An Efficient NIZK Scheme for Privacy-Preserving Transactions over Account-Model Blockchain

Shunli Ma[1,2,4], Yi Deng[1,2,4], Debiao He[3], Jiang Zhang[4], and Xiang Xie[5]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China
[4] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, China
[5] Juzix Technology Co. Ltd., Shenzhen, China

Abstract. We introduce the abstract framework of decentralized smart contracts system with balance and transaction amount hiding property under the ACCOUNT architecture. To build a concrete system with such properties, we utilize a homomorphic public key encryption scheme and construct a *highly efficient* non-interactive zero knowledge (NIZK) argument based upon the encryption scheme to ensure the validity of the transactions. Our NIZK scheme is *perfect* zero knowledge in the common reference string model, while its soundness holds in the random oracle model. Compared to previous similar constructions, our proposed NIZK argument dramatically improves the time efficiency in generating a proof, at the cost of relatively longer proof size.

1 Introduction

Bitcoin [Nak08], as the first widely successful decentralized digital currency, has drawn a lot of attention to the conception of blockchain. A blockchain is a tamper-proof digital ledger of transactions with chronological order maintained by distributed consensus nodes (called miners). The miners reach consensus not only on the transactions (e.g., money transfer records or other data) but also on the involving computations (e.g., validate or update the transactions). This guarantees the blockchain to possess decentralization, verifiability and immutability. Due to these properties, blockchain has been used in the design of systems for data storage [KMH+17], provenance [LST+17,XSA+17], sharing economy [XSC+17], dynamic key management [LCC+17], supply chain finance and so forth.

Although the blockchain can provide a powerful abstraction for the design of distributed protocols, the security and privacy issues (e.g., the leakage of user real identity, transaction amount and balance) should not be ignored from the protection of users' interests. Among these security and privacy concerns, hiding the transaction amount and balance is especially important when designing

a blochchain-based system involving economic dealings (e.g., sharing economy or supply chain finance system). Here, we take the blockchain-driven supply chain finance (BDSCF) system [OHHH17] as an example to specify the potential threats without a protection mechanism for money transfer records.

The BDSCF system was proposed to cut unnecessary costs during the deal appears between a supplier and a buyer who trust different supply chain finances (SCFs). Due to the integration of blockchain into supply chain finance system, SCFs (as the distributed miners) collectively maintain a general ledger (see Figure 1) which avoids complicated data synchronism across the participating SCFs and eliminates the inefficiencies in financial flaws. Consequently, it helps the company financing make a higher profits and lower cost. Although BDSCF can enhance the efficiency of trading processes among supply chain partners and improve the buyer-supplier relation during the payment process, the disclosure of the transferred and balance in general ledger to SCFs which may leak key trade secrets of the suppliers. That is, the price of products from different suppliers involved in the general ledger can be estimated by analysing transaction records and balance in account. As a result, the suppliers' incentives to adopt this blockchain-based mechanism will be diminished for their dinterests are compromised, which seriously limits the application and scalability of BDSCF.

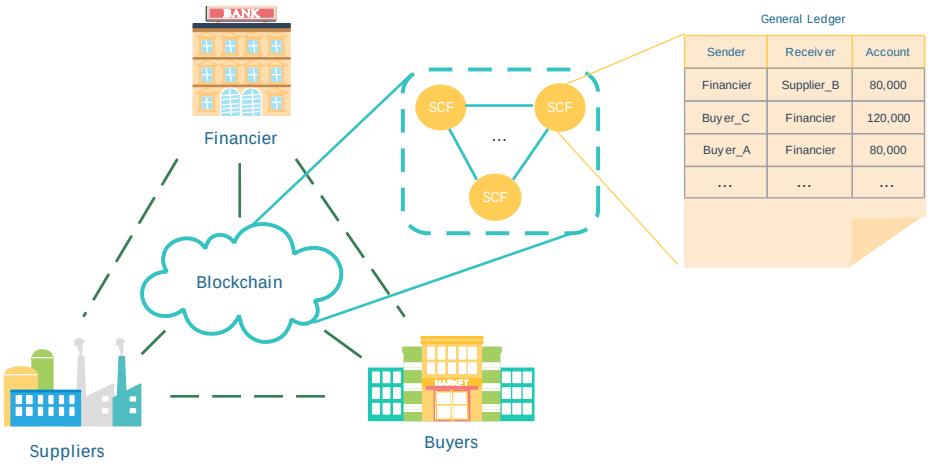

Fig. 1: The architecture of blockchain-based supply chain finance system

In order to protect suppliers' commercial interests, we consider a direct but efficient method, i.e., hiding the transferred and balance involved in the ledger. If we can conceal the amount in both the user's account and the transaction, the

threats of amount-change analysed by compromised SCFs or other adversaries will be mitigated.

There has been progress in designing privacy-preserving schemes (e.g., Confidential Transaction [Max15], Zerocash [BCG+14], Monero [Sab13]), details of which will be described in the Section 1.2. Most of them focus on hiding the transaction accounts via several cryptographic techniques (e.g., cryptographic commitment, zero-knowledge proof, ring signature, etc). Notice that the coins of them are in Bitcoin's UTXO (Unspent Transaction Outputs) architecture and a user's balance is the sum of all outputs regulated by wallet. In the UTXO architecture, your wallet will simultaneously create a new address for the change you are owed when greater coins are sent to another user. Subsequently, the emergency of Ethereum [Woo14] has introduced an innovation architecture (the ACCOUNT architecture), which relies on global state storage of accounts, balances, code and storage (i.e. the user's balance now is kept as global state). Analogous to a bank account, there is a debit and corresponding credit to the states with a transaction.

When considering the privacy of user's balance, previous UTXO-based researches may not work for the following reasons. Firstly, the cryptographic commitment scheme may bring about the difficulty for the concurrent balance-updating in the system. Secondly, high computational complexity greatly restricts their application in the lightweight but widespread used devices (e.g. mobile phone). Finally, none of them support the smart contract system of Ethereum, which offers more flexible and arbitrary trading operations running in the blockchain. Thus, we are motivated to propose a mechanism with the ACCOUNT architecture for creating an expressive decentralized smart contract (DSC) system with the above hiding and updating.

In order to achieve hiding and timely updating operations to the balance, we employ the homomorphic encryption (HE) schemes. Both the amount of transferred records and balance are encrypted by the HE algorithms and stored in ciphertext. The homomorphism of HE allows the miners to directly update the balance in ciphertext without the need of decryption, that is, given encryptions $E(v_1), E(v_2), \cdots, E(v_t)$ of the balance $v_1, v_2, \cdots, v_t$, the miners can efficiently compute a ciphertext of $f(v_1, v_2, \cdots, v_t)$, where $f(\cdot)$ is an efficiently computable function (this function is mainly related to addition or substraction operation in our paper). In addition, we propose a zero knowledge (ZK) proof tool to prove two basis statements required by a transaction. One is "equivalence" (i.e. Alice's balance decreases v and Bob's should correctly add v when Alice transfers money v to Bob) and the other is "enough"(i.e. Alice's balance should not be less than v if she want to transfer money v to others). Thus, in this paper, we not only find an applicable HE scheme, but also design the corresponding ZK scheme to support DSC system with the balance hiding property.

1.1 Our contributions

In this section, we summarize the contributions of this paper as follows:

1. The main contribution of this research is to introduce a priori mechanism enabling programmability (i.e. decentralized smart contract) with balance hiding property under the ACCOUNT architecture. This mechanism can be applied in various financial scenarios and can also work when a system involves economic dealings or even change in digital assets.
2. We utilize a public key encryption scheme with homomorphic property to hide the balance and transaction amount, and design a non-interactive zero knowledge (NIZK) scheme to prove the validity of the transactions. The in-depth security proof shows that our proposed scheme is provably secure under the random oracle model.
3. We analyze the performance of the proposed scheme both in asymptotic and practical terms, and also implement it on the personal computer. The encouraging result indicates that our scheme is practicable and maneuverable in the mentioned actual applications.

1.2 Related Work

In this subsection, we briefly review some existing cryptographic techniques around the privacy protection in the blockchain, however which are not suitable to the demand of balance confidentiality and timely updating in our system.

Bitcoin Core Developer Gregory Maxwell [Max15] first conceptualizes Confidential Transaction as a solution for keeping the transaction amounts unrevealed. Their solution is based on the Pedersen commitment scheme [P+91], where the transaction amounts are masked by random blinding factors before sent to the recipients and lately notarized by the recipients. The clear thing is that, these masked amounts still can be used for certain types of calculations, which means that all inputs and outputs of a transaction can be added up respectively and these two sums can be compared to ensure trade-off during the verifying process without revealing the real values.

Ring Confidential Transaction (RingCT) is another variant CT approach for hiding transaction amounts. Collaborated with the linkable ring signature scheme [LWW04], Monero [NM+16] (another proof-of-work cryptocurrency) achieves the requirements of decentralization, privacy and anonymity. Similar to [Max15], the RingCT scheme improves the privacy of the blockchain by allowing the amounts sent in a transaction to be concealed in an anonymous set. In addition, the linkable mechanism is equipped to ensure any double-spending behaviors can be detected timely.

