Practical Fully Simulatable Oblivious Transfer with Sublinear Communication

Bingsheng Zhang[1], Helger Lipmaa[2], Cong Wang[3], and Kui Ren[1]

[1] State University of New York at Buffalo, United States
[2] University of Tartu, Estonia
[3] City University of Hong Kong, China

Abstract During an adaptive k-out-of-N oblivious transfer (OT), a sender has N private documents, and a receiver wants to adaptively fetch k documents from them such that the sender learns nothing about the receiver's selection and the receiver learns nothing more than those chosen documents. Many fully simulatable and universally composable adaptive OT schemes have been proposed, but those schemes typically require $\mathcal{O}(N)$ communication in the initialization phase, which yields $\mathcal{O}(N)$ overall communication. On the other hand, in some applications, the receiver just needs to fetch a small number of documents, so the initialization cost dominates in the entire protocol, especially for 1-out-of-N OT. We propose the first fully simulatable adaptive OT with sublinear communication under the DDH assumption in the plain model. Our scheme has $\mathcal{O}(N^{1/2})$ communication in both the initialization phase and each transfer phase. It achieves better (amortized) overall communication complexity compared to existing schemes when $k = \mathcal{O}(N^{1/2})$.
Keywords. Adaptive oblivious transfer, fully simulatable security, sublinear communication, zero knowledge batch argument.

1 Introduction

Data outsourcing and online shopping have become during the recent years. To address the related information security and privacy concerns, many cryptographic protocols have been studied to accomplish tasks with minimal information disclosure. Consider the case that an online store sells digital goods, such as movies, books, music, etc. The buyer wants to purchase some of them without revealing his/her choices. Here, we assume that there is a uniform price for those goods in the same category, e.g. movies. [1] Oblivious transfer (OT) is a handy primitive, which has found its usage in many security applications with this kind of privacy requirements as in the aforementioned case.

OT family mainly consists of 1-out-of-2 OT, denoted as OT_1^2, 1-out-of-N OT, denoted as OT_1^N, k-out-of-N OT, denoted as OT_k^N and k-out-of-N OT with adaptive queries (also known as adaptive k-out-of-N OT), denoted as $\mathsf{OT}_{k\times 1}^N$.

[1] It is also possible to protect the buyer's privacy even if all the goods are paid at unique prices. This problem is addressed by priced OT, see [33] for details.

OT_1^2 is widely used in secure multi-party/two-party computation, for example, it serves as an important building block of Yao's garbled circuit [34] in two-party *secure function evaluation* (SFE). Based on earlier work of Lipmaa [26], Ishai and Paskin [18] showed how to privately evaluate a branching program with OT_1^2. OT_1^N also has rich applications in financial cryptography, for instance, one can use OT_1^N to construct simultaneous contract signing schemes [29].

We focus on $\mathsf{OT}_{k\times 1}^N$ as well as its special case, OT_1^N when $k = 1$. During an $\mathsf{OT}_{k\times 1}^N$ protocol, a sender has N private documents, and a receiver can adaptively fetch k documents from them such that the sender learns nothing about the receiver's selection and the receiver learns nothing more than those k documents. The notion of $\mathsf{OT}_{k\times 1}^N$ was first introduced by Naor and Pinkas [30] who also gave several applications, including oblivious search. In an oblivious search protocol, the server owns a sorted database that the client wants to search. Given the element that the client is searching for, they invoke an $\mathsf{OT}_{k\times 1}^N$ protocol using binary search, where $k = \log N$. After the protocol execution, the client can determine whether the element is in the database while the server only has revealed limited database information ($\log N$ elements).

There is always a trade-off between security and efficiency. Due to bandwidth limitation, most applications employ OT schemes with so-called *half-simulation* security, where the sender's and receiver's security are handled separately. Such OT_1^N can achieve logarithmic communication [10] or sublinear computation [27] and OT_k^N can achieve optimal rate [1]. The receiver's security is defined as indistinguishability of the sender's view of the protocol when the receiver's choices are different. The sender's security follows the real-world/ideal-world paradigm and guarantees that for any malicious receiver in the real world there is a receiver in an ideal world where OT is implemented by a trusted party. However, this security definition is vulnerable to the selective failure attack [30]. Namely, the sender is able to cause protocol failure based on some property of the receiver's selection. Based on an arbitrary semisimulatable OT protocol, Laur and Lipmaa proposed a consistent OT protocol [24] of virtually the same complexity that allows to detect selective failures, but still does not obtain ideal security.

On the other hand, all existing *fully simulatable* and *universally composable* adaptive OT schemes typically require $\mathcal{O}(N)$ communication in the initialization phase. The huge initialization cost is not acceptable in many applications, especially when the receiver is only required to fetch a small number of documents. For example, in 1-out-of-N OT the initialization cost dominates the entire protocol, so the overall communication cost becomes $\mathcal{O}(N)$. Can we make OT more communication-efficient without sacrificing its security level? In this paper, we try to answer this question by investigating a practical fully simulatable $\mathsf{OT}_{k\times 1}^N$ scheme with sublinear communication.

Our Contribution and Related Work. In theory, one can transform any secure OT protocol in semi-honest model to an OT protocol that is secure against malicious adversaries by plug-in *zero-knowledge* (ZK) proofs/arguments. To achieve sublinear communication, we may use *probabilistically checkable proofs*

Scheme	Init Cost	Transfer Cost	Assumption	Security
Prot. 3.1 [31]	$\mathcal{O}(N)$	$\mathcal{O}(N^{1/2})$	DDH	Full Sim
[5]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	q-Power DDH + q-Strong DH	Full Sim
[13]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DLIN + q-Hidden LRSW	UC
[20]	$\mathcal{O}(N)$	$\mathcal{O}(N)$	Dec. n-th Residuosity/DDH	Full Sim
[19]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	Dec. Residuosity + q-DDHI	Full Sim
[33]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DLIN + q-Hidden SDH + q-TDH	UC
[21]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DDH	Full Sim
[14]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	3-DDH + DLIN	Full Sim
[22]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DDH/DLIN/DCR/QR/LWE	Full Sim
[35]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DDH/Dec. n-th Residuosity	Full Sim
this work	$\mathcal{O}(N^{1/2})$	$\mathcal{O}(N^{1/2})$	DDH	Full Sim

Table1. Comparison of $\mathsf{OT}^N_{k\times 1}$ schemes. The trivial factor $\log N$ is ignored.

(PCP), e.g., [4] or a sublinear ZK argument [17,28]. The problem with such approaches is that the OT protocol has to be reduced to some NP-complete language, which is neither efficient nor practical. In this paper, we propose the first fully simulatable $\mathsf{OT}^N_{k\times 1}$ scheme with $\mathcal{O}(\sqrt{N})$ communication in both the initialization phase and each transfer phase based on the standard DDH assumption. When $k = \mathcal{O}(\sqrt{N})$, our $\mathsf{OT}^N_{k\times 1}$ scheme has better amortized overall communication complexity compared to existing schemes. In order to achieve sublinear communication complexity, we constructed a few efficient batch ZK arguments, such as masked multi-exponentiation batch argument (c.f. Sect. 4.3, below). We use Lim's multi-exponentiation algorithm in our implementation, and a benchmark is given at the end of this paper.

We now give a survey on recent *fully simulatable* and *universally composable* $\mathsf{OT}^N_{k\times 1}$ schemes. As shown by Canetti and Fischlin [6], an OT cannot be realized in UC security without additional trusted setup assumptions. All the UC-secure $\mathsf{OT}^N_{k\times 1}$ schemes mentioned here are in *common reference string* (CRS) model, i.e. $\mathcal{F}_{crs}$-hybrid model. Whereas many fully simulatable $\mathsf{OT}^N_{k\times 1}$ schemes in this survey, as well as our construction, are realized in the plain model. Table 1 lists several existing $\mathsf{OT}^N_{k\times 1}$ schemes, together with our proposed scheme for comparison. In 2007, Camenisch, Neven and shelat [5] proposed $\mathsf{OT}^N_{k\times 1}$ under the q-strong Diffie-Hellman and q-power decisional Diffie-Hellman assumptions in bilinear groups. They used signatures as a key ingredient in their scheme. Later, Green and Hohenberger [12] showed an $\mathsf{OT}^N_{k\times 1}$ in random oracle model under decisional bilinear Diffie-Hellman assumption. In their scheme, the sender encrypts message m_i by identity-based encryption under identity i. The receiver executes a blind key extraction protocol such that he/her can obliviously obtain the secret key of any identity. In 2008, Green and Hohenberger [13] introduced another OT that achieves UC security in the $\mathcal{F}_{crs}$-hybrid model, using a Groth-Sahai non-interactive ZK (NIZK) proof for pairing product equations. The scheme is based on the decisional linear and q-Hidden LRSW assumptions. Jarecki and

Liu [19] simplified the Camenisch et al. construction to a fully simulatable OT under the composite decisional residuosity and q-decisional Diffie-Hellman Inversion assumptions. Rial, Kohlweiss and Preneel [33] presented an adaptive priced OT that achieves UC security using "assisted decryption". In 2009, Kurosawa and Nojima [20] gave adaptive OT constructions based on Paillier and ElGamal encryption schemes. Later, Kurosawa, Nojima and Phong [21] improved the scheme [20] by increasing the complexity of initialization phase. In 2011, Green and Hohenberger [14] proposed another fully simulatable OT under decisional 3-party Diffie-Hellman assumption. Recently, Kurosawa et al. [22] and Zhang [35] generalized the scheme in [21] to various schemes with different assumptions.

