Fair-Anonymity: A Novel Fairness Notion for Cryptocurrency

Taishi Higuchi and Akira Otsuka

Institute of Information Security (IISEC), Kanagawa, Japan

Abstract. In recent years, there has been a growing demand for using tokens of public blockchains like Bitcoin for legitimate transactions. However, the lack of authoritative guarantees on these tokens raises concerns about their potential misuse in criminal activities. Conversely, the introduction of full transparency regulation may stifle the highly innovative cryptocurrency community. This paper introduces a novel concept of fairness, termed Fair-Anonymity, which allows regulatory authorities to probabilistically trace the payer's ID with the pre-agreed probability based solely on the total amount of the transaction, even when divided into smaller transactions. The Fair-Anonymity protocol can be applied to many blockchains by adding proof to the transaction, in which public verifiers can verify the result. Our scheme cryptographically enforces the revealing probability using k-out-of-n Committed Oblivious Transfer, ensuring that neither the sender nor the receiver can manipulate the probability or alter the committed values, thus disincentivizing illegal high-value transactions. Conversely, enterprises accepting only tokens with Fair-Anonymity proofs can externally demonstrate their commitment to lawful operations.

Keywords: Blockchain, Security, Electronic-cash, Cryptocurrency, Fairness, Anonymity, Traceability, Oblivious transfer.

1 Introduction

The concept of anonymous electronic cash (e-cash) was first introduced by Chaum [6], who proposed using blind signatures to enable user anonymity while preventing double-spending. Since then, the coexistence of anonymity and accountability (traceability) has been recognized as a fundamental challenge in e-cash schemes. Subsequent research has explored various approaches to balancing these two properties. Brickell et al. [3] introduced the concept of trustee-based tracing, where a trusted third party (the trustee) can trace transactions when necessary while preserving user anonymity under normal circumstances. They proposed two types of tracing: owner tracing, which allows the trustee to identify the owner of a specific coin, and coin tracing, which reveals the transaction history of a given coin. To prevent misuse, their system uses a distributed trust model with multiple trustees. Building on this work, Camenisch et al. [5] developed an endorsed e-cash system that enables fair exchange of e-cash for goods in both online and offline settings. However, their online scheme has limitations regarding user anonymity due to transaction linkability. Belenkiy et al. [1] introduced an e-cash scheme employing a trusted third party called the "judge", who retrieves the identity of the defrauder after detection of a double-spending. In recent years, various approaches have been proposed to address the coexistence of anonymity and accountability [10, 22].

In attempts to achieve both anonymity and traceability (or accountability) by authorities, particularly in prior research closely related to our approach [22], a method is employed where a limit is set on the amount of transfers per month, and anonymity is nullified if this limit is exceeded. However, this does not eliminate the possibility that users, especially malicious ones, could form groups and make anonymous transfers up to the threshold times the number of users by making transfers slightly below the threshold amount.

David C. Wyld et al. (Eds): SIGML, CNSA, MoWiN, CCSEIT, NIAI, AIAP – 2025
pp. 51-67, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2024.150203

1.1 Challenges and Our Solution

We propose a novel scheme named "Fair-Anonymity", which enables authorities to probabilistically trace IDs based on the value of a paid coin of an e-cash system. In our Fair-Anonymity scheme, the probability of anonymity being nullified increases as the transfer amount approaches the threshold, rendering the aforementioned evasion techniques ineffective. Furthermore, even if coins (or transactions) are split, the probability function can be designed so that the sum of probabilities remains consistent with the original transaction, preserving this key property. Fair-Anonymity is also theoretically applicable to any blockchain tokens such as Bitcoin and Ethereum, in which the transfer amount in a transaction is regarded as a coin. The transcript of each execution of the protocol leaves a cryptographic trace on the blockchain, serving as an indelible credential of a legitimate transaction. It provides a flexible tracing mechanism where authorities can trace users with a pre-agreed probability that is automatically determined based on the transaction amount. This probability cannot be manipulated, as the final probability depends solely on the transferred amount, remaining invariant even when transactions are split into smaller denominations. This enables a balance between regulatory requirements, such as anti-money laundering and taxation, and the protection of user privacy.

In the Fair-Anonymity scheme, users can participate by registering their identity with the authorities and obtaining an ID issued by them. Our Fair-Anonymity system supports the decentralization of authorities through cryptographic techniques involving multiple organizations, thereby reducing dependence on a single third party. To construct the Fair-Anonymity scheme, we introduced a new k-out-of-n Committed Oblivious Transfer as a variant of the efficient 2-round OT_n^k protocol proposed by Lai et al. [19], enabling practical performance.

The probabilistic nature of our Fair-Anonymity protocol effectively disincentivizes high-value illegal transactions without compromising the convenience of honest users. Criminals, even with a small probability, are deterred by the risk of their identities being traced by authorities, whereas legitimate users remain largely unaffected by occasional transaction tracking.

1.2 Our Contribution

Our research contributions are as follows:

1. Fairness in Transaction Amounts:
   We introduce a novel concept of "Fairness" as a property where the probability of revealing a user's (payer's) identity is determined solely by the total transaction amount, regardless of how the transaction is divided. This ensures that the probability remains unaffected by multiple low-value transactions, thereby mitigating split attacks. Fairness can be incorporated into existing transactions as accompanying proof information, making it adaptable to various blockchains. The accompanying proof with the transaction can be publicly verified, while only the designated authority can trace the user's identity based on a probability determined by the total transaction amount.
2. Achieving Perfect Fairness with Exponential Saturation Functions:
   We demonstrate that by utilizing Exponential Saturation Functions (ESFs), we can achieve "Perfect Fairness," where identity disclosure is determined exclusively by the total transaction amount. In constructing Fair-Anonymity, the ESF parameters must be approximated with integers, introducing an error ϵ. We identify a trade-off between minimizing this error and reducing the proof size.

3. Modification of Committed Oblivious Transfer (COT):
   We propose improvements to the original efficient k-out-of-n Oblivious Transfer (COT) protocol [19]. Our modification enables public verification of the correctness of the message without revealing itself, while ensuring that the probability determined by the transaction amount cannot be manipulated.

With this "fair" probabilistic ID tracing scheme, we achieve the following outcomes: (1) Malicious users will fear that even a single illicit transaction could be detected, leading to their identification by authorities. (2) Honest users, on the other hand, will not be negatively impacted by probabilistic tracing, as not all of their transactions will be monitored or linked.

In scenarios where payers use this protocol over a long period, the probability of tracing their ID increases due to the accumulation of transaction amounts. However, this can be mitigated by periodically refreshing the scheme, as described in [22]. This creates a situation where malicious users are disincentivized from using Fair-Anonymity for transactions, while honest users are encouraged to adopt this scheme. Consequently, Fair-Anonymity can establish a crime-free, transparent economic zone, clearly separated from the illicit economic sphere.

2 k-out-of-n Committed Oblivious Transfer

Oblivious Transfer (OT) is a crucial component in constructing protocols that are secure and protect privacy, such as contract signing, private information retrieval, and secure function evaluation. An OT scheme is a two-party protocol between a sender S and a receiver R, where the sender possesses multiple secrets, and the receiver wishes to select and obtain some of them. The receiver acquires the secrets without revealing their choice to the sender, and the sender remains unaware of which secrets the receiver obtained. The first OT scheme was proposed by Rabin [21], and since then, more general forms such as 1-out-of-2 OT (OT_2^1), 1-out-of-n OT(OT_n^1), and general k-out-of-n OT (OT_n^k) have been introduced [20, 8, 17, 16]. Committed Oblivious Transfer (COT) is an extension of OT that additionally involves commitments. In COT, the sender is committed to the input messages and the receiver is committed to the choice index before the OT protocol is executed. The COT protocol provides additional security guarantees compared to plain OT. The commitments prevent the sender from changing the messages and the receiver from changing the choice index during the protocol. We introduce the new schemes of k-out-of-n Committed Oblivious Transfer (COT_n^k) with a small modification on the efficient 2-round OT_n^k protocol proposed by Lai et al. [19].

