SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2019, Vol. 62 032109:1–032109:12
https://doi.org/10.1007/s11432-018-9611-3

A regulated digital currency

Yanbing WU[1*], Haining FAN[2], Xiaoyun WANG[1] & Guangnan ZOU[3]

[1]*Institute for Advanced Study, Tsinghua University, Beijing* 100084, *China;*
[2]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China;*
[3]*Space Star Technology Co., Ltd, Beijing* 100086, *China*

Received 13 June 2018/Accepted 13 September 2018/Published online 29 January 2019

Abstract The decentralized digital currency Bitcoin is an anonymous alternative to the centralized banking system and enjoys widespread and increasing adoption. Since Bitcoin created, many other electronic currencies have been developed. We propose a protocol for an electronic currency for making anonymous payments that can be supervised by an auditor, who has sole access to the transaction history. Other electronic currencies provide only anonymity, which is convenient for making illegal transactions without regulation. For users, miners, and banks, the transactions of our electronic currency are anonymous, and only auditors can see how it is used. We make use of POW (prove-of-work) technique that allows for distributed decision-making within a network, namely the Bitcoin blockchain protocol. We combine the POW and blockchain technology of Bitcoin to give better protection against double-spending attacks.

Keywords Blockchain, Bitcoin, anonymous, digital currency, regulated, proof-of-work

Citation Wu Y B, Fan H N, Wang X Y, et al. A regulated digital currency. Sci China Inf Sci, 2019, 62(3): 032109, https://doi.org/10.1007/s11432-018-9611-3

1 Introduction

In 2008, Nakamoto [1] proposed the digital currency Bitcoin as an anonymous alternative to the central banking system. Since then, Bitcoin has been widely known. For example, in July 2017, it had a market value of more than 5 billion. To track the balance and build confidence in the currency, all currency transactions are stored in a distributed public ledger, namely the blockchain, instead of using a centralized entity, such as bank. To receive, store, and use Bitcoin, people maintain encrypted identities called addresses, which correspond to the public keys of the elliptic curve digital signature algorithm (ECDSA). Addresses and transactions are anonymous, and if addresses cannot be linked to their owners, they can usually be guaranteed by using new, unlinked addresses. This commitment to financial privacy, in particular, has sparked intense interest in Bitcoin.

However, recent studies have raised serious questions about Bitcoin's built-in privacy protection. They show that a user's transactions and address can be linked together by analyzing transaction diagrams in the exposed blockchain [2–5]. To make matters worse, Ref. [6] shows that addresses can be linked to IP addresses, thus completely eliminating the anonymity of their owners. This raises the question that whether a user, Alice, can re-establish her financial privacy. Of course, Alice can generate a new, unlinkable address, but moving her money to it will link the new address back to her old address. Therefore, Alice needs a way to send funds from an anonymous address to another address, in an unlinkable way. So far, many commercial hybrid services have been set up to help Alice. For example, she can sends her money to a mixed service, which after deducting a small fee, will mix her money with the money of other

* Corresponding author (email: wuyb14@mails.tsinghua.edu.cn)

random users and return it to her. Therefore, no passive observer can trace the ownership of an address through the blockchain, unlike the de-anonymizing method described in [2–5]. Note that a passive observer cannot even distinguish between other contemporaneous unmixed trades in the blockchain. This is desirable because it provides a lot of anonymity and allows people to reasonably deny their involvement in the mix.

The first generation of hybrid products, however, has two serious flaws that the Bitcoin community is familiar with. First, users need to believe blindly that operators will not steal their money. In fact, ther are many allegations of theft in the Bitcoin community. Second, hybrid services know and may keep records on how funds are used. Government agencies may then violate, incentivize or force the disclosure of these written records [7, 8]. These daunting shortcomings have led to new decentralization methods that guarantee greater security and anonymity [9]. Most of these methods require that all mixed transactions are made in a single atomic transaction with multiple inputs and outputs. This prevents malicious peers from suspending the agreement after receiving their funds, thus leaving other peers unpaid. Form the characteristics of the group, however, trading enmasse in the chain is very easy to recognize, which introduces two serious limitations for a mixed service. First, the user's anonymity is limited by the number of users who are participating in the same hybrid transaction. Second, the bundled mixed transactions are obvious in the blockchain, depriving users of any possible way to deny their involvement in the mix.

Several academic papers [5, 10], and blockchain developers [11, 12] have proved the weakness of the anonymity of Bitcoin. As a result, the community has responded to this, by putting forward two key ways to improve the currency's anonymity: (1) new anonymity schemes compatible with Bitcoin [9,13–19] and (2) new anonymous encrypted currencies independent of Bitcoin [16, 20]. As we will see, some of these developments provide efficient solutions [9,15,17–19], some achieve limited security and anonymity, and some provide powerful anonymity but are slow [13, 14, 21].

Zerocash [16] and Zerocoin [20] provide anonymous payments with a novel type of cryptographic proof (zk-SNARK). The disadvantage of Zerocash is that the computing resources required for zk-SNARK are very high, making it unsuitable for practical use. Monero [22] uses a technology called ring signing in which transactions are copied to multiple users so they all appear valid. That makes it extremely difficult to track the source of a coin. Ref. [23] proves that Monero is not really anonymous.

In this paper, we propose a protocol for an electronic currency for making anonymous payments that can be supervised by an auditor, who has sole access to the transaction history. Other electronic currencies provide only anonymity, which is convenient for making illegal transactions without regulation. For users, miners and banks, the transactions of our electronic currency are anonymous, and only auditors can see how it is used. Based on anonymity and regulation, our electronic currency protocol can also prevent double-spending attacks. In the rest of this paper, we provide an overview of some of the technologies that we use as building blocks for our protocol, followed by a detailed description of our system, and a brief security analysis.

2 Blind signature

Many public key signature protocols have blind signature schemes. Here are some examples. In each case, the message to be signed is contained in m and is considered some legitimate input to the signature function. For example, consider that Alice has a letter that should be signed by Bob, but Alice does not want Bob to know what is in the letter. She could put the letter and some carbon paper in an envelope and send it to Bob. Bob would sign the outside of the envelope without opening it, and send it back to Alice. Then Alice could open it to get the letter signed by Bob via the carbon paper.

