Hindawi
Security and Communication Networks
Volume 2017, Article ID 4746586, 10 pages
https://doi.org/10.1155/2017/4746586

WILEY | Hindawi

Research Article

Research on a New Signature Scheme on Blockchain

Chao Yuan, Mi-xue Xu, and Xue-ming Si

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

Correspondence should be addressed to Chao Yuan; yc_xxgcdx@163.com

Received 7 May 2017; Accepted 20 July 2017; Published 21 August 2017

Academic Editor: Xiaojiang Du

With the rise of Bitcoin, blockchain which is the core technology of Bitcoin has received increasing attention. Privacy preserving and performance on blockchain are two research points in academia and business, but there are still some unresolved issues in both respects. An aggregate signature scheme is a digital signature that supports making signatures on many different messages generated by many different users. Using aggregate signature, the size of the signature could be shortened by compressing multiple signatures into a single signature. In this paper, a new signature scheme for transactions on blockchain based on the aggregate signature was proposed. It was worth noting that elliptic curve discrete logarithm problem and bilinear maps played major roles in our signature scheme. And the security properties of our signature scheme were proved. In our signature scheme, the amount will be hidden especially in the transactions which contain multiple inputs and outputs. Additionally, the size of the signature on transaction is constant regardless of the number of inputs and outputs that the transaction contains, which can improve the performance of signature. Finally, we gave an application scenario for our signature scheme which aims to achieve the transactions of big data on blockchain.

1. Introduction

Since the emergence of Bitcoin [1], blockchain as the core technology of Bitcoin has attracted more and more attention. As a combination of a variety of technologies such as distributed data storage, peer-to-peer network, consensus mechanism, and cryptographic algorithm, blockchain has broad prospects of application.

There are still some flaws on blockchain where privacy preserving and performance are two important aspects. When achieving the characteristics of blockchain, preserving the privacy is the focus of academic research. In this field, Monero and Zcach are representative projects where ring signature, zero-knowledge proof, and other cryptographic technologies play important roles. In addition, achieving rapid trading to meet realistic demands is another challenge that blockchain faces. In this field, lightning network is widely recognized, but there are also some flaws in its theories and implement.

Meanwhile, we know big data has been used in many fields. However, there are still many flaws in the storage, transmission, transaction, and privacy preserving of big data. And blockchain was considered to be an ideal technology for solving these flaws. Thus, we applied our new signature scheme to the transactions of big data on blockchain.

Our Contributions. In this work, we make three contributions in view of the privacy preserving and performance on blockchain.

(1) We introduce some existing contributions to the privacy preserving on blockchain, including CoinJoin in Dash, ring signature in Monero, and zero-knowledge proof in Zcash.

(2) We introduce some cryptographic technologies which are favorable for privacy preserving and performance on blockchain, including elliptic curve cryptography (ECC), bilinear maps, and aggregation signature. And then we propose a new signature scheme for the transaction on blockchain in which the amount will be hidden especially in the transactions which include multiple inputs and outputs. Additionally, the size of the signature on transaction is constant regardless of the number of inputs and outputs that

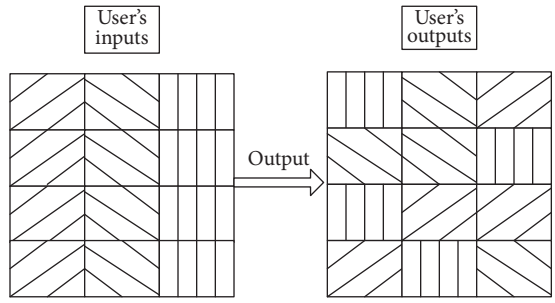

Figure 1: CoinJoin technique.

the transaction contains, which can improve the performance of the signature. And we give the security analysis of our new signature scheme.

(3) We propose an application scenario for our signature scheme which aims to achieve the transaction of big data on blockchain.

Paper Organization. The rest of the paper is organized as follows. Section 2 introduces some projects which aimed at the privacy preserving on blockchain. And the basic building blocks that will be used in our signature scheme are also introduced. In Section 3, the core of our new signature scheme which aimed at hiding the amount of transactions is introduced. The main contribution of this paper is the new signature scheme on blockchain based on aggregate signature that will be described in Section 4, and a formal security analysis for our proposed scheme will also be presented. In Section 5, a simple application of our signature scheme is introduced with respect to transactions of big data. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Privacy Preserving on Blockchain

Dash. 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 of 0.1, 1, 10, and 100 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 in Figure 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.

Monero. In Dash, there is still the risk that the master nodes are controlled by malicious attackers, which may lead to the disclosure privacy of the users. In order to solve this problem, a hybrid cryptographic scheme that does not depend on the central nodes was proposed in Monero. There are two technologies in Monroe: one is called *stealth address* and the other is called *ring signature* [2, 3].

Stealth address is 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. And 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 randomly selected. When verifying a signature, the public keys of the other users and the parameters in the signature are needed.

Zcash. 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 [4].

In Zcash, a noninteractive zero-knowledge proof [5, 6] was used, which is 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 us discuss the simplest case, assuming that the amount in Zcash is fixed, such as 1BTC. Then the process of coinage is equivalent to the fact that the user pours 1BTC into an escrow pool and then writes 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:

(1) Give the serial number.

(2) 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 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 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a map with the following properties [7, 8]:

(i) Bilinear: $\forall u \in \mathbb{G}_1, v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_n : e(u^a, v^b) = e(u, v)^{ab}$.

(ii) Nondegenerate: 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 $\mathbb{G}_T$.

