IFSA Publishing

Blockchain and Cryptocurrency

Published by IFSA Publishing, S. L., 2026
https://bc-ifsa-journal.com/

Private Blockchain Anonymization-Deanonymization System Preserving Anonymity for the Net

Eligijus Sakalauskas, Antanas Bendoraitis, Syeda Roushan Arshid, Aušrys Kilčiauskas, Aleksejus Michalkovič, Lina Dindienė, Kęstutis Lukšys
Kaunas University of Technology, Scientific Group of 'Cryptography and blockchain systems',
Studentų str. 50, Kaunas, Lithuania
Tel.: + 370 698 784 77
E-mail: eligijus.sakalauskas@ktu.lt

Received: 22 Dec. 2025 /Revised: 19 Mar. 2026 /Accepted: 20 Mar. 2026 /Published: 23 Mar. 2023

Abstract: In a private blockchain, it is essential to provide not only anonymization of users but also deanonymization for certain parts of the Network, such as audit organizations, investment companies, and others, while maintaining anonymity for the other parts of the Network. Traditionally, the anonymization in private blockchain is performed by implementing ring signatures based on elliptic curve cryptography. The deanonymization problem is not considered in connection with the anonymization problem. In this paper, we present a unified anonymization-deanonymization system. Anonymization is based on an arbitrary set of single-user addresses generated independently and secretly. These addresses correspond to the user's private and public keys used for transaction creation based on the Schnorr signature. Deanonymization is based on the Schnorr multi-signature scheme and Non-Interactive Zero Knowledge proof. The presented solution is an integration of Schnorr signature, Schnorr multi-signature, and Schnorr-based Non-Interactive Zero Knowledge Proof. Security and effectiveness analysis are presented.

Keywords: Private blockchain, Transaction's anonymity, Ring signatures, Anonymization, Deanonymization.

1. Introduction

Anonymization is a relevant topic in private blockchain, alongside other confidentiality requirements. For example, this problem in the Monero blockchain is addressed by Ring Signatures (RS) based on Elliptic Curve Cryptography (ECC) [1-3]. On this background, the Elliptic Curve Digital Signature Algorithm (ECDSA) was created and standardised.

Nevertheless, such anonymity can be interpreted as partial anonymity on the Network, since the actual transaction signer is inevitably among the other ring signers, who are members of the ring created by the actual transaction signer. A shortcoming of Monero RS-based anonymity is that it should be performed during transaction signing, which requires additional computational resources. To increase this kind of anonymity, it is required to increase the number of ring members. At the same time, computational resources are increasing. Therefore, there is a need to develop a more anonymous transaction system that requires fewer computational resources.

However, there are also situations in which the opposite process, such as deanonymization, is required. For example, income from anonymous transactions should be redirected to the Investment Company (IC) by revealing the identity of the cryptocurrency owner.

In the case of Monero RS, deanonymization does not require additional computational resources because all funds are held in the transaction creator's account.

In this paper, we present an alternative approach to achieve actual anonymity and a more effective realization based on the classical Schnorr signature together with a deanonymization method based on the Schnorr multi-signature. Deanonymization requires providing a small amount of additional information to IC, together with the required signature for this additional information.

We consider a blockchain system based on the Unspent Transactions Output (UTxO) paradigm [4].

In Section 2, we present an anonymization-deanonymization method based on RS and used in the Monero blockchain. In Section 3, the proposed method of anonymization-deanonymization based on Schnorr signature and Schnorr multi-signature is presented. Section 4 is supported by example, and Section 5 includes security considerations. In Section 6, the conclusions are presented.

2. Anonymization-deanonymization based on Ring Signatures (RS)

This anonymization method is used in the Monero blockchain, providing the transaction's creator with anonymity. RS is a type of digital signature that can be performed by any member of a set of users who each have a pair of public and private keys [5]. Therefore, a message signed with an RS is endorsed by someone in a particular set of people. One of the security properties of a ring signature is that it should be computationally infeasible to determine which of the set's members' keys was used to produce the signature. In RS, there is no way to revoke the anonymity of an individual signature, and any set of users can be used as a signing set without additional setup.

Ring signatures are signer-ambiguous. In a ring signature scheme, there are no prearranged groups of users, there are no procedures for setting, changing, or deleting groups, there is no way to distribute specialized keys, and there is no way to revoke the anonymity of the actual signer (unless he decides to expose himself).

The only assumption is that each member is already associated with the public key of some standard signature scheme.

To produce a ring signature, the actual signer declares an arbitrary set of possible signers that includes himself and computes the signature entirely by himself using only his private key and the others' public keys.

