User-Controlled, Auditable, Cross-Jurisdiction Sharing of Healthcare Data Mediated by a Public Blockchain

Xiaohu Zhou, Vitor Jesus, Yonghao Wang and Mark Josephs
School of Computing and Digital Technology
Birmingham City University, Birmingham, UK
{vitor.jesus, yonghao.wang, mark.josephs}@bcu.ac.uk
xiaohu.zhou@mail.bcu.ac.uk

***Abstract*—We tackle the problem of sharing eHealth data across different jurisdictions. As a general rule, and due to the sensitive nature of the information, different national regulations impose severe limits on what can be exchanged, even in case of emergencies. Furthermore, different systems in different jurisdictions do not communicate. We propose BRUE as a scheme that allows eHealth data to be securely exchanged, with the data subject always in the position of mediation. We combine several technologies, namely, Blockchain, OAuth and User-Managed Access, and concept Receipts, to achieve our aim.**

***Index Terms*—eHealth data exchange, blockchain, smart-contracts, distribute, cross-jurisdiction**

I. Introduction

People like to travel and may visit different doctors across different jurisdictions (such as countries). eHealth data of individuals is managed by the diverse health service providers and stored in different locations. Although there are many agreements between different jurisdictions (such as countries), in general they do not allow eHealth data to be shared externally. Often, sharing is not even allowed within the same country between different healthcare providers. There are several reasons but, as a general theme, it is due to a lack of trust or data disclosure considerations stemming from compliance and regulations. There is a consensus that healthcare data is sensitive personal information that must be well protected.

In this paper, we tackle the problem of sharing eHealth data across different jurisdictions. The overall challenge stems from three fundamental problems. *First*, there needs to be full accountability (e.g., non-repudiation) when sharing data; accountability also refers to the possibility of someone sharing healthcare data without the authorisation of the patient. *Second*, since there is no global infrastructure to discover the locations of eHealth data, it must be a truly decentralised scheme. *Third*, before sharing eHealth data, explicit consent must be obtained from the data subject. The eHealth data custodian needs to be able to demonstrate that appropriate measures had been taken to address these problems if there is an audit or breach in the future.

We tackle this problem by, first, centring all the information exchange and control on the data subject. The user is who mediates all steps and is, effectively, the (cryptographically) trusted communication channel between all the parties. Second, we combine a set of technologies and standards, each addressing a particular requirement. For authorisation and managing access to the records, we use and extend OAuth's-based User-Managed Access (UMA) [1]. For a trusted and confidential communication channel, distributed discoverability and overall accountability, we use a public blockchain able to run smart-contracts. To handle the requirement of demonstration of valid consent, we use Kantara's Consent Receipts [2]. This combination of technologies motivates the name of our scheme: Blockchain, Receipts and UMA for eHealth data exchange (BRUE).

To illustrate the problem, we informally analyse a simple working scenario of international eHealth data exchange.

Alice (A) has had heart trouble since she was born in France. She has registered a GP (FGP) in her original city. Then she moved to the UK after 10 years. When she was 20 years old and travelled to Canada, she fell sick and visited a GP (CGP) to get e temporary treatment in Canada. After that trip, she returned to the UK and visited her British doctor (BGP, where GP stands for "General Practitioner"). She would like to share the British GP with her previous medical data which is stored respectively in Canada and France. However, she doesn't want to simply give British GP her personal credential but would rather authorize British GP which could gain access using GP's credential.

In the scenario, there are four entities: the *data subject* A, a *requesting party* BGP, and *data controllers* which are healthcare service providers FGP and CGP. To note that A, BGP, CGP, and FGP, are independent parties located in different jurisdictions. Some of them are not known to the others; their only point of contact is their relationship with A, the patient and data subject. Furthermore, we assume, as is overwhelmingly the case, that there is no global system in place that allows all parties to directly communicate, find each other, or self-certify. In other words, BGP needs to access A's medical data from FGP and CGP but FGP and CGP do not recognise BGP so A needs to intermediate the request and grant access. A, in turn, needs to authenticate against FGP

and CGP.

In the remainder of the paper, Section II reviews the literature on eHealth data exchanges and highlights our contribution of this paper. Section III formally defines the problem and introduces notation and terminology. Section IV presents our approach to the problem. In Section V, we describe and discuss a proof-of-concept application of our approach based on Ethereum. In Section VI we conclude our paper and discuss future work.

II. Related Work

Healthcare is a sensitive domain that poses and processes a large amount of personal medical data daily. Regarding the requirement of eHealth data exchange, the application of blockchain (Distribute ledger) technology in eHealth data exchange is a continuous hot topic which has been significantly discussed in the literature. We show relevant work of distributed ledger in eHealth data exchange in this section.

A. Blockchain in Healthcare Data Exchange

Prior literature about eHealth data exchange with distributed ledger commonly review the current research situation and evaluate the existing distributed application. This raises a problem of performance feasibility as there is no any implementation work prove in practise. Angraal et al. [3] assess several healthcare applications based on blockchain and point out that key limitations of blockchain technology expanded to large-scale production deployment in future research are system scalability, security and cost-effectiveness. Zhang et al. [4] define a set of evaluation metrics for blockchain-based healthcare decentralized applications to guild the development of blockchain applications in the healthcare domain, which include cost-effectiveness, patient-centred care model, system scalability, interoperability, user identification, and Turing-complete operations. However, those metrics concern only the requirements of HIPAA[1], the key Healthcare regulation in the United States. McGhin et al. [5] compare nine types of existing blockchain-based applications in healthcare and give tips on how blockchain technology meet which requirements of the healthcare industry. They also present limitations and technical issues of blockchain technology, such as mining incentives and standardization, although its applications have potential benefits for the healthcare industry. Mackey et. al. [6] recommend a 'fit-for-purpose' framework as a guide of health blockchain application design.