(iii) 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. Aggregate Signature. There, **U** means a set of users, each user $u \in \mathbf{U}$ has a signature key pair $(\mathrm{PK}_u, \mathrm{SK}_u)$, and $\mathbf{U}_1 \subseteq \mathbf{U}$ means the users whose signatures will be aggregated. Each user $u \in \mathbf{U}_1$ 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 cannot be in the set **U** or can be distrusted by the user in the collection **U**, who has access to the user's public key, message, and their home signature but cannot 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 a verifier can make sure that each user signs their own messages [7, 8] when σ and each message are obtained.

2.4. Elliptic Curve. Assume that $\mathbb{F}_q$ has characteristic greater than 3. An elliptic curve E over $\mathbb{F}_q$ is the set of all solutions $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ to an equation $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_q$, and $4a^2 + 27b^2 \neq 0$, together with a special point ∞ called the point at infinity. It is well known that E is an abelian group with the point ∞ serving as its identity element. The rules for group addition are summarized below [9].

(1) Let $P = (x_1, y_1) \in E$; then $-P = (x_1, -y_1)$. If $Q = (x_2, y_2) \in E$, $Q \neq -P$, then $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{if } P = Q. \end{cases} \tag{1}$$

If $\mathbb{F}_q$ is a field of characteristic 2, an elliptic curve E of zero j-invariant over $\mathbb{F}_q$ is the set of all solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ to an equation $y^2 + cy = x^3 + ax + b$, where $a, b, c \in \mathbb{F}_q$, $c \neq 0$, together with the point at infinity ∞. The rules for group addition are summarized below.

(2) Let $P = (x_1, y_1) \in E$; then $-P = (x_1, y_1 + c)$. If $Q = (x_2, y_2) \in E, Q \neq -P$, then $P + Q = (x_3, y_3)$, where

$$x_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)^2 + x_1 + x_2 & \text{if } P \neq Q \\ \dfrac{x_1^4 + a^2}{c^2} & \text{if } P = Q, \end{cases}$$

$$y_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + y_1 + c & \text{if } P \neq Q \\ \left(\dfrac{x_1^2 + a}{c}\right)(x_1 + x_3) + y_1 + c & \text{if } P = Q. \end{cases} \tag{2}$$

If $\mathbb{F}_q$ is a field of characteristic 2, an elliptic curve E of *nonzero j-invariant* over $\mathbb{F}_q$ is the set of all solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ to an equation $y^2 + xy = x^3 + ax^2 + b$, where $a, b \in \mathbb{F}_q, b \neq 0$, together with the point at infinity ∞. The rules for group addition are summarized below.

(3) Let $P = (x_1, y_1) \in E$; then $-P = (x_1, y_1 + x_1)$. If $Q = (x_2, y_2) \in E, Q \neq -P$, then $P + Q = (x_3, y_3)$, where

$$x_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)^2 + \dfrac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a & \text{if } P \neq Q \\ x_1^2 + \dfrac{b}{x_1^2} & \text{if } P = Q, \end{cases} \tag{3}$$

$$y_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1 & \text{if } P \neq Q \\ x_1^2 + \left(x_1 + \dfrac{y_1}{x_1}\right)x_3 + x_3 & \text{if } P = Q. \end{cases}$$

3. Core of the New Signature Scheme

When transactions are generated on blockchain, cryptographic signatures are used to judge the legality of the transactions and the identities of the senders [10]. Furthermore, the signature algorithms are aimed at privacy preserving of the transactions, including the addresses of both sides and transaction amount. For example, in Bitcoin, ECDSA [11, 12], RIPEMD [13, 14], and SHA256 [15, 16] are used to make signatures for the transactions. In Section 3.1, we will design a scheme which is the core of our new signature scheme. The amount of transactions which include multiple inputs and outputs can be hidden using this scheme.

3.1. Basic Scheme. Without loss of generality, we deal with a single transaction, which is divided into inputs and outputs; the details are shown in Figure 2.

As shown in Figure 2, the transaction contains n inputs and m outputs. Accessibly, we have $\sum_{i=1}^{n} \mathrm{in}_i = \sum_{j=1}^{m} \mathrm{out}_j$.

For each i and $j, 1 \leq i \leq n, 1 \leq j \leq m$; in order to hide in_i and out_j, this paper uses ECC to make an operation for them. We choose G as the generator of $\mathbb{F}_p$, and the transfer forms of in_i and out_j are $I_j = \mathrm{in}_j \cdot G$ and $O_j = \mathrm{out}_j \cdot G$. And according to the operation rules of the elliptic curve, the following equations are true [17]:

$$\sum_{i=1}^{n} \mathrm{in}_i \cdot G = \sum_{i=1}^{n} I_i = \left(\sum_{i=1}^{n} \mathrm{in}_i\right) \cdot G$$
$$\sum_{j=1}^{m} \mathrm{out}_j \cdot G = \sum_{j=1}^{m} O_j = \left(\sum_{j=1}^{m} \mathrm{out}_j\right) \cdot G. \tag{4}$$

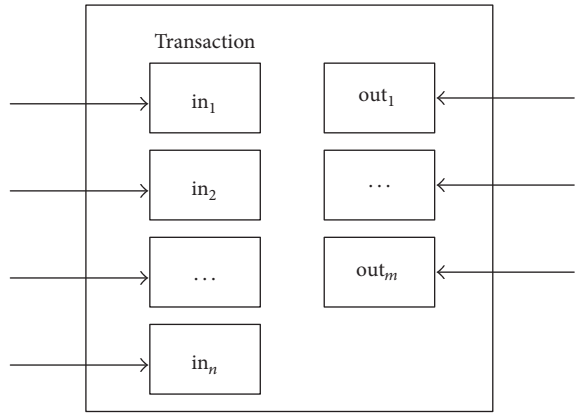

Figure 2: Model of single transaction.

