A Provably Secure Network Protocol for Private Communication with Analysis and Tracing Resistance [*]

Chao Ge
Department of Electronic Engineering
Tsinghua University
Beijing
gechao@amss.ac.cn

Wei Yuan
State Key Laboratory of Mathematical Sciences
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing
wyuan@math.ac.cn

Ge Chen
State Key Laboratory of Mathematical Sciences
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing
chenge@amss.ac.cn

Yanbin Pan
State Key Laboratory of Mathematical Sciences
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing
panyanbin@amss.ac.cn

Yuan Shen
Department of Electronic Engineering
Tsinghua University
Beijing
shenyuan_ee@tsinghua.edu.cn

Abstract

Anonymous communication networks have emerged as crucial tools for obfuscating communication pathways and concealing user identities. However, their practical deployments face significant challenges, including susceptibility to artificial intelligence (AI)-powered metadata analysis, difficulties in decentralized architectures, and the absence of provable security guarantees. To address these issues, this paper proposes a novel decentralized anonymous routing protocol with resistance to tracing and traffic analysis. The protocol eliminates dependencies on the threshold model and trusted third-party setups, ensuring indistinguishable identity privacy even in highly adversarial environments. Different from traditional empirical security analysis of anonymous networks, this paper rigorously proves indistinguishable identity privacy for users even in extremely adversarial environments. Furthermore,

[*] This material has been submitted to submitted to the IEEE Journal on Selected Areas in Communications.

This paper was supported by the National Key Research and Development Program of China (2022YFA1004600), the National Natural Science Foundation of China (12288201, 12071465).

simulations confirm its practical feasibility, demonstrating both security and efficiency. By achieving information sharing with privacy preservation, the proposed protocol offers a provably secure solution for privacy-preserving communication in digital environments.

***K*eywords** Anonymous network · Provable security · Privacy preservation · Communication security · Traffic analysis resistance

1 Introduction

The rapid development of wireless communication technology has driven an exponential increase in terminal devices, intensifying the social reliance on ubiquitous network access. While facilitating diverse applications, these technologies simultaneously pose significant privacy risks due to the widespread transmission of sensitive data [1–4]. The emergence of 6G networks will heighten this concern by generating, storing, and processing vast amounts of data, including precise geolocation tracking and predictive user profiling. This challenge has spurred urgent research into communication security and privacy preservation.

Traditional cryptographic methods ensure content confidentiality and integrity but are inadequate against emerging threats. Modern challenges require broader protection that extends to communication metadata, including temporal patterns, frequency characteristics, and relationship dynamics within communication processes [2, 3, 5–7]. As artificial intelligence (AI) reshapes cyberattack strategies, adversaries are increasingly targeting metadata inference over content theft [1, 3, 8]. By leveraging advanced data analytics and machine learning, they extract behavioral patterns from communication processes to deduce critical and sensitive information [9, 10]. This is particularly vital in next-generation networks, where metadata can expose operational patterns and user behaviors through sophisticated correlation and AI-driven inference attacks.

To address risks of identity traceability and data linkability, anonymous communication networks have become a key research focus [11–28]. Anonymous communication networks enable users to communicate without revealing their identities, locations, or behavioral patterns, thereby enhancing confidentiality. They are applied in secure military operations, privacy-preserving networks, and anonymous social platforms that promote free expression. Also, they play a vital role in e-commerce by preventing third-party tracking, as well as in e-democracy, online surveys, where anonymity ensures impartiality and data authenticity. These networks balance information sharing with privacy preservation, with prominent examples including the onion routing (Tor), invisible internet project (I2P), and Nym Mixnet. *Tor* is a connection-oriented system using multi-hop proxy sequences of volunteer nodes [14, 15]. It offers low latency, high anonymity, and ease of deployment, making it the most widely used anonymous network [16]. However, due to its lack of traffic obfuscation, it is susceptible to activity pattern detection and de-anonymization via website fingerprinting and end-to-end correlation attacks [16–18]. *I2P* is a hidden network that employs garlic routing, a variant of onion routing, with one-way encryption for end-to-end communication [19]. Using short-lived links, it reduces third-party tracking risks [20] and replaces Tor's centralized directories with a Distributed Hash Table (DHT), eliminating reliance on a central authority. However, secure DHT design remains challenging [29], and I2P is vulnerable to de-anonymization by global adversaries conducting traffic analysis [21–24]. Built on the Loopix protocol, *Nym Mixnet* offers superior metadata protection through cryptographic reordering and independent message routing [25, 26, 30]. Despite challenges in bandwidth, computation, and latency, Mixnets are critical for privacy-preserving communication [27, 28]. The evolving Nym network extends to a universal incentivized Mixnet for anonymous email and messaging.

Among these networks, preserving the anonymity of routing information remains a fundamental security challenge. These networks mainly depend on server-router interactions. However, bidirectional interactions introduce security vulnerabilities. Shi and Wu [31] propose the Non-Interactive Anonymous Router (NIAR) scheme. By eliminating interaction-related risks, NIAR ensures security even in the presence of untrusted nodes, a key capability that motivates our work. We defer the brief introduction of NIAR to Subsection 2.2.

Motivations: In summary, current anonymous routing systems, particularly interactive protocols, provide substantial privacy benefits by effectively obfuscating communication paths, thereby preventing adversaries from identifying the source or destination of messages. However, the security of these anonymous routing systems relies on the assumption that a majority of routing nodes remain uncompromised, known as the *threshold model* [32, 33]. As a result, these protocols guarantee anonymity only under this strict condition; if an adversary compromises a significant fraction of nodes, privacy protections weaken. The NIAR scheme, operating non-interactively, is built on a well-defined mathematical security model that enables theoretical proofs and provides a solid foundation for security analysis. Nevertheless, this scheme relies on a trusted initial setup, posing practical challenges in fully decentralized settings. Additionally, its computational complexity scales quadratically with the number of participants, leading to inefficient routing computation.

Motivated by these limitations, this paper designs a decentralized private communication network protocol that eliminates reliance on both the threshold model and a trusted initial setup. The protocol is provably secure, with its resistance to analysis and tracing verified through formal proofs. Notably, the mathematically grounded security ensures resilience against evolving adversarial strategies, as its rigorously verified cryptographic invariants remain robust against future attacks. By overcoming the limitations of existing systems, our protocol establishes a novel scheme for privacy-preserving communication in fully decentralized and untrusted environments, offering a theoretically sound and practically feasible approach.

Contributions: For the significant but challenging private and covert communications, we design a decentralized and threshold-model-free protocol to conceal communication paths and identities of communicating parties. The main contributions are outlined as follows.

- We propose a decentralized mechanism to dynamically generate routing configurations in the initial setup. This approach overcomes the critical reliance on a trusted initial setup inherent in NIAR schemes and enables practical operation in fully decentralized environments, thereby facilitating adaptability to dynamic network conditions.

- We present a detailed implementation of a decentralized anonymous routing protocol that eliminates reliance on the threshold model and any trusted initial setup. This protocol ensures indistinguishable identity privacy even in extremely adversarial environments. This marks a significant advancement over existing systems constrained by strict trust requirements.

- Provable security has long been a fundamental challenge in anonymous networks, remaining unresolved due to inherent cryptographic complexity. Different from traditional empirical security analysis of anonymous networks, this paper rigorously proves indistinguishable identity privacy for users even in extremely adversarial environments. This result establishes a formidable defense against both present and future adversarial strategies, offering a level of assurance unattainable by empirically driven approaches.

Simulation results confirm the practical applicability of the proposed protocol, demonstrating that it is both theoretically secure and practically efficient.

Structure: Section 2 presents the preliminaries of this paper, including the application scenario, the building blocks of our protocol, and the hardness assumptions for the theoretical security evaluation. Section 3 introduces the decentralized anonymous communication protocol, which operates independently of any threshold model. Subsequently, the theoretical security results of the protocol are detailed in Section 4. Section 5 provides simulation results and a detailed analysis to demonstrate the protocol's practical efficiency. Finally, conclusions are drawn in Section 6.

Notation: Throughout this paper, we use $\lambda \in \mathbb{N}^{+}$ to denote the security parameter which measures the input size of the computational problem. By convention, the security parameter is input in unary form, denoted as 1^{λ}, representing a string consisting of λ ones. The notation $\texttt{poly}(\lambda)$ denotes a *polynomial function* in λ, meaning that its growth rate is bounded by some polynomial in λ. The notation $\texttt{negl}(\lambda)$ denotes a *negligible function* in λ. A function is considered negligible if it decreases more rapidly than the inverse of any polynomial as λ grows, a property critical for ensuring that certain probabilities become vanishingly small as the security parameter increases. For a given prime q, let $\mathbb{Z}_q$ denote the ring of integers modulo q, which is the set $\{0, 1, \ldots, q-1\}$ equipped with addition and multiplication operations modulo q. Let $\mathbb{Z}_q^{\times}$ denote the multiplicative group of units of $\mathbb{Z}_q$, consisting of all non-zero elements $\{1, 2, \ldots, q-1\}$, since these elements are invertible when q is prime. Moreover, $\mathbb{Z}_q^{n \times m}$ denotes the set of all $n \times m$ matrices with elements from $\mathbb{Z}_q$. As introduced in [34], let $\{q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, \mu\}$ denote a bilinear group, where: i) $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of order q, whose generators are g_1, g_2, and g_T, respectively; ii) $\mu : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear map, and $g_T = \mu(g_1, g_2)$. For $t \in \{1, 2, T\}$, we define the notation $\llbracket a \rrbracket_t = g_t^a \in \mathbb{G}_t$. For convenience, we use $\llbracket a \rrbracket_{1,2}$ to represent the pair $(\llbracket a \rrbracket_1, \llbracket a \rrbracket_2)$. For vectors, let $\boldsymbol{\xi} \in \mathbb{Z}_q^{1 \times \ell}$ be a row vector and $\boldsymbol{\beta} \in \mathbb{Z}_q^{\ell \times 1}$ be a column vector. We denote $\llbracket \boldsymbol{\xi} \rrbracket_1 = (\llbracket \xi_1 \rrbracket_1, \ldots, \llbracket \xi_\ell \rrbracket_1)$, a vector of group elements in $\mathbb{G}_1$, $\llbracket \boldsymbol{\beta} \rrbracket_2 = (\llbracket \beta_1 \rrbracket_2, \ldots, \llbracket \beta_\ell \rrbracket_2)^T$, a vector of group elements in $\mathbb{G}_2$. Using the bilinear map μ, the inner product is computed as $\llbracket \langle \boldsymbol{\xi}, \boldsymbol{\beta} \rangle \rrbracket_T = \mu(\llbracket \boldsymbol{\xi} \rrbracket_1, \llbracket \boldsymbol{\beta} \rrbracket_2) \in \mathbb{G}_T$, which is also written as $\llbracket \boldsymbol{\xi} \rrbracket_1 \llbracket \boldsymbol{\beta} \rrbracket_2$. The operator $\leftarrow$ represents the random sampling of an output of a randomized algorithm, $\xleftarrow{\$}$ represents the uniform sampling of an element from a set. We operate within a standard computational model and define an adversary $\mathcal{A}$ as *probabilistic polynomial time* (p.p.t.), meaning that $\mathcal{A}$ executes in polynomial time relative to the security parameter. Unless otherwise specified, all algorithms are probabilistic. The probability that $\mathcal{A}$ outputs 1 in an experiment **Exp** is denoted by $\Pr[1 \leftarrow \mathcal{A}(\textbf{Exp})]$. For two bit-strings $\boldsymbol{x}$ and $\boldsymbol{y}$, their concatenation is represented as $\boldsymbol{x} \| \boldsymbol{y}$.

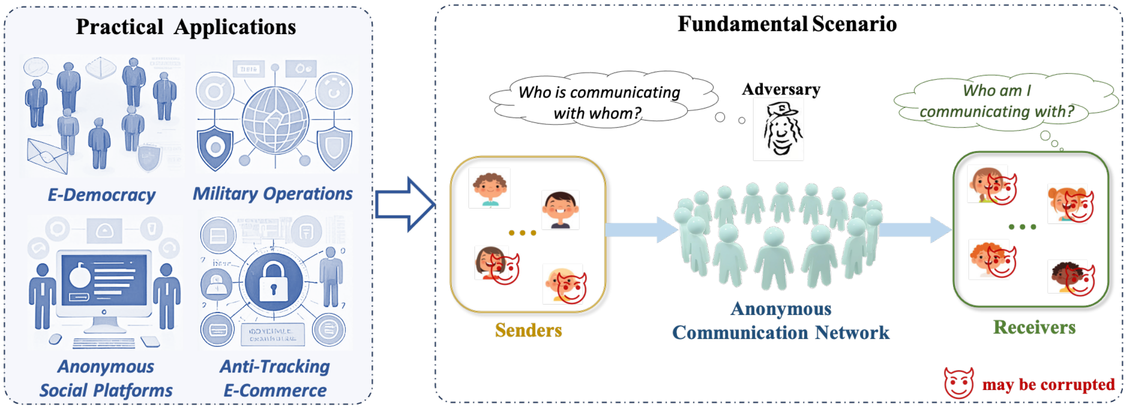

Figure 1: The application scenario of our private communication network protocol.

2 Preliminaries

In this section, we first describe the application scenario of our private communication protocol. Following this, we review the NIAR scheme proposed in [31], and the correlated pseudorandom function (CPRF), which serves as the foundational building block of our protocol. Finally, we present the relevant hardness assumptions for theoretical security evaluation.

2.1 Scenario Description

To address the privacy-preserving challenges of latency-tolerant remote communication networks, we design a decentralized private communication protocol for asynchronous message transmission. In this application scenario, the protocol ensures anonymity among all honest senders such that no participant can distinguish the identity of its communication source. Furthermore, it achieves unlinkability against adversarial entities equipped with comprehensive monitoring capabilities and traffic analysis techniques.

Without loss of generality, we consider the foundational scenario and formalize an n-to-n communication paradigm, as illustrated in Fig. 1. The fundamental communication scenario involves n senders and n receivers, where each sender intends to communicate anonymously with a unique receiver. To conceal the identities of participants, all communications undergo a permutation process. Specifically, let $\boldsymbol{\pi} \in S_n$ represent the permutation map between senders and receivers. For sender $i \in [n]$, let $\boldsymbol{\pi}(i)$ denote the receiver with whom sender i intends to communicate. This permutation process is executed by an untrusted router, which, by design, is unable to access any knowledge regarding the mapping information $\{i, \boldsymbol{\pi}(i)\}_{i\in[n]}$. Additionally, certain senders and receivers may be corrupted, allowing an adversary to gain knowledge of the permutation information known to corrupted participants. It is assumed that all participants comply with the specified routing protocol, although they may inadvertently leak information. Our objective is to design private communication networks that can effectively conceal the identities of participants, even when a subset of corrupted participants may collude with the untrusted router. Let $\mathcal{K}_S \subseteq [n]$ denote the set of corrupted senders and $\mathcal{K}_R \subseteq [n]$ denote the set of corrupted receivers. Let $\mathcal{H}_S := [n] \setminus \mathcal{K}_S$ and $\mathcal{H}_R := [n] \setminus \mathcal{K}_R$ denote the set of honest senders and receivers, respectively. It is noteworthy that the generation of π exhibits permutation invariance with respect to the order of senders. Without loss of generality, we can assume that $\mathcal{K}_S = \{m{+}1, \cdots, n\} \subset \{m'{+}1, \cdots, n\} = \mathcal{K}_R$, where $m' \leq m$, denoting the number of honest receivers and senders, respectively.

