Discretionary Access Control Method to Protect Blockchain Privacy

Jie Yang, Min-Sheng Tan(✉), and Lin Ding

School of Computer, University of South China, Hengyang 421001, China
tanminsheng65@163.com

Abstract. Blockchain technology has been widely concerned by scholars and industry since it was put forward, and Banks, Internet of Things, Supply Chain, Government and Medical Industry have proposed using blockchain technology to solve their problems, respectively. However, there are some difficulties in the deployment of blockchain products. One important reason is privacy protection. In order to protect blockchain privacy, discretionary access control method is proposed, and the corresponding model and algorithm are given. Encryption algorithm is used to encrypt the blockchain transaction transactions to privacy transactions. The encryption key and access rights are encapsulated by Lagrange polynomial to form secret information sent to authorized users. Extracting enough secret information, authorized user groups work together to calculate the decryption key, and then obtain the transaction transactions plaintext and finally implement consensus mechanism to verify the transaction. Secret information safely self-destruct immediately if exceed effective time. Authorized users and effective time are entirely determined by the owner of the transaction. This paper realizes key distributed securely, achieves discretionary access control and fine-grained access control and provides strong privacy protection.

Keywords: Privacy · Blockchain · Discretionary access control

1 Introduction

Bitcoin has attracted the attention of many scholars since it was put forward. Blockchain technology is the underlying technology of encrypting encrypted digital currency. It is composed of distributed database system (also known as distributed ledger), peer-to-peer network (P2P) and applications. It has been applied to banking, Internet of Things (IoT), key supply chain, government, medical and other industries. The ledger is completely open, and users' privacy is protected only through virtual name, which hides identity information to some extent.

Privacy in blockchain [1–3] mainly considers transaction privacy and identity privacy. Transaction privacy mainly refers to the content of transaction transactions including transaction amount or transaction mode only accessed by designated users [3]. Identity privacy mainly refers to the inability to track the relationship between participants and infer the relationship between participants' real identity and transactions.

H. Ning (Ed.): CyberDI 2019/CyberLife 2019, CCIS 1137, pp. 161–174, 2019.
https://doi.org/10.1007/978-981-15-1922-2_11

With the increasing of blockchain technology application, the privacy problems revealed behind it become more and more serious. Scholars have found that trading chart analysis [2, 5], network graph analysis [4, 5], trading fingerprint identification [6], DoS attack [7], address clustering [5, 8], Sybil attack [9] and other methods can achieve the de-anonymity of Bitcoin.

In order to protect users' privacy, mixing services divided into centralized mixing services and decentralized mixing services, ring signature and non-interactive zero-knowledge proof were proposed.

The structure of centralized mixing services [3, 10] is shown in Fig. 1.

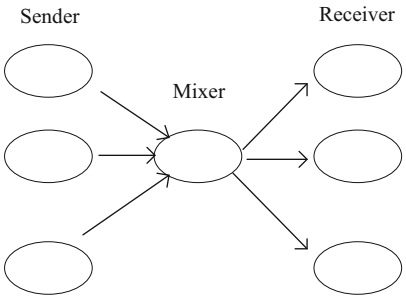

Fig. 1. Architecture of centralized mixing services

The sender encrypts the message(M) with receiver's public key(K_{PR}), encrypts the ciphertext and the receiver's address(R) with the mixer's public key(K_{MR}), then sends it to the mixer. The mixer sends the message decrypted with his private key(K_{MS}) to the receiver. At last the receiver decrypts the ciphertext with his private key(K_{RS}) to get the M. The whole process is expressed as follows:

$$E_{K_{MP}}(E_{K_{RP}}(M), R) \rightarrow D_{K_{MS}}(E_{K_{MP}}(E_{K_{RP}}(M), R)) \rightarrow D_{K_{RS}}(E_{K_{RP}}(M))$$