We emphasize that Prot. 3.1 in [31] is essentially different from our scheme. In [31], the sender first 'commits' the documents to the receiver in the initialization phase. This step takes $\mathcal{O}(N)$ communication, because each "commitment" serves as an encryption and the receiver should be able to extract (or decrypt) the committed document from it later. Therefore, it is not possible to directly plug a succinct commitment scheme in the Naor-Pinkas scheme.

2 Preliminaries

Let $[n] := \{1, \ldots, n\}$. By $\boldsymbol{a}$, we denote a vector $\boldsymbol{a} = (a_1, \ldots, a_n)^T$. When S is a set, $a \leftarrow_\$ S$ means that a is uniformly and randomly chosen from S. Let λ be the security parameter. By $A \overset{c}{\approx} B$, we mean that A and B are computationally indistinguishable. We abbreviate *probabilistic polynomial-time* as p.p.t. and let $\mathrm{poly}(\cdot)$ be a polynomially-bounded function.

Elliptic Curves Over $\mathbb{F}_p$. The implementation of our scheme is based on elliptic curve groups for efficiency. Let $\sigma := (p, a, b, g, q, \zeta)$ be the elliptic curve domain parameters over $\mathbb{F}_p$, consisting of a prime p specifying the finite field $\mathbb{F}_p$, two elements $a, b \in \mathbb{F}_p$ specifying an elliptic curve $E(\mathbb{F}_p)$ defined by $E : y^2 \equiv x^3 + ax + b \pmod{p}$, a base point $g = (x_g, y_g)$ on $E(\mathbb{F}_p)$, a prime q which is the order of g, and an integer ζ which is the cofactor $\zeta = \#E(\mathbb{F}_p)/q$. We denote the cyclic group generated by g by $\mathbb{G}$, and it is assumed that the DDH assumption holds over $\mathbb{G}$, that is for all p.p.t. adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\mathcal{A}) = \left| \Pr \left[\begin{array}{l} x, y \leftarrow_\$ \mathbb{Z}_q; b \leftarrow_\$ \{0,1\}; h_0 = g^{xy}; \\ h_1 \leftarrow_\$ \mathbb{G} : \mathcal{A}(g, g^x, g^y, h_b) = b \end{array} \right] - \frac{1}{2} \right| \leq \epsilon(\lambda) \ ,$$

where $\epsilon(\cdot)$ is a negligible function.

Security Definition (Fully Simulation Security). We use the same security definition as in [30,5,14]. Let $(\mathscr{S}_I, \mathscr{R}_I, \mathscr{S}_T, \mathscr{R}_T)$ be an $\mathsf{OT}^N_{k\times 1}$ protocol. Let S_*, R_* be private states. During the initialization phase, the sender sets $S_0 \leftarrow \mathscr{S}_I(m_1, \ldots, m_N)$, and the receiver sets $R_0 \leftarrow \mathscr{R}_I()$. During the ℓ-th transfer phase, $\ell \in [k]$, the sender sets $S_\ell \leftarrow \mathscr{S}_T(S_{\ell-1})$, and the receiver sets $(R_\ell, m^*_{\sigma_\ell}) \leftarrow \mathscr{R}_T(R_{\ell-1}, i_\ell)$, where $i_\ell \in [N]$ is the index of the message to be

received. $m^*_{\sigma_\ell} = m_{\sigma_\ell}$ if retrieval succeeds, $m^*_{\sigma_\ell} = \perp$ if fails. The security of an $\mathsf{OT}^N_{k\times 1}$ scheme is defined in the real-world/ideal-world paradigm with static corruption, i.e. the adversary $\mathcal{A}$ can only choose to corrupt either the sender or the receiver at the beginning of the experiment.

Real experiment. In experiment $\mathbf{Real}_{\hat{\mathbf{S}},\hat{\mathbf{R}}}(N, k, m_1, \ldots, m_N, \mathcal{I})$, a presumably cheating sender $\hat{\mathbf{S}}$ is given messages $(m_1, \ldots, m_N)$ as input and interacts with a presumably cheating receiver $\hat{\mathbf{R}}(\mathcal{I})$, where $\mathcal{I}$ is a selection algorithm that on input messages $\{m_{i_t}\}_{t=1}^{\ell-1}$ outputs the index i_ℓ of the next message to be queried. In the initialization phase, $\hat{\mathbf{S}}$ and $\hat{\mathbf{R}}$ output the initial states S_0 and R_0. In the ℓ-th transfer phase, for $\ell \in [k]$, the sender runs $S_\ell \leftarrow \hat{\mathbf{S}}(S_{\ell-1})$, and the receiver runs $(R_\ell, m^*_{i_\ell}) \leftarrow \hat{\mathbf{R}}(R_{\ell-1})$. After the k-th transfer, the output of the experiment $\mathbf{Real}_{\hat{\mathbf{S}},\hat{\mathbf{R}}}$ is the tuple (S_k, R_k).

We define the honest sender algorithm $\mathbf{S}$ as the one that runs $\mathscr{S}_I(m_1, \ldots, m_N)$ in the initialization phase, runs $\mathscr{S}_T()$ during each transfer phase, and returns $S_k = \emptyset$ as its final output. The honest receiver algorithm $\mathbf{R}$ runs $\mathscr{R}_I()$ in the initialization phase, runs $\mathscr{R}_T(R_{\ell-1}, i_\ell)$ during the ℓ-th transfer phase, where the index i_ℓ is generated by $\mathcal{I}$, and returns $R_k = (m_{i_1}, \ldots, m_{i_k})$ as its final output.

Ideal experiment. In experiment $\mathbf{Ideal}_{\hat{\mathbf{S}}',\hat{\mathbf{R}}'}(N, k, m_1, \ldots, m_N, \mathcal{I})$, the presumably cheating sender $\hat{\mathbf{S}}'$ and the presumably cheating receiver $\hat{\mathbf{R}}'$ communicate with the ideal functionality $\mathcal{F}_{OT}^{N\times 1}$. In the initialization phase, $\hat{\mathbf{S}}'(m_1, \ldots, m_N)$ sends messages $m_1^*, \ldots, m_N^*$ to $\mathcal{F}_{OT}^{N\times 1}$. In the ℓ-th transfer phase, $\ell \in [k]$, $\hat{\mathbf{R}}'(\mathcal{I})$ sends to $\mathcal{F}_{OT}^{N\times 1}$ an index i_ℓ^*. $\mathcal{F}_{OT}^{N\times 1}$ then sends a tag '$\mathsf{Received}$' to $\hat{\mathbf{S}}'$, and $\hat{\mathbf{S}}'$ replies a bit $b_\ell \in \{0, 1\}$ to $\mathcal{F}_{OT}^{N\times 1}$. If $b_\ell = 1$ and $i_\ell^* \in [N]$, $\mathcal{F}_{OT}^{N\times 1}$ sends $m^*_{i_\ell^*}$ to $\hat{\mathbf{R}}'$; otherwise, it sends $\perp$ to $\hat{\mathbf{R}}'$. After the k-th transfer, the output of the experiment $\mathbf{Ideal}_{\hat{\mathbf{S}}',\hat{\mathbf{R}}'}$ is the tuple (S_k, R_k).

We define the honest sender algorithm $\mathbf{S}'(m_1, \ldots, m_N)$ as the one that sends $m_1, \ldots, m_N$ to $\mathcal{F}_{OT}^{N\times 1}$ in the initialization phase, and sends $b_\ell = 1$ during each transfer phase, and returns $S_k = \emptyset$ as its final output. The honest receiver $\mathbf{R}'$ submits the indices i_ℓ that generated by $\mathcal{I}$ to $\mathcal{F}_{OT}^{N\times 1}$, and returns $R_k = (m_{i_1}, \ldots, m_{i_k})$ as its final output.

Sender Security. An $\mathsf{OT}^N_{k\times 1}$ is sender-secure if for every real-world p.p.t. receiver $\hat{\mathbf{R}}$, there exists an ideal-world p.p.t. receiver $\hat{\mathbf{R}}'$, s.t. for every $N = \mathrm{poly}(\lambda)$, $k \in [N]$, $(m_1, \ldots, m_N)$, selection algorithm I, and p.p.t. distinguisher $\mathbf{D}$,

$$\mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}(N, k, m_1, \ldots, m_N, \mathcal{I}) \quad \overset{c}{\approx} \quad \mathbf{Ideal}_{\mathbf{S}',\hat{\mathbf{R}}'}(N, k, m_1, \ldots, m_N, \mathcal{I}) \ .$$

Receiver Security. An $\mathsf{OT}^N_{k\times 1}$ is receiver-secure if for every real-world p.p.t. sender $\hat{\mathbf{S}}$, there exists an ideal-world p.p.t. sender $\hat{\mathbf{S}}'$, s.t. for every $N = \mathrm{poly}(\lambda)$,

$k \in [N]$, $(m_1, \ldots, m_N)$, selection algorithm $\mathcal{I}$, and p.p.t. distinguisher $\mathbf{D}$,

$$\mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}(N, k, m_1, \ldots, m_N, \mathcal{I}) \quad \overset{c}{\approx} \quad \mathbf{Ideal}_{\hat{\mathbf{S}}',\mathbf{R}'}(N, k, m_1, \ldots, m_N, \mathcal{I}) \ .$$

Definition 1. $\mathsf{OT}^N_{k\times 1}$ *is fully simulatable iff it is both sender- and receiver-secure.*

Special Honest Verifier Zero-knowledge Argument. Let $\mathcal{R}$ be a polynomial time decidable binary relation, we say w is a witness for a statement x if $(x, w) \in \mathcal{R}$. We define the language $\mathcal{L} := \{x \mid \exists w : (x, w) \in \mathcal{R}\}$ as the set of all statements x that have a witness w for the relation $\mathcal{R}$. Let a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ be two p.p.t. interactive algorithms. Denote $\tau \leftarrow \langle \mathcal{P}(x, w), \mathcal{V}(x)\rangle$ as the public transcript produced by $\mathcal{P}$ and $\mathcal{V}$. After the protocol, $\mathcal{V}$ accepts iff $\Phi(x, \tau) = 1$, where Φ is a predicate function.

