Research on Security for Wireless Body Area Network Data in Cloud Storage Based on Blockchain

Yanfang Gao1,2, Lan Yao1*, Yansong Ma1

1School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning, 110169, China
2Medical IT Department, Neusoft Corporation, Shenyang, Liaoning, 110179, China
*yaolan@mail.neu.edu.cn

Abstract. Cloud computing enables users with limited resources to accomplish complex computation. Meanwhile, the massive human body sensing data generated by Wireless Body Area Network (WBAN) can be stored in cloud servers. However, cloud storage of human body sensing data is challenged by some issues. Users need to ensure that cloud servers can return the results and that they are correct. To address this problem, based on the argument system of commitment scheme, this paper designs a DMP verification algorithm. By compiling the hash of the header node into a constraint set and verifying the accuracy of the data of the intermediate node through a hash list, the algorithm reduces the compiling time of the intermediate node and improves the verification efficiency.

1. Introduction

Wireless Body Area Network (WBAN) is a small human-centered network [1]. WBAN collects physiological information of human body through embedded sensor nodes on skin surface, clothing and around human body. All kinds of information collected by WBAN, such as blood pressure, ECG, body temperature, brain wave and even blood parameters, are regarded as personal sensitive data [2]. These data are used by hospitals, schools, government departments, disease research and other industries. However, with the rapid development of WBAN, WBAN produces a large amount of data, and the storage capacity of sensor nodes is limited. Uploading WBAN data to cloud storage server solves the problem that WBAN cannot store a large amount of data [3-4].

When WBAN data are uploaded to the cloud storage server, the data are shared by multiple users. The data are no longer stored and used locally, but are completely controlled by the cloud server. Therefore, the data stored on the cloud server may face the following serious security challenges. (a) Data may be intercepted and tampered maliciously in the process of uploading to cloud servers. (b) Data of human parameters are usually privacy. When data is stored in the cloud, it is necessary to ensure that data are not visible to cloud storage service providers. Otherwise, cloud storage service providers may maliciously disseminate data, and the data privacy is destroyed. (c) Assuming that cloud storage service providers are untrustworthy, data integrity may be challenged by modifying and deleting data that is used less frequently on cloud servers due to self-interest and other reasons. (d) For unauthorized organizations, their access to data should be avoided. Due to multi-party sharing, different authorized organizations have different requirements for data processing. How to coordinate data processing and avoid data modification by individual organizations without multi-party consent is an important issue currently facing.
Blockchain technology revolutionizes the "Byzantine General Problem", has the security characteristics of irreversible, unforgettable and completely traceable, and achieves a trustless consensus network system [5-6]. Blockchain system generally consists of application layer, contract layer, incentive layer, consensus layer, network layer and data layer [7]. Blockchain technology has the characteristics of distributed storage and consensus mechanism. It has advantages in solving the security problems of access control and integrity verification.

2. Access control method based on blockchain technology
Aiming at the problem of data access control, this paper designs an access control framework based on blockchain technology, as shown in Figure 1. There are five roles in the framework: Data Uploader (DUP), Data Accessor (DAP), Data Access Authorization Center (DAAC), Blockchain Network (BCN), cloud storage service provider (CSSP). BCN is logically a blockchain network composed of set $U=\{U_1, U_2, U_3,\ldots, U_n\}$. Physically, $U_i$ is a user node that has been granted cloud storage service by DAAC. It uses blockchain to store access requests. It has dual identities of DUP and DAP. It can upload and access data. It can generate access requests, control access and verify the results of BCN’s access request control.

![Figure 1. Framework of access control based on BCN.](image-url)