According to (4), we can verify $\sum_{i=1}^{n} \text{in}_i = \sum_{j=1}^{m} \text{out}_j$ by $\sum_{i=1}^{n} I_i = \sum_{j=1}^{m} O_j$. Because the attackers cannot get in_i and out_j through I_i and O_j, the amount of transaction can be hidden by this scheme. The following introduces the homomorphic proof and the drawback of this scheme [18].

Homomorphic Proof of the 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 [19, 20].

Proof. For each $i, 1 \le i \le n$, as defined in basic scheme, $I_i = \text{in}_i \cdot G$. According to the operation rules of the elliptic curve, the following equations are true:

$$\left(\sum_{i=1}^{n} \text{in}_i\right) \cdot G = \sum_{i=1}^{n} \text{in}_i \cdot G$$
$$\sum_{i=1}^{n} \text{in}_i \cdot G = \sum_{i=1}^{n} \text{in}_i \cdot G. \tag{5}$$

We can obtain that

$$\left(\sum_{i=1}^{n} \text{in}_i\right) \cdot G = \sum_{i=1}^{n} \text{in}_i \cdot G. \tag{6}$$

The left side of (6) means the addition followed by an encryption operation; correspondingly the right side means the encryption operation followed by addition. So we can obtain that our basic scheme is additive homomorphic. □

The Drawback of the Basic Scheme. Our basic scheme can hide the amount of the 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 [21, 22], eclipse attack [23], and stubborn mining attack [24]. There are similar drawbacks in our basic scheme.

A malicious attacker impedes u inputs and v outputs, which satisfy the fact that $\sum_{i=1}^{u} \text{in}'_i = \sum_{j=1}^{v} \text{out}'_j$. And in the normal network, the sum of all the inputs is

$$Is = \sum_{i=1}^{n} \text{in}_i - \sum_{j=1}^{u} \text{in}'_j. \tag{7}$$

The sum of all the outputs is

$$Os = \sum_{i=1}^{m} \text{out}_i - \sum_{j=1}^{v} \text{out}'_j, \tag{8}$$

where the elements of sets $\{\text{in}'_j\}_{1\le j\le u}$ and $\{\text{out}'_j\}_{1\le j\le v}$ are contained in sets $\{\text{in}_i\}_{1\le i\le n}$ and $\{\text{out}_i\}_{1\le i\le m}$.

Because we know that $\sum_{i=1}^{u} \text{in}'_i = \sum_{j=1}^{v} \text{out}'_j$ and $\sum_{i=1}^{n} \text{in}_i = \sum_{j=1}^{m} \text{out}_j$, 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 Scheme. Recall that elliptic curve on the finite group $\mathbb{F}_p$ is specified by tuple $\langle p, a, b, G, n\rangle$, $G = (g_x, g_y)$ which is the generator of $\mathbb{F}_p, n \cdot G = \mathcal{O}$. The modified scheme is performed as follows.

(1) Compute $I_i = \text{in}_i \cdot G,\ i = 1, 2, \ldots, n, O_j = \text{out}_j \cdot G,\ j = 1, 2, \ldots, m$.

(2) For each $i, 1 \le i \le n$, randomly select $d_i \in \mathbb{Z}_p$, and compute $iR_i = d_i \cdot G, ih_i = H(iR_i \parallel \text{in}_i)$, and $is_i = d_i \cdot ih_i + \text{in}_i$. And randomly select $t_j \in \mathbb{Z}_p$, and compute $oR_j = t_j \cdot G$, $oh_j = H(oR_j \parallel \text{out}_j)$, and $os_j = t_j \cdot oh_j + \text{out}_j$; the transfer forms of inputs and outputs are $\sum_{i=1}^{n} is_i$ and $\sum_{j=1}^{m} os_j$.

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

$$\sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} is_i \cdot G = \sum_{j=1}^{m} oh_j \cdot oR_j - \sum_{j=1}^{m} os_j \cdot G. \tag{9}$$

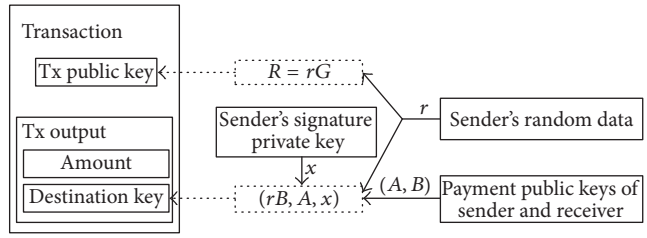

Figure 3: Basic transaction structure.

Proof of the feasibility of the modified scheme will be given in the Appendix.