Definition 2. *We say $(\mathcal{P}, \mathcal{V})$ is a perfectly complete argument for a relation $\mathcal{R}$ if for all non-uniform p.p.t. interactive adversaries $\mathcal{A}$ it satisfies*

- *Perfect completeness:* $\Pr[(x, w) \leftarrow \mathcal{A}; \tau \leftarrow \langle \mathcal{P}(x, w), \mathcal{V}(x)\rangle : (x, w) \in \mathcal{R} \vee \Phi(x, \tau) = 1] = 1$*;*
- *Computational soundness:* $\Pr[x \leftarrow \mathcal{A}; \tau \leftarrow \langle \mathcal{A}, \mathcal{V}(x)\rangle : x \notin \mathcal{L} \wedge \Phi(x, \tau) = 1] \approx 0$.

Denote $\mathcal{V}(x; r)$ as the verifier $\mathcal{V}$ on input x, given r as the randomness. An argument $(\mathcal{P}, \mathcal{V})$ is *public coin* if the verifier $\mathcal{V}$ picks his challenges randomly and independently of the messages sent by the prover $\mathcal{P}$.

Definition 3. *A public coin argument $(\mathcal{P}, \mathcal{V})$ is called a perfect special honest verifier zero-knowledge (SHVZK) argument for a relation $\mathcal{R}$ if there exists a p.p.t. simulator $\mathcal{S}$ such that for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$\begin{aligned}\Pr[(x, w, r) \leftarrow \mathcal{A}; &\tau \leftarrow \langle \mathcal{P}(x, w), \mathcal{V}(x; r)\rangle : (x, w) \in \mathcal{R} \wedge \mathcal{A}(\tau) = 1] \\ &= \Pr[(x, w, r) \leftarrow \mathcal{A}; \tau \leftarrow \mathcal{S}(x; r) : (x, w) \in \mathcal{R} \wedge \mathcal{A}(\tau) = 1] \ .\end{aligned}$$

We define the SHVZK *argument of knowledge* similiarly to the definition of [15,16,2]; namely, given an adversary that produces an acceptable argument with probability p, there exists a witness-extended emulator that produces a similar argument with probability p and outputs a witness. The standard definition of "*proofs of knowledge* (PoK)" by Bellare and Goldreich [3] does not work for "*arguments of knowledge* (AoK)". See [8] for more discussion of this issue and an alternative definition of knowledge soundness.

Definition 4. *A public coin argument $(\mathcal{P}, \mathcal{V})$ has a witness extended emulator if for all p.p.t. $\mathcal{P}^*$ there exists an expected polynomial time emulator $\mathcal{X} = \mathcal{X}^{\mathcal{P}^*}$ such that for all non-uniform polynomial time adversaries $\mathcal{A}$,*

$$\Pr\begin{bmatrix}(x, \psi) \leftarrow \mathcal{A}; \\ \tau \leftarrow \langle \mathcal{P}^*(x, \psi), \mathcal{V}(x)\rangle : \\ \mathcal{A}(\tau) = 1\end{bmatrix} = \Pr\begin{bmatrix}(x, \psi) \leftarrow \mathcal{A}; \\ (\tau, w) \leftarrow \mathcal{X}^{\langle \mathcal{P}^*(x,\psi), \mathcal{V}(x)\rangle}(x, \psi) : \\ \mathcal{A}(\tau) = 1 \wedge (\Phi(x, \tau) = 0 \vee (x, w) \in \mathcal{R})\end{bmatrix} \ .$$

Here, $\mathcal{X}$ has access to a transcript oracle $\langle \mathcal{P}^*(x,\psi), \mathcal{V}(x)\rangle$ that can be rewound to a particular round and run again with the verifier using fresh randomness. Let ψ be the state of $\mathcal{P}^*$, including the randomness. Whenever $\mathcal{P}^*$ is able to make a convincing argument with state ψ, the emulator $\mathcal{X}$ can extract a witness w.

3 Building Blocks

Additively Homomorphic Public-key Cryptosystem. The lifted ElGamal public-key cryptosystem consists of the following 4 p.p.t. algorithms:

- $\mathsf{Gen}_{\mathsf{gk}}(1^\lambda)$: inputs a security parameter λ, and outputs $\sigma := (p, a, b, g, q, \zeta)$.
- $\mathsf{Gen}_{\mathsf{pkc}}(\sigma)$: picks $sk \leftarrow_{\$} \mathbb{Z}_q^*$, sets $pk := h = g^{sk}$, and outputs (pk, sk).
- $\mathsf{Enc}_{pk}(m; r)$: outputs $e := (e_1, e_2) = (g^r, g^m h^r)$.
- $\mathsf{Dec}_{sk}(e)$: outputs $\mathsf{DL}_g(e_2 \cdot e_1^{-sk})$, where $\mathsf{DL}_g(x)$ is the discrete logarithm of x. (Note that since $\mathsf{DL}_g(\cdot)$ is not efficient, the message space should be a small set, say $\{0,1\}^\xi$, for $\xi \leq 30$.)

It is well known that lifted ElGamal encryption scheme is IND-CPA secure under the DDH assumption. It is additively homomorphic: $\mathsf{Enc}_{pk}(m_1; r_1) \cdot \mathsf{Enc}_{pk}(m_2; r_2) = \mathsf{Enc}_{pk}(m_1 + m_2; r_1 + r_2)$.

Additively Homomorphic Succinct Vector Commitment. In our protocol, we use a generalized version of the Pedersen commitment scheme [32]. The generalized Pedersen commitment scheme consists of the following 4 algorithms:

- $\mathsf{Gen}_{\mathsf{gk}}(1^\lambda)$: inputs security parameter λ, and outputs $\sigma := (p, a, b, g, q, \zeta)$.
- $\mathsf{Gen}_{\mathsf{ped}}(\sigma)$: outputs distinct generators $ck := (g_1, \ldots, g_n, f)$.
- $\mathsf{Com}_{ck}(\boldsymbol{m}; r)$: outputs a commitment $c := f^r \prod_{i=1}^n g_i^{m_i}$ for $\boldsymbol{m} \in \mathbb{Z}_q^n$ and $r \in \mathbb{Z}_q$.
- $\mathsf{Open}_{ck}(c)$: outputs $\boldsymbol{m} \in \mathbb{Z}_q^n, r \in \mathbb{Z}_q$ such that $c = f^r \prod_{i=1}^n g_i^{m_i}$. Open also receives some private information that was created during the commitment.

The generalized Pedersen commitment is perfect hiding and computationally binding if the discrete logarithm problem is hard in $\mathbb{G}$. It is additively homomorphic: $\mathsf{Com}_{ck}(\boldsymbol{m_1}; r_1) \cdot \mathsf{Com}_{ck}(\boldsymbol{m_2}; r_2) = \mathsf{Com}_{ck}(\boldsymbol{m_1} + \boldsymbol{m_2}; r_1 + r_2)$.

In the plain model, if Alice wants to commit N elements to Bob, the best communication complexity with generalized Pedersen commitment scheme is $\mathcal{O}(\sqrt{N})$. Namely, Bob first sends to Alice $n := \sqrt{N}$ commitment keys ck, and Alice commits and sends to Bob $\sqrt{N}$ commitments.

4 Fully Simulatable $\mathsf{OT}^N_{k\times 1}$ With Square-root Communication

We now propose a fully simulatable $\mathsf{OT}^N_{k\times 1}$ protocol with a square-root overall communication complexity. The basic idea comes from the classic KO *private*

Common input: $\sigma := (p, a, b, g, q, \zeta) \leftarrow \mathsf{Gen}_{\mathsf{gk}}(1^\lambda)$.
Sender's input: $\boldsymbol{M} := \{m_{i,j}\}_{i,j\in[n]} = \{\boldsymbol{m}_j\}_{j\in[n]}$, where $\boldsymbol{m}_j = (m_{1,j}, \ldots, m_{n,j})^T$.
Receiver's output: $i_\ell := m_{x_\ell, y_\ell}$ for the ℓ-th transfer phase, where $x_\ell, y_\ell \in [n]$.

Initialization Phase:

1. $\mathbf{R}$ computes $(pk, sk) \leftarrow \mathsf{Gen}_{\mathsf{pkc}}(\sigma)$ and $ck \leftarrow \mathsf{Gen}_{\mathsf{ped}}(\sigma)$, and sends pk, ck to $\mathbf{S}$.
2. $\mathbf{S}$ picks $\boldsymbol{r} \leftarrow_\$ \mathbb{Z}_q^n$ and sends $\{c_i := \mathsf{Com}_{ck}(\boldsymbol{m}_i; r_i)\}_{i=1}^n$, where $\boldsymbol{r} := (r_1, \ldots, r_n)^T$.