2.1 Definitions of k-out-of-n Committed Oblivious Scheme

In this paper, we let $\mathbb{G}$ be a multiplicative cyclic group of prime order p. We say $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an admissible bilinear map.

Suppose that both a sender S and a receiver R engage in the COT_n^k protocol defined below. A set of messages $\boldsymbol{M} = m_1, \cdots, m_n$ is held by the sender, and a set of indices $\boldsymbol{G} = l_1, \cdots, l_k$, where $k \leq n$, is chosen by the receiver.

Definition 1 (k-out-of-n Committed Oblivious Transfer (COT_n^k)). *The COT_n^k protocol is executed between the parties S and R. Initially, there exist public commitments $\mathrm{Com}_S(m_1, r_1), \ldots, \mathrm{Com}_S(m_n, r_n)$, and $\mathrm{Com}_R(l_1, s_1), \ldots, \mathrm{Com}_R(l_k, s_k)$. S inputs $m_1, \ldots, m_n$ and $r_1, \ldots, r_n$, while R inputs $l_1, \ldots, l_k$ and $s_1, \ldots, s_k$. At the end of the protocol, R receives*

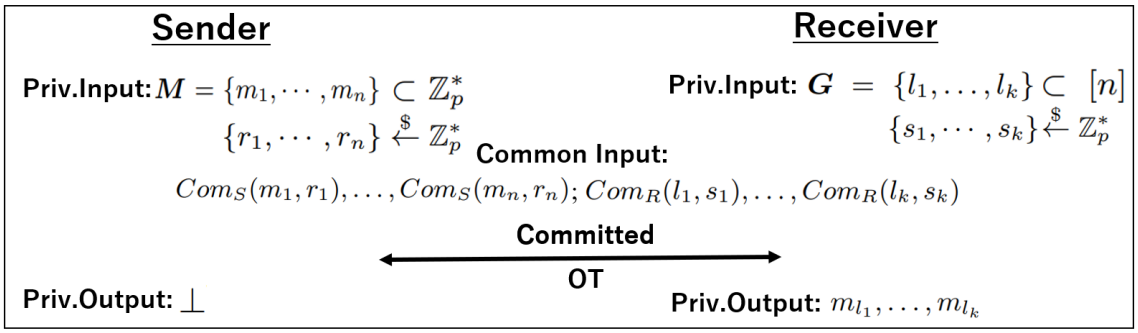

Fig. 1: Overview of k-out-of-n Commited Oblivious Transfer

$m_{l_1}, \ldots, m_{l_k}$. *S learns nothing about* $l_1, \ldots, l_k$ *while R remains unaware of the unchosen messages.*

An overview of COT_n^k is shown in Fig. 1. In order to construct our proposed Fair-Anonymity protocol (in Section 4), we slightly modify the k-out-of-n COT and define the k-out-of-n Committed Oblivious Transfer of Commitments ($\mathrm{C^2OT}_n^k$). In $\mathrm{C^2OT}_n^k$, the sender selects messages $m_1, \ldots, m_n$ in plaintext, and the receiver receives the chosen messages in their encrypted form, $Enc(m_{l_1}), \ldots, Enc(m_{l_k})$.

Definition 2 (**k-out-of-n Committed Oblivious Transfer of Commitments** ($\mathrm{C^2OT}_n^k$)**).** *The* $\mathrm{C^2OT}_n^k$ *protocol is executed between the parties S and R. Initially, both parties take a generator* g_c*, along with the commitments* $\mathrm{Com}_S(m_1, r_1), \ldots, \mathrm{Com}_S(m_n, r_n)$*, and* $\mathrm{Com}_R(l_1, s_1), \ldots, \mathrm{Com}_R(l_k, s_k)$ *as public inputs. S inputs* $m_1, \ldots, m_n$ *and* $r_1, \ldots, r_n$*, while R inputs* $l_1, \ldots, l_k$ *and* $s_1, \ldots, s_k$*. At the end of the protocol, R receives* ${g_c}^{m_{l_1}}, \ldots, {g_c}^{m_{l_k}}$*. S learns nothing about* $l_1, \ldots, l_k$*, while R remains unaware of the unchosen messages.*

The modification makes the original COT_n^k verifiable by allowing the receiver to confirm that the original messages corresponding to the ciphertexts received by him are indeed the ones sent by the sender. This enables a public verifier in the Fair-Anonymity protocol to verify that the received ciphertexts correspond to the original messages without directly revealing the messages themselves.

2.2 Construction of k-out-of-n Committed Oblivious Scheme

The concrete COT_n^k scheme is illustrated in Fig.2. A trusted third party establishes the system by choosing a security parameter λ and a random value α as the system's secret key. It then generates the system parameter $SP = (\mathbb{BG}, g, h, g_1, \cdots, g_n, h_1, \cdots, h_n)$, where $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ and $g, g_i, h, h_i \in \mathbb{G}$, which are publicly known. First, the receiver selects a random value $s \in \mathbb{Z}_p^*$ as its secret key and a set $\boldsymbol{G} = \{l_1, l_2, \cdots, l_k\}$. It then uses the Aggregation algorithm to compute a token $\mathrm{P}(\boldsymbol{G}) = g^{\frac{s}{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)}}$, and computes a proof information $\Sigma = h^{\frac{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)\cdot\alpha^{n-k}}{s}}$ for its choice set $\boldsymbol{G}$. Next, after receiving a request from the receiver, the sender first checks whether $e(\mathrm{P}(\boldsymbol{G}), \Sigma) = e\left(g, h^{\alpha^{n-k}}\right)$. If the check fails, the protocol aborts. Otherwise, the sender selects a random value $r \in \mathbb{Z}_p^*$ and computes a ciphertext CT for the secrets as $\tilde{c}_0 = \mathrm{P}(\boldsymbol{G})^r = g^{\frac{rs}{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)}}$, along with, for each $i = 1, 2, \cdots, n : \tilde{c}_i = e\left(g^{\frac{1}{\alpha+i}}, h\right)^r \cdot m_i$. After receiving the encrypted secrets

CT from the sender, the receiver computes, for each $i \in \boldsymbol{G}$, $m_i = \tilde{c}_i \cdot e\left(\tilde{c}_0, h^{\frac{(\alpha+l_1)\cdots(\alpha+l_n)}{(\alpha+i)}}\right)$. After decryption, the receiver obtains only k secrets with indices in $\boldsymbol{G}$ from the sender. Finally, the receiver outputs new commitments, allowing verification of all received messages using membership proofs such as One-out-of-Many proofs (see Appendix A.1).

As stated in the definition, when modifying the COT_n^k scheme into the $\mathrm{C^2OT}_n^k$ scheme, the message m_i received by the receiver is transformed into an encrypted form, such as $m_i \to g_T^{m_i}$ where $g_T \in \mathcal{G}_T$. This ensures that the receiver obtains the messages in their encrypted form. However, as this modification only requires replacing the messages with their ciphertexts and does not involve significant changes to the overall construction, we omit the explanation here (This explanation will be provided when presenting the construction of the Fair-Anonymity protocol in Section 4).

3 Fairness

In this section, we propose a novel concept of *Fairness* and demonstrate that a specific function satisfies the definition of this fairness property. This property is crucial, as a Fair-Anonymity scheme based on a probability distribution function satisfying this property allows authorities to identify the sender's ID with a probability that depends solely on the transfer amount, irrespective of how the coins are divided (transaction splitting).