However, the CT-based schemes uses blinding factors for inputs and outputs, which are picked in special so that they add up correctly. This may cause lower randomness and reduce the security of the whole scheme. In addition, the blinding factors may need to be somehow synchronized to both sides, which may lead to concurrency problems and have slightly difficulty when implementing into a financial system (e.g. BDSCF).

Another cryptographic method is zero-knowledge proof. Zerocash [BCG+14] employs the zero-knowledge succinct non-interactive argument of knowledge (zk-SNARKs) [BSCG+13] and cryptographic commitment schemes to reach the un-

linked transaction and confidential amount. The transfer transaction consists of a cryptographic commitment to a new coin, which specifies the coin's value, owner address and unique serial number. When consuming the input coins, zero-knowledge proofs and serial numbers are needed to prove the ownership of the input coins and the trade-off between the inputs and outputs. Recently, Zerocash can achieve the highest level of privacy protection and anonymity of the cryptocurrency based on UTXO architecture. However, when using this method in our account-based system, there are two main drawbacks. One is that the cryptographic commitments generated by the one-way hash functions do not support the ACCOUNT architecture, since homomorphic operations are not considered while Zerocash was designed. The other is that the proof generation process in this scenario is rather expensive which leads to the worse efficiency and not suitable for the lightweight devices (e.g. mobile phones).

Instead of UTXO architecture, Ethereum [Woo14] introduce the ACCOUNT architecture (mentioned in Section 1) and a decentralized arbitrary user-defined programming system running in the blockchain, named of smart contract system. Followed the idea of smart contract, Kosba et al. [KMS+16] implements a cryptographic suite that can blind transactions with programmable logic. It applies smart contract to store the committed coins generated by the users and determine the payout distribution. Once the users open the commitments and uncover the information to the manager (who is trusted not to disclosed the user's private data), the manager then interact with the smart contract to generate new coins and pay to the recipients. The new coins will lately be submitted to the blockchain with zero knowledge proofs for its legality. This scenario provides programmability without exposing explicit transaction information to the public. However, since the manager always knows users' quotes, this scheme is not suitable for the privacy protection in terms of transaction amount and balance in our scenario.

1.3 Organization

We organize the remainder of this paper as follows. Section 2 contains background materials such as bilinear pairings, homomorphic encryption, Σ-protocols, non-interactive zero knowledge proofs and some complexity assumptions. In section 3, we describe our NIZK scheme, including the construction with its corresponding proof. Section 4 discusses the concrete instantiation of our scheme and demonstrates a comparison with previous scheme. Section 5 concludes this paper and gives future directions.

2 Preliminaries

In this section we give basic definitions of cryptographic primitives including required tools and complexity assumptions, along with some properties if necessary.

Notations. If n is an integer, we denote $[n] = \{1, \ldots, n\}$. For any set S, $x \leftarrow_{\$} S$ means sampling uniformly at random some element x from the set S. Besides, for any distribution D, $x \leftarrow_{\$} D$ means sampling x from the probability distribution D, and $v \in_R D$ denotes that variable v is uniformly random in D. We write $y = A(x; r)$ to represent that an algorithm A takes input x and randomness r, output y. The formula $y \leftarrow A(x)$ means picking randomness r uniformly at random and setting $y = A(x; r)$.

In this paper, we denote by n the security parameter, and abbreviate probabilistic polynomial-time as PPT. A function $\epsilon(n)$ is negligible in n if $\epsilon(n) = o(1/n^c)$ for all $c \in \mathbb{N}$. $\epsilon(n) = \mathsf{negl}(n)$ denotes that $\epsilon(n)$ is a negilible function in n, and $\epsilon(n) = \mathsf{poly}(n)$ denotes that $\epsilon(n)$ is a polynomial function in n.

For a group G, we denote by $||G||$ the size of its arbitrary element.

For any two distribution ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$, we write $\{X_n\}_{n\in\mathbb{N}} \overset{c}{\approx} \{Y_n\}_{n\in\mathbb{N}}$ to represent the two distribution ensembles are computational indistinguishable with security parameter n.

2.1 Cryptographic primitives

Bilinear groups. We call $\mathcal{G}_{bp}(1^n)$ the bilinear group generator which takes a security parameter as input and outputs a description of a bilinear group $\mathsf{gk} = (p, G_1, G_2, G_T, e, g_1, g_2)$ such that p is a n-bit prime. We follow the notation of [BLS01]:

- G_1, G_2, G_T are multiplicative cyclic groups of order p. The elements g_1, g_2 generates G_1, G_2 respectively.
- $e : G_1 \times G_2 \to G_T$ is a nondegenerate bilinear map, and $e(g_1, g_2)$ generates G_T.
- $\phi : G_2 \to G_1$ is a computable isomophism, and $g_1 = \phi(g_2)$.
- $\forall a, b \in \mathbb{Z}, e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- It is efficient to compute group operations, compute the bilinear map, and decide the membership in G_1, G_2 and G_T.

Remark 1. In some cases, $G_1 = G_2 = G$ and $g_1 = g_2 = g$, where the bilinear group generator outputs (p, G, G_T, e, g). Under different intractability problems, the respective multiplicative groups are of prime order or composite order, for instance, subgroup decision problem needs groups of composite order, and decision linear problem needs groups of prime order. However, Freeman in his work [Fre10] proposed an abstract framework to convert some pairing-based cryptosystems from composite-order groups to prime-order groups.

DLIN assumption. With $g_1 \in G_1$ discribed above, let f, h, g be its arbitrary generators. For a triple $(s_1, s_2, s_3) \in G_1^3$ w.r.t the basis (f, h, g), if there exist $r, s \in \mathbb{Z}_p$ such that $s_1 = f^r, s_2 = h^s, s_3 = g^{r+s}$, we call the triple linear. The decision linear assumption proposed in [BBS04] states that no PPT algorithm can distinguish g^{r+s} from g' (where $g' \leftarrow_{\$} G_1$).

Definition 1 (DLIN Assumption). *The decision linear assumpion (DLIN) holds in G_1 if for all non-uniform* PPT $\mathcal{A}$ *we have*

$$\left| \begin{array}{l} \Pr[f,h,g \leftarrow_{\$} G_1, r,s \leftarrow_{\$} \mathbb{Z}_p : \mathcal{A}(f,h,g,f^r,h^s,g^{r+s}) = 1] \\ -\Pr[f,h,g,g' \leftarrow_{\$} G_1, r,s \leftarrow_{\$} \mathbb{Z}_p : \mathcal{A}(f,h,g,f^r,h^s,g') = 1] \end{array} \right| = \mathsf{negl}(n)\,.$$

A public-key encryption(PKE) scheme consists of three PPT algorithms (KGen, $\mathsf{Enc}, \mathsf{Dec}$) which indicates key generation, encryption, and decryption. We require that $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^n)$ and for any valid plaintext m and randomness r, $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m;r)) = m$. A PKE scheme is IND-CPA secure(a.k.a. semantically secure [GM82]) if

$$\Pr\left[\begin{array}{c} b \leftarrow_{\$} \{0,1\}, (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^n) \\ m_0, m_1 \leftarrow_{\$} \{0,1\}^n, c = \mathsf{Enc}_{\mathsf{pk}}(m_b;r) \;:\; b = b' \\ b' \leftarrow \mathcal{A}(1^n,\mathsf{pk},c) \end{array}\right] = \mathsf{negl}(n)\,.$$

In this paper, we use a PKE scheme with homomorphic property, called Homomorphic Encryption (abbreviate as HE), to hide the balance and transaction.

Definition 2. *The HE scheme comprises a triple of algorithms* ($\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}$)*:*

- $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^n)$*:* $h \leftarrow_{\$} G_1, x, y \leftarrow_{\$} \mathbb{Z}_p, \mathsf{sk} = (x,y), \mathsf{pk} = (X,Y) = (g_1^x, g_1^y, g_1, h)$.
- $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)$*: Randomly sample* $r,s \leftarrow_{\$} \mathbb{Z}_p$*, set* $C_1 = X^r, C_2 = Y^s, C_3 = g_1^{r+s} \cdot h^m$*, then* $C = (C_1, C_2, C_3)$.
- $m = \mathsf{Dec}_{\mathsf{sk}}(C)$*: Parse* C *into* (C_1, C_2, C_3)*, compute* $h_m = C_3/(C_1^{1/x} \cdot C_2^{1/y})$. *One can efficiently get* $m = log_h^{h_m}$ *if the plaintext space is small.*

The correctness of the cryptosystem is straightforward. Note that for efficient decryption we require the message space to be small for solving the discrete logrithm problem. Assuming DLIN assumption, our encryption scheme is IND-CPA secure.

Remark 2. The third part of the ciphertext, $C_3 = g_1^{r+s} \cdot h^m$, employs the form of Perdersen commitment [P+91]. While the whole ciphertext owns a form like linear encryption posed in [BBS04], there are significant differences in some respects including the message space and the pair of keys.

q-SDH assumption. With the bilinear group $\mathsf{gk} = (p, G_1, G_2, G_T, e, g_1, g_2) \leftarrow \mathcal{G}_{bp}(1^n)$, we pay attention to the q-strong Diffie-Hellman (q-SDH) assumption proposed by Boneh and Boyen in [BB04]. Later the work of [TS10] gives more information about the assumption.