The remaining of literature propose theoretical approaches without/with performance prove in practise, which can be classified by the data storage. Some approaches attempt to store all data with a public, or private, or consortium blockchain instead of traditional data storage; other approaches use a mixed approach in data storage. MA-ABS [7] scheme is a typical approach for medical data exchange with blockchain, which is using blockchain to exchange encapsulated electronic medical record (EMR) with an attribute-based signature scheme authorized by multiple authorities. In this approach, although the exchanged message is endorsed by participants without any information disclosure, it has a scalability problem due to a significant volume of storage of EMRs data in the blockchain. Another example is a blockchain-based secure and privacy-preserving personal health information sharing (BSPP) [8] scheme for diagnosis improvement. It uses a private blockchain to store eHealth data and a consortium blockchain to record the secure indexes of eHealth data. These approaches have benefits on access control and confidentiality because of storing all data in the blockchain, they are not compliant with regulations and specifications, such as EU/GDPR[2], which requires data subjects to have the right to erase personal information. However, depending on the type of blockchain, it may not be possible to modify or delete after-the-fact.

MedBlock [9] is a blockchain-based information management system, which attempts to provide large scale data retrieval and share without extra costs and network congestion. While it does not consider participants' incentive in the system, MeDShare [10] is a blockchain application to improve security and data authentication in medical data sharing. MedRec [11], in turn, built upon existing databases that support open standards of healthcare exchange to facilitate data share and authentication. The key focus is on designing mining incentives and is not focused on the security issue of the existing database.

Esposito et al. [12] suggest an approach that uses a conventional or distributed database to store medical data and an online chain to record the hash value of those data. This scheme is not implemented in practise [13]. Zhang et al. [14] recommend a use case 'Decentralized Application for Smart Health' (DASH) to solve problems of storage requirements, privacy and scalability. It needs to verify or prove by experiments for feasibility and concern the auditability. They also propose a hybrid on-chain/off-chain framework to improve the security and scalability in clinical data sharing, named FHIRchain [15], to address requirements of FHIR [16], an industry standard of healthcare information exchange.

B. Contribution of This Paper

In the design of BRUE, we focus on auditability, confidentiality, and distributedness in cross-jurisdiction eHealth data exchange. Firstly, minimise the information exchanged in the sharing network. It is designed to verify identity information in the local jurisdiction and it is not necessary to check and share identity information cross-jurisdiction. Secondly, we propose a scheme where lightweight and short-lived authorisation tokens are created and shared between entities to access eHealth data cross region. For the identity information, a token without identity information is produced by authorization party to represent the verified identity. All entities share and verify these tokens with each other to create an individual identity while not disclosing information. Tokens are shared through blockchain with smart-contracts, which are system-agnostic

[1]HIPAA: https://www.hhs.gov/hipaa/index.html

[2]GDPR: https://gdpr.eu

for the existing infrastructure. It improves system scalability. Furthermore, since tokens can be revoked at any time with immediate effect, it promotes compliance with virtually all regulations. Thirdly, we reuse the notion of Personal Data Receipt from the Data Protection communities [17]. BRUE provides acknowledgement Receipts for all operations of the involved entities. Receipts not only meet accountability requirements (for data controllers) but also provide the data subject with means to trace accesses to the past. Finally, cryptographic access is required for any exchange between two entities. For this matter, we adapt the well-known Diffie-Hellman secret sharing scheme [18] to be used in a blockchain providing the useful result of proving, beyond any doubt, that the two parties were engaged. It further provides forward-secrecy and confidentiality.

III. Problem Statement

The key requirements of a distributed cross-jurisdiction eHealth data exchange architecture needs to target *auditability*, *non-repudiation*, *confidentiality*, and *compatibility*.

A. Overall Perspective

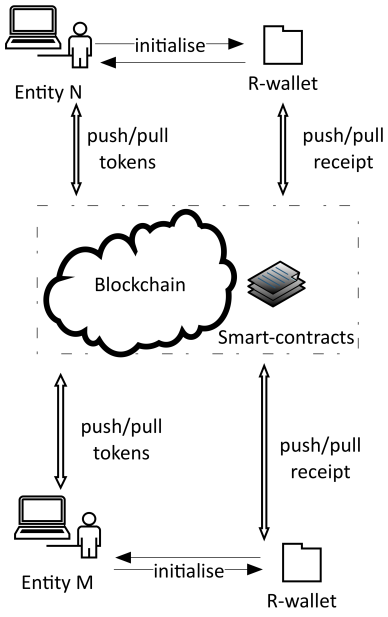


Figure 1: Data exchange between two parties.

Figure 1 shows actions for a simple data exchange between only two parties, N and M. All data flow of transactions from entities go through the blockchain network (for provenance and accountability). A receipt of each transaction is also produced following the data flow. In the sketch, N and M start by preparing R-wallet to receive receipts[3]; N then requests services from M by invoking smart-contracts running on the blockchain. M processes work pulled from the blockchain (via notifications) to complete the request from N. When M returns the expected outcome and pushes the result to the blockchain, N then obtains the outcome from the blockchain itself. The fact that the communication channel is the blockchain itself guarantees traceability.

