Block-based Access Control for Blockchain-based Electronic Medical Records (EMRs) Query in eHealth

Xiaoshuai Zhang, Stefan Poslad, Zixiang Ma
School of Electrical Engineering and Computer Science
Queen Mary University of London, London, E1 4NS, United Kingdom
Email: {xiaoshuai.zhang, stefan.poslad, zixiang.ma}@qmul.ac.uk

***Abstract*—In this paper, we propose an access control solution for exchanging Blockchain-based Electronic Medical Records (EMRs) called BBACS (Block-based Access Control Scheme) that includes an access model and an access scheme. Unlike the existing Blockchain-oriented access schemes for EMRs, our access model can omit the agent layer (gateway) in order to authorise users' access with block level granularity, whilst maintaining compatibility with the underlying Blockchain data structure. Furthermore, the authorisation, encryption, and decryption algorithms presented in BBACS dispense with the need to use a public key infrastructure (PKI) and hence cut down the cost of network construction and improve the computational performance. We validated the efficiency (time cost) of local computation and data transmission (over Wi-Fi) for BBACS using a simulation of BBACS against another Blockchain-oriented access control scheme for EMRs called HDG as our baseline. To the best of our knowledge, BBACS is the first Blockchain-oriented access control solution without the need for an agent (or gateway) design, supporting granular authorisation (block level), that has been proposed for secure EMRs management in eHealth.**

***Index Terms*—access control, EMRs management, privacy preservation, Internet of Things.**

I. Introduction

The global digital health or eHealth market is estimated to be worth over 500 billion dollars by 2025 in the United States alone [1]. Pivotal to the growth in eHealth, is a growth in the management of electronic medical records (EMRs). EMRs tend to be highly distributed in terms of who (doctor, nurse, administrator etc.) has modified what and where this is done [2]. For example, the use of Internet of Things (IoT) can allow the wearable eHealth devices to be used outside healthcare centres, enabling not just health providers, but also health users, to monitor their own health status anywhere and anytime [2–4].

In this scenario, EMR security including secure storage and access control is a major requirement for EMR management in eHealth [3, 5], yet is quite challenging to achieve because of the highly distributed and fragmented nature of EMRs and the range of providers and users who are authorised to access the EMRs. Recently, the Blockchain model (structure) is being investigated as a potential solution for EMR management since Blockchain is a highly distributed data structure suitable for EMR storage and queries. Furthermore, a feature of a Blockchain is that it inherently enables all the operations (add, query and modify) in EMRs to be verified and recorded through a consensus of all the parties involved [6]. However, one of the key challenges here for EMR's use of a Blockchain is that the initial focus of the Blockchain design is not to limit unauthorised, granular access to avoid specific confidential parts of an EMR implemented using a Blockchain [7]. It means the Blockchain design can protect the data integrity of EMRs but has no data access control.

As EMRs hold personal information about patients that can be confidential and private to the stakeholders, new approaches to constructing an access control solution to EMRs is needed. However, the current research, which applies Blockchain to EMRs, usually also supports authentication to validate users' identities but without any authorisation design to determine what the users can access. For example, a doctor can only look through the EMRs of the patients he (or she) is responsible for. For a nurse, he (or she) should not know the diagnosis, the drug injection doses or other private data (e.g. social security numbers and home addresses) of the patients he (or she) is not taking care of. On the other hand, the agent (or gateway) design used in much current research may not be suitable for mid-scale (or small-scale) eHealth service providers (companies) as they cannot afford to build a network infrastructure to implement the agent structure. These challenges motivate us to present a new EMR access control solution to achieve precise access control (block level granularity) for EMRs queries, without the need for agent support.

Related Work. Research regarding the security for applying Blockchain in EMRs is currently still in its infancy. For example, [8] and [9] use Blockchain's feature to implement the consent management in eHealth but there is no access control consideration. [10] proposes an access architecture, Healthcare Data Gateway (HDG) for Blockchain use in eHealth. However, it requires a gateway to support access control. Furthermore, the information exchange scheme designed is quite complex, which may be too heavy to be executed in low-resource IoT devices used in eHealth. [11] discusses a permission control method for a Blockchain used to implement secure EMRs in different use case scenarios. A simple system structure for applying Blockchain in eHealth is presented, but no detailed scheme or algorithm is proposed in the article.