2.2 Non-Interactive Anonymous Router

As introduced in [31], the NIAR scheme consists of the following procedures:

- $(\{\boldsymbol{ek}_i, \boldsymbol{rk}_i\}_{i\in[n]}, \boldsymbol{tk}) \leftarrow \texttt{NIAR.Setup}(1^\lambda, n, \boldsymbol{\pi})$. First, the one-time trusted setup takes the security parameter 1^λ, the sender/receiver number n, and the routing permutation $\boldsymbol{\pi}$. Subsequently, it outputs the sender keys $\{\boldsymbol{ek}_i\}_{i\in[n]}$, the receiver keys $\{\boldsymbol{rk}_i\}_{i\in[n]}$, and a token $\boldsymbol{tk}$ for the router to encode $\boldsymbol{\pi}$.
- $\boldsymbol{ct}_{i,t} \leftarrow \texttt{NIAR.Enc}(\boldsymbol{ek}_i, \boldsymbol{msg}_{i,t}, t)$. For $t = 1, 2, \cdots$, sender i encrypts its message $\boldsymbol{msg}_{i,t}$ with its secret key $\boldsymbol{ek}_i$ and sends ciphertext $\boldsymbol{ct}_{i,t}$ to the router.

- $\{\boldsymbol{ct}'_{i,t}\}_{i\in[n]} \leftarrow \texttt{NIAR.Rte}\left(\boldsymbol{tk}, \{\boldsymbol{ct}_{i,t}\}_{i\in[n]}\right)$. The router uses the token $\boldsymbol{tk}$ to transform the ciphertexts and then forwards $\boldsymbol{ct}'_{i,t}$ to receiver i.
- $\boldsymbol{msg}_{i,t} \leftarrow \texttt{NIAR.Dec}(\boldsymbol{rk}_i, \boldsymbol{ct}'_{i,t})$. Receiver $\boldsymbol{\pi}(i)$ uses its secret key $\boldsymbol{rk}_i$ to decrypt $\boldsymbol{ct}'_{i,t}$, thereby retrieving the plaintext $\boldsymbol{msg}_{i,t}$.

The indistinguishability security of the NIAR scheme is defined as follows. Let $\mathcal{A}$ denote a non-uniform p.p.t. adversary and $\mathcal{C}$ denote a challenger. Consider the following experiment $\textbf{NIAR.Exp}^{(b)}(1^\lambda)$ indexed by $b \in \{0,1\}$:

- $n, \mathcal{K}_S, \mathcal{K}_R, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1 \leftarrow \mathcal{A}\left(1^\lambda\right)$: $\mathcal{A}$ outputs $\boldsymbol{\pi}_0, \boldsymbol{\pi}_1 \in S_n$ satisfying $\{(i, \boldsymbol{\pi}_0(i)) : i \in \mathcal{K}_S\} = \{(i, \boldsymbol{\pi}_1(i)) : i \in \mathcal{K}_S\}$, and then sends them to the challenger $\mathcal{C}$.
- $\mathcal{C}$ selects $b \xleftarrow{\$} \{0,1\}$ and runs $\left(\{\boldsymbol{ek}_i, \boldsymbol{rk}_i\}_{i\in[n]}, \boldsymbol{tk}\right) \leftarrow \texttt{Setup}(1^\lambda, n, \boldsymbol{\pi}_b)$. Then, $\mathcal{C}$ gives $\{\boldsymbol{ek}_i\}_{i\in\mathcal{K}_S}, \{\boldsymbol{rk}_j\}_{j\in\mathcal{K}_R}$, and $\boldsymbol{tk}$ to $\mathcal{A}$.
- $\mathcal{A}$ is allowed to make a polynomial number of queries. In the k-th query, $\mathcal{A}$ selects and sends two sets of plaintexts $\{\boldsymbol{msg}^0_{i,t}\}_{i\in\mathcal{H}_S}$ and $\{\boldsymbol{msg}^1_{i,t}\}_{i\in\mathcal{H}_S}$ that satisfy $\boldsymbol{msg}^0_{\boldsymbol{\pi}_0^{-1}(i)} = \boldsymbol{msg}^1_{\boldsymbol{\pi}_1^{-1}(i)}, \forall i \in \mathcal{K}_R \cap \boldsymbol{\pi}_0(\mathcal{H}_S) = \mathcal{K}_R \cap \boldsymbol{\pi}_1(\mathcal{H}_S)$. Then, $\mathcal{C}$ returns $\{\texttt{Enc}\left(\boldsymbol{ek}_i, \boldsymbol{msg}^b_{i,t}, t\right)\}_{i\in\mathcal{H}_S}$ to $\mathcal{A}$.

The NIAR scheme is secure if and only if experiments $\textbf{NIAR.Exp}^{(0)}(1^\lambda)$ and $\textbf{NIAR.Exp}^{(1)}(1^\lambda)$ are computationally indistinguishable for any p.p.t. adversary. Based on the standard Decisional Linear assumption in certain bilinear groups, the NIAR scheme is provably secure [31]. However, it relies on a trusted initial setup, which encodes routing information via a central trusted authority. Therefore, it is incapable of addressing the risk of single-point failure and lacks dynamic adaptability. In this paper, we aim to enable all participants to collaboratively generate the dynamic routing in a distributed manner without a trusted central node. Moreover, each participant can access only its own routing information, thereby preventing information leakage and ensuring secure communication as well as the privacy of multi-party participation.

2.3 Correlated pseudorandom Function

Here we introduce the Correlated Pseudorandom Function (CPRF), which serves as an essential building block of our protocol. Following [31, 35], we use the CPRF as a random number generator. For distinct periods $t = 0, 1, \cdots$, each sender $i \in [n]$ independently computes a secret $K_i(t)$ by running the CPRF, which is parameterized by the security parameter 1^λ, the number of senders n, and a prime q, denoted by

$$(K_1(t), \cdots, K_n(t)) \leftarrow \texttt{CPRF}(1^\lambda, n, q),$$

where $K_1(t), \cdots, K_n(t)$ satisfy the following conditions:

C1. $K_i(t) \in \mathbb{Z}_q$;

C2. $\sum_i K_i(t) = 0$;

C3. for any non-uniform p.p.t. adversary $\mathcal{A}$,

$$\left| \Pr\left[1 \leftarrow \mathcal{A}\left(\textbf{CPRF.Exp}^{(0)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\textbf{CPRF.Exp}^{(1)}\right)\right] \right| = \texttt{negl}(\lambda),$$

where $\textbf{CPRF.Exp}^{(b)},\ b \in \{0,1\}$ is defined as:

- $\mathcal{A}$ sends a subset $\mathcal{K} \subset [n]$ with $|\mathcal{K}| \leq n-2$ to the challenger $\mathcal{C}$, implying that there exists at least two trusted senders;
- $\mathcal{C}$ samples $b \xleftarrow{\$} \{0,1\}$. After running $\texttt{CPRF}(1^\lambda, n, q)$, $\mathcal{C}$ returns $\{K_i\}_{i\in\mathcal{K}}$ to $\mathcal{A}$;
- $\mathcal{A}$ submits distinct t. If $b = 0$, $\mathcal{C}$ returns $\{K_j(t)\}_{j\notin\mathcal{K}}$. If $b = 1$, $\mathcal{C}$ randomly samples $\{d_j \in \mathbb{Z}_q\}_{j\notin\mathcal{K}}$ that satisfy $\sum_{j\notin\mathcal{K}} d_j = -\sum_{i\in\mathcal{K}} K_i(t)$, then returns $\{d_j\}_{j\notin\mathcal{K}}$.
- $\mathcal{A}$ outputs 0 or 1.

Remark 1 *Condition (C3) guarantees that any non-uniform p.p.t. adversary's view in **CPRF.Exp**$^{(0)}$ and **CPRF.Exp**$^{(1)}$ are computationally indistinguishable, implying that$\{K_i\}_{i\in\mathcal{K}}$ are computationally indistinguishable from random elements satisfying $\sum_{j\notin\mathcal{K}} d_j = -\sum_{i\in\mathcal{K}} K_i(t)$.*

As introduced in [36–38], CPRFs satisfying conditions (C1)-(C3) can be constructed with specific pseudorandom function family (PRF). A PRF refers to a collection of functions designed to produce outputs that are computationally indistinguishable from truly random outputs when evaluated with a randomly chosen key [36, 38]. Formally, let $\{0,1\}^*$ denote the set of all binary strings, representing possible keys, and $\{0,1\}^\lambda$ denote the set of binary strings of length λ. A PRF is defined as a map: $\mathtt{PRF} : \{0,1\}^* \times \{0,1\}^\lambda \mapsto \mathbb{Z}_q$. Let W_0 denote the probability that any p.p.t. adversary correctly identifies an output as being generated by $\mathtt{PRF}$, and let W_1 denote the probability that the adversary correctly identifies an output as being drawn uniformly at random from $\mathbb{Z}_q$. A PRF is considered secure if and only if the advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathtt{PRF}}(\lambda) := |W_0 - W_1|$ is negligible. For $1 \leq i < j \leq n$, k_{ij} are chosen at random. Let $k_{ij} := k_{ji}$ for $i > j$. The derived key K_i is constructed as

$$K_i := \sum_{j \neq i} (-1)^{j<i} \mathtt{PRF}(k_{ij}), \tag{1}$$

where $(-1)^{j<i}$ is the indicator function. As proven in [31, 36], this construction (1) satisfies conditions (C1)-(C3) and the following Lemma 1 holds.

Lemma 1 *Suppose the pseudorandom function family* $\mathtt{PRF}$ *is secure. Under construction (1), it holds that*

$$\mathrm{Adv}_{\mathcal{A}}^{\mathtt{CPRF}}(\lambda) := \left| \Pr\left[1 \leftarrow \mathcal{A}\left(\textbf{\textit{CPRF.Exp}}^{(0)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\textbf{\textit{CPRF.Exp}}^{(1)}\right)\right] \right| \leq \frac{n(n-1)}{2} \mathrm{Adv}_{\mathcal{A}}^{\mathtt{PRF}}(\lambda).$$

By Lemma 1, we note that if the underlying PRF is secure, then the CPRF constructed by (1) is also secure. Since $\mathrm{Adv}_{\mathcal{A}}^{\mathtt{PRF}}$ is negligible and $\frac{n(n-1)}{2}$ is polynomial, we have $\frac{n(n-1)}{2}\mathrm{Adv}_{\mathcal{A}}^{\mathtt{PRF}}(\lambda)$ is also negligible.

2.4 External Decisional Linear Assumption

Different from the NIAR scheme based on the Decisional Linear assumption [31], we construct our protocol on the external decisional linear (XDLin) assumption [39, 40], defined as follows.

Definition 1 (XDLin) *Let* $\mathrm{pp}_{\mathbb{G}} := \{q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mu\} \leftarrow \mathcal{G}(1^\lambda)$ *be a bilinear group generated by algorithm* $\mathcal{G}(1^\lambda)$*. The external decisional linear assumption holds for* $\mathcal{G}$ *if and only if the following distributions* P_0 *and* P_1 *are computationally indistinguishable. For* $x \in \{0,1\}$,

- $P_0 := (\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket ac \rrbracket_{1,2}, \llbracket bd \rrbracket_{1,2}, \llbracket c+d \rrbracket_x)$ *with* $a, b, c, d \xleftarrow{\$} \mathbb{Z}_q$;
- $P_1 := (\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket ac \rrbracket_{1,2}, \llbracket bd \rrbracket_{1,2}, \llbracket e \rrbracket_x)$ *with* $a, b, c, d, e \xleftarrow{\$} \mathbb{Z}_q$,

that is, for any p.p.t. $\mathcal{A}$,

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda) := \left| \Pr\left[1 \leftarrow \mathcal{A}(P_0)\right] - \Pr\left[1 \leftarrow \mathcal{A}(P_1)\right] \right| = \mathtt{negl}(\lambda).$$

The XDLin assumption asserts the hardness of distinguishing a specific element in $\mathbb{G}_T$ from a random element, given certain elements in $\mathbb{G}_1$ and $\mathbb{G}_2$. The advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda)$ is negligible.

3 Design of the Decentralized Anonymous Communication Protocol

In this section, we present the design of our threshold-model-free decentralized anonymous communication protocol in two phases (Phase I: Setup; Phase II: Communication). First, we introduce a decentralized setup method without any trusted authority. Subsequently, we detail the communication procedures of our anonymous router. We remark that the protocol operates in a fully decentralized manner and does not rely on threshold models. To streamline presentation, we present only the plaintext version of the anonymous routing protocol. This protocol can be easily extended to support encrypted communications through integration with standard cryptographic primitives. The roadmap of our protocol is illustrated in Fig. 2.

3.1 Decentralized Setup Process

To achieve initialization without any trusted third-party assumptions, the setup process of our protocol involves the following steps.

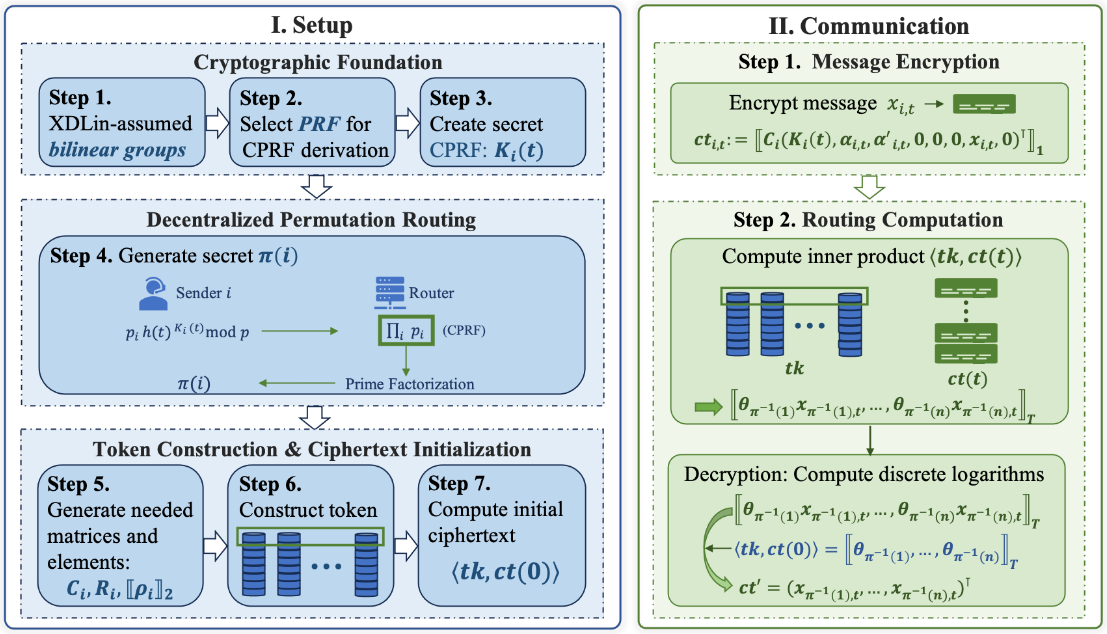
I. Setup
Cryptographic Foundation
Step 1. XDLin-assumed bilinear groups
Step 2. Select PRF for CPRF derivation
Step 3. Create secret CPRF: $K_i(t)$
Decentralized Permutation Routing
Step 4. Generate secret $\pi(i)$
Sender i
Router
$p_i\, h(t)^{K_i(t)} \bmod p$
$\Pi_i\, p_i$
(CPRF)
$\pi(i)$
Prime Factorization
Token Construction & Ciphertext Initialization
Step 5. Generate needed matrices and elements: $C_i, R_i, [\![\rho_i]\!]_2$
Step 6. Construct token
Step 7. Compute initial ciphertext $\langle tk, ct(0)\rangle$
II. Communication
Step 1. Message Encryption
Encrypt message $x_{i,t}$
$ct_{i,t} := [\![C_i(K_i(t), \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0)^\top]\!]_1$
Step 2. Routing Computation
Compute inner product $\langle tk, ct(t)\rangle$
tk
$ct(t)$
$[\![\theta_{\pi^{-1}(1)} x_{\pi^{-1}(1),t}, \dots, \theta_{\pi^{-1}(n)} x_{\pi^{-1}(n),t}]\!]_T$
Decryption: Compute discrete logarithms
$[\![\theta_{\pi^{-1}(1)} x_{\pi^{-1}(1),t}, \dots, \theta_{\pi^{-1}(n)} x_{\pi^{-1}(n),t}]\!]_T$
$\langle tk, ct(0)\rangle = [\![\theta_{\pi^{-1}(1)}, \dots, \theta_{\pi^{-1}(n)}]\!]_T$
$ct' = (x_{\pi^{-1}(1),t}, \dots, x_{\pi^{-1}(n),t})^\top$

Figure 2: The roadmap of the proposed private communication network protocol.