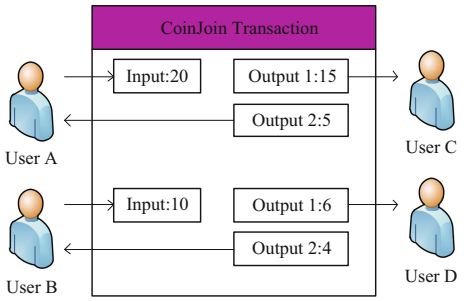

Fig. 2. Decentralized mixing services [3]

Centralized mixing services protect identity privacy, but needed to wait for enough participants to online perform interactive mixing services, so the delay is quite high and the sender needs to pay an expensive cost. Its privacy security depends on the loyalty of the mixer. Decentralized mixing services (as shown in Fig. 2 [3]) is divided into Coinjoin [11] and Secure Multi-Party Computation (MPC) [12], removing third-party mixer, but there is still high latency, don't protect transaction privacy and don't provide fine-grained access control.

Monero [13] represented using ring signature technology uses digital signature technology rather than any central manager, and protect transaction privacy and identity privacy at the same time. However, Feng Q [3] point out that it requires thousands of bytes of space to storage transaction transactions and there only limited external output in actual transactions, as the size of signatures is proportional to the number of participants.

Zerocoin [14] uses non-interactive zero-Knowledge proof to hide the relationship between payment source and transaction to protect users' privacy. Zerocash [15] improves Zerocoin [14], providing both identity privacy and transaction privacy. It achieves strong anonymity and provides the highest level of privacy protection so far, but at the expense of the high computational cost to generate transaction proof.

In order to solve the privacy problem of blockchain, this paper combining encryption algorithm, and authorization strategy proposes security discretionary access control method (SDAC) and implements the corresponding algorithms, which simply and efficiently achieve transaction encryption, fine-grained autonomous access control and access time control, thus realizing strong privacy protection of blockchain. The contributions of this paper are as follows:

- An effective authorization method is designed to realize blockchain discretionary access control.
- The number of elements in the authorized users' set can be controlled by trader to realize fine-grained access control.
- For the first time, the simple and efficient algorithms are used to encrypt blockchain transaction transactions while protecting both transaction privacy and identity privacy.

2 Basic Knowledge for SDAC

2.1 Threshold Secret Sharing

Secret sharing is an important issue in the field of information security, and an important method for key management. Shamir [16] first proposes threshold secret sharing method (t, n) ($0 < t \leq n$). Secret S is divided into n parts, which are shared by n participants. Each participant keeps one part. Only when more than t participants cooperate with each other, the secret S can be recovered. When less than, it cannot.

2.2 Lagrange Polynomial

A polynomial in the form:

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 \tag{1}$$

is called Lagrange polynomial. N different points can reconstruct an n − 1 Lagrange polynomial:

$$F(x) = \sum_{k=0}^{n-1} l_k(x)y_k \tag{2}$$

where,

$$l_k(x) = \prod_{0 \le j \le n-1, j \ne k} \frac{x - x_j}{x_k - x_j} \tag{3}$$

3 SDAC Core Goals and Related Assumptions

In this section, the basic definitions, core goals and assumptions of SDAC are given in turn.

3.1 SDAC Basic Definition

(1) Definition 1. Privacy Transactions: It denotes an encrypted data structure consisting of input and output. Detailed data structure is given in Sect. 5.1.
(2) Definition 2. Blockchain Self-Destroy Object (C_{sdo}): It denotes an encrypted string used to encapsulate secret information and reconstruct the decryption key, and may be leaked during the transmission in P2P network.
(3) Definition 3. Validity Period: It denotes a lifetime and authorized users can access data objects, during it, but beyond the critical point, immediately cannot. SDAC method protects blockchain privacy security during and after the lifetime of self-destructive objects.

3.2 SDAC Core Goals

(1) Discretionary Access Control. It refers to that the owner of transaction transactions decides to authorize users who can scan it without relying on any other entity or user, but unauthorized users cannot.
(2) Fine-grained Access Control. A transaction can be visited by a user set, and different transaction transactions can be visited by different user sets. The number of elements in the authorized user sets can go from zero to any value.

