Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 5, September 2017, pp. 754-762
DOI: 10.23940/ijpe.17.05.p18.754762

A New Aggregate Signature Scheme in Cryptographic Currency

Chao Yuan*, Mixue Xu, Xueming Si

State Key Laboratory of Mathematical Engineering and Advanced Computing, Information Engineering University, Zhengzhou 450001, China

Abstract

With the rise of Bitcoin, cryptographic currencies have attracted more and more attention. Subsequently, other cryptographic currencies were gradually created, such as Zcash, Moreno, Dash and so on. In cryptographic currency, privacy preserving and expansion are two key technical points. In terms of privacy preserving, more effective solutions were proposed in Zcach, Moreno, Dash and other cryptographic currencies systems, in which ring signature, zero knowledge proof and other cryptographic techniques played important roles. But these schemes mainly considered protecting the addresses of both sides of the transaction. In terms of expansion, lightning network and other projects also give solutions. But most of these projects will bring other problems. In this paper, a signature scheme based on the aggregate signature and the elliptic curve algorithm is proposed to hide the transaction value of a single sender and receiver in the transactions which contain multiple inputs and outputs. This signature scheme achieves the purpose of privacy preserving from the transaction value. Further, the correctness proof and security analysis are given in this paper. In addition to that, another signature scheme that combines aggregation signature with bilinear ring signature is proposed. This aggregate ring signature scheme gives another attempt to solve the problem of expansion in the cryptographic currency system only using cryptographic technologies. At the same time, the sender's addresses can be hidden. Similarly, we also confirmed the correctness of this signature scheme.

Keywords: Bitcoin; Cryptographic Currency; Privacy Preserving; Aggregation Signature; Ring Signature; Expansion

(Submitted on March 24, 2017; Revised on June 17, 2017; Accepted on August 20, 2017)

1. Introduction

With Bitcoin proposed by Satoshi Nakamoto [17] in 2008, cryptographic currencies has attracted more and more attention. Inspired by Bitcoin, more and more cryptographic currencies were born. Some of these cryptographic currencies are used to solve potential problems in Bitcoin. In Bitcoin system, privacy preserving is an important problem. To solve this problem, Dash, Monero, Zcash and other cryptographic currencies were created. For example, Dash uses a technique known as CoinJoin which protects from guessing relevancy of transactions from the values. Monero uses the ring signature to protect the transaction addresses. Zcash used zero-knowledge proof to hide transaction information.

Expansion is another problem in Bitcoin and other cryptographic currencies. In Bitcoin system, the size of each block is limited to 1*M*. Now the size of each transaction is mostly more than 250 bytes. Each block can accommodate up to 4000 transactions under the conditions that a new block is created in every 10 minutes. In the Bitcoin system, only 6-7 transactions can be processed per second. This is far less realistic. In some other cryptographic currency systems, the transaction performance has improved, but there is still a great distance from the practical. To solve this problem, some solutions were presented. Lightning networks are more recognized in these solutions. There are three key technologies in Lightning Network, namely RSMC (Recoverable Sequence Maturity Contract), HTLC(Hashed Timelock Contract) and lightning network. The foundation of the Lightning Network is the two-way micro-payment channel between the two parties. The RSMC defines the most basic work of the two-way micropayment channel. RSMC only supports the simplest unconditional payment of funds, HTLC further achieve the conditional payment of funds, therefore the distribution of channel balance becomes more complex. And finally, based on HTLC, the ultimate goal is achieved which is Lightning Network.

* Corresponding author.
E-mail address: yc_xxgcdx@163.com.

Many transactions on cryptographic currencies include multiple inputs and multiple outputs. In Bitcoin, transactions which include multiple inputs and multiple outputs are very common. And these transactions involve signatures on many different messages generated by many different users. The current processing scheme is to provide a signature for each input which greatly increases the size of the transaction. An aggregate signature scheme enables us to achieve precisely this type of compression. Suppose each of n users has a public-private key pair (PK_i, SK_i). User u_i signs input In_i to obtain a signature σ_i. Then there is a public aggregation algorithm that takes as input all of $\sigma_1,\sigma_2,\ldots,\sigma_n$ and outputs a short compressed signature σ. Anyone can aggregate the n signatures. Moreover, the aggregation can be performed incrementally. This will shorten the original n signatures into one which reduces the size of the transaction.

Our contributions. In this work, we made three contributions in view of the aggregate signature in cryptographic currencies.