The receipt as proof of an intermediate transaction is generated and follows the data flow in both directions. The peer-to-peer data exchange between N and M terminates at this point. The receipt generated while the data flow progresses is composed of the individual audit records. Should a concern or dispute arise in the future, this receipt holds all the evidence needed to keep all participants accountable.

Figure 2 shows a general scenario of a cross-jurisdiction eHealth data exchange flows with multiple parties: data subject, a requesting party, the data controller, and a verifier/authorizer. Each entity contributes to the overall data flow work done. The topology of participants is established before it starts because of services request and data flow direction. It is also static for its duration. We assume, without loss of work generalisation and rejection, that the graph is acyclic when each entity is a requester or responder. In other words, data flows the graph with a path by order such that no entity is a requester or a responder twice. If a particular data exchange workflow uses the same participant twice at different times, the model has a different node in the graph. BRUE is agnostic in terms of the actual eHealth data format.

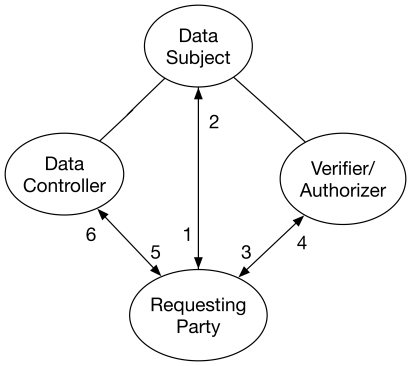


Figure 2: Multi-party data exchange flow.

B. notation

We use the following notation in our proposed approach:

- DS is the *data subject* who owns eHealth data.
- RO stands for *resource owner* who, on behalf of the data subject, can process authorization.
- RqP is the *requesting party* that wants to access the health records.
- AS is an *authorization server* which is an organisation which is authorized to manage access rights of health records. We assume, for simplicity, that each jurisdiction has only one AS.
- LAS is a *local authorization server* so that AS is located in the same jurisdiction as RqP.
- RAS is a *remote authorization server* which means AS is located in a different jurisdiction as RqP.
- RS are *resource servers* which hold the actual healthcare records – such as hospitals.

[3]We reuse the familiar term of "wallet" as the (digital) container of a receipt

- LRS is a *local resource server* located in the same jurisdiction as RqP.
- RRS is a *remote resource server* located in a different jurisdiction than RqP.
- PT is a *permission token* which provides a proof of service request permission for RqP authorized by data subject.
- VIT stands for *verified identity token* which is an authorization identity proof after individual identity verification; it is authorized by LAS for RqP.
- RPT represents a *requesting party token* which is an authorization proof of the data access permission with specified conditions by AS to RqP.
- $k_{N,M}$ is a *shared secret key* of entity N and M in the data flow following the Diffie-Hellman key exchange method.
- $R_{N,M}$ is a service request sent from entity N to M.
- $R_1 \| R_2$ denotes concatenation of request R_1 and R_2.
- I_N expresses the identity of entity N.
- $\text{sign}_N(R_{N,M})$ is a request $R_{N,M}$ signed by entity N.
- $\text{enc}_{k_{N,M}}(R_{N,M})$ presents a request $R_{N,M}$ encrypted with key $k_{N,M}$.
- $RE_{N,M}$ is a receipt of transaction between entity N and M.
- $\text{hash}(R)$ is a digest of request R using a one-way collision-resistant function (a "hash").

We use the following data formats in our signalling diagram:

- permission token

$$PT = \text{enc}_{k_{DS,RqP}}(\text{sign}_{DS}(PT))$$

  Entity RO is sending a token PT to RqP as a response for request $R_{RqP,DS}$ and, to assure auditability, it signs the message. For confidentiality, RO encrypts the token with their common secret key $k_{DS,RqP}$. A plaintext of PT includes consent status of $R_{RqP,DS}$, URL of LAS, duration of permission, start date of permission, original jurisdiction, jurisdiction of destination, expired data of permission, name of DS, RO, LAS and RqP.
- verified identity token

$$VIT = \text{enc}_{k_{LAS,RqP}}(\text{sign}_{LAS}(VIT))$$

  Entity LAS sends a token VIT to RqP for request of identity and authorization validation as a response proof. VIT is signed by LAS and encrypted with their common secret key $k_{LAS,RqP}$. A plaintext of VIT is made of consent status of $R_{RqP,DS}$, URL of RAS, status of identity verification, duration of permission, start date of permission, expired data of permission, original jurisdiction, jurisdiction of destination, name of DS, RO, LAS and RqP.
- requesting party token

$$RPT = \text{enc}_{k_{RAS,RqP}}(\text{sign}_{RAS}(RPT))$$

  Entity RAS shares a token RPT with RqP after verification. RPT is signed and encrypted by RAS with their common secret key $k_{RAS,RqP}$. The original RPT is composed of consent status of $R_{RqP,DS}$, URL of RRS, duration of permission, start date of permission, expired data of permission, original jurisdiction, jurisdiction of destination, name of DS, RO, RRS, LAS and RqP.
- receipt

$$RE_{N,M} = \text{sign}_N(R_{N,M}) \| \text{hash}(R_{N,M})$$

$$RE_{N,M} = \text{sign}_N(PT) \| \text{hash}(PT)$$

$$RE_{N,M} = \text{sign}_N(VIT) \| \text{hash}(VIT)$$

$$RE_{N,M} = \text{sign}_N(RPT) \| \text{hash}(RPT)$$

  After N shares information with M, a receipt of delivery is produced that includes this encrypted transaction and a digest.