Let m be the message that Alice want Bob sign, and (PK, sk) be Bob's asymmetric encryption/ decryption key pair. r is a random element chosen from the group M. Alice encrypts r using Bob's public key to form the blinding factor (PK(r)). She computes the product of the serial number with this

blinding factor to form the blinded coin serial number:

$$m' = m \cdot \mathrm{PK}(r).$$

Bob in turn signs the blinded serial number with its private key:

$$s' = \mathrm{sk}(m').$$

He returns the coin to Alice, who removes blinding factor:

$$s = s' \cdot r^{-1}.$$

Alice now has a coin signed with Bob's private key:

$$\begin{aligned} s &= \mathrm{sk}(m') \cdot r^{-1} \\ &= \mathrm{sk}(m \cdot \mathrm{PK}(r)) \cdot r^{-1} \\ &= (\mathrm{sk}(m) \cdot \mathrm{sk}(\mathrm{PK}(r))) \cdot r^{-1} \\ &= (\mathrm{sk}(m) \cdot r) \cdot r^{-1} \\ &= \mathrm{sk}(m) \cdot (r \cdot r^{-1}) \\ &= \mathrm{sk}(m) \cdot 1 \\ &= \mathrm{sk}(m). \end{aligned}$$

3 Bitcoin

Bitcoin is an encrypted currency and a payment system. It was invented by an unidentified programmer or a group of programmers under the name Satoshi Nakamoto. Bitcoin was introduced into a crytography mailing list on October 31, 2008, and was released as open source software in 2009. There are various theories and speculations about Nakamoto's identity, but none has been confirmed. The system is peer-to-peer and trades are made directly between users without intermediaries. These transactions are verified by network nodes and recorded in a public distributed ledger called the blockchain, which uses Bitcoin as the unit of account. Because the system operates without a central repository or a single administrator, the U.S. Treasury classified Bitcoin as a decentralized virtual currency. Bitcoin is often called the first cryptocurrency, although there were previous systems. It is more accurately described as the first decentralized digital currency. Bitcoin is the largest of its kind.

4 Serial number of a coin

According to the method shown in Figure 1, a random seed is used by an auditor to generate many public and private key pairs such as $(\mathrm{sk}_1, \mathrm{PK}_1), (\mathrm{sk}_2, \mathrm{PK}_2), \ldots, (\mathrm{sk}_i, \mathrm{PK}_i)$. The resulting public–private key pairs match each other, meaning that the information encrypted using PK_i can be decrypted using sk_1 to get the plaintext information.

We propose a concrete solution for generating public-private key pairs, as shown in Figures 2 and 3. The purpose of our design is that users can encrypt the trading history of a coin with a public key $\mathrm{PK}_{\mathrm{serial}_i}$, The ayditor has the corresponding private key $\mathrm{sk}_{\mathrm{serial}_i}$, which it can use to decrypt and view the transaction history. According to the design, the public key is the serial number of the coin. Moreover, the auditor does not need to save all the coin's private keys $\mathrm{sk}_{\mathrm{serial}_i}$. It just needs to use the master private key and the seeds to view the entire transaction history.

5 System overview

Our payment system is totally anonymous, in that transactions between users are anonymous and transactions between a user and a bank are anonymous. Our payment system adds auditors, who can monitor

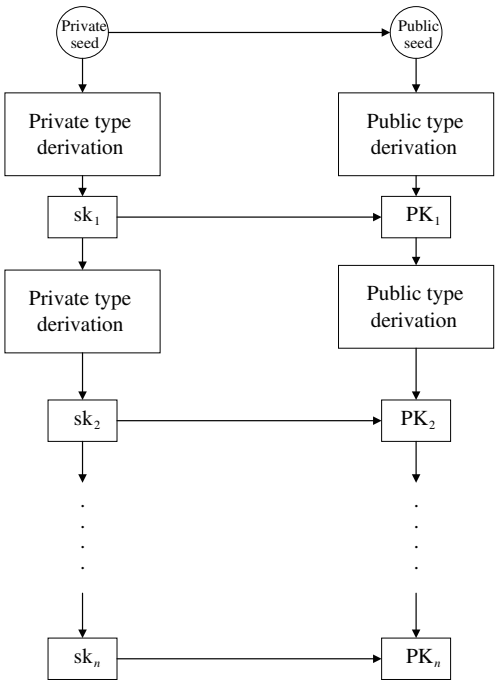

Figure 1 Generate a serial number of a coin.

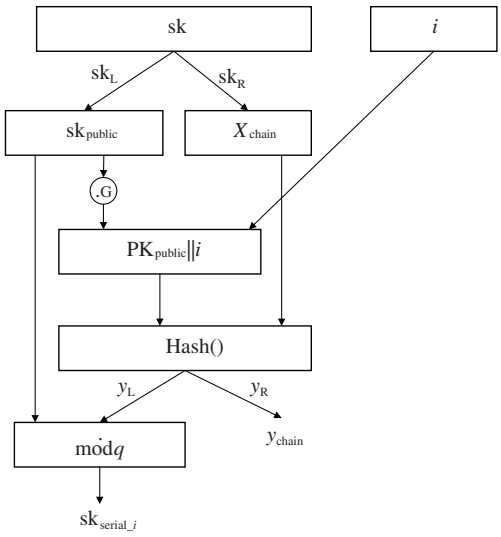

Figure 2 Generate a private key.

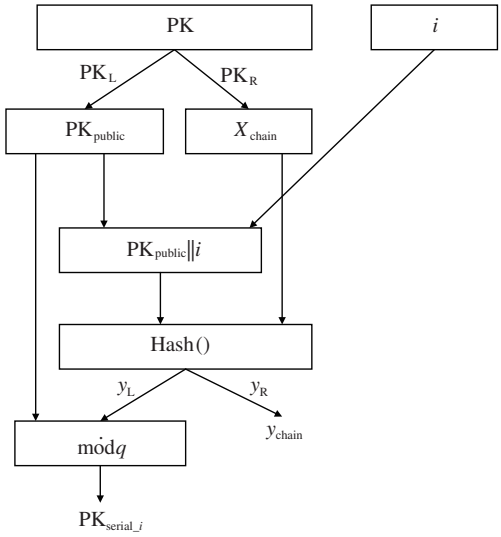

Figure 3 Generate a public key.

the flow of money. Users and banks cannot see this flow so the payment system is totally anonymous for users. Our payment system uses blind signature technology, public key signatures, blockchains, and POW technology.

