INFORMATICA, 2024, Vol. 35, No. 3, 601–616

DOI: https://doi.org/10.15388/24-INFOR564

Confidential Transaction Balance Verification by the Net Using Non-Interactive Zero-Knowledge Proofs

Aušrys KILČIAUSKAS[1], Antanas BENDORAITIS[2], Eligijus SAKALAUSKAS[1,*]

[1] *Department of Applied Mathematics, Kaunas University of Technology, Studentu 50-324A, Kaunas LT51368, Lithuania*
[2] *Faculty of Informatics, Kaunas University of Technology, Lithuania*
e-mail: ausrys.kilciauskas@ktu.lt, antanas.bendoraitis@ktu.edu, eligijus.sakalauskas@ktu.lt

Received: March 2024; accepted: June 2024

Abstract. One of the main trends for the monitoring and control of business processes is to implement these processes via private blockchain systems. These systems must ensure data privacy and verifiability for the entire network here denoted by 'Net'. In addition, every business activity should be declared to a trusted third party (TTP), such as an Audit Authority (AA), for tax declaration and collection purposes.

We present a solution for a confidential and verifiable realization of transactions based on the Unspent Transaction Output (UTxO) paradigm. This means that the total sum of transaction inputs (incomes) In must be equal to the total sum of transaction outputs (expenses) Ex, satisfying the balance equation $In = Ex$. Privacy in a private blockchain must be achieved through the encryption of actual transaction values. However, it is crucial that all participants in the network be able to verify the validity of the transaction balance equation. This poses a challenge with probabilistically encrypted data. Moreover, the inputs and outputs are encrypted with different public keys. With the introduction of the AA, the number of different public keys for encryption can be reduced to two. Incomes are encrypted with the Receiver's public key and expenses with the AA's public key.

The novelty of our realization lies in taking additively-multiplicative, homomorphic ElGamal encryption and integrating it with a proposed paradigm of modified Schnorr identification providing a non-interactive zero-knowledge proof (NIZKP) using a cryptographically secure h-function. Introducing the AA as a structural element in a blockchain system based on the UTxO enables effective verification of encrypted transaction data for the Net. This is possible because the proposed NIZKP is able to prove the equivalency of two ciphertexts encrypted with two different public keys and different actors.

This integration allows all users on the Net to check the UTxO-based transaction balance equation on encrypted data. The security considerations of the proposed solution are presented.

Key words: blockchain, transactions, unspent transaction output, confidentiality, verifiability.

*Corresponding author.

1. Introduction

The global trends, current tendencies, and frontiers of blockchain technology have been reported in Boakye *et al.* (2022) where it is noted that blockchain technology research in finance has become dominated by studies on crowdfunding, entrepreneurial finance, bitcoin, entrepreneurship, fintech, and venture capital. Their overview covers 157 articles. However, the interaction between financial activity and the tax collection system is not sufficiently outlined in this and other studies.

This paper is a continuation of our previous research based on a tax declaration scheme using blockchain confidential transactions (Sakalauskas *et al.*, 2023), based on the Unspent Transaction Output (UTxO) paradigm. This scheme includes the main blockchain actors: Senders, the Receiver, the Audit Authority (AA), and the Net, and provides confidentiality of transactions while at the same time ensuring their verifiability for the Net. Any sum Received by the Receiver is denoted by income i, and any sum spent by the Receiver is denoted by expense e. The honesty of a transaction is based on the balance between the total sums of income and expense, which we denote by In and Ex. This means that, to ensure the honesty of transaction, the balance equation $In = Ex$ must hold. Privacy in a private blockchain must be achieved through the encryption of actual transaction values. But at the same time, all the Net must be able to verify the validity of the transaction balance equation, which is impossible for probabilistically encrypted data.

So far, ciphertext equivalency proofs have been broadly used in cloud computing.

Guomin *et al.* (2010) present a probabilistic public key encryption scheme based on a bilinear group where anyone can verify whether two ciphertexts are encryptions of the same message. The applications of this include searchable encryption and the partitioning of encrypted data. In their scheme, verifying the equivalency of ciphertexts requires bilinear map operations. Such operations require more computational power than exponential operations in ElGamal encryption. Their presented solution does not support encryptor authorization, which is an important part of our application.

The issue of searching among encrypted data is discussed in Canard *et al.* (2012) with an approach based on the ElGamal system, using bilinear maps. This approach extends public-key encryption by having the following functionality: given a plaintext, a ciphertext, and a public key, it is universally possible to check whether the ciphertext encrypts the plaintext under the key. This approach, like the previous one, lacks authorization. This capacity could be valuable when storing encrypted transaction balances, and utilizing the technique outlined in our paper, in the cloud.

In Hongbo *et al.* (2019), the problem of protecting data privacy using cloud storage is considered. The most effective solution is to encrypt data before uploading it to the cloud. The authors introduced a new notion of identity-based encryption with an equivalency test which supported flexible authorization using bilinear pairings. The experimental results presented by the authors show that their scheme is efficient and can satisfy various types of searches of encrypted data.

The problem of plaintext equality, which consists in determining whether ciphertexts hold the same value, is considered in Blazy *et al.* (2021). Their approach generates two

ciphertexts using a probabilistic public-key encryption scheme by the same prover. For a proof of plaintext equality, the authors proposed sigma protocols that led to non-interactive zero-knowledge proofs.

In Zhao *et al.* (2022), a public key encryption scheme is proposed which supports authorized equality tests on ciphertexts in the dual server model. In this scheme, the primary server and secondary server must get authorization from users before performing a sequential equality test on ciphertexts. This scheme provides security against keyword guessing attacks and is a further improvement on the schemes proposed before.