IV. Architecture

Our approach provides the feasibility of cross-jurisdiction eHealth data exchange. It produces receipt record as participants' consent integrated with UMA standards to target auditability and compatibility; in addition to, runs on a public blockchain to achieve non-repudiation and confidentiality with blockchain's features. In this section, we present how our approach achieves these.

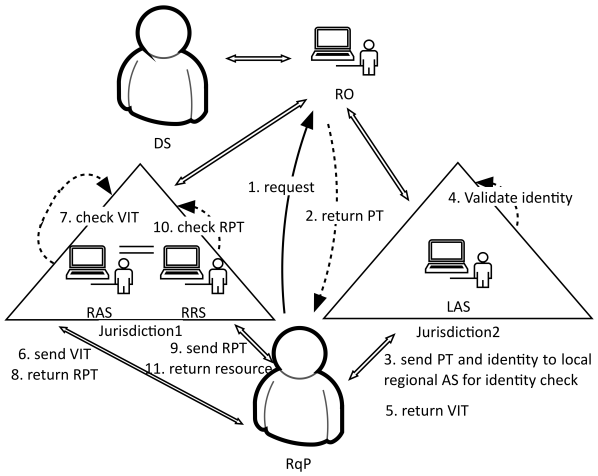


Figure 3: The representation of proposed approach structure

Figure 3 shows the sequence of actions in BRUE. RqP is located in jurisdiction 2 (e.g., it's a foreign hospital) who needs to access eHealth data of data subject from RRS located in jurisdiction 1. A completed data flow is shown as below. At first, RqP requests service from the data subject. RO processes this request on behalf of the data subject. RO checks service request and then returns a PT. Then, RqP pulls PT and then sends to LAS with identity information for identity verification. LAS returns a VIT to RqP after verification. RqP uses VIT to RAS for identity and authorization check in the cross-jurisdiction. RAS then gives a RPT to RqP for final progress of eHealth data access. RqP shares RPT with RRS to get eHealth data. After permission check, RRS returns the required eHealth data to RqP. The whole data flow terminates

at this point. During transactions progresses, all authorized tokens are pushed to the blockchain firstly and then pulled by entities always mediated by the blockchain. In the blockchain, smart-contracts are triggered by transactions to process tokens sharing. A receipt is always generated following transaction of new information, such as a new token generation and push.

A. Key Management

We develop a variant of the Diffie-Hellman key exchange method to produce secret key, on-the-fly, for token exchange encryption. This provides full accountability of requests and non-repudiation. Whereas writing in a blockchain requires a secret key (tied to the specific blockchain), *reading* from a public blockchain is open and unaccountable. This key, $k_{N,M}$, is generated as in Figure 4.

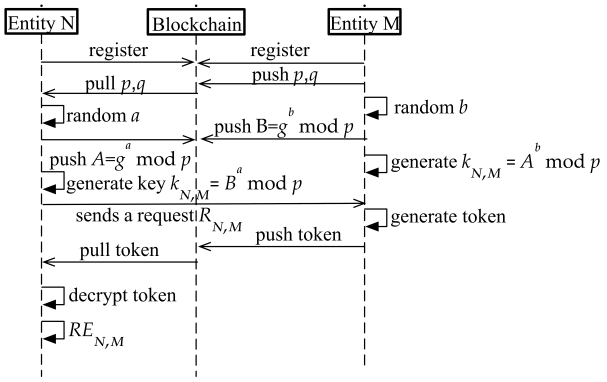


Figure 4: Diffie-Hellman variant over a public blockchain.

Entity M, as a requester, pushes a modulus p and base g to blockchain for entity N. Entity N then pulls them from blockchain. M and N publicly agree to use p and g for key generation. After that, M and N respectively select a secret random integer a or b. N pushes A to blockchain for M and M sends B to N through blockchain. Then, N pulls B from blockchain and generate secret key $k_{N,M}$ with calculation. M does the similar work as N to get a common secret key $k_{N,M}$. Thus, they have a shared secret key $k_{N,M}$.

B. Token Exchange

Tokens are designed to grant access; in the process, a record is generated in the form of a receipt that a party keeps in its possession if later is audited. Entities share tokens with each other only via a public blockchain running smart-contracts – see Algorithm 1. Algorithm 1 presents how an entity shares a token with another entity running smart-contract on a public blockchain. We define two kinds of tokens. One is generated between RO or AS with RqP. When RO or AS generates a new signed token, they send it to blockchain encrypted by key $k_{RO,RqP}$ or $k_{AS,RqP}$. Then, RqP gets it from the blockchain and decrypts it with a shared key $k_{RO,RqP}$ or $k_{AS,RqP}$. Another token is used between RqP and AS or RS. RqP shares an authorisation token PT with LAS, or shares VIT with RAS, or shares RPT with RS through blockchain. Before the token exchange, RqP needs to sign and encrypt the token. For example, a shared token PT between RqP with LAS is delivered as $\text{enc}_{k_{LAS,RqP}}(\text{sign}_{RqP}(PT))$. After the token is pushed, AS or RS pulls the protected tokens from the blockchain and decrypts with key $k_{AS,RqP}$ or $k_{RS,RqP}$.