In the conventional EMR management field (without

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

Blockchain), there have been various attempts to address the access control demand for EMRs in eHealth. [12] proposes an access control scheme for eHealth based upon elliptic curve cryptography (ECC) but there is no consideration or design to control the access granularity in the proposed authorisation algorithm. [13] constructs a detailed access control scheme, called ESPAC, to implement block level granularity authorisation for data queries in eHealth. The major cryptographic method in the scheme construction is based upon attribute-based encryption (ABE). However, the attribute-based encryption is time-consuming as it comprises bilinear pairing [14], which is too heavy-weight to be supported by current resource-constrained IoT devices in eHealth. On the other hand, ESPAC is similar to HDG, i.e., ESPAC also needs to deploy an agent to be used to authorise users' access.

Our Contribution. Compared with existing access control schemes for EMRs in eHealth [10–13], we highlight three novel contributions. First, to the best of our knowledge, our scheme is the first Blockchain-oriented access control scheme without requiring agent (or gateway) structure support. Second, the authorisation, encryption and decryption in our scheme do not rely on public key encryption or a public key infrastructure (PKI), thus lowering the computation time needed. Third, compared with the schemes in [11, 12], BBACS implements an access control scheme to validate the data queries with block level granularity.

Organisation. The remainder of the paper is organised as follows. In section II, preliminaries, in order to understand our access control scheme, i.e., the general Blockchain and the elliptic curve Diffie-Hellman assumption are discussed. Then in Section III, our proposed access model is presented. After that, we discuss the detailed design of our access control (authorisation) scheme in Section IV. Section V describes the results of our performance simulations for the proposed scheme, which is followed by the final Section VI that presents the conclusions of our work.

II. Preliminaries

A. General Blockchain

A general Blockchain consists of many blocks linked with hash values [7]. In this model, each block contains data, a cryptographic hash value (h) and a timestamp (ts), and the data can contain several attributes and their values. The hash value of one block is produced by the previous hash value and the data in the current block. It means that the hash value is applied for establishing the link between two blocks and any fallacious and distorted data would be figured out by verifying the hash value. A general Blockchain model is demonstrated as follows (Figure. 1).

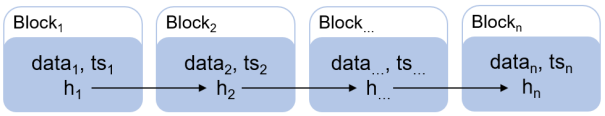

Fig. 1: The general Blockchain model

In this model, the hash values are generated by two rules:

$$h_i = \begin{cases} Hash(data_1||ts_1) & , \ i = 1 \\ Hash(h_{i-1}||data_i||ts_i), & i = 2...n \end{cases}$$

B. Elliptic Curve Diffie-Hellman (ECDH) Problem

Let $E_p(a, b)$ be a cryptographic secure elliptic curve with the prime order and a base point G. For any point $P \in E_p(a, b)$ and a random $u \in_R \mathbb{Z}_p^*$, any probabilistic polynomial-time algorithm $\mathcal{A}$ computes u with its advantage:

$$Adv^{ECDH}_{\mathcal{A},E_p(a,b)} = Pr[c = u | u \in_R \mathbb{Z}_p^*, c = \mathcal{A}(P, uP)]$$

The ECDH assumption can hold if for any probabilistic polynomial-time algorithm $\mathcal{A}$, its advantage $Adv^{ECDH}_{\mathcal{A},E_p(a,b)}$ is negligible.

III. Proposed Access Model

The two entities in the presented model (Figure. 2) are users and the EMR server. The two entities are located in the same trusted network (cloud). Then, we introduce the two entities of our access model in Figure. 2 and illustrate the functions of each entity.