The modified scheme greatly avoids the drawback in the basic scheme. If a malicious attacker impedes u inputs and v inputs, which satisfy the fact that $\sum_{i=1}^{u} \text{in}_i = \sum_{j=1}^{v} \text{out}_j$, 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

$$\sum_{i=1}^{n-u} ih'_i \cdot iR'_i - \sum_{i=1}^{n-u} is'_i \cdot G = \sum_{j=1}^{m-v} oh'_j \cdot oR'_j - \sum_{j=1}^{m-v} os'_j \cdot G, \tag{10}$$

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

4. New Signature Scheme on Blockchain

In Section 3, we proposed a new scheme which aimed at hiding the amount of the transactions on blockchain which contain multiple inputs and outputs. Based on this, we designed a new signature scheme that can protect the amount of transactions and keep the size of signatures constant regardless of the number of inputs and outputs. Recall that elliptic curve E on the finite group $\mathbb{F}_p$ is specified by tuple $\langle p, a, b, G, n\rangle$. The base groups are $\mathbb{G}_1$ and $\mathbb{G}_2$, their respective generators are g_1 and g_2, the computable isomorphism ψ *is* from $\mathbb{G}_2$ to $\mathbb{G}_1$, and the bilinear map is $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with target group $\mathbb{G}_T$. Let $\mathscr{H}_s : \{0,1\}^* \to \mathbb{F}_q$, $\mathscr{H}_p : E(\mathbb{F}_q) \to E(\mathbb{F}_q)$.

4.1. Basic Signature Scheme

Key Generation. A particular user picks random $x \xleftarrow{R} \mathbb{Z}_p$, $a \in E$ and computes $v = g_2^x$, $A = aG$. The user's signature public key and signature private key are $v \in \mathbb{G}_2$ and $x \in \mathbb{Z}_p$. The user's payment public key and payment private key are $A \in E$ and $a \in E$.

Signing. We suppose that the sender wants to send a payment to a particular receiver whose payment public key is B. The sender generates a random $r \in [1, n-1]$ and computes a one-time public key $P = \mathscr{H}s(rB)G + A$ and then computes $\sigma = P^x$. The signature is $\sigma \in \mathbb{G}_1$. $R = r \cdot G$ is also packed somewhere into the transaction.

Verification. Given the sender's payment public key v, and the signature σ, the receiver computes $P' = \mathscr{H}s(b \cdot R) \cdot G + A$ and then accepts if $e(\sigma, g_2) = e(P', v)$ holds.

We know that $b \cdot R = b \cdot r \cdot G = r \cdot B$; then $P' = P$. And through the rules of the bilinear maps, we obtain that $e(\sigma, g_2) = e(P^x, g_2) = (P, g_2^x) = (P', v)$. Figure 3 gives the structure of our basic signature scheme.

As shown in Figure 3, we give the basic signature scheme [2, 25]. In order to achieve the purpose of improving the performance of the signature scheme, we combine the aggregate signature with our basic signature scheme and propose a modified signature scheme in Section 4.2.

4.2. Modified Signature Scheme

Key Generation. For the aggregate subset of users $\mathbf{U}_1 \subseteq \mathbf{U}$, assign to each user an index i, ranging from 1 to $k = |\mathbf{U}_1|$. Each user $u_i \in \mathbf{U}_1$ picks random $x_i \xleftarrow{R} \mathbb{Z}_p$, $a_i \in E$ and computes $v_i = g_2^{x_i}$, $A_i = a_i \cdot G$. The signature public key and signature private key of u_i are $v_i \in \mathbb{G}_2$ and $x_i \in \mathbb{Z}_p$. The payment public key and payment private key of u_i are $A_i \in E$ and $a_i \in E$.

Signing. For each i, $1 \le i \le k$, we suppose that u_i wants to send a payment to a particular receiver whose payment public key is B_i. And u_i generates a random $r_i \in [1, n-1]$ and computes a one-time public key $P_i = \mathscr{H}s(r_i B_i)G + A_i$ and then computes $\sigma_i = P_i^{x_i}$. The signature is $\sigma_i \in \mathbb{G}_1$. $R_i = r_i \cdot G$ is also packed somewhere into the transaction.

Aggregation. Compute $\sigma \leftarrow \prod_{i=1}^{k} \sigma_i$; the aggregate signature is $\sigma \in \mathbb{G}_1$.

Aggregate Verification. We are given an aggregate signature $\sigma \in \mathbb{G}_1$ for an aggregating subset $\mathbf{U}_1 \subseteq \mathbf{U}$ indexed as before and are given the original $P_i = \mathscr{H}s(r_i \cdot B_i) \cdot G + A_i$ and public keys $v_i \in \mathbb{G}_2$ for all users $u_i \in \mathbf{U}_1$. To verify the aggregate signature σ, compute $P'_i = \mathscr{H}s(b_i \cdot R_i) \cdot G + A_i$ for $1 \le i \le k$ and accept if $e(\sigma, g_2) = \prod_{i=1}^{k} e(P'_i, v_i)$ holds.

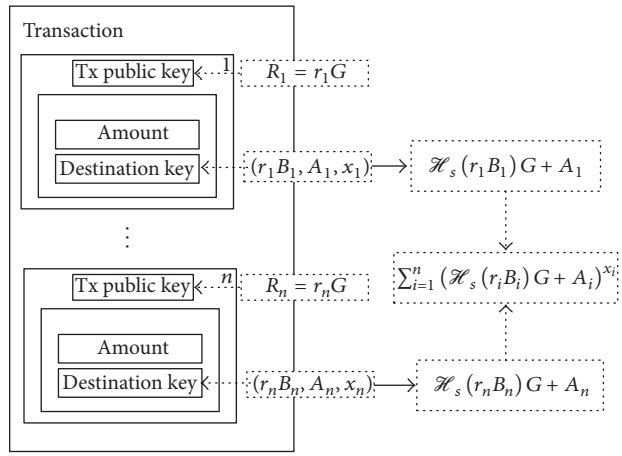

Figure 4: Aggregate transaction structure.