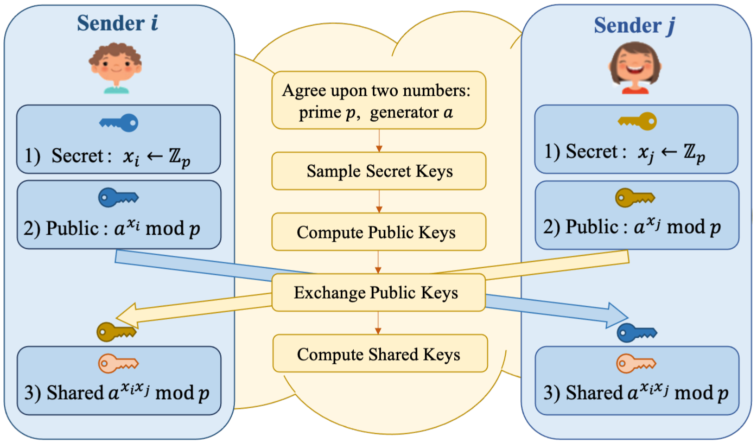
Sender i
1) Secret : $x_i \leftarrow \mathbb{Z}_p$
2) Public : $a^{x_i} \bmod p$
3) Shared $a^{x_i x_j} \bmod p$
Agree upon two numbers: prime p, generator a
Sample Secret Keys
Compute Public Keys
Exchange Public Keys
Compute Shared Keys
Sender j
1) Secret : $x_j \leftarrow \mathbb{Z}_p$
2) Public : $a^{x_j} \bmod p$
3) Shared $a^{x_i x_j} \bmod p$

Figure 3: The DH key exchange protocol for senders i, j in Step 3 of the setup process.

Step 1: Select a bilinear group that satisfies the XDLin assumption $\{q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, \mu\}$. Here we note that the generators g_1 and g_2 randomly selected and undergo dynamic updates in compliance with the security protocol.

Step 2: Select a pseudorandom function family $\texttt{PRF} : \{0,1\}^* \times \{0,1\}^\lambda \mapsto \mathbb{Z}_q$.

Step 3: Select a sufficiently large prime p to support the Diffie-Hellman (DH) protocol [41]. For $1 \le i < j \le n$, senders i and j jointly generate the secret key k_{ij} using the DH protocol, whose security is based on the difficulty of solving the discrete logarithm problem. As described in Fig. 3, the DH protocol enables two parties to securely establish a secret key known only to them within an insecure network environment. To be more specific, senders i and j agree on a public generator a. Then, sender i randomly selects a private key $x_i \xleftarrow{\$} \mathbb{Z}_p$, computes $a^{x_i} \pmod p$, and sends $a^{x_i} \pmod p$ to sender j. Subsequently, sender j randomly selects $x_j \xleftarrow{\$} \mathbb{Z}_p$, computes $a^{x_j} \pmod p$, and sends $a^{x_j} \pmod p$ to sender i. Afterwards, senders i and j can compute the shared secret key $k_{ij} := a^{x_i x_j} \pmod p$. Let $k_{ij} := k_{ji}$ for $i > j$. Hence, each sender $i \in [n]$ can compute $K_i(t) := \sum_{j \neq i} (-1)^{j<i} \texttt{PRF}(k_{ij}, t) \pmod p$. It holds that $\sum_{i=1}^n K_i(t) = 0$.

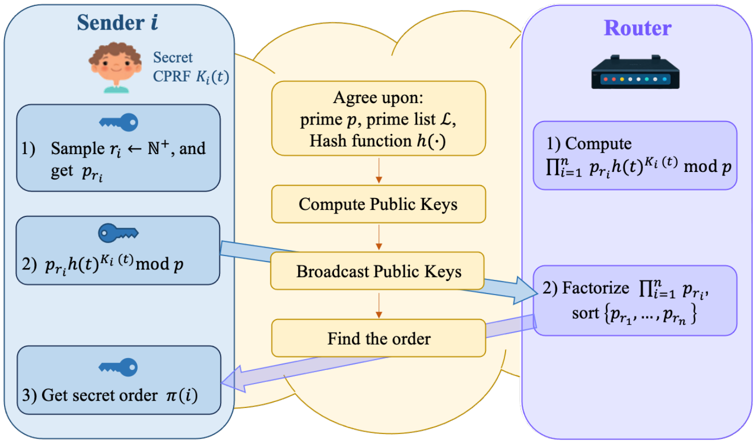
Sender i
Secret CPRF $K_i(t)$
1) Sample $r_i \leftarrow \mathbb{N}^+$, and get p_{r_i}
2) $p_{r_i}h(t)^{K_i(t)}$mod p
3) Get secret order $\pi(i)$
Agree upon: prime p, prime list $\mathcal{L}$, Hash function $h(\cdot)$
Compute Public Keys
Broadcast Public Keys
Find the order
Router
1) Compute $\prod_{i=1}^{n} p_{r_i}h(t)^{K_i(t)}$ mod p
2) Factorize $\prod_{i=1}^{n} p_{r_i}$, sort $\{p_{r_1}, \ldots, p_{r_n}\}$

Figure 4: The generation of permutation mapping $\boldsymbol{\pi}$ in Step 4 of the setup process.

Step 4: All senders collaboratively generate a permutation $\boldsymbol{\pi} \in S_n$ in a decentralized and anonymous manner, without a trusted third party. The detailed process is described in Fig. 4. Given a public prime list $\mathcal{L}$ and a Hash function $h(t)$, sender i randomly selects a positive integer $r_i \in \mathbb{N}^+$. Denote the r_i-th element of $\mathcal{L}$ by p_{r_i}. It is worth mentioning that p_{r_i} should not be excessively large. With the public Hash function $h(t)$, sender i computes $p_{r_i}h(t)^{K_i(t)} \pmod p$, and then sends the result to the router. After receiving $\{p_{r_1}h(t)^{K_1(t)}, \cdots, p_{r_n}h(t)^{K_n(t)}\}$, the router computes

$$\prod_{i=1}^{n} p_{r_i}h(t)^{K_i(t)} \pmod p = (\prod_{i=1}^{n} p_{r_i})h(t)^{\sum_{i=1}^{n} K_i(t)} \pmod p = \prod_{i=1}^{n} p_{r_i}.$$

Subsequently, the router performs prime factorization on $\prod_{i=1}^{n} p_{r_i}$, sorts the factors, and publishes the ordered list. Each sender i identifies the position of p_{r_i} in the sorted list, denoted as $\boldsymbol{\pi}(i)$. Hence, $\boldsymbol{\pi}$ is the desired permutation. We note that each sender i knows only $\boldsymbol{\pi}(i)$, and the router has no knowledge of $\boldsymbol{\pi}$.

Remark 2 *If there are repetitions among $p_{r_1}, \cdots, p_{r_n}$, repeat the above steps until there are no repetitions.*

Step 5: For $i \in [n]$, sender i randomly samples $\theta_i \overset{\$}{\leftarrow} \mathbb{Z}_q^\times$, and invertible matrix $C_i, R_i \overset{\$}{\leftarrow} (\mathbb{Z}_q)_{8\times 8}^\times$ which satisfy

$$R_i C_i = \mathbf{I}_8,$$

where $\mathbf{I}_8$ is the identity matrix of order 8. Meanwhile, all senders collaboratively generate $\{[\![\rho_1]\!]_2, \cdots, [\![\rho_n]\!]_2\}$, while ensuring that no single sender can dominate or predict the outcome. The detailed process is described in Fig. 5. To be more specific, first, each sender generates a random number c_i, which is initially kept confidential. Given the Hash function $h(\cdot)$, each sender computes a commitment to the random number c_i, denoted by $h_i = h(c_i)$. The commitment h_i is then publicly disclosed. Once all senders have shared their commitments $h_1, \cdots, h_n$, each sender reveals the respective c_i. Other senders can verify the integrity of the revealed values by checking if $h(c_i) = h_i$, ensuring that c_i has not been altered post-commitment. After collecting all verified values $\{c_i\}_{i\in[n]}$, the shared seed s is computed as $s = h(c_1\|\cdots\|c_n)$. Utilizing the shared seed s as a foundation, $\{h(s\|i)\}_{i\in[n]}$ are public keys generated by all senders, where i is a unique index corresponding to each sender. Subsequently, $h(s\|i)$ is mapped to $\mathbb{G}_2$ via the standardized Hash-to-Curve algorithm [42], ensuring deterministic and secure encoding of the input data into a valid group element within $\mathbb{G}_2$. Let $[\![\rho_i]\!]_2 :=$ Hash-to-Curve $(h(s\|i)) \in \mathbb{G}_2$. Hash-to-curve is a standardized algorithm for uniformly mapping hash function outputs to points on an elliptic curve. Its security can be formally established based on mathematical foundations. The mapping satisfies computational indistinguishability and provides resilience against chosen-input attacks, where adversarially selected inputs do not affect the statistical distribution of the mapped points. Here we remark that $\{[\![\rho_i]\!]_2\}_{i\in[n]}$ are public and collectively generated by all senders, while the values of $\{\rho_i\}_{i\in[n]}$ remain secret.

Step 6: For $i \in [n]$, sender i computes n tokens. The j-th token corresponding to sender i is computed by

$$\boldsymbol{tk}_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i\delta_{j,\boldsymbol{\pi}(i)}, 0)\ R_i]\!]_2, \tag{2}$$

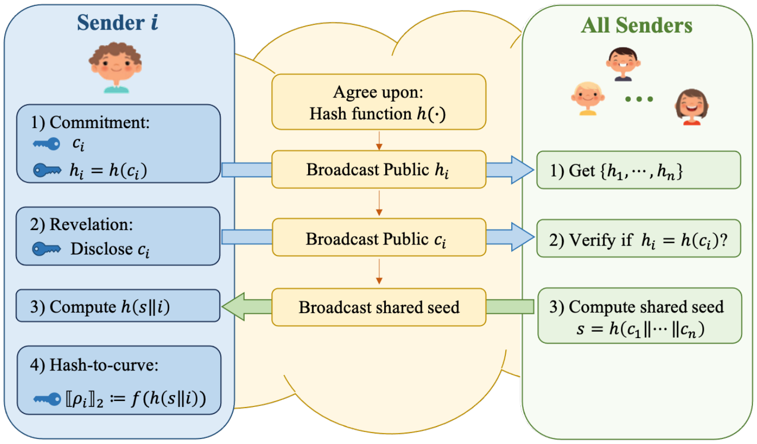

Figure 5: The generation of $\{[\![\rho_1]\!]_2, \cdots, [\![\rho_n]\!]_2\}$ in Step 5 of the setup process.

where $\beta_{ij}, \gamma_{ij} \xleftarrow{\$} \mathbb{Z}_q$, $\delta_{j,\boldsymbol{\pi}(i)} = 1$ if $j = \boldsymbol{\pi}(i)$, and otherwise $\delta_{j,\boldsymbol{\pi}(i)} = 0$. Then, sender i sends $\{\boldsymbol{tk}_i^1, \cdots, \boldsymbol{tk}_i^n\}$ to the router. After receiving

$$\begin{pmatrix} \boldsymbol{tk}_1^1 \\ \vdots \\ \boldsymbol{tk}_1^n \end{pmatrix}, \begin{pmatrix} \boldsymbol{tk}_2^1 \\ \vdots \\ \boldsymbol{tk}_2^n \end{pmatrix}, \cdots \begin{pmatrix} \boldsymbol{tk}_n^1 \\ \vdots \\ \boldsymbol{tk}_n^n \end{pmatrix},$$

the router reassembles them row by row. Let

$$\boldsymbol{tk}^j := \left(\boldsymbol{tk}_1^j, \cdots, \boldsymbol{tk}_n^j\right)$$

denote the j-th row, which is a row vector of length $8n$. Therefore, we can obtain the entire routing token

$$\boldsymbol{tk} := \left(\boldsymbol{tk}^1, \cdots, \boldsymbol{tk}^n\right)^\top.$$

Step 7: Let $\alpha_{i,0}, \alpha'_{i,0} \xleftarrow{\$} \mathbb{Z}_q$ be random samples. For $i \in [n]$, sender i computes and sends

$$\boldsymbol{ct}_{i,0} := [\![C_i \left(K_i(0), \alpha_{i,0}, \alpha'_{i,0}, 0, 0, 0, 1, 0\right)^\top]\!]_1$$

to the router. Let $\boldsymbol{ct}(0) := (\boldsymbol{ct}_{1,0}, \cdots, \boldsymbol{ct}_{n,0})^\top$. Subsequently, the router computes and stores the inner product

$$\langle \boldsymbol{tk}, \boldsymbol{ct}(0)\rangle = \left([\![\theta_{\boldsymbol{\pi}^{-1}(1)}]\!]_T, \cdots, [\![\theta_{\boldsymbol{\pi}^{-1}(n)}]\!]_T\right)^\top.$$

In summary, the setup process takes the security parameter 1^λ and the number of senders n as input, and outputs a token $\boldsymbol{tk}$ and an initial ciphertext, which can be denoted by

$$(\boldsymbol{tk}, \langle \boldsymbol{tk}, \boldsymbol{ct}(0)\rangle) \leftarrow \texttt{Setup}(1^\lambda, n).$$

3.2 Routing Steps in Communication Process

Based on the setup process, the routing steps of the communication process can be outlined as follows.

Step 1: Let $x_{i,t}$ be the t-th message that sender i wants to send. With $\alpha_{i,t}, \alpha'_{i,t} \xleftarrow{\$} \mathbb{Z}_q$, sender i computes

$$\boldsymbol{ct}_{i,t} := [\![C_i \left(K_i(t), \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0\right)^\top]\!]_1, \tag{3}$$

and sends it to the router. The router then collects $\{\boldsymbol{ct}_{i,t}\}_{i\in[n]}$ Let $\boldsymbol{ct}(t) := (\boldsymbol{ct}_{1,t}, \cdots, \boldsymbol{ct}_{n,t})^\top$.