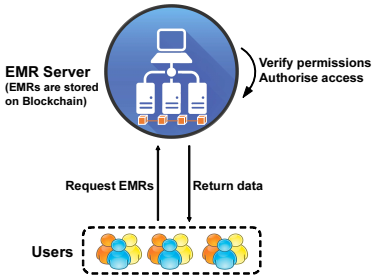

Fig. 2: The proposed access model

1) Users: The users represent data inquirers e.g. doctors, patients and data analysts. The data inquirers initiate queries to obtain data from the EMR server. In general, the EMRs of each patient are constructed by several blocks. A unique token is allocated to each block when the block is added to the patient's EMRs. As the focus of our scheme is access authorisation during data exchange, the token distribution is not yet discussed in this paper. Hence, we assume that the token for each block on the Blockchain has been distributed to the corresponding valid data requesters (e.g. the patient, doctors and data analysts) by the medical management authority. Note that whilst a patient usually queries the personal blocks; the doctors and data analysts normally query many blocks from different patients' EMRs. All the access permissions of the queries will be verified through the EMR server.

2) EMR server: There are two functions designed for the EMR server in our model. First, the EMR server is used to store all the electronic medical records (EMRs) for eHealth in the Blockchain. Another function of EMR server is to verify the access permissions and authorise the block(s) access. If the data inquirers possess all the required permissions, the EMR server can authorise the access then return the queried data (blocks) to the users.

IV. Proposed Authorisation Scheme

In this section, we first propose our access control (authorisation) scheme, BBACS (Block-based Access Control Scheme) and then prove its correctness. After that, the security of our proposed scheme BBACS is analysed.

A. The Proposed Scheme

1) ***Setup****(λ):* This procedure outputs public parameters pp with the security parameter λ using the following steps.

1. Select a secure elliptic curve $E_p(a,b)$ and a base point G on $E_p(a,b)$, where $p \in \{0,1\}^\lambda$ is a big prime, and a, b are the parameters of the elliptic curve.

2. Generate a random integer token $TK \in \{0,1\}^\lambda$ for each block BLK (EMR) on the Blockchain and write the token TK in the block BLK. Note that this step should be executed when a new block is created and added to the Blockchain to keep the integrity of the Blockchain. Furthermore, all the tokens have been distributed to the corresponding users correctly before users query blocks from the EMR server.

3. Select a symmetric encryption algorithm, e.g., AES (Advanced Encryption Standard).

4. Select one secure cryptographic hash function, $H : \{0,1\}^* \rightarrow \{0,1\}^\lambda$.

5. Output the public parameters pp to finish the *Setup* phase, $pp = (E_p(a,b), G, AES, H)$.

2) ***Query****(pp):* In this phase, the users prepare the data query Q via the following steps.

1. Prepare the sequence S_{id} including all the indexes of the queried blocks. For example, if a user want to request several (n) blocks on the Blockchain, sequence S_{id} should contain all the indexes of the requested blocks n: $S_{id} = \{BLK_{id_i} | i = 1...n\}$.

2. Prepare the sequence S_{TK} including all the tokens of requested blocks. For the given sequence S_{id} in step 1, $S_{TK} = \{TK_{id_i} | i = 1...n\}$.

3. Use the sequence S_{TK} to calculate a value $k = TK_{id_1} \oplus TK_{id_2} \oplus ... \oplus TK_{id_{n-1}} \oplus TK_{id_n}$.

4. Calculate a point multiplication on elliptic curve $E_p(a,b)$ with k and the base point G to obtain a new point $P = kG$.

5. Send the query $Q = (S_{id}, P)$ to the EMR server.

3) ***Authorise****(pp, Q):* The EMR server validates the access permissions provided in Q from the user via the following steps.

1. Repeat step 2 in the *Query* phase with the indexes $S_{id} \in Q$ to obtain the token sequence $S'_{TK} = \{TK'_{id_i} | i = 1...n\}$ from the Blockchain.

2. Use the sequence S'_{TK} to calculate the $k' = TK'_{id_1} \oplus TK'_{id_2} \oplus ... \oplus TK'_{id_{n-1}} \oplus TK'_{id_n}$.