Using the properties of the bilinear map, the left side of the verification equation expands:

$$\begin{aligned} e(\sigma, g_2) &= \prod_{i=1}^{k} e\left(\prod_{i=1}^{k} \sigma_i, g_2\right) = \prod_{i=1}^{k} e\left(\prod_{i=1}^{k} P_i^{x_i}, g_2\right) \\ &= \prod_{i=1}^{k} e\left(P_i^{x_i}, g_2\right) = \prod_{i=1}^{k} e\left(P_i', g_2^{x_i}\right) \\ &= \prod_{i=1}^{k} e\left(P_i', v_i\right). \end{aligned} \tag{11}$$

Figure 4 gives the structure of our aggregate transaction structure.

As shown in Figure 4, the signature is kept constant regardless of the number of inputs and outputs that the transaction contains. Then we combine the core of the new signature scheme proposed in Section 3.2 with the modified signature scheme to a new signature scheme which will be described in Section 4.3.

4.3. New Signature Scheme

Key Generation. For the aggregate subset of users $\mathbf{U}_1 \subseteq \mathbf{U}$, assign to each user an index i, ranging from 1 to $k = |\mathbf{U}_1|$. Each user $u_i \in \mathbf{U}_1$, picks random $x_i \xleftarrow{R} \mathbb{Z}_p$, $a_i \in E$, and computes $v_i = g_2^{x_i}$, $A_i = a_i \cdot G$. The user's signature public key and signature private key are $v_i \in \mathbb{G}_2$ and $x_i \in \mathbb{Z}_p$. The user's payment public key and payment private key are $A_i \in E$ and $a_i \in E$.

Signing. For each i, $1 \le i \le k$, we suppose that u_i wants to send a payment to a particular receiver whose payment public key is B_i. And u_i generates a random $r_i \in [1, n-1]$ and computes a one-time public key $P_i = \mathscr{H}s(r_i B_i) G + A_i$ and then computes $\sigma_i = P_i^{x_i}$. The signature is $\sigma_i \in \mathbb{G}_1$. $R_i = r_i \cdot G$ is also packed somewhere into the transactions. And compute $AI_i = Ia_i \cdot G$, $AO_j = Oa_j \cdot G$.

Aggregation. Compute $\sigma \leftarrow \prod_{i=1}^{k} \sigma_i$; the aggregate signature is $\sigma \in \mathbb{G}_1$. For each i, $1 \le i \le k$, randomly select $d_i \in \mathbb{Z}_p$ and compute $iR_i = d_i \cdot G$, $ih_i = H(iR_i \parallel Ia_i)$, and $is_i = d_i \cdot ih_i + Ia_i$; the transfer form of input is $\sum_{i=1}^{n} is_i$.

Aggregate Verification. We are given an aggregate signature $\sigma \in \mathbb{G}_1$ for an aggregating subset $\mathbf{U}_1 \subseteq \mathbf{U}$ indexed as before and are given the original $P_i = \mathscr{H}s(r_i \cdot B_i) \cdot G + A_i$ and public keys $v_i \in \mathbb{G}_2$ for all users $u_i \in \mathbf{U}_1$. To verify the aggregate signature σ, compute $P_i' = \mathscr{H}s(b_i \cdot R_i) \cdot G + A_i$ for $1 \le i \le k$ and accept if $e(\sigma, g_2) = \prod_{i=1}^{k} e(P_i', v_i)$ holds. And randomly select $t_j \in \mathbb{Z}_p$, compute $oR_j = t_j \cdot G$, $oh_j = H(oR_j \parallel Oa_j)$, and $os_j = t_j \cdot oh_j + Oa_j$; the transfer form outputs are $\sum_{j=1}^{m} os_j$. Figure 5 gives the structure of our new transaction structure.

4.4. Security of the New Signature Scheme. It is easy to show that the security of our new signature scheme is equivalent to the traditional bilinear aggregate signature. As the aggregate chose-key security model which was proposed in [7], the security of aggregate signature schemes is equivalent to the nonexistence of an adversary capable of existentially forging an aggregate signature. Existential forgery here means that the adversary attempts to forge an aggregate signature on a subtransaction of his choice by other subtransactions in a particular transaction. The adversary $\mathscr{A}$ is given a single public key. His goal is the existential forgery of an aggregate signature. We give the adversary power to choose all public keys except the challenge public key. The adversary is also given access to a signing oracle on the challenge key. His advantage $\mathrm{AdvAggSig}_{\mathscr{A}}$ is defined to be his probability of success in the following game [7, 26].

Setup. The aggregate forger $\mathscr{A}$ is provided with a public key PK_1, generated at random.

Queries. Proceeding adaptively, $\mathscr{A}$ requests signatures with PK_1 on the subtransaction of his choice.

Response. Finally, $\mathscr{A}$ outputs $k-1$ additional public keys PK_2, $\ldots$, PK_k. These keys, along with the initial key PK_1, will be