In particular, other possible signers may have chosen their private keys solely to conduct e-commerce over the internet. They may be completely unaware that a stranger uses their public key to produce a ring signature on a message they have never seen and would not wish to sign.

A set of possible signers is named a ring. The ring member who produces the actual signature is the signer, and each of the other ring members is a non-signer.

A ring signature scheme is set-up free: The signer does not need the knowledge, consent, or assistance of the other ring members to put them in the ring - all he needs is knowledge of their regular public keys. The size of the signature depends on the number of ring members.

Verification must satisfy the usual soundness and completeness conditions. It is required that the signatures be signer-ambiguous in the sense that the verifier should be unable to determine the identity of the actual signer in a ring of size r with probability greater than $\frac{1}{r}$.

The construction based on ECDSA provides unconditional anonymity in the sense that even an infinitely powerful adversary with access to an unbounded number of chosen-message signatures produced by the same ring member cannot guess his identity with any advantage, and cannot link additional signatures to the same signer.

In Monero, RS are implemented using elliptic curve cryptography. These curves are defined over a finite field $\mathbb{F}p$, where p is a prime number. The field formed by the set $\{0, 1, 2, \dots, p-1\}$, with arithmetic operations $(+, \cdot)$ calculated $(mod\ p)$.

Typically, elliptic curves are defined as the set of points (x, y) satisfying a *Weierstraß* equation:

$$y^2 = x^3 + ax + b \quad \text{where } a, b, x, y \in \mathbb{F}\text{p}. \tag{1}$$

However, the cryptocurrency Monero uses a special curve known to offer improved security over other commonly used NIST curves, as well as excellent performance of cryptographic primitives. The curve used belongs to the category of so-called *Twisted Edwards* curves, which are commonly used [1]. In EC, the special points addition is defined, creating an abelian group of these points.

A generator G of EC is a point on the curve such that for every other point P in EC there exists k such that $P = kG$.

Public key cryptography algorithms can be devised in an analogous way to modular arithmetic.

Let k be a randomly selected number satisfying $1 < k < N_{EC}$, where N_{EC} is a number or EC points. Number k represents a *private key*. Then the corresponding *public key* $K = kG$, where K is an EC point.

Typically, a cryptographic signature is performed on a cryptographic hash of a message.

2.1. Signature

Assume that Alice has the private/public key pair (k, K). To sign an arbitrary message M univocally, she could execute the following steps [2]:

1. Calculate a hash of the message using a cryptographically secure hash function, $h = H(M)$

2. Generate a random integer r such that $1 < r < N$ and compute $P = (x, y) = rG$.
   If $x = 0$ generate another random integer.
3. Calculate $s = r^{-1}(h + xk)(mod\ N)$. If $s = 0$, then go to the previous step and repeat
4. The signature is $\sigma = (x, s)$.

2.2. Verification

Any third party can verify the signature by calculating

$$u_1 = s^{-1}h \tag{2}$$

$$u_2 = s^{-1}r \tag{3}$$

$$Q = u_1G + u_2K \tag{4}$$

2.3. Correctness

The correctness of the scheme can be derived from the fact that

$$\begin{aligned} Q &= u_1G + u_2K \\ &= s^{-1}hG + s^{-1}rkG \\ &= s^{-1}(h + xk)G \end{aligned} \tag{5}$$

Since $s = r^{-1}(h + xk)$, it follows that $r = s^{-1}(h + xk)$, whereby it is proved that

$$Q = rG \tag{6}$$

In the Monero blockchain, the anonymization is performed using ring signatures (RS) presented in [3].

2.4. Anonymization

The simplified version of this scheme is presented below, assuming that we have the same number of keys for any value of the first index i. Assume that we have a set of public keys $\{K_{i,j}\}$ for $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$. Furthermore, we also assume that for each *i,* there is an index π_i such that the signer knows the private key k_{i,π_i} corresponding to K_{i,π_i}.

In what follows, we will use *M* for the hash of the message concatenated with keys $K_{i,j}$.

2.5. Ring Signature

1. For each $i = 1, \dots, n$:
   (a) generate a random value $\alpha_i \in_R \mathbb{Z}_q$
   (b) set $c_{i,\pi_i} = H_n(M, \alpha_i G, i, \pi_i)$
   (c) for $j = \pi_i + 1, \dots, m-1$ generate random numbers $r_{i,j} \in_R \mathbb{Z}_q$ and compute,

$$c_{i,j+1} = H_n\big(M, r_{i,j}G - c_{i,j}K_{i,j}, i, j\big) \tag{7}$$

2. For $i = 1, \dots, n$ generate random numbers $r_{i,m} \in_R \mathbb{Z}_q$ and compute

$$c_1 = H_n\big(r_{1,m}G - c_{1,m}K_{1,m}, \dots, r_{n,m}G - c_{n,m}K_{i,m}\big) \tag{8}$$