- We introduce some existing contributions to the privacy preserving in the cryptographic currencies, including CoinJoin in Dash, ring signature in Monero, and zero knowledge proof in Zcash.
- We propose a signature scheme which combine ECC with aggregate signature. This scheme can hide the transaction value, and the size of the signature on transaction is constant regardless of the number of inputs and outputs the transaction contains, which can improve the performance of signature. And we give the correctness proof and security analysis of this scheme.
- We propose another signature scheme that combines bilinear ring signature with aggregate signature. While protecting the sender's addresses in the transaction, the size of the signature on transaction is constant regardless of the number of inputs and outputs the transaction contains, which can improve the performance of signature. And we give the correctness proof of this scheme.

2. Preliminaries

2.1. Privacy Protection in Cryptographic currencies

2.1.1. Feature Selection Results

Dash uses a technique known as CoinJoin. In a nutshell, the CoinJoin mixes multiple transactions of multiple users to a single transaction through some master nodes. In Dash, each user picks an address, and then sends it to the master node to mix with other addresses. Transactions can only be made with amount 0.1, 1, 10 and100 which increases the difficulty for the attackers to guess the relevance of transactions from the amount of transactions. At the same time, the master nodes are required to ensure out-of-order output. As shown is Fig.1, different lines represent different users and every amount is 10 DASH. DASH is the currency unit in this system. By mixing, the user who is represented by the vertical line makes a transaction of 10 DASH to the user who is represented by the line from top left to bottom right, while it is hard for others to find this transaction from the confused transactions.

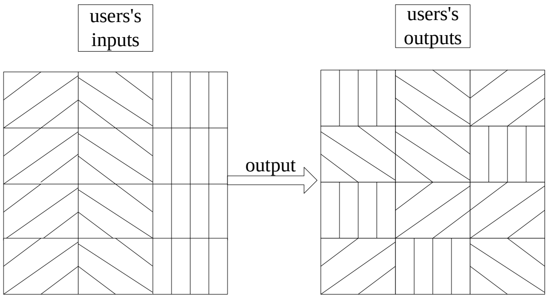

Figure 1. process of CoinJoin

2.1.2. Feature Selection Results

In Dash, there is still the risk that the master nodes are controlled and involved in coins with malicious users, which may lead to the disclosure of user's privacy to a certain extent. In order to solve this problem, Monroe proposed an encryption hybrid scheme that does not depend on the central nodes. There are two technologies in Monroe: *stealth address* and *ring signature* [18].

Stealth address is used to solve the problem of relevance of input addresses and output addresses. Each time the sender makes a transaction, a one-time public key using the elliptic curve via the receiver's address will be computed. The sender then sends out this public key along with an additional message on blockchain. The receivers can detect each transaction based on its own private key to determine whether the sender has already sent out the transaction. When the receiver wants to use the transaction, it can calculate a private key of signature based on their own private key and transaction information. Then the transaction is signed by the private key of signature.

In addition, Monroe proposed a *ring signature* scheme. Whenever the sender wants to make a transaction, the transaction will be signed by the sender's private key and the public keys of other users will be randomly selected. When verifying a signature, the public keys of the other users and the parameters in the signature are needed [4].

2.1.3. Feature Selection Results

A new scheme with zero-knowledge proof was proposed in Zcash, which allows users to hide transaction information only by interacting with the cryptographic algorithm itself so that all transactions are created equally [22].

In Zcash, a non-interactive zero-knowledge proof [3,19] was used, called zk-SNARK. Here, we do not go into the details of zk-SNARK, but generally describe how to use this technology in Zcash. Let's discuss the simplest case, assuming that the amount in Zcash is fixed, such as 1BTC. Then the process of coinage is equivalent to that the user pours 1BTC into an escrow pool, and then write a commitment which can be calculated by the serial number and user's private key to a list. When the user wants to spend the money, two steps need to be done:

- give the serial number
- use zk-SNARK to prove that it holds the user's private key to generate this commitment

2.2. Bilinear Pairings

There, $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative cyclic groups of prime order p, g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$. ψ is a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$. A bilinear pairing is defined as to be $\mathcal{G}=(n,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,g_1,g_2)$where $\mathbb{G}_1=\langle g_1\rangle,\mathbb{G}_2=\langle g_2\rangle$ and $\mathbb{G}_T$ are multiplicative groups of order n. Let e:$\mathbb{G}_1\times\mathbb{G}_2\rightarrow\mathbb{G}_T$ be a map with the following properties [2,18]:

- **Bilinear:** $\forall u\in\mathbb{G}_1, v\in\mathbb{G}_2$ and $a,b\in\mathbb{Z}_n : e\left(u^a,v^b\right)=e(u,v)^{ab}$
- **Non-degenrate:** There exists $u\in\mathbb{G}_1$, $v\in\mathbb{G}_2$ such that $e(u, v)\neq\mathcal{O}$, where $\mathcal{O}$ means the identity of $\mathbb{G}_T$
- **Computability:** There is an efficient algorithm to compute e(u,v) for all $u\in\mathbb{G}_1$, $v\in\mathbb{G}_2$

2.3. Computational Co-Diffie-Hellman

Given $g_2, g_2^a \in\mathbb{G}_2$ and $h\in\mathbb{G}_1$ compute $h^b\in\mathbb{G}_1$[10].