(3) Strong privacy protection. Protect both transaction privacy and identity privacy at the same time.
(4) Low Computability and Efficiency. Transaction generated is more efficient than Monero. Computational cost of SDAC method is lower than Zerocash.

3.3 SDAC Assumptions

(1) Blockchain Client Security and Trusted. Blockchain client can execute script program and consensus mechanism correctly.
(2) Authorized Users Trusted. Authorized users are related to transaction. In order to ensure their own benefits, they are credible during the validity period.
(3) Cipher and Cryptographic Algorithm Trusted. They are the basis of this paper.

4 SDAC Model and Description

4.1 SDAC Model

The SDAC model is shown in Fig. 3. The model consists of three entities: privacy owner, users of the system and potential attackers.

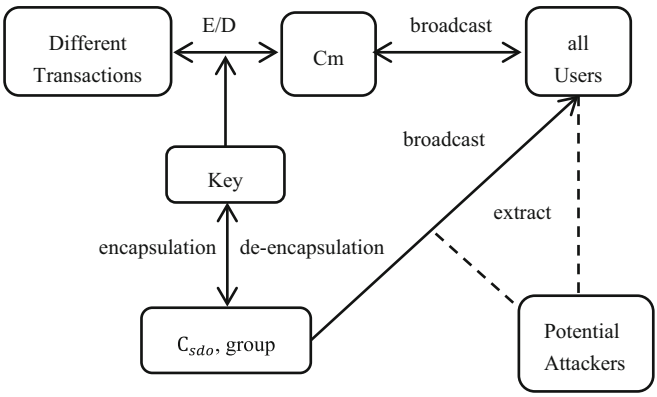

Fig. 3. SDAC model

- Privacy Owner: Sender and receiver of transaction.
- All Users: All participants in blockchain system, which are divided into authorized users and unauthorized users. Authorized users are allowed to access privacy transaction transactions, responsible for verifying transactions and implementing consensus mechanism. The $R = \{r_i | i \in N\}$ denotes the set of authorized users, which can be encrypted to *group*.

- Potential Attackers: All users in the system are likely to be attackers at different time. Authorized users attempt to save C_{sdo} to recover the decryption key at any time. Other users attempt to launch various attacks on P2P networks and authorized users in order to get C_{sdo}.

The first stage: Transaction transactions encrypted, key encapsulated and distributed phase (corresponding to the right and down direction's arrow in Fig. 3). The privacy owner divides the transaction transactions according to different access rights, encrypts the transactions with the key to form a privacy transaction and broadcasts it to the blockchain system. The encryption algorithm uses the Advanced Encryption Standard (AES). The privacy transactions with the same access rights use the same key and with different access rights use different keys. The keys are encapsulated to form C_{sdo} and then send the C_{sdo} to authorized users.

The second stage: Key de-encapsulated, transaction transactions decrypted, and consensus mechanism implemented phase (corresponding to the arrow up and left in Fig. 3). This stage mainly is traversed phase by the authorized users (reverse process of the first stage). Authorized users get decryption key through a series of processing, then decrypt transaction transactions to plaintext and then implement consensus mechanism.

4.2 SDAC Description

Transaction Establishment: The traders make a deal, give the parameter τ, call setup(τ) to generate the plaintext of the transaction transactions locally.

Generate the Key: The traders give the security parameter κ and Ke which is the receiver key of the previous round of generating privacy transactions, call setkey(κ, Ke), generate the receiver key Ke', and at last generate the secret information S which is presented access rights.

Generate Privacy Transactions: Call Esecret(m, S), encrypt the transaction transactions plaintext into Privacy Transactions Cm and broadcast it to the whole blockchain system.

Authorizing Access: Giving the R, threshold δ and time stamp t, call SDACAR(R, δ, S, t) to get C_{sdo}, call SDACE(R) to get *group* and then broadcast C_{sdo} and *group* to the whole blockchain system. Because secret S has been encrypted by access user's public key, unauthorized user(s) cannot decrypt it, thus achieving strong discretionary access control.

