An Efficient and Distributed Data Storage and Sharing Method Based on Blockchain

Yizhe Sun[1], Shiyou Chen[2], Yadong Fang[3], Wei Xu[3], Qun Luo[1], and Lanlan Rui[2(✉)]

[1] National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China
{yzsun1015,luoqun}@bupt.edu.cn

[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
{chenshiyou,llrui}@bupt.edu.cn

[3] Network Technology Research Center, China Unicom Research Institute, Beijing, China
{fangyd,xuwei}@inspur.com

Abstract. With the development of intelligent medical, electronic medical records (EMR) promote the information management of Internet of Medical Things (IoMT) data. As the need for sharing IoMT data between organizations grows, EMR faces challenges in terms of reliability of the third-party storage, availability of cross-domain authentication, and efficiency of access control in mobile network. In this paper, we propose a blockchain-based medical data sharing model to guarantee the security and efficiency of cross-domain sharing of medical privacy data. Firstly, heterogeneous data of IoMT are classified, and the lightweight data are mixed encrypted and stored in the blockchain full nodes. Then considering the availability and scalability, we introduce the light nodes deployed on mobile edge computing (MEC) platforms to provide distributed access control proxy service. In the CP-ABE access control algorithm, we decouple the relationship between attributes and decryption time, and further design a token mechanism in improved CP-ABE algorithm. Finally, security analysis proves that our model performs well in terms of privacy. Simulation results show that the encryption and decryption performance and key storage space of our CP-ABE access control algorithm are better than traditional algorithms.

Keywords: Blockchain · MEC · IoMT · Data sharing · Access control

1 Introduction

With the development of Internet of Things (IoT) technology, wise information technology of medical (WITMED) has an increasing demand for data sharing. Recently, many medical institutions store and share electronic medical records (EMR) in the centralized cloud, so as to reduce the operation and maintenance costs [1]. However, once data are uploaded into the widely deployed third-party cloud, patients are worried that records may be exposed to malicious entities and even destroyed [2].

Q. Liu et al. (Eds.): CENet 2022, LNEE 961, pp. 421–430, 2022.
https://doi.org/10.1007/978-981-19-6901-0_44

In order to solve the above problems, the decentralization and tamper-proof characteristics of the blockchain can be used to construct a secure storage solution for private data. Blockchain allows different and distrustful gatherings to share data without central supervision [3], building a foundation of trust at a lower cost. But asymmetric encryption in blockchain requires that the same message is encrypted differently for users of different transmission conversations, which brings unnecessary cost, does not provide sufficient support for the scalability of IoT terminals. Ciphertext policy attribute based encryption (CP-ABE) combines encryption and access control, and only encrypts once in the face of different access requests, and is one of the most suitable fine-grained access control technologies in an open environment.

But there are problems that decryption time of CP-ABE increases linearly with the increase of users, and the key storage overhead also brings a huge cost for the system. Besides, the computation-intensive encryption and centralized access control in smart contract cannot support well to blockchain distributed storage. Therefore, we deploy frequent computing services on the edge network to avoid the blockchain performance bottleneck.

In this paper, we propose a lightweight data storage method based on blockchain. Considering the performance of blockchain, we deploy light nodes on the edge computing platform and run improved access control services, reducing routing overhead and request latency. Besides, we introduce token mechanism to reduce the key management work. The rest of the paper is organized as follows: Sect. 2 explains related work. Sect. 3 describes system model and sharing process. Section 3.2 shows security analysis. Section 4 shows simulation results. Section 5 gives conclusion.

2 Related Work

In 2017, UCSD medical center proposed to use cloud computing technology to serve healthcare infrastructure and move its EMR to the cloud hosted by Epic, allowing users to freely share and access their EMR data anytime, anywhere, on any device. Zhang et al. in [4] proposed a medical data sharing scheme on the cloud, used searchable encryption technology to protect data privacy, but the efficiency of encryption and search is poor. [5] and [6] concluded that MEC and fog calculation can be used to handle a large number of security and time sensitive data generated by IoT, which can reduce end-to-end response time.

