The ISC Int'l Journal of Information Security
ISeCure

October 2025, Volume 17, Number 3 (pp. 1–14)
http://www.isecure-journal.org

Presented at the ISCISC'2025 in Tehran, Iran.

Efficient Pairing-Free Adaptable k-out-of-n Oblivious Transfer Protocols **

Keykhosro Khosravani [1,*] Taraneh Eghlidos [2] Mohammad Reza Aref [3]
[1] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[2] Electronics Research Institute, Sharif University of Technology, Tehran, Iran
[3] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO.

Keywords:
Oblivious Transfer (OT), adaptable Oblivious Transfer, privacy-preserving, secure multiparty computation, offline precomputation

Type:

doi:

ABSTRACT

Oblivious Transfer (OT) is one of the fundamental building blocks in cryptography that enables various privacy-preserving applications. Constructing efficient OT schemes has been an active research area. This paper presents three efficient two-round pairing-free k-out-of-n oblivious transfer protocols with standard security. Our constructions follow the minimal communication pattern: the receiver sends k messages to the sender, who responds with n+k messages, achieving the lowest data transmission among pairing-free k-out-of-n OT schemes. Furthermore, our protocols support adaptivity and enable the sender to encrypt the n messages offline, independent of the receiver's variables, offering significant performance advantages in one-sender-multiple-receiver scenarios. We provide security proofs under the Computational Diffie-Hellman (CDH) and RSA assumptions, without relying on the Random Oracle Model. Our protocols combine minimal communication rounds, adaptivity, offline encryption capability, and provable security, making them well-suited for privacy-preserving applications requiring efficient oblivious transfer.

© 2025 ISC. All rights reserved.

1 Introduction

Oblivious Transfer (OT) is a fundamental cryptographic primitive that enables secure two-party computation. In its simplest form, known as 1-out-of-2 OT, one party (the sender) holds two messages M_0 and M_1, while the other party (the receiver) holds a bit b. At the end of the protocol execution, the receiver learns M_b but cannot obtain any information on M_{1-b}, and gains no knowledge of the receiver's selected bit b. Despite its seeming simplicity, OT is a powerful building block for constructing secure multiparty computation (SMPC) protocols [1] and various other privacy-preserving applications, like private set intersection [2, 3], and location-based services [4]. Since its introduction by Rabin in 1981 [5], Oblivious Transfer (OT) has been widely researched[6, 7] and has become a vital tool in modern cryptography. It allows secure computations while maintaining the confidentiality of inputs and outputs.

The concept of k-out-of-n Oblivious Transfer (OT)

* Corresponding author.
**The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.
Email addresses: keykhosro_khosravani@ee.sharif.edu, teghlidos@sharif.edu, aref@sharif.edu
ISSN: 2008-2045 © 2025 ISC. All rights reserved.

emerged as a generalization of the fundamental 1-out-of-2 OT primitive introduced by Rabin in 1981 [5]. This generalization, first proposed by Brassard, Crépeau, and Robert [8]. Over the years, numerous efficient constructions and security models for k-out-of-n OT have been explored [9–13], driven by its potential applications in areas such as secure database querying [14], private information retrieval [15], privacy-preserving data mining[16] ,and data transmission[17]. In the pairing model, Lai et al. [18] achieved the lowest communication cost, where the receiver sends 3 group elements to the sender, who responds with n+1 group elements. In pairing-free schemes, the lowest communication cost involves the receiver sending k group elements to the sender and the sender sending n+k group elements to the receiver[19, 20]. While these constructions optimize communication complexity, designing efficient k-out-of-N OT protocols under stronger security assumptions remains an active research direction[21, 22].

Our contribution

In this paper, we present three pairing-free k-out-of-n oblivious transfer schemes with several useful features. The key contributions of this paper are as follows:

1. *Offline encryption.* Our protocols allow the sender to pre-encrypt messages independent of the receivers' inputs, improving the overall online execution performance.
2. *One-sender multiple-receiver support.* Because the encryption of messages is independent of the receiver's inputs, our schemes can be applied in scenarios where a single sender communicates with multiple receivers, enhancing their practicality in real-world settings.
3. *Adaptivity.* Receivers can choose their inputs in an adaptive manner, ensuring flexibility during the execution of the protocol.
4. *Pairing-free design.* Unlike previous schemes such as [18, 23], which rely on costly pairing operations, our constructions avoid such expensive computations, making them practical for resource-constrained environments.
5. *Efficiency.* The proposed schemes achieve efficient communication and computation costs, making them suitable for lightweight or resource-constrained devices such as IoT nodes.
6. *New Computational Problems.* We have proposed two new computational problems, one based on the Computational Diffie-Hellman (CDH) assumption and another based on the RSA assumption, and proved their hardness relative to standard assumptions through polynomial-time reductions.
7. *Provable security.* We provide formal security proofs under the standard model, guaranteeing strong security assurances without relying on the Random Oracle Model.

Our proposed pairing-free k-out-of-n oblivious transfer schemes offer several advantages over previous works. In contrast to the schemes [18, 23], which rely on costly pairing operations, our constructions eliminate the need for such expensive computations, making them more suitable for resource-constrained environments. Furthermore, our protocols support offline precomputation, enabling the sender to encrypt messages independent of the receiver's inputs. This feature not only makes our schemes usable in one-sender-multiple-receiver scenarios, but also enhances the efficiency of online execution of the protocol. In Section 6, we provide a comprehensive comparison of our protocols with other k-out-of-n oblivious transfer schemes that share similar features, such as adaptivity and offline precomputation capabilities [17, 19, 24–27].