3.1 Definition of Fairness

Let $\boldsymbol{C}$ and $\boldsymbol{U}$ represent the sets of all coins and all users, respectively. For any coin $c \in \boldsymbol{C}$, let $c.v$ denote the value of the coin. A payment system is considered *fair* if the following condition holds.

Definition 3 (ϵ-Fairness). *Suppose we have a probabilistic extractor $\mathcal{E} : \boldsymbol{C} \to \boldsymbol{U} \cup \perp$ that extracts from any coin c the spender's user ID $u \in \boldsymbol{U}$ when successful, or returns nothing ($\perp$) otherwise.*

Then, for all $\epsilon \in \mathbb{R}_{\geq 0}$, we say that a payment system satisfies ϵ-`fairness` if for all coins $c_1, c_2, c_3 \in \boldsymbol{C}$ the following holds:

$$c_1.v = c_2.v + c_3.v \Rightarrow |\Pr[\mathcal{E}(c_1) \neq \perp] - \Pr[\mathcal{E}(c_2) \neq \perp \vee \mathcal{E}(c_3) \neq \perp]| \leq \epsilon \tag{6}$$

When $\epsilon = 0$, we say that a payment system satisfies perfect fairness.

The intuition of this definition is the following. Any division of coins or payments does not affect the probabilistic traceability with at most ϵ fluctuation. ϵ-Fairness ensures that the process of identifying the spender from a coin is not affected by the division of coins, within a tolerance level represented by ϵ. The smaller the value of ϵ, the fairer the system is considered.

3.2 Exponential Saturation Function

Next, we introduce the exponential saturation function which satisfies the condition of the fairness property. The form of the function is

$$p(x) = 1 - e^{-\frac{x}{K}} \tag{7}$$

where $K \in \mathbb{R}$ is a rate constant.

– **Inputs:**
  - A trusted third party runs the Setup algorithm as follows. Given a security parameter λ, this algorithm generates a bilinear group $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ with two generators $g, h \in \mathbb{G}$. Then it randomly chooses $\alpha \in \mathbb{Z}_p^*$ as the system secret key and computes $g_i = g^{\frac{1}{\alpha+i}}, h_i = h^{\alpha^i}$ for all $i \in [n]$. A system parameter SP consists of $(\mathbb{BG}, g, h, g_1, g_2, \cdots, g_n, h_1, h_2, \cdots, h_n)$.
  - S holds a set of secrets $\boldsymbol{M} = \{m_1, \cdots, m_n\} \subset \mathbb{Z}_p^*$ and random numbers $\{r_1, \cdots, r_n\} \xleftarrow{\$} \mathbb{Z}_p^*$.
  - R holds his choice set $\boldsymbol{G} = \{l_1, \ldots, l_k\} \subset [n]$ and random numbers $\{s_1, \cdots, s_k\} \in \mathbb{Z}_p^*$.
  - Input Common Commitments [a]:
    $Com_S(m_1, r_1), \ldots, Com_S(m_n, r_n)$; $Com_R(l_1, s_1), \ldots, Com_R(l_k, s_k)$.

– **Execute OT_n^k Protocol:**

1. $R \to S$: Given a choice set $\boldsymbol{G} = \{l_1, \cdots, l_k\}$ and SP, R picks a random $s \in \mathbb{Z}_p^*$ as his secret key sk and uses the Aggregation algorithm [11] to compute $\mathrm{P}(\boldsymbol{G})$ together with Σ where

$$\mathrm{P}(\boldsymbol{G}) = g^{\frac{s}{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)}}, \tag{1}$$

$$\Sigma = h^{\frac{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)\cdot\alpha^{n-k}}{s}}. \tag{2}$$

2. $S \to R$: S runs the Encrypt algorithm and generates commitments as follows. Given a set $T = (\mathrm{P}(\boldsymbol{G}), \Sigma)$, a set of secrets $\boldsymbol{M} = \{m_1, \cdots, m_n\}$ and the system parameter SP, it first performs the verification algorithm as: $e(\mathrm{P}(\boldsymbol{G}), \Sigma) = e\left(g, h^{\alpha^{n-k}}\right)$. If the equation does not hold, it aborts. Otherwise, it accepts $|\boldsymbol{G}| = k$. Then for a random parameter $r \xleftarrow{\$} \mathbb{Z}_p^*$, it computes the ciphertext set $CT = \{\tilde{c}_i\}_{i=1,\ldots,n}$ for the messages as

$$\tilde{c}_0 = \mathrm{P}(\boldsymbol{G})^r = g^{\frac{rs}{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)}} \tag{3}$$

   together with, for each $i \in [n]$:

$$\tilde{c}_i = e\left(g^{\frac{1}{\alpha+i}}, h\right)^r \cdot m_i \tag{4}$$

3. R decrypts the received ciphertexts as follows. Given the ciphertext $CT = \{\tilde{c}_0, \tilde{c}_1, \cdots, \tilde{c}_n\}$, a choice set $\boldsymbol{G} = \{l_1, \cdots, l_k\}$, a secret key s and the system parameter SP, only for each $i \in \boldsymbol{G}$, R can compute

$$m_i = \tilde{c}_i \cdot e\left(\tilde{c}_0, h^{\frac{(\alpha+l_1)(\alpha+l_2)\cdots(\alpha+l_k)}{(\alpha+i)}}\right)^{-\frac{1}{s}} \tag{5}$$

– **Outputs and Verification:** R verifies that the values of the received messages are the same as the values of committed ones.

[a] S and R should perform the PoK for S : $((m_i, r_i); Com_S(m_i, r_i))$ for all $i \in [n]$ and the PoK for R : $((l_i, s_i); Com_R(l_i, s_i))$ for all $i \in [k]$ respectively

Fig. 2: The construction of COT_n^k scheme.

Lemma 1. *Suppose we have a payment system with a probabilistic extractor* $\mathcal{E} : \boldsymbol{C} \to \boldsymbol{U}$ *or* $\perp$ *as in Definition 3 such that its probability is determined by exponential saturation function of the input coin value. More concretely, we define a probability function* $p : \mathbb{R}_{\geq 0} \to [0, 1]$ *as*

$$p(c.v) = \Pr[\mathcal{E}(c) \neq \perp] = 1 - e^{-\frac{c.v}{K}} \tag{8}$$

for some rate constant $K \in \mathbb{R}_{\geq 0}$.

Then, for any $n > 0$ *division of coin* c *to* $c_1, \ldots, c_n$ *such that* $c.v = c_1.v + \cdots + c_n.v$, *the payment system satisfies perfect fairness for all* $c.v, K \in \mathbb{R}_{\geq 0}$.

proof By using probability of complementary event that $p(X \cup Y) = 1-(1-p(X))(1-p(Y))$ for independent variables X, Y. We have

$$\begin{aligned} p(c.v) = 1 - e^{-\frac{c.v}{K}} &= 1 - \prod_{i=1}^{n} (1 - p(c_i.v)) \\ \Leftrightarrow 1 - p(c.v) &= \prod_{i=1}^{n} (1 - p(c_i.v)) \\ \log(1 - p(c.v)) &= \sum_{i=1}^{n} \log (1 - p(c_i.v)) \end{aligned} \tag{9}$$

By substituing p with exponential sturation function as:

$$\begin{aligned} p(c_i.v) = 1-e^{-\frac{c_i.v}{K}}, \quad p(c.v) &= 1 - e^{-\frac{c.v}{K}} \\ \log(e^{-\frac{c.v}{K}}) &= \sum_{i=1}^{n} \log \left(e^{-\frac{c_i.v}{K}}\right) \\ \therefore c.v &= \sum_{i=1}^{n} c_i.v \end{aligned} \tag{10}$$

4 Fair-Anonymity

In this section, we propose the Fair-Anonymity scheme constructed by combining our construction of COT_n^k with the exponential saturation function satisfying the fairness property.