Step 2: With the routing token $\boldsymbol{tk}$, the router computes the inner product $\langle \boldsymbol{tk}, \boldsymbol{ct}(t)\rangle$. By equation (2) and equation (3), we have

$$\langle \boldsymbol{tk}_i^j, \boldsymbol{ct}_{i,t}\rangle = [\![\rho_j K_i(t) + \theta_i \delta_{j,\boldsymbol{\pi}(i)} x_{i,t}]\!]_T.$$

Hence, we can obtain that

$$\begin{aligned}\langle \boldsymbol{tk}, \boldsymbol{ct}(t)\rangle &= \left(\sum_{i=1}^{n}\langle \boldsymbol{tk}_i^1, \boldsymbol{ct}_{i,t}\rangle, \cdots, \sum_{i=1}^{n}\langle \boldsymbol{tk}_i^n, \boldsymbol{ct}_{n,t}\rangle\right)^\top \\ &= \left([\![\theta_{\boldsymbol{\pi}^{-1}(1)} x_{\boldsymbol{\pi}^{-1}(1),t}]\!]_T, \cdots, [\![\theta_{\boldsymbol{\pi}^{-1}(n)} x_{\boldsymbol{\pi}^{-1}(n),t}]\!]_T\right)^\top. \end{aligned} \tag{4}$$

Recall from **Step** 7 of the setup process that the router has stored an initial ciphertext $\langle \boldsymbol{tk}, \boldsymbol{ct}(0)\rangle$. By computing the discrete logarithm of $[\![\theta_{\boldsymbol{\pi}^{-1}(i)} x_{\boldsymbol{\pi}^{-1}(i),t}]\!]_T$ with respect to $[\![\theta_{\boldsymbol{\pi}^{-1}(i)}]\!]_T$, the router can obtain

$$\boldsymbol{ct}'(t) := \left(x_{\boldsymbol{\pi}^{-1}(1),t}, \cdots, x_{\boldsymbol{\pi}^{-1}(n),t}\right)^\top.$$

In summary, the routing process is denoted as

$$\boldsymbol{ct}'(t) \leftarrow \texttt{Rte}\left(\boldsymbol{tk}, \boldsymbol{ct}_{1,t}, \cdots, \boldsymbol{ct}_{n,t}\right).$$

4 Theoretical Results of Security

In this section, we first extend the XDLin assumption to broader scenarios. Following this, we rigorously demonstrate the *provable security* of the proposed protocol through several sequences of *security experiments*.

As stated in Section 1, provable security plays a critical role in establishing the theoretical soundness of security mechanisms in cryptography and secure communication systems. By providing formal mathematical proofs, provable security provides quantifiable guarantees against specified adversarial models. Unlike heuristic or empirical security approaches relying on observed attack resistance, provable security guarantees that breaking the protocol would require solving a well-studied computational hardness problem [43,44].

Furthermore, provable security frameworks establish precise definitions of security goals, such as indistinguishability, and adversary capabilities, eliminating ambiguous notions of “security through obscurity" [45]. Therefore, provable security not only identifies potential vulnerabilities during the design phase, but also facilitates comparative analysis. From an evolutionary perspective, it has become indispensable for standardizing cryptographic protocols, as evidenced by its mandatory inclusion in modern algorithm specifications [45], making it a cornerstone of protocol design.

4.1 Theoretical Results Extended by XDLin Assumption

As stated in Subsection 2.4, the XDLin assumption asserts that an adversary cannot efficiently distinguish between a valid linear combination of group elements and a random one in certain bilinear groups. Building upon this assumption, we can further extend this “computational indistinguishability" to a broader range of scenarios. Hence, the following propositions are derived, forming the security foundation for the construction of our protocol.

Proposition 1 *Assume the XDLin assumption holds for $\mathcal{G}$. For $m = \texttt{poly}(\lambda) \geq 1$, consider the following distributions:*

- $P_0^{(1)} := ([\![a]\!]_{1,2}, [\![b]\!]_{1,2}, [\![a\boldsymbol{k}]\!]_{1,2}, [\![b\boldsymbol{d}]\!]_{1,2}, [\![\boldsymbol{k}+\boldsymbol{d}]\!]_x)$ *with* $a, b \xleftarrow{\$} \mathbb{Z}_q$ *and* $\boldsymbol{k}, \boldsymbol{d} \xleftarrow{\$} \mathbb{Z}_q^{m\times 1}$*;*
- $P_1^{(1)} := ([\![a]\!]_{1,2}, [\![b]\!]_{1,2}, [\![a\boldsymbol{k}]\!]_{1,2}, [\![b\boldsymbol{d}]\!]_{1,2}, [\![\boldsymbol{k}'+\boldsymbol{d}]\!]_x)$ *with* $a, b \xleftarrow{\$} \mathbb{Z}_q$ *and* $\boldsymbol{k}, \boldsymbol{k}', \boldsymbol{d} \xleftarrow{\$} \mathbb{Z}_q^{m\times 1}$.

For any p.p.t. $\mathcal{A}$, there holds

$$\left|\Pr\left[1 \leftarrow \mathcal{A}(P_0^{(1)})\right] - \Pr\left[1 \leftarrow \mathcal{A}(P_1^{(1)})\right]\right| \leq m\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda).$$

The proof of Proposition 1 is provided in Appendix A. We remark that Proposition 1 extends the XDLin assumption to the vector case. Since $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda)$ is negligible, we can obtain that $m\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda)$ is also negligible. Hence, no p.p.t. adversary can distinguish $P_0^{(1)}$ from $P_1^{(1)}$ with non-negligible advantage.

On this basis, the following Proposition 2 further imposes a sum constraint on the vectors $\boldsymbol{k}, \boldsymbol{k}'$.

Proposition 2 *Assume the XDLin assumption holds for $\mathcal{G}$. For $m = \texttt{poly}(\lambda) \geq 1$ and any $t \in \mathbb{Z}_q$, consider the following distributions:*

- $P_0^{(2)} := ([\![a]\!]_{1,2}, [\![b]\!]_{1,2}, [\![a\boldsymbol{k}]\!]_{1,2}, [\![b\boldsymbol{d}]\!]_{1,2}, [\![\boldsymbol{k}+\boldsymbol{d}]\!]_x)$ *with* $a, b \xleftarrow{\$} \mathbb{Z}_q$ *and* $\boldsymbol{k}, \boldsymbol{d} \xleftarrow{\$} \mathbb{Z}_q^{m\times 1}$*;*

- $P_1^{(2)} := ([\![a]\!]_{1,2}, [\![b]\!]_{1,2}, [\![a\boldsymbol{k}]\!]_{1,2}, [\![b\boldsymbol{d}]\!]_{1,2}, [\![\boldsymbol{k}' + \boldsymbol{d}]\!]_x)$ *with* $a, b \xleftarrow{\$} \mathbb{Z}_q$ *and* $\boldsymbol{k}, \boldsymbol{k}', \boldsymbol{d} \xleftarrow{\$} \mathbb{Z}_q^{m\times 1}$,

satisfying $\sum_{i=1}^m k_i = \sum_{i=1}^m k'_i = t$, *where* k_i, k'_i *are the i-th elements of* $\boldsymbol{k}$ *and* $\boldsymbol{k}'$, *respectively. For any p.p.t.* $\mathcal{A}$, *there holds*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}(P_0^{(2)})\right] - \Pr\left[1 \leftarrow \mathcal{A}(P_1^{(2)})\right]\right| \leq (m-1)\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda).$$

Under the XDLin assumption, for any p.p.t. adversary, the correlation in $P_0^{(2)}$ remains computationally indistinguishable from the randomized structure in $P_1^{(2)}$, even with the sum constraint $\sum_{i=1}^m k_i = \sum_{i=1}^m k'_i = t$. The proof in Appendix B demonstrates that the distinguishing advantage is bounded by $(m-1)\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda)$, which is also negligible.

Based on Proposition 2, the following Proposition 3 provides $\mathcal{A}$ with an additional query capability.

Proposition 3 *Assume the XDLin assumption holds for* $\mathcal{G}$. *For* $m = \texttt{poly}(\lambda) \geq 1$ *and any* $t \in \mathbb{Z}_q$, *consider the following game* $(\triangle)$:

- $\mathcal{A}$ *receives the distributions* $P_0^{(2)}$ *and* $P_1^{(2)}$ *as defined in Proposition 2.*
- $\mathcal{A}$ *uniformly samples elements* $\{r_1, \cdots, r_m\}$ *from the group* $\mathbb{G}_x$, *where* $x \in \{1, 2\}$, *and sends them to the challenger* $\mathcal{C}$. *Then,* $\mathcal{C}$ *returns* $\{r_1^a, \cdots, r_m^a\}$ *to* $\mathcal{A}$.

For any p.p.t. $\mathcal{A}$, *it holds that*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}^{(\triangle)}(P_0^{(2)})\right] - \Pr\left[1 \leftarrow \mathcal{A}^{(\triangle)}(P_1^{(2)})\right]\right| \leq (m-1)\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda).$$

The proof of Proposition 3 is presented in Appendix C. We can observe that Proposition 3 involves additional interactions on the basis of Proposition 2. The adversary's ability to query the challenger with elements from $\mathbb{G}_x$ and receive their images under exponentiation by a simulates limited oracle access. Under the XDLin assumption, Proposition 3 demonstrates that $\mathcal{C}$'s response does not leak additional information that would aid $\mathcal{A}$ in distinguishing the two distributions $P_0^{(2)}$ and $P_1^{(2)}$.

Furthermore, the following Proposition 4 establishes the equivalence between two interactive games, denoted as $(\triangle)$ and $(\diamond)$.

Proposition 4 *The games* $(\triangle)$ *and* $(\diamond)$ *are the same in procedures except for: in* $(\diamond)$

- *After* $\mathcal{A}$ *receives* $P_0^{(2)}$ *and* $P_1^{(2)}$, *as defined in Proposition 2,* $\mathcal{C}$ *sends the pair* $\left\{(r_1, r_1^{\frac{1}{a}}), \cdots, (r_m, r_m^{\frac{1}{a}})\right\}$ *to* $\mathcal{A}$, *where* $\{r_1, \cdots, r_m\}$ *are uniformly sampled elements of* $\mathbb{G}_x$.

For any p.p.t. $\mathcal{A}$, *it holds that*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}^{(\diamond)}(P_0^{(2)})\right] - \Pr\left[1 \leftarrow \mathcal{A}^{(\diamond)}(P_1^{(2)})\right]\right| \leq (m-1)\mathrm{Adv}_{\mathcal{A}}^{\mathrm{XDLin}}(\lambda).$$

The proof of Proposition 4 is presented in Appendix D. Proposition 4 demonstrates that the specific modification to $\mathcal{C}$'s behavior in $(\diamond)$ does not affect $\mathcal{A}$'s ability to distinguish between distributions $P_0^{(2)}$ and $P_1^{(2)}$.

4.2 A Theoretical Result of Indistinguishability Security

Following [43, 44], provable security can be established through the formalization of security experiments that test the security of cryptographic schemes. These experiments typically simulate interactions between a challenger and an adversary: the challenger generates keys and responds to the adversary's queries, such as encryption, decryption, or signing, while the adversary attempts to break the scheme under a specific attack model, such as chosen plaintext or ciphertext attacks. The core of the experiment is to quantify the adversary's probability of distinguishing between experiments and to prove that this advantage is negligible by reducing it to the hardness of a mathematical problem. These experiments can provide mathematically grounded assurances that the protocol can withstand both theoretical scrutiny and practical cryptanalysis.

In the following, we first define the security experiments for our protocol. We say that the protocol is indistinguishably secure if and only if no non-uniform p.p.t. adversary $\mathcal{A}$ can distinguish between the outputs of any two permutations. To simplify the exposition without loss of generality, we demonstrate that no non-uniform p.p.t. adversary $\mathcal{A}$ can distinguish between any permutation $\boldsymbol{\pi}$ and the identity mapping. Given that any permutation $\boldsymbol{\pi}$ is indistinguishable from the identity mapping, indistinguishability between any two permutations immediately follows. Hence, we consider the following two experiments indexed by $b \in \{0, 1\}$. When $b = 0$, the permutation $\boldsymbol{\pi}$ is applied. When $b = 1$, the identity mapping is applied. In $\mathbf{Exp}^{(b)}$:

- $\mathcal{A}$ interacts with $\mathcal{C}$ to generate a permutation $\boldsymbol{\pi} \in S_m$, which is subsequently extended to a permutation in S_n by setting $\boldsymbol{\pi}(j) = j$ for all $j > m$. Then, $\mathcal{A}$ sends $\boldsymbol{\pi}_0 = \boldsymbol{\pi}$, and $\boldsymbol{\pi}_1 = \mathrm{id} : i \to i$ to $\mathcal{C}$.
- $\mathcal{C}$ selects $b \xleftarrow{\$} \{0, 1\}$ uniformly at random. Then, $\mathcal{C}$ runs the setup algorithm $(\boldsymbol{tk}, \langle \boldsymbol{tk}, \boldsymbol{ct}(0)\rangle) \leftarrow \mathtt{Setup}(1^\lambda, n)$, for $i \in \mathcal{H}_S$,
$$\boldsymbol{tk}_i^j = \begin{cases} [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i \delta_{j,\boldsymbol{\pi}(i)}, 0)\, R_i]\!]_2, & b = 0 \\ [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i \delta_{j,i}, 0)\, R_i]\!]_2, & b = 1 \end{cases}.$$
$\mathcal{C}$ sends $\boldsymbol{tk}$ and $\langle \boldsymbol{tk}, \boldsymbol{ct}(0)\rangle$ to $\mathcal{A}$.
- $\mathcal{A}$ makes $Q\ (= \mathtt{poly}(\lambda))$ queries. In the t-th query, $\mathcal{A}$ selects a set of plaintexts $\{x_{i,t}\}_{i<m}$ and sends them to $\mathcal{C}$. Then, $\mathcal{C}$ computes
$$\boldsymbol{ct}_{i,t} = \begin{cases} [\![C_i \left(K_i(t), \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0\right)^\top]\!]_1, & b = 0 \\ [\![C_i \left(K_i(t), \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{\boldsymbol{\pi}^{-1}(i),t}, 0\right)^\top]\!]_1, & b = 1 \end{cases}.$$
and returns $(\boldsymbol{ct}_{1,t}, \cdots, \boldsymbol{ct}_{m,t})$ to $\mathcal{A}$.
- By checking the obtained ciphertexts, $\mathcal{A}$ outputs 0 or 1.

By proving that $\mathbf{Exp}^{(0)}$ and $\mathbf{Exp}^{(1)}$ are computationally indistinguishable, we can conclude the indistinguishability security of our protocol, and the following Theorem 1 holds.

Theorem 1 (Indistinguishability Security) *Suppose the pseudorandom function family* $\mathtt{PRF}$ *is secure and the XDLin assumption holds for* $\mathcal{G}$*, it holds that*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Exp}^{(0)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Exp}^{(1)}\right)\right]\right| \\ \leq 2\Big[\frac{m(m+1) + Qm}{q} + \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CPRF}}(\lambda) + \left[Q(m^2 + m - 2) + m^2\right] \mathrm{Adv}^{\mathrm{XDLin}}(\lambda)\Big].$$

Remark 3 *We note that the number of honest senders,* m*, and the number of adversary's queries,* Q*, are both polynomial in* λ*, whereas* q *is typically a large prime that can be exponential in* λ*. Hence, the term* $\frac{m(m+1)+Qm}{q}$ *becomes negligible as* λ *increases. For the XDLin-related term* $\left[Q(m^2 + m - 2) + m^2\right] \mathrm{Adv}^{\mathrm{XDLin}}(\lambda)$*, since the coefficient* $Q(m^2 + m - 2) + m^2$ *is polynomial and* $\mathrm{Adv}^{\mathrm{XDLin}}(\lambda)$ *is negligible under the XDLin assumption, the entire term remains negligible. Combined with the security of the PRF,* $\mathrm{Adv}^{\mathrm{CPRF}}(\lambda)$ *is also negligible. Therefore, we can conclude that equation (**??**) holds.*

Remark 4 *Theorem 1 illustrates that the adversary's distinguishing advantage between* $\boldsymbol{Exp}^{(0)}$ *and* $\boldsymbol{Exp}^{(1)}$ *is negligible, indicating that no p.p.t. adversary can effectively distinguish honest senders with non-negligible probability. Through formal proofs, we provide a provable security guarantee for the proposed protocol. Provable security has long been a fundamental challenge in privacy-preserving communication networks, remaining unresolved due to intrinsic cryptographic complexity. Our results fundamentally demonstrate the theoretical feasibility of constructing provably secure anonymous networks, addressing key challenges in simultaneously ensuring computational efficiency and rigorous security certification.*

Next, we present a comprehensive and detailed proof of Theorem 1. The outline of the proof is described in Fig. 6.