Definition 3 (q-SDH Assumption). *The q-Strong Diffie-Hellman (q-SDH) assumption associated to a bilinear group* gk *holds if for all non-uiform* PPT $\mathcal{A}$*, we have*

$$\Pr[x \leftarrow_{\$} \mathbb{Z}_p : (c, g_1^{1/(x+c)}) \leftarrow \mathcal{A}(g_1, g_1^x, \ldots, g_1^{x^q}, g_2, g_2^x)] = \mathsf{negl}(n)\,;$$

where $c \in \mathbb{Z}_p$.

In a signature scheme, there exist a triple of polynomail-time algorithms $(KeyGen, Sign, Verify)$ for generating keys, signning, and verifing signatures, respectively. The conditions should be satisfied:

- $(sk, vk) \leftarrow KeyGen(1^n)$,
- $Verify_{vk}(m, Sign_{sk}(x)) = 1$.

As to the security of signature schemes, we only consider existential unforgeability under a weak chosen message attack. In this model, the advesary submits q queries $m_1, \ldots, m_q$ to the challenger for asking their signatures. The challenger runs $(sk, vk) \leftarrow KeyGen(1^n)$ and sends vk to the adversary, together with signatures $\sigma_1, \ldots, \sigma_q$ on $m_1, \ldots, m_q$. We say the adversary wins if it outputs a signature σ' such that $Verify_{vk}(m', \sigma') = 1$ and $m' \notin \{m_1, \ldots, m_q\}$. A signature scheme is said to be secure under a weak chosen message attack if no PPT adversary wins the game with non-negligible probability.

Definition 4 (Boneh-Boyen Signature). *Boneh-Boyen signature consists of three polynomial-time algorithms:*

- $(sk, vk) \leftarrow KeyGen(1^n)$*: The randomized key generation algorithm takes the security parameter n as input, randomly choose $\lambda \leftarrow_{\$} \mathbb{Z}_p$, set $(sk, vk) = (\lambda, g_2^{\lambda})$.*
- $\sigma \leftarrow Sign_{sk}(m)$*: The deterministic signing algorithm uses the private signing key sk and input m. It outputs $\sigma = g_1^{\frac{1}{\lambda+m}}$.*
- $\{0, 1\} \leftarrow Verify_{vk}(m, \sigma)$*: Given the public verification key vk, the deterministic verification algorithm outputs 1 if $e(\sigma, vk \cdot g_2^m) = e(g_1, g_2)$, and 0 otherwise.*

Under the q-SDH assumption, the Boneh-Boyen signature scheme is secure against existential forgery under a weak chosen message attack, which is sufficient enough for our goal. For more detail information on this proof, see [BB04].

Σ-Protocol. Let $R = \{(x, w)\}$ be a binary relation which can be efficiently computed such that $|w| = \mathsf{poly}(n)(|x|)$. Here, x is a statement and w is a witness. Let $L_R = \{x : \exists w \text{ s.t. } (x, w) \in R\}$ be an NP language. A Σ-protocol $\Pi = (a, c, z)$ introduced in [Cra96] is a 3-round public-coin protocol between two efficient parties (P, V): the prover P sends the first message $a \leftarrow \mathsf{P}(x)$; when received a, the verifier V sends $c \leftarrow_{\$} \{0, 1\}^n$ to P; the prover's last message $z \leftarrow \mathsf{P}(x, a, c)$. The transcript (a, c, z) is accepting iff. $\mathsf{V}(x, a, c, z) = 1$. For more information about Σ-protocols, see [HL10,Dam10]. Formally:

Definition 5 (Σ-Protocol). *A 3-round public-coin protocol $\Pi = (a, c, z)$ is a Σ-protocol for language L_R if the following conditions hold:*

- *Completeness: If P and V execute the protocol on input x and private input w to P in which $(x, w) \in R$, then V always accepts.*
- *Special soundness: For any statement x, given two accepting transcripts on input x: $(a, c, z), (a, c', z')$ where $c \neq c'$, there exists a PPT algorithm Ext which can compute the witness w s.t. $(x, w) \in R$.*

– *Special honest verifier zero knowledge (SHVZK): There exists a* PPT *algorithm* Sim*, on input x and a challenge c, can perfectly simulate the conversations between the honest* P, V *on input x. Formally speaking,*

$$\Big\{\mathsf{Sim}(x,c)\Big\}_{x\in L_R, c\in\{0,1\}^n} \equiv \Big\{ < \mathsf{P}(w), \mathsf{V}(c) > (x)\Big\}_{x\in L_R, c\in\{0,1\}^n};$$

where $\mathsf{Sim}(x,c)$ *represents the output of simulator* Sim *on input x and c, and* $< \mathsf{P}(w), \mathsf{V}(c) > (x)$ *denotes the real output transcript of the protocol.*

NIZK argument. A non-interactive argument system for a relation R consists of two efficient parties: a prover P and a verifier V. Taking (x, w) as input, P produces a proof π, and sends it to V. The verifier V takes as input (x, π) and outputs 1 if the proof is acceptable and output 0 if rejecting the proof. We call (P, V) a non-interactive argument system for R if it owns the completeness and soundness properties defined below.

A non-interactive zero knowledge (NIZK) argument system proposed in [BFM88] is a non-interactive argument system which leaks no information to the verifier except the validity of the statement.

Definition 6 (NIZK **Arguments).** *A triple of* PPT *algorithms (*K*,* P*,* V*) is called a* NIZK *argument system for language L_R if the conditions described below hold:*

– *Completeness: For all $crs \leftarrow \mathsf{K}(1^n)$ and all $(x, w) \in R$, we have:*

$$\Pr[\pi \leftarrow \mathsf{P}(x, w, crs) : \mathsf{V}(x, \pi, crs) = 1] = 1 - \mathsf{negl}(n)\,.$$

– *(Adaptive) Soundness: For all non-uniform* PPT *prover* P^**, the probability*

$$\Pr[crs \leftarrow \mathsf{K}(1^n), (x, \pi) \leftarrow \mathsf{P}^*(crs) : x \notin L_R \wedge \mathsf{V}(x, \pi, crs) = 1] = \mathsf{negl}(n)\,.$$

– *(Adaptive) Zero-Knowledge: There exists a* PPT *simulator $S = (S_1, S_2)$, such that for all stateful non-uniform* PPT *adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we have*

$$\left| \begin{array}{l} \Pr\left[\begin{array}{c} crs \leftarrow \mathsf{K}(1^n) \\ (x, w) \leftarrow \mathcal{A}_1(crs) \\ \pi \leftarrow \mathsf{P}(crs, x, w) \end{array} : \begin{array}{c} (x, w) \in R \wedge \\ \mathcal{A}_2(crs, \pi) = 1 \end{array} \right] \\ - \Pr\left[\begin{array}{c} (crs, td) \leftarrow S_1(1^n) \\ (x, w) \leftarrow \mathcal{A}_1(crs) \\ \pi \leftarrow S_2(crs, x, td) \end{array} : \begin{array}{c} (x, w) \in R \wedge \\ \mathcal{A}_2(crs, \pi) = 1 \end{array} \right] \end{array} \right| = \mathsf{negl}(n)\,.$$

We call the NIZK argument perfect zero-knowledge if the above probability equals 0*.*

The above definition discribes the NIZK argument in the common reference string (CRS) model which is generated by a trusted third party. Using Fiat-Shamir heuristic [FS86] and a secure hash function H, a Σ-protocols can be transformed into a NIZK argument in the following way: P computes a, applies

H to a and obtains the challenge $c = \mathsf{H}(a)$, then computes z according to the Σ-protocol and send the proof (a, c, z) to V. One can prove the property of soundness and zero-knowledge of the new protocol in the random oracle (RO) model [BR93] where we replace H by a random oracle in the way of [FS86].

We will construct NIZK in the common reference string model by applying Fiat-Shamir heuristic to a Σ-protocol, which allows us to achieve perfect zero knowledge without relying on a random oracle, though the soundness of our construction is proved in the random oracle model.

2.2 Decentralized smart contracts over blockchains

A smart contract is a piece of code which is stored in the blockchain network on each participant node. It can be seen as a digital version of a traditional contract. The property of decentralization of blockchain has improved the development of smart contracts. Assume in a payment system which owns the ACOUNT architecture, user A want to transfer t coins to user B. Then we can deploy the transfer action and some necessary checks in the blockchain as a smart contract to automatically execute the operation in the following way. User A posts a transaction on the blockchain that basically says

Transfer t of my coins to B, and σ is a signature of t.

Being triggered by this message, the smart contract first checks the validity of the signature, and that A has more than t coins, If so does the transfer action and publishes the transation on the blockchain, otherwise it ignores the transation.

In the simplified transaction above, anyone can learn the money t being transfered from A to B (i.e. there is no guarantee in the privacy of users' balance and transaction amount). But we can get around this problem by changing the verification procedure accordingly deployed in the smart contract. Suppose that every user's balance is encrypted with a homomorphic encryption scheme $E(\cdot)$ and saved on the ledger in the form of ciphertext. A could post the transaction as follows.

Transfer $E(t)$ of my coins to B, here is a non-interactive zero knowledge proof π to prove the correctness of $E(t)$ and that my balance is larger than t.

In next section, we will introduce the abstract framework of a decentralized smart contracts system that allows the users to transfer money with privacy of balance and transaction amount and give a concrete construction of its main building block, a NIZK argumment system.