4.1 Overview of Fair-Anonymity Protocols

The Fair-Anonymity protocol consists of System Setup, Registration Protocol, Execution Protocol, and Tracing Protocol.

- **System Setup:** First, in the setup phase, the authority generates a set of IDs using a salt known only to itself. Here, the authority can be a single organization or a decentralized system that requires consensus among multiple organizations using cryptographic techniques. Since the set of IDs is generated using a salt known only to the authority, third parties cannot compute them.
- **Registration Protocol:** Users who want to use the Fair-Anonymity protocol interact with the authority to prove their identity and obtain an ID. This is equivalent to opening a bank account.

- **Execution Protocol:** In the execution protocol, the user interacts online with a verifier (on the blockchain). The authority is offline and not involved in this interaction. First, the user and the verifier execute COT, where the user is the sender and the verifier is the receiver. The user selects a set of messages consisting of either the encryption of their ID or random values, and keeps them secret. The ratio of the ID's ciphertext to random values is specified by a probability distribution function that satisfies fairness, such as an exponential saturation function (introduced in Section 3) with the total amount as a variable (if the transfer amount is large, the ratio of the ID's ciphertext increases according to the probability function, which is an increasing function for x $\geq$ 0). The verifier selects k labels to receive. Then, the user sends k out of the n messages specified by the verifier using COT. Due to the properties of COT, neither the message values nor the specified label values can be changed from the initially chosen ones. If there are no issues during the protocol execution, the verifier publishes the received values (ID ciphertext or random values) as part of the transcript on the blockchain with a signature. We emphasize that the verifier and those who can see the transcript can only know the values received by the verifier, but cannot distinguish whether the values are ciphertexts or mere random values, thus preserving the user's anonymity.
- **Tracing Protocol:** In the tracing protocol, the authority can trace a specific transcript on the blockchain at any time. Since a specific transfer transaction using the Fair-Anonymity protocol includes the sender's ID ciphertext with the pre-agreed probability according to the transfer amount, the authority can identify the user by searching a narrow set of IDs using the salt known only to the authority (if it is an ID ciphertext and not a random value). Those who do not know the salt must perform an exhaustive search on the set $\mathbb{Z}_p^*$, and if the set is sufficiently large, anonymity is not compromised.

4.2 System Setup

Firstly, Authority $\mathcal{A}$ generates a subset $\boldsymbol{U} = \{u_i\}_{i=1,\ldots,N} \subset \mathbb{Z}_p^*$, using a value of salt known only to $\mathcal{A}$. Assuming the DDH assumption holds, it is impossible to calculate the user ID from g^u for some group g for those who do not know the salt, while $\mathcal{A}$, knowing the salt, can perform a brute-force search in the $\boldsymbol{U}$ space [1]. Next, $\mathcal{A}$ generates a set of Pedersen commitments $\boldsymbol{A} = \{Com_A(u_i, s_i)\}_{i=1,\ldots,N}$ from the set of user IDs $\boldsymbol{U}$ as follows:

$$Com_A(u_i, s_i) = g_T^{u_i} h_T^{s_i}, \text{ where } \quad g_T, h_T \quad \in \mathbb{G}_T \quad \text{and} \quad s_i \xleftarrow{\$} \mathbb{Z}_p^* \tag{11}$$

This set of commitments enables users, in the Fair-Anonymity protocol to be constructed next, to provide a zero-knowledge proof that they know their ID is in the correct ID set.

4.3 Registration Protocol

The Registration protocol is executed between $\mathcal{A}$ and User $\mathcal{U}$ interactively. First, $\mathcal{U}$ shows his certificate of identity to $\mathcal{A}$. $\mathcal{A}$ verifies it. After the verification, $\mathcal{A}$ picks up an ID $u \leftarrow \boldsymbol{U}$ for him and finds the corresponding commitment $Com_A(u_\rho, s_\rho)$ of $u = u_\rho$ at the ρ-th position in the entire set of $\boldsymbol{A}$. Then, $\mathcal{A}$ sends $\mathcal{U}$ the triple of (u, ρ, s_ρ).

4.4 Execution Protocol

Next, we give the protocol of executing Fair-Anonymity between a user $\mathcal{U}$ and a public verifier $\mathcal{V}$. The overview of this protocol is shown in Fig. 3. For simplicity, the protocol is

[1] Note that the size of the set $\boldsymbol{U}$ is $|\boldsymbol{U}| \ll p$.

constructed using $\mathrm{C^2OT}_n^k|_{k=1}$. Note that the protocol can similarly be constructed with higher efficiency for the general case where $k > l$. (The value of k is automatically selected based on the transfer amount.)

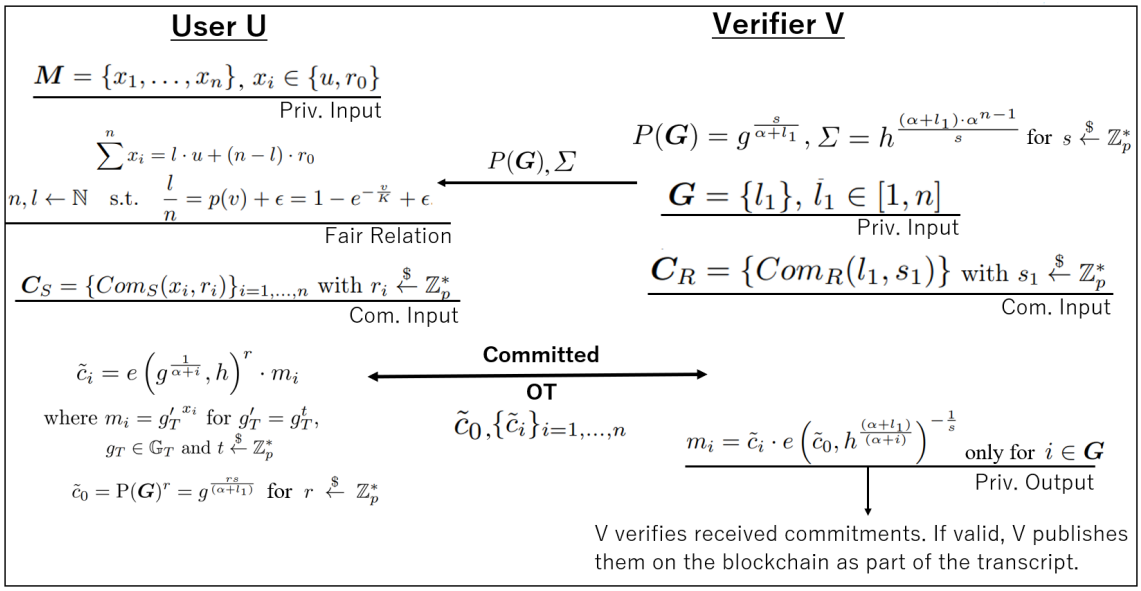

Fig. 3: Overview of Execution Protocol of Fair-Anonymity

Consider $\mathcal{U}$, already registered with an authority $\mathcal{A}$, executes the execution protocol with a public verifier $\mathcal{V}$ for his coin c of value $c.v$.

