital signatures. *Ring signatures* allow a signer to sign data anonymously. That is to say, that the signature is mixed with other groups (named ring), and no one (except the actual signer) knows which member signed the message [12].
6) **Security of data:** To save the data from hackers, we are using a double encryption scheme. We encrypt the data using lightweight *ARX algorithms* and then encrypt the data again using the public key of the receiver [13]. Also, we are using *Diffie-Hellman key exchange* technique to transfer the public keys and therefore to get the keys is almost impossible for an attacker [14].

III. Our Proposed System

Our system consists of five parts:

- Overlay network
- Cloud storage
- Healthcare providers
- Smart contracts
- Patient equipped with IoT devices.

1) **Overlay network:** An overlay is a peer to peer network that is based on a distributed architecture. The nodes connected to the network could be a computer, smartphone, tablet or any other IoT device as well. To increase network scalability and avoid network delay, we group the nodes in the form of many clusters. Each cluster has one Cluster Head. Each cluster head has a unique public key which is shared with other clusters using the Diffie-Hellman algorithm. The cluster head also contains the information about the public keys of nodes which are allowed to access data from the network. These networks only contain the hash of the blocks (not the original data which is stored in the cloud). Each block also contains the hash of the previous block, and therefore we can treat it similar to the blockchain.
2) **Cloud Storage:** Instead of saving the IoT healthcare data over blockchain directly, we use cloud storage servers to save the patient data. The cloud storage groups user's data in identical blocks associated with a unique block number. Cloud storage is connected to overlay networks, once the data stored in a block, the block is encrypted using the shared public key of the user, and cloud server sends the hash of the data blocks to the overlay network.
3) **Healthcare providers and Patients:** Healthcare providers are appointed by insurance companies or by patients to perform medical tests or to provide medical treatments. Healthcare services provide treatment to patients once they receive an alert from the network. Patients themselves are the owners of their personal data and responsible for granting, denying or revoking data access from any other parties, such as insurance company or health care providers [15].
4) **Smart contracts:** Smart contracts allow the creation of agreements in any IoT devices which is executed when the given conditions in the contract are met [16]. Consider we set the condition for the highest and lowest level of patient blood pressure. Once readings are received from the wearable device that do not follow the indicated range, the smart contract will send an alert message to the authorized person or healthcare provider and also store the abnormal data into the cloud so that healthcare providers can receive the patient blood pressure readings as well when needed in real time.

IV. Cryptographic techniques used in the model

Instead of only one type of encryption technique, we use both encryption schemes, Symmetric and Asymmetric for different purposes. Symmetric algorithms (Private key encryption) use the same key for both encryption of plaintext and decryption of ciphertext, whereas asymmetric algorithms (Public key encryption) use different keys for encryption of plaintext and decryption of ciphertext. We use the variable name k_{sym} for the private key or symmetric key in our algorithms, and the same key will be used for encryption and decryption on both side of the transmission.

An asymmetric encryption sender will have one key pair (sk_{priv}, sk_{pub}), and receiver will have another key pair (rk_{priv}, rk_{pub}). Data can be encrypted using receivers public key rk_{pub} and can be decrypted using her private key rk_{priv}. Generally, we use abbreviation plaintext (P) for the normal data file and ciphertext (C) for the encrypted data file.

A. ARX Encryption Algorithm:

In our model, we are using a particular branch of the Symmetric key, called ARX algorithms to encrypt the data for Blockchain. These algorithms are made of the simple operations Addition, Rotation and XOR and support lightweight encryption for small devices. One example of the latest good ARX cipher is SPECK [17], [18], designed by National Security Agency (NSA), US in June 2013. SPECK is a family of lightweight block ciphers with the Feistel-like structure in which each block is divided into two branches, and both

branches are modified at every round. Each block size is divided into two parts, left half and right half.

B. Digital Signature:

We add a digital signature to the data for authentication purposes. Digital signatures are the public-key primitives of message authentication. Each user has a public-private key pair. Generally, the key pairs used for signing/verifying and the key pairs used for encryption/decryption are different. In our case sender will have one key pair $(sk_{s_{priv}}, sk_{s_{pub}})$, and receiver will have another key pair $(rk_{s_{priv}}, rk_{s_{pub}})$. The senders private key $sk_{s_{priv}}$ is used to sign the data, and the key is referred to the signature key while senders public key $sk_{s_{pub}}$ is used for verification on the receiver side of the transmission. Signer feeds the data or plaintext into the *Hash Function* and generates the hash value $hash_p$. Hash value $hash_p$ of plaintext and signature key $sk_{s_{priv}}$ are then fed to the signature algorithm and sent along with the encrypted data. During the verification process, the verifier generates the hash value $hash_r$ of the received data from the same hash function. Using the Verification algorithm and signers public key, he also extracts the original hash value $hash_p$ of plaintext and if the value of $hash_p$ and $hash_r$ are same then data is verified and not changed during the transmission process.