We organize the rest of this paper as follows. In Section 2, we review some preliminary concepts, hard assumptions, and their respective proofs. Section 3 presents a discussion of related works. Section 4 describes the constructions of our three proposed pairing-free k-out-of-n oblivious transfer schemes. The security proofs for these schemes are provided in Section 5. In Section 6, we compare our proposed schemes with other existing pairing-free k-out-of-n oblivious transfer protocols. Finally, we provide a concluding summary and outline potential future research in Section 7.

2 preliminaries

This section introduces and defines core concepts that are essential throughout the paper: Oblivious Transfer, a variant of the Generalized Computational Diffie-Hellman (CDH) and RSA assumptions, along with essential lemmas employed in Section 5. We also define the notations used in the paper.

2.1 Notations

For an elliptic curve $\mathbf{E}$ defined over a finite field $\mathbb{F}_p$, let $N' = \#\mathbf{E}(\mathbb{F}_p)$ denote the number of points on $\mathbf{E}$. If G is a generator of this elliptic curve, then for any integer $a \in \mathbb{Z}_{N'}$, we define $[a]G$ as the point on the elliptic curve obtained by scalar multiplication of the generator G with a, i.e.,

$$[a]G = \overbrace{G + G + \cdots + G}^{a \text{ times}}.$$

Table 1 provides a summary of the notations and symbols that will be used in the subsequent sections of this paper.

Table 1. Table of Notations

Notation	Description
n	# of sender messages
k	# of messages the receiver wishes to obtain.
σ_i	Index of the i-th message that the receiver selects.
G	Generator of an additive group (elliptic curve).
g	Generator of a multiplicative group.
$\mathbf{E}$	Elliptic curve used in Scheme $A.2$.
N'	# of points on the elliptic curve $\mathbf{E}$.
p', q'	prime numbers described in scheme $A.1$.
p'', q''	prime numbers described in scheme $A.2$.
p, q	prime numbers described in schemes $B.1, B.2$.
(N, e, d)	RSA parameters used in Schemes $B.1$ and $B.2$.

2.2 Oblivious Transfer

Oblivious Transfer (OT) is a two-party cryptographic protocol, first introduced by Rabin [5] in 1981. It has evolved into three main varieties:

- 1-out-of-2 OT: In this variant, two parties are involved: a sender and a receiver. The sender possesses a pair of messages, m_0 and m_1, while the receiver has a private bit b. At the end of the protocol, the receiver learns m_b but nothing regarding m_{1-b}, and the sender gains no knowledge about the value of b.
- 1-out-of-n OT: This is a generalization of the 1-out-of-2 OT, where the sender possesses n messages $m_1, m_2, \ldots, m_n$, and the receiver holds an integer $r \in \{1, 2, \ldots, n\}$. Upon completion, the receiver learns m_r without obtaining any information about the other messages, while the sender remains unaware of the value of r.
- k-out-of-n OT: Further extending the concept, this variant allows the receiver to obtain k messages out of the n messages held by the sender. Specifically, the sender holds n messages $m_1, m_2, \ldots, m_n$, and the receiver possesses k integers $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$. Following the protocol, the receiver's output consists of $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ without obtaining any information about the remaining messages, while the sender remains oblivious to the values of $\sigma_1, \sigma_2, \ldots, \sigma_k$.

These variants of Oblivious Transfer enable secure two-party computation and serve as fundamental building blocks for various cryptographic protocols.

2.3 System Model

In Secure Multiparty Computation, security guarantees are typically analyzed under different adversarial models. The two most common models are the *semi-honest* model and the *malicious* model.

- Semi-honest model: In the semi-honest (also known as honest-but-curious) model, adversaries are assumed to follow the protocol specification correctly, but they try to learn additional information from the transcript of the execution. In other words, a semi-honest adversary does not deviate from the prescribed steps, but may attempt to infer private inputs of other parties based on its entire view of the protocol execution.
- Malicious model: In the malicious model, adversaries are allowed to arbitrarily deviate from the protocol specification. A malicious adversary may send malformed or inconsistent messages, abort prematurely, or collude with other corrupted parties in order to learn private information or disrupt the computation. Protocols secure in this model provide stronger guarantees, but are generally less efficient than those designed for the semi-honest setting.

It is well known that two-party protocols secure in the semi-honest model can, in principle, be compiled into protocols secure against malicious adversaries by requiring parties to provide proofs that their actions are consistent with the prescribed protocol execution[28]. However, this generic transformation typically decreases efficiency, motivating the design of dedicated protocols secure in the malicious setting.

Oblivious Transfer protocols assume an authenticated point-to-point communication channel between two parties. The channel ensures delivery and integrity of messages, while confidentiality is guaranteed by the cryptographic design of the protocol itself.

2.4 Computational Assumptions

This section outlines the core computational hardness assumptions that underpin the security proofs in this work: the Computational Diffie-Hellman assumption and the RSA assumption.

Computational Diffie-Hellman (CDH) assumption[29]: Consider a cyclic group $\mathbf{G}$ of prime order p with randomly chosen generator g. The Computational Diffie-Hellman assumption states that, given (g, g^a, g^b) for a, b randomly chosen from Z_p, it is computationally infeasible to compute $g^{ab} \mod p$.

Our construction is implemented in groups where the Computational Diffie-Hellman problem (CDH) is believed to be hard. For our work, we utilize a variant of the Generalized Multi-Variant Computational Diffie-Hellman problem, which we explain further in Section 5.

RSA assumption: Given N, e and $m^e \mod N$ such that N is the product of two large random prime numbers p, q of approximately equal size, and $gcd(e, (p-1)(q-1)) = 1$ finding m is computationally hard.