Proof 1 (Proof of Theorem 1) *Following the framework of security experiments, we construct several sequences of hybrid experiments to gradually establish a switch between* $\boldsymbol{Exp}^{(0)}$ *and* $\boldsymbol{Exp}^{(1)}$*. By proving that each consecutive pair of hybrid experiments is computationally indistinguishable, we can obtain the indistinguishability of* $\boldsymbol{Exp}^{(0)}$ *and* $\boldsymbol{Exp}^{(1)}$*. We now introduce the detailed hybrid experiments.*

(1) Hybrid Experiment $\boldsymbol{Hyb}^{(0)}$***:***

First, we define $\boldsymbol{Hyb}^{(0)}$*, which is identical to* $\boldsymbol{Exp}^{(0)}$ *except for the calculation of* $(\boldsymbol{ct}_{1,t}, \cdots, \boldsymbol{ct}_{m,t})$*. In* $\boldsymbol{Hyb}^{(0)}$*, when processing the* t*-th* $\mathtt{Enc}$ *query, the challenger* $\mathcal{C}$ *replaces the element* $K_i(t)$ *in equation (3) with a uniformly random*

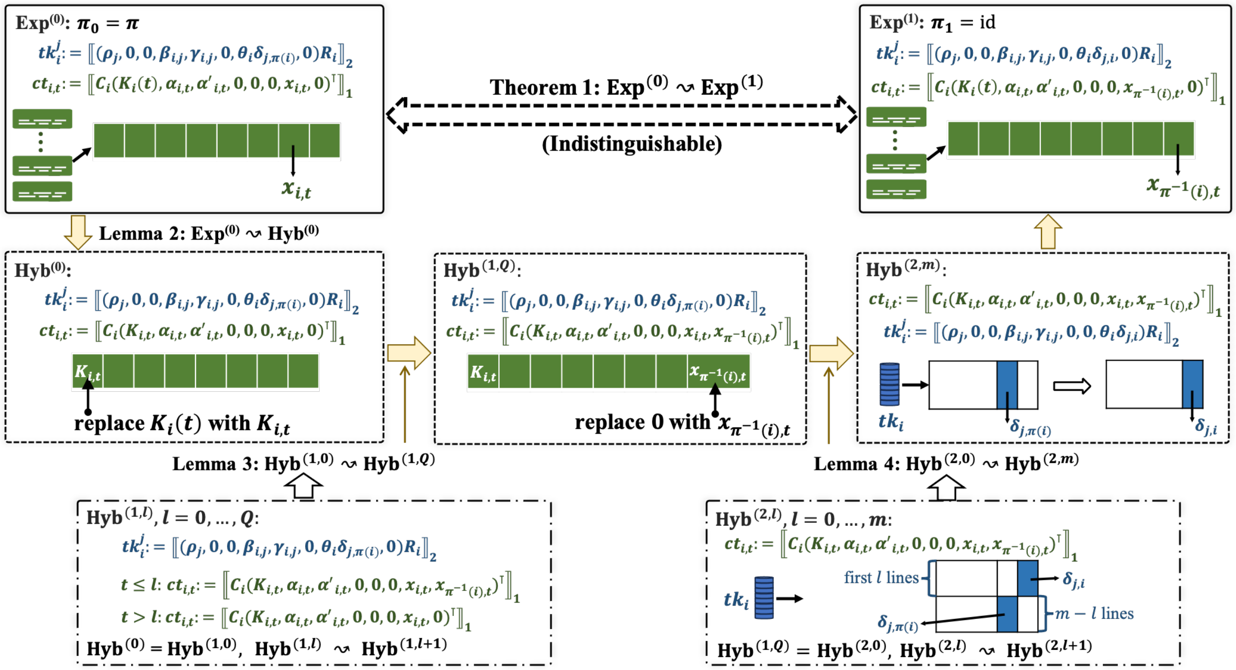
$\mathbf{Exp}^{(0)}$: $\pi_0 = \pi$
$tk_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i\delta_{j,\pi(i)}, 0)R_i]\!]_2$
$ct_{i,t} := [\![C_i(K_i(t), \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0)^\top]\!]_1$
$x_{i,t}$
Theorem 1: $\mathbf{Exp}^{(0)} \rightsquigarrow \mathbf{Exp}^{(1)}$
(Indistinguishable)
$\mathbf{Exp}^{(1)}$: $\pi_1 = \mathrm{id}$
$tk_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i\delta_{j,i}, 0)R_i]\!]_2$
$ct_{i,t} := [\![C_i(K_i(t), \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{\pi^{-1}(i),t}, 0)^\top]\!]_1$
$x_{\pi^{-1}(i),t}$
Lemma 2: $\mathbf{Exp}^{(0)} \rightsquigarrow \mathbf{Hyb}^{(0)}$
$\mathbf{Hyb}^{(0)}$:
$tk_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i\delta_{j,\pi(i)}, 0)R_i]\!]_2$
$ct_{i,t} := [\![C_i(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0)^\top]\!]_1$
$K_{i,t}$
replace $K_i(t)$ with $K_{i,t}$
$\mathbf{Hyb}^{(1,Q)}$:
$tk_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i\delta_{j,\pi(i)}, 0)R_i]\!]_2$
$ct_{i,t} := [\![C_i(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, x_{\pi^{-1}(i),t})^\top]\!]_1$
$K_{i,t}$
$x_{\pi^{-1}(i),t}$
replace 0 with $x_{\pi^{-1}(i),t}$
$\mathbf{Hyb}^{(2,m)}$:
$ct_{i,t} := [\![C_i(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, x_{\pi^{-1}(i),t})^\top]\!]_1$
$tk_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, 0, \theta_i\delta_{j,i})R_i]\!]_2$
tk_i
$\delta_{j,\pi(i)}$
$\delta_{j,i}$
Lemma 3: $\mathbf{Hyb}^{(1,0)} \rightsquigarrow \mathbf{Hyb}^{(1,Q)}$
Lemma 4: $\mathbf{Hyb}^{(2,0)} \rightsquigarrow \mathbf{Hyb}^{(2,m)}$
$\mathbf{Hyb}^{(1,l)}$, $l = 0, \dots, Q$:
$tk_i^j := [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i\delta_{j,\pi(i)}, 0)R_i]\!]_2$
$t \le l$: $ct_{i,t} := [\![C_i(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, x_{\pi^{-1}(i),t})^\top]\!]_1$
$t > l$: $ct_{i,t} := [\![C_i(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0)^\top]\!]_1$
$\mathbf{Hyb}^{(0)} = \mathbf{Hyb}^{(1,0)}$, $\mathbf{Hyb}^{(1,l)} \rightsquigarrow \mathbf{Hyb}^{(1,l+1)}$
$\mathbf{Hyb}^{(2,l)}$, $l = 0, \dots, m$:
$ct_{i,t} := [\![C_i(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, x_{\pi^{-1}(i),t})^\top]\!]_1$
tk_i
first l lines
$\delta_{j,i}$
$m - l$ lines
$\delta_{j,\pi(i)}$
$\mathbf{Hyb}^{(1,Q)} = \mathbf{Hyb}^{(2,0)}$, $\mathbf{Hyb}^{(2,l)} \rightsquigarrow \mathbf{Hyb}^{(2,l+1)}$

Figure 6: The outline of Theorem 1's proof.

*element $K_{i,t} \xleftarrow{\$} \mathbb{Z}_q$, subject to the constraint $\sum_{i=1}^m K_{i,t} = \sum_{i=1}^m K_i(t)$. This leads to the following Lemma 2, which shows the indistinguishability of **Exp**$^{(0)}$ and **Hyb**$^{(0)}$.*

***Lemma 2** Suppose the pseudorandom function family* PRF *is secure, then for any non-uniform p.p.t. $\mathcal{A}$, **Exp**$^{(0)}$ and **Hyb**$^{(0)}$ are computationally indistinguishable. Formally,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Exp}^{(0)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Hyb}^{(0)}\right)\right]\right| \le \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CPRF}}(\lambda).$$

The proof of Lemma 2 can be easily derived from the security of CPRF and Lemma 1 introduced in Subsection 2.3.

(2) Hybrid Experiments Hyb**$^{(1,\ell)}, \ell = 0, 1, \cdots, Q$**:

*Then, we define a sequence of hybrid experiments **Hyb**$^{(1,\ell)}$ for $\ell = 0, 1, \cdots, Q$. The difference between **Hyb**$^{(1,\ell)}$ and **Hyb**$^{(0)}$ lies in the last element in the calculation of $\boldsymbol{ct}_{i,t}$ in equation (3). In **Hyb**$^{(1,\ell)}$,*

- *for queries with $t > \ell$, $\mathcal{C}$ computes $\boldsymbol{ct}_{i,t}$ by*

$$\boldsymbol{ct}_{i,t} = [\![C_i\left(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, 0\right)^\top]\!]_1;$$

- *for queries with $t \le \ell$, $\mathcal{C}$ computes $\boldsymbol{ct}_{i,t}$ by*

$$\boldsymbol{ct}_{i,t} = [\![C_i\left(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, 0, x_{i,t}, x_{\boldsymbol{\pi}^{-1}(i),t}\right)^\top]\!]_1.$$

*Note that **Hyb**$^{(1,0)}$ is identical to **Hyb**$^{(0)}$. The following Lemma 3 illustrates that **Hyb**$^{(1,Q)}$ and **Hyb**$^{(1,0)}$ are computationally indistinguishable.*

***Lemma 3** Suppose the pseudorandom function family* PRF *is secure and the XDLin assumption holds for $\mathcal{G}$, then for any non-uniform p.p.t. $\mathcal{A}$, **Hyb**$^{(1,0)}$ and **Hyb**$^{(1,Q)}$ are computationally indistinguishable. Formally,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Hyb}^{(1,0)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Hyb}^{(1,Q)}\right)\right]\right| \le 2Q\left[\frac{m}{q} + (m-1)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda)\right].$$

The proof of Lemma 3 also employs a sequence of hybrid experiments, with detailed steps provided in Appendix E.

We briefly denote $\boldsymbol{Hyb}^{(1,Q)}$ *as* $\boldsymbol{Hyb}^{(2)}$*. Subsequently, we define the following sequence of* $\boldsymbol{Hyb}^{(2,\ell)}$ *for* $\ell = 0, 1, \cdots, m$*.*

(3) Hybrid Experiments $\boldsymbol{Hyb}^{(2,\ell)}, \ell = 0, 1, \cdots, m$***:***

The difference between $\boldsymbol{Hyb}^{(2,\ell)}$ *and* $\boldsymbol{Hyb}^{(2)}$ *lies in the last two elements in the calculation of* $\boldsymbol{tk}_i^j$ *in equation (2). In* $\boldsymbol{Hyb}^{(2,\ell)}$*, for* $1 \leq i \leq m$*,*

- *for* $1 \leq j \leq \ell$*,*
$$\boldsymbol{tk}_i^j = [\![(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, 0, \theta_i \delta_{j,i})\, R_i]\!]_2;$$

- *for* $\ell + 1 \leq j \leq m$*,*
$$\boldsymbol{tk}_i^j = [\![\big(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i \delta_{j,\boldsymbol{\pi}(i)}, 0\big)\, R_i]\!]_2.$$

Note that $\boldsymbol{Hyb}^{(2,0)}$ *is identical to* $\boldsymbol{Hyb}^{(2)}$*. We further demonstrate the indistinguishability of* $\boldsymbol{Hyb}^{(2,m)}$ *and* $\boldsymbol{Hyb}^{(2,0)}$ *by the following Lemma 4.*

Lemma 4 *Suppose the pseudorandom function* PRF *is secure and the XDLin assumption holds for* $\mathcal{G}$*, then for any non-uniform p.p.t.* $\mathcal{A}$*,* $\boldsymbol{Hyb}^{(2,0)}$ *and* $\boldsymbol{Hyb}^{(2,m)}$ *are computationally indistinguishable. Formally,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Hyb}^{(2,0)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\boldsymbol{Hyb}^{(2,m)}\right)\right]\right| \leq 2m\left[\frac{m+1}{q} + (Qm + m - Q)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda)\right].$$

The proof of Lemma 4 is deferred to the Appendix F.

Combining Lemmas 2, 3, and 4, we can conclude that Theorem 1 holds, thereby establishing the indistinguishability of $\boldsymbol{Exp}^{(0)}$ *and* $\boldsymbol{Exp}^{(1)}$*.*

Theorem 1 establishes a formal bound on the adversary's distinguishing advantage between $\mathbf{Exp}^{(0)}$ and $\mathbf{Exp}^{(1)}$. This bound quantifies the security guarantees of our protocol, demonstrating that the adversary's ability to distinguish the two experiments is negligible, given that the employed PRF is secure and the XDLin assumption holds. Hence, our protocol's untraceability and unlinkability can be rigorously proven via this indistinguishability result.

5 Simulations

In this section, we present some simulations and tests for our proposed protocol. First, we describe the experimental environment and specific configurations. Subsequently, we provide the communication efficiency evaluation results, focusing on two critical metrics: encryption/decryption speed and ciphertext transmission efficiency. Additionally, detailed discussion and analysis of the test results are provided to validate the performance of our protocol.

5.1 Experimental Configuration

The simulation experiments were conducted on a virtual machine running Ubuntu 22.04, equipped with GCC 11.4.0, 2 GB of memory, and an Intel i5-8265U processor operating at 1.60 GHz. Cryptographic primitives were instantiated using the BLS12-381 curve, a pairing-friendly elliptic curve, which guarantees 128-bit security under standardized cryptographic assumptions. All bilinear pairing operations and elliptic curve arithmetic were implemented with the RELIC library, a high-performance cryptographic toolkit optimized for modular exponentiation and pairing-based protocols.

5.2 Results and Analysis

The security of our protocol is rigorously demonstrated through formal proofs in Section 4. Since operational performance is crucial for the real-world deployment of cryptographic algorithms, this section focuses on evaluating the communication efficiency of the proposed protocol and demonstrating its practical applicability. To be more specific, the communication efficiency is assessed through two primary metrics, encryption and decryption speeds, as well as transmission efficiency. The detailed results are presented below.

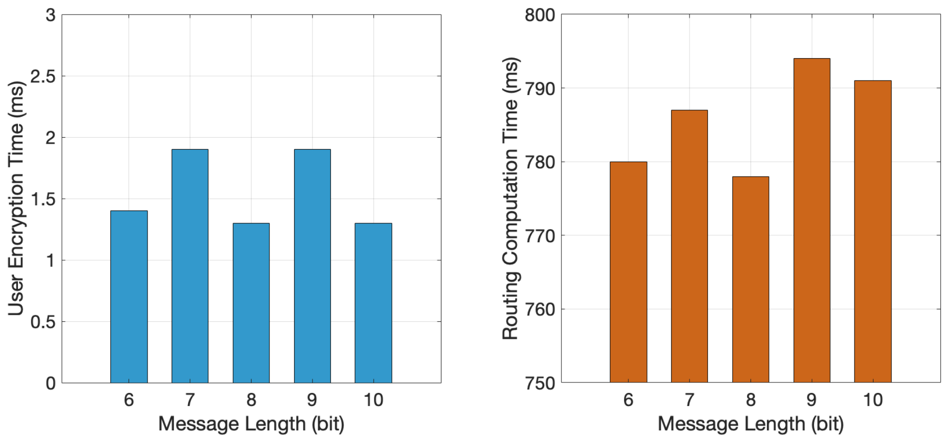

Figure 7: Encryption and routing computation time for 10 users with varying message lengths.

5.2.1 Encryption and Decryption Speeds

To comprehensively investigate the encryption and decryption performance, two sets of tests are conducted under varying conditions.