3. For $i = 1, \dots, n$:
(a) for $j = 1, \dots, \pi_i - 1$generate random numbers $r_{i,j} \in_R \mathbb{Z}_q$ and compute

$$c_{i,j+1} = H_n\big(M, r_{i,j}G - c_{i,j}K_{i,j}, i, j\big) \tag{9}$$

Here, we interpret references to $c_{i,1}$ as c_1, see the previous step.
(b) set $r_{i,\pi_i} = \alpha_i + k_{i,\pi_i}c_{i,\pi_i}$
The ring signature is

$$\sigma_{RS} = \big(c_1, r_{1,1}, r_{1,2}, \dots, r_{1,m}, \dots, r_{n,m}\big) \tag{10}$$

2.6. Verification

As in the previous section, let m denote the hash of the message to be signed, together with the corresponding set of signing keys. The verification of a given signature is performed as follows:

1. For each $i = 1, \dots, n$ and $j = 1, \dots, m$ compute:

$$R'_{i,j+1} = r_{i,j}G - c'_{(i,j)}K_{i,j} \tag{11}$$

$$c'_{i,j+1} = H_n\big(M, R'_{i,j+1}, i, j\big) \tag{12}$$

   Interpret any $c'_{i,1}$ as c_1
2. Compute $c'_i = H\big(R'_{1,m}, \dots, R'_{n,m}\big)$

The signature will be valid if $c'_1 = c_1$.

2.7. Correctness

1. For $j \neq \pi_i$ and for all i, we can readily see that $c'_{i,j+1} = c_{i,j+1}$.
2. When $j = \pi_i$, for all i

$$\begin{aligned} R'_{i,j+1} &= r_{i,j}G - c'_{i,j}K_{i,j} \\ &= (\alpha_i + ki, \pi_i c'_i, \pi_i)\ G - c'_i, \pi_i K_i, \pi_i \end{aligned} \tag{13}$$

In other words, $c'_{i,\pi_i} = H_n(M, \alpha_i G, i, \pi_i) = c_{i,\pi+1}$. Therefore, we can conclude that the verification step identifies correctly valid signatures.

2.8. Deanonymization

Deanonymization is performed in a trivial way. Alice simply transfers a required sum from her account to IC and signs her transaction with her private key without using any other public keys required for RS creation.

2.9. Efficiency

Most of the proposed algorithms have an asymptotic output size $O(n)$, i.e., the size of the resulting signature increases linearly with the size of

the input (number of public keys) corresponding to the number of chosen ring members. The computational cost is estimated by identifying the operations that require significant computational resources. Such an operation in ECC is a multiplication of EC point by the number, i.e. rG, where r is a number in the field $\mathbb{F}_p$, and G is an EC generator. This operation requires a significant computational resources and, in some sense, is analogous to the exponentiation operation used in the classical ElGamal cryptosystem The number of EC point multiplications by scalars in $\mathbb{F}_p$ required for RS implementation is denoted by N_{ECO}. According to the construction presented in Section 2.5, this number depends on the number n of ring members and is equal to

$$N_{RS} = 4n^3. \quad (14)$$

As we see, to provide more anonymity, a greater number of ring members is needed, and cubically more of EC point multiplication operations of EC points are required.

The other drawback is that these operations must be carried out during the transaction execution, i.e., online.

3. Anonymization-deanonymization based on the Schnorr Signature Approach

This approach is based on Schnorr signature for anonymization [6] and Schnorr multi-signature for deanonymization [7]. As usual, every user has a private and public key pair, which we denote by $(\mathrm{PrK}, \mathrm{PuK})$ for signature creation and verification, respectively. Moreover, every user of blockchain can generate as many $(\mathrm{PrK}, \mathrm{PuK})$ pairs as required.

Let p be a large prime of order 2^{2048}, and $\boldsymbol{Z_p^*}$ is a multiplicative group with operation $mod\ p$. Let $\boldsymbol{G_q}$ be a subgroup of $\boldsymbol{Z_p^*}$ of order q, where q is prime. Then all elements in $\boldsymbol{G_q}$ except 1 are generators. Let $g \in \boldsymbol{G_q}$ be a generator in $\boldsymbol{G_q}$, then the order of the generator g is q.

In the Schnorr signature scheme, public parameters (PP) consist of the following public parameters $PP = (p, q, g)$. In our approach, user anonymization is achieved by generating as many blockchain accounts as required. This simple procedure therefore conceals the true identity of the account owner.