To ensure anonymity, we stipulate that banks cannot use the serial number of a coin to associate the coin with a user when a user creats and uses a coin. When a coin is added to the blockchain, anyone can view the serial number of the coin, but there is no information in the coin parameters to identify the current owner of the coin or the previous trading participants. When trading with a coin, users who are parties to the transaction need to share information. The transaction history of the coin includes all its owners, which is added to the properties of the coin. However, only auditors can see the coin history. Nobody else can obtain any information from the blockchain. If the parties in the transaction keep it confidential, no one can learn anything about the transaction through the blockchain.

In our payment system, a coin is just like this

$$\begin{aligned}\mathrm{Coin}_1 = \langle & \mathrm{PK}_{\mathrm{serial}_i}||\mathrm{val}_n||\mathrm{PK}_{\mathrm{trade}_n}||\mathrm{count}|| \\ & \mathrm{Sig}_{\mathrm{sk}_{\mathrm{trade}_{n-1}}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{trade}_n}||\mathrm{val}_n)|| \\ & \mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{user}_n}}(r_{\mathrm{trade}_n}))|| \\ & \mathrm{Sig}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{user}_n}}(r_{\mathrm{trade}_n}))))\rangle,\end{aligned}$$

where $\mathrm{PK}_{\mathrm{serial}_i}$ is the serial number of the coin and val_n is the value of the coin. $\mathrm{PK}_{\mathrm{trade}_n}$ is the public key randomly generated during the n-th transaction. $\mathrm{Sig}_{\mathrm{sk}_{\mathrm{trade}_{n-1}}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{trade}_n}||\mathrm{val}_n)$ is a signature, which is used the previous owner's private key that generated randomly to sign the serial number of the coin and the next owner's public key that generated randomly and the value of the coin. $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{user}_n}}(r_{\mathrm{trade}_n}))$ uses the serial number of the coin which is randomly generated by the auditor who has the corresponding private key to encrypt user_n's signature with a random number. $\mathrm{Sig}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{user}_n}}(r_{\mathrm{trade}_n}))))$ is a signature which uses the next owner's private key that is generated randomly to sign $\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{user}_n}}(r_{\mathrm{trade}_n})))$.

5.1 Minting a coin

Figure 4 shows how Alice gets the bank to sign the first coin transaction blindly with its private signature key for a particular coin value. The process is as follows:

(1) Alice sends a request to the auditor that she wants a coin to beminted.

(2) After the auditor receives Alice's mint request, it will use the method introduced in Section 4 to generate private key $\mathrm{sk}_{\mathrm{serial}_i}$ randomly. The corresponding public key $\mathrm{PK}_{\mathrm{serial}_i}$ will be used as the serial number of the coin and to encrypt the transaction information.

(3) The auditor encrypts $\mathrm{PK}_{\mathrm{serial}_i}$ and the signature $\mathrm{Sig}_{\mathrm{sk}_{\mathrm{audit}}}(\mathrm{PK}_{\mathrm{serial}_i})$ with Alice's public key, and then sends it to Alice.

(4) Alice randomly generates $\mathrm{sk}_{\mathrm{trade}_1}$ and the corresponding $\mathrm{PK}_{\mathrm{trade}_1}$. Alice constructs the coin's transaction data $m = \mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{trade}_1}||\mathrm{val}$ and then blinds this value, forming $m' = m \cdot \mathrm{Enc}_{\mathrm{PK}_{\mathrm{val}}}(r_{\mathrm{blind}})$.

(5) Alice sends $m' = m \cdot \mathrm{Enc}_{\mathrm{PK}_{\mathrm{val}}}(r_{\mathrm{blind}})$ to the bank along with the value of the coin she requires.

(6) When the bank receives Alice's mint request, it blindly signs $m' = m \cdot \mathrm{Enc}_{\mathrm{PK}_{\mathrm{val}}}(r_{\mathrm{blind}})$. The bank has numerous signature keys for different coin values (for example \$1, \$5, \$10), and uses the private signature key ($\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}$) corresponding to the val that Alice requires. The bank deducts the corresponding value from Alice's account, checks that the blinded message $m' = m \cdot \mathrm{Enc}_{\mathrm{PK}_{\mathrm{val}}}(r_{\mathrm{blind}})$ adheres to certain parameters (such as message length), and use its private signature key corresponding to val ($\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}$) to sign the coin's blinded transaction $m' = m \cdot \mathrm{Enc}_{\mathrm{PK}_{\mathrm{val}}}(r_{\mathrm{blind}})$. The bank gets s' by blindly signing m', that is $s' = \mathrm{Sig}_{\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}}(m')$.

(7) The bank sends $s' = \mathrm{Sig}_{\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}}(m')$ to Alice.

(8) Alice uses the random number r_{blind} to remove the blinding of the bank's signature s' to get the final signature:

$$s = s' \cdot r_{\mathrm{blind}}^{-1} = \mathrm{Sig}_{\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{trade}_1}||\mathrm{val}),$$

which has been signed with the bank's private signature key ($\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}$) for value (val). Now, Alice has the private key $\mathrm{sk}_{\mathrm{trade}_1}$ of the coin. She is the owner of the coin. She does not need to disclose any information about the serial number of the coin to the bank. Only Alice knows the private key $\mathrm{sk}_{\mathrm{trade}_1}$ of the coin, which she uses to give the coin to another user in the network. Alice randomly generates r_{trade_1}, which is used to record the transaction data, which only auditor can view.

(9) Alice broadcasts

$$\begin{aligned}\mathrm{Coin} = \langle & \mathrm{PK}_{\mathrm{serial}_i}||\mathrm{val}||\mathrm{PK}_{\mathrm{trade}_1}||\mathrm{count}=0|| \\ & \mathrm{Sig}_{\mathrm{sk}_{\mathrm{bank}_{\mathrm{val}}}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{trade}_1}||\mathrm{val})|| \\ & \mathrm{Sig}_{\mathrm{sk}_{\mathrm{audit}}}(\mathrm{PK}_{\mathrm{serial}_i})||\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1}))|| \\ & \mathrm{Sig}_{\mathrm{sk}_{\mathrm{trade}_1}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sig}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1}))))\rangle\end{aligned}$$