3. Repeat step 4 in the *Query* phase to obtain the point $P' = k'G$.

4. If the two points $P = P'$ holds, it means the user has the correct access permissions to be authorised to access the queried blocks S_{id}, otherwise, the EMR server should deny the query Q from the user.

4) ***Encrypt****(pp, Q, k'):* The EMR server encrypts the queried data via the following steps.

1. Prepare the queried blocks M based upon the sequence $S_{id} \in Q$ from the user then calculate the hash value $H_M = H(M)$ of the data M.

2. Use $AES \in pp$ to encrypt M and $H(M)$ with the key k' to output the ciphertext $C = AES_{k'}(M, H_M)$. Besides, $AES'_{k'}$ is defined as the decryption process to decrypt $C = AES_{k'}(M, H_M)$ to recover the plain data M, i.e., $M = AES'_{k'}(C = AES_{k'}(M, H_M))$.

3. Return the ciphertext C to the user (data requester) to finish the authorisation and data transmission.

5) ***Decrypt****(pp, k, C):* If the user has all the access permissions (tokens) for the queried blocks, the user is authorised to access these during the *Authorise* phase and can then decrypt the ciphertext C generated by the *Encrypt* phase via the following steps.

1. Decrypt C to with the key k to retrieve the plaintext $(M, H_M) = AES'_k(C) = AES'_k(AES_{k'}(M, H_M))$.

2. If the condition $H(M) = H_M$ holds, the algorithm outputs M, otherwise, it outputs $\perp$.

The following Figure. 3 shows the work flow of our scheme.

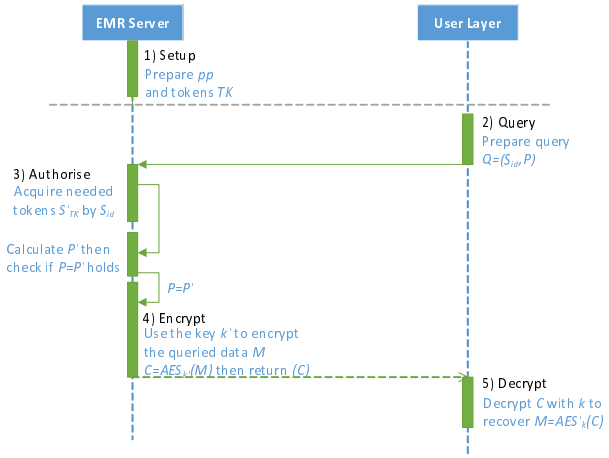

Fig. 3: The work flow of the proposed scheme BBACS

B. Correctness

In the *Authorise* phase, the two points $P = P'$ is the key condition for a user to pass the authorisation phase.

$$
\begin{aligned}
& P = P' \\
\Leftrightarrow\ & kG = k'G \\
\Leftrightarrow\ & (TK_{id_1} \oplus ... \oplus TK_{id_n})G = (TK'_{id_1} \oplus ... \oplus TK'_{id_n})G \\
\Leftrightarrow\ & TK_{id_1} \oplus ... \oplus TK_{id_n} = TK'_{id_1} \oplus ... \oplus TK'_{id_n} \\
\Leftrightarrow\ & \{TK_{id_1}, ..., TK_{id_n}\} = \{TK'_{id_1}, ..., TK'_{id_n}\}.
\end{aligned}
$$

It means the user can pass the *Authorise* phase if and only if this user has offered all the required access permission tokens in Q correctly for the requested blocks.

In the *Decrypt* phase, the user can succeed to decrypt the ciphertext C and ensure $H_M = H(M)$ if the condition $k = k'$ is fulfilled. According to the correctness analysis for the *Authorise* phase, the condition $k = k'$ can hold if

the user is authorised to access the queried blocks successfully. To be detailed, the condition $k = k'$ is equivalent to $\{TK_{id_1}, TK_{id_2}, ..., TK_{id_n}\} = \{TK'_{id_1}, TK'_{id_2}, ..., TK'_{id_n}\}$, i.e., the sequences $S_{TK} = S'_{TK}$. Therefore, the keys k and k' in the phases *Encrypt* and *Decrypt* respectively are identical.

Hence, the authorised user can decrypt the ciphertext C with the key k correctly.