The ℓ-th Transfer Phase:

1. $\mathbf{R}$ sets two unit vectors $\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^n$ according to (x_ℓ, y_ℓ), i.e., $u_{x_\ell} = 1$, $v_{y_\ell} = 1$ and the rest are 0's. He picks $\boldsymbol{t_u}, \boldsymbol{t_v} \leftarrow_\$ \mathbb{Z}_q^n$, and sends $\boldsymbol{e_u} := \{e_u^{(i)} := \mathsf{Enc}_{pk}(u_i; t_u^{(i)})\}_{i=1}^n$ and $\boldsymbol{e_v} := \{e_v^{(i)} := \mathsf{Enc}_{pk}(v_i; t_v^{(i)})\}_{i=1}^n$ to $\mathbf{S}$, where $\boldsymbol{u} := (u_1, \ldots, u_n)^T$, $\boldsymbol{v} := (v_1, \ldots, v_n)^T$, $\boldsymbol{t_u} := (t_u^{(1)}, \ldots, t_u^{(n)})^T$ and $\boldsymbol{t_v} := (t_v^{(1)}, \ldots, t_v^{(n)})^T$.
2. $\mathbf{R}$ proves to $\mathbf{S}$ that $\boldsymbol{e_u}, \boldsymbol{e_v}$ encrypt unit vectors by *unit vector PoK* (c.f. Sect. 4.1).
3. $\mathbf{S}$ picks $\boldsymbol{t_s} \leftarrow_\$ \mathbb{Z}_q^n, \boldsymbol{a}, \boldsymbol{t_a} \leftarrow_\$ \mathbb{Z}_q^n, t_z \leftarrow_\$ \mathbb{Z}_q$. $\mathbf{S}$ sets for $i \in [n]$, $c_a^{(i)} = \mathsf{Com}_{ck}(a_i; t_a^{(i)})$, $w_i = \mathsf{Enc}_{pk}(a_i; t_s^{(i)}) \prod_{j=1}^n (e_u^{(j)})^{m_{j,i}}$, and $z = \mathsf{Enc}_{pk}(0; t_z) \prod_{j=1}^n (e_v^{(j)})^{a_j}$.
4. $\mathbf{S}$ sends $\boldsymbol{c_a}, \boldsymbol{w}, z$ to $\mathbf{R}$.
5. $\mathbf{S}$ proves to $\mathbf{R}$ that
   - $\boldsymbol{w}, \boldsymbol{e_u}, \boldsymbol{c}, \boldsymbol{c_a}$ are consistent by *masked multi-exp batch AoK* (c.f. Sect. 4.3);
   - $\boldsymbol{e_u}, z, \boldsymbol{c_a}$ are consistent by *multi-exp AoK* (c.f. Sect. 4.2, below).
6. $\mathbf{R}$ computes $g^{a_{y_\ell}} = z_2 \cdot z_1^{-sk}$ and then returns m_{i_ℓ} as $\mathsf{DL}_g(w_{x_\ell,2} \cdot w_{x_\ell,1}^{-sk} \cdot (g^{a_{y_\ell}})^{-1})$.

Figure1. Fully Simulatable $\mathsf{OT}_{k\times 1}^N$ With Square-root Communication

information retrieval (PIR) scheme [23]. Intuitively, when not concerned about privacy, the receiver sends two n-dimensional unit-vectors $\boldsymbol{u}, \boldsymbol{v}$ to the sender, where $n = \sqrt{N}$. The sender computes and sends $m^* = \boldsymbol{u} \cdot \boldsymbol{M} \cdot \boldsymbol{v}^T$ to the receiver, where $\boldsymbol{M} = \{m_{i,j}\}_{i,j\in[n]}$ is the sender's database.

Both the ElGamal encryption scheme and the generalized Pedersen commitment scheme are based on elliptic curves, so the membership of a group element is efficiently decidable. Hence, during our protocol, if the message consists of group elements/generators, the parties always first check their group membership. We will not mention this step in the protocol description explicitly.

We give the protocol description in Fig. 1. Our $\mathsf{OT}_{k\times 1}^N$ scheme consists of the initialization phase and the transfer phase. If the prover is honest, then w_i encrypts $a_i + \sum m_{j,i} u_j = a_i + m_{x_\ell,i}$ and z encrypts $\sum a_j v_j = a_{y_\ell}$, and thus the verifier can retrieve m_{x_ℓ,y_ℓ} as claimed.We show how to construct the SHVZK proofs/arguments in the following sections. All the SHVZK proofs/arguments should be compiled to general ones via a standard transformation by using commitments and a public coin flipping protocol, e.g. [9]. To keep the exposition simple, we will not explicitly mention the transformation in the protocol.

4.1 SHVZK Unit Vector Proof

Now we show how to construct the SHVZK Unit Vector Proof that is used in our $\mathsf{OT}^N_{k\times 1}$ scheme. In a Unit Vector Proof, given an encrypted vector $\boldsymbol{e} := (e_1, \ldots, e_n)^T = (\mathsf{Enc}_{pk}(b_1; r_1), \ldots, \mathsf{Enc}_{pk}(b_n; r_n))^T \in (\mathbb{G} \times \mathbb{G})^n$, the prover wants to convince the verifier that $\boldsymbol{b} := (b_1, \ldots, b_n)^T \in \mathbb{Z}_q^n$ is a unit vector, i.e., there is exactly one $i \in [n]$ such that $b_i = 1$ and $\forall j \neq i : b_j = 0$. Considering the lifted ElGamal encryption, we have $e_i := (e_{i,1}, e_{i,2}) = (g^{r_i}, g^{b_i} h^{r_i})$. As depicted in Fig. 2, we give a ZK proof of knowledge (PoK) of $\boldsymbol{b}, \boldsymbol{r} \in \mathbb{Z}_q^n$ such that for $i \in [n]$, $e_i = \mathsf{Enc}_{pk}(b_i; r_i)$, $b_i \in \{0, 1\}$ and $\sum_{i=1}^n b_i = 1$, where $\boldsymbol{r} := (r_1, \ldots, r_n)^T$. The proof uses $\vee, \wedge$ compositions [**?**] of the basic Σ protocol to prove that e_i encrypts 0 or 1, based on DDH tuple proof technique [7].

Common input: Group information σ and the public key $pk := h$.
Prover's private input: $\boldsymbol{b} \in \{0,1\}^n$ and $\boldsymbol{r} \in \mathbb{Z}_q^n$.
Statement: $\{(e_{i,1}, e_{i,2}) := (g^{r_i}, g^{b_i} h^{r_i})\}_{i=1}^n$. Let $E_1 := \prod_{i=1}^n e_{i,1}$, $E_2 := \prod_{i=1}^n e_{i,2}$.

1. Since $b_i \in \{0,1\}$, let $\bar{b}_i := 1 - b_i$. For $i \in [n]$, the prover picks random $s_i, \rho_{i,\bar{b}_i}, z_{i,\bar{b}_i} \leftarrow_\$ \mathbb{Z}_q$ and computes $a_{i,1}^{(b_i)} = g^{s_i}, a_{i,2}^{(b_i)} = h^{s_i}, a_{i,1}^{(\bar{b}_i)} = g^{z_{i,\bar{b}_i}} e_{i,1}^{-\rho_{i,\bar{b}_i}}$ and $a_{i,2}^{(\bar{b}_i)} = h^{z_{i,\bar{b}_i}} \cdot (e_{i,2} \cdot g^{-\bar{b}_i})^{-\rho_{i,\bar{b}_i}}$. He picks random $s \leftarrow_\$ \mathbb{Z}_q$, sets $A_1 = g^s, A_2 = h^s$, and sends $\{(a_{i,1}^{(0)}, a_{i,2}^{(0)}), (a_{i,1}^{(1)}, a_{i,2}^{(1)})\}_{i=1}^n$ and (A_1, A_2) to the verifier.
2. The verifier picks random challenge $\rho \leftarrow_\$ \mathbb{Z}_q^*$ and sends ρ to the prover.
3. For $i \in [n]$, the prover sets $\rho_{i,b_i} = \rho - \rho_{i,\bar{b}_i}$ and $z_{i,b_i} = r_i \rho_{i,b_i} + s_i$. He computes $Z = \rho \cdot \sum_{i=1}^n r_i + s$, and sends $\{\rho_{i,0}, z_{i,0}, z_{i,1}\}_{i=0}^n$ and Z to the verifier.

Verification:

1. The verifier checks $E_1^\rho A_1 = g^Z \wedge (E_2/g)^\rho A_2 = h^Z$. For $i \in [n]$, the verifier computes $\rho_{i,1} = \rho - \rho_{i,0}$ and checks

$$e_{i,1}^{\rho_{i,0}} a_{i,1}^{(0)} = g^{z_{i,0}} \wedge e_{i,2}^{\rho_{i,0}} a_{i,2}^{(0)} = h^{z_{i,0}} \wedge e_{i,1}^{\rho_{i,1}} a_{i,1}^{(1)} = g^{z_{i,1}} \wedge (e_{i,2}/g)^{\rho_{i,1}} a_{i,1}^{(1)} = h^{z_{i,1}}.$$

Figure2. Public Coin SHVZK Unit Vector Proof

Theorem 1. *The protocol depicted in Fig. 2 is a* 3*-move public coin perfect special honest verifier zero-knowledge proof of knowledge of* $\boldsymbol{b}$ *and* $\boldsymbol{r}$ *such that* $e_i = \mathsf{Enc}_{pk}(b_i; r_i) \wedge b_i \in \{0,1\} \wedge \sum_{i=1}^n b_i = 1$.