In the Schnorr signature scheme, a key pair $(\mathrm{PrK}, \mathrm{PuK})$ is generated using PP in the following way:

$$\mathrm{PrK} = x \leftarrow rand(\boldsymbol{Z_q}), \quad (15)$$

$$\mathrm{PuK} = g^x\ mod\ p = a. \quad (16)$$

In (15), private key x is generated at random in the semiring $\boldsymbol{Z_q} = \{0, 1, 2, \dots, q-1\}$, where addition and multiplication operations mod q are defined. Moreover, all arithmetical operations in the exponents can be reduced modulo q. Therefore, all exponents are in the set $\boldsymbol{Z_q}$.

The public key a is calculated using a discrete exponential function (DEF), which is considered as a classical conjectured one-way function (OWF). This function is used to ensure security against cryptanalysis by classical computers when public parameters p and q are chosen sufficiently large, i.e., p has a 2048-bit length, and q is more than 1024-bit length.

Assume that the blockchain account address is computed traditionally, i.e., using a secure (collision-resistant) cryptographic function. H-function. This function is specified in the blockchain architecture. Symbolically, we denote it by

$$Addr = H(\mathrm{Puk}). \quad (17)$$

As we see $\mathrm{PrK} = x$ and $\mathrm{PuK} = a$ are linked with the Alice account, we name it the main account.

3.1. Anonymization

Let Alice wish to be anonymous on the Net, except to her business partners, by generating at random N pairs of public and private keys to create N anonymous accounts. First, she generates a set of $N-1$ private key $\mathrm{PrK}_{n-1} = x_{n-1}, n = 1, 2, \dots, N$. The last private key is computed by the expression

$$x_N = x - x_1 - x_2 - \dots - x_N - 1\ mod\ q. \quad (18)$$

Hence, evidently, we have:

$$\mathrm{PrK} = x = x_1 + x_2 + \dots + x_N\ mod\ q. \quad (19)$$

Then, using (16) and (17), public keys and addresses are computed.

The set of addresses and public keys is known to the nodes of the Net. But the Net does not know that the set of addresses $\{Addr_n\}$ and the corresponding set of public keys $\{a_n\}, n = 1, 2, \dots, N$ belong to Alice. To ensure anonymity, Alice uses her anonymous addresses to conduct transactions between her partners. All transactions are signed using the Schnorr signature scheme.

To be self-contained, we present the Schnorr signature scheme to sign the transaction Tx_i

$$j_i \leftarrow \mathrm{rand}(\boldsymbol{Z_q}). \quad (20)$$

$$r_i = g^{j_i} mod\ p. \quad (21)$$

$$h_i = H(\mathrm{Tx}_i || r_i). \quad (22)$$

$$s_i = j_i + x_i \cdot h_i\ mod\ q. \quad (23)$$

Alice's signature on Tx_i is $\sigma_i = (r_i, s_i)$.

The verification of the signature σ_i is performed with Alice $\text{PuK}_i = a_i$ by verifying the following identity

$$g^{s_i} \bmod p = r_i \cdot (a_i)^{h_i} \bmod p. \quad (24)$$

All nodes in the net can verify the validity of the transaction signature by using (24) and acquiring the public key a_i. This verification is secure under the DL assumption in the random oracle model.

For more clarity, let us assume that Alice created two anonymous account addresses $\{Addr_1, Addr_2\}$ as indicated in Fig.1, corresponding to $(PrK_1 = x_1, PuK_1 = a_1)$ and $(PrK_2 = x_2, PuK_2 = a_2)$ respectively.

Let Alice decide to invest some income in her addresses $\{Addr_1, Addr_2\}$ to the Investment Company (IC). Then she creates two transactions Tx1, Tx2, where investment sums are outputs (expenses) of these transactions. These transactions are signed by Schnorr signature using Alice's private keys $PrK_1 = x_1$ and $PrK_2 = x_2$ generated by her in advance. Using (20)-(23) two signatures are computed:

$$\sigma_1 = (r_1, s_1); \ \sigma_2 = (r_2, s_2). \quad (25)$$

IC, together with the received sums in Tx1 and Tx2, verifies signatures on these transactions using (24). But IC does not know that these transactions are sent from Alice's anonymous accounts. IC can also believe that it receives funds from the different investors.

3.2. Deanonymization

After Alice sends Tx1 and Tx2 to the IC, she must prove that this amount of money belongs to her and that she is an investor. For this purpose, Alice uses Schnorr-based Non-Interactive Zero Knowledge Proof (NIZKP) to demonstrate that she knows the $PrK = x$, as defined in (19), without revealing it. The general idea of an NIZKP is illustrated in Fig. 1.