C. Ring Signature:

We use *Ring signature* technology, which allows a signer to sign data in an anonymous way. The signature is mixed with other groups (named ring) and no one (except actual signer) knows which member signed the message. Ring Signature was originally proposed by Rivest in 2001 [19]. A user desiring to mix his transaction sends a request to the Blockchain network. The request comprises of the public key $sk_{s_{pub}}$. After receiving the request the network sends back a certain amount of public keys $(sk1_{s_{pub}}, sk2_{s_{pub}}, sk3_{s_{pub}}, sk4_{s_{pub}}|$ which are collected from other users $(u_1, u_2...u_N)$ who also applied for mixing service, including $sk_{s_{pub}}$. Using ring signatures in our model we can get two important security properties. They are *Signers Anonymity* and *Signature Correctness*.

In our proposed system, we need to transfer the public key through the network. To make data more secure, we also share the public key secretly. To share the public key $sk_{s_{pub}}$ safely along the network we are using the *Diffie-Hellman key exchange technique*.

V. Algorithms

In our encryption Algorithm 1, we encrypt the $data\_file$ by using the symmetric key k_{sym} and produce a ciphertext file C. After encryption, we use double encryption technique and encrypt the key k_{sym} by using the public key cryptography. We use the receivers public key rk_{pub} to encrypt the symmetric key k_{sym} and send the encrypted key along with the ciphertext C. We denote the encrypted symmetric key with C_k.

Algorithm 1 Data Encryption

1: **function** ENCRYPTION ($data\_file$)
2: **if** user confirm data preservation over blockchain **then**
3: Generate a symmetric key k_{sym}
4: $C \leftarrow Encrypt_{sym}\ (data\_file, k_{sym})$
5: $C_k \leftarrow Encrypt_{asym}\ (k_{sym}, rk_{pub})$
6: **else**
7: Do nothing
8: **end if**
9: **end function**

For the digital signature senders can use two keys $(sk_{s_{pub}}, sk_{s_{priv}})$ that is different from the encryption/decryption keys. To add the digital signature, the sender first passes the data file to the *Hash Function* and creates the hash value $hash_p$ of the data. Then he signs the data using his private key $sk_{s_{priv}}$ by passing the value of the private key and hash value $hash_p$ to the Signature Algorithm. The signers public key $sk_{s_{pub}}$ can be used to verify data on the receiver side. To apply the Anonymity of the patient or user, we add ring signature in our Algorithm 2.

Algorithm 2 Ring Signature and Public Key Sharing

1: **function** SIGNATURE ($data\_file$)
2: **if** user chose anonymity over blockchain **then**
3: Generate a asymmetric public-private key pair $(sk_{s_{pub}}, sk_{s_{priv}})$
4: $hash_p \leftarrow$ calculate hash of the $data\_file$
5: Create the Digital Signature using $hash_p$ and signers private key $sk_{s_{priv}}$
6: Share the public key $sk_{s_{pub}}$ to the receiver using Diffie-Hellman key exchange
7: Mix the signature with other network group to form a ring
8: **end if**
9: **end function**

The user will ask the network for other accounts who also want to add ring signatures to their transactions. The network will then provide them with a set of users who also wish to apply ring signatures. The sender's transactions are then mixed with other users transactions and sent over the network. No one will be able to identify the original signer of the ring group. The process is described in the block diagram (Figure 1) of our model.

To decrypt the ciphertext data C (Algorithm 3), we need the symmetric key k_{sym}. The symmetric key was encrypted using the public key rk_{pub} of the receiver, and therefore receivers private key rk_{priv} can only decrypt the symmetric key. We first decrypt the C_k using the private key rk_{priv} of the receiver and get the original symmetric key k_{sym}. We apply the key to the ciphertext C and get the original plaintext or data file.