C. Security Analysis

In this section, we analyse the security of our access control scheme BBACS from two angles, data confidentiality and integrity.

1) Confidentiality: There are two potential attacks may occur in the communications between the users and the EMR server based upon BBACS.

(i) The user tries to query a block but without the corresponding access permission. For example, the user plays the following game to query several blocks from the EMR server.

In the *Query* phase, the index sequence of the queried blocks is defined as $S_{id} = \{BLK_{id_1}, BLK_{id_2}, ..., BLK_{id_n}\}$. However, the user does not have the access permission (token) TK_{id_n} of the last block BLK_{id_n}. The partial access tokens that the user owns are presented as the sequence $\{TK_{id_1}, TK_{id_2}, ..., TK_{id_{n-1}}\}$ for the queried blocks $\{BLK_{id_1}, BLK_{id_2}, ..., BLK_{id_{n-1}}\}$. To construct a valid query, the user counterfeits the access token $TK^*_{id_n} \in \{0,1\}^\lambda$ for the block BLK_{id_n} and then then organises the fake token sequence $S^*_{TK} = \{TK_{id_1}, TK_{id_2}, ..., TK_{id_{n-1}}, TK^*_{id_n}\}$. After that, the user follows step 3 and step 4 in the *Query* phase of BBACS to calculate the point $P^* = (TK_{id_1} \oplus TK_{id_2} \oplus ... \oplus TK_{id_{n-1}} \oplus TK^*_{id_n})G$. Finally, the user sends the query $Q^* = (S_{id}, P^*)$ to the EMR server.

Next, for the EMR server in the *Authorise* phase, the correct token sequence for the queried blocks S_{id} is $S'_{TK} = \{TK'_{id_1}, TK'_{id_2}, ..., TK'_{id_{n-1}}, TK'_{id_n}\}$. Then the EMR server follows the step 2 and the step 3 in *Authorise* phase to calculate the point $P^* = (TK'_{id_1} \oplus TK'_{id_2} \oplus ... \oplus TK'_{id_{n-1}} \oplus TK'_{id_n})G$. Based upon the correctness analysis above (see Section IV.B), the condition that allows the user to be authorised successfully is $P' = P^*$. Meanwhile, the condition $P' = P^*$ is equivalent to $S'_{TK} = S^*_{TK}$. Even though $\forall i \in \{1..n-1\}, TK_{id_i} = TK'_{id_i}$ $(TK_{id_i} \in S^*_{TK}, TK'_{id_i} \in S'_{TK})$ holds, the advantage $Pr[TK'_{id_n} = TK^*_{id_n}]$ is $\frac{1}{2^\lambda}$ to satisfy the condition $TK'_{id_n} = TK^*_{id_n}$, where λ represents the security parameter in the *Setup* phase. Note that $\frac{1}{2^\lambda}$ is quite small as λ is big enough so that the advantage $Pr[TK'_{id_n} = TK^*_{id_n}] = \frac{1}{2^\lambda}$ is negligible. It means that the probability of $S'_{TK} = S^*_{TK}$ is negligible as well. Hence, the user cannot be authorised to access the queried data in the *Authorise* phase (see Section IV.A) since $Pr[P^* = P'] = Pr[S^*_{TK} = S'_{TK}] = Pr[TK'_{id_n} = TK^*_{id_n}] = \frac{1}{2^\lambda}$ is negligible based upon the above analysis.

(ii) The communications between the user and the EMR server could be eavesdropped upon by an evil attacker, who can acquire all the data of the communications between the user and the EMR server. The data contains the user's query $Q = (S_{id}, P)$ and the response data (C).

First, the attacker cannot retrieve the plain data M with the user's query Q and the returned data (C) because of two reasons. Firstly, the attacker cannot acquire the token sequence S_{TK} from the communications. Furthermore, based upon the ECDH problem, the attacker cannot determine the value k with $P(= kG), G$ and pp. Secondly, the attacker cannot recover the plaintext M from the ciphertext C without the correct AES decryption key k (or k').