Proof. For perfect completeness, if $b_i \in \{0,1\}$, it is easy to verify that all the equations hold. For soundness, we have to construct an extractor $\mathcal{X}$ that runs on $\langle \mathcal{P}^*, \mathcal{V} \rangle$ to get a transcript. It rewinds the protocol to the challenge phase and runs it with fresh challenges until it has 2 acceptable proofs. Assuming the prover $\mathcal{P}$ has probability of $p(\lambda)$ of making an acceptable proof, so the extractor $\mathcal{X}$ will take an average of $2/p(\lambda)$ rewinds, which is polynomial running time. Thus, there is overwhelming probability that we have transcripts with 2 different challenges $\rho^{(1)}, \rho^{(2)}$. From those transcripts, the extractor can extract the knowledge $\boldsymbol{b}$ and $\boldsymbol{r}$. Namely, for each $i \in [n]$, we have at least one different pair between

$(\rho_{i,0}^{(1)}, \rho_{i,0}^{(2)})$ and $(\rho_{i,1}^{(1)}, \rho_{i,1}^{(2)})$. Assume $\rho_{i,x}^{(1)}, \rho_{i,x}^{(2)}$ are different, we can compute $r_i = (z_{i,x}^{(1)} - z_{i,x}^{(2)})/(\rho_{i,x}^{(1)} - \rho_{i,x}^{(2)})$. Subsequently, $\mathcal{X}$ can extract b_i by checking e_i. Hence, we have constructed an extractor $\mathcal{X}$ that outputs $\boldsymbol{b}$ and $\boldsymbol{r}$.

For perfect zero-knowledge, we construct a simulator $\mathcal{S}$ that on challenge ρ outputs simulated proof that is indistinguishable from a real proof with challenge ρ. On challenge ρ, for $i \in [n]$, $\mathcal{S}$ randomly picks $\rho_{i,0}, z_{i,0}, z_{i,1}, Z \leftarrow_{\$} \mathbb{Z}_q$ and computes $\rho_{i,1} = \rho - \rho_{i,0}, A_1 = g^Z E_1^{-\rho}, A_2 = h^Z (E_2/g)^{-\rho}, a_{i,1}^{(0)} = g^{z_{i,0}} e_{i,1}^{-\rho_{i,0}}, a_{i,2}^{(0)} = h^{z_{i,0}} e_{i,2}^{-\rho_{i,0}}, a_{i,1}^{(1)} = g^{z_{i,1}} e_{i,1}^{-\rho_{i,1}}, a_{i,2}^{(1)} = h^{z_{i,1}} (e_{i,2}/g)^{-\rho_{i,1}}$. $\mathcal{S}$ outputs

$$\tau^* := \left(\{(a_{i,1}^{(0)}, a_{i,2}^{(0)}), (a_{i,1}^{(1)}, a_{i,2}^{(1)})\}_{i=1}^n, (A_1, A_2), \rho, \{\rho_{i,0}, z_{i,0}, z_{i,1}\}_{i=0}^n \right) .$$

Note that simulated $z_{i,0}, z_{i,1}, Z$ have the same distribution as in the real proof, because s_i, s are uniformly random. It is easy to see that ρ_0 and ρ_1 have identical distribution of them in a real proof. Finally, we argue that $\{(a_{i,1}^{(0)}, a_{i,2}^{(0)}), (a_{i,1}^{(1)}, a_{i,2}^{(1)})\}_{i=1}^n, (A_1, A_2)$ are uniquely determined for fixed $\rho_0, \rho_0, z_{i,0}, z_{i,1}, Z$. Therefore, we have shown that the distribution of simulated τ^* is identical to τ in a real proof. □

4.2 Multi-exponentiation Argument

In Fig. 3, we give an argument of knowledge of $\boldsymbol{m} := (m_1, \ldots, m_n)^T, \boldsymbol{r} := (r_1, \ldots, r_n)^T \in \mathbb{Z}_q^n$ and $t \in \mathbb{Z}_q$ such that $v = \mathsf{Enc}_{pk}(0; t) \prod_{j=1}^n e_j^{m_j}$ and $c_i = \mathsf{Com}_{ck}(m_i; r_i)$, for $i \in [n]$.

Common input: Group information σ and pk, ck.
Statement: $\boldsymbol{e} \in (\mathbb{G} \times \mathbb{G})^n, v \in (\mathbb{G} \times \mathbb{G})$ and $\boldsymbol{c} \in \mathbb{G}^n$.
Prover's private input: $\boldsymbol{m}, \boldsymbol{r} \in \mathbb{Z}_q^n$ and $t \in \mathbb{Z}_q$.

1. The prover picks $\boldsymbol{x}, \boldsymbol{y} \leftarrow_{\$} \mathbb{Z}_q^n, t' \leftarrow_{\$} \mathbb{Z}_q$ and sends $v' := \mathsf{Enc}_{pk}(0; t') \prod_{i=1}^n e_i^{x_i}$, $u_i := \mathsf{Com}_{ck}(x_i; y_i)$ to the verifier.
2. The verifier picks a random challenge $\rho \leftarrow_{\$} \mathbb{Z}_q^*$ and sends ρ to the prover.
3. The prover sends $\boldsymbol{w} := \rho \cdot \boldsymbol{m} + \boldsymbol{x}$, $\hat{t} = \rho \cdot t + t'$ and $\boldsymbol{z} := \rho \cdot \boldsymbol{r} + \boldsymbol{y}$ to the verifier.

Verification:

1. The verifier checks $c_i^\rho u_i = \mathsf{Com}_{ck}(w_i; z_i) \wedge v^\rho v' = \mathsf{Enc}_{pk}(0; \hat{t}) \prod_{i=1}^n e_i^{w_i}$.

Figure3. Public Coin SHVZK Multi-exponentiation Argument

Theorem 2. *The protocol depicted in Fig. 3 is a* 3*-move public coin perfect special honest verifier zero-knowledge argument of knowledge of* $\boldsymbol{m}, \boldsymbol{r}, t$ *such that* $v = \mathsf{Enc}_{pk}(0; t) \prod_{j=1}^n e_j^{m_j} \wedge c_i = \mathsf{Com}_{ck}(m_i; r_i)$.

Proof. For perfect completeness, it is easy verify that all the equations hold. Now we prove soundness and show that the protocol is an argument of knowledge (AoK). Since $\rho \in \mathbb{Z}_q^*$ is randomly chosen, by Schwartz-Zippel lemma, the

prover has negligible probability of convincing the verifier unless all ρ related terms match on each side of the equality. Now we construct the witness-extended emulator $\mathcal{X}$ runs $\langle \mathcal{P}^*, \mathcal{V} \rangle$ to get a transcript. If the prover $\mathcal{P}$ has probability $p(\lambda)$ of making an acceptable argument, the black-box witness-extended emulator $\mathcal{X}$ also has success probability $p(\lambda)$ to produce an accepting argument. It rewinds the protocol to the challenge phase and runs it with fresh challenges until it has 2 acceptable arguments. Since the prover $\mathcal{P}$ has probability $p(\lambda)$ of making an accepting argument in the first place, the emulator $\mathcal{X}$ will take an average of $2/p(\lambda)$ rewinds, which is polynomial running time. Again, there is overwhelming probability that we have transcripts with 2 different challenges $\rho^{(1)}, \rho^{(2)}$. After obtaining $\boldsymbol{w}^{(\eta)} = \rho^{(\eta)} \cdot \boldsymbol{m} + \boldsymbol{x}$, $\hat{t}^{(\eta)} = \rho^{(\eta)} \cdot t + t'$, $\boldsymbol{z}^{(\eta)} = \rho^{(\eta)} \cdot \boldsymbol{r} + \boldsymbol{y}$ for $\eta \in \{1, 2\}$, $\mathcal{X}$ computes $\boldsymbol{m} = (\boldsymbol{w}^{(1)} - \boldsymbol{w}^{(2)})/(\rho^{(1)} - \rho^{(2)})$, $t = (\hat{t}^{(1)} - \hat{t}^{(2)})/(\rho^{(1)} - \rho^{(2)})$ and $\boldsymbol{r} = (\boldsymbol{z}^{(1)} - \boldsymbol{z}^{(2)})/(\rho^{(1)} - \rho^{(2)})$. Hence, we have extracted a valid witness $\boldsymbol{m}, r, t$ for the statement.