In addition to using the cloud as a storage medium, many papers considered blockchain as a viable technology to improve data sharing and storage systems due to its tamper-proof, decentralized, and traceable properties [7]. Su et al. [8] designed a secure financial data sharing model by taking advantage of the distributed storage, decentralized management and non-tamper characteristics of blockchain. In this model, access control strategy is deployed on blockchain platform, which solves the hidden danger of financial data security and the problem of financial risk control. In [9], Xia et al. proposed a medical data sharing framework based on blockchain called MedShare, which can track data behavior and reverse access to offending entities in case of violation of data permission. However, the scheme does not consider data from heterogeneous terminals, which is easily cause the performance bottleneck of the blockchain. [10] summarized the impact on blockchain and IoT on the distributed evolution of cloud

computing. It can be seen that the solution for private data sharing is shifting from the early cloud storage to blockchain. At present, most of the access control to data is integrated on the chain, or rely on a unified authentication center. For large and frequent IoT data sharing, this is not a good solution that can take advantage of the distributed and non-tamper characteristics of the blockchain. For blockchain performance issues, [11] suggested that the combination of mobile edge computing and blockchain will become a hot spot.

Data need fine-grained access control in the process of multi-organization sharing. In order to realize fine-grained data access control in the process of sharing, [12] proposed a new attribute-based encryption (ABE) method for ciphertext policy called BSW. Different from traditional role-based access control, ABE implements permission control based on user's own conditions or attributes. However, decryption requires twice for each property and secret key (SK) pair, which is inefficient. Waters proposed a more efficient CP-ABE scheme called Waters [13]. It uses security groups with lower computational cost to reduce decryption time under the assumption of Bilinear Diffie-Hellman Exponent security, but the description of attributes has certain limitations. Chen proposed the CGW scheme [14], which is more secure than Waters and BSW in the asymmetric environment for pairing operation, so it is more difficult to calculate and consumes more time. [15] designed a SPIRC scheme, which supports the revocation of user's specific attributes instead of all the attributes of the user, and there is no need to redistribute the key in the revocation process.

3 System Architecture

This section describes a data sharing method based on blockchain, and also designs cp-ABE access control algorithm in the process of data storage and access.

3.1 System Overview

As shown in Fig. 1, our architecture called IoMTShare can be divided into three layers: mobile terminal, blockchain edge and blockchain storage.

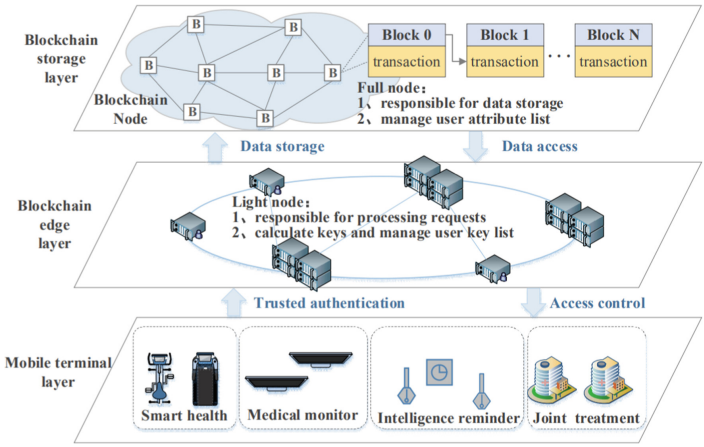

Fig. 1. Architecturel of IoMTShare

The mobile terminal layer uses the IoMT to generate and consume data. The terminal layer has two roles, namely the data requester (DR) and the data owner (DO). DO owns the data and defines the access structure with attribute. Only the DR that meets the access control structure can decrypt the data. The terminal layer can also perform signature verification on messages to ensure the integrity of private data.

The blockchain edge layer contains edge light nodes with interaction and computing capabilities. The edge light nodes are responsible for deal with communication between full nodes and terminals, dynamically calculate key and verifying access control tokens. This architecture places access control computing service at the edge light node closer to the user, and is more friendly to the terminal.

The blockchain storage layer's full nodes maintain the user encrypted EMR data. It also responsible for receiving blockchain edge layer data storage and access requests, and reaching a consensus to prevent malicious nodes from forging and tampering with data.