Lemma 1. *Let* $\mathbf{G}$ *be a finite cyclic group of order* k*, and* g *be a generator of* G*. For an integer* α *with* $\gcd(\alpha, k) = 1$*,* g^α *is also a generator of* $\mathbf{G}$*.*

Lemma 2. *In a finite cyclic group* $\mathbf{G}$ *of prime order* p*, every element other than the identity is a generator.*

Remark 1. To create a cyclic group of prime order q, one can choose a prime q and an arbitrary small integer r such that $p = rq + 1$ is also prime. Since $\varphi(p) = rq$, where φ is Euler's totient function, we can randomly select an element g from Z_p^* such that $g^r \not\equiv 1 \pmod p$ and $g^q \equiv 1 \pmod p$. This ensures that the order of g is q, which means g is a generator of a cyclic group $\mathbf{G}$ of prime order q.

3 Related work

The problem of constructing efficient k-out-of-n Oblivious Transfer (OT) protocols has been a significant focus in cryptography since the primitive's generalization from 1-out-of-2 OT [5, 8]. Research efforts have primarily aimed at optimizing key performance metrics: the number of communication rounds, computational complexity, communication bandwidth, and the strength of the security model. Existing schemes can be broadly categorized based on their underlying cryptographic assumptions, particularly those that rely on bilinear pairings and those that are pairing-free.

Pairing-based Schemes. A line of research has utilized bilinear pairings to construct OT protocols, focusing on minimizing communication overhead. This approach achieves the lowest known communication cost for k-out-of-n OT: The receiver sends only a constant number of group elements (e.g., 3 [18]) and the sender responds with $n+1$ elements. Similarly, Guo *et al.* [23] introduced a pairing-based subset membership encryption scheme applicable to OT. However, this communication efficiency comes at the cost of computationally expensive pairing operations, which are often prohibitive for resource-constrained environments. Furthermore, their protocol requires a Trusted Third Party (TTP) to generate the public parameters.

Pairing-free Schemes. The majority of efficient OT constructions avoid pairings, basing their security on assumptions like the Computational Diffie-Hellman (CDH), Decisional Diffie-Hellman (DDH), or RSA problems. A primary efficiency goal in this category is to minimize communication rounds, computation cost ,and the number of communicated group elements. Several protocols [17, 24, 25] operate in three rounds: The sender initiates the protocol by sending n elements, the receiver subsequently sends k elements, and the sender concludes it by responding with k elements. However, a more efficient communication pattern is achieved by state-of-the-art two-round protocols [19, 26, 27]. In this superior two-round structure, the receiver initiates the protocol by sending k elements, and the sender concludes it by responding with $n + k$ elements. This decrease in rounds minimizes latency and interaction, leading to a more practical and efficient protocol execution, especially in high-latency networks.

Several works have explored additional features beyond basic efficiency. The concept of *adaptivity*, where the receiver can choose their selection indices interactively during the protocol, has been addressed in various designs [19]. Another impactful feature is *offline precomputation* or *offline encryption*, where the sender can pre-process and encrypt their messages independent of the receiver's inputs. This not only improves online performance, but also naturally enables *one-sender-to-multiple-receivers* scenarios, a significant advantage for real-world deployment [19, 26].

A critical differentiator among schemes is their security foundation. Many efficient protocols rely on the Random Oracle Model (ROM) for their security proofs [17, 19, 26]. In contrast, practical proofs in the standard model are often considered more robust and desirable. Furthermore, some schemes [24, 25] achieve standard model security in three rounds, but this sacrifices both round efficiency and practical applicability by introducing an extra round of latency and requiring messages themselves to be exponentiated.

Our Stand. This paper contributes to the line of pairing-free, standard-model k-out-of-n OT protocols. Our three proposed schemes (A.1, A.2, B.1) are designed for high practicality, achieving a combination of efficiency and simplicity that is well-suited for resource-constrained environments. As summarized in Table 2, our protocols achieve the minimal communication pattern, among pairing-free protocols, of $n + 2k$ group elements in just two rounds, matching the most efficient prior pairing-free work [19]. Crucially, our Scheme B.1 achieves the lowest computational cost among its peers, a key advantage for energy-limited devices. Furthermore, unlike some other schemes [17, 19, 26, 27], our constructions avoid complex operations: they rely solely on efficient modular arithmetic and pseudorandom number generator. This focus on algorithmic simplicity combined with minimal rounds, minimal communication ,and computational overhead, makes our protocols particularly attractive for lightweight applications and IoT ecosystems.

Scheme A.1

Sender $(m_1, \ldots, m_n)$ **Receiver** $(\sigma_1, \ldots, \sigma_k)$

$q', p' = 2q' + 1$ are prime,

$\langle g \rangle$ is a subgroup of $Z^*_{p'}$ of order q'

$r, \alpha_1, \alpha_2, \ldots, \alpha_n \in_R Z^*_{q'}$

$SP = \{q', g, g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_n}\}$

$s_1, s_2, \ldots, s_k \in_R Z^*_{q'}$

$(g^{\alpha_{\sigma_1}})^{s_1}, \ldots, (g^{\alpha_{\sigma_k}})^{s_k}$ ←

for $i = 1, 2, \ldots, n$

$c_i \overset{p'}{\equiv} m_i \cdot g^{r\alpha_i}$

$c_1, \ldots, c_n, (g^{s_1\alpha_{\sigma_1}})^r, \ldots, (g^{s_k\alpha_{\sigma_k}})^r$ →

for $j = 1, 2, \ldots, k$

$(g^{rs_j\alpha_{\sigma_j}})^{-s_j^{-1}} \overset{p'}{\equiv} g^{-r\alpha_{\sigma_j}}$

$m_{\sigma_j} \overset{p'}{\equiv} c_{\sigma_j} \cdot g^{-r\alpha_{\sigma_j}}$

Figure 1. Scheme A.1: The construction of k-n OT based on Discrete logarithm in mult. group