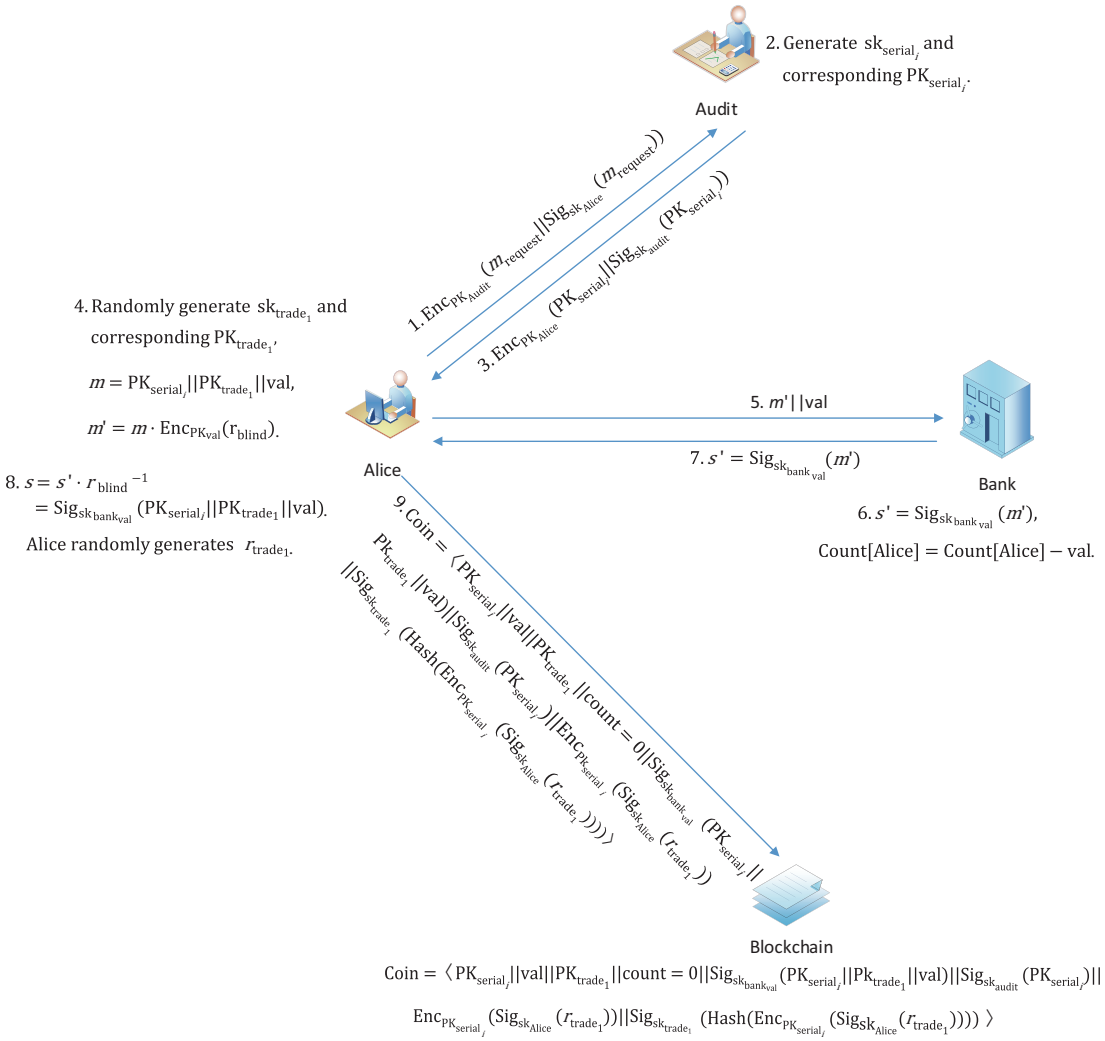

Figure 4 (Color online) The process of exchanging coin from a bank.

to the network. A miner can see if the coin is valid by checking whether PK_{serial_i} is consistent with the signature of the auditor, whether val, PK_{trade_1} and PK_{serial_i} are consistent with the signature of the bank and whether the signature $Sig_{sk_{trade_1}}(Hash(Enc_{PK_{serial_i}}(Sig_{sk_{Alice}}(r_{trade_1}))))$ can be validated by the public key PK_{trade_1}. If valid, the coin will be packed into the blockchain. No one except the auditor can link the coin to Alice.

The communication between Alice and the bank is through a secure channel. Because the bank uses a blind signature, it cannot link the serial number of the coin with Alice. Nobody except the auditor can check the association between the serial number and Alice through the blockchain. Alice must submit the exact coin format, otherwise the miners will not pack it into the blockchain.

5.2 Anonymous coin transfer

Each coin in the blockchain has a public key PK_{trade_n}. The corresponding private key sk_{trade_n} is known only by the coin's owner. If the coin's current owner needs to trade, they use the private key sk_{trade_n} to sign the public key $PK_{trade_{n+1}}$. The corresponding a private key $sk_{trade_{n+1}}$ is randomly generated by the coin's next owner. The new coin is packaged by miners into the blockchain. Once there are six confirmed blocks, the trading can be determined a success. In a transaction, a coin worth val can be split into two coins, a coin of value val_1 and a coin of value val_2, so that any amount can be transferred from one user to another. Each new coin has a different count attribute, which is used to identify it uniquely. The