3.2 The Data Secure Storage Process

(1) System preparation: DO generates public/private key pair PK_{DO} and SK_{DO}. SK_{DO} used for decryption and signature, PK_{DO} used for encryption and authentication. DO brings its public key PK_{DO} and attribute information to apply for a digital certificate. Edge light node executes the CP-ABE Setup steps as shown in Algorithm 1 to generate the PK and MK of CP-ABE for DO.

Algorithm 1: CP-ABE Setup:

Input λ // Safety parameters
Output PK, MK //public key and master key
1 **Select** $par := (p, G, H, G_T, e, g, h) \leftarrow 1^{\lambda}$
2 **Select** $a \leftarrow Z^*_p$, $k, m \leftarrow Z_p$
3 **Compute** $h_1 := h^a$, $h_2 := e(g, h)^{ka+m}$
4 PK= $\{h, h_1, h_2\}$
5 MK= $\{g, h, a, g^k, g^m\}$

The safety parameter is used to calculate the prime order to generate three groups of bilinear pairings. It also generates generators g and h for groups G and H at the same time. The linear secret sharing matrix M has m rows and n columns. Then full node signs DO's public key PK_{DO} and writes them to the user's certificate, along with CP-ABE's PK and attributes A_{dr}. MK is stored and managed by light node. The certificate has a unique ID that maps to the MK and user public key to form a user key list. Finally, light node returns the certificate to the user encrypted with DO's public key. It should be noted that the system preparation phase only runs once. The edge light node registers the UKL of user, in other words, it maps MK, PK_{DO} to CID. The user locally generates symmetric encryption key Kd.

(2) Encryption: The DO's original data are always owned by itself. Under the balance of privacy and service, the condensed text data of EMR are collected and calculate the digest, and then use SK_{DO} to sign it. Then use the symmetric key Kd to encrypt signed data. The symmetric key encryption of data can ensure the efficiency of encryption. Finally, calculate the hash of the key to generate the access control token ACT.

$$DC = E_{Kd}(Sign(Hash(Data), SK_{DO}, t)||Data) \tag{1}$$

$$ACT = Hash(PK_{DO}||Kd) \tag{2}$$

where DC is the encrypted data and t is the timestamp. In addition to encrypting data, the access control algorithm CP-ABE encrypts the key. Exactly, data owner uses the write access structure WT and PK to encrypt PK_{DO} and Kd to obtain the ciphertext WS. Only users who meet the WT can decrypt the encrypted data.

$$WS = Encrypt(PK, PK_{DO}||Kd, WT) \tag{3}$$

Algorithm 2: CP-ABE Encrypt:

Input PK, msg, WT
Output WS
Select $s_1, s_2 \leftarrow Z_p$
Compute $ct_0 := (h_1^{s1}, h_1^{s2}, h^{s1+s2})$
//linear secret sharing calculation on WT
for each $i = 1$ to m and $l = 1$ to 3 **do**
//Linear shared matrix M has m rows and n columns
  for each j=1 to n **do**
    $ct_{i,l} := H(\pi(i)l1)^{s1} \cdot H(\pi(i)l2)^{s2}$
    $\cdot \prod_{j=1}^{n} [H(0jl1)^{s1} \cdot H(0jl2)^{s2}]^{M_{i,j}}$
  end for
  $ct_i := ct_{i,1}, ct_{i,2}, ct_{i,3}$
end for
$ciphertext = h_2^{s1} \cdot h_2^{s2} \cdot msg$
$WS = \{ct_0, ct_1, \cdots, ct_m, ciphertext\}$

(3) Sending request: DO sends an upload request to the edge light node. The upload request contains CID, ACT, WS, and DC. Edge light node maps the ACT and DO's attributes to CID in UPL registration.
(4) The edge light node only saves UKL, so it sends WS, DC and UPL to the connected full node.
(5) Blockchain storage: The full node first needs to verify the validity of the request according to signature, and check the time stamp t. If the difference between t and the current time is more than two round-trip time, it is considered as an illegal request. Then full node parses the CID, UPL, WS and DC in the request. After consensus, full node return result.