1. First, the pair of two integers $l, n \in \mathbb{N}$ are determined based on the probability distribution $p(v)$, which depends on the coin value $v = c.v$,

$$n, l \leftarrow \mathbb{N} \quad \text{s.t.} \quad \frac{l}{n} = p(v) + \epsilon = 1 - e^{-\frac{v}{K}} + \epsilon. \tag{12}$$

   where $\epsilon \in \mathbb{R}$ represents the small fluctuation due to approximating real numbers with rational numbers. Note that the probability function is one of the exponential saturation functions that satisfy the fairness property (shown in Section 3).
2. $\mathcal{U}$ and $\mathcal{V}$ start the $\mathrm{C^2OT}_n^1$ protocol, where Sender is $\mathcal{U}$ and Verifier is $\mathcal{V}$ in this case, as follows:
   (a) **Input**: First, $\mathcal{U}$ chooses the set of n messages $\boldsymbol{M} = \{x_1, \ldots, x_n\}$ where $x_i \in \{u, r_0\}$ s.t.

$$\sum_{i=1}^{n} x_i = l \cdot u + (n-l) \cdot r_0 \tag{13}$$

   where $u \in \boldsymbol{U}$ is the $\mathcal{U}$'s ID and $r_0 \in \mathbb{Z}_p^*$ is a randomly chosen number. [2] For his chosen messages, $\mathcal{U}$ calculates the corresponding commitments $\boldsymbol{C}_S = \{Com_S(x_i, r_i)\}_{i=1,\ldots,n}$ with $r_i \xleftarrow{\$} \mathbb{Z}_p^*$
   Next, $\mathcal{V}$ chooses his choice set $\boldsymbol{G} = \{l_1\}$, $l_1 \in [1, n]$ (because we consider now the case $k = 1$) and calculates the commitment $\boldsymbol{C}_R = \{Com_R(l_1, s_1)\}$ with $s_1 \xleftarrow{\$} \mathbb{Z}_p^*$.
   The two sets of the commitments $\boldsymbol{C}_R$ and $\boldsymbol{C}_S$ are common inputs.

[2] We assumed $|\boldsymbol{U}| \ll p$ and hence $\Pr[r_0 \in \boldsymbol{U}] < \mathrm{negl}(\lambda)$.

(b) **Execute**:

– $\mathcal{V} \rightarrow \mathcal{U}$: Given a choice set $\boldsymbol{G} = \{l_1\}$ and the system parameters SP,$\mathcal{V}$ picks a random $s \xleftarrow{\$} \mathbb{Z}_p^*$ as his secret key and uses the Aggregation Algorithm to compute $\mathrm{P}(\boldsymbol{G})$ and Σ where

$$P(\boldsymbol{G}) = g^{\frac{s}{\alpha+l_1}}, \quad \Sigma = h^{\frac{(\alpha+l_1)\cdot\alpha^{n-1}}{s}} \tag{14}$$

$\mathcal{V}$ sends them to $\mathcal{U}$.

– $\mathcal{U} \rightarrow \mathcal{V} : \mathcal{U}$ runs the Encrypt algorithm and generates commitments as follows. Given a set $T = (\mathrm{P}(\boldsymbol{G}), \Sigma)$, a set of secrets $\boldsymbol{M} = \{x_1, \cdots, x_n\}$ and the system parameter SP, it first performs the verification algorithm as: $e(\mathrm{P}(\boldsymbol{G}), \Sigma) = e\left(g, h^{\alpha^{n-k}}\right)$. If the equation does not hold, it aborts. Otherwise, it accepts $|\boldsymbol{G}| = k$. Then for a random parameter $r \xleftarrow{\$} \mathbb{Z}_p^*$, it computes the ciphertext set $\tilde{\boldsymbol{C}} = \{\tilde{c}_i\}_{i=1,\ldots,n}$ for the messages as

$$\tilde{c}_0 = \mathrm{P}(\boldsymbol{G})^r = g^{\frac{rs}{(\alpha+l_1)}}, \quad \tilde{c}_i = e\left(g^{\frac{1}{\alpha+i}}, h\right)^r \cdot m_i \tag{15}$$

where $m_i = {g'_T}^{x_i}$ for $g'_T = g_T^t$, $g_T \in \mathbb{G}_T$ and $t \xleftarrow{\$} \mathbb{Z}_p^*$. Note that in this point the construction of $\mathrm{C}^2\mathrm{OT}_n^k$ protocol is slightly different from that of the COT_n^k protocol, changing the message x_i encrypted by ElGamal Encryption to ${g'_T}^{x_i}$. This enables only $\mathcal{A}$, knowing the secret salt, to trace the user's ID. Using Verifiable Encryption [7] (see Appendix A.2), $\mathcal{U}$ performs the PoK for the $\tilde{c}_i =_{eq} Com_S(x_i)$ for all $i \in [n]$ to $\mathcal{V}$, outputting the proof set $\{\pi_i\}_{i=1,\ldots,n}$ (defined in Section 5).

– $\mathcal{V}$ decrypts the received ciphertexts as follows. Given the ciphertexts $\tilde{\boldsymbol{C}} = \{\tilde{c}_0, \tilde{c}_1, \cdots, \tilde{c}_n\}$, the choice set $\boldsymbol{G} = \{l_1\}$, the secret key s and the system parameter SP, only for the $i \in \boldsymbol{G}$, here just for the l_1, $\mathcal{V}$ can decrypt

$$m_i = \tilde{c}_i \cdot e\left(\tilde{c}_0, h^{\frac{(\alpha+l_1)}{(\alpha+i)}}\right)^{-\frac{1}{s}}. \tag{16}$$

(c) **Verify**:

$\mathcal{V}$ verifies the received proof set $\{\pi_i\}_{i=1,\ldots,n}$. If the verification fails $\mathcal{V}$ aborts.

3. After the execution of $\mathrm{C}^2\mathrm{OT}_n^k$, $\mathcal{U}$ proves to $\mathcal{V}$ that the two relations $\mathscr{R}_1$ and $\mathscr{R}_2$ defined in the following Section 5 hold - for $\mathscr{R}_1$ all of the $\{x_i\}_{i=1,\ldots,n}$ in the commitments $\boldsymbol{C}_S$ are indeed selected from either u or r_0 (see the equation (23)), and for $\mathscr{R}_2$ the equation (24) for the exponential saturation function holds.
4. At the end, if $\mathcal{U}$ clears all verifications, $\mathcal{V}$ publishes the results of those proofs as a transcript on the blockchain. As a result, the $\mathcal{U}$'s coin can be considered "fair" with the transcript.

4.5 Tracing Protocol

The Authority $\mathcal{A}$ can, at any time, refer to the transcripts of users published on the blockchain and combine these with the secret value of salt known only to $\mathcal{A}$, to probabilistically trace the ID of the user who made the coin payment. However, the actual probability of $\mathcal{A}$ being able to identify the user ID is given by $p(v) \pm \epsilon$.

5 Security Notions of Fair-Anonymity

In this section, we define the required security notions of Fair-Anonymity - *Completeness*, *Soundness*, *Anonymity*, and *Fair Traceability*. First, we introduce the relevant lemmas and convenient notations that will be used for the definition and the proof sketch of the security notions. Next, we define two cryptographic relations that Fair-Anonymity must satisfy and the four security notions above. Finally, we prove that the construction of Fair-Anonymity in Section 4 satisfies these security notions.