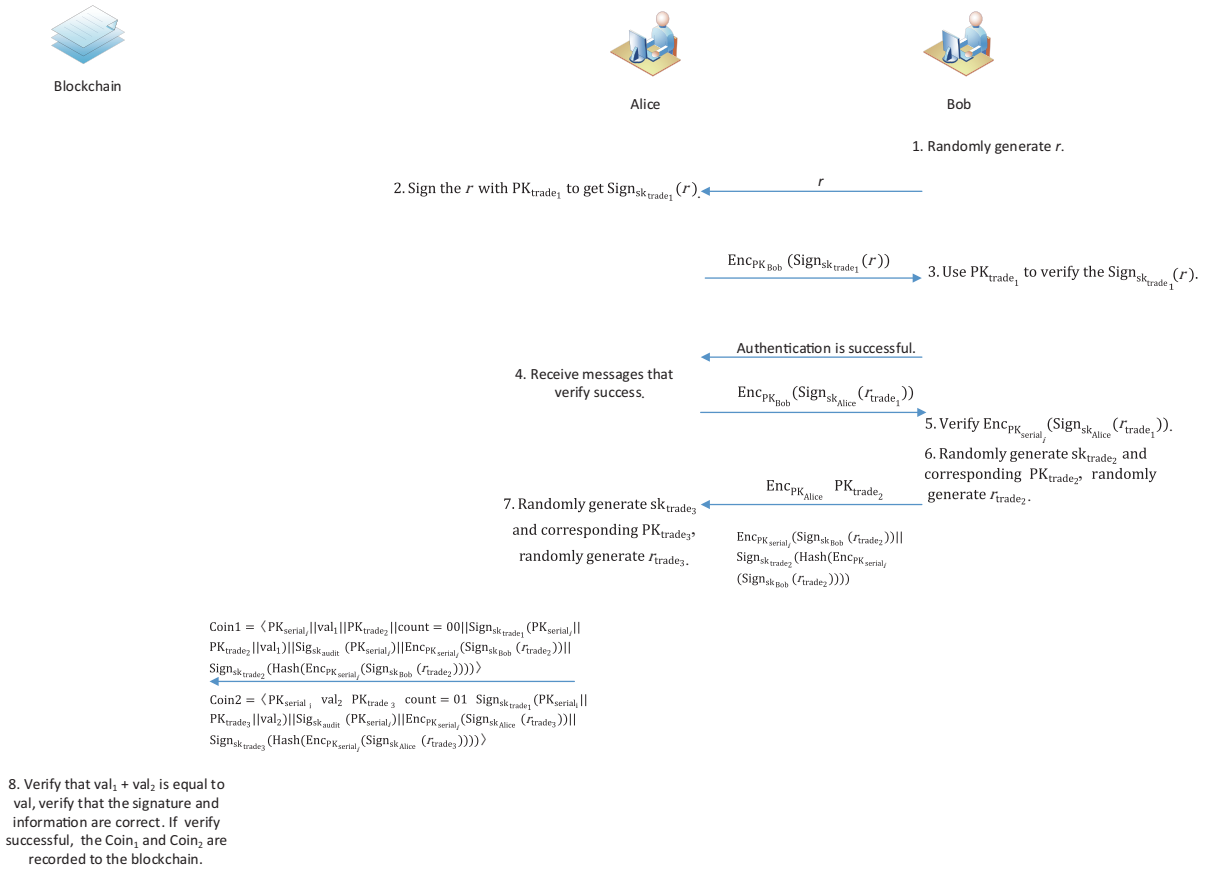

Figure 5 (Color online) Coin transfer protocol.

count attribute of a coin minted from a bank is 0. When a coin is split into two new coins, the count attributes of the new coins are 00 and 01. If these are split, the count attributes of the new coins would be 000, 001, 010, 011.

The first transaction is always signed by the bank whose public key $\mathrm{PK}_{\mathrm{val}}$ for value val is known to all participants. The bank use blind signature to binds the public key that corresponding to randomly generated private key and serial number of the coin, and inserts it into the coin property. When a coin needs to be traded, the coin's current owner uses the private key sign the next owner's random generated private key corresponding public key and inserts into the property of the coin. We then make use of a global ledger and the POW algorithm to lock in valid coins into the blockchain and to prevent the double-spending.

In order for Alice to transfer a coin to another user or spend coins at a merchant premises she needs to fulfill two requirements. The first one is Alice must be able to prove to Bob that she has the private key of the current coin that the coin's public key can be found in the blockchain. The second is Bob submit the correct coin format to the miners, after miners confirmed, the new coin will be packed into the blockchain. The deal was successful only when Bob saw the new coin packed into the blockchain and additional six blocks of confirmation.

Figure 5 shows in detail the coin transfer protocol between Alice and Bob. The process is as follows.

(1) Bob randomly generates r and sends it to Alice. This r will be used to verify that Alice owns the private key of the coin.

(2) After receiving r, Alice uses the private key $\mathrm{sk}_{\mathrm{trade}_1}$ that corresponds to public key $\mathrm{PK}_{\mathrm{trade}_1}$, which can be found in the blockchain, to sign r. She gets the signature $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_1}}(r)$ and sends it to Bob.

(3) Bob use the public key $\mathrm{PK}_{\mathrm{trade}_1}$, which can be found in the blockchain, to verify the signature $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_1}}(r)$. If the validation is successful, Bob can conclude that Alice is the owner of the current coin, since only the coin's current owner knows the coin's private key $\mathrm{sk}_{\mathrm{trade}_1}$. Bob sends a message

stating that authentication has been successful to Alice. If the validation fails, Bob cancels the deal.

(4) When Alice receives the authentication message, she uses Bob's public key $\mathrm{PK}_{\mathrm{Bob}}$ to encrypt the previous messages $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1})$, then she sends $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bob}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1}))$ to Bob.

(5) When Bob receives $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bob}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1}))$, then he decrypts it with his own private key $\mathrm{sk}_{\mathrm{Bob}}$ and gets $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1})$. He uses $\mathrm{PK}_{\mathrm{serial}_i}$, which can be found in the blockchain, to encrypt $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1})$ to verify that the result of the encryption $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_1}))$ is consistent with what is in the blockchain. If the validation is successful, he continues with the subsequent steps. If the validation fails, he cancels the transaction. The validation checks the previous transactions to see whether Alice added her signature information to the coin. This facilitates audits. If Bob's validation failed but he still traded with Alice, he would be punished. This ensures that the transaction history in the blockchain for the auditor is true and reliable.

(6) Bob randomly generates $\mathrm{sk}_{\mathrm{trade}_2}$ and the corresponding $\mathrm{PK}_{\mathrm{trade}_2}$. $\mathrm{sk}_{\mathrm{trade}_2}$ is the private key for the new coin, that will receive for the amount due to him. $\mathrm{PK}_{\mathrm{trade}_2}$ corresponding to $\mathrm{sk}_{\mathrm{trade}_2}$ will be placed in the blockchain and used to verify that he is the owner of the coin. Bob randomly generates r_{trade_2}, and he generates $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Bob}}}(r_{\mathrm{trade}_2}))$ and $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_2}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Bob}}}(r_{\mathrm{trade}_2}))))$. Bob uses Alice's public key $\mathrm{PK}_{\mathrm{Alice}}$ to encrypt r_{trade_2} to get $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Alice}}}(\mathrm{PK}_{\mathrm{trade}_2})$. Then Bob sends $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Alice}}}(\mathrm{PK}_{\mathrm{trade}_2})$, $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Bob}}}(r_{\mathrm{trade}_2}))$ and $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_2}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Bob}}}(r_{\mathrm{trade}_2}))))$ to Alice.