- Test 1 is designed to evaluate the impact of *message length* and is therefore performed with a *fixed number of users* while *varying the message length*.
- Test 2 is designed to assess the impact of *the number of users* and is therefore conducted with a *varying number of users* while *keeping the message length fixed*.

Test 1: In this set of tests, the number of users is fixed at 10, while the message length per communication is varied across $6 - 10$ bits. The results are summarized in TABLE 1 and Fig. 7.

Table 1: Encryption and routing computation time for 10 users with varying message lengths.

Message Length (bit)	User Encryption Time (ms)	Routing Computation Time (ms)
6	1.4	780
7	1.9	787
8	1.3	778
9	1.9	794
10	1.3	791

Note: The above results are based on single-threaded benchmark testing. According to the protocol design, multi-threaded implementations are supported. In practical deployments, there is significant potential for improving computational efficiency through methods such as parallel computing, GPU acceleration, and memory expansion..

As observed from TABLE 1 and Fig. 7, the user encryption time exhibits minor fluctuations, ranging from 1.3 to 1.9 ms, indicating that it is largely independent of message length within this range. Similarly, the routing computation time exhibits only minor fluctuations between 778 and 791 ms, suggesting stability across the tested message lengths. These results demonstrate that the computational cost associated with encryption and routing does not scale significantly with small increases in message length. Furthermore, a larger routing memory, capable of storing more extensive exponent tables for $g_T^{\theta_i}$, could facilitate the transmission of longer messages per communication, potentially improving efficiency.

Test 2: In the second set of tests, the message length is fixed at 8 bits per user, while the number of users varies from 5 to 25. The results are presented in TABLE 2 and Fig. 8.

From TABLE 2 and Fig. 8, we can similarly conclude that the encryption time per user is observed to remain stable, ranging between 1.3 and 1.6 ms, further confirming its independence from the number of users. In contrast, the routing computation time exhibits polynomial growth with an increasing number of users, rising from 204 ms for 5 users to 4833 ms for 25 users. This suggests that while individual encryption operations are efficient and scalable, the routing process becomes a performance bottleneck as the system scales to support more users. This issue similarly exists in the

Table 2: Encryption and routing computation time for varying numbers of users with a fixed 8-bit message length.

User Number	User Encryption Time (ms)	Routing Computation Time (ms)
5	1.5	204
10	1.3	778
15	1.6	1733
20	1.6	3080
25	1.6	4833

Note: The above results are based on single-threaded benchmark testing. According to the protocol design, multi-threaded implementations are supported. In practical deployments, there is significant potential for improving computational efficiency through methods such as parallel computing, GPU acceleration, and memory expansion.

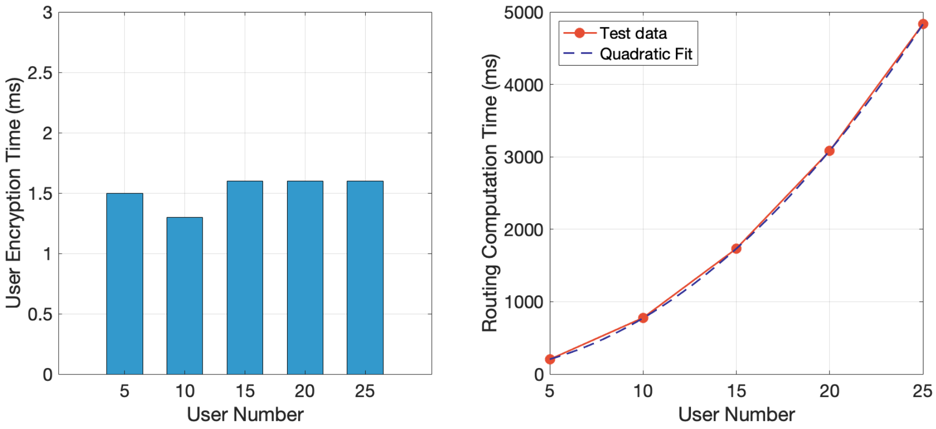

Figure 8: Encryption and routing computation time for varying numbers of users with a fixed 8-bit message length.

NIAR scheme. Nevertheless, the decentralized design of our protocol allows this limitation to be mitigated through integration with an efficient grouping strategy, which will be the focus of our future work.

Compared to the NIAR scheme in [31], given that each sender transmits a message of b bits and each message spans l bytes, our protocol requires $4nb/l$ group multiplications during the encryption process, whereas the NIAR scheme requires $17nb$ group multiplications. Regarding the routing computation time, the primary computational cost in the decryption phase arises from bilinear pairing operations. We therefore compare the number of bilinear pairings required by each scheme. The NIAR scheme incurs $64n^2b$ pairing operations, while our protocol requires only n^2b/l pairings, representing a significant reduction in computational cost, especially as l increases.

5.2.2 Transmission Efficiency

According to equation (3), each encryption operation in our protocol generates a ciphertext represented as a 1×8 vector in $\mathbb{G}_1$. Since each element in $\mathbb{G}_1$ requires 49 bytes of storage, the total ciphertext size per user amounts to 392 bytes. For a plaintext message of length l bytes per user, the resulting plaintext-to-ciphertext expansion factor is calculated as $392/l$.

In contrast, the NIAR construction in [31] encrypts each bit into a single $\mathtt{MCFE}^{ffh}$ ciphertext, as outlined in the $\mathtt{MCFE}^{ffh}$ framework. An $\mathtt{MCFE}^{ffh}$ ciphertext consists of an $\mathtt{MCFE}$ ciphertext combined with the output of $\mathtt{FE.KGEN}$. According to the $\mathtt{MCFE}$ construction, an $\mathtt{MCFE}$ ciphertext is composed of $[\![c_1]\!]$, $[\![c_2]\!]$ and $[\![\tilde{c}]\!]$, with respective lengths of $2b$, 2, and 2 group elements, resulting in a total length of $2b + 4$. Here, $b = 1$ when encrypting bit-by-bit. Additionally, the output of $\mathtt{FE.KGEN}$ has a length of 2 group elements, resulting in a total $\mathtt{MCFE}^{ffh}$ ciphertext length of $2b + 6$. For a single plaintext bit ($b = 1$), the resulting ciphertext thus contains 8 group elements in $\mathbb{G}_1$. Using the provided test curve parameters, where each $\mathbb{G}_1$ element occupies 392 bytes, the ciphertext size is calculated to be 3136 bytes, yielding an expansion ratio of 3136 for a 1-bit plaintext. This is markedly higher than the expansion factor of our proposed protocol.

In summary, our protocol demonstrates significantly greater communication efficiency than the NIAR scheme, both in terms of ciphertext expansion and computational cost, offering substantial improvements for practical applications.

6 Conclusion

Private and covert communication is crucial in meeting the ever-evolving security demands of the interconnected digital era. Anonymous routing technology emerges as a pivotal tool, aiming to obfuscate communication paths and participant identities, and resist sophisticated traffic analysis attacks. This paper proposes a decentralized anonymous communication network protocol that fundamentally eliminates dependence on threshold trust models and trusted setups, which are common vulnerabilities in existing systems. The protocol's resistance to analysis and traceability is validated through formal security proofs, which remains inherently robust in the face of evolving adversarial strategies. Furthermore, simulation results demonstrate that the proposed protocol is also efficient in practical deployment.

Several issues remain to be addressed in future work. First, the proposed protocol addresses the identity protection of trusted senders. In future work, we will further explore integrating Private Information Retrieval methods to achieve anonymous protection for receivers, thereby realizing bidirectional identity untraceability. Second, the protocol's applicable scale is currently limited. However, its decentralized nature allows for integration with an efficient grouping strategy, thereby further optimizing the computational efficiency of routing computation and enhancing the protocol's dynamic adaptability.

A Proof of Proposition 1

Proposition 1 can be considered as a vector version of the XDLin assumption. We begin by proving the case for $m = 2$ via the contradiction method.

For $m = 2$, we have

$$P_0 = \left(\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket \begin{pmatrix} ak_1 \\ ak_2 \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} bd_1 \\ bd_2 \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} k_1 + d_1 \\ k_2 + d_2 \end{pmatrix} \rrbracket_x \right), a, b, k_1, k_2, d_1, d_2 \xleftarrow{\$} \mathbb{Z}_q;$$

$$P_1 = \left(\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket \begin{pmatrix} ak_1 \\ ak_2 \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} bd_1 \\ bd_2 \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \rrbracket_x \right), a, b, k_1, k_2, d_1, d_2, e_1, e_2 \xleftarrow{\$} \mathbb{Z}_q.$$

Let

$$P_1' := \left(\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket \begin{pmatrix} ak_1 \\ ak_2 \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} bd_1 \\ bd_2 \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} k_1 + d_1 \\ e_1 \end{pmatrix} \rrbracket_x \right), a, b, k_1, k_2, d_1, e_1 \xleftarrow{\$} \mathbb{Z}_q.$$

With the XDLin assumption, the indistinguishability of P_0 and P_1 is equivalent to the indistinguishability of P_1 and P_1'. Suppose there exists a p.p.t. $\mathcal{A}$ such that the advantage $|\Pr[1 \leftarrow \mathcal{A}(P_1)] - \Pr[1 \leftarrow \mathcal{A}(P_1')]|$ is non-negligible. This would imply that $\mathcal{A}$ could distinguish the two distributions defined in Definition 1, violating the XDLin assumption. Therefore, we can conclude that P_1 and P_1' are computationally indistinguishable, establishing Proposition 1 for the case $m = 2$. The argument generalizes to arbitrary m by inductively applying the same reasoning to each additional dimension. Thus, we can conclude that Proposition 1 holds.

B Proof of Proposition 2

Given that $(\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket a\boldsymbol{l} \rrbracket_{1,2}, \llbracket b\boldsymbol{e} \rrbracket_{1,2}, \boldsymbol{y})$, where $a, b \xleftarrow{\$} \mathbb{Z}_q$, $\boldsymbol{l}, \boldsymbol{e} \xleftarrow{\$} \mathbb{Z}_q^{(m-1)\times 1}$, and $\boldsymbol{y} = \llbracket \boldsymbol{l} + \boldsymbol{e} \rrbracket_x$ or $\llbracket \boldsymbol{l}' + \boldsymbol{e} \rrbracket_x$, $\boldsymbol{l}' \xleftarrow{\$} \mathbb{Z}_q^{(m-1)\times 1}$. Denote the i-th element of $\boldsymbol{l}$, $\boldsymbol{e}$, $\boldsymbol{y}$ by l_i, e_i, y_i, respectively. Let

$$Q := \Big(\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket a\boldsymbol{l} \| a(t - \sum_{i=1}^{m-1} l_i) \rrbracket_{1,2}, \llbracket b\boldsymbol{e} \| b(r - \sum_{i=1}^{m-1} e_i) \rrbracket_{1,2}, \boldsymbol{y} \| (t + r - \sum_{i=1}^{m-1} y_i) \Big), r \xleftarrow{\$} \mathbb{Z}_q.$$

We note that when $\boldsymbol{y} = \llbracket \boldsymbol{l} + \boldsymbol{e} \rrbracket_x$, Q is equivalent to P_0 and when $\boldsymbol{y} = \llbracket \boldsymbol{l}' + \boldsymbol{e} \rrbracket_x$, Q is equivalent to P_1. Meanwhile, condition $\sum_{i=1}^{m} k_i = t = \sum_{i=1}^{m} k_i'$ in Proposition 2 is satisfied. Therefore, by Proposition 1, Proposition 2 can be directly derived.

C Proof of Proposition 3

The indistinguishability of the target games in Proposition 3 is formally established via a security reduction to the game variant $(\bigtriangledown)$, defined as:

- $\mathcal{A}$ receives

- $P_0^{(2)} := (\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket a\boldsymbol{k} \rrbracket_{1,2}, \llbracket b\boldsymbol{d} \rrbracket_{1,2}, \llbracket \boldsymbol{k}+\boldsymbol{d} \rrbracket_x)$, where $a, b \xleftarrow{\$} \mathbb{Z}_q$ and $\boldsymbol{k}, \boldsymbol{d} \xleftarrow{\$} \mathbb{Z}_q^{m\times 1}$,
- $P_1^{(2)} := (\llbracket a \rrbracket_{1,2}, \llbracket b \rrbracket_{1,2}, \llbracket a\boldsymbol{k} \rrbracket_{1,2}, \llbracket b\boldsymbol{d} \rrbracket_{1,2}, \llbracket \boldsymbol{k}'+\boldsymbol{d} \rrbracket_x)$, where $a, b \xleftarrow{\$} \mathbb{Z}_q$ and $\boldsymbol{k}, \boldsymbol{k}', \boldsymbol{d} \xleftarrow{\$} \mathbb{Z}_q^{m\times 1}$,

satisfying $\sum_{i=1}^m k_i = \sum_{i=1}^m k_i' = t$, where k_i, k_i' are the i-th elements of $\boldsymbol{k}$ and $\boldsymbol{k}'$, respectively.

- $\mathcal{A}$ uniformly samples elements $\{z_1, \cdots, z_m\}$ from $\mathbb{Z}_q$, and sends them to $\mathcal{C}$. Subsequently, $\mathcal{C}$ returns $\{\llbracket az_1 \rrbracket_x, \cdots, \llbracket az_m \rrbracket_x\}$ to $\mathcal{A}$.

We note that $\mathcal{A}$ gains no computational advantage in extracting the secret exponent a, since any efficient extraction of a would imply breaking the discrete logarithm problem. Combining with Proposition 2, we can conclude that for any p.p.t. $\mathcal{A}$, the two distributions $P_0^{(2)}$ and $P_1^{(2)}$ in game variant ($\bigtriangledown$) are computationally indistinguishable. Meanwhile, game ($\bigtriangleup$) exhibits strictly greater computational hardness than ($\bigtriangledown$), as the adversary has access to more information in ($\bigtriangledown$). Therefore, $P_0^{(2)}$ and $P_1^{(2)}$ remain computationally indistinguishable in game ($\bigtriangleup$), which confirms that Proposition 3 holds.

D Proof of Proposition 4

First, we observe that the adversarial interaction where $\mathcal{A}$ uniformly samples elements $\{r_1, \cdots, r_m\}$ from the group $\mathbb{G}_x$ and submits them to $\mathcal{C}$, who then returns $\{r_1^a, \cdots, r_m^a\}$, is computationally indistinguishable from the scenario where $\mathcal{C}$ directly transmits the pairs $\{(r_1, r_1^a), \cdots, (r_m, r_m^a)\}$ to $\mathcal{A}$. The equivalence comes from the fact that the information ultimately obtained by $\mathcal{A}$ in both interactions is identically distributed, provided that the selection of $\{r_1, \cdots, r_m\}$ is random and cannot be controlled by $\mathcal{A}$. Additionally, based on the discrete logarithm problem, transmitting the pairs $\{(r_1, r_1^a), \cdots, (r_m, r_m^a)\}$ to $\mathcal{A}$ is equivalent to transmitting the pairs $\left\{(r_1, r_1^{\frac{1}{a}}), \cdots, (r_m, r_m^{\frac{1}{a}})\right\}$. The equivalence lies in that $\mathcal{A}$ cannot efficiently deduce a or a^{-1} from the transmitted elements in both cases. Therefore, Proposition 4 holds.

E Proof of Lemma 3

To prove Lemma 3, it suffices to show that for any $\ell < Q$,

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(1,\ell)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(1,\ell+1)}\right)\right]\right| \leq 2\left[\frac{m}{q} + (m-1)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda)\right]. \tag{5}$$

To this end, we consider the following hybrid experiments to establish a switch between $\mathbf{Hyb}^{(1,\ell)}$ and $\mathbf{Hyb}^{(1,\ell+1)}$.