In this framework, the key generation and signature method of each node adopts the digital signature method without trusted center [8]. $H$ is a one-way hash function, $(P, P')$ are two secure large primes, $g$ is the generator of $GF(P)$ with order $P'$, $2^{511}<P<2^{512}$. $Q$ is the prime factor of $P'^{-1}$, $\alpha$ is the generator of $GF(P')$ with order $Q$, $2^{159}<Q<2^{160}$, and the set of BCN user nodes is $U=\{U_1, U_2, U_3,\ldots, U_n\}$. Subsequently, the key of each node is generated. For the user node $U_i$, $i \in [1,n]$, $Id_i$ is the unique identification number $Id_i \in [1,Q-1]$ of $U_i$, $Id_i$ is open to the user node of BCN. The construction of each user node in the set $U$ is $f(x)=(a_0+a_1x+a_2x^2+\ldots+a_{n-2}x^{n-2}+1) \mod Q$. $f(x)$ is polynomial of degree $n(2+1)$, where $0 < a_k < Q$, $k=0, 1, 2, \ldots, n/2+1, a_i$ is kept in secret by $U_i$, the private key is $SK_i = \alpha^{f(0)} = a_0$, the public key is $PK_i = g^{a_i^{f(0)}} \mod P$, and the group public key is $y = g^{a_i^{f(0)}a_i^{f(0)}\cdot\ldots\cdot a_i^{f(0)}} \mod P$. $U_i$ computes $V_{ij}, Y_{ij}$ for other user members $U_j$, $v_{ij} = \alpha^{f(Id_j)} \mod P^*$, $y_{ij} = g^{v_{ij}} \mod P$, $U_i$ publicly sends computed $V_{ij}$ and $Y_{ij}$ to other user nodes of BCN.

When each node receives an access request, it generates a sub-signature. $U_i$ sends an access request $O$ to BCN. $U_i$, the user node of BCN, looks at the access request $O$, generates a random number $d_i$, computes $r_i = a_i^{d_i}$, and sends $r_i$ to BCN to express its consent to the access request. User nodes that agree to access requests constitute set $B=\{U_1, U_2, U_3,\ldots, U_n\}$. If $w > n/2+1$, it means that more than 50% of BCN nodes agree to this access request. The node $U_i$ that agrees to access the request $O$ generates a sub-signature using its own key $SK_i$ and a $\alpha^{f(Id_j)}$ received by each node. The computation of the sub-signature $S_i$ is shown in (1).

$$s_i = \left\{ \left\lfloor \frac{\alpha^{f(0)} \cdot \prod_{j \in B, i \in B} f_j(Id_i) \prod_{j \in B} -Id_i}{ld_i - ld_j} \right\rfloor \right\} * H(o) * a_i^{k_i}$$

(1)

After $U_i$ generates the sub-signature, the random number $k_i$ is selected to satisfy $\text{gcd}(k_i, P^*)=1$, and
the $z_i$ and $s_i$ are computed according to:

$$z_i = g^{k_i} \mod P$$

and

$$s_i' = k_i^{-1} (s_i - \alpha^{f(0)} \mod P').$$

At this time, the message $(O, s_i, r_i, z_i, s_i')$ can be sent to $U_j$. After receives the sub-signatures from each node, $U_j$ computes $(R, S)$, and computes $R$ according to (2) and $S$ according to (3).

$$R = \prod_{i=r}^{\frac{n+2}{2}+1} r_i \mod P'$$

(2)

$$S = \prod_{i=r}^{\frac{n+2}{2}+1} s_i \mod P'$$

(3)

At this time, $U_j$ generates its own digital signature for access request $O$. $U_j$ generates random number $e_j$ in interval $[1, P-1]$, and computes $l_j$, $z_j$ according to:

$$l_j = g^{e_j} \mod P,$$

$$z_j = (\alpha^{f(0)} \mod O' - e_j \cdot l_j) \mod (P-1),$$

where $z_j \in [1, P-2]$, $O' = H(O)$. Subsequently, each BCN nodes can verify the authenticity of group signatures by:

$$g^{s} = y^{\gamma e_{224} + R} \mod P',$$

and if the verification is successful, the access request is stored in its own block.

The consensus mechanism in BCN is implemented by improving Raft algorithm. Firstly, the leader node is selected through the heartbeat mechanism interaction information. Subsequently, the leader node makes consistency judgment by the length of blockchain and the hash value of the last block in the blockchain. If not, block replenishment is performed. Finally, the blockchains of each node are consistent. After leader node make the blockchains of each node in BCN consistent, and when blocks need to be added, BCN nodes need to verify the blocks by using the hash value of blocks. If most of the blocks in BCN are validated successfully, the blocks will be recorded. The Leader node then sends the blocks to the cloud storage service.