(7) Alice receives $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Alice}}}(\mathrm{PK}_{\mathrm{trade}_2})$ and uses her private key $\mathrm{sk}_{\mathrm{Alice}}$ to decrypt $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Alice}}}(\mathrm{PK}_{\mathrm{trade}_2})$ to get $\mathrm{PK}_{\mathrm{trade}_2}$. Then Alice randomly generates $\mathrm{sk}_{\mathrm{trade}_3}$ and corresponding $\mathrm{PK}_{\mathrm{trade}_3}$. $\mathrm{sk}_{\mathrm{trade}_3}$ is the private key of the next new coin, and the $\mathrm{PK}_{\mathrm{trade}_3}$ corresponding to $\mathrm{sk}_{\mathrm{trade}_3}$ will be placed in the blockchain, where it can be used to verify that she is the owner of the coin. Alice randomly generates r_{trade_3} and she generates $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_3}))$ and $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_3}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_3}))))$. Alice broadcasts

$$
\begin{aligned}
\mathrm{Coin}_1 = {} & \langle \mathrm{PK}_{\mathrm{serial}_i} || \mathrm{val}_1 || \mathrm{PK}_{\mathrm{trade}_2} || \mathrm{count} = 00 || \\
& \mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_1}}(\mathrm{PK}_{\mathrm{serial}_i} || \mathrm{PK}_{\mathrm{trade}_2} || \mathrm{val}_1) || \mathrm{Sig}_{\mathrm{sk}_{\mathrm{audit}}}(\mathrm{PK}_{\mathrm{serial}_i}) || \\
& \mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Bob}}}(r_{\mathrm{trade}_2})) || \\
& \mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_2}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Bob}}}(r_{\mathrm{trade}_2}))))\rangle, \\
\mathrm{Coin}_2 = {} & \langle \mathrm{PK}_{\mathrm{serial}_i} || \mathrm{val}_1 || \mathrm{PK}_{\mathrm{trade}_3} || \mathrm{count} = 01 || \\
& \mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_1}}(\mathrm{PK}_{\mathrm{serial}_i} || \mathrm{PK}_{\mathrm{trade}_3} || \mathrm{val}_2) || \mathrm{Sig}_{\mathrm{sk}_{\mathrm{audit}}}(\mathrm{PK}_{\mathrm{serial}_i}) || \\
& \mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_3})) || \\
& \mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_3}}(\mathrm{Hash}(\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Alice}}}(r_{\mathrm{trade}_3}))))\rangle
\end{aligned}
$$

to the network.

(8) A miner verifies that $\mathrm{val}_1 + \mathrm{val}_2$ is equal to val and verifies that the signature and information are correct. If successful, Coin_1 and Coin_2 will be recorded in the blockchain.

When Bob sees that Coin_1 has been recorded in the blockchain, he can be confirm that he has received a coin for amount val_1 from Alice, because only Bob knows that Coin_1 has private key r_{trade_2}. It shows to others that he is the coin's owner. Now Bob can pay Coin_1 to anyone. If Alice and Bob do not disclose the details of the trade to anyone else, the trade will remain anonymous, except to the auditor. No one except the auditor can relate the coin to Alice and Bob through the blockchain.

5.3 Withdraw coin

If someone wants to deposit a coin in a bank account, they can send an exchange request to the bank and prove to the bank that they are the owner of the coin. After the bank validates the coin's ownership, it will add val to the user's account and publish a message on the blockchain to indicate that the coin has been exchanged. The bank cannot obtain the prior transaction history of the coin from its serial number, and nobody except the auditor can get any transaction information about the coin from the blockchain.

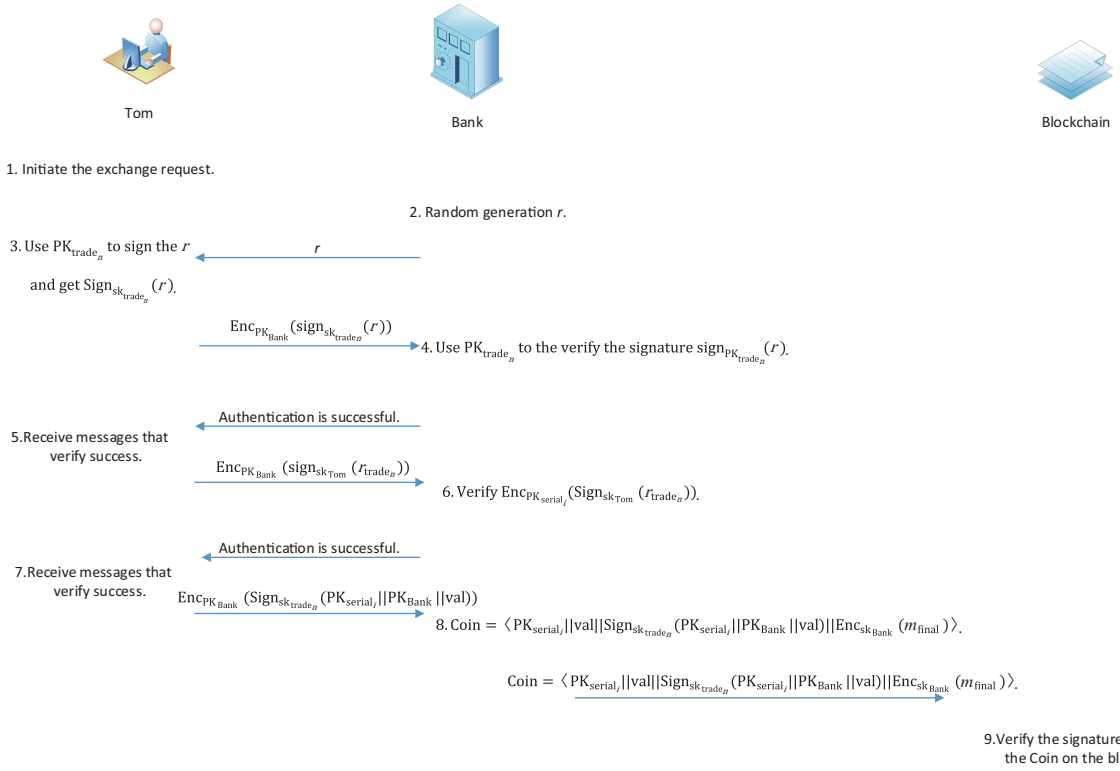

Figure 6 (Color online) Coin exchange protocol.