Transaction Authentication: Authorized user(s) take out Hash(*source*), if it is the same as some transaction's hash, then continue to perform the following operations, otherwise, judge it is false transactions and vote to refuse it. Look up the global bookkeeping, if the Hash(*source*) is the same as Hash(*source*) of other privacy transactions, judge it is double-spent attack and vote to refuse it too. Call SDACgetkey($C_{sdo}, group$) function, reconstruct secret S, call decryption algorithm, get transaction plaintext, and then implement consensus mechanism.

5 SDAC Algorithms

SDAC combines blockchain, authorization strategy and information encryption to achieve security discretionary access control method for blockchain. The main symbols and description of SDAC algorithm are shown in Table 1.

Table 1. Main symbols in SDAC Algorithms

Symbol	Description	Symbol	Description	Symbol	Description
In	Transaction input	Out	Transaction output	E	Encryption algorithm
Hash	Hash function	Ke/Ke′	Key	D	Decryption algorithm
Change	Change output	sign	Transaction's signature	*ni*/*Ni*	Input/output amount

5.1 SDAC Data Structure

For simplicity, the transaction transactions plaintext m in SDAC is similar to Bitcoin, as shown in Fig. 4. The ini (i < N+) is the total benefit of UserA through every mining or trading. The input and output meet:

$$\text{sum}(ni) = sum(Nj) + num\,(i, j \in N+) \tag{4}$$

The m is consist of $\text{Hash}(source), Time, ni, Sign, Fi, Ni, pb, Ti$, namely:

$$m = (\text{Hash}(source), Time, ni, Sign, Fi, Ni, pb, Ti) \tag{5}$$

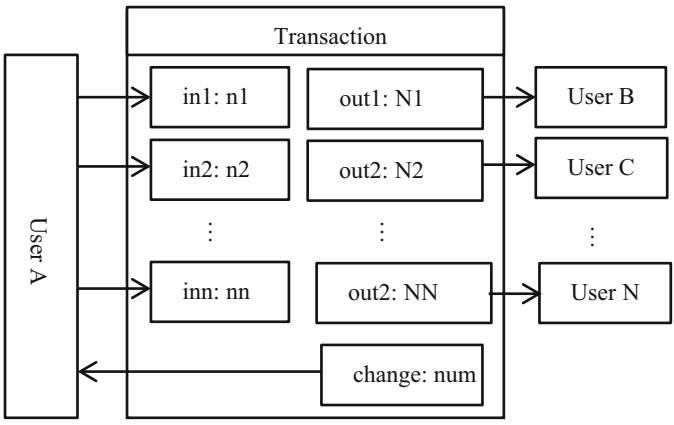

Fig. 4. Transactions

where, Hash($source$) denotes the hash value of the source, *Time* denotes the time when the transaction transactions is generated, Sign is the sender's signature of the transaction transactions, *Fi* denotes that the input comes from the *Fi*th output of the previous transaction transactions, *pb* denotes receivers' public key, *Ti* denotes that this is the *Ti*th output.

Cm denotes Privacy transactions, which consist of input and output. The input data structure of *Cm* in SDAC method is shown in Fig. 5. The output structure of *Cm* is shown in Fig. 6. $E_{Ke}()/E_{Ke'}()$ denotes that using key $Ke/Ke^{'}$ to encrypt.

Hash($source$)	$E_{Ke}(Time)$	$E_{Ke}(ni)$	$E_{Ke}(Sign)$	$E_{Ke}(Fi)$

Fig. 5. Structure of input

$E_{Ke'}(Ni)$	$E_{Ke'}(pb)$	Ti

Fig. 6. Structure of output

5.2 SDAC Algorithms Constructions

(1) Transaction establishment algorithm: setup$(\tau) \rightarrow$ m.
(2) Randomized algorithm: Random$() \rightarrow (0,1)^{\lambda}$.
(3) Key generation algorithm: setkey$(\kappa) \rightarrow$ S.

---

setkey$(\kappa, Ke) \rightarrow S$

Input: κ, Ke
Output: S

1 Random( ) $\rightarrow Ke^{'}$
2 $S = (Ke, Ke^{'})$

---

(4) Privacy transactions generation algorithm:Esecret$(m, S) \rightarrow Cm$.