Algorithm 1 Smart-contract of token exchange

Input: $Token\_Rec$[]
▷ an encrypted signed token reported by a participant
Id_Rec[]
▷ identity information reporting by the recipient to pull token
Output: Indicating if a token is shared successfully

```
if Token_Rec[] !=∅ then
    func(saveToken)            ▷ save Token into blockchain
    return True
end if
Blockchain.push(Id_Rec[])
    ▷ upload identity to blockchain for search and pull token
if Blockchain.push(Id_Rec[]) !=∅ then
    func(getToken)
    for each Id_Rec[] ∈Token_Rec[] do
        while Id_Rec[signature]==Token_Rec[signature] do
            return Token_Rec[]
        end while
    end for
else
    return False
end if
```

C. Receipt Management

Our approach use receipts for transaction records to assure auditability. When a new information (and with a token) is shared between entities, a receipt is generated. For the BRUE, there are two kinds of receipts:

- Receipt for services request. If RqP starts a service request to access eHealth data for DS, a receipt is generated and a hash digest of the request is included.
- Receipt for new token generated and shared. When entity N generates a token and then pushes it into the blockchain, a receipt is returned to N which has the hash value of this transaction.

D. Protocol

The protocol consists of five phases: initialisation, service request, identity verification, authorization verification and resource exchange. See signalling diagram in Figure 5.

1) Initialisation: The initialisation phase is a preparation step that includes entity delegation, key generation, entity registration and receipt wallet (R-wallet) initialisation.

- *delegation* The data subject needs to authorize a resource owner and select some authorization servers and resource servers. RO is only delegated as the representative of DS. DS and RO can be the same individual or different entities. For instance, if DS is a child, RO could be a parent. Furthermore, there is only one AS required

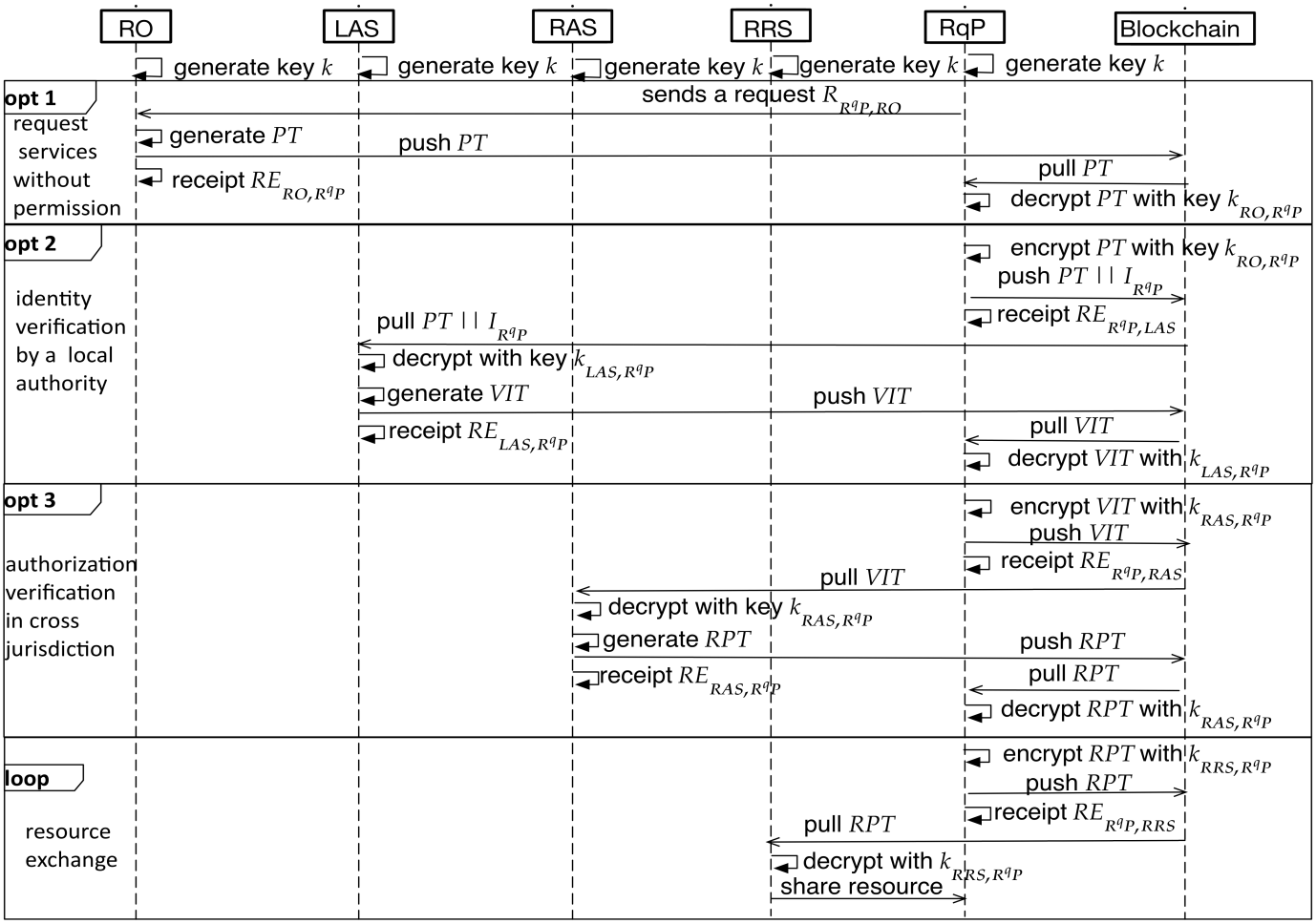


Figure 5: Signalling diagram of data request and exchange.

and some RS in the same region. The number of RS is dependent on the selection from DS.