In Dong *et al.* (2023), it is pointed out that existing encryption schemes can potentially suffer from information leakage when dealing with multiple ciphertexts due to the need for pairwise equality tests. In this paper, encryption is supported by a multi-ciphertext equality test with proxy-assisted authorization. The proposed scheme incorporates the functionality of the multi-ciphertext equality test into the encryption schemes outlined above. It allows a single equality test to be performed on multiple ciphertexts to determine whether the underlying plaintexts are equal.

In the reviewed papers, different techniques are considered, but the proposed solutions do not cover the problem we are examining in this paper. Ensuring privacy in the realization of transactions requires that the inputs and outputs of such transactions are encrypted with different public keys, and that these encryptions are performed by different actors. All Senders encrypt their transactions with the Receiver's public key, and the Receiver declares her expenses to the AA by encrypting them with the AA's public key. To validate the balance, additively-multiplicative homomorphic encryption must be integrated into the system. We are using the well-known probabilistic ElGamal encryption paradigm, transformed to become additively-homomorphic, using also an established technique. This issue has not been considered in our reviewed literature.

As outlined in the literature, the problem of deciding whether several ciphertexts are computed from the same plaintext is called the ciphertext equivalency problem. Thus, the integrity of the transaction can be verified by providing a ciphertext equivalency proof for the total *In* and *Ex* values, which are encrypted. In the case of an honest transaction, $In = Ex$, and the Receiver must prove that multiple ciphertexts of the (encrypted) total *In* and *Ex* values are the same. This eventuality is also not considered in the outlined literature, since it arises when encrypted *In* and *Ex* values are made by different actors and with different public keys.

In the reviewed techniques, no homomorphic property is required for balance verification, is realized. Such techniques can be used in a further step to store confidential data in the cloud. It is also known that encryption based on the bilinear pairing approach requires more computation power than the ElGamal encryption-based approach. Given the large number of transactions in the blockchain, optimizing the effectiveness of the ciphertext equivalency proof on the user side is desirable.

In this paper, we propose a more efficient method for the verification of balances by the Net in confidential transactions, compared to our previous publication (Sakalauskas *et al.*, 2023). The background of our approach is the application of modified, probabilistic, additively-multiplicative homomorphic ElGamal encryption together with a modified

Schnorr identification method (Freeman, 2011; Boneh and Shoup, 2023). The benefit of this approach is that the public parameters for both cryptographic methods are the same. The modification of ElGamal encryption involves transforming it into an additively-multiplicative homomorphic encryption, as proposed by Bunz *et al.* (2018).

The innovation in our realization is based on the integration of modified, probabilistic and additively-multiplicative homomorphic ElGamal encryption with our proposed construction of a modified Schnorr identification. This construction is extended to a non-interactive zero-knowledge proof (NIZKP) using a cryptographically secure h-function (Boneh and Shoup, 2023). By introducing an AA as a structural element in blockchain system based on UTxO, the verification of encrypted transaction data for the Net can be dealt with effectively.

This is achieved when the proposed NIZKP construction proves the equivalency of two ciphertexts encrypted with two different public keys. This equivalency shows that two ciphertexts correspond to encryption of the same plaintext, either In or Ex when $In = Ex$.

Efficiency is assured by using NIZKP, thereby reducing twofold the number of encryptions required for Senders and the number of decryptions realized by the Receiver, compared with our previous publication (Sakalauskas *et al.*, 2023). This is achieved by omitting the encryption of random parameters sent by Senders to the Receiver and, hence, avoiding the decryption of these parameters for the Receiver. For example, if a transaction has M inputs, then the validity verification requires M encryptions and M decryptions, respectively. All encryption and decryption methods require at least two modular exponentiations. In this paper, we are using NIZKP which requires only four modular exponentiations instead of $2M$ encryptions/decryptions of random parameters.

The approach presented here is compatible with mobile e-wallets, and has the capacity to realize transactions offline. The realization of e-wallet transactions in the presence of observers is treated in Sakalauskas *et al.* (2017, 2018), and Muleravičius *et al.* (2019). In this case, a digital currency for money transfers can be installed in the customer's e-wallet.

The main contributions of this paper are:

- A scheme for verifying the balance of confidential transactions for the Net is presented based on the UTxO paradigm.
- The additively-multiplicative probabilistic ElGamal encryption scheme is integrated with our proposed modification of a Non-Interactive Zero Knowledge Proof (NIZKP).
- NIZKP integration allows for a reduction in the number of encryption and decryption operations versus the previous scheme proposed by the authors.
- The security considerations are presented.

The innovation of our realization is based on the integration of probabilistic, additively-multiplicative homomorphic ElGamal encryption with our proposed modification of a Non-Interactive Zero-Knowledge Proof (NIZKP) based on Schnorr identification, and on the introduction of an AA as a structural element in a UTxO-based blockchain, thus facilitating a more effective verification of encrypted transaction data by the Net. This integration allows all users on the Net to check the balance equation for UTxO-based transactions from encrypted data.

In Section 2, an overall description of the proposed scheme is presented. In Section 3, an introduction to ElGamal encryption is given, and the construction of additively-multiplicative homomorphic encryption is presented. The construction of a confidential transaction is presented in Sections 4 and 5 using our proposed modification of Schnorr identification. In Section 6, security and efficiency analyses are provided. Section 7 gives conclusions, and at the end, a list of references is presented.

2. Overall Description of the Transaction Scheme

To be self-contained, we present here some material from Sakalauskas *et al.* (2023). We consider multiple-input and multiple-output blockchain transactions based on the UTxO paradigm, which is the fundamental building block of cryptocurrency transactions in blockchain systems. Transactions based on a UTxO have certain inputs and outputs (Pinna *et al.*, 2018).