2.4. Decision Co-Diffie-Hellman

Given $g_2,g_2^a\in\mathbb{G}_2$ and $h,h^b\in\mathbb{G}_2$ output yes if $a=b$ and no otherwise. When the answer is yes, we say that $\left(g_2,g_2^a,h,h^b\right)$ is a co-Diffie-Hellman hard [12].

2.5. Ring Signature

Ring signatures were first suggested by Rivest et al, who introduced the RST scheme in 2001[21]. Ring signatures were created in response to the limitations of group signatures. And in particular, they offer honestly participating users with unconditional anonymity, which are formed without a complex setup procedure or the requirement for a group manager. They simply require users to be part of an existing public key infrastructure [20].

Ring signatures are constructed in a way that the ring can only be completed, and therefore verify correctly. The signer has knowledge of some secret information, most commonly a private key corresponding to one of the public keys in the ring. In the signature generation algorithm, a number is generated at random for each of the other public keys in the ring, and then the signer uses the knowledge of their own private key, or some other 'trapdoor information' to close the ring. Ring signatures offer users a type of anonymity by hiding transactions within a set of others' transactions. If there are many users contributing very similar amounts to the ring, the ring is said to have good liquidity, meaning the transactions can occur quickly. Those transactions can be effectively mixed with a high resistance to attempted mixing analysis attacks [6].

2.6. Aggregation Signature

$\boldsymbol{U}$ means a set of users; each user $u \in \boldsymbol{U}$ has a signature key pair (PK_u, SK_u), $\boldsymbol{U}_1 \subseteq \boldsymbol{U}$ means the users whose signatures will be aggregated. Each user $u \in \boldsymbol{U}_1$ generates a signature σ_u for the message M_u they select, and then these signatures are grouped into a single signature by an aggregate community, which can be not in the set $\boldsymbol{U}$, or can be distrusted by the user in the collection $\boldsymbol{U}$, who has access to the user's public key, message, and their home signature, but can't access any private key. The result of the aggregate signature is σ whose length is the same as any single signature. Aggregate signatures have the property that allows a verifier to make sure that each user signs their own messages when σ, the identity of the participants and their corresponding messages are obtained [5].

3. Transaction Value Signature Scheme

Ideally, the transaction contents that include transaction values, transaction addresses and others, are usually hidden through cryptographic signature. The current scheme partially solves this problem. For example, in Monero, the transaction addresses are hidden by the ring signature. In section 3.1, we will design a basic signature scheme to hide the transactions values in cryptographic currencies. In section 3.2, a modified signature scheme which combines the basic signature scheme with aggregate signature will be shown.

3.1. Basic Signature Scheme

Without loss of generality, we deal with a single transaction, which is divided into inputs and outputs, the details shown in Fig.2.

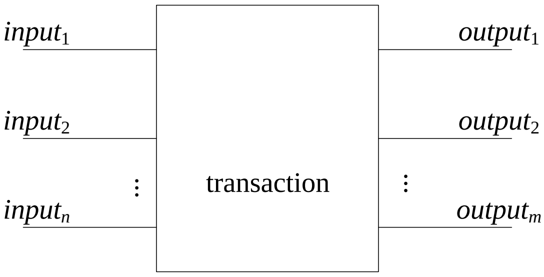

Figure 2. single transaction

As shown in Fig.2, the transaction contains n inputs and m outputs. Accessibly, we have $\sum_{i=1}^{n} input_i = \sum_{j=1}^{m} output_j$. For each i and j, $1 \le i \le n, 1 \le j \le m$,in order to hide $input_i$ and $ouput_j$, this paper uses ECC to make an operation for them. We choose G as the generator of $\mathbb{F}_p$, and the transfer form of $input_i$ and $ouput_j$ are $in_i = input_i \cdot G$ and $out_j = output_j \cdot G$ .And according to the operation rules of the elliptic curve, the following equations are true [14].

$$\sum_{i=1}^{n} \left(input_i \cdot G\right) = \sum_{i=1}^{n} in_i = \left(\sum_{i=1}^{n} input_i\right) \cdot G \tag{1}$$

$$\sum_{j=1}^{m} \left(output_j \cdot G\right) = \sum_{j=1}^{m} out_j = \left(\sum_{j=1}^{m} output_j\right) \cdot G \tag{2}$$

According to equations (1) and (2), we can verify $\sum_{i=1}^{n} input_i = \sum_{j=1}^{m} output_j$ by $\sum_{i=1}^{n} in_i = \sum_{j=1}^{m} out_j$. Because the attackers cannot get $input_i$ and $output_j$ through in_i and out_j, the value of transaction can be hidden by this scheme. The following introduces the homomorphic proof and the drawback of this scheme [12,13].