Scheme A.2

Sender $(m_1, \ldots, m_n)$ **Receiver** $(\sigma_1, \ldots, \sigma_k)$

$p'', N' = \#\mathbf{E}(\mathbb{F}_{p''})$ are prime,

G is a generator of $\mathbf{E}(\mathbb{F}_{p''})$

$r, \alpha_1, \alpha_2, \ldots, \alpha_n \in_R Z^*_{N'}$

$SP = \{p'', G, N', [\alpha_1]G, \ldots, [\alpha_n]G\}$

$s_1, s_2, \ldots, s_k \in_R Z^*_{N'}$

$[s_1\alpha_{\sigma_1}]G, \ldots, [s_k\alpha_{\sigma_k}]G$ ←

for $i = 1, 2, \ldots, n$

$c_i = m_i + [r\alpha_i]G$

$c_1, \ldots, c_n, [rs_1\alpha_{\sigma_1}]G, \ldots, [rs_k\alpha_{\sigma_k}]G$ →

for $j = 1, 2, \ldots, k$

$[s_j^{-1}rs_j\alpha_{\sigma_j}]G = [r\alpha_{\sigma_j}]G$

$m_{\sigma_j} = c_{\sigma_j} - [r\alpha_{\sigma_j}]G$

Figure 2. Scheme A.2: The construction of k-n OT based on the Discrete logarithm in additive groups

4 k-out-of-n OT schemes

In this section, we describe three efficient k-out-of-n Oblivious Transfer protocols with standard security proofs. In all these schemes a sender possesses n messages, $m_1, m_2, \ldots, m_n$, and a receiver wishes to recover k messages $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ out of those n messages, where $\Omega = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ are the k indices chosen by the receiver.

4.1 Scheme A

In this section, we present two k-out-of-n OT constructions, both of which rely on the hardness of the Discrete Logarithm Problem (DLP) for their security guarantees. The first construction (Section 4.1.1) employs a multiplicative group, while the second construction (Section 4.1.2) utilizes an additive group.

4.1.1 Construction A.1

Let q' and $p' = 2q' + 1$ be fixed prime numbers, and let g be a generator of a cyclic multiplicative group $\mathbf{G}$ of order q', which is a subgroup of $Z^*_{p'}$. The sender generates the parameters p', q' and g and shares these values with the receiver. All arithmetic operations mentioned hereafter are performed modulo p'. Scheme A.1 is depicted in Figure 1.

The sender randomly selects n distinct integers $\alpha_1, \alpha_2, \ldots, \alpha_n \in_R Z^*_{q'}$, and then computes $g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_n}$ and publishes these values on a bulletin board, accessible to both parties. The system parameters generated by the sender are denoted as $SP = \{q', g, g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_n}\}$. The protocol is executed using the following steps:

1. The receiver selects k integers $s_1, s_2, \ldots, s_k$, randomly from $Z^*_{q'}$, then computes and sends the values $(g^{\alpha_{\sigma_1}})^{s_1}, \ldots, (g^{\alpha_{\sigma_k}})^{s_k}$ to the sender.
2. The sender selects an integer r, randomly from $Z^*_{q'}$. For each $i = 1, 2, \ldots, n$, the sender encrypts the message m_i as $c_i = m_i \cdot g^{r\alpha_i}$, then computes the values $(g^{s_1\alpha_{\sigma_1}})^r, (g^{s_2\alpha_{\sigma_2}})^r, \ldots, (g^{s_k\alpha_{\sigma_k}})^r$ and sends them to the receiver.
3. For each $j = 1, 2, \ldots, k$, the receiver first computes $(g^{rs_j\alpha_{\sigma_j}})^{s_j^{-1}} = g^{r\alpha_{\sigma_j}}$, where s_j^{-1} denotes the multiplicative inverse of s_j modulo q'. Then, using the Extended Euclidean algorithm, the receiver computes $g^{-r\alpha_{\sigma_j}}$. Finally, the receiver can recover m_{σ_j} as follows: $m_{\sigma_j} = c_{\sigma_j} \cdot g^{-r\alpha_{\sigma_j}}$.

4.1.2 Construction A.2

Let p'' be a fixed prime number, and G be a generator of a prime order elliptic curve $\mathbf{E}$ of order $N' = \#\mathbf{E}(\mathbb{F}_{p''})$ defined over finite field $\mathbb{F}_{p''}$. The sender generates the parameters $p'', \mathbf{E}$ and G and shares these values with the receiver. All arithmetic operations mentioned hereafter are performed in the elliptic curve group $\mathbf{E}(\mathbb{F}_{p''})$. Scheme A.2 is depicted in Figure 2. The sender has n messages, $m_1, m_2, \ldots,$ and m_n, which are points on the elliptic curve $\mathbf{E}(\mathbb{F}_{p''})$. The sender randomly selects n distinct integers $\alpha_1, \alpha_2, \ldots, \alpha_n \in_R Z^*_{N'}$ and computes $[\alpha_1]G, [\alpha_2]G, \ldots, [\alpha_n]G$ and publishes these values on a bulletin board accessible to both parties. The system parameters generated by the sender are de-

Scheme B.1

Sender $(m_1, \ldots, m_n)$ **Receiver** $(\sigma_1, \ldots, \sigma_k)$

p, q are primes,
$N = pq$, $\phi(N) = (p-1)(q-1)$
$e \in \mathbb{Z}^*_{\phi(N)}$, $ed \equiv 1 \mod \phi(N)$
$x \in \mathbb{Z}^*_N$
$SP = \{N, e, x^e \bmod N\}$
$\alpha_1, \ldots, \alpha_n$:$n$ consecutive primes
with predefined prime $\alpha_1 \geq 3$.