Our scheme defines the following actors: the transaction creator, Alice; the Audit Authority (AA); and the network of users, denoted as the Net. The Net is divided into three parts: the Bobs, $B_1, B_2, \ldots, B_M$ transferring money to Alice and providing her with income; the Larries $L_1, L_2, \ldots, L_N$ receiving money from Alice and thus representing Alice's expenses; and other Net nodes verifying the transaction's validity, composing and validating blocks, etc.

The present solution is an integration of several approaches. It is known that the trustworthiness of transactions relies on the balance between income (inputs) and expenses (outputs). Bunz *et al.* (2018) present a method to assure the Net of the confidentiality and verifiability of transactions using a modification of ElGamal encryption (ElGamal, 1985). This technique transforms multiplicatively homomorphic ElGamal encryption into additively homomorphic encryption. We will use this technique to create confidential and verifiable transactions for the Net. According to the UTxO paradigm, a valid transaction requires that the sum of all inputs be equal to the sum of all outputs. Change leftover after expenses is sent to the transaction creator as one of the outputs.

For business processes, it is important to ensure the confidentiality of transaction amounts. Confidentiality means that transaction data must be encrypted using a secure probabilistic encryption method. The challenge here lies in ensuring the honesty of transactions with encrypted incomes and expenses. Even when these values are equal, their ciphertexts may differ due to probabilistic encryption. As a result, detecting balance violations becomes problematic. Let us consider a transaction, created by Alice, consisting of income and expenses. Let us assume that Alice received incomes $i_1, i_2, \ldots, i_M$ from several Bobs $B_1, B_2, \ldots,$ and B_M, respectively. Then the total income is $i = i_1 + i_2 + \cdots + i_M$. Alice transfers part of her total income i to several Larries $L_1, L_2, \ldots, L_N$ by disbursing expenses $e_1, e_2, \ldots, e_N$, respectively. If the sum of expenses $e' = e_1 + e_2 + \cdots + e_N$ is less than the total income, then she transfers the change value, which we denote by e_{N+1}, to herself. If the transaction is honest, then the following balance equation must hold

$$i = i_1 + i_2 + \cdots + i_M = e_1 + e_2 + \cdots + e_N + e_{N+1} = e' + e_{N+1} = e. \tag{1}$$

In an open blockchain, i.e. in an open distributed ledger, all nodes in the Net can verify the balance of the transaction. If this balance holds good, the transaction is assumed valid. However, in private blockchain transactions, data should be confidential. In these cases, the Net cannot directly verify the validity of a transaction.

In this study, we will use the following notation for the private key PrK and public key PuK of two actors. For Alice: $\mathrm{PrK_A} = x$, $\mathrm{PuK_A} = a$ and for the AA: $\mathrm{PrK_{AA}} = z$, $\mathrm{Puk_{AA}} = \beta$.

We assume that the AA is a Trusted Third Party (TTP) for all the Net providing tax accountancy services. All actors must declare to the AA all actual transaction data by encrypting it with the AA's $\mathrm{Puk}_{AA} = \beta$ to ensure the confidentiality of their business activity. In addition, in our example, it means that all Bobs must encrypt their expenses by Alice's $\mathrm{PuK_A} = a$. Then only Alice can decrypt her received income and verify their correctness. When Alice is transferring her expenses to her Larries, and the leftover change to herself, she encrypts them with corresponding Larries' public keys. We do not consider this stage in this paper.

We will deal with Alice's encrypted income values i_m, $m = 1, 2, \ldots, M$, encrypted by her $\mathrm{PuK_A} = a$ and ciphertexts $c_{a,im}$, respectively. All Alice's expenses e_n, $n = 1, 2, \ldots, N$ are declared to the AA by encrypting them with $\mathrm{Puk_{AA}} = \beta$ represented by ciphertexts $c_{\beta,en}$, respectively.

3. Additively-Multiplicative Homomorphic ElGamal Encryption

Our solution relies on probabilistic asymmetric ElGamal encryption by transforming it into additively-multiplicative homomorphic encryption (Bunz *et al.*, 2018). To realize ElGamal encryption/decryption, public parameters must be shared on the Net. Probabilistic encryption is performed by the Sender using the Receiver's public key and a randomly generated number, thus providing different ciphertexts even for the encryption of the same plaintext. The ciphertext is sent to the Receiver who can decrypt it using the same shared public parameters and his/her private key to obtain the corresponding plaintext. ElGamal encryption has so-called multiplicatively homomorphic properties: the encrypted product of plaintexts is equal to the product of corresponding ciphertexts.

Let $Z_p^* = \{1, 2, 3, \ldots, p-1\}$ be a multiplicative cyclic group of order $p-1$, where p is prime, and multiplication is performed mod p. Then let there be a cyclic subgroup G_q of order q where q is prime and any generator of this group we denote by g. Let *Mes* be a message to be encrypted and m – the image of a reversible 1-to-1 function, transforming *Mes* to m in Z_p^*. We denote public parameters in ElGamal encryption by

$$PP = (p, g). \tag{2}$$

The ElGamal encryption system uses the discrete exponential function (DEF) defined by the generator g in G_q and provides the following isomorphic mapping DEF_g: $Z_q \rightarrow G_q$, where $Z_q = \{0, 1, 2, \ldots, q-1\}$ is a ring with addition, subtraction, and multiplication

operations mod q. For any integer $i \in Zq$:

$$DEF_g(i) = g^i \bmod p. \tag{3}$$

Further, we omit the notation mod p, except in some special cases. A private-public key pair (PrK, PuK) is computed using public parameters PP in (2) and *DEF*. Encryption is performed using the Receiver's PuK, and decryption, correspondingly, with the Receiver's PrK. Let Alice's private-public key pair be ($\mathrm{PrK_A} = x$, $\mathrm{PuK} = a$) and the AA's private-public key pair be ($\mathrm{PrK_{AA}} = z$, $\mathrm{PuK} = \beta$). Key generation for both actors, Alice and the AA, is performed in the following way:

1. $\mathrm{PrK_A} = x$ and $\mathrm{PrK_{AA}} = z$ are randomly generated integers in the set Z_q:

$$x \leftarrow randi(Z_q); \qquad z \leftarrow randi(Z_q). \tag{4}$$

2. $\mathrm{PuK_A} = a$ and $\mathrm{PuK_{AA}} = \beta$ are computed using *DEF*:

$$a = g^x \bmod p; \qquad \beta = g^z \bmod p. \tag{5}$$

Let Alice be a Sender and encrypt transaction data d corresponding to a single income or expense with the AA's $\mathrm{PuK_{AA}} = \beta$. Then ciphertext $c_\beta = (\varepsilon_\beta, \delta_\beta)$ is obtained by the following two steps:

1. Generate a random integer $l \in Z_q$.
2. Compute two components ε_β and δ_β of the ciphertext c_β

$$c_\beta = (\varepsilon_\beta, \delta_\beta) = \left(d\beta^l, g^l\right). \tag{6}$$

Decryption is performed using the Receiver's $\mathrm{PrK_{AA}} = z$

$$d = \varepsilon_\beta \cdot (\delta_\beta)^{-z}. \tag{7}$$

We denote encryption and decryption functions by *Enc*( ) and *Dec*( ), respectively. Then, formally, encryption and decryption operations are expressed in the following way:

$$Enc(\beta, l, d) = c_\beta; \qquad Dec(z, c_\beta) = d. \tag{8}$$

ElGamal encryption has the following multiplicative isomorphic property. Let $d_1, d_2 \in Z_q$. Then, for the encryption of two plaintexts d_1 and d_2, two random numbers k, l are generated, yielding two ciphertexts $c_{\beta,1}$ and $c_{\beta,2}$:

$$c_{\beta,1} = Enc(\beta, k, d_1) = (\varepsilon_{\beta,1}, \delta_{\beta,1}); \qquad c_{\beta,2} = Enc(\beta, l, d_2) = (\varepsilon_{\beta,2}, \delta_{\beta,2}). \tag{9}$$

The encryption of a product $d = d_1 \cdot d_2$ with the random parameter $j = k + l \bmod q$ yields a ciphertext $c_{\beta,12}$, equal to the product of two ciphertexts $c_{\beta,1}$ and $c_{\beta,2}$ in Z_p^*, i.e.

$$Enc(\beta, j, d_1 \cdot d_2) = c_{\beta,12} = Enc(\beta, k, d_1) \cdot Enc(\beta, l, d_2) = c_{\beta,1} \cdot c_{\beta,2}. \tag{10}$$

Then

$$c_{\beta,12} = c_{\beta,1} \cdot c_{\beta,2}. \tag{11}$$

According to (10) and (11), this encryption is a multiplicative homomorphism of plaintexts. The value of the multiplied ciphertexts is equal to multiplied value of the underlying transactions.

To verify the validity of the transaction based on balance equation (1), we need to obtain the additively-multiplicative homomorphic encryption by transforming transaction data in the following way:

$$D_1 = g^{d_1}; \qquad D_2 = g^{d_2}, \tag{12}$$

where d_1, d_2 are limited by the upper bound to $2^{32} - 1$ as noted in Sakalauskas *et al.* (2023).

Then (10) can be rewritten in a form denoting the ciphertexts of encrypted transformed data in capital letters C

$$Enc(\beta, j, D_1 \cdot D_2) = C_{\beta,12} = Enc(\beta, k, D_1) \cdot Enc(\beta, l, D_2) = C_{\beta,1} \cdot C_{\beta,2}, \tag{13}$$

where

$$\begin{aligned} C_{\beta,12} &= (\varepsilon_{\beta,1}, \delta_{\beta,1}) \cdot (\varepsilon_{\beta,2}, \delta_{\beta,2}) = (\varepsilon_{\beta,1} \cdot \varepsilon_{\beta,2}, \delta_{\beta,1} \cdot \delta_{\beta,2}) \\ &= \left(g^{(d_1+d_2) \bmod q} \cdot \beta^{(k+l) \bmod q}, g^{(k+l) \bmod q}\right). \end{aligned} \tag{14}$$

The last step allows us to verify transaction balance in (1) by verifying the multiplied encrypted income and expenses with different public keys and proving that these ciphertexts are equivalent using NIZKP.

To create Alice's confidential and verifiable transaction, all her actual incomes $i_1, i_2, \ldots, i_M$ must be transformed to the numbers $I_1, I_2, \ldots, I_M$ using (12) and expenses $e_1, e_2, \ldots, e_{N+1}$, to the numbers $E_1, E_2, \ldots, E_{N+1}$, respectively.

$$I_m = g^{i_m}, \quad m = 1, 2, \ldots, M, \qquad En = g^{e_n}, \quad n = 1, 2, \ldots, N+1. \tag{15}$$

The total transformed income is denoted by I, and the total transformed expenses, by E. Then, referencing homomorphic equation (14), we obtain

$$I_1 \cdot I_2 \cdot \cdots \cdot I_M = I; \qquad E_1 \cdot E_2 \cdot \cdots \cdot E_{N+1} = E. \tag{16}$$

The balance equation (1) can then be rewritten as

$$I = E. \tag{17}$$

4. Creation of Confidential and Auditable Transactions

In the proposed solution, all the information about actual transaction data is available to Alice as a transaction creator and to the AA.