Figure 6 shows in detail the coin transfer protocol in which Tom give a coin to a bank. The process is as follows.

(1) Tom initiates the exchange request with the bank.

(2) The bank randomly generates r, and sends r to Tom. It asks Tom to sign the r with $\mathrm{PK}_{\mathrm{trade}_n}$.

(3) When Tom receives r, he uses $\mathrm{PK}_{\mathrm{trade}_n}$ to sign the r to get $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(r)$. Tom uses $\mathrm{PK}_{\mathrm{Bank}}$ to encrypt $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(r)$ to get $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(r)$, which he sends to the bank.

(4) When the bank receives $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(r)$, it uses $\mathrm{sk}_{\mathrm{Bank}}$ to decrypt it and use $\mathrm{PK}_{\mathrm{trade}_n}$ to the verify the signature $\mathrm{sign}_{\mathrm{PK}_{\mathrm{trade}_n}}(r)$. If the validation is successful, it continues the subsequent steps. If the validation fails, it cancels the transaction.

(5) When Tom receives a message that the validation was successful, he uses the bank's public key $\mathrm{PK}_{\mathrm{Bank}}$ to encrypt the previous messages $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Tom}}}(r_{\mathrm{trade}_n})$. Then he sends $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Tom}}}(r_{\mathrm{trade}_n}))$ to the bank.

(6) When the bank receives $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Tom}}}(r_{\mathrm{trade}_n}))$, it decrypts it with its own private key $\mathrm{sk}_{\mathrm{Bank}}$ to get $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Tom}}}(r_{\mathrm{trade}_n})$. It uses $\mathrm{PK}_{\mathrm{serial}_i}$, which can be found in the blockchain, to encrypt the $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Tom}}}(r_{\mathrm{trade}_n})$ to verify that the result of the encryption $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{serial}_i}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{Tom}}}(r_{\mathrm{trade}_n}))$ is consistent with the value in the blockchain. If the validation is successful, continues with the subsequent steps. If the validation fails, it cancels the transaction.

(7) When Tom receives a messages that the second validation was successful, he uses the private key $\mathrm{sk}_{\mathrm{trade}_n}$ to sign the message $\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val}$ to get $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val})$. Then he uses the bank's public key $\mathrm{PK}_{\mathrm{Bank}}$ to encrypt it to get $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val}))$. Tom sends the message $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val}))$ to the bank.

(8) When the bank receives $\mathrm{Enc}_{\mathrm{PK}_{\mathrm{Bank}}}(\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val}))$, it decrypts it with its own private key $\mathrm{sk}_{\mathrm{Bank}}$to get $\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val})$. The bank broadcasts

$$\mathrm{Coin} = \langle \mathrm{PK}_{\mathrm{serial}_i}||\mathrm{val}||\mathrm{Sign}_{\mathrm{sk}_{\mathrm{trade}_n}}(\mathrm{PK}_{\mathrm{serial}_i}||\mathrm{PK}_{\mathrm{Bank}}||\mathrm{val})||\mathrm{Enc}_{\mathrm{sk}_{\mathrm{Bank}}}(m_{\mathrm{final}})\rangle$$

to the network.

(9) Miners verify the signature. If the verification is successful, the coin will be recorded in the blockchain.

When the bank sees that the coin has been packed into the blockchain and passed six confirmation blocks, it will add val to Tom's account. When users see the m_{final} information in the blockchain, then the coin has been terminated and can no longer be used. No one except the auditor can get any information about the flow of the coin from the blockchain, so the transaction is anonymous.

5.4 Auditing process

When the audit department needs to check the coin information and the coin flow direction on the blockchain, the audit department can use the previously saved private key $\text{sk}_{\text{serial}_i}$ to decrypt the owner information in the coin (such as $\text{Enc}_{\text{PK}_{\text{serial}_i}}(\text{Sign}_{\text{sk}_{\text{Bob}}}(r_{\text{trade}_2}))$) and get the corresponding owner information (such as $\text{Sign}_{\text{sk}_{\text{Bob}}}(r_{\text{trade}_2})$) to know the current owner of the coin (such as Bob). According to the count information (such as count = 01) in the coin, the previous coin (such as 0) can be found. The owner of the previous coin can be obtained through the same method, which can be traced back to the original ownership of coin and the complete flow of coin.

6 Security and performance analysis

In the following subsections, we discuss some of the security issues associated with our protocol.

6.1 Transferability

Only the owner of a coin knows the coin's current private key $\text{sk}_{\text{trade}_n}$. No one can get it from the blockchain. The owner of a coin can sign a random number r_{trade_n} using the coin's private key. The receiver can verify that the sender is the owner of the coin using the public key $\text{PK}_{\text{trade}_n}$ from the blockchain. In a trade, the owner of a coin, the buyer, can prove that they own the coin to the seller using the coin's private key $\text{sk}_{\text{trade}_n}$. The seller randomly generates private key $\text{sk}_{\text{trade}_{n+1}}$ and the current owner sign the corresponding public key $\text{PK}_{\text{trade}_{n+1}}$. The buyer produces a coin in the correct format with the correct information and broadcast it to the entire network. Then miners pack the coin into the blockchain. After six confirmation blocks, the trade is successful. No one can obtain the coin's private key from the blockchain. No one can trade with the coin without the coin's private key or without modifying the coin on the blockchain. The parties do not need to contact the bank or the auditor to complete the deal.

6.2 Anonymity

A coin's trading history is encrypted using $\text{PK}_{\text{serial}_i}$ and is recorded on the blockchain. Only the auditor has the private key $\text{PK}_{\text{serial}_i}$ corresponding to private key $\text{sk}_{\text{serial}_i}$, so only the auditor can see the coin's trading history. Other users, including the bank, are unable to get the trading history of the coin. Although the auditor can see the trading history of coin, the auditor cannot modify the coin arbitrarily because they do not have the corresponding private key $\text{sk}_{\text{trade}_n}$ for the public key $\text{PK}_{\text{trade}_n}$ of the coin. Thus, the auditor can see the history of the transaction, but cannot change the coin or the transaction history on the blockchain. Ordinary users, including both trading parties, are unable to get any information about the trading history from the blockchain. The parties only know only their own transaction details, and other users know nothing.