The security of a signature algorithm must be demonstrated before the algorithm is put into application. At present, taking the development of quantum computing into account, it is necessary to study the homomorphism of the basic signature scheme [6].

- Homomorphic Proof of Basic Signature Scheme

Homomorphic property is an important target to evaluate the security of an algorithm, especially considering that quantum computer gets rapid development. We can easily prove that our basic scheme satisfies additive homomorphism [7,9].

Proof:

For each *i*, $1 \le i \le n$, as defined in basic scheme, $in_i = input_i \cdot G$. According to the operation rules of the elliptic curve, the following equations are true.

$$\left(\sum_{i=1}^{n} input_i\right) \cdot G = \sum_{i=1}^{n} input_i \cdot G \tag{3}$$

$$\sum_{i=1}^{n} input_i \cdot G = \sum_{i=1}^{n} input_i \cdot G \tag{4}$$

We can obtain that

$$\left(\sum_{i=1}^{n} input_i\right) \cdot G = \sum_{i=1}^{n} input_i \cdot G \tag{5}$$

The left side of the equation (5) represents the addition followed by an encryption operation; the right side represents the encryption operation followed by addition. So we can obtain that our basic scheme is additive homomorphisic.

- The Defect of Basic Transaction Signature Scheme

Our basic scheme can hide the amount of transactions, which contain multiple inputs and outputs. But there are also opportunities for the attackers to acquire the amount. On Bitcoin system, there has been mature attack algorithms, such as selfish mining attack [23,8], eclipse attack[11], stubborn mining attack[16],etc. There are similar drawbacks in our basic scheme.

If a malicious attacker impedes *u* inputs and *v* outputs, it satisfies that $\sum_{i=1}^{u} input_i^{'} = \sum_{j=1}^{v} output_j^{'}$. And in the normal network, the sum of all the inputs is

$$Is = \sum_{i=1}^{n} input_i - \sum_{j=1}^{u} input^{'}_{j} \tag{6}$$

The sum of all the outputs is

$$Os = \sum_{i=1}^{m} output_i - \sum_{j=1}^{v} output^{'}_{j} \tag{7}$$

Where the elements of sets $\{input^{'}_{j}\}_{1\le j\le u}$ and $\{output^{'}_{j}\}_{1\le j\le v}$ are contained in sets $\{input_i\}_{1\le i\le n}$ and $\{output_i\}_{1\le i\le m}$.

Because we know that $\sum_{i=1}^{u} input^{'}_{i} = \sum_{j=1}^{v} output^{'}_{j}$ and $\sum_{i=1}^{n} input_i = \sum_{j=1}^{m} output_j$ .It can be obtained that $Is = Os$ .So we can also verify that $Is \cdot G = Os \cdot G$.

In order to modify our basic scheme, this paper combines aggregate signature with the basic scheme to obtain a modified scheme.

3.2. Modified Signature Scheme

Recall that elliptic curve on the finite group $\mathbb{F}_p$ is specified by tuple $< p, a, b, G, n >$, $G = (g_x, g_y)$ is the generator of $\mathbb{F}_p$, $n \cdot G = \mathcal{O}$. The modified scheme is performed as following.

Compute $I_i = input_i \cdot G, i = 1, 2, \cdots, n$, $O_j = output_j \cdot G, j = 1, 2, \cdots, m$. For each *i*, $1 \le i \le n$, randomly select $d_i \in \mathbb{Z}_p$, computes $iR_i = d_i \cdot G$, $ih_i = H(iR_i \parallel input_i)$ and $is_i = d_i \cdot ih_i + input_i$. And randomly select $t_j \in \mathbb{Z}_p$, compute $oR_j = t_j \cdot G$,

$oh_j = H\left(oR_j \,\|\, output_j\right)$ and $os_j = t_j \cdot oh_j + output_j$, the transfer form of inputs and outputs are $\sum_{i=1}^{n} is_i$ and $\sum_{j=1}^{m} os_j$.