(6) Return result: Edge light node receives a successful or failed write message and forwards it to DO.

In summary, access control and symmetric encryption are mixed forms of encryption, which ensures the security and efficiency of data storage. In addition, edge light node can also improve storage service quality.

3.3 The Data Efficient Access Process

(1) DR encrypts its certificate and sends access request with it to the nearest edge light node. The edge light node forwards the information. The full node returns the certificate verification result.

(2) Full node finds the ACT in UPL and WS according to the CID. If DO is in the remote domain, the steps are the same, because the blockchain is synchronized with the entire ledger, and then full node returns the encrypted WS, ACT to the edge full node which forward request information (Fig. 2).

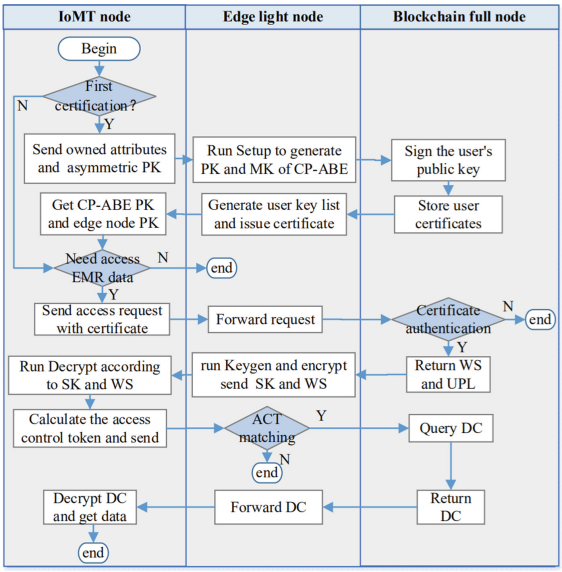

Fig. 2. Data access process of IoMTShare

(3) Edge light node finds the MK of DO in UKL according to the CID in request information, and calculates CP-ABE's SK according to the MK and DR's attribute A_{dr} in the certificate, SK generation step is shown in Algorithm 3, then edge light node sends the dynamically generated encrypted SK and WS from full node to DR.

$$SK = KeyGen(MK_{do,}A_{dr}) \tag{4}$$

Algorithm 3: CP-ABE KeyGen:

Input MK, Attribute
Output SK
1 **Select** $r_1,r_2,r_3 \leftarrow Z_p$
2 **Select** $k_1,k_2 \leftarrow Z_p$
3 $sk_0 := (h^{k_1 r_1}, h^{k_2 r_2}, h^{r_1+r_2})$
4 **for** each $y \in Attribute$
5 **Select** $r_y \leftarrow Z_p$
6 **for** $t = 1$ to 2 //H function maps bit to group G
7 $sk_{y,t} := H(y1t)^{\frac{k_1 r_1}{a}} \cdot H(y2t)^{\frac{k_2 r_2}{a}} \cdot H(y3t)^{\frac{r_1+r_2}{a}} \cdot g^{\frac{r_y}{a}}$
8 $kp_t := g^k \cdot H(011t)^{\frac{k_1 r_1}{a}} \cdot H(012t)^{\frac{k_2 r_2}{a}} \cdot H(013t)^{\frac{r_1+r_2}{a}} \cdot g^{\frac{r_3}{a}}$
9 **end for**
10 $sk_y := (sk_{y,1}, sk_{y,2}, g^{-r_y})$
11 **end for**
12 $kp := (kp_1, kp_2, g^m \cdot g^{-r_3})$
13 $SK=\{sk_0, sk_1, \cdots, sk_y, kp\}$

(4) (DR uses SK and WS sent by edge light node in step 3 to decrypt Kd, PK_{DO} and SK_{DO}. Then DR computes access control token ACT_{DR} and sends it to the edge light node. If the user attributes do not meet the requirements of WT, the decryption key will not be obtained.

$$PK_{DO}||SK_{DO}||Kd = Decrypt(SK, WS) \tag{5}$$

$$ACT_{DR} = Hash(PK_{DO}||SK_{DO}||Kd) \tag{6}$$

Algorithm 4: CP-ABE Decrypt:

Input SK, WS
Output msg
1 $I=\{i|i \in \{1, \cdots, m\}, \pi(i) \epsilon M'\}$ //M' is attribute in SK
2 $U=\{\mu_i\}_{i \in I}$ // Linear shared matrix M in ct
3 **if** $\sum_{i \in I} \mu_i (M)_i = (1, 0, \cdots, 0)$ **then**
4 $GT_1 := ciphertext \cdot e(\prod_{i \in I} ct_{i,1}^{\mu_i}, sk_{0,1})$
5 $\cdot e(\prod_{i \in I} ct_{i,2}^{\mu_i}, sk_{0,2}) \cdot e(\prod_{i \in I} ct_{i,3}^{\mu_i}, sk_{0,3})$
6 $GT_2 := e(kp \cdot \prod_{i \in I} sk_{\pi(i),1}^{\mu_i}, ct_{0,1})$
7 $\cdot e(kp \cdot \prod_{i \in I} sk_{\pi(i),2}^{\mu_i}, ct_{0,2}) \cdot e(kp \cdot \prod_{i \in I} sk_{\pi(i),3}^{\mu_i}, ct_{0,3})$
8 **end if**
9 $msg = GT_1 / GT_2$ //msg is $K_{sign}||K_{verify}||Kd$

(5) Edge light node compares the ACT_{DR} sent by DR with the ACT sent by full node in step (2). If they are consistent, submit a data access request to the full node.

(6) The full node queries the data, and searches the data object in the corresponding index list according to the CID. If the search succeeds, the full node will also record the queried transactions in the ledger for audit.
(7) The full node sends the requested data DC to the edge light node after data auditing.
(8) Edge light node sends DC to DR. DR verifies the signature through SK_{DO} generated in step 4, and use Kd decrypts DC to obtain the requested data.

This section shows how to complete data sharing. The purpose of introducing token mechanism in this section is to reduce key management overhead.

4 Simulation Results and Analysis

In this section, we compare the encryption and decryption time and key storage space with BSW [12], waters [13], CGW [14] and SPIRC [15]. We also compare the service provider request time delay with MedShare and MedBlock. We use 128bit AES symmetric encryption and 1024bit RSA.

4.1 The Time Consuming of Encrypt and Decrypt

As shown in Fig. 3, The left is the encryption comparison, the right is the decryption comparison, and the blue line represents our CP-ABE. In the encryption phase, our time takes less than 1 s to encrypt 50 groups of policy. For decryption phase, BSW, Waters and SPIRC schemes increase almost linearly as the number of attributes increases, but our scheme always takes about 0.08 s because we only need a constant number of pairing operations.

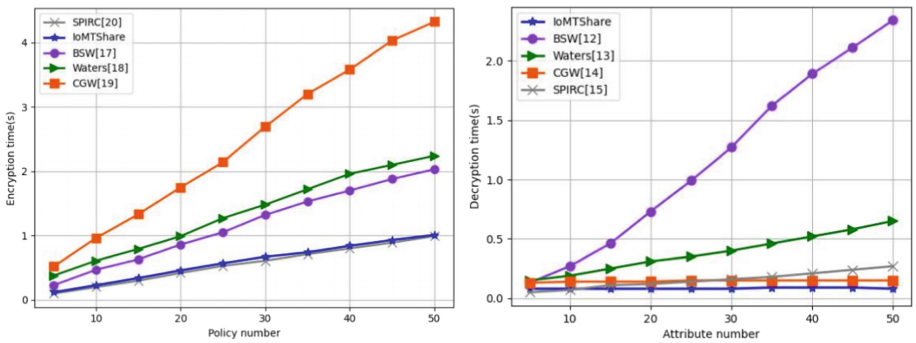

Fig. 3. The encrypt and decrypt time comparison

4.2 Latency of Data Sharing Requests

Different from MedShare [9] and MedBlock [10], our granularity of access control is based on attribute rather than role, so we use R to transform them. When a user has 2 or more attributes, IoMT performs better (Fig. 4).