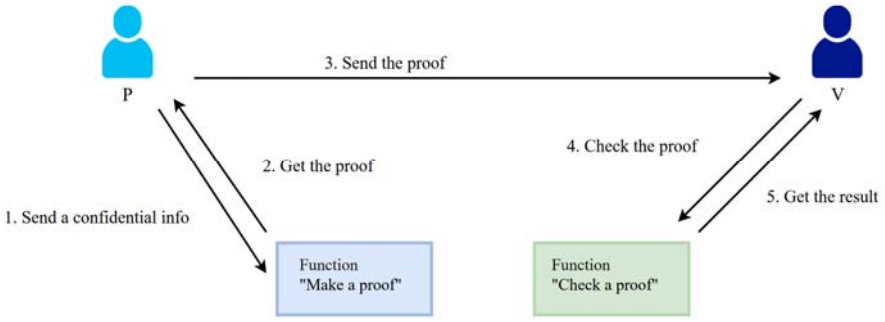

Fig. 1. The general idea of NIZKP.

The idea of a Schnorr-based NIZKP is similar to that of the signature scheme, in that the three-round interactive Schnorr identification protocol is transformed into a single-round non-interactive proof using the Fiat-Shamir heuristic [11].

Following (16) and (19), when $N = 2$, the following equations hold

$$x = x_1 + x_2 \bmod q \quad (26)$$

$$g^x = g^{x_1 + x_2} = a_1 \cdot a_2 = a \bmod p \quad (27)$$

Then, having two signatures (σ_1, σ_2)Alice computes the multiplication of these signatures in a special way, yielding a Schnorr multi-signature. This special multiplication operation, we denote by $*$. As a result, the Schnorr multi-signature is computed in the following way:

$$\begin{aligned} \sigma_{12} &= (\sigma_1 * \sigma_2) = (r_1, s_1) * (r_2, s_2) = \\ &= (r_1 \cdot r_2 \bmod p, s_1 + s_2 \bmod q) = \\ &= (r_{12}, s_{12}). \end{aligned} \quad (28)$$

The Schnorr multi-signature is secure under the DL assumption in a random oracle model. In other words, any polynomial-time adversary cannot produce a forgery with non-negligible advantage based on the challenger's responses to hash and signing queries. This property is known as Existential Unforgeability under Chosen-Message Attack (EUF-CMA).

However, Schnorr multi-signature can be created by anyone on the Net, and, therefore, it itself cannot be a proof of Alice's transactions Tx1 and Tx2. To prove that she is a creator of Tx1 and Tx2, Alice is using Schnorr-based NIZKP to the verifier IC.

To start with, Alice generates at random a number $j \leftarrow rand(\mathbb{Z}_q)$ and computes

$$r = g^j \bmod p \quad (29)$$

Using a collision-resistant H-function, Alice computes the following h-value playing the role of the challenge in Schnorr identification:

$$h = H(Tx1, Tx2, \sigma_1, \sigma_2, \sigma_{12}, a_1, a_2, a || r) \quad (30)$$

She now calculates

$$s = j + x \cdot h \tag{31}$$

Then NIZKP of the prover data we express as follows:

$$\eta = (r, s) \tag{32}$$

The prover Alice sends the following data to the verifier IC:

$$\{Tx1, Tx2, \sigma_1, \sigma_2, \sigma_{12}, a_1, a_2, a, \eta\} \rightarrow IC \tag{33}$$

IC, after receiving this data, performs the following verifications.

1. Are transactions $\{\mathrm{Tx}_1, \mathrm{Tx}_2\}$ on the account of IC.
2. Verifies signatures (σ_1, σ_2) on these transactions using (24).
3. IC Schnorr multi-signature σ_{12} by checking the validity of the following identity

$$g^{s_{12}} mod\ p = r_{12} \cdot (a_1)^{h_1} \cdot (a_2)^{h_2} mod\ p \tag{34}$$

4. IC verifies if

$$a_1 \cdot a_2 = a\ mod\ p \tag{35}$$

5. Then, IC verifies if NIZKP is correct using $\eta = (r, s)$ and checking the identity

$$g^s = r \cdot (a)^h mod\ p \tag{36}$$

The last equation can be rewritten in the form

$$g^{s_1+s_2}\ mod\ p = r_1 \cdot r_2 \cdot (a_1 \cdot a_2)^h \tag{37}$$

If (35) or (37) holds, then Alice proved to IC that she knows her $PrK = x$ satisfying (26) without revealing its value. Consequently, Alice proved that transactions Tx1 and Tx2 belong to her and that she is an investor. This scheme is summarized in Fig. 2.