For perfect zero-knowledge, we have to construct a simulator $\mathcal{S}$ on challenge ρ outputs the simulated argument that is indistinguishable from a real argument with challenge ρ. On challenge ρ, the simulator $\mathcal{S}$ randomly picks $\boldsymbol{w}, \boldsymbol{z} \leftarrow_{\$} \mathbb{Z}_q^n$ and $\hat{t} \leftarrow_{\$} \mathbb{Z}_q$. $\mathcal{S}$ computes $v' = \mathsf{Enc}_{pk}(0; \hat{t}) \prod_{i=1}^{n} e_i^{w_i} \cdot v^{-\rho}$ and $u_i = \mathsf{Com}_{ck}(w_i; z_i) \cdot c_i^{-\rho}$. $\mathcal{S}$ outputs $\tau^* := (v', \boldsymbol{u}, \rho, \boldsymbol{w}, \hat{t}, \boldsymbol{z})$. Since $\boldsymbol{x}, \boldsymbol{y}, t$ are uniformly random in a real argument, the distribution of simulated $\boldsymbol{w}, \hat{t}, \boldsymbol{z}$ is identical to the distribution of them in a real argument. Furthermore, $v', \boldsymbol{u}$ are uniquely determined for fixed $\rho, \boldsymbol{w}, \hat{t}, \boldsymbol{z}$; therefore, simulated τ^* has the same distribution as τ in a real argument. □

4.3 Masked Multi-exponentiation Batch Argument

In this section, we propose the masked multi-exponentiation batch argument. Given two vectors of ciphertexts $\boldsymbol{e} := (e_1, \ldots, e_n)^T \in (\mathbb{G} \times \mathbb{G})^n$, $\boldsymbol{v} := (v_1, \ldots, v_\ell)^T \in (\mathbb{G} \times \mathbb{G})^\ell$ and two vectors of commitments $\boldsymbol{c} := (c_1, \ldots, c_\ell)^T \in \mathbb{G}^\ell$ and $\boldsymbol{u} := (u_1, \ldots, u_\ell)^T \in \mathbb{G}^\ell$, as depicted in Fig. 4, we will give an argument of knowledge of $\boldsymbol{M} := \{m_{j,i}\}_{j,i=1}^{n,\ell} \in \mathbb{Z}_q^{n \times \ell}$, $\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d} \in \mathbb{Z}_q^\ell$ such that for $i \in [\ell]$,

$$v_i = \mathsf{Enc}_{pk}(s_i; t_i) \prod_{j=1}^{n} e_j^{m_{j,i}} \quad , \quad u_i = \mathsf{Com}_{ck}(s_i; d_i) \quad \text{and} \quad c_i = \mathsf{Com}_{ck}(\boldsymbol{m}_i; r_i)$$

where $\boldsymbol{m}_i := (m_{1,i}, \ldots, m_{n,i})^T$, $\boldsymbol{r} := (r_1, \ldots, r_\ell)^T$, $\boldsymbol{s} := (s_1, \ldots, s_\ell)^T$, $\boldsymbol{t} := (t_1, \ldots, t_\ell)^T$ and $\boldsymbol{d} := (d_1, \ldots, d_\ell)^T$.

Theorem 3. *The protocol depicted in Fig. 4 is a 3-move public coin perfect special honest verifier zero-knowledge argument of knowledge of $\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d}$ such that for $i \in [\ell]$,*

$$v_i = \mathsf{Enc}_{pk}(s_i; t_i) \prod_{j=1}^{n} e_j^{m_{j,i}} \quad , \quad u_i = \mathsf{Com}_{ck}(s_i; d_i) \quad \text{and} \quad c_i = \mathsf{Com}_{ck}(\boldsymbol{m}_i; r_i) \ .$$

Common input: Group information σ and pk, ck.
Statement: $\boldsymbol{e} \in (\mathbb{G} \times \mathbb{G})^n, \boldsymbol{v} \in (\mathbb{G} \times \mathbb{G})^\ell$ and $\boldsymbol{c}, \boldsymbol{u} \in \mathbb{G}^\ell$.
Prover's private input: $\boldsymbol{M} \in \mathbb{Z}_q^{n \times \ell}$ and $\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d} \in \mathbb{Z}_q^\ell$.

1. The prover picks $\boldsymbol{x} \leftarrow_\$ \mathbb{Z}_q^n, s_0, t_0, r_0, d_0 \leftarrow_\$ \mathbb{Z}_q$ and sends $v_0 := \mathsf{Enc}_{pk}(s_0; t_0) \prod_{i=1}^n e_i^{x_i}$, $c_x := \mathsf{Com}_{ck}(\boldsymbol{x}; r_0)$ and $u_0 := \mathsf{Com}_{ck}(s_0; d_0)$ to the verifier.
2. The verifier randomly picks a challenge $\rho \leftarrow_\$ \mathbb{Z}_q^*$ and sends ρ to the prover.
3. Set $\boldsymbol{\rho} := (\rho, \rho^2, \ldots, \rho^\ell)^T$. The prover sends $\boldsymbol{w} := \boldsymbol{M} \cdot \boldsymbol{\rho} + \boldsymbol{x}$, $t' = \boldsymbol{t}^T \cdot \boldsymbol{\rho} + t_0$, $s' = \boldsymbol{s}^T \cdot \boldsymbol{\rho} + s_0$, $d' = \boldsymbol{d}^T \cdot \boldsymbol{\rho} + d_0$ and $r' := \boldsymbol{r}^T \cdot \boldsymbol{\rho} + r_0$ to the verifier, where $\boldsymbol{w} := (w_1, \ldots, w_n)^T$.

Verification:

1. The verifier checks

$$u_0 \prod_{i=1}^{\ell} u_i^{\rho^i} = \mathsf{Com}_{ck}(s'; d') \wedge c_x \prod_{i=1}^{\ell} c_i^{\rho^i} = \mathsf{Com}_{ck}(\boldsymbol{w}; r') \wedge v_0 \prod_{i=1}^{\ell} v_i^{\rho^i} = \mathsf{Enc}_{pk}(s'; t') \prod_{i=1}^{n} e_i^{w_i}.$$

Figure4. Public Coin SHVZK Masked Multi-exponentiation Batch Argument

Proof. For perfect completeness, it is easy to verify that all the equations hold. Now we prove soundness and show that the protocol is an argument of knowledge (AoK), by showing that it has a witness-extended emulator. Since $\rho \in \mathbb{Z}_q^*$ is randomly chosen, by Schwartz-Zippel lemma, the prover has negligible probability of convincing the verifier unless all ρ^i related terms match on each side of the equality for all $i \in [\ell]$. The witness-extended emulator $\mathcal{X}$ runs $\langle \mathcal{P}^*, \mathcal{V} \rangle$ to get a transcript. If the prover $\mathcal{P}$ has probability $p(\lambda)$ of making an acceptable argument, the black-box witness-extended emulator $\mathcal{X}$ also has success probability $p(\lambda)$ to produce an accepting argument. It rewinds the protocol to the challenge phase and runs it with fresh challenges until it has $\ell + 1$ acceptable arguments. Since the prover $\mathcal{P}$ has probability $p(\lambda)$ of making an accepting argument in the first place, the emulator $\mathcal{X}$ will take an average of $\frac{\ell+1}{p(\lambda)}$ rewinds, which takes $\mathrm{poly}(\lambda)$ running time. Again, there is overwhelming probability that we have transcripts with $\ell + 1$ different challenges. The $\ell + 1$ different challenges give us a $(\ell + 1) \times (\ell + 1)$ transposed Vandermonde matrix

$$\boldsymbol{V} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \rho^{(1)} & \rho^{(2)} & \cdots & \rho^{(\ell+1)} \\ \vdots & \vdots & \ddots & \vdots \\ (\rho^{(1)})^\ell & (\rho^{(2)})^\ell & \cdots & (\rho^{(\ell+1)})^\ell \end{pmatrix}.$$

Note that $\boldsymbol{V}$ is invertible because $\rho^{(1)}, \ldots, \rho^{(\ell+1)}$ are different, and $\mathcal{X}$ computes $\boldsymbol{V}^{-1}$. Let $\boldsymbol{M_x}$ be the $n \times (\ell + 1)$ matrix that is the column $\boldsymbol{x}$ concatenated at the left side of $\boldsymbol{M}$ and denote $\boldsymbol{W}$ as the matrix that consists of columns $(\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(\ell+1)})$. We have $\boldsymbol{W} = \boldsymbol{M_x} \cdot \boldsymbol{V}$, and $\mathcal{X}$ can compute $\boldsymbol{M_x} = \boldsymbol{W} \cdot$

$\boldsymbol{V}^{-1}$. Similarly, $\mathcal{X}$ can extract $\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d}$; hence, $\mathcal{X}$ has extracted a valid witness $\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d}$ for the statement.