$s_1, s_2, \ldots, s_k \in_R Z^*_N$

$\alpha_{\sigma_1}(x^e)s_1^e, \ldots, \alpha_{\sigma_k}(x^e)s_k^e$ ←

for $i = 1, 2, \ldots, n$
$c_i = m_i \cdot x\alpha_i^d$

$c_1, \ldots, c_n, \alpha_{\sigma_1}^d x s_1, \ldots, \alpha_{\sigma_k}^d x s_k$ →

for $j = 1, 2, \ldots, k$
$\alpha_{\sigma_j}^d x s_j s_j^{-1} = \alpha_{\sigma_j}^d x$
$m_{\sigma_j} = c_{\sigma_j} \cdot (\alpha_{\sigma_j}^d x)^{-1}$

Figure 3. Scheme B.1: The construction of k-n OT based on RSA

Scheme B.2: Adaptive Version of Scheme B.1

Sender $(m_1, \ldots, m_n)$ **Receiver** ()

p, q are prime,
$N = pq$, $\phi(N) = (p-1)(q-1)$
$e \in \mathbb{Z}^*_{\phi(N)}$, $ed \equiv 1 \pmod{\phi(N)}$
$x \in \mathbb{Z}^*_N$, $SP = \{N, e, x^e \bmod N\}$
$\alpha_1, \ldots, \alpha_n$ are consecutive primes
with predefined prime $\alpha_1 \geq 3$.

for $i = 1, 2, \ldots, n$
$c_i = m_i \cdot x\alpha_i^d$

$c_1, \ldots, c_n$ →

for step $j = 1, 2, \ldots, k$
$s_j \in_R Z^*_N$
choose σ_j adaptively

$\alpha_{\sigma_j}(x^e)s_j^e$ ←

raise to power of d

$\alpha_{\sigma_j}^d x s_j$ →

$\alpha_{\sigma_j}^d x s_j s_j^{-1} = \alpha_{\sigma_j}^d x$
$m_{\sigma_j} = c_{\sigma_j} \cdot (\alpha_{\sigma_j}^d x)^{-1}$

Figure 4. Scheme B.2 : Adaptive OT construction based on Scheme B.1

noted as $SP = \{p'', G, N', [\alpha_1]G, [\alpha_2]G, \ldots, [\alpha_n]G\}$. The protocol is executed using the following steps:

1. The receiver randomly selects k integers $s_1, s_2, \ldots, s_k$ from $Z^*_{N'}$, then computes and sends the values $[s_1\alpha_{\sigma_1}]G, [s_2\alpha_{\sigma_2}]G, \ldots, [s_k\alpha_{\sigma_k}]G$ to the sender.
2. The sender randomly selects an integer $r \in_R Z^*_{N'}$. For each $i = 1, 2, \ldots, n$, the sender encrypts the message m_i as $c_i = m_i + [r\alpha_i]G$. Subsequently, the sender computes $[rs_1\alpha_{\sigma_1}]G, [rs_2\alpha_{\sigma_2}]G, \ldots, [rs_k\alpha_{\sigma_k}]G$ and sends these values to the receiver.
3. For each $j = 1, 2, \ldots, k$, the receiver first computes $[s_j^{-1}rs_j\alpha_{\sigma_j}]G = [r\alpha_{\sigma_j}]G$, where s_j^{-1} denotes the multiplicative inverse of s_j modulo N'. Then, the receiver can recover m_{σ_j} for each $j = 1, 2, \ldots, k$ as follows: $m_{\sigma_j} = c_{\sigma_j} - [r\alpha_{\sigma_j}]G$.

4.2 Scheme B.1

This Scheme is based on the RSA algorithm, which relies on the computational difficulty of factoring large composite numbers. Let a and b be two distinct, large prime numbers greater than n, carefully chosen by the sender in such a way that $p = 2a + 1$ and $q = 2b + 1$ are prime numbers as well. Let $N = pq$ and $\phi(N) = (p-1)(q-1) = 4ab$, and $\alpha_1, \alpha_2, ..., \alpha_n$ be a sequence of n consecutive prime numbers, with $\alpha_1 = 3$ or any predefined prime greater or equal than 3. This choice is used for two reasons: First, it is essential that each α_i is co-Prime to N to ensure correct computation within the multiplicative group $\mathbb{Z}^*_N$. Second, using consecutive primes allows both parties to independently generate the same set of primes algorithmically from the agreed-upon starting point α_1, eliminating the communication overhead that would be required to negotiate a random set of n primes. All arithmetic operations mentioned hereafter are performed modulo N. Scheme B.1 is depicted in Figure 3.

The sender randomly selects an integer e from the set $\mathbb{Z}^*_{\phi(N)}$, where its elements are all integers k satisfying $1 \leq k < \phi(N)$ and $gcd(k, \phi(N)) = 1$. Subsequently, the sender employs the Extended Euclidean algorithm to compute d, such that $ed \equiv 1 \pmod{\phi(N)}$. This ensures that d is the multiplicative inverse of e modulo $\phi(N)$. The sender also randomly selects an integer x from the set $\mathbb{Z}^*_N$, in such a way that $x^4 \neq 1 \pmod N$ and computes $x^e \bmod N$, then publishes the values e and x^e on a public bulletin board, while keeping the values p, q, d, and x secret. The protocol is executed using the following steps:

1. The receiver randomly selects k integers $s_1, s_2, \ldots, s_k \in_R Z^*_N$, then computes and sends values $\alpha_{\sigma_1}(x^e)s_1^e, \ldots, \alpha_{\sigma_k}(x^e)s_k^e$ to the sender.
2. For each $i = 1, 2, \ldots, n$, the sender encrypts

the message m_i as $c_i = m_i \cdot x\alpha_i^d$. Subsequently, the sender computes $(\alpha_{\sigma_1} \cdot x^e s_1^e)^d, \ldots, (\alpha_{\sigma_k} \cdot x^e s_k^e)^d = \alpha_{\sigma_1}^d x s_1, \ldots, \alpha_{\sigma_k}^d x s_k$ and sends these values to the receiver.