- **Feasibility of The Modified Scheme.**

Given $\left(I_i, O_j\right)_{1\le i\le n;1\le j\le m}$, $\left\{iR_i\right\}_{1\le i\le n}$, $\left\{ih_i\right\}_{1\le i\le n}$, $\left\{oR_j\right\}_{1\le j\le m}$, $\left\{oh_j\right\}_{1\le j\le m}$ and the transfer form $\sum_{i=1}^{n} is_i$ and $\sum_{j=1}^{m} os_j$,we can obtain that

$$\sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} is_i \cdot G = \sum_{j=1}^{m} oh_j \cdot oR_j - \sum_{j=1}^{m} os_j \cdot G \tag{8}$$

Proof

$$\begin{aligned} & \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} is_i \cdot G \\ &= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} \left(r_i \cdot ih_i + input_i\right) \cdot G \\ &= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} r_i \cdot ih_i \cdot G + input_i \cdot G \\ &= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} r_i \cdot ih_i \cdot G + input_i \cdot G \\ &= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} ih_i \cdot iR_i + \sum_{i=1}^{n} input_i \cdot G \\ &= \sum_{i=1}^{n} in_i \end{aligned} \tag{9}$$

$$\begin{aligned} & \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} os_i \cdot G \\ &= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} \left(t_i \cdot oh_i + output_i\right) \cdot G \\ &= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} t_i \cdot oh_i \cdot G + output_i \cdot G \\ &= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} t_i \cdot oh_i \cdot G + output_i \cdot G \\ &= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} oh_i \cdot oR_i + \sum_{i=1}^{m} output_i \cdot G \\ &= \sum_{i=1}^{m} out_i \end{aligned} \tag{10}$$

Because we know that $\sum_{i=1}^{n} I_i = \sum_{i=1}^{m} O_i$,it can be obtained that $\sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} is_i \cdot G = \sum_{j=1}^{m} oh_j \cdot oR_j - \sum_{j=1}^{m} os_j \cdot G$

The modified transaction signature scheme greatly avoids the defect in the new signature scheme.If a malicious attacker impedes u' *inputs* and v' *inputs*, and satisfy that $\sum_{i=1}^{u} input_i = \sum_{j=1}^{v} output_j$,then $\sum_{i=1}^{n} ih_i \cdot iR_i$, $\sum_{i=1}^{n} is_i \cdot G$, $\sum_{j=1}^{m} oh_j \cdot oR_j$ and $\sum_{j=1}^{m} os_j \cdot G$ will change as well. And we cannot get

$$\sum_{i=1}^{n-u} ih'_i \cdot iR'_i - \sum_{i=1}^{n-u} is'_i \cdot G = \sum_{j=1}^{m-v} oh'_j \cdot oR'_j - \sum_{j=1}^{m-v} os'_j \cdot G \tag{11}$$

Where $\left\{ih'_i\right\}_{1\le i\le n-u'}$ is the set which is obtained from the set $\left\{ih_i\right\}_{1\le i\le n}$ removing the elements impeded. The relationship also applies to $\left\{iR'_i\right\}_{1\le i\le n-u'}$ and $\left\{iR_i\right\}_{1\le i\le n}$, $\left\{is'_i\right\}_{1\le i\le n-u'}$ and $\left\{is_i\right\}_{1\le i\le n}$, $\left\{oh'_j\right\}_{1\le j\le m-v'}$ and $\left\{oh_j\right\}_{1\le j\le m}$, $\left\{oR'_j\right\}_{1\le j\le m-v'}$ and $\left\{oR_j\right\}_{1\le j\le m}$, $\left\{os'_j\right\}_{1\le j\le m-v'}$ and $\left\{os_j\right\}_{1\le j\le m}$. So it will not pass verification, then the attack will not success.

Our modified transaction value signature scheme can hide the transaction values. Besides, the size of the signature on transaction is constant regardless of the number of inputs and outputs the transaction contains, which can improve the performance of signature. But this scheme only takes into account the transaction values, ignoring transaction addresses, signatures and other components in a complete transaction. Furthermore, we give another signature scheme which combines the aggregate signature and ring signature in section 3.3 and section 3.4. This scheme can process the entire transaction. And the transaction addresses are hidden in this scheme while the size of the signature on transaction is constant regardless of the number of inputs and outputs the transaction contains.

3.3. Bilinear Ring Signatures scheme

We first introduce a ring signature scheme based on bilinear mapping proposed by Dan Boneh etc[8]. The ring signature scheme comprises three algorithms: Key Generation, Ring Signing, and Ring Verification. Let g_1, g_2 are generators of groups $\mathbb{G}_1$, $\mathbb{G}_2$ respectively. And e:$\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear map. A computable isomorphism ψ: $\mathbb{G}_2 \rightarrow \mathbb{G}_1$ exists, with $\psi(g_2) = g_1$. Again we use a full-domain hash function H: $\{0,1\}^* \rightarrow \mathbb{G}_1$. The security analysis views H as a random oracle[1].