Schnorr-based NIZKP is standardized by the RFC 8235 document for both finite fields and elliptic curves. It is secure under the DL assumption in the random oracle model.

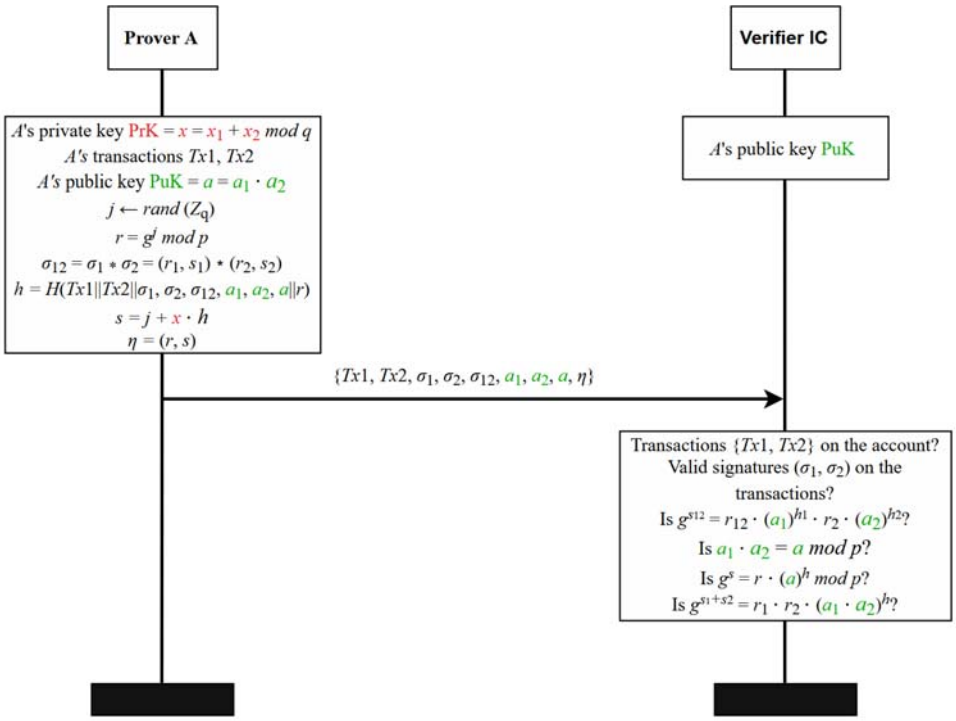

Fig. 2. Deanonymization scheme in our proposal.

4. Example

Let us consider the case when Alice has an account address $Addr$ created according to (17). Then, Alice created two account addresses $\{Addr_1, Addr_2\}$ with corresponding private and public keys to ensure anonymity, as it is depicted in Fig. 3.

According to the Unspent Transactions Output (UTxO) paradigm, Tx1 is realised in $Addr$, with two cryptocurrency incomes $m_1 = 2000$ and $m_2 = 3000$. Hence, the total sum of incomes is $m_{12} = 5000$. The same sum $m_{34} = 5000$ is spent, and is denoted by expenses $m_3 = 1000$ and $m_4 = 4000$. The amount m_3 is transferred to Emily (E), whereas the amount m_4, representing expenses, is transferred by Alice to her anonymous address $Addr_1$.

Let Alice receive two sums $m_6 = 1000$ and $m_7 = 2000$ in the other anonymous address $Addr_2$ in transaction Tx2. She transfers all the received sum $m_8 = 3000$ to IC by this transaction.

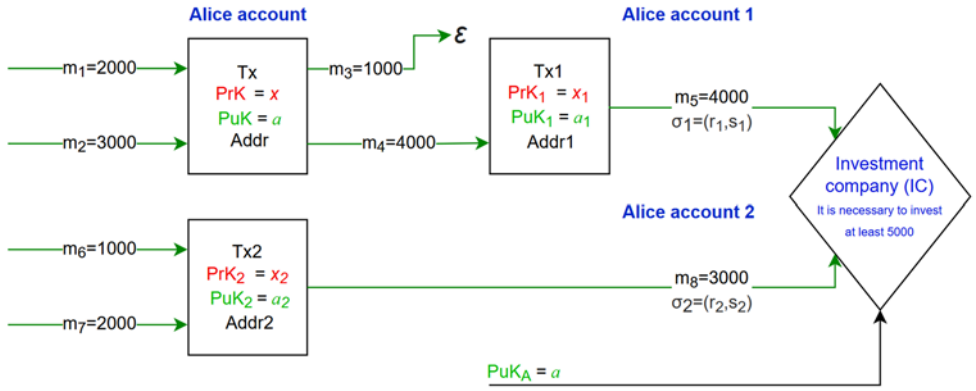

Fig. 3. Anonymization-Deanonymization scheme in a Private Blockchain.