3. For each $j = 1, 2, \ldots, k$, the receiver first employs the Extended Euclidean algorithm to compute s_j^{-1}. then, the receiver computes $\alpha_{\sigma_j}^d x s_j s_j^{-1} = \alpha_{\sigma_j}^d x$. Finally, to decrypt c_{σ_j} for each $j = 1, 2, \ldots, k$, the receiver computes $m_{\sigma_j} = c_{\sigma_j} \cdot (\alpha_{\sigma_j}^d x)^{-1}$, where $(\alpha_{\sigma_j}^d x)^{-1}$ is the multiplicative inverse of $\alpha_{\sigma_j}^d x$ modulo N, which is also computed using the Extended Euclidean algorithm.

4.3 Characteristics and Optimizations

constructed schemes inherit the following features:

- *Adaptivity*: All of the proposed schemes are capable of being used as adaptive Oblivious Transfer protocols. Initially, the sender encrypts n messages and sends them to the receiver. To recover each message, the receiver follows the protocol for a single choice, sends the corresponding data to the sender, and the sender responds with one message based on the protocol. To illustrate, we present the adaptive version of Scheme B.1 in Figure 5. The same transformation can be applied to the other schemes to achieve adaptivity.
- *Precomputation*: Our schemes enable the sender to precompute the encryption of n messages offline, independent of the receiver's parameters and choices. The precomputation offers two key advantages: Improved efficiency by reducing computational overhead during protocol execution, and support multi-receiver scenarios by allowing precomputation and broadcast of encrypted messages to multiple receivers.

Moreover, the system can be further optimized by introducing a pseudorandom function $F : \{1, \ldots, n\} \to G$, where G represents the cyclic group utilized in our cryptosystem. With this function, the receiver no longer needs to publish n separate parameters. Instead, both the sender and the receiver can independently compute $F(i)$ for each $i \in \{1, 2, \ldots, n\}$. This modification significantly reduces the size of the public key and consequently enhances the overall efficiency of the cryptosystem. However, it is important to note that for IoT systems, this optimization presents a trade-off. While it reduces communication overhead, it necessitates additional hardware resources for implementing function F, which may be a considerable constraint in resource-limited IoT devices.

5 security proofs

To establish the security of our oblivious transfer schemes, we introduce two new computational problems and prove their hardness via formal reductions: one to the Computational Diffie-Hellman problem and another to the RSA problem.

Problem 1. Alternative Generalized Computational Diffie-Hellman problem (AGCDH): Let g be a randomly chosen generator of a cyclic group $\mathbf{G}$ of prime order p. Given $(g, g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_k}, g^{\alpha_{k+1}}, g^{r\alpha_1}, g^{r\alpha_2}, \ldots, g^{r\alpha_k})$ for $r, \alpha_1, \alpha_2, \ldots, \alpha_{k+1}$ randomly chosen from $\{0, 1, \ldots, p-1\}$, it is computationally infeasible to compute $g^{r\alpha_{k+1}}$.

Proof. To derive a contradiction, we begin by supposing the existence of a polynomial-time algorithm for the AGCDH, This would allow us to construct a polynomial-time solver for the CDH problem, demonstrating that $CDH \preceq AGCDH$.
We assume that there exists an efficient algorithm $\mathcal{A}_1$ that can solve the AGCDH. A CDH solver $S(g, g^r, g^x)$ can be constructed as follows:

1. S generates k random integers $\beta_1, \beta_2, \ldots, \beta_k$ from $\mathbb{Z}_p^*$.
2. S calls $\mathcal{A}_1$ as a subroutine with the input $(g, g^{\beta_1}, g^{\beta_2}, \ldots, g^{\beta_k}, g^x, (g^r)^{\beta_1}, (g^r)^{\beta_2}, \ldots, (g^r)^{\beta_k})$.
3. $\mathcal{A}_1$ returns g^{rx}
4. S outputs g^{rx} as the solution to the CDH problem instance (g, g^r, g^x).

It follows that if there exists an efficient algorithm $\mathcal{A}_1$ that can solve the AGCDH, then we can use it to construct an efficient solver S for the Computational Diffie-Hellman (CDH) problem. In other words, the CDH is reducible to the AGCDH problem. If the CDH problem is considered to be hard, then the AGCDH must also be considered at least as hard, since solving the AGCDH would allow us to efficiently solve the CDH problem as well. □

Problem 2. Generalized Blinded RSA (GBRSA): In an RSA algorithm with parameters (N, e, d), given non-identity random elements $x, \beta_1, \beta_2, \ldots, \beta_k, \beta_{k+1}$ from $\mathbb{Z}_N^*$, it is computationally hard to compute $x\beta_{k+1}^d$ given $(N, e, x^e, \beta_1, \beta_2, \ldots, \beta_k, \beta_{k+1}, x\beta_1^d, x\beta_2^d, \ldots, x\beta_k^d)$.

Proof. We show that if there exists a polynomial-time algorithm for the GBRSA, then we can use it to solve the RSA in polynomial-time, which means $RSA \preceq GBRSA$.
We assume that there exists an efficient algorithm $\mathcal{A}_2$ that can solve the GBRSA. RSA solver $S(e, N, y = m^e)$ can be constructed as follows:

1. S generates $k+1$ random integers $A_1, A_2, \ldots, A_k, x$ from $\mathbb{Z}_N^*$.

2. S calls $\mathcal{A}_2$ as a subroutine with the input $(N, e, x^e, A_1^e, A_2^e, \ldots, A_k^e, y, xA_1, xA_2, \ldots, xA_k)$.
3. $\mathcal{A}_2$ returns $xy^d \equiv x(m^e)^d \equiv xm \pmod N$
4. S outputs $x^{-1}xm \equiv m \pmod N$ as the solution to the RSA problem instance (e, N, m^e).