- *registration* After participants delegation, entities are required to register. This step is executed based on UMA. All entities are required to register in the blockchain and initialise their own R-wallets. An R-wallet is a client application used to aggregate all generated receipts. RS is also required to register in its local AS. Therefore, AS manages a list of RS. In other words, LAS has a list of LRS and RAS has another list for RRS. A receipt is generated to record the registration of RS for auditability principle.
- *key generation* The secret key of an entity is generated using the described variant of Diffie-Hellman key exchange scheme over a blockchain. This key is a common shared between two parties. For RO and RqP, the key is $k_{RO,RqP}$. The key of AS and RqP is $k_{AS,RqP}$ and the key of RS and RqP is $k_{RS,RqP}$.

2) Services request: The service request phase happens between RO and RqP and starts the data flow. RqP starts a service request $R_{RqP,RO}$ to RO without any permission for eHealth data access of DS. A receipt follows. As seen in Figure 5, RO checks $R_{RqP,RO}$ and then generates a token PT. PT is signed and encrypted by RO with a secret key $k_{RO,RqP}$. RO pushes this token PT to blockchain and RqP pulls it from the blockchain. RqP gets the information of permission authorization and AS. At the same time, a consent receipt is produced following the PT push. To note that this transaction is also recorded in the blockchain.

3) Identity verification: To meet the requirement of confidentiality, identity information cannot be shared across jurisdictions. It needs to be conducted by an authority located in the same region with the identity owner. In Figure 5, RqP is required to verify the identity at the local authorization server LAS before any data is accessed. RqP signs and encrypts PT by itself with another secret key $k_{LAS,RqP}$ to share with LAS, following with the identity. In other words, RqP finally shares $PT \| I_{RqP}$ with LAS through blockchain. Then, LAS gets shared information through decryption with key $k_{LAS,RqP}$. When information of $PT \| I_{RqP}$ is verified, LAS sends a new protected token VIT to the blockchain. VIT is an identity verification proof with permission information for RqP. RqP pulls VIT from blockchain and decrypts VIT to get information of RAS and permission. There are two kinds

of receipts involved in it in the purpose of transaction evidence. One receipt is produced when a token PT is pushed by RqP. The other is produced when VIT is pushed by LAS.

4) Authorization verification: Entity RqP starts the request cross-jurisdiction with identity proof token VIT. RqP needs to sign and encrypt VIT with key $k_{RAS,RqP}$ to share with RAS through blockchain. Here a receipt generated for the transaction of token VIT sharing. RAS pulls VIT and decrypts it with key $k_{RAS,RqP}$ for authorization verification. When RAS confirms the information from VIT, RAS shares a new token RPT with RqP for final data access. RPT is signed by RAS and encrypted with key $k_{RAS,RqP}$. After that, RqP pulls and decrypts RPT to confirm and get information of RRS and permission. A receipt is produced for the transaction of a token RPT push by RAS. The authorization verification phase ends.

5) Resource exchange: RqP is now able to access the eHealth data of DS from RRS. RqP shares RPT with RRS with its signature and encryption. A receipt follows token sharing. RRS pulls RPT from blockchain and decrypts with key $k_{RRS,RqP}$ to confirm whether RqP is authorised to access specific eHealth data. Then, RRS shares the protected eHealth data with RqP following a receipt as a transaction record proof.

E. Discussion

We discuss our approach by revisiting our key requirements: auditability, non-repudiation, confidentiality, and compatibility.

1) Auditability: BRUE is designed to audit transactions of entities. Regarding our protocol, it can be expressed in three ways: (i) Data flows need to go through smart-contracts running on a public blockchain and requires token PT. The trace of a transaction is timely recorded in the blocks generated in the blockchain that assures if entities need to view a transaction after the fact. All existing blocks in a chain are practically impossible to modify or destroy because of blockchain properties. (ii) Transaction records require cryptographic signatures which assures authenticity and supports auditability. (iii) A receipt $RE_{N,M}$ is produced as evidence of new information sharing. Receipts are returned to entities after they sent a request or token to others. In the receipt, a digest hash of transaction is embedded. It prevents the modification of the receipt after-the-fact. This signed receipt is also an evidence record for the purpose of transaction auditing.

2) Non-Repudiation: Entities need to sign the exchanged information before a transaction starts. We follow the principle that who shares information is who signs it. This guarantees non-repudiation. Furthermore, the nature of a public blockchain also guarantees non-repudiation since a transaction can only be sent with private keys which are attested at the initialisation phase.

3) Confidentiality: Confidentiality is a key requirement for eHealth data exchange. The encryption operation with a secret key is a first option to target the confidentiality principle. We use the Diffie-Hellman key exchange method to generate a secret shared key for entities which can be made as strong as desired. We also use protected tokens to share with entities, instead of the actual personal data, to minimize the possibility of data disclosure. Besides, tokens require a validity period and can be revoked at any time with immediate effect.

4) Compatibility: BRUE is designed to reuse the OAuth/UMA framework and extended with receipts running on a public blockchain. It relies on an open protocol that is compliance with data sharing protection regulations, such as GDPR. Besides, our proposed protocol provides feasibility to compatibly apply in cross-jurisdiction for eHealth data exchange without requiring a well-known discovering point (Rendez-vous points). This jurisdiction can be a country or a specified federation of some entities or a city. The scope of jurisdiction is flexibly defined by the data subject or resource owner. Numbers of RS are not restricted involved in the protocol for data sharing. RqP can require access to eHealth data from some RS located in some different or one jurisdiction. The compatibility of RS joint and services request for RqP are stronger.

V. Evaluation

In this section, we present and discuss the results of a proof-of-concept implementation of BRUE.