To prove ownership of m_5 and m_8, a combination of Schnorr signatures, Schnorr multi-signatures, and NIZKPs is used to perform the anonymization and deanonymization processes, as presented in Section 3.

5. Security Considerations

The security of anonymization is based on the security of the Schnorr signature scheme.

The security of the Schnorr signature relies on the following two pillars:

- The discrete logarithm (DL) assumption: given a generator g and the public key $a = g^x mod\ p$, no efficient adversary can gain an advantage in obtaining the private key x.
- Fiat-Shamir and random oracle assumptions: the Schnorr signature is obtained from the Fiat-Shamir transform of the Schnorr sigma identification protocol, and uses a collision-resistant H-function, which acts as a random oracle to ensure protection against existential forgeries.

As usual, we understand the adversary's advantage as a probability to improve the random guess of a discrete logarithm value, i.e.

$$AdvDL(g,a,p) = \left|\Pr(\hat{x} = x) - \frac{1}{q}\right|$$

When we say that a certain cryptographic primitive is secure under the DL assumption, we mean that the advantage $AdvDL(g,a,p)$ is negligible, i.e., for any fixed $d > 0$, the advantage $AdvDL(g,a,p)$ tends to zero faster than n^d tends to infinity, where n is the size of q in bits, after a certain n_0, i.e. $\lim_{n\to+\infty}(AdvDL(g,a,p)\cdot n^d) = 0$.

An effective adversary that forges Schnorr signatures can also efficiently compute discrete logarithms in large groups. In other words, any adversary who can efficiently forge a Schnorr signature must have a non-negligible advantage $AdvDL(g,a,p)$. Therefore, this adversary can also be used to solve for x in (16). According to Shor's results [9], such adversaries are possible on quantum computers. Hence, existing cryptographic methods should be replaced by post-quantum cryptographic (PQC) methods in the near future [10].

Therefore, the security of the Schnorr signature scheme against cryptanalytic attacks implemented by classical computers is based on the difficulty of the discrete logarithm problem (DLP) and on the H-function modelled as a random oracle [8]. In real life, random oracles do not exist, and the cryptographic primitives constructed in this model cannot be used directly. Therefore, in practice, the requirement to model H-function as a random oracle is replaced with collision-resistant H-functions.

The security of deanonymization relies on the security of the Schnorr multi-signature scheme and Schnorr-based NIZKP of knowledge of Alice $PrK = x$ satisfying (19), (26), (27).

However, in a multi-user environment, the rogue-key attack can be arranged [12]. In this case, the possible scenario is the following. Anyone on the Net can compute a rogue-key and create a Schnorr multi-signature without using the signatures of other signers. But this attack is prevented according to the results presented in [12]. Referring to [12], the rogue-key attack is naturally prevented in our case, since in the verification equation (34), the public keys a_1 and a_2 are raised by different exponents h_1 and h_2, respectively. Moreover, in the alternative scenario, it is senseless for Alice to act against herself.

The security of Schnorr-based NIZKP based on (32) also relies on the discrete logarithm (DL) assumption, the Fiat-Shamir heuristics, and the random oracle model. This means that an efficient adversary who can impersonate a legitimate user must possess the corresponding secret key and is therefore able to efficiently solve the discrete logarithm problem (DLP) in large groups. As mentioned above, the group $\boldsymbol{Z_p^*}$ and the subgroup $\boldsymbol{G_q}$ are large. Moreover, the same adversary can also efficiently recover x in (19) if they gain access to all but one of the secret values x_i. However, the semiring $\boldsymbol{Z_q}$ is an additive group with the addition operation mod q. Therefore, it preserves the uniform distribution of the sum, given that all the summands are chosen independently and uniformly at

random. This means that the probability of guessing the PrK in (19) is independent of the number of created anonymous accounts N, used for deanonymization and is negligible under the DL assumption. Obviously, the same holds for (32) as well.

Also, the identity of a suspicious user can be restored using the special soundness property of the Schnorr sigma identification protocol in [11]: given two accepting conversations for the statement a_i the IC can compute the value of the witness x_i corresponding to it. Therefore, the IC can identify a dishonest user [11]. This property is preserved in an NIKZP adaptation.

Moreover, due to the soundness property of the Schnorr e-signature, the verifier accepts a false conversation with a negligible probability. Any adversary that can make the verifier accept a false conversation with non-negligible probability can also be used to solve DLP.

Finally, the direct attack is to recover the private key x in $\boldsymbol{Z_q}$ and to perform NIZKP in (32). However, due to the DL assumption, the adversary's advantage $AdvDL(g, a, p)$ is negligible if q is greater than 2^{1024}.