In the framework designed in this paper, each node in BCN can vote on the request, control the validity of the request collaboratively, and record the request through the block. The improved Raft algorithm in BCN makes the blockchains of each node consistent, and then verifies the authenticity of the blocks to ensure that the blocks uploaded to the cloud storage service are authentic and credible.

3. Integrity verification based on blockchain technology

3.1. Data Structure Design of Block

WBAN data are uploaded to CSSP for storage. An example of WBAN data storage format is shown in Table 1. Temperature, blood pressure and heartbeat are the data collected by BCN user nodes at a certain time. The vector of data $D$ is expressed as (data number, user number, body temperature, blood pressure, heartbeat).

| data number | user number | body temperature (°C) | blood pressure (mmhg) | Heartbeat (times/Min) |
|-------------|-------------|------------------------|------------------------|-----------------------|
| 160000      | 15110       | 37.5                   | 120                    | 65                    |
| 16001      | 15012       | 36.0                   | 80                     | 80                    |
| 160002      | 15057       | 36.5                   | 110                    | 70                    |

The way CSSP stores WBAN data is to combine an ordered array with a Merkle tree whose data numbers are $M$-spaced between array nodes. Each location of an ordered array corresponds to a Merkle tree. When adding data, it only needs to modify the Merkle tree corresponding to the array node. The data stored in ordered array node is (data number, Merkle tree root node pointer), and M represents the upper limit of the number of leaf nodes stored in the Merkle tree. Merkle tree nodes are divided into internal nodes and leaf nodes. The hash algorithm used by internal nodes is MD5. The hash algorithm used by leaf nodes is homomorphic hash algorithm. The internal node contains information $(R, hash value)$, and the leaf node contains information $(R, hash value, encrypted data D')$. The leaf node $R$ is $(D'$ data number - corresponding ordered array node data number). The $R$ of the internal node is the larger $R$ in the sub-nodes. Homomorphic hash values of leaf nodes are computed
using the homomorphic hash algorithm mentioned in reference.

Data D (data number, user number, blood pressure, body temperature, heartbeat) is encrypted with SHK as data D'(data number, user number, SHK(blood pressure), SHK(body temperature), SHK(heartbeat). SHK uses the data sharing key used by BCN nodes to encrypt data. Data parts of D’ are SHK (blood pressure), SHK (body temperature), SHK (heartbeat), these three data can be expressed by a matrix of 1*m. The data number and user number are mainly used as a unique identifier and are not encrypted, so it is initialized with m=3, which is shown in (4).

\[ \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \]

(4)

Among them, the data part of the modification request X(data number, SHK(blood pressure), SHK(body temperature), SHK(heartbeat)) can be expressed by matrix FX, as shown in (5).

\[ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \]

(5)

When CSSP processes the modification request X, after finding the encrypted data D' according to the data number in X, compute D'+X=D'', and the data parts addition is shown in (6).

\[ \begin{pmatrix} d_1 + x_1 \\ d_2 + x_2 \\ d_3 + x_3 \end{pmatrix} \]

(6)

It can be proved that hK(FD+FX)=hK(FD)*hK(FX), that is the algorithm satisfies multiplication homomorphism, where K is a homomorphic hash key. When M is set to 4 by CSSP, the storage structure is shown in Figure 2.

Figure 2. CSSP storage.

Blockchain is linked by blocks. The block header contains the parent hash - the hash value of the previous block; the timestamp - indicating that the block does exist at a certain time; the root node hash of the Merkle tree - the root node hash value of the Merkle tree in the block after computing; and the current block hash - MD5 (parent hash + timestamp + root node hash of the Merkle tree). When each block is generated, the hash of the previous block is first used as the parent hash, and then the current block hash is computed, which is connected with the previous block through the parent hash to form a chain structure.

Blocks contain Merkle trees - when blocks are added to the blockchain, Merkle trees are generated by access requests stored in the hash table; hash tables - access requests are put into the hash table of blocks after BCN validation; and authentication path arrays - two-dimensional array representation of authentication paths of Merkle leaf nodes. The leaf node of the Merkle tree stores access requests. The initial block is called the creation block, which is the first block in the blockchain of each BCN user node. The Merkle tree is empty, the parent hash value is empty, the time stamp is the time specified by
ICEMCE 2019
IOP Conf. Series: Journal of Physics: Conf. Series 1314 (2019) 012199
doi:10.1088/1742-6596/1314/1/012199

BCN, and the hash value is the result obtained by MD5 algorithm after processing the time stamp. The blockchain structure is shown in Figure 3.