Key Generation. For a particular user A, picks random $x \xleftarrow{R} Z_p$, and computes $v \leftarrow g_2^x$. The user's public key is $v \in \mathbb{G}_2$. The user's secret key is $x \in Z_p$.

Ring Signing. Given public keys $v_1,\ldots, v_n \in \mathbb{G}_2$, a transaction $Tx=\{0,1\}^*$ and a private key x corresponding to one of the public keys v_s for some s, choose random $a_i \xleftarrow{R} Z_p$ for all $i \neq s$. Compute $tx=H(Tx) \in \mathbb{G}_1$ and set

$$\sigma_s = \left(tx \big/ \psi\left(\prod\nolimits_{i \neq s} v_i^{a_i} \right) \right)^x \tag{12}$$

For all $i \neq s$, let $\sigma_i = g_1^{a_i}$,output the ring signature $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle \in \mathbb{G}_1^n$.

Ring Verification. Given public keys $v_1,\ldots,v_n \in \mathbb{G}_2$, a transaction $Tx=\{0,1\}^*$, and a ring signture σ, compute $tx=H(Tx)$ and verify that $e(tx, g_2) = \prod\nolimits_{i=1}^n e(\sigma_i, v_i)$.

- Proof that signature verification works

If a signature $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle \in \mathbb{G}_1^n$ on a transaction Tx was indeed generated by A then

$$\begin{aligned}
\prod\nolimits_{i=1}^n e(\sigma_i, v_i) &= e\left(tx \big/ \psi\left(\prod\nolimits_{i \neq s} v_i^{a_i} \right), v_s \right)^{1/x} \cdot \prod\nolimits_{i \neq s} e\left(g_1^{a_i}, v_i \right) \\
&= e\left(tx \big/ \psi\left(\prod\nolimits_{i \neq s} v_i^{a_i} \right), g_2^x \right)^{1/x} \cdot \prod\nolimits_{i \neq s} e\left(g_1^{a_i}, v_i \right) \\
&= e\left(tx \big/ \psi\left(\prod\nolimits_{i \neq s} g_2^{x_i a_i} \right), g_2^x \right)^{1/x} \cdot \prod\nolimits_{i \neq s} e\left(g_1^{a_i}, v_i \right) \\
&= e\left(tx \big/ \prod\nolimits_{i \neq s} g_1^{x_i a_i}, g_2 \right) \cdot \prod\nolimits_{i \neq s} e\left(g_1^{x_i a_i}, g_2 \right) \\
&= e\left(tx \big/ \prod\nolimits_{i \neq s} g_1^{x_i a_i}, g_2 \right) \cdot e\left(\prod\nolimits_{i \neq s} g_1^{x_i a_i}, g_2 \right) \\
&= e(tx, g_2)
\end{aligned} \tag{13}$$

In section 3.4, we combine the aggregate signature with the bilinear ring signature scheme, whose purpose is to reduce the transaction size to one when the transactions include multiple inputs multiple outputs.

3.4. Aggregate ring signature scheme

The bilinear ring signature scheme can protect the address of the sender. But, when a transaction includes multiple input addresses, there will be multiple signatures. In order to reduce the size of signature of the transaction, we try to combine the aggregate signature with the bilinear ring signature. The aggregate ring signature scheme also comprises three algorithms: Aggregate Key Generation, Aggregate Ring Signing, and Aggregate Ring Verification. Recall g_1, g_2 are generators of groups $\mathbb{G}_1$, $\mathbb{G}_2$ respectively. And e: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear map. A computable isomorphism ψ: $\mathbb{G}_2 \rightarrow \mathbb{G}_1$ exists, with $\psi(g_2) = g_1$. Again we use a full-domain hash function H:$\{0,1\}^* \rightarrow \mathbb{G}_1$. The security analysis views H as a random oracle.

Key Generation. For user A_j (j =1,2,…,m), picks random $x_j \xleftarrow{R} Z_p$, and computes $v_j \leftarrow g_2^{x_j}$. The user's public key is $v_j \in \mathbb{G}_2$. The user's secret key is $x_j \in Z_p$.

Signing. Given public keys $v_1^1, v_1^2, \cdots, v_1^n, \cdots, v_m^1, v_m^2, \cdots, v_m^n \in \mathbb{G}_2$, m transactions $Tx_i, Tx_2, \cdots, Tx_m = \{0,1\}^*$ and private keys $\{x_1, x_2, \cdots, x_m\}$ corresponding to m public keys $\{v_1, v_2, \cdots, v_m\}$, choose random $a_j^i \xleftarrow{R} Z_p, j = 1,2,\cdots,m; i = 1,2,\cdots,n$ for all $i \neq s_j$. Compute $tx_i = H(Tx_i) \in \mathbb{G}_1$ and set[2].

$$\sigma_s = \left(tx \middle/ \psi\left(\prod_{i \neq s} v_i^{a_i} \right) \right)^x \tag{14}$$

For all $i \neq s_j$, let $\sigma_j^i = g_1^{a_j^i}$, output the ring signature $\sigma = \left\langle \prod_{j=1}^m \sigma_1^j, ..., \prod_{j=1}^m \sigma_n^j \right\rangle \in \mathbb{G}_1^n$.