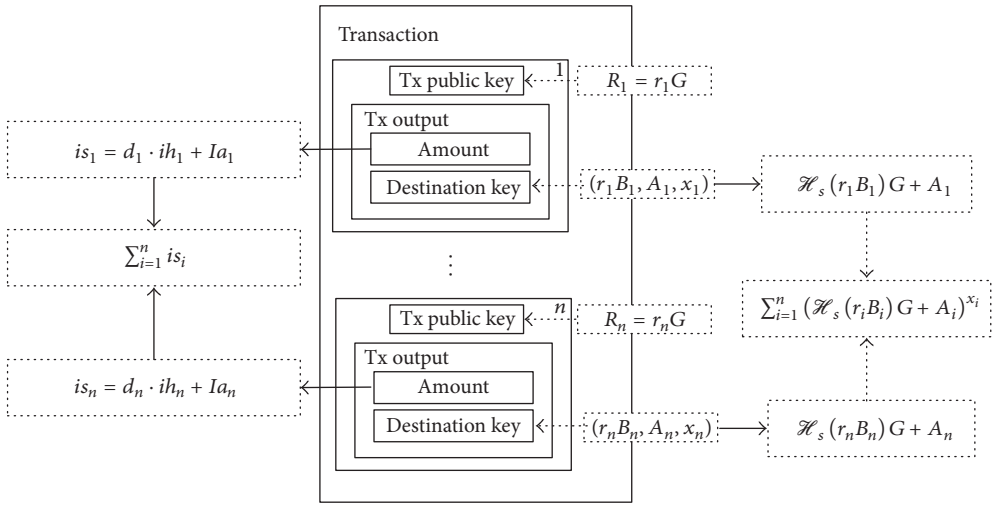

Figure 5: New transaction structure.

included in $\mathscr{A}$'s forged aggregate. $\mathscr{A}$ also outputs subtransaction $T_1, \ldots, T_k$, finally, an aggregate signature σ by the k users, each on his corresponding subtransaction.

The forger wins if the aggregate signature σ is a valid aggregate on subtransactions $T_1, \ldots, T_k$ under keys $\mathrm{PK}_1, \ldots, \mathrm{PK}_k$, and σ is nontrivial.

Definition 1. An aggregate forger $\mathscr{A}(t, q_H, q_s, N, \epsilon)$-breaks an N-user aggregate signature scheme in the aggregate chosen-key model if the following conditions are met:

(1) $\mathscr{A}$ runs in time at most t.

(2) $\mathscr{A}$ makes at most q_H queries to the hash function and at most q_S queries to the signing oracle.

(3) $\mathrm{AdvAggSig}_{\mathscr{A}}$ is at least ϵ.

(4) Forged aggregate signature is by at most N users.

An aggregate signature scheme is $(t, q_H, q_s, N, \epsilon)$-secure. It is against existential forgery in the aggregate chosen-key model if no forger $(t, q_H, q_s, N, \epsilon)$-breaks it. The next theorem shows that this simple constraint is sufficient for proving security in the chosen-key model.

Theorem 2. *Let $(\mathbb{G}_1, \mathbb{G}_2)$ be a (t', ϵ')-bilinear group pair for co-Diffie-Hellman, with each group of order p, with respective generators g_1 and g_2, with an isomorphism computable from $\mathbb{G}_2$ to $\mathbb{G}_1$, and with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Then the bilinear aggregate signature scheme on $(\mathbb{G}_1, \mathbb{G}_2)$ is $(t, q_H, q_s, N, \epsilon)$-secure against existential forgery in the aggregate chosen-key model for all t and ϵ satisfying $\epsilon \geq e(q_s + N) \cdot \epsilon'$ and $t \leq t' - c_{\mathbb{G}_1}(q_H + 2q_s + N + 4) - (N - 1)$, where e is the base of natural logarithms, and exponentiation and inversion on $\mathbb{G}_1$ take time $c_{\mathbb{G}_1}$.*

Besides, the security of the scheme which is used to hide the amount of the transactions has been analyzed in Section 3.2. So, we can get that our signature scheme satisfies unforgeability and other security properties.

5. Application of Signatures Scheme

Big data brings many benefits to our lives. At the same time, there are some drawbacks in big data. Firstly, the utilization of data is poor. Large amounts of data are in the idle state, occupying a lot of storage space. Secondly, there are a lot of drawbacks in the security and privacy of the data. The use of big data exposes personal privacy and other security problems, while big data may be used to do illegal activities by criminals. At the same time, there are some drawbacks in the transmission efficiency and transmission accuracy of data. Blockchain is considered to be an ideal solution to these problems. Based on this, we try to apply our signature scheme to the transactions of big data [27].

5.1. Infrastructure of Transaction of Big Data on Blockchain. Here, we consider the transactions of big data on blockchain. The infrastructure is based on the P2P network which is the network model of blockchain [28]. And we give the model of the infrastructure in Figure 6.

We consider the inputs and outputs of a particular transaction, which consists of data inputs, data outputs, and the corresponding amount of outputs and amount of inputs which are described in Figure 7.

Setup. Recall that elliptic curve on the finite group $\mathbb{F}_p$ is specified by tuple $\langle p, a, b, G, n \rangle$.

Key Generation. For the aggregate subset of users $\mathbf{U}_1 \subseteq \mathbf{U}$, assign to each user an index i, ranging from 1 to $n = |\mathbf{U}_1|$. Each user $u_i \in \mathbf{U}_1$ picks random $x_i \xleftarrow{R} \mathbb{Z}_p$, $a_i \in E$ and computes $v_i = g_2^{x_i}$, $A_i = a_i \cdot G$. The signature public key and signature private key of u_i are $v_i \in \mathbb{G}_2$ and $x_i \in \mathbb{Z}_p$. The payment public key and payment private key of u_i are $A_i \in E$ and $a_i \in E$.