On the other hand, the modification in $S_{id} \in Q$ will lead to the incorrect token sequence S'_{TK} in the *Authorise* phase so that the point P' is changed as well. If the attacker modifies the point $P \in Q$ directly, the point P cannot be matched with the point P' calculated by the EMR server in the *Authorise* phase. As a result, the attacker cannot pass through the *Authorise* phase in our scheme BBACS based upon our security analysis.

2) Integrity: If the attacker intercepts the communications between the user and the EMR server, the attacker can challenge the integrity via forging the returned data C.

If the attacker distorts the original ciphertext C to C^*, the user can still decrypt C^* with the user's key k to retrieve the plaintext (M^*, H^*_M). However, based upon the AES algorithm, the decrypted $AES'_k(C^*) = (M^*, H^*_M)$ is changed and the attacker cannot control M^* and H^*_M to make $H(M^*) = H^*_M$ hold via tampering C^*. As a result, the step 2 in the *Decrypt* phase outputs $\perp$.

Overall, our scheme BBACS can offer sufficient security to ensure that both the confidentiality of the transported data and the data integrity are protected based upon the security analysis above when attacks happen to the communications between users and the EMR server.

V. Performance Simulation

In this section, the two parts to be illustrated are as follows. The first one is to compare the theoretical cryptographic operations. And the second one is about the experimental time efficiency. In the second part, the computational time efficiency of our scheme BBACS is evaluated with respect to the time cost for transmitting encrypted data over Wi-Fi. The reason for evaluating the time cost of data transmission over Wi-Fi is that the lower time cost of data transmission via Wi-Fi can lead to lower power cost, especially for those resource-constrained eHealth devices.

As there is yet no clear best practice to be used as a baseline for comparison, we select an access control scheme based upon an agent for Blockchain-based EMRs system named HDG [10] as our baseline. Note that compared with the HDG scheme that needs an agent to support access control, our scheme BBACS can verify and authorise the data access without the agent.

A. Theoretical Comparison

The major cryptographic operation used in our scheme BBACS and the HDG scheme [10] is scalar multiplication. Hence, we denote by O_{mul} an operation of the scalar multiplication for the following comparison. The result is shown in the Table I. In terms of the encryption part, our scheme BBACS

requires less scalar multiplication operations. Furthermore, BBACS does not use any complicated public-key cryptographic operation in the decryption. Note that there are other

TABLE I: Theoretical comparison of the used cryptographic operation

	BBACS	HDG
Encryption	$2O_{mul}$	$5O_{mul}$
Decryption	$0O_{mul}$	$1O_{mul}$

cryptographic operations and algorithms used in the schemes BBACS and HDG, e.g. XOR operation, hash summary and AES. However, when compared with the denoted operation O_{mul}, the time complexity of these operations and algorithms is negligible [15]. Therefore, these low time complexity operations and algorithms are not taken into account in our theoretical comparison.

B. Simulation Comparison

In this section, our simulation result for comparing the time efficiency of local computation and transmission in the two schemes is demonstrated. Note that the entire two schemes are simulated to acquire the result. Besides, all the test results are averaged over 10 runs for each scheme. The devices for our simulation use are a conventional computer with an Intel i5-4200H processor running at 3.30GHz, and a Raspberry Pi 2 as a low-resource eHealth IoT device.

The first simulation is performed on the mentioned conventional computer. We vary the number of the blocks requested by the user to compare the time consumption of the encryption and decryption algorithms in the two schemes (BBACS and HDG). The simulation is implemented based upon MIRACL [16], which can support all the necessary cryptographic operations for the two schemes. Note that the block size we used is 16 Kbytes and the length of the index BLK_{id_i} and token TK_{id_i} for each block is 256 bits. We assume the queried EMRs of patients vary from 2 to 10 with 10 blocks in each patient's EMRs. The number of the queried blocks set in the simulation varies from 20 to 100 with 10 runs for each number of the queried blocks to finally calculate the average results. Furthermore, the cryptographic security level [17] of all the implemented experiments is equivalent (128-bit security).