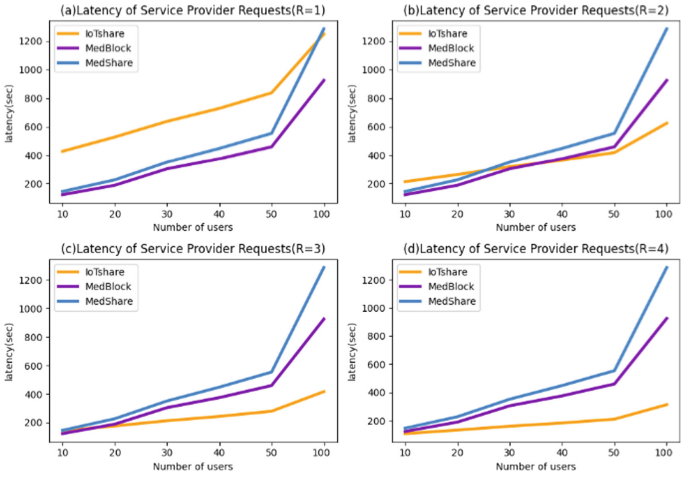

Fig. 4. Scheduling result of multi-level queue window scheduling module

5 Conclusion

In this paper, we design a data sharing architecture based on MEC and blockchain. We further improve the CP-ABE's decryption time. To save key storage space, we introduce a token mechanism. For future work, we'd like to solve attributes revocation question.

Acknowledgements. The work is supported by the Ministry of Industry and Information Technology project entitled "Blockchain based anti-counterfeiting and traceability platform for industrial products".

References

1. Cao, Y., Fan, W., Wang, Y., Yi, K.: Querying Shared Data with Security Heterogeneity. In: 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD 2020), pp. 575–585 (2020)
2. Vincent, R., Dimitri, V., Paolo, V., Bert, L., Riccardo, L., Wouter, J.: Analysis of architectural variants for auditable blockchain-based private data sharing. In: 34th ACM/SIGAPP Symposium on Applied Computing (SAC 2019), pp. 346–354 (2019)
3. Alberto, C., Marco, P., Stefano, S.: Mobile edge cloud network design optimization. IEEE/ACM Trans. Netw. **25**(3), 1818–1831 (2017)
4. Xu, C., Wang, N., Zhu, L.: Achieving searchable and privacy-preserving data sharing for cloud-assisted E-healthcare system. IEEE Internet Things J. **6**(5), 8345–8356 (2019)
5. Subhav, P., Abhishek, D.: CHARIOT: goal-driven orchestration middleware for resilient IoT systems. ACM Trans. Cyber-Phys. Syst. **2**(3), 1–37 (2018)
6. Ashkan, Y., Caleb, F.: All one needs to know about fog computing and related edge computing paradigms: a complete survey. J. Syst. Architect. **98**, 289–330 (2019)
7. Abu-Elezz, I., Hassan, A., Nazeemudeen, A.: The benefits and threats of blockchain technology in healthcare: a scoping review. Int. J. Med. Inform. **142**, 172–182 (2020)

8. Su, Z., Wang, H., Shi, X.: A financial data security sharing solution based on blockchain technology and proxy re-encryption technology. In: 2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI), pp. 462–465. (2020)
9. Xia, Q., Sifah, E.B., Asamoah, K.O.: MeDShare: trust-less medical data sharing among cloud service providers via blockchain. IEEE Access **1**, 124–136 (2017)
10. Fan, K., Wang, S., Ren, Y., Li, H., Yang, Y.: MedBlock: efficient and secure medical data sharing via blockchain. J. Med. Syst. **42**(8), 1–11 (2018). https://doi.org/10.1007/s10916-018-0993-7
11. Zhang, R., Xue, R., Liu, L.: Security and privacy on blockchain. ACM Comput. Surv. **52**(3), 1–34 (2019)
12. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP 2007), pp. 321–334 (2007)
13. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
14. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_20
15. Sethia, D., Saran, H., Gupta, D.: CP-ABE for selective access with scalable revocation: a case study for mobile-based healthfolder. Int. J. Netw. Secur. **20**(4), 689–701 (2018)