The Bobs in the transaction transfer their expenses as Alice's income $i_1, i_2, \ldots, i_M$, and using (15) computed values $I_1, I_2, \ldots, I_M$. Then the Bobs encrypt them using Alice $\mathrm{PuK}_A = a$ with the randomly generated numbers $k_1, k_2, \ldots, k_M$, thus obtaining ciphertexts $C_{a,I1}, C_{a,I2}, \ldots, C_{a,IM}$, respectively. All Bobs send $(C_{a,I1}, C_{a,I2}, \ldots, C_{a,IM})$ to Alice.

Alice, after receiving ciphertexts $C_{a,I1}, C_{a,I2}, \ldots, C_{a,IM}$ from the Bobs, decrypts them using her $\mathrm{PrK} = x$ and obtains numbers $I_1, I_2, \ldots, I_M$ corresponding to the actual income values $i_1, i_2, \ldots, i_M$ transformed according to (15).

Let us consider that according to the agreement between parties, values $i_1, i_2, \ldots, i_M$ are bounded to $2^{32} - 1$. The computation of $i_1, i_2, \ldots, i_M$ does not, therefore, require us to solve a general discrete logarithm problem in Z_q, which is assumed unfeasible given security requirements. It is enough to perform a search in the set with $2^{32} - 1$ values. Moreover, Alice can reduce the search area with preliminary knowledge about the expected sums to be received. For example, let Alice know that the sums received from her Bobs do not exceed 10.000. Then the search area can be bound by 2^{16} instead of $2^{32} - 1$.

According to (10), Alice multiplies all ciphertexts $(C_{a,I1}, C_{a,I2}, \ldots, C_{a,IM})$, thus obtaining the ciphertext

$$C_{a,I} = C_{a,I1} \cdot C_{a,I2} \cdot \cdots \cdot C_{a,IM}. \tag{18}$$

Referring to the multiplicative homomorphic property (10), the ciphertext $C_{a,I}$ corresponds to the encrypted value I, equal to the multiplication of transformed incomes (15).

$$C_{a,I} = Enc(a, k, I) = (\varepsilon_{a,I}, \delta_{a,I}) = \left(Ia^k, g^k\right), \tag{19}$$

where $k = k_1 + k_2 + \cdots + k_M \bmod q$. Notice that in our construction it is not necessary to have any knowledge about k and its additive components. This was necessary for the method proposed in our previous publication (Sakalauskas *et al.*, 2023), where $k_1, k_2, \ldots, k_M$ were additionally encrypted by the Bobs with Alice's $\mathrm{PuK_A} = a$. Then Alice decrypted them using her $\mathrm{PrK_A} = x$. Therefore, actors made M additional encryptions and M additional decryptions. As we said above, it is not here necessary to perform these steps since we have proposed using NIZKP instead.

After this step, Alice defines expenses $e_1, e_2, \ldots, e_{N+1}$ and transforms them to $E_1, E_2, \ldots, E_{N+1}$ by applying (15).

We do not consider Alice's tax declaration to the AA and her encrypted transfer of expenses to her Larries, since these were presented in the previous paper (Sakalauskas *et al.*, 2023).

Alice multiplies all expenses $(E_1, E_2, \ldots, E_{N+1}) \bmod p$, computing the value

$$E = E_1 \cdot E_2 \cdot \cdots \cdot E_{N+1}. \tag{20}$$

Then the secret random integer $l \leftarrow randi(Z_q)$ is generated by Alice and used for E value encryption using the AA's PuK $= \beta$. Thus, the following ciphertext is obtained:

$$C_{\beta,E} = Enc(\beta, l, E) = (\varepsilon_{\beta,E}, \delta_{\beta,E}) = \left(E\beta^l, g^l\right). \tag{21}$$

Alice must prove to the Net that ciphertexts $C_{a,I}$, and $C_{\beta,E}$ encrypt the same value $I = E$ using NIZKP, thus proving that the transaction is valid and honest.

5. A Zero-Knowledge Proof – ZKP

A ZKP is based on our proposed modifications of the classical Schnorr identification protocol realized in the form of NIZKP (Boneh and Shoup, 2023).

To be self-contained we present the classical ZKP. Recall that the public parameter is a pair, $PP = (p, g)$. Let Alice be a Prover intending to prove to any Verifier (say, the Net) that she knows her private key $\mathrm{PrK_A} = x$ by declaring her public key $\mathrm{PuK}_A = a$. Then $\mathrm{PuK}_A = a$ is called a statement (St), and $\mathrm{PrK_A} = x$ is a witness for a. An interactive ZKP consists of three steps:

1. Alice generates $u \leftarrow randi(Z_q)$, computes the commitment

$$t = g^u,$$

   and sends t to the Verifier;
2. The Verifier generates $h \leftarrow randi(Z_q)$ and sends it to Alice;
3. Alice computes a response

$$r = xh + u \bmod q,$$

   and sends r to the Verifier.

The Verifier verifies the following identity of terms to be convinced that Alice knows her $\mathrm{PrK_A} = x$.

$$g^r = a^h \cdot t.$$

This technique is generalized to allow NIZKP to convince the Net that two different ciphertexts $C_{a,I}$ in (18) and $C_{\beta,E}$ in (21) are obtained by encryption of the same plaintext with different public keys, namely $\mathrm{PuK}_A = a$ and $\mathrm{PuK}_{AA} = \beta$. We assume that the verifier Net is a so-called honest verifier, and the proof represents the so-called Honest Verifier ZKP.