3 NIZK Argument and DSC Scheme

In this section, we introduce the framewrok of a decentralized smart contract (DSC) system with the property of hiding balance and transaction amount and present a new NIZK argument for the two basic statements introduced in section 1 to fulfill the DSC system. We also prove the correctness and security of

the NIZK argument. With respect to the "equivalence" statement, the basic idea is that we first construct a Σ-protocol to prove the given two ciphertexts corresponding to some transaction amount own a same plaintext which is encrypted with an HE scheme. Then using Fiat-Shamir heuristic method, we build a NIZK protocol between the two parties. As the second statement, "enough", we utilize the technique borrowed from [CC+08] to construct a range proof. The main idea of the range proof is that for a secret $t \in [0, u^l]$, the prover writes it in u-ary notation (i.e., $t = \sum_{j=0}^{l} t_j \cdot u^j$) and shows that each element t_j in the range $[0, u)$. Now the key technique to use is a set membership proof protocol. We get the full NIZK scheme acting as a building block in our DSC system when put the two proofs together. Note that we also put forward a system public parameter generated once serving as common reference string in the NIZK argument which can be reused in other proofs.

3.1 Decentralized smart contract system

Suppose a NIZK argument with a prover P and a verifier V, we deploy the verification procedure in the blockchain to obtain a smart contract which can automatically do the transfer operation. Following is a formal description of a DSC system.

3.2 The construction of NIZK and its security

For the sake of simplicity, we only consider two parties A and B in the smart contracts. Suppose that the plaintext space is $[0, 2^{\mathcal{L}})$, where $\mathcal{L} = 10 \times l$. In order to construct a concrete NIZK argument, we leave the implementation of Setup and PartyInitial in DSC system to the NIZK argument system:

Setup. $(p, G_1, G_2, G_T, e, g_1, g_2) \leftarrow G_{bp}(1^n)$ is a bilinear group as described in Section 2.1. Let $h = g_1^{\omega}$ be another generator of G_1, where $\omega \leftarrow_{\$} \mathbb{Z}_p$. Let $g_T = e(g_1, g_2)$ be a generator of G_T. Given a key pair $(sk = \lambda, vk = g_2^{\lambda})$ of Boneh-Boyen signature scheme, we compute the signatures of the integers between 0 and $2^{10} - 1$:

$$\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{2^{10}-1}) = (g_1^{\frac{1}{\lambda}}, g_1^{\frac{1}{\lambda+1}}, \ldots, g_1^{\frac{1}{\lambda+2^{10}-1}});$$

and the following bilinear maps:

$$T = (T_0, T_1, \ldots, T_{2^{10}-1}) = (e(\sigma_0, g_2), e(\sigma_1, g_2), \ldots, e(\sigma_{2^{10}-1}, g_2));$$

The public parameter now is the tuple of $PP = (p, G_1, G_2, G_T, e, g_1, h, g_2, g_T, vk, \sigma, T)$ which also serves as a common reference string[1].

[1] In order to improve the prover's efficiency, we precompute σ, T in the Setup procedure.

Decentralized Smart Contract System

- **Setup** The algorithm Setup produces a system public parameter:
  - input: security parameter n
  - output: a system public parameter PP
- **PartyInitial** The algorithm PartyInitial generates every user's (say A) information using a homomorphic encryption scheme:
  - input: PP
  - output: $PK_A, SK_A, C_A = \mathsf{Enc}_{PK_A}(t_A)$

  The public key PK_A also links to the address for receiving coins. Only the ciphertext C_A of A's balance t_A is stored in the account book.
- **Transfer** The algorithm Transfer is invoked when some party A transfer t coins to B.
  - input: $PP, t, t_A, PK_A, C_A, PK_B$
  - A generates a transfer statement x and according NIZK proof $\pi \leftarrow \mathsf{P}(x, PP, w)$, and posts them to the blockchain.
- **Redeem** The algorithm Redeem deployed in the blockchain for automatically transfer will be triggered by the Transfer algorithm.
  - input: PP, x, π
  - if $\mathsf{V}(PP, x, \pi) = 1$, then T finds the sender A and the receiver B from the statement x, publishes the transaction and does the tranfer operation:

    Update A's balance to $C'_A = C_A / C_t$ and B's balance to $C'_B = C_B \cdot \hat{C}_t$;

    otherwise, T ignores it.

Fig. 2: DSC System

PartyInitial. Parties in the protocol use the homomorphic encryption described in Definition 2. Consider a party A, its public key, private key, and encryption algorithm is as follows:

- Private key: $SK_A = (x_{A1}, x_{A2}) \in \mathbb{Z}_p^2$,
- Public key: $PK_A = (X_{A1}, X_{A2}) \in G_1^2$, where $X_{A1} = g_1^{x_{A1}}, X_{A2} = g_1^{x_{A2}}$,
- Encryption: $\mathsf{Enc}_{PK_A}(m; (y_1, y_2)) = (C_1 = X_{A1}^{y_1}, C_2 = X_{A2}^{y_2}, C_3 = g_1^{y_1+y_2} \cdot h^m)$, where (y_1, y_2) denotes the randomness. For any valid ciphertext c, one who has corresponding private key can decrypt it efficiently, since the plaintext space is $[0, 2^{\mathcal{L}})$ where $2^{\mathcal{L}} \ll q$. In this paper, we consider the plaintext space of size $[0, 2^{30})$ (i.e., $\mathcal{L}$ is set to be 30).

Proof generation by P. Party A with balance t_A does the following operations, when transfering t to party B:

1. From the account book, A gets the ciphertext of t_A, $\tilde{C} = (\tilde{C}_1, \tilde{C}_2, \tilde{C}_3) = (X_{A1}^{\tilde{y}_1}, X_{A2}^{\tilde{y}_2}, g_1^{\tilde{y}_1+\tilde{y}_2} \cdot h^{t_A})$. Note that A probably does not know $\tilde{y}_1, \tilde{y}_2$. After randomly sampling $y_1, y_2 \leftarrow_{\$} \mathbb{Z}_p$, A generates the following ciphertext of t under A's public key (X_{A1}, X_{A2}):

$$C = (C_1, C_2, C_3) = (X_{A1}^{y_1}, X_{A2}^{y_2}, g_1^{y_1+y_2} \cdot h^t).$$

   With the same randomness y_1, y_2, A generates the ciphertext of t under B's' public key:

$$\hat{C} = (\hat{C}_1, \hat{C}_2, \hat{C}_3 = C_3) = (X_{B1}^{y_1}, X_{B2}^{y_2}, g_1^{y_1+y_2} \cdot h^t).$$

2. Define the language L proved by P as follows:
   The statement $x = (C, \hat{C}, PK_A, PK_B, \tilde{C}) \in L$ if there exists a witness $w = (sk_A = (x_{A1}, x_{A2}), y_1, y_2, t_A, t)$, such that
   (a) $\frac{C_i}{\tilde{C}_i} = \left(\frac{X_{Ai}}{X_{Bi}}\right)^{y_i}$, for $i = 1, 2$;
   (b) $C_3 = g_1^{y_1+y_2} \cdot h^t$;
   (c) $\frac{\tilde{C}_3}{C_3} = \tilde{C}_1^{\frac{1}{x_{A1}}} \cdot \tilde{C}_2^{\frac{1}{x_{A2}}} \cdot g_1^{-y_1-y_2} \cdot h^{t_A - t}$;
   (d) $t \in [0, 2^{\mathcal{L}}), t' = t_A - t \in [0, 2^{\mathcal{L}})$,
   where $t = \sum_{j=0}^{l-1} t_j \cdot (2^{10})^j, t' = \sum_{j=0}^{l-1} t'_j \cdot (2^{10})^j, 0 \le t_j, t'_j < 2^{10}$;
   OR there exists $\omega \in \mathbb{Z}_p$, such that
   (e) $h = g_1^{\omega}$.
3. Taking PP as common input, A generates a NIZK proof for the above statement with private input (sk_A, y_1, y_2, t_A, t) in the following way:
   For the proof generation of Equation (a), (b), (c), a Σ-protocol can be used. Equation (d) can be proved by utilizing the range proof in [CC+08]. Equation (e) holding a trapdoor ω is designed for the simulator.
   Randomly sample $r_1, r_2, \ell, k \leftarrow_{\$} \mathbb{Z}_p$, compute $R_i = \left(\frac{X_{Ai}}{X_{Bi}}\right)^{r_i}, i = 1, 2$.
   For $j = 0, 1, \ldots, l-1$, randomly sample $v_j, v'_j, s_j, w_j, q_j, m_j \leftarrow_{\$} \mathbb{Z}_p$, then compute:

$$V_j = \sigma_{t_j}^{v_j}, V_j' = \sigma_{t_j'}^{v_j'};$$
$$D_1 = \prod_{j=0}^{l-1} \left(h^{(2^{10})^j \cdot s_j}\right) \cdot g_1^{r_1+r_2};$$
$$D_2 = \prod_{j=0}^{l-1} \left(h^{(2^{10})^j \cdot w_j}\right) \cdot \tilde{C}_1^{\ell} \cdot \tilde{C}_2^{k} \cdot g_1^{-r_1-r_2};$$
$$a_j = T_{t_j}^{-s_j \cdot v_j} \cdot g_T^{q_j}, a_j' = T_{t_j'}^{-w_j \cdot v_j'} \cdot g_T^{m_j};$$

Randomly sample $\hat{c} \leftarrow_{\$} \mathbb{Z}_p, \hat{z} \leftarrow_{\$} \mathbb{Z}_p$, and set $\alpha = g_1^{\hat{z}}/h^{\hat{c}}$.