Lemma 2. *(Membership proof) Take a Pedersen commitment $c = g^m h^r$ with a message m, a randomness r and a set of n commitments $\{c_i\}_{i=1,\dots,n}$ for $c_i = g^{m_i} h^{r_i}$. If a prover knows that the message m of the commitment c corresponds to the message m_l of $c_l \in \{c_i\}$ he can prove this to the verifier without revealing any information about the witness (m, r, l).*

First, we define our unique notation for describing the relations between commitments as follows:

$$c_1 =_\omega c_2 \overset{\text{def}}{\Leftrightarrow} x_1 = x_2 \text{ for } c_1 = g^{x_1} h^{r_1} \text{ and } c_2 = g^{x_2} h^{r_2}, \tag{17}$$

$$c \in_\omega \{c_1, \dots c_n\} \overset{\text{def}}{\Leftrightarrow} \exists\ l \in [n], c =_\omega c_l \in \{c_i\}_{1,\dots,n}. \tag{18}$$

And we define the proof of knowledge for membership π_{mem} as follows:

$$\pi_{mem}(c \in_\omega \{c_i\}_{1,\dots,n}) \overset{\text{def}}{=} \text{PoK}\left(x, r, l; c \in \{c_i\}_{1,\dots,n}\right), \tag{19}$$

$$\pi_m(c_1 =_\omega c_2) \overset{\text{def}}{=} \text{PoK}\left(x_1, r_1, x_2, r_2; c_1 =_\omega c_2\right). \tag{20}$$

Lemma 3. *(Verifiable Encryption [7]) Take a Pedersen commitment $c = \text{Com}(m, r) = g^m h^r$ and an ElGamal ciphertext $\tilde{c} = \mathsf{Enc}(m', r') = g_T^{m'} h_T^{r'}$ for a plaintext m with a randomness $r, r' \in \mathbb{Z}_p^*$. If a prover knows that the message m of the commitment c equals the message m' of the ciphertext $\tilde{c}$ he can prove this to the verifier without revealing any information about the witness (m, r, m', r').*

With the notation "$=_{eq}$" defined as

$$\tilde{c} =_{eq} c \overset{\text{def}}{\Leftrightarrow} m = m' \text{ for } \tilde{c} = g_T^m h_T^r \text{ and } c = g^{m'} h^{r'} \tag{21}$$

for $r, r', m, m' \in \mathbb{Z}_p^*$, $(g, h) \in \mathbb{G}$, and $(g_T, h_T) \in \mathbb{G}_T$, we define the proof of knowledge for the equality π_{eq} as follows:

$$\pi_{eq}(c =_\omega \tilde{c}) \overset{\text{def}}{=} \text{PoK}\left((x, r, r'); c =_\omega \tilde{c}\right) = (v, \tilde{v}, z_1, z_2, z_3) \tag{22}$$

Next, we introduce two relations for the security notions. In the Fair-Anonymity scheme, the probability of an authority $\mathcal{A}$ being able to trace a User's ID is determined by the value of the User's coin. During the one-time System Setup, $\mathcal{A}$ generates the set of commitments for all user IDs, $\boldsymbol{A} = \{Com_A(u_i)\}_{i=1,\dots,N}$. In the Registration protocol, a user $\mathcal{U}$ presents their real identity to $\mathcal{A}$, and then sends $\mathcal{U}$ their user ID u, and the label ρ indicating the position of the commitment corresponding to u in the set $\boldsymbol{A}$ together with r_ρ. In the Execution phase, $\mathcal{U}$ executes probabilistic message transmission to a public verifier $\mathcal{V}$ using COT_n^k (the message being information about u or a uniform random number). In the C^2OT_n^k, $\mathcal{U}$ chooses n messages $\{x_1, \dots, x_n\}$ and calculates the corresponding commitment set $\boldsymbol{C}_S = \{Com_S(x_i)\}_{i=1,\dots,n}$ as common input. Each message is u itself or

a uniform random number r_0. After the execution of $\mathrm{C^2OT}_n^k$, $\mathcal{U}$ proves that the following two relations $\mathscr{R}_1$ and $\mathscr{R}_2$ defined below hold. $\mathscr{R}_1$ is the relation that all $\{x_i\}_{i=1,\ldots,n}$ in the commitments $\boldsymbol{C}_S$ are indeed values of either u or r_0. $\mathscr{R}_2$ denotes the relation for the commitments $\boldsymbol{C}_S$ of n messages $\{x_i\}_{i=1,\ldots,n}$, where l of these messages have values equal to u, and the remaining $n-l$ messages are valued at r_0.

1. **Relation $\mathscr{R}_1$:** For the $\mathcal{U}$'s witness (u, r^*, ρ) and a fresh commitment $c^* = Com_{\mathcal{U}}(u, r^*) = g^u h^{r^*}$, where r^* is a fresh random number, there exists $\rho \in [N]$ such that $Com_{\mathcal{U}}(u, r^*) =_\omega Com_A(m_\rho, r_\rho)$, i.e.,

$$\begin{aligned}(u, r^*, r_\rho, r_\sigma, \rho, \sigma) \in \mathscr{R}_1 \Leftrightarrow \pi \leftarrow PoK\big((u, r^*, r_\rho, r_\sigma, \rho, \sigma); c^* = Com_{\mathcal{U}}(u, r^*) \\ \wedge\, ^\exists\rho \in [N] \ s.t.\ c_\rho = Com_A(u, r_\rho) \\ \wedge\, ^\exists\sigma \in [n] \ s.t.\ c_\sigma = Com_S(u, r_\sigma)\big) \\ \text{is an accepting proof.}\end{aligned} \tag{23}$$

2. **Relation $\mathscr{R}_2$:** In $\mathrm{C^2OT}_n^k$ for the commitments $\boldsymbol{C}_S$ for massages $\{x_1, \ldots, x_n\}$, the condition $\#\{i | c^* = c_i, i \in [n]\} = l (> 0)$ should be satisfied. For $c^* = Com_{\mathcal{U}}(u, r^*)$ and $c^\dagger = Com_{\mathcal{U}}(r_0, r^\dagger)$, $\mathscr{R}_2$ is defined as:

$$\begin{aligned}(u, r^*, r_0, r^\dagger, r_1, \ldots, r_n) \in \mathscr{R}_2 \Leftrightarrow \pi \leftarrow PoK\big((u, r^*, r_0, r^\dagger, r_1, \ldots, r_n); \\ c_i =_\omega c^* \text{ or } c^\dagger \quad \text{for all } i \in [n] \\ \wedge \prod_{i=1}^{n} Com(x_i, r_i) = (c^*)^l (c^\dagger)^{n-l} Com(0, v)\big)\end{aligned} \tag{24}$$

where $v = \sum_{i=1}^n r_i - l r^* - (n-l) r^\dagger$ and π is an accepting proof[3].

For the proofs of the relations, the following four proofs of knowledge of the relation between a witness $w \in \boldsymbol{W}$ and a statement $x \in \boldsymbol{X}$ are required, where $\boldsymbol{X}$ and $\boldsymbol{W}$ are the sets of witnesses and statements, respectively.

1. PoK-1: $((u, r^*, \rho); c^* \in_\omega \boldsymbol{A})$
2. PoK-2: $((u, r^*, \sigma); c^* \in_\omega \boldsymbol{C}_S)$
3. PoK-3: $\big((u, r_0), (r_1, \ldots, r_n);\ c_i = \mathrm{Com}(u, r_i) \text{ or } \mathrm{Com}(r_0, r_i) \text{ for all } i \in [n]\big)$
4. PoK-4: $\big((u, r_0), r^*, r^\dagger, (r_1, \ldots, r_n);\ \prod_{i\in[n]} c_i = (c^*)^n (c^\dagger)^{n-l} Com(0, v)\big)$
   where $v = \sum_{i=1}^n r_i - l \cdot r^* - (n-l) \cdot r^\dagger$

For the proof of $\mathscr{R}_1$, the PoKs of (1)-(3) are required while the proof of $\mathscr{R}_2$ requires (1),(2),(4). Let us denote by $\mathscr{R} = (\mathscr{R}_1 \wedge \mathscr{R}_2)$.

Nest, we define the required security notions of the Fair-Anonymity.

Definition 4 (Completeness). *Completeness refers to the fact that a prover can always provide a valid proof except for negligible probability for a statement with a witness. More formally, given a statement $x = (c^*, c^\dagger, \boldsymbol{C}_S)$ and a witness $w = (u, r^*, r_0, r^\dagger, r_1, \ldots, r_n, \rho, \sigma)$, for every honest user $\mathcal{U}$ and honest verifier $\mathcal{V}$, and for every security parameter $\lambda > 0$, the following holds:*

$$\Pr\left[\mathcal{V}(x, \pi) = 1 \mid \pi \leftarrow \mathcal{U}(x, w), (x, w) \in \mathscr{R}\right] \geq 1 - \mathsf{negl}(\lambda). \tag{25}$$

where the probability is taken over the random coin-flips by $\mathcal{U}$.