If there exists an efficient algorithm $\mathcal{A}_2$ that can solve the GBRSA, then we can construct an efficient algorithm S for the RSA problem. In other words, RSA is reducible to the GBRSA, which implies that the GBRSA must be considered at least as hard as the RSA problem. □

Equipped with the hardness of these new problems, we proceed to define the security model for oblivious transfer and present the security proofs of our constructions, which are founded upon these hardness assumptions.

In k-out-of-n OT schemes the sender possesses n messages, $m_1, m_2, \ldots, m_n$, and the receiver wishes to recover k of those messages, $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$, where $\sigma_1, \sigma_2, \ldots, \sigma_k$ are k indices chosen by the receiver.

So far, we have presented three schemes for semi-honest parties with the following security requirements:

- Receiver's Privacy: It is computationally infeasible for the sender to distinguish between $I = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ and any other arbitrary set $I' = \{\sigma'_1, \sigma'_2, \ldots, \sigma'_k\}$ of the same size[19].
- Sender's Security: The receiver cannot recover any message m_j for $j \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$

In the rest of this section, we are going to provide Security proofs for our schemes:

5.1 Security of Scheme A.1

Lemma 3. *n scheme A.1, the receiver's choices are unconditionally secure.*

Proof. Based on Lemma 2, since $\gcd(q', \alpha_i) = 1$ for each $i \in 1, 2, \ldots, n$, it implies that g^{α_i} is a generator of the group $\mathbf{G}$ of order q'. When the sender receives $E = (g^{\alpha_{\sigma_i}})^{s_i}$, this value can be potentially a mask for any element in the set $B = \{g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_n}\}$, because all elements in B are generators of $\mathbf{G}$. As g^{α_i} and g^{α_j} are generators of the group $\mathbf{G}$, for any two distinct indices $1 \leq i \neq j \leq n$, there exist integers s_i and s_j such that $g^{\alpha_i s_i} = g^{\alpha_j s_j}$. Consequently, the received value E can potentially mask any element of the set B, and the receiver's choice σ_i is hidden from the sender. Therefore, the receiver's choices are unconditionally secure, meaning that the sender has no information about the receiver's choice, even with unlimited computational power, as E can mask any element of the set B equally likely.

□

Lemma 4. *In scheme A.1, the sender's security is conditional, subject to AGCDH problem.*

Proof. Suppose $1 \leq j \leq n$ is not an element of the set $\Omega = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, but the receiver can recover m_j from executing the protocol, defined in Scheme A.1. If the receiver can recover m_j from the received ciphertext $c_j = m_j \cdot g^{r\alpha_j}$, one can then effectively recover $g^{r\alpha_j}$ by computing $c_j \cdot m_j^{-1} = g^{r\alpha_j}$. Since the receiver is semi-honest, it follows the exact execution of the protocol. Therefore, by the end of the protocol, it possesses the set $T = \{g^{\alpha_1}, \ldots, g^{\alpha_n}, \sigma_1, \ldots, \sigma_k, s_1, \ldots, s_k, g^{s_1\alpha_{\sigma_1}}, \ldots, g^{s_k\alpha_{\sigma_k}}, g^{rs_1\alpha_{\sigma_1}}, \ldots, g^{rs_k\alpha_{\sigma_k}}, c_1, \ldots, c_n\}$, which comprises public parameters, the receiver's choices, the receiver's secret values, and the transcript of the protocol. In a semi-honest setup, if the receiver can recover the extra data m_j, it means there exists a polynomial-time algorithm $\mathcal{R}_1$ that the receiver executes to recover m_j. However, we prove that there exists no probabilistic polynomial-time (PPT) algorithm $\mathcal{R}_1$ to recover the extra data m_j. Therefore, there exists no semi-honest receiver who can recover m_j.

Suppose there exists a PPT algorithm $\mathcal{R}_1$ that can recover m_j for $j \notin \Omega$. We construct an algorithm $\mathcal{A}_3$ that can solve the AGCDH problem using $\mathcal{R}_1$ as a subroutine. Given an AGCDH instance $(g, A_1, A_2, \ldots, A_k, x, A_1^r, A_2^r, \ldots, A_k^r)$ $\mathcal{A}_3$ proceeds as follows:

1. Computes the public parameters PP through this process: It submits $(A_1, A_2, \ldots, A_k, x)$ as $(g^{\alpha_{\sigma_1}}, g^{\alpha_{\sigma_2}}, \ldots, g^{\alpha_{\sigma_k}}, g^{\alpha_{\sigma_j}})$ to the bulletin board. For the remaining $n - k - 1$ values in PP, $\mathcal{A}_3$ selects random non-identity elements from the cyclic group $\mathbf{G}$.
2. randomly selects k integers $s_1, s_2, \ldots, s_k$ from $\mathbb{Z}_{q'}^*$, collectively referred to as the set S. It then computes $(g^{\alpha\sigma_1})^{s_1}, \ldots, (g^{\alpha_{\sigma_k}})^{s_k}$, and denotes them as the set A.
3. Utilizing the values $s_1, s_2, \ldots, s_k$ from set S, computes $(A_1^r)^{s_1}, (A_2^r)^{s_2}, \ldots, (A_k^r)^{s_k}$, collectively denoting them as set B. Furthermore, generates n random values to serve as the ciphertexts C.
4. Constructs the transcript $T' = \{PP, \Omega, S, A, B, C\}$ in the simulated world, which is indistinguishable from the real-world transcript T.
5. Executes $\mathcal{R}_1$ with input T' as a subroutine to obtain m_j.

6. Computes $c_j \cdot m_j^{-1} = x^r$ and outputs it as the solution to the AGCDH problem with input $(g, A_1, A_2, \ldots, A_k, x, A_1^r, A_2^r, \ldots, A_k^r)$.