6.3 Double-spending

Each coin has a serial number as its unique ID. A coin has a public key $\text{PK}_{\text{trade}_n}$, which is in the blockchain, and the current owner of the coin has corresponding private key $\text{sk}_{\text{trade}_n}$. When a coin is transferred, the previous owner of the coin uses the private key to sign the new public key that corresponds to the private key randomly generated by the next owner of the coin. After the transaction, the previous coin

and the next coin have formed a chain. If someone wants to modify a coin in the blockchain, they have to modify subsequent coins in the chain too. After a coin's private key has been used to sign the next public key and the next coin is packaged in the blockchain, the previous private key is no longer usable and cannot be used to sign another public key. Our blockchain uses a POW consensus algorithm. Once there are six confirmation blocks in the blockchain, it is difficult to modify the block in the blockchain, which prevents double-spending attacks.

6.4 Performance analysis

Our protocol generates 4 encrypted communications, 2 public and private keys, 1 blind signature, 3 signatures, and 1 asymmetric encryption during mining a coin. In the process of anonymous coin transfer, our agreement conducts 4 encrypted communication, 2 private-public keys, 3 signatures and 3 asymmetric encryption. In the withdraw coin process, our protocol conducts 6 encrypted communications, 1 signature and 1 asymmetric encryption. Our protocols have a very small overhead in terms of communication and computation, so in practice they are available and very efficient.

Zk-snark is adopted in the Zcash. The mathematics of Zcash is very complicated, and it requires 1 set of initialization parameters that need to be destroyed after use. At present, the generation and destruction of initialization parameters are controlled by several zero dollar sponsors, so there are some risks. Another big limitation of the Zcash is the efficiency of zk-snark. Although the proof of zk-snark is verified quickly, it still takes a long time to produce proof and requires a large amount of memory. Although Monroe's currency is relatively efficient, there is still a big loophole in privacy. Ref. [23] analyzed the privacy of Monroe's currency and concluded that Monroe's currency could be tracked.

Compared with existing anonymous digital currency protocols, our protocol has a good performance in both anonymity and performance. Not only that, but our agreement can also have a regulatory function on an anonymous basis, something that existing digital currencies do not.

7 Conclusion

In this paper, we propose an anonymous payment electronic currency protocol with the supervision of the audit department. The electronic currency is designed in order to achieve the user can use anonymous electronic currency and convenient auditors to better regulation electronic currency transaction history. The previous electronic currency only do the anonymity, anonymity leads to illegal use of money and non-regulation. So in this paper, the design of electronic money will not only satisfy the electronic money when use the anonymity and joined the auditing department, realize the anonymity of electronic money, but also can be better to monitor the electronic money. For users, miners and banks, electronic currency is anonymous, and only regulators can get the flow and use of electronic money. On the basis of anonymity and regulation, our electronic currency protocol can also prevent double-spending attacks.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2017YFA0303903), Zhejiang Province Key R&D Project (Grant No. 2017C01062), and National Cryptography Development Fund (Grant No. MMJJ20170121).

References

1 Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. Consulted, 2009. https://bitcoin.org/bitcoin.pdf
2 Androulaki E, Karame G O, Roeschlin M, et al. Evaluating user privacy in Bitcoin. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2013. 34–51
3 McEliece R J, Sarwate D V. On sharing secrets and Reed-Solomon codes. Commun ACM, 1981, 24: 583–584
4 Reid F, Harrigan M. An analysis of anonymity in the Bitcoin system. In: Proceedings of the 3rd International Conference on Privacy, Security, Risk and Trust, 2011. 1318–1326
5 Ron D, Shamir A. Quantitative analysis of the full Bitcoin transaction graph. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2013
6 Biryukov A, Pustogarov I. Bitcoin over tor isn't a good idea. In: Proceedings of IEEE Symposium on Security and Privacy, 2015. 122–134

7 The Guardian. Secrets, lies and Snowden's email: why I was forced to shut down lavabit. Commun ACM, 2014. https://www.theguardian.com/commentisfree/2014/may/20/why-did-lavabit-shut-down-snowden-email
8 Henze M, Hiller J, Hohlfeld O, et al. Moving privacy-sensitive services from public clouds to decentralized private clouds. In: Proceedings of IEEE International Conference on Cloud Engineering Workshops, 2016. 130–135
9 Ruffing T, Moreno-Sanchez P, Kate A. CoinShuffle: practical decentralized coin mixing for Bitcoin. In: Proceedings of European Symposium on Research in Computer Security, 2014
10 Meiklejohn S, Pomarole M, Jordan G, et al. A fistful of Bitcoins: characterizing payments among men with no names. In: Proceedings of Conference on Internet Measurement Conference, 2013. 127–140
11 Chainalysis Inc. Chainalysis: blockchain analysis. 2016. https://www.chainalysis.com/
12 Elliptic Enterprises Limited. Elliptic: the global standard for blockchain intelligence. 2016. http://www.elliptic.co/
13 Barber S, Boyen X, Shi E, et al. Bitter to better – how to make bitcoin a better currency. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2012
14 Bissias G, Ozisik A P, Levine B N, et al. Sybil-resistant mixing for Bitcoin. In: Proceedings of the Workshop on Privacy in the Electronic Society, 2014. 149–158
15 Bonneau J, Narayanan A, Miller A, et al. Mixcoin: anonymity for Bitcoin with accountable mixes. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2014
16 Sasson E B, Chiesa A, Garman C, et al. Zerocash: decentralized anonymous payments from Bitcoin. In: Proceedings of IEEE Symposium on Security and Privacy, 2014. 459–474
17 Saxena A, Misra J, Dhar A. Increasing anonymity in Bitcoin. In: Proceedings of Financial Cryptography and Data Security Workshop, 2014. 122–139
18 Valenta L, Rowan B. Blindcoin: blinded, accountable mixes for Bitcoin. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2015
19 Ziegeldorf J H, Grossmann F, Henze M, et al. Coinparty: secure multi-party mixing of Bitcoins. In: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, 2015
20 Miers I, Garman C, Green M, et al. Zerocoin: anonymous distributed e-cash from Bitcoin. In: Proceedings of IEEE Symposium on Security Privacy, 2013. 397–411
21 Maxwell G. Coinjoin: Bitcoin privacy for the real world. 2013. https://bitcointalk.org/index.php?topic=279249.0
22 Noether S, Mackenzie A, Monero Research Lab. Ring confidential transactions. Ledger, 2016, 1: 1–18
23 Kumar A, Fischer C, Tople S, et al. A traceability analysis of Monero's blockchain. 2017. doi: 10.1007/978-3-319-66399-9_9