For perfect zero-knowledge, we have to construct a simulator $\mathcal{S}$ on challenge ρ outputs the simulated argument that is indistinguishable from a real argument with challenge ρ. On challenge ρ, the simulator $\mathcal{S}$ randomly picks $\boldsymbol{w} \leftarrow_{\$} \mathbb{Z}_q^n$ and $r', s', t', d' \leftarrow_{\$} \mathbb{Z}_q$. $\mathcal{S}$ computes $u_0 = \mathsf{Com}_{ck}(s'; d')/(\prod_{i=1}^{\ell} u_i^{\rho^i})$, $v_0 = (\mathsf{Enc}_{pk}(s'; t') \prod_{i=1}^{n} e_i^{w_i})/(\prod_{i=1}^{\ell} v_i^{\rho^i})$ and $c_x = \mathsf{Com}_{ck}(\boldsymbol{w}; r')/(\prod_{i=1}^{\ell} c_i^{\rho^i})$. $\mathcal{S}$ outputs $\tau^* := (u_0, v_0, c_x, \rho, \boldsymbol{w}, r', s', t', d')$. Since $\boldsymbol{x}, r_0, s_0, t_0, d_0$ are uniformly random in a real argument, the distribution of simulated $\boldsymbol{w}, r', s', t', d'$ is identical to the distribution of them in a real argument. Furthermore, u_0, v_0, c_x are uniquely determined for fixed $\rho, \boldsymbol{w}, r', s', t', d'$, therefore, simulated τ^* is identical to the distribution of τ in a real argument. □

4.4 Security Analysis of Our $\mathsf{OT}_{k\times 1}^N$ Scheme

In this section, we examine the security of our $\mathsf{OT}_{k\times 1}^N$ scheme in Fig. 1. Since w_i in step 3 of the transfer phase is masked by a_i, it does not reveal information about $\boldsymbol{M}$; therefore, the receiver can only decrypt one document in each transfer phase. In our security proof of fully simulation, we don't consider the initialization phase and transfer phase as separated experiments. One may add argument of knowledge of the openings of commitment $\boldsymbol{c}$ [15] in the initialization phase in order to exact the sender's input $\boldsymbol{M}$ in the initialization phase. Note that it is the receiver's responsibility to choose correct commitment key ck to achieve the binding property. Since, the order of $\mathbb{G}$ is q, the sender only needs to check group membership of ck to guarantee that his commitments will not reveal anything information about the messages even if the receiver is cheating. Its formal security proof is given in App. A.

4.5 Implementation and Efficiency

In terms of communicational efficiency, it is clear that the proposed $\mathsf{OT}_{k\times 1}^N$ scheme (shown in Fig. 1) costs $\mathcal{O}(\sqrt{N})$ in both the initialization phase and each transfer phase. Let $k = 1$, as far as we know, our proposed OT_1^N is the first fully simulatable OT_1^N that achieves $\mathcal{O}(\sqrt{N})$ communication complexity. The computation complexity of our proposed $\mathsf{OT}_{k\times 1}^N$ scheme is $\mathcal{O}(N)$ in both initialization phase and each transfer phase. As mentioned before, since the scheme uses lifted ElGamal encryption, the message space should be small enough to compute discrete logarithm, e.g., $m_i \in \{0,1\}^{\xi}$, where $\xi \leq 30$.

In practical implementation, the actual complexity of our protocol is smaller. Since the protocol only uses multi-exponentiation operations in both homomorphic operations and commitments. We employ Lim's multi-exponentiation algorithm to reduce the actual computation. In [25], Lim showed how to compute a product of n exponentiations using only $\mathcal{O}(\frac{n}{\log n})$ multiplications. We implemented the proposed $\mathsf{OT}_{k\times 1}^N$ scheme on elliptic curve group over $\mathbb{F}_p$. The performance benchmark is tested with the 192-bit elliptic curve domain parameters

	Initialization phase			Each transfer phase		
DB size	**S**'s r.t. (s)	**R**'s r.t. (s)	Comm. (byte)	**S**'s r.t. (s)	**R**'s r.t. (s)	Comm. (byte)
1×10^4	0.06	0.045	4065	0.98	1.16	44320
2.5×10^5	0.29	0.565	20165	4.9	7.24	220320
1×10^6	0.59	1.975	40290	9.7	17.68	440320
2.5×10^7	2.92	43.555	201290	48.61	223.25	2200320
1×10^8	5.83	171.24	402540	96.94	786.77	4400320

Table2. Performance Benchmark. (r.t. stands for running time. Messages are chosen from $\{0,1\}^{10}$, and the network delay is not considered.)

recommended by NIST p192, where $p = 2^{192} - 2^{64} - 1$, which gives about 96-bit security level. In order to save communication bandwidth, we also used the standard point compression technique: a point on $E(\mathbb{F}_p)$ is represented by its x coordinate together with the least significant bit of its y coordinate. The code is implemented in C++, using *Multi-precision Integer and Rational Arithmetic C/C++ Library* (MIRACL) crypto SDK. All the tests are performed on a linux desktop with an Intel Core $i5$-2400 CPU running at 3.10 GHz. Table 2 depicts the sender's and receiver's running time (in seconds) as well as the communication complexity (in bytes) in both initialization phase and each transfer phase. We can see our scheme is very efficient even with relatively large database size.

5 Conclusions

In this paper, we proposed an efficient $\mathsf{OT}^N_{k\times 1}$ scheme in the plain model. It achieves fully simulatable security with $\mathcal{O}(\sqrt{N})$ communication in both the initialization phase and each transfer phase. Ideally, the scheme is dedicated to 1-out-of-N oblivious transfer, whereas it also achieves better (amortized) communication, comparing with existing schemes when $k = \mathcal{O}(N^{1/2})$, which covers majority OT usage cases. We also implemented and highly optimized the proposed scheme, and its perform benchmark shows very impressive results. When k is very large, say $\mathcal{O}(N)$, we recommend the user to adopt ORAM based two-party computation schemes, e.g. [11], so the cost of each transfer is minimum after the setup phase. We would like to further reduce the communication complexity of fully simulatable OT^N_1 in our future research.

Acknowledgements. The second author was supported by Estonian Research Council, the Tiger University Program of the Estonian Information Technology Foundation, and European Union through the European Regional Development Fund. The last author was supported in part by US National Science Foundation under grants CNS-1262277 and CNS-1116939.

References

1. Multi-query Computationally-Private Information Retrieval with Constant Communication Rate. In: PKC (2010)
2. Bayer, S., Groth, J.: Efficient Zero-knowledge Argument for Correctness of a Shuffle. In: EUROCRYPT (2012)
3. Bellare, M., Goldreich, O.: On Defining Proofs of Knowledge. In: CRYPTO (1993)
4. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.P.: Short PCPs Verifiable in Polylogarithmic Time. In: CCC (2005)
5. Camenisch, J., Neven, G., Shelat, A.: Simulatable Adaptive Oblivious Transfer. In: EUROCRYPT (2007)
6. Canetti, R., Fischlin, M.: Universally Composable Commitments. In: CRYPTO (2001)
7. Chaum, D.: Zero-Knowledge Undeniable Signatures (extended abstract). In: EUROCRYPT (1990)
8. Damgård, I., Fujisaki, E.: A Statistically Hiding Integer Commitment Scheme Based on Groups with Hidden Order. In: ASIACRYPT (2002)
9. Damgård, I., Goldreich, O., Okamoto, T., Wigderson, A.: Honest Verifier vs Dishonest Verifier in Public Cain Zero-Knowledge Proofs. In: CRYPTO (1995)
10. Gentry, C., Ramzan, Z.: Single-Database Private Information Retrieval with Constant Communication Rate. In: ICALP (2005)
11. Gordon, S.D., Katz, J., Kolesnikov, V., Krell, F., Malkin, T., Raykova, M., Vahlis, Y.: Secure Two-party Computation in Sublinear (amortized) Time. In: CCS (2012)
12. Green, M., Hohenberger, S.: Blind Identity-Based Encryption and Simulatable Oblivious Transfer. In: ASIACRYPT (2007)
13. Green, M., Hohenberger, S.: Universally Composable Adaptive Oblivious Transfer. In: ASIACRYPT (2008)
14. Green, M., Hohenberger, S.: Practical Adaptive Oblivious Transfer from Simple Assumptions. In: TCC (2011)
15. Groth, J.: Linear Algebra with Sub-linear Zero-Knowledge Arguments. In: CRYPTO (2009)
16. Groth, J.: A Verifiable Secret Shuffle of Homomorphic Encryptions. Journal of Cryptology 23, 546–579 (2010)
17. Groth, J.: Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In: ASIACRYPT (2010)
18. Ishai, Y., Paskin, A.: Evaluating Branching Programs on Encrypted Data. In: TCC (2007)
19. Jarecki, S., Liu, X.: Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: TCC (2009)
20. Kurosawa, K., Nojima, R.: Simple Adaptive Oblivious Transfer without Random Oracle. In: ASIACRYPT (2009)
21. Kurosawa, K., Nojima, R., Phong, L.T.: Efficiency-improved fully simulatable adaptive OT under the DDH assumption. In: SCN (2010)
22. Kurosawa, K., Nojima, R., Phong, L.T.: Generic Fully Simulatable Adaptive Oblivious Transfer. In: ACNS (2011)
23. Kushilevitz, E., Ostrovsky, R.: Replication is NOT Needed: Single Database, Computationally-Private Information Retrieval. In: FOCS (1997)
24. Laur, S., Lipmaa, H.: On the Feasibility of Consistent Computations. In: PKC (2010)

25. Lim, C.H.: Efficient Multi-exponentiation and Application to Batch Verification of Digital Signatures (2000), online Tech. Report: http://dasan.sejong.ac.kr/ chlim/pub/multiexp.ps
26. Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: ISC (2005)
27. Lipmaa, H.: First CPIR Protocol with Data-Dependent Computation. In: ICISC (2009)
28. Lipmaa, H.: Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In: TCC (2012)
29. Liskova, L., Stanek, M.: Efficient Simultaneous Contract Signing. In: SEC (2004)
30. Naor, M., Pinkas, B.: Oblivious Transfer with Adaptive Queries. In: CRYPTO (1999)
31. Naor, M., Pinkas, B.: Computationally Secure Oblivious Transfer. Journal of Cryptology 18, 1–35 (2005), http://dx.doi.org/10.1007/s00145-004-0102-6
32. Pedersen, T.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: CRYPTO (1991)
33. Rial, A., Kohlweiss, M., Preneel, B.: Universally Composable Adaptive Priced Oblivious Transfer. In: Pairing (2009)
34. Yao, A.: Protocols for Secure Computations (Extended Abstract). In: FOCS (1982)
35. Zhang, B.: Simulatable Adaptive Oblivious Transfer with Statistical Receiver's Privacy. In: ProvSec (2011)