Let $a = (R_1, R_2, \{V_j, V_j'\}_{j=0}^{l-1}, D_1, D_2, \{a_j, a_j'\}_{j=0}^{l-1}, \alpha)$ represent the first message of a Σ-protocol. Applying H to a,

$$\tilde{c} = \mathsf{H}(a);$$

where H represents a random oracle which can be instantiated by a secure hash function.

Let $c = \tilde{c} + \hat{c}$ represent the challenge value of a Σ-protocol.

Compute (all modulo p):

$$z_1 = r_1 - c \cdot y_1; \qquad z_2 = r_2 - c \cdot y_2;$$
$$z_{v_j} = q_j - c \cdot v_j; \qquad z_{v_j'} = m_j - c \cdot v_j';$$
$$z_{t_j} = s_j - c \cdot t_j; \qquad z_{t_j'} = w_j - c \cdot t_j';$$
$$z_\ell = \ell - \frac{c}{x_{A1}}; \qquad z_k = k - \frac{c}{x_{A2}};$$

Finally, A sends to B the proof:

$$\pi = \Big(R_1, R_2, \{V_j, V_j'\}_{j=0}^{l-1}, D_1, D_2, \{a_j, a_j'\}_{j=0}^{l-1}, \alpha, c, z_1, z_2, \{z_{v_j}, z_{v_j'}\}_{j=0}^{l-1}, \{z_{t_j}, z_{t_j'}\}_{j=0}^{l-1}, z_\ell, z_k, \hat{z}\Big).$$

Proof verification by V. Upon receiving a proof π, the verifier V parses π into the form as above, then computes $\tilde{c}$ and $\hat{c} = c - \tilde{c}$. With the common input PP, $\forall i = 1, 2; j = 0, 1, \ldots, l-1$, V checks whether the following conditions

hold:

$$R_i = \left(\frac{C_i}{\hat{C}_i}\right)^c \cdot \left(\frac{X_{Ai}}{X_{Bi}}\right)^{z_i}; \tag{1}$$

$$D_1 = \prod_{j=0}^{l-1} \left(h^{(2^{10})^j \cdot z_{t_j}}\right) \cdot C_3^c \cdot g_1^{z_1+z_2}; \tag{2}$$

$$D_2 = \prod_{j=0}^{l-1} \left(h^{(2^{10})^j \cdot z_{t'_j}}\right) \cdot \left(\frac{\tilde{C}_3}{C_3}\right)^c \cdot \tilde{C}_1^{z_\ell} \cdot \tilde{C}_2^{z_k} \cdot g_1^{-z_1-z_2}; \tag{3}$$

$$a_j = e(V_j, vk)^c \cdot e(V_j, g_2)^{-z_{t_j}} \cdot g_T^{z_{v_j}}, a'_j = e(V'_j, vk)^c \cdot e(V'_j, g_2)^{-z_{t'_j}} \cdot g_T^{z_{v'_j}}; \tag{4}$$

$$g_1^{\hat{z}} = \alpha \cdot h^{\hat{c}}; \tag{5}$$

Theorem 1. *Assuming the DLIN, q-SDH assumptions, the protocol described above is a* NIZK *argument with perfect completeness, perfect zero-knowledge and computational soundness in the RO model. Furthermore, perfect zero-knowledge holds in the standard CRS model.*

Proof. We prove each direction separately.

Perfect Completeness. Perfect completeness follows by direct verification, see appendix A for more details.

Soundness. The soundness follows from the property of special soundness of Σ-protocols and the unforgeability of the Boneh-Boyen signature. If a PPT prover P^* generates an accepted argument π for an invalid statement, where

$$\pi = \Big(a = \big(R_1, R_2, \{V_j, V'_j\}_{j=0}^{l-1}, D_1, D_2, \{a_j, a'_j\}_{j=0}^{l-1}, \alpha\big), c, \\ z_1, z_2, \{z_{v_j}, z_{v'_j}\}_{j=0}^{l-1}, \{z_{t_j}, z_{t'_j}\}_{j=0}^{l-1}, z_\ell, z_k, \hat{z}\Big);$$

Then, we construct such an extractor Ext: Upon seeing the argument, Ext rewinds P^* to the oracle query $H(a)$ that returned $\tilde{c}$. It then reprogram the random oracle such that $\tilde{c}' = H(a)$ with $\tilde{c} \neq \tilde{c}'$ and continue the execution of P^* with the modified random oracle. In expected polynomial time, another valid argument appears:

$$\pi' = (a, c' = \tilde{c}' + \hat{c}, z'_1, z'_2, \{z'_{v_j}, z'_{v'_j}\}_{j=0}^{l-1}, \{z'_{t_j}, z'_{t'_j}\}_{j=0}^{l-1}, z'_\ell, z'_k, \hat{z}).$$

The witness can be extracted by computing (for $i = 0, 1; j = 0, 1, \ldots, l-1$):

$$y_i = \frac{z_i - z'_i}{c' - c}, t_j = \frac{z_{t_j} - z'_{t_j}}{c' - c}, t'_j = \frac{z_{t'_j} - z'_{t'_j}}{c' - c}, x_{A1} = \frac{c' - c}{z_\ell - z'_\ell}, x_{A2} = \frac{c' - c}{z_k - z'_k}.$$

Conditioned on the extracted witness, if $t \notin [0, 2^{\mathcal{L}})$ or $t' \notin [0, 2^{\mathcal{L}})$, then we can successfully attack the Boneh-Boyen signature in a weak chosen message attack model with non-negligible probability, taking P^* as a subroutine.
A contradiction occurs.

Perfect Zero-Knowledge. Unlike using the standard Fiat-Shamir heuristic method, in our construction, we prove perfect zero-knowlege without relying on a random oracle. To prove the zero-knowlege, we construct a simulator Sim to prove statement $h = g_1^w$, see Fig. 3.

Simulator for the New NIZK Argument

- Just do like the procedure of **Setup** and output: $(PP, td) = ((p, G_1, G_2, G_T, e, g_1, h, g_2, g_T, vk, \sigma, T), \omega)$ where $h = g_1^{\omega}$.
- Choose randomly $t, t' \leftarrow_{\$} [0, 2^{\mathcal{L}}); v_j, v_j' \leftarrow_{\$} \mathbb{Z}_p$, and write t, t' in base-2^{10}:

$$t = \sum_{j=0}^{l-1} (2^{10})^j \cdot t_j, t' = \sum_{j=0}^{l-1} (2^{10})^j \cdot t_j',$$

  then set $V_j = \sigma_{t_j}^{v_j}, V_j' = \sigma_{t_j'}^{v_j'}$, where $j \in \{0, 1, \ldots, l-1\}$.
- Choose $c, z_1, z_2, z_{v_j}, z_{v_j'}, z_{t_j}, z_{t_j'}, z_\ell, z_k, u \leftarrow_{\$} \mathbb{Z}_p$, and compute ($i = 1, 2; j = 0, 1, \ldots, l-1$):

$$R_i = \left(\frac{C_i}{\hat{C}_i}\right)^c \cdot \left(\frac{X_{Ai}}{X_{Bi}}\right)^{z_i};$$

$$D_1 = \prod_{j=0}^{l-1} \left(h^{(2^{10})^j \cdot z_{t_j}}\right) \cdot C_3^c \cdot g_1^{z_1+z_2};$$

$$D_2 = \prod_{j=0}^{l-1} \left(h^{(2^{10})^j \cdot z_{t_j'}}\right) \cdot \left(\frac{\tilde{C}_3}{C_3}\right)^c \cdot \tilde{C}_1^{z_\ell} \cdot \tilde{C}_2^{z_k} \cdot g_1^{-z_1-z_2};$$

$$a_j = e(V_j, vk)^c \cdot e(V_j, g_2)^{-z_{t_j}} \cdot g_T^{z_{v_j}}, a_j' = e(V_j', vk)^c \cdot e(V_j', g_2)^{-z_{t_j'}} \cdot g_T^{z_{v_j'}};$$

$$\alpha = g_1^u;$$

- Compute $\tilde{c} = \mathsf{H}(R_1, R_2, \{V_j, V_j'\}_{j=0}^{l-1}, D_1, D_2, \{a_j, a_j'\}_{j=0}^{l-1}, \alpha)$, and let $\hat{c} = c - \tilde{c}, \hat{z} = u + \hat{c} \cdot \omega$. Then output the simulated argument:

$$\pi = \Big(R_1, R_2, \{V_j, V_j'\}_{j=0}^{l-1}, D_1, D_2, \{a_j, a_j'\}_{j=0}^{l-1}, \alpha, c, z_1, z_2, \{z_{v_j}, z_{v_j'}\}_{j=0}^{l-1}, \{z_{t_j}, z_{t_j'}\}_{j=0}^{l-1}, z_\ell, z_k, \hat{z}\Big).$$