E.0.1 Hybrid Experiment $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$

$\widetilde{\mathbf{Hyb}}^{(1,\ell)}$ is defined as a modification of $\mathbf{Hyb}^{(1,\ell)}$, and the only difference lies in the calculation of $\{\boldsymbol{ct}_{1,\ell+1}, \cdots, \boldsymbol{ct}_{m,\ell+1}\}$. In $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$, the third-to-last element 0 is replaced by a randomly selected element $\zeta_i \xleftarrow{\$} \mathbb{Z}_q^\times$. Specifically, $\mathcal{C}$ computes $\boldsymbol{ct}_{i,\ell+1}$ by

$$\boldsymbol{ct}_{i,\ell+1} := \llbracket C_i \left(K_{i,\ell+1}, \alpha_{i,\ell+1}, \alpha'_{i,\ell+1}, 0, 0, \zeta_i, x_{i,\ell+1}, 0\right)^\top \rrbracket_1.$$

We can prove the computational indistinguishability of $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$ and $\mathbf{Hyb}^{(1,\ell)}$ through the reduction to the XDLin assumption. Here we demonstrate how to construct a non-uniform p.p.t. adversary $\mathcal{B}$ against the XDLin problem by leveraging the adversary $\mathcal{A}$.

- After $\mathcal{B}$ receives
$$\llbracket \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} a_1\alpha_{1,\ell+1} \\ \vdots \\ a_m\alpha_{m,\ell+1} \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} b_1\alpha'_{1,\ell+1} \\ \vdots \\ b_m\alpha'_{m,\ell+1} \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} \rrbracket_1,$$
it needs to distinguish whether $Y_i = \alpha_{i,\ell+1} + \alpha'_{i,\ell+1}$ or $Y_i = \alpha_{i,\ell+1} + \alpha'_{i,\ell+1} + \zeta_i$.
- $\mathcal{B}$ calls $\texttt{Setup}(1^\lambda, n)$ to obtain $\boldsymbol{tk}$.

- For each $i \in [n]$, $\mathcal{C}$ samples an invertible matrix $W_i \xleftarrow{\$} (\mathbb{Z}_q)^{\times}_{8\times 8}$. Let

$$C_i = W_i \begin{pmatrix} 1 & & & & & & & \\ & a_i & & & & & & \\ & & b_i & & & & & \\ & & & a_i & & & & \\ & & & & a_i & & & \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & a_i & \\ & & & & & & & a_i \end{pmatrix},$$

  and

$$R_i \qquad = \qquad \mathrm{diag}\left(1, \frac{1}{a_i b_i}, \cdots, \frac{1}{a_i b_i}\right) \begin{pmatrix} 1 & & & & & & & \\ & b_i & & & & & & \\ & & a_i & & & & & \\ & & & b_i & & & & \\ & & & & b_i & & & \\ 0 & -b_i & -a_i & 0 & 0 & a_i b_i & 0 & 0 \\ & & & & & & b_i & \\ & & & & & & & b_i \end{pmatrix} W_i^{-1}.$$

  We have $R_i C_i = \mathbf{I}_8$.
  We remark that the j-th token corresponding to sender i is computed by

$$\begin{aligned} \boldsymbol{tk}_i^j &:= [\![\left(\rho_j, 0, 0, a_i b_i \beta_{i,j}, a_i b_i \gamma_{i,j}, 0, a_i b_i \delta_{j,\boldsymbol{\pi}(i)}, 0\right) R_i]\!]_2 \\ &= [\![\left(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \delta_{j,\boldsymbol{\pi}(i)}, 0\right) \begin{pmatrix} 1 & & & & & & & \\ & b_i & & & & & & \\ & & a_i & & & & & \\ & & & b_i & & & & \\ & & & & b_i & & & \\ 0 & -b_i & -a_i & 0 & 0 & a_i b_i & 0 & 0 \\ & & & & & & b_i & \\ & & & & & & & b_i \end{pmatrix} W_i^{-1}]\!]_2 \end{aligned}$$

  Since $\beta_{ij}, \gamma_{ij} \xleftarrow{\$} \mathbb{Z}_q$, we have $a_i b_i \beta_{ij}$ and $a_i b_i \gamma_{ij}$ are still random sampled elements. Here we note that $\theta_i = a_i b_i$.
- $\mathcal{A}$ makes queries $\{x_{i,t}\}_{i \le m}$. When $t = \ell + 1$, $\boldsymbol{ct}_{i,\ell+1}$ is computed by

$$\boldsymbol{ct}_{i,\ell+1} := [\![W_i \left(K_{i,\ell+1}, a_i \alpha_{i,\ell+1}, b_i \alpha'_{i,\ell+1}, 0, 0, Y_i, a_i x_{i,\ell+1}, 0\right)^{\top}]\!]_1.$$

  Otherwise, $\mathcal{B}$ computes the ciphertexts in the same way as in $\mathbf{Hyb}^{(1,\ell)}$, and returns them to $\mathcal{A}$.
- $\mathcal{B}$ outputs the same guess that $\mathcal{A}$ outputs.

Observe that if $Y_i = \alpha_{i,\ell+1} + \alpha'_{i,\ell+1}$, then $\mathcal{A}$'s view is identically distributed as in $\mathbf{Hyb}^{(1,\ell)}$; otherwise, if $Y_i = \alpha_{i,\ell+1} + \alpha'_{i,\ell+1} + \zeta_i$, $\mathcal{A}$'s view is identically distributed as in $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$. By Proposition 2, we can obtain

$$\left| \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(1,\ell)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\widetilde{\mathbf{Hyb}}^{(1,\ell)}\right)\right] \right| \le \frac{m}{q} + (m-1)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda).$$

Therefore, the experiments $\mathbf{Hyb}^{(1,\ell)}$ and $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$ are computationally indistinguishable. Next, we aim to demonstrate that $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$ is equivalent to the following experiment $(*)$.

E.0.2 Hybrid Experiment $(*)$

The difference between games $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$ and $(*)$ lies in the calculation of $\boldsymbol{ct}_{i,\ell+1}$. In $(*)$,

$$\boldsymbol{ct}_{i,\ell+1} := [\![C_i \left(K_{i,\ell+1}, \alpha_{i,\ell+1}, \alpha'_{i,\ell+1}, 0, 0, \zeta_i, x_{i,\ell+1}, x_{\boldsymbol{\pi}^{-1}(i),\ell+1}\right)^{\top}]\!]_1.$$

By computing the product of

$$C_i \begin{pmatrix} \boldsymbol{I}_5 & \\ & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{x_{\boldsymbol{\pi}^{-1}(i),\ell+1}}{\zeta_i} & 0 & 1 \end{pmatrix} \end{pmatrix}$$

and $\left(K_{i,\ell+1}, \alpha_{i,\ell+1}, \alpha'_{i,\ell+1}, 0, 0, \zeta_i, x_{i,\ell+1}, 0\right)^\top$, we observe that the ciphertext in the exponent g_1 remains

$$C_i \left(K_{i,\ell+1}, \alpha_{i,\ell+1}, \alpha'_{i,\ell+1}, 0, 0, \zeta_i, x_{i,\ell+1}, x_{\boldsymbol{\pi}^{-1}(i),\ell+1}\right)^\top.$$

Moreover, the product

$$\begin{pmatrix} \boldsymbol{I}_5 & \\ & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{x_{\boldsymbol{\pi}^{-1}(i),\ell+1}}{\zeta_i} & 0 & 1 \end{pmatrix} \end{pmatrix} R_i C_i \begin{pmatrix} \boldsymbol{I}_5 & \\ & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{x_{\boldsymbol{\pi}^{-1}(i),\ell+1}}{\zeta_i} & 0 & 1 \end{pmatrix} \end{pmatrix} = \mathbf{I}_8.$$

Hence, we conclude that $\widetilde{\mathbf{Hyb}}^{(1,\ell)}$ and $(*)$ are equivalent. Formally,

$$\Pr\left[1 \leftarrow \mathcal{A}\left(\widetilde{\mathbf{Hyb}}^{(1,\ell)}\right)\right] = \Pr\left[1 \leftarrow \mathcal{A}\left(*\right)\right].$$

Symmetrically to the first step, we can eliminate ζ_i and obtain that

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(*\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(1,\ell+1)}\right)\right]\right| \leq \frac{m}{q} + (m-1)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda).$$

Therefore, equation (5) in Lemma 3 holds, demonstrating that $\mathbf{Hyb}^{(1,Q)}$ and $\mathbf{Hyb}^{(1,0)}$ are computationally indistinguishable.

F Proof of Lemma 4

To prove Lemma 4, it suffices to show that

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell+1)}\right)\right]\right| \leq 2\left[\frac{m+1}{q} + (Qm+m-Q)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda)\right]. \quad (6)$$

To this end, we consider the following hybrid experiments $\mathbf{Hyb}^{(2,\ell,1)}$, $\mathbf{Hyb}^{(2,\ell,2)}$, and $\mathbf{Hyb}^{(2,\ell,3)}$ to establish a switch between $\mathbf{Hyb}^{(2,\ell)}$ and $\mathbf{Hyb}^{(2,\ell+1)}$.

F.0.1 Hybrid Experiment $\mathbf{Hyb}^{(2,\ell,1)}$

The difference between $\mathbf{Hyb}^{(2,\ell,1)}$ and $\mathbf{Hyb}^{(2,\ell)}$ lies in the $(\ell+1)$-th row of token $\boldsymbol{tk}_i$. In $\mathbf{Hyb}^{(2,\ell,1)}$,

$$\boldsymbol{tk}_i^{\ell+1} := \llbracket \left(\rho_{\ell+1}, 0, 0, \beta_{i,\ell+1}, \gamma_{i,\ell+1}, \xi_i, \delta_{\ell+1,\boldsymbol{\pi}(i)}, 0\right) R_i \rrbracket_2,$$

where $\xi_i \xleftarrow{\$} \mathbb{Z}_q^\times$. Similar to the proof of Lemma 3, leveraging the elements $\beta_{i,\ell+1}$ and $\gamma_{i,\ell+1}$, we can construct a non-uniform p.p.t. adversary $\mathcal{B}$ against the XDLin problem by leveraging the adversary $\mathcal{A}$.

- After $\mathcal{B}$ receives

$$\llbracket \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} a_1\beta_{1,\ell+1} \\ \vdots \\ a_m\beta_{m,\ell+1} \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} b_1\gamma_{1,\ell+1} \\ \vdots \\ b_m\gamma_{m,\ell+1} \end{pmatrix} \rrbracket_{1,2}, \llbracket \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} \rrbracket_2,$$

  it needs to distinguish whether $Y_i = \beta_{i,\ell+1} + \gamma_{i,\ell+1}$ or $Y_i = \beta_{i,\ell+1} + \gamma_{i,\ell+1} + \xi_i$.

- $\mathcal{B}$ calls $\texttt{Setup}(1^\lambda, n)$ to obtain $\boldsymbol{tk}$.

- For each $i \in [n]$, $\mathcal{C}$ samples an invertible matrix $W_i \xleftarrow{\$} (\mathbb{Z}_q)^{\times}_{8\times 8}$. Let

$$C_i = W_i \begin{pmatrix} 1 & & & & & & & \\ & b_i & & & & & & \\ & & b_i & & & & & \\ & & & b_i & & -b_i & & \\ & & & & a_i & -a_i & & \\ & & & & & a_ib_i & & \\ & & & & & & b_i & \\ & & & & & & & b_i \end{pmatrix},$$

and

$$R_i = \mathrm{diag}\left(1, \frac{1}{a_ib_i}, \cdots, \frac{1}{a_ib_i}\right) \begin{pmatrix} 1 & & & & & & & \\ & a_i & & & & & & \\ & & a_i & & & & & \\ & & & a_i & & 1 & & \\ & & & & b_i & 1 & & \\ & & & & & 1 & & \\ & & & & & & a_i & \\ & & & & & & & a_i \end{pmatrix} W_i^{-1}.$$

We have $R_iC_i = \mathbf{I}_8$.
When $j \neq \ell + 1$, the j-th token corresponding to sender i is computed by

$$\begin{aligned} \boldsymbol{tk}_i^j &:= [\![\big(\rho_j, 0, 0, a_ib_i\beta_{i,j}, a_ib_i\gamma_{i,j}, 0, a_ib_i\delta_{j,\boldsymbol{\pi}(i)}, 0\big)\, R_i]\!]_2 \\ &= [\![\big(\rho_j, 0, 0, \beta_{i,j}, \gamma_{i,j}, 0, \delta_{j,\boldsymbol{\pi}(i)}, 0\big) \begin{pmatrix} 1 & & & & & & & \\ & a_i & & & & & & \\ & & a_i & & & & & \\ & & & a_i & & 1 & & \\ & & & & b_i & 1 & & \\ & & & & & 1 & & \\ & & & & & & a_i & \\ & & & & & & & a_i \end{pmatrix} W_i^{-1}]\!]_2 \end{aligned}$$

Here we note that $a_ib_i\beta_{i,j},\ a_ib_i\gamma_{i,j}$ remain random samples and $\theta_i = a_ib_i$. When $j = \ell + 1$,

$$\boldsymbol{tk}_i^{\ell+1} := [\![\big(\rho_{\ell+1}, 0, 0, a_i\beta_{i,\ell+1}, b_i\gamma_{i,\ell+1}, Y_i, a_i\delta_{\ell+1,\boldsymbol{\pi}(i)}, 0\big) W_i^{-1}]\!]_2$$

- $\mathcal{B}$ computes the ciphertexts in the same way as in $\mathbf{Hyb}^{(2,\ell)}$, and returns them to $\mathcal{A}$.
- $\mathcal{B}$ outputs the same guess that $\mathcal{A}$ outputs.

Observe that if $Y_i = \beta_{i,\ell+1} + \gamma_{i,\ell+1}$, then $\mathcal{A}$'s view is identically distributed as in $\mathbf{Hyb}^{(2,\ell)}$; otherwise, if $Y_i = \beta_{i,\ell+1} + \gamma_{i,\ell+1} + \xi_i$, $\mathcal{A}$'s view is identically distributed as in $\mathbf{Hyb}^{(2,\ell,1)}$. Based on Proposition 1 and the XDLin assumption, we can obtain that

$$\left| \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell,1)}\right)\right] \right| \leq \frac{m}{q} + m\mathrm{Adv}^{\mathrm{XDLin}}(\lambda).$$

Next, we demonstrate that $\mathbf{Hyb}^{(2,\ell,1)}$ is equivalent to the following $\mathbf{Hyb}^{(2,\ell,2)}$.