[3] The relation of $c_i =_\omega c^*$ or $c^\dagger$ can be proven with OR-Proofs [14, 9]

Definition 5 (Soundness). *Soundness guarantees that the prover can give a proof that verifies for a false statement only with negligible probability. More formally, given a statement* $x = (c^*, c^\dagger, \boldsymbol{C}_S)$ *, for any PPT adversaries* $\mathcal{A}$ *and honest verifier* $\mathcal{V}$*, and for every security parameter* $\lambda > 0$*, the following holds:*

$$\Pr\left[\mathcal{V}(x, \pi') = 1 \mid \pi' \leftarrow \mathcal{A}(x)\right] < \mathsf{negl}(\lambda). \tag{26}$$

where the probability is taken over the random coin-flips by $\mathcal{A}$*.*

Definition 6 (Anonymity). *Anonymity guarantees that the outcome of the execution of the protocol does not non-negligibly increase the probability of identifying the user's ID* u *from the coin* c*. More formally, given a statement* $x = (c^*, c^\dagger, \boldsymbol{C}_S)$ *and a witness* $w = (u, r^*, r_0, r^\dagger, r_1, \ldots, r_n, \rho, \sigma)$*, for all PPT adversaries* $\mathcal{A}$*, and for all coin* $c \in \boldsymbol{C}$ *and for every security parameter* $\lambda > 0$*, it holds:*

$$\left|\Pr[u \leftarrow \mathcal{A}(c, x, \pi) \mid \pi \leftarrow \mathcal{U}(x, w)] - \Pr[u \leftarrow \mathcal{A}(c, x)]\right| \leq negl(\lambda) \tag{27}$$

where the probability is taken over the internal coin-flips of $\mathcal{U}$ *and* $\mathcal{A}$*.*

Definition 7 (Fair Traceability). *Fair Traceability guarantees Authority to trace the user's id* $u \in \boldsymbol{U}$ *involoved in a coin* $c \in \boldsymbol{C}$ *with an accepting fairness proof* $(x, \pi) \in \boldsymbol{X} \times \boldsymbol{\Pi}$ *with pre-agreed fair probability* $p(c.v)$ *determined by the coin value* $c.v$*. More formally, let* $\boldsymbol{K}$ *be the set of all possible secret keys. Let* $p : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ *be a probability function satisfing* ϵ*-Fairness in Definition 3. There exists an efficient probabilistic extractor* $\mathcal{E} : \boldsymbol{K} \times \boldsymbol{C} \times \boldsymbol{X} \times \boldsymbol{\Pi} \rightarrow \boldsymbol{U}$*, which takes a secret key* $sk \in \boldsymbol{K}$*, a valid coin* $c \in \boldsymbol{C}$*, and an accepting statement-proof pair* $(x, w) \in \mathscr{R}$ *as inputs and outputs a user id* $u \in \boldsymbol{U}$*. Given a statement* $x = (c^*, c^\dagger, \boldsymbol{C}_S)$ *and a witness* $w = (u, r^*, r_0, r^\dagger, r_1, \ldots, r_n, \rho, \sigma)$*, for every honest user* $\mathcal{U}$ *and for the extractor* $\mathcal{E}$*, it holds:*

$$\left|\Pr\left[\mathcal{E}(sk, c, x, \pi) = u \mid \pi \leftarrow \boldsymbol{\mathcal{U}}(w, x)\right] - p(c.v)\right| < \epsilon \tag{28}$$

Finally, we claim that the construction of Fair-Anonymity in Section 4 satisfies these security notions.

Theorem 1 (Fair-Anonymity).
The construction described in Section 4 satisfies Completeness (def. 4), Soundness (def. 5), Anonymity (def. 6) and Fair Traceability (def. 7).

The proof of this theorem is rather straightforward from the above discussions. Here, we will give only the proof sketch. The complete proof will appear in the full version of this paper.

Proof (sketch). We give the proof scketch that the construction of Fair-Anonymity satisfies the four security notions of Completeness, Soundness, Anonymity, and Fair Traceability as follows. Completeness follows directly from that of the Membership Proof, e.g. One-out-of-many Proof [15], in the PoK-1 - PoK-4, which implies $\mathcal{V}$ accepts if the all of the four proofs are accepting. In the same way, Soundness is implied from the soundness of PoK-1 to PoK-4. Adversary who breaks the Soundness (def. 5) has to break at least one of the soundness PoK-1 to PoK-4. Thus, this probability is negligible in λ Anonymity means that the probability of user information leaking from the published transcript (c, x, π) is negligible. From the Zero-knowledge property of PoK-1 to PoK-4, ensuring that the probability of distinguishing the witnesses is negligible. Noting that u is contained in the witness information, Anonymity is directly implied. Lastly, regarding Fair Traceability,

the probability that u is transmitted via $\mathrm{C^2OT}_n^k$ approximates the function $p(x)$ (7), which satisfies the fairness property based on the (n, l) pair in equations (12) - (13) in the Execution protocol. Indeed, there is an error due to approximating $p(x)$, defined over the real numbers $\mathbb{R}$, with integer pairs $n, l \in \mathbb{N}$, but this value is upper-bounded by $\epsilon \leq \frac{1}{2n}$ which decreases as n increases.

References

1. Olivier Blazy, Sébastien Canard, Georg Fuchsbauer, Aline Gouget, Hervé Sibert, and Jacques Traoré. Achieving Optimal Anonymity in Transferable E-Cash with a Judge. In Progress in Cryptology – AFRICACRYPT 2011, volume 6737, pages 206–223. 2011. Lecture Notes in Computer Science.
2. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short Accountable Ring Signatures Based on DDH. In Computer Security – ESORICS 2015, Lecture Notes in Computer Science, pages 243–265. Springer International Publishing, 2015.
3. Ernie Brickell, Peter Gemmell, and David Kravitz. Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change. 1993.
4. Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards Privacy in a Smart Contract World. In Financial Cryptography and Data Security, Lecture Notes in Computer Science, pages 423–443, Cham, 2020. Springer International Publishing.
5. Jan Camenisch, Anna Lysyanskaya, and Mira Meyerovich. Endorsed E-Cash. pages 101–115, May 2007. ISSN: 2375-1207.
6. David Chaum. Blind Signatures for Untraceable Payments. In Advances in Cryptology, pages 199–203. Springer US, 1983.
7. David Chaum and Torben Pryds Pedersen. Wallet Databases with Observers. In Advances in Cryptology — CRYPTO' 92, volume 740, pages 89–105. Springer, 1993.
8. Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-Out-of-n Oblivious Transfer Schemes with Adaptive and Non-adaptive Queries. In Public Key Cryptography - PKC 2005, Lecture Notes in Computer Science, 2005.
9. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In Advances in Cryptology — CRYPTO '94, Lecture Notes in Computer Science, pages 174–187, 1994.
10. Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. Balancing Privacy and Accountability in Blockchain Identity Management, 2020.
11. Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys. In Pairing-Based Cryptography – Pairing 2007, pages 39–59. Springer Berlin Heidelberg, 2007.
12. Benjamin E. Diamond. Many-out-of-Many Proofs and Applications to Anonymous Zether. pages 1800–1817. IEEE, May 2021.
13. Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short Lattice-Based One-out-of-Many Proofs and Applications to Ring Signatures. In Applied Cryptography and Network Security, Lecture Notes in Computer Science. Springer International Publishing, 2019.
14. Marc Fischlin, Patrick Harasser, and Christian Janson. Signatures from Sequential-OR Proofs. In Advances in Cryptology – EUROCRYPT 2020, Lecture Notes in Computer Science. Springer International Publishing, 2020.
15. Jens Groth and Markulf Kohlweiss. One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin. In Elisabeth Oswald and Marc Fischlin, editors, Advances in Cryptology - EUROCRYPT 2015, Lecture Notes in Computer Science, pages 253–280. Springer, 2015.
16. Fuchun Guo, Yi Mu, and Willy Susilo. Subset Membership Encryption and Its Applications to Oblivious Transfer. IEEE Transactions on Information Forensics and Security, 9(7):1098–1107, July 2014. Conference Name: IEEE Transactions on Information Forensics and Security.
17. Fuchun Guo, Yi Mu, Willy Susilo, and Vijay Varadharajan. Membership Encryption and Its Applications. In Information Security and Privacy, Lecture Notes in Computer Science, pages 219–234. Springer, 2013.
18. Aram Jivanyan. Lelantus : Towards Confidentiality and Anonymity of Blockchain Transactions From Standard Assumptions. 2019.
19. Jianchang Lai, Yi Mu, Fuchun Guo, Rongmao Chen, and Sha Ma. Efficient k-out-of-n oblivious transfer scheme with the ideal communication cost. 2018.
20. Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. pages 245–254. ACM, May 1999.