Signing. For each i, $1 \leq i \leq k$, we suppose that u_i wants to send a payment to a particular receiver whose payment public key

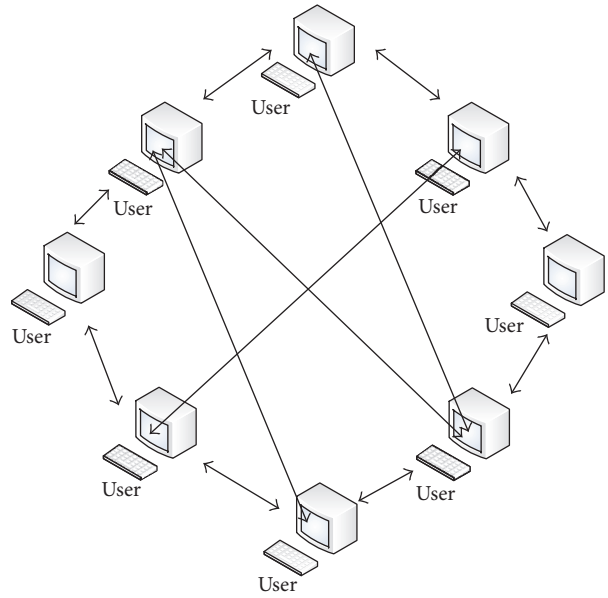

Figure 6: Infrastructure of transaction of data.

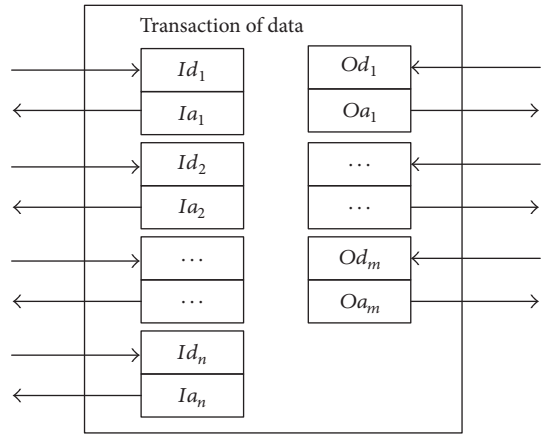

Figure 7: Single transaction of data.

is B_i. And u_i generates a random $r_i \in [1, n-1]$ and computes a one-time public key $P_i = \mathcal{H}s(r_i B_i \parallel Od_i)G + A_i$ and then computes $\sigma_i = P_i^{x_i}$. The signature is $\sigma_i \in \mathbb{G}_1$. $R_i = r_i \cdot G$ is also packed somewhere into the transactions. And compute $AI_i = Ia_i \cdot G$, $AO_j = Oa_j \cdot G$.

Aggregation. Compute $\sigma \leftarrow \prod_{i=1}^{n} \sigma_i$; the aggregate signature is $\sigma \in \mathbb{G}_1$. For each i, $1 \le i \le n$, randomly select $d_i \in \mathbb{Z}_p$ and compute $iR_i = d_i \cdot G$, $ih_i = H(iR_i \parallel Ia_i)$, and $is_i = d_i \cdot ih_i + Ia_i$; the transfer form of input is $\sum_{i=1}^{n} is_i$.

Aggregate Verification. We are given an aggregate signature $\sigma \in \mathbb{G}_1$ for an aggregating subset $\mathbf{U}_1 \subseteq \mathbf{U}$ indexed as before and are given the original $P_i = \mathcal{H}s(r_i \cdot B_i) \cdot G + A_i$ and public keys $v_i \in \mathbb{G}_2$ for all users $u_i \in \mathbf{U}_1$. To verify the aggregate signature σ, compute $P_i' = \mathcal{H}s(b_i R_i \parallel Od_i)G + A_i$ for $1 \le i \le n$ and accept if $e(\sigma, g_2) = \prod_{i=1}^{n} e(P_i', v_i)$ holds. And randomly select $t_j \in \mathbb{Z}_p$, compute $oR_j = t_j \cdot G$, $oh_j = H(oR_j \parallel Oa_j)$, and $os_j = t_j \cdot oh_j + Oa_j$; the transfer form outputs are $\sum_{j=1}^{m} os_j$.

5.2. Performance of Signature Scheme on Transaction of Big Data

Aggregate Signing Time. In a single signature, one hash operation, one modular power multiplication, and one multiplication operation are implemented. Let σ be an aggregate of the n signatures $\sigma_1, \ldots, \sigma_n$. The time to verify the aggregate signature σ is linear in n. And one multiplication with aggregation is implemented [29].

Aggregate Verification Time. In a single verification, k times hash operations and $n + 1$ bilinear maps operations are implemented. Let σ be an aggregate of the n signatures

$\sigma_1, \ldots, \sigma_n$. The time to verify the aggregate signature σ is linear in n.

Signature Space. Let σ be an aggregate of the n signatures $\sigma_1, \ldots, \sigma_n$. The space of the signature will be $1/n$ of the normal signature.

6. Concluding

In this paper, we have proposed a new signature scheme for the transactions on blockchain based on aggregate signature and ECC. Through our new signature scheme, the amount will be hidden when the transactions contain multiple inputs and outputs [30]. Besides, the size of the signature for the transactions will keep constant regardless of the number of inputs and outputs that the transaction contains. We have shown the validity of our new signature scheme. More importantly, the security of our new signature scheme is analyzed. Currently there is no scheme which achieves both hiding the amount of the transactions and constant-size signature when the transaction contains multiple inputs and outputs. Furthermore, we have given an application scenario for our signature scheme which aimed at achieving the transaction of big data on blockchain. And the performance of the signature scheme in the application scenarios was analyzed.

There are still many interesting problems to be solved. For example, it would be valuable to explore the possibility of achieving a signature scheme which combines our scheme with ring signature. Using our scheme to construct a practical complete application is also another interesting problem [31, 32].