Fig. 3: Simulator

Parse the argument into 3 parts:

$$\pi = \big(a = (R_1, R_2, \{V_j, V_j'\}_{j=0}^{l-1}, D_1, D_2, \{a_j, a_j'\}_{j=0}^{l-1}, \alpha), c, z = (z_1, z_2, \{z_{v_j}, z_{v_j'}\}_{j=0}^{l-1}, \{z_{t_j}, z_{t_j'}\}_{j=0}^{l-1}, z_\ell, z_k, \hat{z})\big).$$

For the sake of clarity and convenience, we denote the simulated argument by

$$\boldsymbol{\pi} = \big(\mathfrak{a} = (\mathcal{R}_1, \mathcal{R}_2, \{\mathcal{V}_j, \mathcal{V}'_j\}_{j=0}^{l-1}, \mathcal{D}_1, \mathcal{D}_2, \{\mathfrak{a}_j, \mathfrak{a}'_j\}_{j=0}^{l-1}, \boldsymbol{\alpha}), \mathfrak{c}, \\ \mathfrak{z} = (\mathfrak{z}_1, \mathfrak{z}_2, \{\mathfrak{z}_{v_j}, \mathfrak{z}_{v'_j}\}_{j=0}^{l-1}, \{\mathfrak{z}_{t_j}, \mathfrak{z}_{t'_j}\}_{j=0}^{l-1}, \mathfrak{z}_\ell, \mathfrak{z}_k, \hat{\mathfrak{z}})\big).$$

Observe that $\hat{c} \leftarrow_{\$} \mathbb{Z}_p$ is independent of a, $c = \mathsf{H}(a) + \hat{c}$ is uniformly distributed in $\mathbb{Z}_p$, and that $\mathfrak{c}$ is also chosen from $\mathbb{Z}_p$ at random in the simulation, thus,

$$\{c\} \equiv \{\mathfrak{c}\}; \tag{6}$$

which indicates the above two distributions are identical.
Set $\mathscr{C} = \{c\} = \{\mathfrak{c}\}$. Conditioned on (6), given $\bar{c} \in \mathscr{C}$, for every $\rho \in \mathbb{Z}_p$, since $\hat{z}$, r_1, r_2, $\ell, k, q_j, m_j, s_j, w_j, v_j, v'_j \leftarrow_{\$} \mathbb{Z}_p$ where $j = 0, 1, \ldots, l-1$, and they are all independent of c, we have

$$\Pr[z_1 = \rho | c = \bar{c}] = \Pr[r_1 - cy_1 \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_2 = \rho | c = \bar{c}] = \Pr[r_2 - cy_2 \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_\ell = \rho | c = \bar{c}] = \Pr[\ell - \frac{c}{x_{A1}} \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_k = \rho | c = \bar{c}] = \Pr[k - \frac{c}{x_{A2}} \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_{v_j} = \rho | c = \bar{c}] = \Pr[q_j - c \cdot v_j \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_{v'_j} = \rho | c = \bar{c}] = \Pr[m_j - c \cdot v'_j \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_{t_j} = \rho | c = \bar{c}] = \Pr[s_j - c \cdot t_j \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[z_{t'_j} = \rho | c = \bar{c}] = \Pr[w_j - c \cdot t'_j \bmod p = \rho | c = \bar{c}] = \frac{1}{p};$$

$$\Pr[\hat{z} = \rho | c = \bar{c}] = \frac{1}{p}.$$

In the simulated argument, under the same condition, given the value $\mathfrak{z}_1, \mathfrak{z}_2, \mathfrak{z}_\ell$, $\mathfrak{z}_k, \mathfrak{z}_{v_j}, \mathfrak{z}_{v'_j}, \mathfrak{z}_{t_j}, \mathfrak{z}_{t'_j}, u \leftarrow_{\$} \mathbb{Z}_p$ which are independent of $\mathfrak{c}$, we have

$$\Pr[\mathfrak{z}_1 = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p}; \Pr[\mathfrak{z}_2 = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p};$$

$$\Pr[\mathfrak{z}_\ell = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p}; \Pr[\mathfrak{z}_k = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p};$$

$$\Pr[\mathfrak{z}_{v_j} = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p}; \Pr[\mathfrak{z}_{v'_j} = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p};$$

$$\Pr[\mathfrak{z}_{t_j} = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p}; \Pr[\mathfrak{z}_{t'_j} = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p};$$

$$\Pr[\hat{\mathfrak{z}} = \rho | \mathfrak{c} = \bar{c}] = \Pr[u + \hat{c} \cdot \omega = \rho | \mathfrak{c} = \bar{c}] = \frac{1}{p}.$$

Set $\mathscr{Z} = \{z^1, z^2, \{z_j^3, z_j^4\}_{j=0}^{l-1}, \{z_j^5, z_j^6\}_{j=0}^{l-1}, z^7, z^8, z^9 : z_i \leftarrow_{\$} \mathbb{Z}_p, i \in [10]\}$. Given $\bar{c} \leftarrow_{\$} \mathscr{C}$, for every $\bar{z} \in \mathscr{Z}$,

$$\Pr[z = \bar{z} | c = \bar{c}] = \Pr[\mathfrak{z} = \bar{z} | \mathfrak{c} = \bar{c}]. \tag{7}$$

Conditioned on (7), given $\bar{c} \in \mathscr{C}, \bar{z} \in \mathscr{Z}$, following from the verification strategy, the messages $R_1, R_2, D_1, D_2, a_j, a'_j, \alpha$ in π are determined where $j = 0, 1, \ldots, l-1$. For $\{V_j, V'_j\}$, we have

$$\Pr[V_j = \mathfrak{g} | c = \bar{c}, z = \bar{z}] = \Pr[\sigma_{t_j}^{v_j} = \mathfrak{g} | c = \bar{c}, z = \bar{z}] = \frac{1}{p};$$

$$\Pr[V'_j = \mathfrak{g} | c = \bar{c}, z = \bar{z}] = \Pr[\sigma_{t'_j}^{v'_j} = \mathfrak{g} | c = \bar{c}, z = \bar{z}] = \frac{1}{p};$$

where $\mathfrak{g} \leftarrow_{\$} G_1$, since $v_j, v'_j \leftarrow_{\$} \mathbb{Z}_p$.
Note that in the simulated argument, for fixed $\bar{c} \in \mathscr{C}, \bar{z} \in \mathscr{Z}$, the messages $\mathcal{R}_1, \mathcal{R}_2, \mathcal{D}_1, \mathcal{D}_2, \mathfrak{a}_j, \mathfrak{a}'_j, \boldsymbol{\alpha}$ are determined according to Sim. For arbitrary $\mathfrak{g} \in G_1, j = 0, 1, \ldots, l-1$,

$$\Pr[\mathcal{V}_j = \mathfrak{g} | \mathfrak{c} = \bar{c}, \mathfrak{z} = \bar{z}] = \Pr[\sigma_{t_j}^{v_j} = \mathfrak{g} | \mathfrak{c} = \bar{c}, \mathfrak{z} = \bar{z}] = \frac{1}{p};$$

$$\Pr[\mathcal{V}'_j = \mathfrak{g} | \mathfrak{c} = \bar{c}, \mathfrak{z} = \bar{z}] = \Pr[\sigma_{t'_j}^{v'_j} = \mathfrak{g} | \mathfrak{c} = \bar{c}, \mathfrak{z} = \bar{z}] = \frac{1}{p};$$

since $v_j, v'_j \leftarrow_{\$} \mathbb{Z}_p$.
Set $\mathscr{A} = \{a^1, a^2, \{a_j^3, a_j^4\}_{j=0}^{l-1}, a^5, a^6, \{a_j^7, a_j^8\}_{j=0}^{l-1}, a^9 : a^1, a^2, a^3, a^4, a^5, a_j^6, a_j^7, a^9 \leftarrow_{\$} G_1, a_j^7, a_j^8 \leftarrow_{\$} G_T\}$. Thus, given $\bar{c} \in \mathscr{C}, \bar{z} \in \mathscr{Z}$, for arbitrary $\bar{a} \in \mathscr{A}$,

$$\Pr[a = \bar{a} | c = \bar{c}, z = \bar{z}] = \Pr[\mathfrak{a} = \bar{a} | \mathfrak{c} = \bar{c}, \mathfrak{z} = \bar{z}]. \tag{8}$$

Combine (7) and (8), we conclude that for any non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\begin{aligned}&\Pr[(x, w) \leftarrow \mathcal{A}_1(1^n), (a, c, \hat{c}, z) \leftarrow \mathsf{P}(x, w, PP) : (x, w) \in R \wedge \mathcal{A}_2(a, c, \hat{c}, z) = 1] \\ &= \Pr[(x, w) \leftarrow \mathcal{A}_1(1^n), (a, c, \hat{c}, z) \leftarrow \mathsf{Sim}(x) : (x, w) \in R \wedge \mathcal{A}_2(a, c, \hat{c}, z) = 1].\end{aligned}$$

(Perfect) Zero-knowledge property is obtained.