Verification. Given public keys $v_1^1, v_1^2, \cdots, v_1^n, \cdots, v_m^1, v_m^2, \cdots, v_m^n \in \mathbb{G}_2$, m transactions $Tx_i, Tx_2, \cdots, Tx_m = \{0,1\}^*$, and a signature σ, compute $tx_j = H(Tx_j) \in \mathbb{G}_1$ and verify that $\prod_{j=1}^m e(tx_j, g_2) = \prod_{i=1}^n e\left(\prod_{j=1}^m \sigma_j^i, \prod_{j=1}^m v_j^i \right)$.

- Proof that signature verification works[15]

If a signature σ on m transactions $Tx_i, Tx_2, \cdots, Tx_m = \{0,1\}^*$ was indeed generated by $A_j, j = 1,2,\ldots,m$ then

$$\begin{aligned}
&\prod_{i=1}^n e\left(\prod_{j=1}^m \sigma_j^i, \prod_{j=1}^m v_j^i \right)^{1/x_j} \\
&= e\left(\prod_{j=1}^m \left(\left(tx_j \middle/ \psi\left(\prod_{i \neq j_s} \left(v_j^i \right)^{a_j^i} \right) \right), \prod_{j=1}^m g_2^{x_j} \right) \right)^{1/x_j} \cdot \prod_{j=1}^m e\left(\prod_{i \neq j_s} g_1^{a_j^i}, \prod_{j=1}^m g_2^{x_j^i} \right) \\
&= e\left(\prod_{j=1}^m \left(\left(tx_j \middle/ \psi\left(\prod_{i \neq j_s} \left(g_2^{x_j^i} \right)^{a_j^i} \right) \right), \prod_{i \neq j_s} g_2 \right) \right) \cdot \prod_{j=1}^m e\left(\prod_{i \neq j_s} g_1^{a_j^i}, \prod_{j=1}^m g_2^{x_j^i} \right) \\
&= e\left(\prod_{j=1}^m \left(\left(tx_j \middle/ \psi\left(\prod_{i \neq j_s} \left(g_2^{x_j^i} \right)^{a_j^i} \right) \right), \prod_{i \neq j_s} g_2 \right) \right) \cdot \prod_{j=1}^m e\left(\prod_{i \neq j_s} g_1^{a_j^i}, \prod_{i \neq j_s} g_2^{x_j^i} \right) \\
&= e\left(\prod_{j=1}^m \left(\left(tx_j \middle/ \psi\left(\prod_{i \neq j_s} \left(g_2^{x_j^i} \right)^{a_j^i} \right) \right), \prod_{i \neq j_s} g_2 \right) \right) \cdot \prod_{j=1}^m e\left(\prod_{i \neq j_s} g_1^{a_j^i x_j^i}, \prod_{i \neq j_s} g_2 \right) \\
&= e\left(\prod_{j=1}^m \left(\left(tx_j \middle/ \psi\left(\prod_{i \neq j_s} \left(g_2^{x_j^i} \right)^{a_j^i} \right) \right), \prod_{i \neq j_s} g_2 \right) \right) \cdot \prod_{j=1}^m e\left(\prod_{i \neq j_s} g_1^{a_j^i x_j^i}, \prod_{i \neq j_s} g_2 \right) \\
&= \prod_{j=1}^m e(tx_j, g_2)
\end{aligned} \tag{15}$$

Our aggregate ring signature scheme can solve the problems of expansion and privacy preserving to a certain extent. But in cryptographic currencies, linkable ring signatures are needed. So we think the combination of aggregate signature and linkable ring signature is important to the cryptographic currencies.