Our comparison result is shown in the next Figure. 4. Since the scheme HDG involves an agent, it uses more scalar multiplication operations and AES algorithms to authorise the access and encrypt required data. As a result, the time efficiency of our scheme BBACS is significantly superior in terms of the encryption. On average, our scheme BBACS consumes 73% less time than the scheme HDG in terms of the encryption. Furthermore, the growth rate of the computational time for the encryption process in BBACS is lower than the relative growth rate in HDG as seen in Figure. 4.

Next, we keep all the above experimental parameters then repeat the simulation on the Raspberry Pi 2. The simulation result depicted in the Figure. 5 shows that the comparison of the computational time cost on the Raspberry Pi 2 is consistent with the result on the conventional computer (Figure. 4). On average, whilst the time consumption of BBACS is only

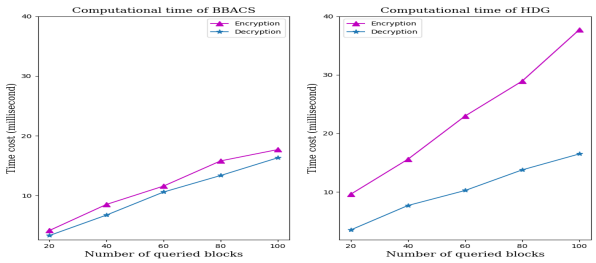

Fig. 4: The computational time cost of encryption and decryption algorithms in the schemes BBACS and HDG on the conventional computer

around 29% that of the scheme HDG in terms of the encryption algorithm, the time cost of decryption in BBACS is 9% lower than that of the HDG scheme.

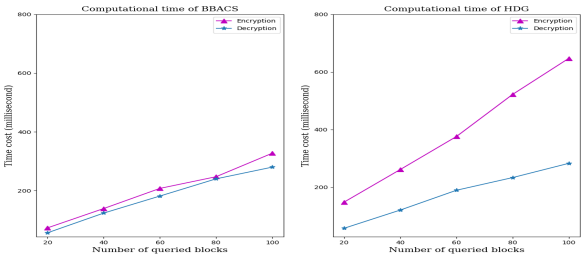

Fig. 5: The computational time cost of encryption and decryption algorithms in the schemes BBACS and HDG on the low-resource IoT device (Raspberry Pi 2)

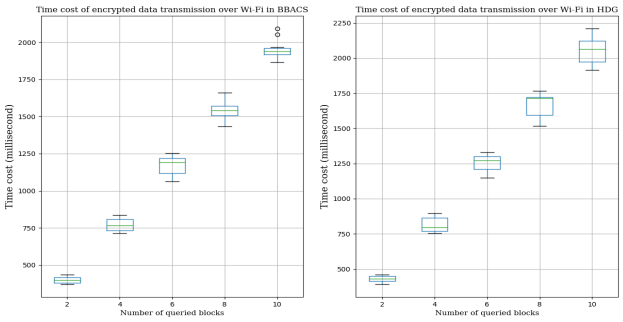

Fig. 6: The comparison of the time cost for transporting encrypted data on the Raspberry Pi 2 in schemes BBACS and HDG

Furthermore, we test the time cost of transporting the encrypted data over Wi-Fi to (and from) the user in the two schemes. Note that the number of the queried blocks is set between 2 to 10; meanwhile, the block size is adjusted to 512 bytes. The data transmission process in our simulation is implemented based upon the socket communication in Python-2.7. The result presented in the Figure. 6 indicates that the time cost for transporting the encrypted data to (and from) the user via Wi-Fi increases linearly with the number of the queried blocks in both two schemes BBACS and HDG. However, the

growth rate of the computational time for BBACS is lower than that in the compared scheme HDG. This is because the point P (see Section IV.A.2) is the only data we use to authorise the access, and the data size of the point P is fixed. However, the scheme HDG needs to transport the whole token list to the agent to authorise the user's data query. The length (data size) of the token list is sensitive to the number of the queried blocks so that it leads to the higher growth rate of time cost in the encrypted data transmission over Wi-Fi in HDG.

The summary of the simulating comparisons and the features of the two schemes is given in Table. II.