A. Implementation

Our implementation was not designed for performance but only to demonstrate the feasibility and completeness of our proposed approach. We used common desktop hardware (Intel Core i7 at 2.9 GHz with 8GB RAM). All source code is open and available on request. The implementation of *initialisation* phase is not important concerned and introduced in here. For the blockchain, we used Ethereum with the default developer settings. We used Truffle[4], a library to use with web applications. In our proof-of-concept, entities share tokens through Ethereum running smart-contracts. There are four forms of information sharing implemented with smart-contracts involving $R_{N,M}$, PT, VIT, and RPT. The following code shows a smart-contract allowing tokens sharing. It includes two functions. Function $saveToken$ is to receive tokens pushed by an entity and then store it in the blockchain; function $getToken$ is to return token information from blockchain to the recipient. Both functions implement the design of token exchange shown in Algorithm 1.

```
contract Token {
  event NewToken (string signature, string
      receiver, string token);
  struct Token {string signature; string
     receiver; string token;}
  Token[] public token;
  mapping (uint => address) tokenToOwner;
  mapping (address => string) signToToken;
```

[4]Truffle: https://www.trufflesuite.com

```
  function saveToken (string memory
      _signature,string memory _receiver,
      string memory _token) public payable
       returns (bool) {
    uint id = token.push(Token(_signature
       ,_receiver,_token)) - 1;
    tokenToOwner[id] = msg.sender;
    signToken[msg.sender]= _signature;
    emit NewToken(_signature,_receiver,
       _token);
    return true;  }
  function getToken (string memory
      _signature,string memory _receiver)
      public view returns (string memory){
    for(uint i=0;i<token.length;i++){
        Token storage myToken = token[i];
        if(keccak256(bytes(_signature))
           == keccak256(bytes(myToken.
           signature)) && keccak256(bytes
           (_receiver)) == keccak256(
           bytes(myToken.receiver))){
               return myToken.token; }
    } return "wrong input or no record";}
}
```

Figure 6 shows the structure of the user interface as below. We used a web front interface. This interface is implemented by JavaScript codes to achieve operations of entities, which include the generation of tokens and receipts. Three kinds of entities were applied: requesting party, data subject (which was also the resource owner), and authority organisation (equivalent to AS and RS). To note that RO is a delegation of DS in BRUE. Both of AS and RS have a similar role in our approach which is to get tokens from the blockchain, generate new tokens and then share with RqP. Therefore, for our proof-of-concept, we combine both roles. A complete representation of the full data flow is shown in Figure 6. Graph arrow represents the direction of data flow and integer represents the sequence of data flow. We introduce the implementation details of the demo separately by roles as below.

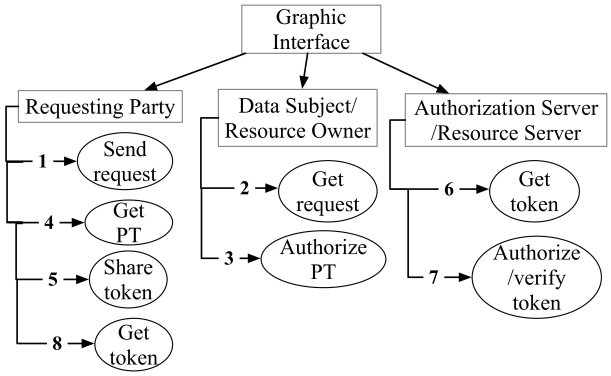
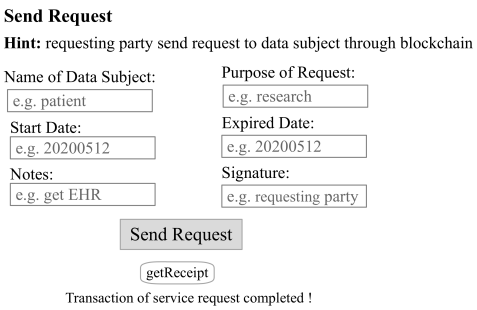


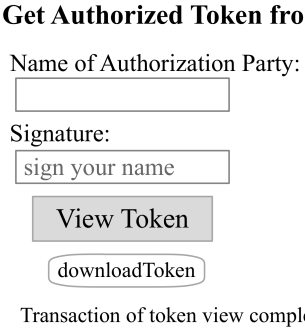
Figure 6: The structure of the proof-of-concept implementation

RqP can send a request without permission, get a token, and share a token. Figure 7 presents screenshots of the interface for RqP. Figure 7a shows a request form for RqP to request authorization of data access. Figure 7b shows how RqP gets a token from the blockchain. Figure 7c shows two operations: one is RqP uploads a token for sharing and the other is RqP shares a token with a signature. A download button of receipt is available for RqP to obtain the corresponding receipt after token sharing.



(a) Interface of service request

(b) Interface of token exchange



(c) Interface of token view and download

Figure 7: Proof-of-concept of BRUE - RqP operations

RO is required to check service request from RqP and then authorize a token for data access. Figure 8 illustrates how RO views the request from RqP and then authorizes this request. If RO permits the request, a signed and encrypted token is generated and shared. RO can download a receipt after its sharing operation.