Referring to Boneh and Shoup (2023), interactive ZKP can be transformed to non-interactive by replacing the random value h generated by the Verifier with the h-value computed by the Prover using a cryptographically secure h-function. This is the basis of the NIZKP scheme.

However, the scheme presented above is insufficient to realize a proof of ciphertext equivalency. We propose the modification of the existing NIZKP to realize two ciphertext equivalency proofs, namely $C_{a,I}$ in (18), (19), and $C_{\beta,E}$ in (20), (21). Recall that $C_{a,I}$ is a ciphertext of plaintext I encryption with Alice's PuK $= a$ and $C_{\beta,E}$ is a ciphertext of plaintext E encryption with the AA's PuK $= \beta$. The statement St of our proposed NIZKP consists of the following:

$$St = \left\{(\varepsilon_{a,I}, \delta_{a,I}), (\varepsilon_{\beta,E}, \delta_{\beta,E}), a, \beta\right\}. \tag{22}$$

The random integers $u \leftarrow randi(Z_q)$ and $v \leftarrow randi(Z_q)$ are generated by Alice, and the value $(-v) \bmod q$ is computed. The proof of ciphertext equivalence is computed using three computation steps:

1. The following commitments are computed:

$$t_1 = g^u \bmod p; \tag{23}$$
$$t_2 = g^v \bmod p; \tag{24}$$
$$t_3 = (\delta_{a,I})^u \cdot \beta^{-v} \bmod p. \tag{25}$$

2. The following h-value is computed using the cryptographically secure h-function H:

$$h = H\left(a \| \beta \| t_1 \| t_2 \| t_3 \|\right). \tag{26}$$

3. Alice, having her $\mathrm{PrK_A} = x$, randomly generates the secret number l for E encryption and computes the following two values:

$$r = x \cdot h + u \bmod q; \tag{27}$$
$$s = l \cdot h + v \bmod q. \tag{28}$$

Then Alice declares the following set of data to the Net:

$$\{a, \beta, t_1, t_2, t_3, r, s\} \rightarrow \text{Net}. \tag{29}$$

To verify the transaction's validity, the Net computes the h-value according to (26) and then verifies three identities:

$$g^r = a^h \cdot t_1; \tag{30}$$
$$g^s = (\delta_{\beta,E})^h \cdot t_2; \tag{31}$$
$$(\varepsilon_{\beta,E})^h \cdot (\varepsilon_{a,I})^{-h} \cdot (\delta_{a,I})^r \cdot \beta^{-s} = t_3. \tag{32}$$

The correctness of (30), (31) is proved by the following identities:

$$g^r = g^{xh+u} = g^{xh} \cdot g^u = \left(g^x\right)^h \cdot g^u = a^h \cdot t_1; \tag{33}$$
$$g^s = g^{lh+\mathrm{v}} = g^{lh} \cdot g^v = \left(g^l\right)^h \cdot g^v = (\delta_{\beta,E})^h \cdot t_2. \tag{34}$$

The correctness of (32) is proved by considering every multiplier separately:

$$(\varepsilon_{\beta,E})^h = \left(E \cdot \beta^l\right)^h = E^h \cdot \beta^{lh}; \tag{35}$$

$$(\varepsilon_{a,I})^{-h} = \left(I \cdot a^k\right)^{-h} = I^{-h} \cdot a^{-kh}; \tag{36}$$

$$\begin{aligned}(\delta_{a,I})^r &= \left(g^k\right)^r = \left(g^{kxh+ku}\right) = \left(g^x\right)^{hk} \cdot \left(g^k\right)^u \\ &= a^{hk} \cdot \left(g^k\right)^u = a^{hk} \cdot (\delta_{a,I})^u;\end{aligned} \tag{37}$$

$$\beta^{-s} = \beta^{-lh-v} = \beta^{-lh} \cdot \beta^{-v}. \tag{38}$$

Notice that k is not known to Alice and is included in $(\delta_{a,I})$. If the transaction is honest, then the transaction balance (1) is satisfied and $I = E$ since. Then $E^h \cdot I^{-h} = 1 \bmod p$, and putting it all together, we obtain:

$$E^h \cdot \beta^{lh} \cdot I^{-h} \cdot a^{-kh} \cdot a^{hk} \cdot (\delta_{a,I})^u \cdot \beta^{-lh} \cdot \beta^{-v} = (\delta_{a,I})^u \cdot \beta^{-v} = t_3. \tag{39}$$

This is the proof to the Net that the balance equation (1) is valid.

6. Security Considerations

It is known that ElGamal encryption possesses semantic security and is secure against eavesdropping attacks. The modification of classical ElGamal encryption for the additively-multiplicative homomorphic encryption does not add or subtract security. It is furthermore proven that the classical Schnorr identification protocol is secure against eavesdropping attacks (Boneh and Shoup, 2023).

The present modification of the classical Schnorr identification scheme is based on two parallel proofs of knowledge, i.e. of $\mathrm{PrK_A} = x$ in (4) and of a secret random parameter l in (21) used for probabilistic encryption. Therefore, these parallel proofs are also secure against eavesdropping attacks. The third proof is the combination of the two previous proofs proving the equivalency of two ciphertexts meaning that total transformed income I in (16) is encrypted by Alice's public key a and the total transformed expense E in (16) is encrypted with the AA's public key β, when, according to (17), $I = E$. This proof neither adds nor subtracts security. Therefore, the proposed scheme is also secure against eavesdropping attacks.