3.3 An optimized verifier.

Instead of verifying equation (4) with computing $4l$ pairing computations, V can select randomly $d_0, d'_0, d_1, d'_1, \ldots, d_{l-1}, d'_{l-1} \leftarrow_{\$} \mathbb{Z}_p$, and check whether the following equation holds:

$$\begin{aligned} a_0^{d_0} a_1^{d_1} \cdots a_{l-1}^{d_{l-1}} (a'_0)^{d'_0} (a'_1)^{d'_1} \cdots (a'_{l-1})^{d'_{l-1}} = \\ e(V_0^{cd_0} V_1^{cd_1} \cdots V_{l-1}^{cd_{l-1}} (V'_0)^{cd'_0} (V'_1)^{cd'_1} \cdots (V'_{l-1})^{cd'_{l-1}}, vk) \cdot \\ e(V_0^{-z_{t_0} d_0} V_1^{-z_{t_1} d_1} \cdots V_{l-1}^{-z_{t_{l-1}} d_{l-1}} (V'_0)^{-z_{t'_0} d'_0} (V'_1)^{-z_{t'_1} d'_1} (V'_{l-1})^{-z_{t'_{l-1}} d'_{l-1}}, g_2) \cdot \\ g_T^{z_{v_0} d_0 + z_{v_1} d_1 + \cdots + z_{v_{l-1}} d_{l-1} + z_{v'_0} d'_0 + z_{v'_1} d'_1 + \cdots + z_{v'_{l-1}} d'_{l-1}}. \end{aligned} \tag{9}$$

Equation (9) only computes 2 pairing computations, which is more efficient than (4), but induces computational completeness property. Next we discuss the equivalence of this two equations.

- (4) $\Rightarrow$ (9): Upon substitution of all the values of $\{a_j\}_{j=0}^{l-1}, \{a'_j\}_{j=0}^{l-1}$ in (4), equation (9) is obtained.
- (9) $\Rightarrow$ (4): Consider equation (9):

$$\begin{aligned} & \mathit{Right\_Side} \\ =& \prod_{j=0}^{l-1} \left(e(V_j^{cd_j}, vk) \cdot e((V'_j)^{cd'_j}, vk) \cdot e(V_j^{-z_j d_j}, g_2) \cdot e((V'_j)^{-z_{t'_j} d'_j}, g_2) \cdot g_T^{z_{v_j} d_j} \cdot g_T^{z_{v'_j} d'_j} \right) \\ =& \prod_{j=0}^{l-1} \left(e(V_j, vk)^{cd_j} \cdot e(V'_j, vk)^{cd'_j} \cdot e(V_j, g_2)^{-z_{t_j} d_j} \cdot e(V'_j, g_2)^{-z_{t'_j} d'_j} \cdot g_T^{z_{v_j} d_j} \cdot g_T^{z_{v'_j} d'_j} \right) \\ =& \prod_{j=0}^{l-1} \left(\left(e(V_j, vk)^c \cdot e(V_j, g_2)^{-z_{t_j}} \cdot g_T^{z_{v_j}} \right)^{d_j} \cdot \left(e(V'_j, vk)^c \cdot e(V'_j, g_2)^{-z_{t'_j}} \cdot g_T^{z_{v'_j}} \right)^{d'_j} \right); \end{aligned}$$

$$\mathit{Left\_Side} = \prod_{j=0}^{l-1} \left((a_j)^{d_j} (a'_j)^{d'_j} \right).$$

if $\mathit{Left\_Side} = \mathit{Right\_Side}$, two cases occur:

1. $\forall j = 0, 1, \ldots, l-1, a_j = e(V_j, vk)^c \cdot e(V_j, g_2)^{-z_{t_j}} \cdot g_T^{z_{v_j}}, a'_j = e(V'_j, vk)^c \cdot e(V'_j, g_2)^{-z_{t'_j}} \cdot g_T^{z_{v'_j}}$, which implies the correctness of (4).
2. There exist some d_j or $d'_j = 0$, which can lead to $a_j \neq e(V_j, vk)^c \cdot e(V_j, g_2)^{-z_{t_j}} \cdot g_T^{z_{v_j}}$ or $a'_j \neq e(V'_j, vk)^c \cdot e(V'_j, g_2)^{-z_{t'_j}} \cdot g_T^{z_{v'_j}}$ for some $j \in [0, l)$. This case happens with probability

$$\begin{aligned} & \sum_{i=1}^{2l} \left(C_{2l}^i \frac{1}{p^i} (1 - \frac{1}{p})^{2l-i} \right) \\ =& 1 - (1 - \frac{1}{p})^{2l} < \frac{2l}{p} \leq \frac{2l}{2^{n-1}}; \end{aligned}$$

which is a negilible probability, since p is a prime with n bits.

Overall, with an overwhelming probability $1 - \frac{2l}{2^{n-1}}$, equation (4) $\Leftrightarrow$ (9).

4 Evaluation

We evaluated our NIZK argument system on a personal computer . In order to show the superriorty of our scheme intuitively, we also took a comparison with prior works.

4.1 Comparison

Let us discuss our protocol and compare it with other existing solutions in both theoretical and practical aspects. Firstly, we focus on the computational complexity in theorical aspects. The system parameter PP generated once for the proof is of the size $||G_2|| + 2^{10} \cdot (||G_1|| + ||G_T||)$ (omited the bilinear group parameters), while the size of the whole proof is $(2l+5) \cdot ||G_1|| + 2l \cdot ||G_T|| + (4l+6) \cdot ||\mathbb{Z}_p||$. Secondly, in the practical performance, we implement our protocol utilizing the MIRACL Library (more precise, miracl 7.0 version), consider the plaintext space $[0, 2^{30})$ and take SHA256 hash function to instantiate our NIZK argument. The experiment is based on coding language C++ on Windows system (Windows 7, 64 bits) with an Inter(R) Core(TM) i7-4770 CPU of 3.40 GHz and 16-GB RAM. We now give a comparison in Table 1 between our scheme and the zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) [BSCTV13] employed by Zerocash [BCG+14]. Additionally, in the public parameter size, we omitted the basic parameters of ECC including e, p, g_1, g_2, g_t, h which only account for a small proportion. Designed for the cloud/verifiable computing, the zk-SNARK protocol owns significant efficiency in the verify process and proof size, but it does not have the according efficiency in the running time of the prove process. On one hand, our protocol has improved a lot, e.g., the running time of Setup (corresponding to the KeyGen phase in Pinocchio) and Proof are improved about 35.7x and amazing 1830x respectively, to get a trade-off between the prover and verifier obtaining two fairly fast algorithms. This result also gives us confidence on applying our scheme in the computation-limited devices like mobile-phones. On the other hand, the size of public parameter is improved 7.8x. Although we don't a get better result in the Verify phase and the size of a proof, since the absolute verifier's running time and proof size are indeed small, it is sufficient to construct a direct and efficient NIZK argument for our DSC scheme.

[2] Parameters $(\kappa, \iota, \beta, \delta)$, polynomials of the security parameters n, are components of a circuit $C : \mathbb{F}^{\beta} \times \mathbb{F}^{\gamma} \to \mathbb{F}^{\delta}$ with κ wires and ι gates. We refer the reader to [BSCTV13] for more information.

		zk-SNARK	This Paper	Improvement
Theoretical	Public Parameter Size	$(6\kappa + 2\beta + \iota + \delta + 29)\|\|G_1\\| + (\iota + 9)\|\|G_2\\|\|^2$	$2^{10}(\|\|G_1\|\| + \|\|G_T\|\|) + \|\|G_2\|\|$	↑
	Proof Size	$7\|\|G_1\|\| + \|\|G_2\|\|$	$(2l + 5)\|\|G_1\|\| + 2l\|\|G_T\|\| + (4l + 6)\|\|\mathbb{Z}_p\|\|$	↓
Practical	Setup	5 min 11s	8.7s	35.7x
	Proof	1 min 59s	64.97ms	1830x
	Verify	5.4ms	48.96ms	0.1x
	Public Parameter Size	3.4MB	0.44MB	7.8x
	Proof Size	288B	3616B	0.1x

Table 1: Comparison with Pinocchio

5 Conclusion

In this paper, We present a main contribution: a decentralized smart contract system with balance and transaction amount hiding under the ACCOUNT architecture. To implement this mechanism enabling programmability, we put forward a homomorphic encryption scheme with the form like Pedersen commitment and construct a contrete NIZK scheme to prove the validity of transactions. In our NIZK argument system, the public parameter serves as the common reference string which is only generated once for multi proofs. With respect to the security, we can achieve the zero-knowledge property in the standard CRS model, while the soundness can be obtained under the RO model. We also demonstrate the practical performence of our NIZK scheme on a personal computer. The result gives our confidence in applying our scheme in practice.

The NIZK scheme employed a range proof. There has been a lot of research on the range proof so far such as [Sce09,CCJT13,CC+08,CLS10,CLZ12,BBB+17]. A future direction is to utilize a new range proof to obtain more efficiency without lose security. In the range proof, we utilize the weak Boneh-Boyen signature scheme. It is also a way to develop our scheme to use alternative signature schemes in the range proof.

Acknowledgements

Shunli Ma and Yi Deng are supported by the National Natural Science Foundation of China (Grant No. 61772521), Key Research Program of Frontier Sciences, CAS (QYZDB-SSW-SYS035), and the Open Project Program of the State Key Laboratory of Cryptology. Debiao He is supported by the National Key Research and Development Program of China (Grant No. 2017YFB0802500). Jiang Zhang is supported by the National Key Research and Development Program of China (Grant No. 2017YFB0802005), the National Natural Science Founda-

tion of China (Grant Nos. 61602046, 61602045, U1536205), and the Young Elite Scientists Sponsorship Program by CAST (2016QNRC001).