![Blockchain structure](image)

**Figure 3. Blockchain structure**

3.2. **Integrity Verification**

According to the block structure of BCN and the storage structure of WBAN data stored in CSSP, the integrity of WBAN data uploaded to CSSP is verified by using the homomorphic characteristics of homomorphic hashing algorithm. In this paper, because all BCN user nodes store access requests, BCN user nodes can verify the integrity of data without delegating a trusted third party, and support the dynamic operation of data.

3.2.1. **Data Integrity Verification Model**

In the data integrity verification model, the upload node $U_j$ encrypts the data with the key SHK, generates the hash value of the encrypted data with the homomorphic hash key $K$, and the BCN verification node $U_i$ randomly selects the data number to verify the data. CSSP stores the encrypted data and the BCN node $U_k$ returns the authentication path to ensure the reliability of the modification request. The data verification model is shown in Figure 4.

![Integrity verification model](image)

**Figure 4. Integrity verification model**

3.2.2. **Data integrity verification scheme**

The data integrity verification scheme consists of five parts: key generation (keyGen), tag generation (TagBlock), challenge (Challenge), proof generation (ProofGen) and proof verification (Verify).

(a) **Key generation stage**: The key generation algorithm DAAC uses the homomorphic encryption algorithm Paillier to complete the generation of key (PHK, SHK).

(b) **Tag generation stage**: When data is uploaded, the uploading node performs homomorphic hashing operation on the encrypted data $D'$ to generate the tag $h_K(D')$. Cloud storage server stores data, generates tags $PD(PD_1, PD_2, PD_3, ...)$, and $PD$ is the authentication path of Merkle tree for data $D'$ in CSSP storage structure. When BCN user node $U_k$ stores modification request $X$, tags $h_K(X)$, $PX (PX_1, PX_2, PX_3, ...)$ are generated, here $PX$ is the authentication path of modification request $X$ in block structure Merkle tree.

(c) **Challenge stage**: Verification node $U_i$ selects data number $NM$ and sends $NM$ to data upload node $U_j$, $U_k$, CSSP.

(d) **Proof generation stage**: After CSSP receives $NM$, it looks up tag $PD(PD_1, PD_2, PD_3, ...)$, sends the $PD$ back to $U_i$, and after upload node $U_j$ receives $NM$, it looks up the tag $h_K(D')$, sends the tag $h_K(D')$ back to $U_i$. When the BCN user node $U_k$ receives $NM$, it looks up the authentication path $PX$
whose request data number is NM, and sends PX to Ui.

(c) Proof verification stage: Verify whether the label $h_k(X)$ is reliable after the node $U_i$ receives $PX$. Assuming that $PD$ is a three-dimensional vector $(PD_1, PD_2, PD_3)$, $PX_3$ is the root hash of Merkle tree in the block, the correctness of $PX$ is verified by (7).

$$PX_3 = MD5(MD5(h_k(X) + PX_1) + PX_2)$$  \hspace{1cm} (7)

If the above formula holds, it shows that the request tag $h_k(X)$ of $U_i$ lookup is correct, and then verifies the data integrity. Assuming that the $PD$ is a three-dimensional vector $(PD_1, PD_2, PD_3)$, it is a Merkle tree root hash in the CSSP storage structure. The data integrity can be verified by (8).

$$PD_i = MD5(MD5(h_k(X) * h_k(D') + PD_2) + PD_1)$$ \hspace{1cm} (8)

If the above formula holds, it means that the data stored in CSSP is not maliciously tampered with, and the data is integrated.

4. Experimental results and analysis

The framework designed in this paper is mainly to ensure that BCN user nodes can collaboratively control access to cloud storage data and ensure the security of cloud storage data. Next, the access control capability of the framework is tested and analyzed, and the data integrity verification scheme proposed in this paper is evaluated. In this paper, the distributed environment is simulated by Docker container. The data set uses the patient physiological information data set collected by WBAN from a hospital in Beijing which has been desensitized.