We refer to the following facts and the proof of statements presented below, as detailed in numerous publications (e.g. Boneh and Shoup, 2023), providing our security considerations.

Fact 1. *Since the group G_q is of prime order, then all elements, except* 1*, are the generators and the order q of G_q is super-polynomial.*

Fact 2. *The DEF in* (3) *is a* 1*-to-*1 *function.*

Fact 3. *The decisional Diffie–Hellman assumption (DDH) holds in the subgroup G_q.*

Fact 4. *Since DDH holds, then the discrete logarithm assumption (DDA) holds in G_q as well. This means that for every probabilistic polynomial time algorithm, the probability to find i in (3) is negligible.*

The following known statements hold.

Statement 1. ElGamal additively-homomorphic encryption achieves semantic security.

Statement 2. Schnorr's identification protocol is secure against eavesdropping attacks.

Statement 3. Schnorr's non-interactive identification protocol is secure against eavesdropping attacks if the challenge component in Schnorr's identification protocol is replaced by h-value computation using H-function modelled as a random oracle.

Now we can turn to the proof of security for a non-interactive zero-knowledge proof of ciphertext equivalency starting from the following clear lemma.

Lemma 1. *Let expenses E be encrypted using the AA's public key β with a secret random parameter l. The obtained ciphertext $C_{\beta,E}$ in (21) can then be decrypted using the parameter l without knowledge of the AA's private key z.*

Proof. For decryption, it is necessary to compute the value $\beta^{-l} \bmod p$, where $-l$ is computed mod q. Then

$$E = (\varepsilon_{\beta,E}) \cdot \beta^{-l} = E\beta^{l} \cdot \beta^{-l}. \quad \square$$

Corollaries.

1. *The AA's private key z and random secret number l are independent secrets representing the plaintext E and the ciphertext $C_{\beta,E}$.*
2. *Since the discrete exponential function is 1-to-1, then for fixed z, the parameter l uniquely represents the plaintext E and the ciphertext $C_{\beta,E}$, and vice versa: for fixed l, the private key z uniquely represents the plaintext E and the ciphertext $C_{\beta,E}$.*

According to Lemma 1 and the Corollaries, the first two steps of the presented modification of NIZKP are based on two independent and parallel proofs of knowledge: one for $\mathrm{PrK_A} = x$ in (4) and another for the secret random parameter l in (21) used for probabilistic encryption. These proofs use the classical Schnorr identification scheme. Therefore, these two parallel proofs are also secure against eavesdropping attacks.

The third proof is the combination of the two previous proofs, demonstrating the equivalence of two ciphertexts. Specifically, this means that the total transformed income I in (16) is encrypted with Alice's public key a, and the total transformed expense E in (16) is encrypted with the AA's public key β. According to (17), it means that $I = E$.

Let us consider the commitment t_3 consisting of the two secret parameters u and v. According to Fact 1, all the elements of G_q, except 1, are generators. We can interpret the

value $\delta_{a,I}$ as a the generator g_1 and β^{-1} as the generator g_2. Then (25) we can rewrite it in the following way:

$$t_3 = (g_1)^u \cdot (g_2)^v.$$

The last equation indicates that t_3 can be represented by generators g_1 and g_2 with two indices u and v, respectively. In other words, t_3 has representation of generator-tuple of length 2, (2-tuples).

Lemma 2. *For all representations of t_3 there are exactly q representations of 2-tuples.*

Proof. Referencing Fact 1, we can find v as a discrete logarithmic function Dlog in the following way:

$$t_3 = \mathrm{Dlog}\big(t_3/(g_1)^{-u}\big).$$

So, there are exactly q possible values of u to make t_3 representations. The lemma is proved. □

The security proof is based on the fact that the Honest Prover will always convince the Honest Verifier about the ciphertexts' equivalency. It is called a completeness proof.

Theorem 1. *The presented modified NIZKP protocol is secure against an eavesdropping attack.*

Proof. By inspecting h-value expression in (26), we see that it depends on the values a, β, t_1, t_2 and t_3. The values a and β are predetermined and cannot be modified. According to Fact 1, the commitments t_1 and t_2 are in 1-to-1 correspondence with the randomly generated parameters u and v. Therefore, by fixing t_1 and t_2, we are also fixing u and v. In this case, referencing Lemma 1, the number of representations of t_3 is exactly 1. Assuming that H-function is secure (collision free), h-value is fixed and determined by a, β, t_1, t_2 and t_3. Consequently, the values r and s in (27) and (28) are also fixed. Therefore, the ciphertext equivalency proof based on equations (30), (31), and (32) satisfies the completeness condition.

Referencing Fact 3, the commitment t_3 does not add any insecurity to this protocol compared to the classical non-interactive Schnorr identification protocol. □

7. Conclusions

A method for the confidential verification of transaction balances by the Net has been presented using the UTxO system, in a manner providing transparency and trustworthiness for blockchain transactions. The honesty of transactions can be verified by anyone on the Net without any knowledge about their actual values.

The proposed scheme provides a proof to the Net that encrypted transaction data satisfies the balance equation (1) without revealing any information about the actual transaction data. It is guaranteed by the semantic security of ElGamal encryption and, in this paper, by a proposed a non-interactive zero-knowledge proof (NIZKP) based on Schnorr identification. Proof of security against eavesdropping attacks for the proposed NIZKP is presented. In the literature, this problem is generally known as a ciphertext equivalency proof.

The novelty of this proposed solution lies in the integration of additively-multiplicative homomorphic ElGamal encryption with our constructed NIZKP and its application to blockchain technology based on a UTxO system. The difference from other known approaches is that our NIZKP is constructed for different actors making independent encryptions with two different public keys.