6. Conclusions

The proposed anonymization-deanonymization system is based on a unified cryptosystem, usually named the ElGamal cryptosystem, and therefore uses the general public parameters.

The alternative method is based on ECC ring signatures, requiring more ECC "exponentiation" operations to sign transactions since Alice must sign transactions for all ring members. In this paper, we are using a unified approach based on the integration of Schnorr signature, Schnorr multi-signature and Schnorr-based Non-Interactive Zero Knowledge Proof (NIZKP) to create anonymization, de-anonymization system in a private blockchain under the recognized security assumptions.

The proposed solution provides much more anonymity as compared with the anonymity provided in Ring Signatures (RS) based on the Elliptic Curve Cryptography (ECC) method used in Monero blockchain.

We estimate the effectiveness of realization between RS and our approach by comparing the number N_{ECO} in (14) of EC point multiplication operations to the number of exponentiation operations used in our proposal, based on Schnorr methods, which we denote by N_{EO}.

In our proposal, the effectiveness of anonymization is significantly higher than in the case of RS. In RS schemes, the signer must account for the fact that the number of operations N_{ECO} grows cubically with the number of users n in the ring, as shown in (14).

In contrast, in our construction, the anonymization is linearly dependent on the number of created anonymity addresses. For a single anonymous address, only two exponentiation operations are required. Moreover, anonymization can be performed offline at any time.

For Schnorr multi-signature, no exponentiation operations are required.

For Schnorr-based NIZKP, only two exponentiation operations are required.

The proposed system achieves significantly higher anonymity than RS-based schemes. In the latter case, when the ring consists of n members, the probability of guessing the transaction creator is $\frac{1}{n}$. In our proposal, this probability is equal to $\frac{1}{NoU}$, where NoU stands for the number of users.

This paper is an extension of a B2C' 2025 conference report and proceedings [13].

References

[1]. D. J. Bernstein, T. Lange, Faster addition and doubling on elliptic curves, in Advances in Cryptology - ASIACRYPT 2007, Lecture Notes in Computer Science, Vol. 4833, *Springer*, 2007, pp. 29–50.

[2]. D. Hankerson, A. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, *Springer*, 2004.

[3]. G. Maxwell, A. Poelstra, Borromean ring signatures, Technical Report, *Blockstream*, 2015. Available at: https://github.com/Blockstream/borromean_paper/raw/master/borromean_draft_0.01_34241bb.pdf (Accessed: 20 March 2026).

[4]. M. M. T. Chakravarty, J. Chapman, K. MacKenzie, O. Melkonian, et al., The extended UTXO model, in Financial Cryptography and Data Security: FC 2020 International Workshops, Lecture Notes in Computer Science, Vol. 12063, *Springer*, 2020, pp. 525–539.

[5]. R. L. Rivest, A. Shamir, Y. Tauman, How to leak a secret, in Advances in Cryptology — ASIACRYPT 2001, Lecture Notes in Computer Science, Vol. 2248, *Springer*, 2001, pp. 552–565.

[6]. C. P. Schnorr, Efficient signature generation by smart cards, *Journal of Cryptology*, Vol. 4, Issue 3, 1991, pp. 161–174.

[7]. S. Micali, K. Ohta, L. Reyzin, Accountable-subgroup multisignatures, in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, *ACM*, 2001, pp. 245–254.

[8]. J. Katz, Y. Lindell, Introduction to Modern Cryptography, 2nd ed., *Chapman and Hall/CRC*, 2015.

[9]. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, *IEEE*, 1994, pp. 124–134.

[10]. Penta Security Inc., Post-quantum cryptography: A new security paradigm for the post-quantum era, *Penta Security Blog*, 5 June 2025. Available at: https://www.pentasecurity.com/blog/security-issue-post-quantum-cryptography/ (Accessed: 20 March 2026).

[11]. D. Boneh, V. Shoup, A Graduate Course in Applied Cryptography, Draft (Version 0.6), 2023. Available at: https://toc.cryptobook.us/book.pdf (Accessed: 20 March 2026).

[12]. M. Bellare, G. Neven, Multi-signatures in the plain public-key model and a general forking lemma, in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, *ACM*, 2006, pp. 390–399.

[13]. E. Sakalauskas, A. Kilčiauskas, A. Bendoraitis, A. Michalkovič, et al., Anonymization-deanonymization System in Private Blockchain, in *Proceedings of the 4th Blockchain and Cryptocurrency Conference (B2C' 2025)*, Innsbruck, Austria, 25–27 November 2025, pp. 6-10.