F.0.2 Hybrid Experiment $\mathbf{Hyb}^{(2,\ell,2)}$

In $\mathbf{Hyb}^{(2,\ell,2)}$, the $(\ell+1)$-th row of token $\boldsymbol{tk}_i$ is replaced by

$$\boldsymbol{tk}_i^{\ell+1} := [\![(0, 0, 0, 0, 0, \xi_i, 0, 0)\, R_i]\!]_2.$$

When computing $\boldsymbol{ct}_{i,t}$, let

$$\boldsymbol{ct}_{i,t} := [\![C_i \big(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, \frac{1}{\xi_i}(\rho_{\ell+1}K_{i,t} + \theta_i\delta_{\ell+1,\boldsymbol{\pi}(i)}x_{i,t}), x_{i,t}, x_{\boldsymbol{\pi}^{-1}(i),t}\big)^\top]\!]_1.$$

It can be validated that the product of tokens and ciphertexts remains the same. Moreover, the product of

$$\begin{pmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ &&&& 1 &&& \\ \frac{-\rho_{\ell+1}}{\xi_i} & 0 & 0 & \frac{-\beta_{i,\ell+1}}{\xi_i} & \frac{-\gamma_{i,\ell+1}}{\xi_i} & 1 & \frac{-\theta_i\delta_{\ell+1,\boldsymbol{\pi}(i)}}{\xi_i} & 0 \\ &&&&&& 1 & \\ &&&&&&& 1 \end{pmatrix} R_i$$

and

$$C_i \begin{pmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ &&&& 1 &&& \\ \frac{\rho_{\ell+1}}{\xi_i} & 0 & 0 & \frac{\beta_{i,\ell+1}}{\xi_i} & \frac{\gamma_{i,\ell+1}}{\xi_i} & 1 & \frac{\theta_i\delta_{\ell+1,\boldsymbol{\pi}(i)}}{\xi_i} & 0 \\ &&&&&& 1 & \\ &&&&&&& 1 \end{pmatrix}$$

equals $\mathbf{I}_8$. Hence, we can obtain that $\mathbf{Hyb}^{(2,\ell,1)}$ is equivalent to $\mathbf{Hyb}^{(2,\ell,2)}$. Formally,

$$\Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell,1)}\right)\right] = \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell,2)}\right)\right].$$

Following this, we demonstrate that $\mathbf{Hyb}^{(2,\ell,2)}$ is computationally indistinguishable from the following $\mathbf{Hyb}^{(2,\ell,3)}$.

F.0.3 Hybrid Experiment $\mathbf{Hyb}^{(2,\ell,3)}$

$\mathbf{Hyb}^{(2,\ell,3)}$ differs from $\mathbf{Hyb}^{(2,\ell,2)}$ in the calculation of $\{\boldsymbol{ct}_{i,t}\}_{i\leq m}$, where $K_{i,t}$ in the third-to-last element is replaced by a randomly selected element $K'_{i,t} \xleftarrow{\$} \mathbb{Z}_q$ satisfying $\sum_{i=1}^m K'_{i,t} = \sum_{i=1}^m K_{i,t}$. In $\mathbf{Hyb}^{(2,\ell,3)}$, the challenger computes $\boldsymbol{ct}_{i,t}$ by

$$\boldsymbol{ct}_{i,t} := [\![C_i\big(K_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, \frac{1}{\xi_i}(\rho_{\ell+1}K'_{i,t} + \theta_i\delta_{\ell+1,\boldsymbol{\pi}(i)}x_{i,t}), x_{i,t}, x_{\boldsymbol{\pi}^{-1}(i),t}\big)^\top]\!]_1.$$

Following the methodology in Appendix E, the indistinguishability of $\mathbf{Hyb}^{(2,\ell,2)}$ and $\mathbf{Hyb}^{(2,\ell,3)}$ can be established through a reduction to the XDLin assumption. Here we demonstrate how to construct a non-uniform p.p.t. adversary $\mathcal{B}$ against game $(\diamond)$ by leveraging adversary $\mathcal{A}$.

- $\mathcal{B}$ interacts with $\mathcal{C}$ to obtain public parameters. Then, $\mathcal{C}$ generates and returns

  $$[\![\rho_{\ell+1}]\!]_{1,2}, [\![u]\!]_{1,2}, [\![\begin{pmatrix} \rho_{\ell+1}K_{1,1} & \cdots & \rho_{\ell+1}K_{1,Q} \\ \vdots & \ddots & \vdots \\ \rho_{\ell+1}K_{m,1} & \cdots & \rho_{\ell+1}K_{m,Q} \end{pmatrix}]\!]_{1,2},$$
  $$[\![\begin{pmatrix} \alpha_{1,1} & \cdots & \alpha_{1,Q} \\ \vdots & \ddots & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,Q} \end{pmatrix}]\!]_{1,2}, [\![\begin{pmatrix} Y_{1,1} & \cdots & Y_{1,Q} \\ \vdots & \ddots & \vdots \\ Y_{m,1} & \cdots & Y_{m,Q} \end{pmatrix}]\!]_1.$$

  Meanwhile, $\mathcal{C}$ sends $\{[\![\frac{\rho_j}{u}]\!]_2\}_{j\in[n]}$ to $\mathcal{B}$. Here we remark that, by our protocol, the generation of $\{\rho_j\}_{j\in[n]}$ cannot be controlled by the adversary and the values of $\{\rho_j\}_{j\in[n]}$ are secret.
- After $\mathcal{B}$ receives the above instances, it needs to distinguish whether $Y_{i,j} = K_{i,j} + \frac{\alpha_{i,j}}{u}$ or $Y_{i,j} = K'_{i,j} + \frac{\alpha_{i,j}}{u}$, where $\sum_{i=1}^m K_{i,t} = \sum_{i=1}^m K_i(t) = \sum_{i=1}^m K'_{i,t}$.
- $\mathcal{B}$ calls $\texttt{Setup}(1^\lambda, n)$. For $1 \leq i \leq m$, when $j \leq \ell$, let

  $$\boldsymbol{tk}_i^j := [\![\Big(\rho_j, -\frac{\rho_j}{u}, 0, \beta_{i,j}, \gamma_{i,j}, 0, 0, \theta_i\delta_{j,i}\Big)\, R_i]\!]_2.$$

When $j = \ell + 1$, let

$$\boldsymbol{tk}_i^{\ell+1} := [\![(0,0,0,0,0,\xi_i,0,0)\, R_i]\!]_2.$$

When $j \geq \ell + 2$, let

$$\boldsymbol{tk}_i^{j} := [\![\Big(\rho_j, -\frac{\rho_j}{u}, 0, \beta_{i,j}, \gamma_{i,j}, 0, \theta_i \delta_{j,\boldsymbol{\pi}(i)}, 0\Big)\, R_i]\!]_2.$$

- After receiving the queries $\{x_{i,t}\}_{i \leq m}$ from $\mathcal{A}$, $\mathcal{B}$ computes

$$\boldsymbol{ct}_{i,t} := [\![C_i\big(Y_{i,t}, \alpha_{i,t}, \alpha'_{i,t}, 0, 0, \frac{1}{\xi_i}(\rho_{\ell+1} K_{i,t} + \theta_i \delta_{\ell+1,\boldsymbol{\pi}(i)} x_{i,t}), x_{i,t}, x_{\boldsymbol{\pi}^{-1}(i),t}\big)^\top]\!]_1,$$

  and returns it to $\mathcal{A}$.

- $\mathcal{B}$ outputs the identical guess output by $\mathcal{A}$.

Observe that if $Y_{i,j} = K_{i,j} + \frac{\alpha_{i,j}}{u}$, then $\mathcal{A}$'s view is identically distributed as in $\mathbf{Hyb}^{(2,\ell,2)}$; otherwise, if $Y_{i,j} = K'_{i,j} + \frac{\alpha_{i,j}}{u}$, $\mathcal{A}$'s view is identically distributed as in $\mathbf{Hyb}^{(2,\ell,3)}$. By the XDLin assumption and Proposition 4, we can obtain that

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell,2)}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{Hyb}^{(2,\ell,3)}\right)\right]\right| \leq \frac{1}{q} + Q(m-1)\mathrm{Adv}^{\mathrm{XDLin}}(\lambda).$$

Symmetrically to the steps above, equation (6) can be proved. Hence, Lemma 4 holds.

References

[1] X. Chen, J. An, Z. Xiong, C. Xing, N. Zhao, F. R. Yu, and A. Nallanathan, “Covert communications: A comprehensive survey,” *IEEE Commun. Surv. & Tutor.*, vol. 25, no. 2, pp. 1173–1198, Apr. 2023.

[2] I. Makhdoom, M. Abolhasan, and J. Lipman, “A comprehensive survey of covert communication techniques, limitations and future challenges,” *Computers & Security*, vol. 120, p. 102784, Sep. 2022.

[3] Z. Chen, L. Zhu, P. Jiang, C. Zhang, F. Gao, J. He, D. Xu, and Y. Zhang, “Blockchain meets covert communication: A survey,” *IEEE Commun. Surv. & Tutor.*, vol. 24, no. 4, pp. 2163–2192, Sep. 2022.

[4] D. Hitaj, G. Pagnotta, B. Hitaj, F. Perez-Cruz, and L. V. Mancini, “Fedcomm: Federated learning as a medium for covert communication,” *IEEE Trans. Depend. Sec. Comput.*, Jun. 2023.

[5] J. Partala, “Provably secure covert communication on blockchain,” *Cryptography*, vol. 2, no. 3, p. 18, Aug. 2018.

[6] Y. Gilad, “Metadata-private communication for the 99%,” *Commun. ACM*, vol. 62, no. 9, pp. 86–93, Aug. 2019.

[7] S. Sasy and I. Goldberg, “Sok: Metadata-protecting communication systems,” *PETS 2024*, 2024.

[8] V.-T. Hoang, Y. A. Ergu, V.-L. Nguyen, and R.-G. Chang, “Security risks and countermeasures of adversarial attacks on AI-driven applications in 6G networks: A survey,” *J. Netw. Comput. Appl.*, p. 104031, Dec. 2024.

[9] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, “AI and 6G security: Opportunities and challenges,” in *EuCNC/6G Summit*, Porto, Portugal, Jul. 2021, pp. 616–621.

[10] V.-L. Nguyen, P.-C. Lin, B.-C. Cheng, R.-H. Hwang, and Y.-D. Lin, “Security and privacy for 6G: A survey on prospective technologies and challenges,” *IEEE Commun. Surv. & Tutor.*, vol. 23, no. 4, pp. 2384–2428, Aug. 2021.

[11] Y. Zhang, W. Liu, and W. Lou, “Anonymous communications in mobile ad hoc networks,” in *INFOCOM’05*, vol. 3, Miami, FL, USA, Mar. 2005, pp. 1940–1951.

[12] M. Edman and B. Yener, “On anonymity in an electronic society: A survey of anonymous communication systems,” *ACM Comput. Surv.*, vol. 42, no. 1, pp. 1–35, Dec. 2009.

[13] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, and C. Diaz, “A survey on routing in anonymous communication protocols,” *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–39, Jun. 2018.

[14] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, “Hiding routing information,” in *Information Hiding*, vol. 1174, Berlin, Heidelberg, Jun. 1996, pp. 137–150.

[15] R. Dingledine, N. Mathewson, and e. a. Syverson, Paul F, “Tor: The second-generation onion router.” in *USENIX security symposium*, vol. 4, San Diego, CA, USA, Aug. 2004, pp. 303–320.

[16] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, “Users get routed: Traffic correlation on Tor by realistic adversaries,” in *CCS ’13*, Taipei, Taiwan, China, Nov. 2013, pp. 337–348.

[17] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, “A new cell-counting-based attack against Tor,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1245–1261, Aug. 2012.

[18] S. J. Murdoch and P. Zieliński, “Sampled traffic analysis by internet-exchange-level adversaries,” in *PETS 2007*, Ottawa, Canada, Jun. 2007, pp. 167–183.

[19] A. Serjantov and P. Sewell, “Passive attack analysis for connection-based anonymity systems,” in *ESORICS 2003*, Gjøvik, Norway, Oct. 2003, pp. 116–131.

[20] B. Zantout, R. Haraty *et al.*, “I2P data communication system,” in *ICN ’11*, St. Maarten, The Netherlands Antilles, Jan. 2011, pp. 401–409.

[21] C. Egger, J. Schlumberger, C. Kruegel, and G. Vigna, “Practical attacks against the I2P network,” in *RAID 2013*, Rodney Bay, St. Lucia, Oct. 2013, pp. 432–451.

[22] M. Herrmann and C. Grothoff, “Privacy-implications of performance-based peer selection by onion-routers: a real-world case study using I2P,” in *PETS 2011*, Waterloo, ON, Canada, Jul. 2011, pp. 155–174.

[23] N. P. Hoang, P. Kintis, M. Antonakakis, and M. Polychronakis, “An empirical study of the I2P anonymity network and its censorship resistance,” in *IMC ’18*, Boston, MA, USA, Oct. 2018, pp. 379–392.

[24] S. H. Jeong, A. R. Kang, J. Kim, H. K. Kim, and A. Mohaisen, “A longitudinal analysis of I2P leakage in the public DNS infrastructure,” in *SIGCOMM ’16*, Florianopolis, Brazil, Aug. 2016, pp. 557–558.

[25] G. Danezis and I. Goldberg, “Sphinx: A compact and provably secure mix format,” in *IEEE S&P 2009*, Oakland, CA, USA, Aug. 2009, pp. 269–282.

[26] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis, “The loopix anonymity system,” in *USENIX security symposium*, Vancouver, BC, Canada, Aug. 2017, pp. 1199–1216.

[27] C. Diaz, S. J. Murdoch, and C. Troncoso, “Impact of network topology on anonymity and overhead in low-latency anonymity networks,” in *PETS 2010*, Berlin, Germany, Jul. 2010, pp. 184–201.

[28] D. Kesdogan, J. Egner, and R. Büschkes, “Stop-and-go-mixes providing probabilistic anonymity in an open system,” in *Information Hiding*, Portland, Oregon, USA, Apr. 1998, pp. 83–98.

[29] D. Chaum, “Blind signatures for untraceable payments,” in *CRYPTO ’82*, 1983, pp. 199–203.

[30] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[31] E. Shi and K. Wu, “Non-interactive anonymous router,” in *EUROCRYPT 2021*, Zagreb, Croatia, Oct. 2021, pp. 489–520.

[32] M. Chauhan, A. K. Singh, and Komal, “Survey of onion routing approaches: advantages, limitations and future scopes,” in *ICCBI ’19*, Madurai, India, Dec. 2019, pp. 686–697.

[33] J. Feigenbaum, A. Johnson, and P. Syverson, “Probabilistic analysis of onion routing in a black-box model,” *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 3, pp. 1–28, Nov. 2012.

[34] J. Groth and A. Sahai, “Efficient non-interactive proof systems for bilinear groups,” in *EUROCRYPT 2008*, Istanbul, Turkey, Apr. 2008, pp. 415–432.

[35] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl, “Correlated pseudorandom functions from variable-density LPN,” in *FOCS ’20*, Durham, NC, USA, Nov. 2020, pp. 1069–1080.

[36] K. Bonawitz and e. a. Ivanov, Vladimir, “Practical secure aggregation for privacy-preserving machine learning,” in *CCS ’17*, Dallas, Texas, USA, Nov. 2017, pp. 1175–1191.

[37] M. Abdalla, F. Benhamouda, and R. Gay, “From single-input to multi-client inner-product functional encryption,” in *ASIACRYPT 2019*, Kobe, Japan, Dec. 2019, pp. 552–582.

[38] O. Goldreich, S. Goldwasser, and S. Micali, “How to construct random functions,” *J. ACM)*, vol. 33, no. 4, pp. 792–807, 1986.

[39] S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters, “Building efficient fully collusion-resilient traitor tracing and revocation schemes,” in *CCS ’10*, Chicago, Illinois, USA, Oct. 2010, pp. 121–130.

[40] J. Tomida, M. Abe, and T. Okamoto, “Efficient functional encryption for inner-product values with full-hiding security,” in *ISC ’16*, Honolulu, HI, USA, Sep. 2016, pp. 408–425.

[41] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Jun. 1976.

[42] T. Icart, “How to hash into elliptic curves,” in *CRYPTO 2009*. Springer, 2009, pp. 303–316.

[43] J. Brendel, C. Cremers, D. Jackson, and M. Zhao, “The provable security of ed25519: theory and practice,” in *IEEE S&P 2021*, San Francisco, CA, USA, Aug. 2021, pp. 1659–1676.

[44] P. Rogaway and T. Shrimpton, “A provable-security treatment of the key-wrap problem,” in *EUROCRYPT 2006*, St. Petersburg, Russia, May 2006, pp. 373–390.

[45] J. Stern, “Why provable security matters?” in *EUROCRYPT 2003*, Warsaw, Poland, May 2003, pp. 449–461.