TABLE II: Comparison of the simulation results and the features for BBACS and HDG

	BBACS	HDG
Computational time cost	Low	High
Network throughput	Low	High
Trusted third party for authorisation	Unnecessary	Necessary

VI. Conclusion

In this paper, we proposed what we believe is the first access model without an agent layer or gateway support to realise the access control to authorise the data access from users to the Blockchain-based EMR server. Compared with the existing work, e.g., HDG, our scheme BBACS does not require the agent or PKI to authorise the access or encrypt/decrypt the queried EMRs. The proposed access control (authorisation) scheme can validate the access permission for each queried block to authorise the data queries from users. As a result, a Blockchain-based EMR server can respond to the data requesters without the assistance of agent(s) or leaking unauthorised private medical records, especially for resource-constrained devices used as in those IoT systems for eHealth. Further work will entail validating the use of the algorithms on additional low resource IoT devices and evaluating the scalability in its use on multiple user EMRs.

Acknowledgement

This research was funded in part by a PhD scholarship funded jointly by the China Scholarship Council and Queen Mary University of London.

References

[1] Transparency Market Research, "Digital Health Market - Global Industry Analysis, Size, Share, Growth, Trends and Forecast, 2017 - 2025," [Online]. Available: https://www.transparencymarketresearch.com/digital-health-market.html, 2017.

[2] W. Leister, M. Hamdi, H. Abie, A. Torjusen, and S. Poslad, "An Evaluation Framework for Adaptive Security for the IoT in eHealth," *International Journal on Advances in Security*, vol. 7, no. 3–4, pp. 93–109, 2014.

[3] M. Al Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of medical systems*, vol. 36, no. 1, pp. 93–101, 2012.

[4] X. Zhang, S. Poslad, and Z. Ma, "A semi-outsourcing secure data privacy scheme for iot data transmission," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*. IEEE, 2017, pp. 1–5.

[5] W. Leister, M. Hamdi, H. Abie, and S. Poslad, "An evaluation scenario for adaptive security in eHealth," in *Proceedings of Fourth International Conference on Performance, Safety and Robustness in Complex Systems and Applications, Nice, France*, vol. 2327, 2014.

[6] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "How blockchain could empower ehealth: An application for radiation oncology," in *VLDB Workshop on Data Management and Analytics for Medicine and Healthcare*. Springer, 2017, pp. 3–6.

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

[8] P. Genestier, S. Zouarhi, P. Limeux, D. Excoffier, A. Prola, S. Sandon, and J.-M. Temerson, "Blockchain for consent management in the ehealth environment: A nugget for privacy and security challenges," *Journal of the International Society for Telemedicine and eHealth*, vol. 5, pp. 24–1, 2017.

[9] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, "Towards using blockchain technology for ehealth data access management," in *Advances in Biomedical Engineering (ICABME), 2017 Fourth International Conference on*. IEEE, 2017, pp. 1–4.

[10] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of medical systems*, vol. 40, no. 10, p. 218, 2016.

[11] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and trustable electronic medical records sharing using blockchain," *arXiv preprint arXiv:1709.06528*, 2017.

[12] J. Liu, Y. Xiao, and C. P. Chen, "Authentication and access control in the internet of things," in *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*. IEEE, 2012, pp. 588–592.

[13] M. Barua, X. Liang, R. Lu, and X. Shen, "Espac: Enabling security and patient-centric access control for ehealth in cloud computing," *International Journal of Security and Networks*, vol. 6, no. 2-3, pp. 67–76, 2011.

[14] R. W. Zhu, G. Yang, and D. S. Wong, "An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices," *Theoretical Computer Science*, vol. 378, no. 2, pp. 198–207, 2007.

[15] T. Eisenbarth and S. Kumar, "A survey of lightweight-cryptography implementations," *IEEE Design & Test of Computers*, vol. 24, no. 6, 2007.

[16] M. Scott, "MIRACL - Multiprecision Integer and Rational Arithmetic C/C++ Library." [Online]. Available: https://github.com/miracl/MIRACL, 2012.

[17] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management part 1: General (revision 3)," *NIST special publication*, vol. 800, no. 57, pp. 1–147, 2012.