Esecret$(m, S) \rightarrow Cm$

Input: m, S
Output: Cm
$Cm = (\text{Hash}(source), \text{E}_{Ke}(Time), \text{E}_{Ke}(ni), \text{E}_{Ke}(Sign), \text{E}_{Ke}(Fi), \text{E}_{Ke'}(Ni), \text{E}_{Ke'}(pb))$

1 $\text{E}_{Ke}(Time)$
2 $\text{E}_{Ke}(ni)$
3 $\text{E}_{Ke}(Sign)$
4 $\text{E}_{Ke}(Fi)$
5 $\text{E}_{Ke'}(Ni)$
6 $\text{E}_{Ke'}(pb)$

---

(5) Authorized user sets encryption algorithm: SDACE$(R) \rightarrow group$.

SDACE$(R) \rightarrow group$

Input: R
Output: $group$

1 Random( ) → key
2 $R = \text{E}_{key}(R)$

---

(6) Authorization algorithm: SDACAR$(R, \delta, S, t) \rightarrow C_{sdo}$

SDACAR$(R, \delta, S, t) \rightarrow C_{sdo}$

Input: R, δ, S, t
Output: C_{sdo}
$C_{sdo} = \{CS_i | i \in N^* \wedge i \leq \| \text{ R } \|\}$

1 **for** each r_i **do**
2 $f_i(pb_i) = a_{\delta-1} pb_i^{\delta-1} + \cdots + a_1 pb_i + S$
3 **end for**
4 $d_i = \text{EECC}_{pb_i}(\text{f}_\text{i}(pb_i), pb_i, key)$
5 $CS_i = (d_i, t)$

---

(7) Key recovery algorithm: SDACgetkey$(C_{sdo}, group) \rightarrow S$

SDACgetkey$(C_{sdo}, group) \rightarrow S$

Input: $C_{sdo}, group$
Output: S
pr denotes private key

1 get current time ct
2 **if** ct<t
3 **for** each r_j **do**
4 **for** each CS_i **do**
5 $((\mathrm{f_i}'(pb_i), pb_i', key') = DeECC_{pr_i}(d_i)$
6 **if** $pb_j = pb_i'$
7 key $= key'$
8 $\mathrm{f_j}(pb_j) = \mathrm{f_i}'(pb_i)$
9 $R = \mathrm{D}_{key}(group)$
10 **end if**
11 **end for**
12 **end for**
13 **for** each r_i **do**
14 $l_i(0) = \prod_{0 \le k \le \delta-1, i \ne k} \frac{-x_i}{x_k - x_i}$
15 $S = \sum_{i=0}^{\delta-1} \mathrm{f_i}(pb_i) l_i(0)$
16 **end for**
17 **else**
18 delete $C_{sdo}, group$
19 **end if**

(8) Plaintext recovery algorithm: Dsecret$(Cm, S) \rightarrow m$

Dsecret$(Cm, S) \rightarrow m$

Input: Cm, S
Output: m

1 $\mathrm{D}_{Ke}(\mathrm{E}_{Ke}(Time))$
2 $\mathrm{D}_{Ke}(\mathrm{E}_{Ke}(ni))$
3 $\mathrm{D}_{Ke}(\mathrm{E}_{Ke}(Sign))$
4 $\mathrm{D}_{Ke}(\mathrm{E}_{Ke}(Fi))$
6 $\mathrm{D}_{Ke'}(\mathrm{E}_{Ke'}(Ni))$
7 $\mathrm{D}_{Ke'}(\mathrm{E}_{Ke'}(pb))$

6 SDAC Comprehensive Analyses

SDAC method encrypts transaction transactions' data structure of blockchain. Comparing with other methods, we analyzed the advantages and disadvantages of this paper shown in Table 2.