4.1. Security Analysis and Experiments of Access Control

(a) Data privacy When each BCN node uploads data to CSSP, it encrypts the data using the shared key SHK. Make sure that CSSP is not visible to data.

(b) Anti-single-point security The framework designed in this paper guarantees that access requests from BCN user nodes are co-controlled by multiple BCN user nodes. Control rights are allocated by threshold signature without trusted center. Access requests can only be processed by CSSP with the consent of most nodes in BCN network.

(c) Accountability mechanism When BCN user nodes make access requests, they need digital signature. After most BCN user nodes agree to the access requests, they need to store the access requests and digital signatures together in blocks. Through digital signature, we can know which user node in BCN network requests access.

(d) Block verification BCN adopts improved Raft algorithm. Leader node maintains the consistency of blockchains of each user node in BCN. If the Leader node forges access requests, the block cannot pass the verification of other user nodes in BCN. The verification fails, and the Leader node automatically abandons the right to account. Block verification guarantees that the Leader nodes in BCN network cannot forge access requests. The foundation of the block verification block of BCN user nodes is that the length of the blockchain of the nodes remains the same. Figure 5 shows the change of the blockchain length of each node in BCN with time after the Leader nodes are selected by BCN. Figure 5 shows that within 100ms after Leader election, the consistency of blockchains of BCN nodes can be guaranteed. The length of blockchain of BCN nodes is the same, which indicates that all nodes participate in access control. The results show that the framework can effectively guarantee the privacy of data and the rationality of access requests. Malicious access requests and unauthorized access requests can not access the data stored in the cloud, and facilitate tracing accountability.
4.2. Data Integrity Verification Experiment

In the experiment of data integrity verification, several nodes are selected and the integrity of multiple data is verified, and the results of integrity verification are counted. Fig. 6 is the result of integrity verification of randomly selected data numbers by BCN user nodes, in which the round node indicates the success of verification and the square node indicates the failure of verification. The experimental results show that the data integrity verification results of different BCN user nodes are consistent, which shows that the integrity verification scheme can support all users to carry out integrity verification.

5. Conclusion

In view of many security problems existing in WBAN data storage in cloud servers, this paper designs an access control framework based on blockchain technology, which restricts the access of multi-users to WBAN-aware data in cloud storage, and then proposes a data integrity verification scheme on the basis of the framework. Finally, the security of the framework is analyzed and tested, and the proposed data integrity verification scheme is evaluated experimentally. The experimental results show the correctness and feasibility of the scheme, which has certain theoretical and practical value.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61602106).

References

[1] Rahman A. F. A., Ahmad R., Ramli S. N. (2014) Forensics readiness for wireless body area network (WBAN) system. In: Proceedings of International Conference on Advanced Communication Technology. Pyeongchang. pp. 177-180.
[2] Kim Y, Lee S. S., Lee S. K. (2016) Coexistence of ZigBee-based WBAN and WiFi for health telemonitoring systems. IEEE Journal of Biomedical & Health Informatics, 20: 222-230.

[3] He D., Zeadally S., Wu L. (2018) Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Systems Journal, 12: 64-73.

[4] Li S., Hong Z., Jie C. (2016) Public auditing scheme for Cloud-Based wireless body area network. In: Proceedings of IEEE/ACM International Conference on Utility & Cloud Computing. Shanghai. pp. 375-381.

[5] Xie H., Wang J. (2016) Study on blockchain technology and its application. Netinfo Security, China. 2016: 192-195.

[6] George P., Ilya S. (2018) Mechanising blockchain consensus. In: Proceedings of 7th ACM SIGPLAN International Conference on Certified Programs and Proofs. Los Angeles. pp. 78-90.

[7] Yuan Y., Wang F. (2016) Blockchain: the state of the art and future trends. Acta Automatica Sinica, China. 42: 481-494.

[8] Krohn M. N., Freedman M. J., Mazieres D. (2004) On-the-fly verification of rateless erasure codes for efficient content distribution. In: Proceedings of IEEE Symposium on Security & Privacy. Berkeley. pp. 226-240.