4. **Conclusions**

We introduced the concept of aggregate signatures and constructed two efficient aggregate signature scheme to solve the problem of privacy preserving and expansion in cryptographic currency. Our first signature scheme named transaction value signature scheme combines ECC with aggregate signature. This signature scheme can hide the transaction values and the size of the signature on transaction is constant regardless of the number of inputs and outputs the transaction contains. Persuasively, we gave the correctness proof and safety analysis of the transaction value signature scheme. And our second signature scheme named aggregate ring signature scheme combines ring signature with aggregate signature. This signature scheme can reduce *n* signatures to one for the transactions which contain *n* input addresses and *m* output addresses in cryptographic currency while protecting the addresses of the senders in transactions. The expansion in cryptographic currencies is a controversial problem. Our aggregate ring signature scheme can solve this problem to a certain degree only by cryptographic technology. Previous schemes to solve the expansion mostly used architectural improvements. This will inevitably bring a certain price. Our signature schemes only used aggregate signature and other cryptographic technologies. Previous schemes for the problem of privacy preserving mainly considered the transaction addresses of both sides. Like in Monero, ring signature was used to protect the address of the sender. But, our modified signature transaction value signature scheme can also hide the transaction value. And our aggregate ring signature scheme can reduce the size of the transaction while protecting the addresses of the senders.

Acknowledgements

This work was partly financially supported through grants from the National Key Research and Development Program (No. 2016YFB0800101, No.2016YFB0800100), State Key Laboratory of Mathematics and Advanced Computing Open Topic (No.2015A14) and National Natural Science Foundation of China (No. 61602512).

References

1. M. Aschbacher, "Finite Group Theory, Second Edition," Cambridge University Press ,2000
2. M. Bellare and P. Rogaway, "The Exact Security of Digital Signatures: How to Sign with RSA and Rabin," In Proceedings of Eurocrypt '96, vol.1070, pp:399-416,1996
3. D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," In AsiaCrypt, pp:514–532, 2001
4. D. Boneh,C. Gentry,B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps ," Lecture Notes in Computer Science, vol.2656,no.1,pp:416-432, 2003
5. D. Boneh, C. Gentry, B. Lynn,and H.Shacham. "A Survey of Two Signature Aggregation Techniques," CryptoBytes, vol.6,no.2,2003
6. M. Blum, P. Feldman, and S. Micali, "Non-Interaciiue Zero Knowledge and Its Applications," Proc. 20th ACM Symposium on Theory of Computing, pp.103-112,1988
7. M.van Dijk, C.Gentry, S.Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," International Conference on Theory and Applications of Cryptographic Techniques ,Vol.2009,no.4, pp.24-43, 2010
8. I. Eyal, "The Miner's Dilemma," In IEEE Symposium on Security and Privacy, pp.89-103,2015
9. C.Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," ACM Symposium on Theory of Computing ,vol.9,no.4 pp.169-178,2009
10. R Granger, "On the Static Diffie-Hellman Problem on Elliptic Curves over Extension Fields," ASIACRYPT 2010
11. E.Heilman, A. Kendler, and A. Zohar, "Eclipse Attacks on Bitcoins Peer-to-Peer Network," Usenix Conference on Security Symposium, USENIX Association, vol.45, no.3, pp.129-144,2015
12. A Joux and V Vitse, "Elliptic Curve Discrete Logarithm Problem over Small Degree Extension Fields," Journal of Cryptology, 2010
13. Don Johnson, Alfred Menezes, and Scott Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA) ," International Journal of Information Security,pp.36-63,2001
14. N. Koblitz, A. Menezes, and S. Vanstone," The State of Elliptic Curve Cryptography," Designs, Codes and Cryptography, vol.19, pp.173–193, 2000
15. S. Micali, K. Ohta, and L. Reyzin, "Accountable-subgroup Multisignatures (Extended Abstract)," In Proceedings of CCS 2001, ACM Press, pp.245-254, 2001
16. K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack," IEEE European Symposium on Security and Privacy, vol.142,no.5, 2016
17. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://Bitcoin.org/Bitcoin.pdf, 2009
18. S. Noether, "Ring Signature Confidential Transactions," https://eprint.iacr.org/2015/1098,2015
19. C. Rackoff and D. R. Simon, "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack," Cryptology - CRYPT0 '91, LNCS 576, pp. 433-444, 1992
20. R. L. Rivest, A. Shamir, and L. M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol.21,no.2, pp.120–126, 1978
21. R. L Rivest, A Shamir, and Y Tauman, "How to Leak a Secret: Theory and Applications of Ring Signatures," In Theoretical Computer Science, pp.164-186,2006
22. E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized Anonymous Payments from Bitcoin," IEEE Syposium on Security and Privacy, pp.459-474,2014
23. A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal Selfish Mining Strategies in Bitcoin," International Conference on Financial Cryptography and Data Security, Springer, Berlin, Heidelberg, pp.515-532, 2016

Chao Yuan is a master student from the School of Information Engineering University. His research interests include cryptography, information security and blockchain.

Mixue Xu is a master student from the School of Information Engineering University. Her research interests include cryptography, information security and blockchain.

Xueming Si is a professor in Information Engineering University.His current research interests include computer networks and security, cryptography, information security and blockchain.