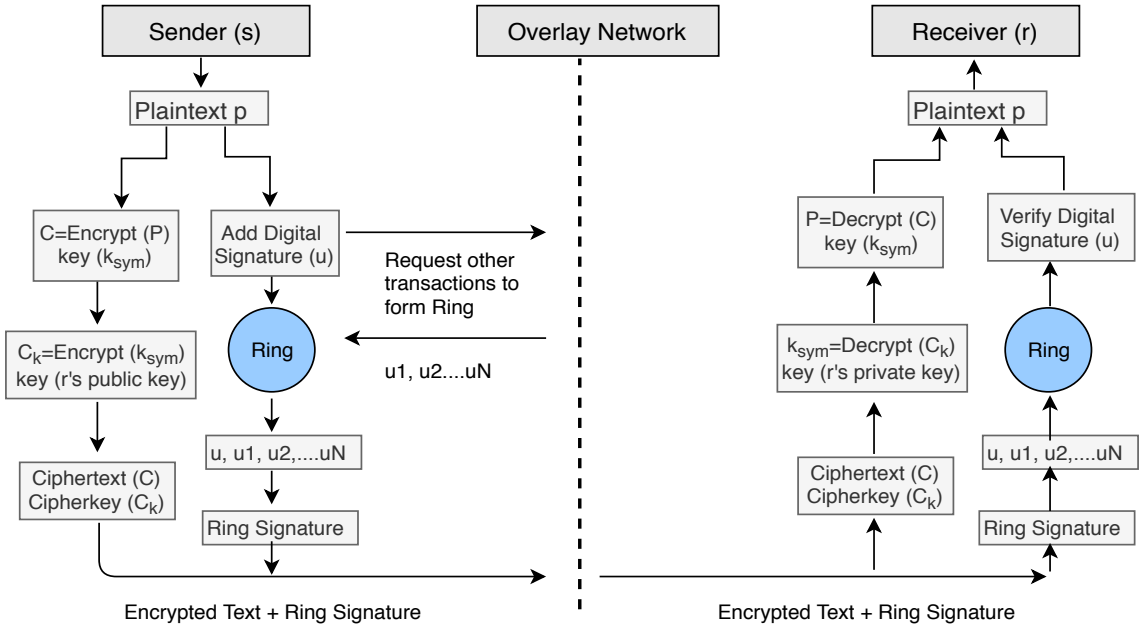

Fig. 1: Block Diagram of Model

Algorithm 3 Data Decryption

Input: Encrypted file C, Encrypted symmetric key (C_k)
Output: Decrypted $data\_file$
function DECRYPTION ($C, C_k, rk_{priv}, k_{sym}$)
    $k_{sym} \leftarrow Decrypt_{asym}\ (C_k, rk_{priv})$
    $data\_file \leftarrow Decrypt_{sym}\ (C, k_{sym})$
end function

During the verification process (Algorithm 4), a Verifier generates the hash value $hash_c$ of received data (ciphertext) using the same hash function. Also, the verifier feeds the digital signature and the verification key into the verification algorithm and extracts the hash value $hash_p$ of original data (plaintext). If both hash values are equal, it means data file is not modified during transfer between sender and receiver.

Algorithm 4 Signature Verification

Input: Encrypted file C, Signers Public key ($sk_{s_{pub}}$)
function VERIFICATION (C, $sk_{s_{pub}}$)
    $hash_c \leftarrow$ calculate hash of the received encrypted data file C to be verified
    Using Public key $sk_{s_{pub}}$ of signer, extract $hash_p$ of senders file
    if $hash_c$ = $hash_p$ **then**
        return C
    else
        return "Signature incorrect"
    end if
end function

VI. Model Implementation

In our system, the patient is equipped with wearable devices such as a blood pressure monitor, insulin pump, or other known devices which are constantly evolving in today's medical world. The health information is sent to smart devices such as a smartphone or tablet for the formatting and aggregation by the application. Once complete, the formatted information is sent to the relevant smart contract for full analysis along with the threshold value. The threshold value decides whether the health reading is NORMAL as per standard readings or not. If the health reading is abnormal, then the smart contract will create an event and send an alert to the overlay network and to the patient. Also, it stores the abnormal readings to cloud servers and cloud server transfer the hash of the stored data to the overlay network. When health data is transferred to the cloud server, the sender adds a digital signature to the data. Overlay network then sends an alert to the health providers. Here, we are not storing the health readings to the overlay network, but we only store the transaction alert to the overlay network.