A Security proof of our $\mathsf{OT}^N_{k\times 1}$ scheme

Theorem 4. *The proposed $\mathsf{OT}^N_{k\times 1}$ scheme (as shown in Fig. 1) is secure against the sender corruption under the DDH assumption.*

Proof. We show that for every real-world cheating p.p.t. sender $\hat{\mathbf{S}}$ there exists an ideal-world cheating p.p.t. sender $\hat{\mathbf{S}}'$ such that for every distinguisher $\mathbf{D}$:

$$\mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}(N,k,m_1,\ldots,m_N,\mathcal{I}) \quad \overset{c}{\approx} \quad \mathbf{Ideal}_{\hat{\mathbf{S}}',\mathbf{R}'}(N,k,m_1,\ldots,m_N,\mathcal{I})$$

Considering a sequence of games $G_0,\ldots,G_4$, where Game $G_0 = \mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}$ and Game $G_4 = \mathbf{Ideal}_{\hat{\mathbf{S}}',\mathbf{R}'}$. We define

$$\mathsf{Adv}[\mathbf{D}] = \left|\Pr[\mathbf{D}(X)=1 : X \overset{\$}{\leftarrow} \mathbf{Ideal}_{\hat{\mathbf{S}}',\mathbf{R}'}] - \Pr[\mathbf{D}(X)=1 : X \overset{\$}{\leftarrow} \mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}]\right| \ .$$

Game G_0: The real-world experiment $\mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}$. By definition, $\Pr[\mathbf{D}(X)=1 : X \overset{\$}{\leftarrow} G_0] = Pr[\mathbf{D}(X)=1 : X \overset{\$}{\leftarrow} \mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}]$.

Game G_1: Game G_1 is the same as Game G_0 except the following. In the first transfer phase, the receiver uses the witness-extended emulator of the *masked multi-exponentiation batch AoK* to extract $\boldsymbol{M^*}, \boldsymbol{r^*}$ that is committed in $\boldsymbol{c}$. If extraction fails, then the protocol aborts. The failure probability is negligible. Furthermore, if the server can open the commitments to a different set $\boldsymbol{M'}, \boldsymbol{r'}$ from what is extracted, then we have broken the blinding property of generalized Pedersen Commitment; namely, the discrete logarithm assumption does not hold, neither does the DDH assumption. Assume the DDH problem is hard over $\mathbb{G}$, we have $\Pr[\mathbf{D}(X)=1 : X \overset{\$}{\leftarrow} G_1] \approx \Pr[\mathbf{D}(X)=1 : X \overset{\$}{\leftarrow} G_0]$.

Game G_2: Game G_2 is the same as Game G_1 except the following. In the initialization phase, the receiver randomly picks pk such that he does not know the discrete logarithm of pk. In the ℓ-th transfer phase, the receiver skips all the decryption steps, and returns $m^*_{i_\ell}$ according $\boldsymbol{M^*}$ that is extracted in Game G_1. Since all the zero-knowledge arguments and proofs are sound, we have $\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_2] \approx \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_1]$.

Game G_3: Game G_3 is the same as Game G_2 except the following. In the ℓ-th transfer phase, the receiver picks two random unit vectors $\boldsymbol{u}, \boldsymbol{v}$, regardless i_ℓ. Since ElGamal encryption is IND-CPA secure under the DDH assumption, we have $\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_3] \approx \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_2]$.

Game G_4: The ideal-world experiment $\mathbf{Ideal}_{\hat{\mathbf{S}}', \mathbf{R}'}$ in which an ideal-world sender $\hat{\mathbf{S}}'$ uses the real-world sender $\hat{\mathbf{S}}$ as a black-box as follows.

1. After receiving $(m_1, \ldots, m_N)$, $\hat{\mathbf{S}}'$ forwards them to $\hat{\mathbf{S}}$.
2. $\hat{\mathbf{S}}'$ acts as the receiver and plays Game G_3 with $\hat{\mathbf{S}}$.
3. In the first transfer phase, $\hat{\mathbf{S}}'$ sends $(m^*_1, \ldots, m^*_N)$ that is extracted in Game G_1 to $\mathcal{F}_{OT}^{n\times 1}$ (for the initialization phase). [2]
4. In the ℓ-th transfer phase, if $\hat{\mathbf{S}}$ behaved in an acceptable way, then $\hat{\mathbf{S}}'$ sends $b_\ell = 1$ to $\mathcal{F}_{OT}^{n\times 1}$. Otherwise, $\hat{\mathbf{S}}'$ sends $b_\ell = 0$ to $\mathcal{F}_{OT}^{n\times 1}$.

To sum up, it is easy to see that

$$\mathsf{Adv}(\mathbf{D}) = \left|\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_4] - \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_0]\right| \leq \epsilon(\lambda) \ ,$$

where $\epsilon(\cdot)$ is a negligible function. □

Theorem 5. *The proposed $\mathsf{OT}^N_{k\times 1}$ scheme (as shown in Fig. 1) is statistically secure against the receiver corruption.*

Proof. We show that for every real-world cheating p.p.t. receiver $\hat{\mathbf{R}}$ there exists an ideal-world cheating p.p.t. receiver $\hat{\mathbf{R}}'$ such that for every distinguisher $\mathbf{D}$:

$$\mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}(N, k, m_1, \ldots, m_N, \mathcal{I}) \quad \overset{c}{\approx} \quad \mathbf{Ideal}_{\mathbf{S}',\hat{\mathbf{R}}'}(N, k, m_1, \ldots, m_N, \mathcal{I})$$

Again, we consider a series of hybrid games $G_0, \ldots, G_4$, where Game $G_0 = \mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}$ and Game $G_4 = \mathbf{Ideal}_{\mathbf{S}',\hat{\mathbf{R}}'}$. We define

$$\mathsf{Adv}[\mathbf{D}] = \left|\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} \mathbf{Ideal}_{\mathbf{S}',\hat{\mathbf{R}}'}] - \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} \mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}]\right| \ .$$

Game G_0: The real-world experiment $\mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}$. By definition, $\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_0] = \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} \mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}]$.

[2] Remark: the experiments do not separate initialization phase and transfer phase.

Game G_1: Game G_1 is the same as Game G_0 except the following. In the ℓ-th transfer phase, the sender uses the knowledge extractor of *unit vector PoK* to extract the plaintext and randomizers of each ciphertext, i.e., those two unit vectors $\boldsymbol{u^*}, \boldsymbol{v^*}$. Subsequently, we know the index i_ℓ^*. If extraction fails, then the protocol aborts. Since the failure probability is negligible, $\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_1] \approx \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_0]$.

Game G_2: Game G_2 is the same as Game G_1 except the following. In each transfer phase, the sender uses the simulator of the *masked multi-exponentiation batch AoK* to prove that $\boldsymbol{w}$ is computed correctly without using $\boldsymbol{M}$. If simulation fails, then the protocol aborts. Since the failure probability is negligible, we have $\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_2] \approx \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_1]$.

Game G_3: Game G_3 is the same as Game G_2 except the following. In the initialization phase, the sender randomly picks $\boldsymbol{\alpha} \leftarrow_\$ \mathbb{Z}_q^n$ and sets $c_i = g^{\alpha_i}$ as fail commitments. Since the distribution of $\boldsymbol{c}$ is unchanged, $\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_3] = \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_2]$.

Game G_4: The ideal-world experiment $\mathbf{Ideal}_{\mathbf{S}', \hat{\mathbf{R}}'}$ in which an ideal-world receiver $\hat{\mathbf{R}}'$ uses the real-world receiver $\hat{\mathbf{R}}$ as a black-box as follows.

1. $\hat{\mathbf{R}}'$ acts as the sender and plays Game G_3 with $\hat{\mathbf{R}}$.
2. In the ℓ-th transfer phase, $\hat{\mathbf{R}}'$ sends i_ℓ^* that is extracted in Game G_1 to $\mathcal{F}_{OT}^{n\times 1}$ and fetches $m_{i_\ell^*}$ from $\mathcal{F}_{OT}^{n\times 1}$. $\hat{\mathbf{R}}'$ prepares $\boldsymbol{M'}$ such that $m'_{i_\ell^*} = m_{i_\ell^*}$ and $\forall j \neq i_\ell^* : m'_j = 0$.
3. Compute $\boldsymbol{w}$ according to $\boldsymbol{M'}$ and complete the rest of the protocol as described in Game G_3.

To sum up, it is easy to see that

$$\mathsf{Adv}(\mathbf{D}) = \left|\Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_4] - \Pr[\mathbf{D}(X) = 1 : X \xleftarrow{\$} G_0]\right| \leq \epsilon(\lambda) \ ,$$

where $\epsilon(\cdot)$ is a negligible function.

□

Theorem 6. *The proposed $\mathsf{OT}_{k\times 1}^N$ scheme (as shown in Fig. 1) is fully simulatable secure under the DDH assumption.*

Proof. By Definition 1, the proposed $\mathsf{OT}_{k\times 1}^N$ framework is fully simulatable secure due to both Theorem. 4 and Theorem. 5.