AS and RS are authorization parties, which get token from blockchain to verify permission or produce a new authorized

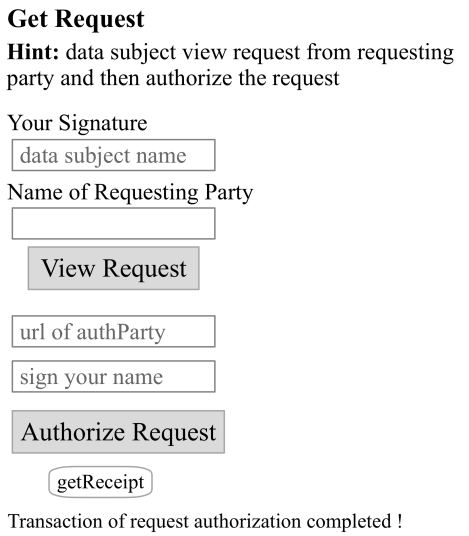


Figure 8: Proof-of-concept of BRUE - RO or DS operations

token for RqP. Figure 9 shows operations of authority party in token receive and authorization. Authority party gets token from blockchain after RqP push (see Figure 9a). Then, a new protected token is generated and uploaded by authority party to share with RqP (see Figure 9b). A receipt record of token sharing is produced and can be download by authority party after the transaction.

B. Discussion

Even though BRUE is technically complex and has several components, its execution can be made simple to non-technical end-users. Whereas our proof-of-concept is not ideal in terms of usability, it does shed light on how simple it can be. Our proof-of-concept also did not show any particular limitation in terms of performance as it is mainly driven by user action. The notion of a wallet of receipts can be simply implemented as a directory with files which are the receipts themselves. This could be converted into a simple mobile application if so desired.

Combining different technologies also proved to be feasible. OAuth/UMA open-source libraries exist which helps future interoperability with eHealth proprietary systems. Diffie-Hellman key exchange scheme is a mature technology to implement key generation. As for the smart-contracts, which we do not discuss in depth given space limitations, they are not very complex and the fact that they use a readily available, and fairly mature, blockchain (Ethereum) shows great promise for real-world trials. We have already shown codes of one smart contract for our proof-of-concept implementation as a sample.

(a) Interface of token view and download

(b) Interface of token authorization

Figure 9: Proof-of-concept of BRUE - AS or RS operations

VI. Conclusions and Future Work

We designed and proposed BRUE for eHealth data exchange and evaluated its feasibility. It relies on open protocols so that it is easier to adopt in existing systems. BRUE seems able to satisfy key requirements such as auditability, confidentiality, non-repudiation, and compatibility when supporting cross-jurisdiction eHealth data exchange. Preliminary results of our prototype show it is not complex to implement and the user-interface can be simple for non-technical users.

Our paper also opens new research directions. On one hand, there essential features that we should be able to accommodate. For example, we need a break-the-glass procedure for emergencies when the data subject cannot give timely consent for resource owner. On the other hand, we did not make any assumptions on the actual eHealth data. Whereas BRUE is agnostic to it, selective disclosure of data, instead of everything, is a crucial feature which requires a data model and fine-grained permissions. Furthermore, the identity of participants is deliberately overlooked. Finally, we did not present a detailed threat model and worked under the assumption that all entities are honest. It is quite possible that BRUE, in its current stage, can be abused or manipulated, particularly in the case of availability and denial-of-service.

Acknowledgment

The authors would like to thank Mark Lizar, Salvatore D'Agostino, Tom Jones, Jan Lindquist and Paul Knowles,

and the wider Kantara ISI WG community, for the insightful comments.

REFERENCES

[1] E. Maler (ed.), "User-managed access (uma) 2.0 grant for oauth 2.0 authorization," *Kantara Initiative Recommendation*, 2018.

[2] M. Lizar and D. Turner (eds.), "Consent receipt specification," *Kantara Initiative Recommendation*, 2017.

[3] S. Angraal, M. H. Krumholz, and L. W. Schulz, "Blockchain technology:application in health care," *Circulation: Cardiovascular Quality and Outcomes*, vol. 10, no. 9, Sep 2017.

[4] P. Zhang, M. A. Walker, J. White, D. C. Schmidt, and G. Lenz, "Metrics for assessing blockchain-based healthcare decentralized apps," in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017, pp. 1–4.

[5] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 62–75, Jun 2019.

[6] T. K. Mackey, T. Kuo, B. Gummadi, A. K. Clauson, G. Church, D. Grishin, K. Obbad, R. Barkovich, and M. Palombini, "'fit-for-purpose?' - challenges and opportunities for applications of blockchain technology in the future of healthcare," *BMC Medicine*, vol. 17, no. 1, p. 68, 2019.

[7] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11 676–11 686, 2018.

[8] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, no. 140, June 2018.

[9] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "Medblock: Efficient and secure medical data sharing via blockchain," *Journal of Medical Systems*, vol. 42, June 2018.

[10] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[11] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016, pp. 25–30.

[12] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, Jan 2018.

[13] X. Zhou, A. Nehme, V. Jesus, Y. Wang, M. Josephs, and K. Mahbub, "Towards blockchain-based auditing of data exchanges," in *Smart Blockchain*. Cham: Springer, 2019, pp. 43–52.

[14] P. Zhang, J. White, D. C. Schmidt, and G. Lenz, "Chapter one - blockchain technology use cases in healthcare," in *Blockchain Technology: Platforms, Tools and Use Cases*, ser. Advances in Computers, P. Raj and G. C. Deka, Eds. Elsevier, 2018, vol. 111, pp. 1 – 41.

[15] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "Fhirchain: Applying blockchain to securely and scalably share clinical data," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 267 – 278, 2018.

[16] HL7, "Fast healthcare interoperability resources (FHIR)," *online: https://www.hl7.org/fhir/overview.html*, 2011.

[17] V. Jesus, "Towards an accountable web of personal information: the web-of-receipts," *IEEE Access,*, vol. 8, 2020.

[18] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.