Table 2. SDAC compared with other schemes

Method	Strong privacy	Delay	Key/evidence management	Discretionary access control	Fine-grained access control
Coinjoin	No	High	Complex	No	No
MPC	Yes	High	Simple	No	No
Menoro	Yes	High	Complex	No	No
Zerocash	Yes	High	Complex	No	No
SDAC	Yes	Lower	Complex	Yes	Yes

(1) Security. The security of SDAC depends on the security of cryptography. An attacker may capture C_{sdo} when it is in the P2P network communication and launch a Sybil attack. Even if achieving the attack, the attacker can recover the S using ECC decryption algorithm only if he gets at least δ different authorized users' private key and then uses AES decryption algorithm and S to get transaction transactions plaintext. Both encryption methods can effectively resist known attacks. According to modern cryptography, Both AES algorithm and ECC algorithm can resist the existing attacks under the existing conditions if the private key doesn't be leaked. So SDAC method is security.

(2) Strong privacy. Strong privacy need consider both transaction privacy and identity privacy.

For the sender:

$$\mathrm{E}_{Ke}(Time, ni, Sign)$$

For the receiver:

$$\mathrm{E}_{Ke'}(Ni, pb)$$

So, the transaction transactions content and transaction amount are hidden, that is SDAC method provides transaction privacy protection.

For the keys:

$$S = (Ke, Ke^{'})$$

$$f_i(pb_i) = a_{\delta-1}pb_i^{\delta-1} + \cdots + a_1 pb_i + S$$

Where, $i \in \{i|0 \leq i \leq \ \| \mathrm{R} \|\}$.

$$\forall f_i(pb_i), \exists \mathrm{EECC}_{pb}(f_i(pb_i))$$

For different transaction:

$$S \neq S^{'}$$

$$f_i(pb_i) \neq f_i^{'}(pb_i)$$

$$\forall pb_i, \exists \mathrm{EECC}_{pb}(f_i(pb_i)) \neq \mathrm{EECC}_{pb}^{'}(f_i(pb_i))$$

To every user, different transaction has different keys, and encrypted amount is different. So, the relationship between participants can't be tracked and the relationship between participants' real identity and transactions can't be inferred, that is SDAC method provides identity privacy protection.

So, SDAC method achieves strong privacy protection.

(3) Delay. Transactions do not require third party trusted institutions, and there is no waiting delay caused by online mixing. There is no need to compute public functions used for encryption, so there is no computation delay caused by multi-party secure computing. There is no no computation delay caused by calculating evidence of non-interactive zero knowledge. The owner only needs to sign the transactions once using the Elliptic Curve Digital Signature Algorithm (ECDSA), so the efficiency of signature is higher than that ring signature. The owner encapsulates the key to $\mathrm{C_{sdo}}$ with Lagrange polynomial and then encrypts it into $\mathrm{C_{sdo}}$ by ECC, and authorizes the user de-encapsulates $\mathrm{C_{sdo}}$ and decrypt $\mathrm{C_m}$, so calculation delay is caused.

(4) Key management. With SDAC method, using different secret key every time, the owner need store a large number of keys, which caused complex key management. In order to reduce it, they can delete it after transaction transactions comfirmed.

(5) Discretionary access control and fine-Grained access control.

$$\mathrm{E}_S(m) = C_m$$

$$f_i(pb_i) = a_{\delta-1}pb_i^{\delta-1} + \cdots + a_1 pb_i + S$$

$$\mathrm{EECC}_{pb_i}(f_i(pb_i)) = f_i^{'}(pb_i)$$

$$\forall r_i \in R, \exists \mathrm{DECC}_{pr}\left(f_i^{'}(pb_i)\right) = f_i(pb_i)$$

$$\forall r_j \notin R, \mathrm{DECC}_{pr}\left(f_i^{'}(pb_i)\right) = f_j\left(pb_j\right) \neq f_i(pb_i)(j > 0)$$

$$S^{'} = \sum_{j=0}^{\delta} l_j(pb_j) f_j(pb_j)$$

$$f_j(pb_j) \neq f_i(pb_i)$$

$$S^{'} \neq S$$

$$\mathrm{D}_{S'}(C_m) \neq m$$

Namely, unauthorized users cannot scan transaction transactions.