The security of the proposed NIZKP against an eavesdropping adversary is proved.

The proposed scheme uses NIZKP, which reduces the number of encryptions and the number of decryptions by a factor of two for income, as compared with the previous authors' results.

We intend future research to focus on implementing a sigma identification protocol for the ciphertext equivalency proof that is secure against active adversary attacks. The other possible direction for the presented methodology is to create new NIZKP schemes based on the potential of the matrix power function to provide security against quantum cryptanalysis attacks. This could involve some contribution to the task of integrating post-quantum cryptography with blockchain technologies.

References

Boakye, E.A., Zhao, H., Ahia, H. (2022). Emerging research on blockchain technology in finance; a conveyed evidence of bibliometric-based evaluations. *The Journal of High Technology Management Research*, 33(2), 100437. https://doi.org/10.1016/j.hitech.2022.100437. https://www.sciencedirect.com/science/article/pii/S1047831022000128.

Boneh, D., Shoup, V. (2023). *A Graduate Course in Applied Cryptography. Draft 0.6*. URL: A Graduate Course in Applied Cryptography (stanford.edu).

Bunz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G. (2018). Bulletproofs: short proofs for confidential transactions and more. In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 315–334.

Blazy, O., Bultel, X., Lafourcade, P., Kempner, O.P. (2021). Generic Plaintext Equality and Inequality Proofs (Extended Version). *Cryptology ePrint Archive*. https://eprint.iacr.org/2021/426.pdf.

Canard, S., Fuchsbauer, G., Gouget, A., Laguillaumie, F. (2012). Plaintext-checkable encryption. In: *Topics in Cryptology–CT-RSA 2012: The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27-March 2, 2012. Proceedings*. Springer, Berlin Heidelberg, pp. 332–348. https://inria.hal.science/docs/00/76/83/05/PDF/PCE_RSA.pdf.

Dong, S., Zhao, Z., Wang, B., Gao, W., Zhang, S. (2023). Certificateless encryption supporting multi-ciphertext equality test with proxy-assisted authorization. *The Journal of High Technology Management Research*, 12(20), 4326. https://www.mdpi.com/2079-9292/12/20/4326.

ElGamal T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *The Journal of High Technology Management Research*, 31(4), 469–472. https://doi.org/10.1109/TIT.1985.1057074.

Freeman, D.M. (2011). *Schnorr Identification and Signatures*. http://web.stanford.edu/class/cs259c/lectures/schnorr.pdf.

Guomin, Y., Chik, T.H., Qiong, H., Duncan, W.S. (2010). Probabilistic public key encryption with equality test. In: *Topics in Cryptology: Cryptographers' Track at the RSA Conference, CT-RSA 2010, San Francisco, March*

1–5: Proceedings, *Lecture Notes in Computer Science*, Vol. 5985, pp. 119–131. https://ink.library.smu.edu.sg/sis_research/7419.

Hongbo, L., Huang, Q., Ma, S., Shen, J., Susilo, W. (2019). Authorized equality test on identity-based ciphertexts for secret data sharing via cloud storage. *IEEE Access*, 7, 25409–25421. https://doi.org/10.1109/TIT.1985.1057074.

Muleravičius, J., Timofejeva, I., Mihalkovich, A., Sakalauskas, E. (2019). Authorized equality test on identity-based ciphertexts for secret data sharing via cloud storage. *IEEE Access*, 30(2), 327–348. https://doi.org/10.15388/Informatica.2019.208.

Pinna, A., Tonelli, R., Orrú, M., Marchesi, M. (2018). A petri nets model for blockchain analysis. *The Computer Journal*, 61(9), 1374–1388. https://doi.org/10.1093/comjnl/bxy001.

Sakalauskas, E., Muleravičius, J., Timofejeva, I., (2017). Computational resources for mobile E-wallet system with observers. In: *2017 Electronics*. IEEE, pp. 1–5. https://doi.org/10.1109/ELECTRONICS.2017.7995226.

Sakalauskas, E., Timofejeva, I., Michalkovič, A., Muleravičius, J. (2018). A simple off-line E-cash system with observers. *Information Technology and Control*, 47(1), 107–117. http://itc.ktu.lt/index.php/ITC/article/view/18021/9326.

Sakalauskas, E., Bendoraitis, A., Lukšaitė, D., Butkus, G., Vitkutė-Adžgauskienė, D. (2023). A petri nets model for blockchain analysis. *Informatica*, 34(4), 603–616. https://content.iospress.com/articles/informatica/infor531.

Zhao, M., Ding, Y., Tang, S., Liang, H., Wang, H. (2022). Public key encryption with authorized equality test on outsourced ciphertexts for cloud-assisted IoT in dual server model. *Wireless Communications and Mobile Computing*, 2022, 1–10. https://www.hindawi.com/journals/wcmc/2022/4462134/.

A. Kilčiauskas is a PhD student of natural sciences' informatics at the Department of Applied Mathematics. His research area is anonymity, confidentiality, and verifiability functionalities implementation in private blockchain transactions.

A. Bendoraitis is a PhD student at the Faculty of Informatics. He has finished a three-semester competence course in cryptography and blockchain systems at the Department of Applied Mathematics.

E. Sakalauskas is a professor at the Department of Applied Mathematics, Kaunas University of Technology. He is the head of the Cryptography and Blockchain Systems research group. The scope of his scientific interests is the creation of new cryptographic methods including post-quantum methods and their security analysis. Other area of activity is cryptographic method application to private blockchain technologies, additional functionality and trustworthiness.