Health Alert Events should also be anonymous, and privacy preserved to the overlay network. We treat this alert as a transaction of the specific user and apply all advance cryptographic techniques according to the algorithm explained in Section V. Here the entity who is sending the information could be treated as a sender and the entity who is receiving the information could be treated as a receiver. Here we only describe the flow of data in our system and do not describe all the encryption/decryption technical details as we already explained cryptographic techniques in above sections by taking a general model of the sender, receiver and network. An overlay network contains the public key information of all connected nodes and hash indexes of the stored data over the cloud. Once the healthcare provider node gets an alert, he/she access the full health reading of patient for which he/she is authorized over the network.

VII. Future Work

This paper takes an initial look at a blockchain-based model glimpsing into any current IoT-based remote monitoring system using known secure crypotographic tools. Since this project is in its infancy, our main future direction for this work is to implement the system in a testable system to provide some real world security and efficiency guarantees apart from what has already been established for all the individual components used. Through community engagement, we also hope to find partners to help bring some of the novel ideas mentioned in this work to become available to the general public.

VIII. Conclusion

In this paper, we introduced a novel blockchain based IoT model to provide advanced security and privacy properties to the current IoT based remote patient monitoring system. Use of blockchain in IoT based models is not straightforward, and therefore we tried to eliminate many challenges and improved the security of healthcare data. Our model provides reliable data communication over the network and storage over the cloud with more advanced and lightweight cryptographic techniques like ARX encryption scheme. We introduce the concept of Ring Signatures which provides important privacy properties like *Signers Anonymity* and *Signature Correctness*. Also, we used a double encryption scheme to make the symmetric key more secure over the network, and we used the concept *Diffie-Hellman key exchange* technique to our blockchain based network which protects our public key from an intruder.

References

[1] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*. IEEE, 2017, pp. 137–141.

[2] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, p. 130, 2018.

[3] E. Karafiloski and A. Mishev, "Blockchain solutions for big data challenges: A literature review," in *IEEE EUROCON 2017-17th International Conference on Smart Technologies*. IEEE, 2017, pp. 763–768.

[4] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352864818301536

[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[6] G. Srivastava, A. D. Dwivedi, and R. Singh, "Phantom protocol as the new crypto-democracy," in *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, 2018, pp. 499–509.

[7] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for iot," *Sensors (Basel, Switzerland)*, vol. 19, no. 2, January 2019. [Online]. Available: http://europepmc.org/articles/PMC6359727

[8] R. M. Parizi and A. Dehghantanha, "On the understanding of gamification in blockchain systems," in *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE, 2018, pp. 214–219.

[9] R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, and A. Singh, "Empirical vulnerability analysis of automated smart contracts security testing on blockchains," in *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering*, ser. CASCON '18. Riverton, NJ, USA: IBM Corp., 2018, pp. 103–113. [Online]. Available: http://dl.acm.org/citation.cfm?id=3291291.3291303

[10] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for iot," in *Proceedings of the second international conference on Internet-of-Things design and implementation*. ACM, 2017, pp. 173–178.

[11] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, "Towards using blockchain technology for iot data access protection," in *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB)*. IEEE, 2017, pp. 1–5.

[12] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with iot. challenges and opportunities," *Future generation computer systems*, vol. 88, pp. 173–190, 2018.

[13] A. D. Dwivedi and G. Srivastava, "Differential cryptanalysis of round-reduced lea," *IEEE Access*, vol. 6, pp. 79 105–79 113, 2018.

[14] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based iot ecosystem using attribute-based encryption," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2017, pp. 1–6.

[15] Z. Huang, X. Su, Y. Zhang, C. Shi, H. Zhang, and L. Xie, "A decentralized solution for iot data trusted exchange based-on blockchain," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 1180–1184.

[16] R. M. Parizi, Amritraj, and A. Dehghantanha, "Smart contract programming languages on blockchains: An empirical evaluation of usability and security," in *Blockchain – ICBC 2018*, S. Chen, H. Wang, and L.-J. Zhang, Eds. Cham: Springer International Publishing, 2018, pp. 75–91.

[17] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK families of lightweight block ciphers," *IACR Cryptology ePrint Archive*, vol. 2013, p. 404, 2013.

[18] A. D. Dwivedi, P. Morawiecki, and G. Srivastava, "Differential cryptanalysis of round-reduced speck suitable for internet of things devices," *IEEE Access*, vol. 7, pp. 16 476–16 486, 2019.

[19] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, vol. 2248. Springer, 2001, pp. 552–565.