Reference

BB04. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Eurocrypt*, volume 3027, pages 56–73. Springer, 2004.

BBB+17. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Efficient range proofs for confidential transactions. *IACR Cryptology ePrint Archive*, 2017:1066, 2017.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Crypto*, volume 3152, pages 41–55. Springer, 2004.

BCG+14. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 459–474, 2014.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM, 1988.

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *Advances in Cryptology–ASICACRYPT 2001*, pages 514–532. Springer, 2001.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

BSCG+13. Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. Snarks for c: Verifying program executions succinctly and in zero knowledge. In *Advances in Cryptology–CRYPTO 2013*, pages 90–108. Springer, 2013.

BSCTV13. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive arguments for a von neumann architecture. *IACR Cryptology ePrint Archive*, 2013:879, 2013.

CC+08. Jan Camenisch, Rafik Chaabouni, et al. Efficient protocols for set membership and range proofs. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 234–252. Springer, 2008.

CCJT13. Sébastien Canard, Iwen Coisel, Amandine Jambert, and Jacques Traoré. New results for the practical use of range proofs. In *European Public Key Infrastructure Workshop*, pages 47–64. Springer, 2013.

CLS10. Rafik Chaabouni, Helger Lipmaa, and Abhi Shelat. Additive combinatorics and discrete logarithm based range protocols. In *ACISP*, volume 10, pages 336–351. Springer, 2010.

CLZ12. Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A non-interactive range proof with constant communication. In *Financial Cryptography*, volume 2012, pages 179–199. Springer, 2012.

Cra96. Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, 1996.

Dam10. Ivan Damgrd. On sigma protocols. http://www.cs.au.dk/~ivan/Sigma.pdf, 2010.

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *Eurocrypt*, volume 6110, pages 44–61. Springer, 2010.

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

GM82. Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377. ACM, 1982.

HL10. Carmit Hazay and Yehuda Lindell. *Efficient secure two-party protocols: Techniques and constructions*. Springer Science & Business Media, 2010.

KMH+17. Henning Kopp, David Mödinger, Franz Hauck, Frank Kargl, and Christoph Bösch. Design of a privacy-preserving decentralized file storage with financial incentives. In *2017 IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2017, Paris, France, April 26-28, 2017*, pages 14–22, 2017.

KMS+16. Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *Security and Privacy*, pages 839–858, 2016.

LCC+17. Ao Lei, Haitham Cruickshank, Yue Cao, Philip Asuquo, Chibueze P. Anyigor Ogah, and Zhili Sun. Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. *IEEE Internet of Things Journal*, PP(99):1–1, 2017.

LST+17. Xueping Liang, Sachin Shetty, Deepak K. Tosh, Charles A. Kamhoua, Kevin A. Kwiat, and Laurent Njilla. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017, Madrid, Spain, May 14-17, 2017*, pages 468–477, 2017.

LWW04. Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *ACISP*, volume 4, pages 325–335. Springer, 2004.

Max15. Gregory Maxwell. Confidential transactions. *URL: https://people. xiph. org/˜ greg/confidential_values. txt (Accessed 09/05/2016)*, 2015.

Nak08. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 2008.

NM+16. Shen Noether, Adam Mackenzie, et al. Ring confidential transactions. *Ledger*, 1:1–18, 2016.

OHHH17. Yaghoob Omran, Michael Henke, Roger Heines, and Erik Hofmann. Blockchain-driven supply chain finance: Towards a conceptual framework from a buyer perspective. https://www.alexandria.unisg.ch/251095/, 2017. [Online].

P+91. Torben P Pedersen et al. Non-interactive and information-theoretic secure verifiable secret sharing. In *Crypto*, volume 91, pages 129–140. Springer, 1991.

Sab13. N Van Saberhagen. Cryptonote v 2.0. https://bytecoin.org/downloads/whitepaper.pdf, 2013. [Online].

Sce09. Antoine Scemama. *A Cryptanalysis of the 2R Cryptosystem and an Improved Commitment Range Proof*. PhD thesis, 2009.

TS10. Naoki Tanaka and Taiichi Saito. On the q-strong diffie-hellman problem. *IACR Cryptology ePrint Archive*, 2010:215, 2010.

Woo14. Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.

XSA+17. Qi Xia, Emmanuel Boateng Sifah, Kwame Omono Asamoah, Jianbin Gao, Xiaojiang Du, and Mohsen Guizani. Medshare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access*, 5:14757–14767, 2017.

XSC+17. Lei Xu, Nolan Shah, Lin Chen, Nour Diallo, Zhimin Gao, Yang Lu, and Weidong Shi. Enabling the sharing economy: Privacy respecting contract based on public blockchain. In *ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 15–21, 2017.

Appendix: The Completeness of Our Protocol

Our Protocol has the perfect completeness property. It is trivial to check the correctness of equation (5).

1. The correctness of equation (1):

$$\left(\frac{C_i}{\tilde{C}_i}\right)^c \cdot \left(\frac{X_{Ai}}{X_{Bi}}\right)^{z_i} = \left(\frac{X_{Ai}}{X_{Bi}}\right)^{cy_i+z_i} = \left(\frac{X_{Ai}}{X_{Bi}}\right)^{r_i} = R_i$$

2. The correctness of equation (2):

$$\begin{aligned}
&\prod_{j=0}^{2}\left(h^{(2^{10})^j\cdot z_{t_j}}\right)\cdot C_3^c\cdot g_1^{z_1+z_2}\\
=&\prod_{j=0}^{2}\left(h^{(2^{10})^j\cdot z_{t_j}}\right)\cdot\left(g_1^{y_1+y_2}\cdot h^t\right)^c\cdot g_1^{z_1+z_2}\\
=&\prod_{j=0}^{2}\left(h^{(2^{10})^j\cdot z_{t_j}}\right)\cdot h^{te}\cdot g_1^{cy_1+cy_2+z_1+z_2}\\
=&h^{\sum_{j=0}^{2}(2^{10})^j\cdot z_{t_j}}\cdot h^{te}\cdot g_1^{cy_1+z_1+cy_2+z_2}\\
=&h^{\sum_{j=0}^{2}(2^{10})^j\cdot(z_{t_j}+et_j)}\cdot g_1^{(cy_1+z_1)+(cy_2+z_2)}\\
=&h^{\sum_{j=0}^{2}(2^{10})^j\cdot s_j}\cdot g_1^{r_1+r_2}\\
=&\prod_{j=0}^{2}\left(2^{10})^j\cdot s_j\right)\cdot g_1^{r_1+r_2}=D_1
\end{aligned}$$

3. The correctness of equation (3):

$$\begin{aligned}
&\prod_{j=0}^{2}\left(h^{(2^{10})^j\cdot z_{t'_j}}\right)\cdot\left(\frac{\tilde{C}_3}{C_3}\right)^c\cdot\tilde{C}_1^{z_\ell}\cdot\tilde{C}_2^{z_k}\cdot g_1^{-z_1-z_2}\\
=&\prod_{j=0}^{2}\left(h^{(2^{10})^j\cdot z_{t'_j}}\right)\cdot\left(\tilde{C}_1^{\frac{1}{x_{A1}}}\cdot\tilde{C}_2^{\frac{1}{x_{A2}}}\cdot g_1^{-y_1-y_2}\cdot h^{t_A-t}\right)^c\cdot\\
&\tilde{C}_1^{\ell-\frac{c}{x_{A1}}}\cdot\tilde{C}_2^{k-\frac{c}{x_{A2}}}\cdot g_1^{-z_1-z_2}\\
=&h^{\sum_{j=0}^{2}(2^{10})^j\cdot z_{t'_j}}\cdot\tilde{C}_1^{\ell}\cdot\tilde{C}_2^{k}\cdot g_1^{-r_1-r_2}=D_2
\end{aligned}$$

4. The correctness of equation (4), for the sake of simplicity we only consider the cae of a_j :

$$\begin{aligned}
&e(V_j,vk)^c\cdot e(V_j,g_2)^{-z_{t_j}}\cdot g_T^{z_{v_j}}\\
=&e(\sigma_{t_j},g_2^{\lambda})^{c\cdot v_j}\cdot e(\sigma_{t_j},g_2)^{-z_{t_j}\cdot v_j}\cdot e(g_1,g_2)^{z_{v_j}}\\
=&e(\sigma_{t_j},g_2)^{cv_j\cdot\lambda}\cdot e(\sigma_{t_j},g_2)^{(ct_j-s_j)\cdot v_j}\cdot e(g_1,g_2)^{q_j-cv_j}\\
=&e(\sigma_{t_j},g_2)^{cv_j\cdot\lambda+cv_j\cdot t_j-s_j\cdot v_j}\cdot e(g_1,g_2)^{q_j-cv_j}\\
=&e(\sigma_{t_j},g_2)^{-s_j\cdot v_j}\cdot e(g_1,g_2)^{q_j}\cdot e(\sigma_{t_j},g_2)^{cv_j\cdot(\lambda+t_j)}\cdot e(g_1,g_2)^{-cv_j}\\
=&a_j\cdot e(g_1,g_2)^{\frac{1}{\lambda+t_j}\cdot cv_j\cdot(\lambda+t_j)}\cdot e(g_1,g_2)^{-cv_j}\\
=&a_j
\end{aligned}$$