Appendix

Proof of the Feasibility of the Modified Scheme

$$\begin{aligned}
&\sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} is_i \cdot G \\
&= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} \left(r_i \cdot ih_i + \mathrm{in}_i\right) \cdot G \\
&= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} r_i \cdot ih_i \cdot G + \mathrm{in}_i \cdot G \\
&= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} r_i \cdot ih_i \cdot G + \mathrm{in}_i \cdot G \\
&= \sum_{i=1}^{n} ih_i \cdot iR_i - \sum_{i=1}^{n} ih_i \cdot iR_i + \sum_{i=1}^{n} \mathrm{in}_i \cdot G = \sum_{i=1}^{n} \mathrm{in}_i.
\end{aligned}$$

$$\begin{aligned}
&\sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} os_i \cdot G \\
&= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} \left(t_i \cdot oh_i + \mathrm{out}_i\right) \cdot G \\
&= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} t_i \cdot oh_i \cdot G + \mathrm{out}_i \cdot G \\
&= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} t_i \cdot oh_i \cdot G + \mathrm{out}_i \cdot G \\
&= \sum_{i=1}^{m} oh_i \cdot oR_i - \sum_{i=1}^{m} oh_i \cdot oR_i + \sum_{i=1}^{m} \mathrm{out}_i \cdot G = \sum_{i=1}^{m} \mathrm{out}_i
\end{aligned} \tag{A.1}$$

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper is supported by National Key Research and Development Program (nos. 2016YFB0800101 and 2016YFB0800100), State Key Laboratory of Mathematics and Advanced Computing Open Topic (no. 2015A14), and National Natural Science Foundation of China (no. 61602512).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2009, https://bitcoin.org/bitcoin.pdf.

[2] N. Saberhagen, "Crypto Note v 2.0," Cryp to Note, 2013.

[3] S. Noether, "Ring signature confidential transactions," 2015, https://eprint.iacr.org/2015/1098.

[4] E. Ben-Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the 35th IEEE Symposium on Security and Privacy, (SP '14)*, pp. 459–474, May 2014.

[5] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," *LNCS 576*, pp. 433–444, 1992.

[6] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC 1988*, pp. 103–112, May 1988.

[7] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Lecture Notes in Computer Science*, vol. 2656 of *Lecture Notes in Comput. Sci.*, pp. 416–432, Springer, 2003.

[8] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "A survey of two signature aggregation techniques," *CryptoBytes*, vol. 6, no. 2, 2003.

[9] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography. An International Journal*, vol. 19, no. 2-3, pp. 173–193, 2000.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[11] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[12] ANSI X9.62, "The elliptic curve digital signature algorithm," Public Key Cryptography for the Financial Services Industry, 1999.

[13] H. Dobbertin, A. Bosselaers, and B. Preneel, "RIPEMD-160: A strengthened version of RIPEMD," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1039, pp. 71–82, 1996.

[14] H. Dobbertin, "RIPEMD with two-round compress function is not collision-free," *Journal of Cryptology*, vol. 10, no. 1, pp. 51–69, 1997.

[15] H. Shariffar, "SHA1 and SHA256 custom instruction design and characterization on Nios II processor," *Journal of the American Oil Chemists Society*, vol. 81, no. 10, pp. 979–987, 2012.

[16] M. Juliato and C. Gebotys, "Tailoring a reconfigurable platform to SHA-256 and HMAC through custom instructions and peripherals," in *Proceedings of the 2009 International Conference on ReConFigurable Computing and FPGAs, ReConFig'09*, pp. 195–200, December 2009.

[17] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[18] A. Joux and V. Vitse, "Elliptic curve discrete logarithm problem over small degree extension fields," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 26, no. 1, pp. 119–143, 2013.

[19] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proceedings of the International Conference on Theory and Applications of Cryptographic Techniques*, vol. 2009, pp. 24–43, Springer, Berlin, Germany, 2010.

[20] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, (STOC '09)*, pp. 169–178, June 2009.

[21] I. Eyal, "The miner's dilemma," in *Proceedings of the 36th IEEE Symposium on Security and Privacy, SP 2015*, pp. 89–103, May 2015.

[22] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*, vol. 9603 of *Lecture Notes in Computer Science*, pp. 515–532, Springer, Berlin, Germany, 2017.

[23] 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.

[24] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: generalizing selfish mining and combining with an eclipse attack," in *Proceedings of the 1st IEEE European Symposium on Security and Privacy*, pp. 305–320, March 2016.

[25] B. Adida, S. Hohenberger, and R. L. Rivest, "Ad-hoc-group signatures from hi-jacked keypairs," In Domacs workshop on Theft in E-Commerce, 2005.

[26] S. Micali, K. Ohta, and L. Reyzin, "Accountable-subgroup multisignatures," in *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, pp. 245–254, Philadelphia, Pa, USA, November 2001.

[27] A. Singh, G. Rumantir, A. South, and B. Bethwaite, "Clustering experiments on big transaction data for market segmentation," in *Proceedings of the 3rd ASE International Conference on Big Data Science and Computing, BIGDATASCIENCE 2014*, August 2014.

[28] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 593–610, 2000.

[29] X. Du, M. Shayman, and M. Rozenblit, "Implementation and performance analysis of SNMP on a TLS/TCP base," in *Proceedings of the 7th IEEE/IFIP International Symposium on Integrated Network Management, IM 2001*, pp. 453–466, Seattle, WA, USA, May 2001.

[30] Y. Xiao, H.-H. Chen, X. Du, and M. Guizani, "Stream-based cipher feedback mode in wireless error channel," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 622–626, 2009.

[31] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[32] X. Yao, X. Han, X. Du, and X. Zhou, "A lightweight multicast authentication mechanism for small scale IoT applications," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3693–3701, 2013.