21. Michael O. Rabin. How To Exchange Secrets with Oblivious Transfer, 1981. Harvard University Technical Report 81.
22. Alin Tomescu, Adithya Bhat, Benny Applebaum, Ittai Abraham, Guy Gueta, Benny Pinkas, and Avishay Yanai. UTT: Decentralized Ecash with Accountable Privacy, 2022. Publication info: Preprint. MINOR revision.

Authors

Taishi Higuchi was born in 1989. He received a Master's degree in Physics from the Tokyo University of Science, Japan, Tokyo, in 2015. He is currently a researcher of Sakura Information Systems Co., Ltd. and a Ph.D Student in the Institute of Information Security (IISEC). His research interests include the blockchain and cryptography.

Akira Otsuka was born in Osaka in 1966. He received B.E. and M.E. from Osaka University in 1989 and 1991 respectively, and Ph.D. degree from University of Tokyo in 2002. From 2002, he was a Post Doctoral Fellow and a Cooperative Researcher at the University of Tokyo. From 2003 to 2005, he was a member of Cryptographic Technique Monitoring Subcommittee at CRYPTREC. From 2005, he was with National Institute of Advanced Industrial Science and Technology (AIST), serves as a Leader of Research Security Fundamentals during 2006 to 2010. During 2007 to 2014, he was a Visiting Professor at Research and Development Initiative, Chuo University. From 2017, he is a Professor at Graduate School of Information Security, Institute of Information Security.

A Cryptographic Techniques for Proof of knowledge

In this appendix, we present the cryptographic components and related knowledge necessary for constructing new k-out-of-n Committed Oblivious Transfer and the novel Fair-Anonymity we propose, which will be demonstrated in subsequent sections. We denote by $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{R}$ the set of natural numbers, integers and real numbers respectively. A function $\epsilon : \mathbb{N} \to \mathbb{R}_{>0}$ is said to be negligible if for every $d \in \mathbb{N}$, there exists $\lambda_0 \in \mathbb{N}$ such that $\epsilon(\lambda) \leq \lambda^{-d}$ for all $\lambda > \lambda_0$.

A.1 One-out-of-Many Proofs

One-out-of-Many proofs, originally introduced by Groth and Kohlweiss [15], allow a prover to demonstrate knowledge of a secret element among a public list of commitments, together with an opening of this commitment to 0. Their impactful research finding has since been employed in subsequent studies [13, 12] and various significant applications [18, 4, 2].

Their proposed scheme is a 3-move public coin special honest verifier zero-knowledge proof (Sigma-protocol) for a list of commitments having at least one commitment that opens to zero. In the scheme, it is not required for the prover to know openings of the other commitments. Their construction works for any additively homomorphic non-interactive commitment schemes such as Pedersen commitments over $\mathbb{Z}_p$, where p is a large prime.

They give a Σ-protocol for knowledge of one out of N commitments $c_0, \ldots, c_{N-1}$ being a commitment to 0. More precisely, a Σ-protocol for the relation is given as

$$R = \left\{ \left(ck, \left(c_0, \ldots, c_{N-1} \right), \left(\ell, r \right) \right) \middle| \begin{array}{l} c_0, \ldots, c_{N-1} \in \mathcal{C}_{ck} \text{ and } \ell \in \{0, \ldots, N-1\} \\ \text{and } r \in \mathbb{Z}_p \text{ and } c_\ell = \mathrm{Com}_{ck}(0, r) \end{array} \right\}. \tag{29}$$

Theorem 2 ([15, Theorem 3]). *One-out-of-Many proof, the Σ-protocol for knowledge of one out of N commitments opening to 0, is perfectly complete. It is (perfect) $(n+1)$-special sound if the commitment scheme is (perfectly) binding. It is (perfect) special honest verifier zero-knowledge if the commitment scheme is (perfectly) hiding.*

A.2 Verifiable Encryption for the Equality of Witnesses with Pedersen Commitment

Here, we introduce a specific Verifiable Encryption scheme [7] that proves the equivalence of the message of a Pedersen commitment with the message in a ciphertext encrypted via ElGamal Encryption, without revealing the message itself. The commitment and the ciphertext are calculated over distinct groups of the same order. The detailed protocol is shown in the Fig. 4.

The following protocol can verify the equivalence of a message committed using Pedersen commitment with a message encrypted via ElGamal Encryption. The public parameters are $(g, h = g^x) \in \mathbb{G}$ and $(g_T, h_T = g_T^{x'}) \in \mathbb{G}_T$ where $x, x' \xleftarrow{\$} \mathbb{Z}_p^*$ and $|\mathbb{G}| = |\mathbb{G}_T|$.

1. **Commitment and Encryption:**
   - The prover, $\mathcal{P}$, takes as input $(g, h), (g_T, h_T)$ and computes a Pedersen commitment and an ElGamal encryption

$$c = g^m h^r, \qquad \tilde{c} = (g_T^m h_T^{r'}, g_T^{r'}) \tag{30}$$

     for a same message $m \in \mathbb{Z}_p^*$ and two random values $r, r' \xleftarrow{\$} \mathbb{Z}_p^*$.
   - $\mathcal{P}$ chooses random values $(r_1, r_2, r_3) \xleftarrow{\$} \mathbb{Z}_p^*$ and computes

$$v = g^{r_1} h^{r_2}, \qquad \tilde{v} = (g_T^{r_1} h_T^{r_3}, g_T^{r_3}). \tag{31}$$

2. **Generation of the Challenge and Response:**
   - The prover generates a challenge hash value $d = \text{Hash}(c, \tilde{c}, v, \tilde{v})$.
   - $\mathcal{P}$ then calculates the responses

$$z_1 = dm + r_1, \quad z_2 = dr + r_2, \quad z_3 = dr' + r_3. \tag{32}$$

3. **Verification:**
   - The verifier receives the proof, which includes $(c, \tilde{c}, v, \tilde{v}, z_1, z_2, z_3)$, and checks the validity of the proof by confirming two equations:

$$v \cdot c^d \stackrel{?}{=} g^{z_1} h^{z_2}, \qquad \tilde{v} \circ \tilde{c}^d \stackrel{?}{=} (g_T^{z_1} h_T^{z_3}, g_T^{z_3}) \tag{33}$$

     where the operation $(x_1, y_1) \circ (x_2, y_2)$ represents $(x_1 x_2, y_1 y_2)$.
   - These checks ensure that the commitments and encryptions are consistent with the response z_1, z_2, z_3. If the equations hold, the verifier can be confident that the committed value in c and the encrypted value in $\tilde{c}$ are indeed the same, without knowing what that value is.

Fig. 4: Verifiable Encryption for the equality of Pedersen commitment and ElGamal Encryption