$\forall r_j \in R$ is determined by the owner, so SDAC can achieve discretionary access control.

$$\| R \| \geq 0 \wedge \| R \| \in N$$

Namely, SDAC can achieve fine-Grained access control.

(6) Low computability and efficiency. The computability and efficiency of SDAC method depend on the computability and efficiency of cryptographic algorithm. The method only use ECDSA once, needn't sign by all participants, which have to in ring signature method, so the size of storage space is smaller than it and the output of transaction transactions don't limit by the number of participants. The method only uses AES algorithm to encrypt and decrypt transaction transactions and authorized users set once, only uses ECC algorithm to encrypt and decrypt C_{sdo} once, respectively. AES algorithm has the characteristics of high efficiency and fast speed. ECC algorithm operates on keys and secret components, having with only a few bytes. From the perspective of cryptography, under the same conditions, the AES algorithm and ECC algorithm working together are two orders of magnitude better than non-interactive zero-Knowledge proof in terms of computational complexity and efficiency.

7 Conclusion

Based on blockchain privacy protection, combining AES, ECC encryption algorithm, ECDSA, Lagrange polynomial and authorization strategy, this paper proposed the SDAC method and implements the corresponding algorithms. SDAC method encrypts the transaction transactions data structure of blockchain, and then transmit the encrypted and encapsulated key to authorized users. They decrypt and de-capsulate to get the key then recover the transaction transactions to implement consensus mechanism. SDAC method can realize fine-grained access control, discretionary access control transaction transactions and strong privacy of blockchain, thus achieving the goal of design protection. Compared with Menoro, SDAC method is more efficient to generate transactions. Compared with Zerocash, its computational cost is lower. The next step is to reduce the complexity of key management.

Acknowledgment. This work was supported by Project 61403183 of the National Science Foundation of China, Project 2017JJ4048 of the Hunan Provincial Natural Science Foundation of China, Project 18A230 of the Hunan Provincial Education Office Science Research of China. Project 20183350502 and 20191550502 of the Hunan Provincial Finance Office Science Research of China.

References

1. Bertram, S.: A privacy-preserving system for data ownership using blockchain and distributed databases. arXiv preprint arXiv:1810.11655 (2018)
2. Chen, L.: Unraveling blockchain based crypto-currency system supporting oblivious transactions: a formalized approach. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, pp. 23–28. ACM, Abu Dhabi (2017)
3. Feng, Q.: A survey on privacy protection in blockchain system. J. Netw. Comput. Appl. (2018)
4. Koshy, P., Koshy, D., McDaniel, P.: An analysis of anonymity in bitcoin using P2P network traffic. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 469–485. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_30
5. Reid, F.: An analysis of anonymity in the bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) Security and Privacy in Social Networks, pp. 197–223. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
6. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 34–51. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_4
7. Biryukov, A.: Bitcoin over tor isn't a good idea. In: 2015 IEEE Symposium on Security and Privacy, pp. 122–134. IEEE (2015)
8. Liao, K.: Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In: 2016 APWG Symposium on Electronic Crime Research (eCrime), pp. 1–13. IEEE (2016)
9. Bissias, G.: Sybil-resistant mixing for bitcoin. In: Proceedings of the 13th Workshop on Privacy in the Electronic Society, pp. 149–158. ACM (2014)
10. Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. In: Gritzalis, D.A. (ed.) Secure Electronic Voting, pp. 211–219. Springer, Boston (2003)
11. Joinmarket-Coinjoin. https://bitcointalk.org/index.php?topic=919116. Accessed 2016
12. Ziegeldorf, J.H.: Coinparty: secure multi-party mixing of bitcoins. In: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, pp. 75–86. ACM (2015)
13. Monero Homepage. https://getmonero.org/. Accessed 10 Sept 2019
14. Miers, I.: Zerocoin: anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy, pp. 397–411. IEEE (2013)
15. Sasson, E.B: Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, pp. 459–474. IEEE (2014)
16. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)