If the receiver could efficiently recover the message m_j for $j \notin \Omega$ in polynomial-time, then the algorithm $\mathcal{A}_3$ could use Scheme A.1 to solve the AGCDH problem, which has been proven to be computationally hard. Therefore, it must be computationally infeasible for the receiver to recover m_j, implying that Scheme A.1 is computationally secure for the sender's security. □

5.2 Security of Scheme A.2

Lemma 5. *In Scheme A.2, the receiver's choices are unconditionally secure and the sender's security is conditionally secure.*

Proof. The proof of this theorem is similar to the proofs presented in Lemmas 3 and 4, with the difference that Scheme A.2 operates on elliptic curve points. Therefore, we omit the proofs. □

5.3 Security of Scheme B.1

Lemma 6. *In Scheme B.1, the receiver's choices are unconditionally secure.*

Proof. The sender receives $x^e \alpha_i s^e$ from the receiver and sends $x\alpha_i^d s$ back to it. Since the sender possesses the values x and x^e, it can compute $(\alpha_i s^e, \alpha_i^d s)$, which hides the receiver's choice α_i, because it can be generated by masking any arbitrary choice α_j with $s' \equiv \alpha_i^d \alpha_j^{-d} s \pmod N$, according to Scheme B.1. Therefore, by observing $(w, w^d) \coloneqq (\alpha_i s^e, \alpha_i^d s)$, one cannot obtain any information about the choice α_i, since there exists some t, which can be used as the masking factor for any arbitrary choice α_j, where

$$(w, w^d) = (\alpha_j t^e, \alpha_j^d t)$$

$$t = w^d \alpha_j^{-d} \bmod N$$

Therefore, $(\alpha_i s^e, \alpha_i^d s)$ can potentially mask any α_j, and the receiver's choice α_i is perfectly hidden from the sender. □

To establish the sender's security for Scheme B.1, we must first prove the following lemma:

Lemma 7. *The GBRSA problem remains computationally hard even when $\beta_1, \beta_2, \ldots, \beta_k, \beta_{k+1}$ are all prime numbers.*

We prove that if there exists a polynomial-time algorithm for the GBRSA problem in the prime setup, then we can use it to solve the RSA problem in polynomial-time, implying that RSA $\preceq$ GBRSA-prime (RSA reduces to GBRSA in the prime setup).

Suppose there exists an efficient algorithm $\mathcal{A}_4$ that can solve the GBRSA problem. We can construct an RSA solver $S(e, N, y = m^e)$ as follows:

1. S generates $k+1$ random integers $A_1, A_2, \ldots, A_k, x$ from $\mathbb{Z}_N^*$, such that for each $1 \leq i \leq k$, A_i^e is prime (refer the reader to Appendix A).
2. If y is not prime, S finds a random θ such that $\theta^e y$ becomes prime. If y is prime, S sets $\theta = 1$.
3. S calls $\mathcal{A}_4$ as a subroutine with the input $(N, e, x^e, A_1^e, A_2^e, \ldots, A_k^e, y\theta^e, xA_1, xA_2, \ldots, xA_k)$.
4. $\mathcal{A}_4$ returns $xy^d\theta^{ed} \equiv x(m^e)^d\theta \equiv xm\theta \pmod N$.
5. S outputs $x^{-1}xm\theta\theta^{-1} \equiv m \pmod N$ as the solution to the RSA problem instance (e, N, m^e).

If there exists an efficient algorithm $\mathcal{A}_4$ that can solve the GBRSA problem in the prime setup, then we can construct an efficient algorithm S for the RSA problem. In other words, RSA is reducible to the GBRSA problem in the prime setup, which implies that it is at least as hard as the RSA problem. Having proved the hardness of the GBRSA problem in the prime setup, we use it to prove the security of this lemma.

Lemma 8. *In Scheme B.1, the sender's security is conditional according to Lemma 7.*

Proof. Let us assume that there exists a polynomial-time algorithm $\mathcal{R}_2$ that allows the semi-honest receiver to recover the plaintext m_j for some $j \notin \Omega$, where $\Omega = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ is the set of indices chosen by the receiver. We show that the existence of such an algorithm $\mathcal{R}_2$ leads to a contradiction, as it can be used to solve the GBRSA problem in the prime setup, which is known to be at least as hard as the RSA problem. If the receiver can recover m_j from the received ciphertext $c_j = m_j \cdot x\alpha_j^d$, where α_j is the j-th prime starting from 3 (or any predefined prime number), one can then compute $m_j^{-1} \cdot c_j = x\alpha_j^d$, effectively recovering the value $x\alpha_j^d$. Since the receiver is semi-honest, it follows the exact execution of the protocol, and ultimately, it obtains the set $T = \{N, e, x^e, \sigma_1, \ldots, \sigma_k, s_1, \ldots, s_k, \alpha_{\sigma_1} x^e s_1^e, \ldots, \alpha_{\sigma_k} x^e s_k^e, \alpha_{\sigma_1}^d x s_1, \ldots, \alpha_{\sigma_k}^d x s_k, c_1, \ldots, c_n\}$, which comprises the public parameters, the receiver's choices, the receiver's secret values, and the transcript of the executed protocol. We construct an algorithm $\mathcal{A}_5$ that uses $\mathcal{R}_2$ as a subroutine to solve the GBRSA problem in the prime setup. Given a GBRSA prime instance $(N, e, x^e, \beta_1, \beta_2, \ldots, \beta_k, \beta_{k+1}, x\beta_1^d, x\beta_2^d, \ldots, x\beta_k^d)$ $\mathcal{A}_5$ proceeds as follows:

1. Constructs the public parameters PP as $\{N, e, x^e\}$.
2. Selects $S = \{s_1, s_2, \ldots, s_k\}$ containing k random integers from $\mathbb{Z}_N^*$, then computes

$\{\beta_1 x^e s_1^e, \ldots, \beta_k x^e s_k^e\}$ and denotes them as set A.

3. Computes $(x\beta_1^d)s_1, \ldots, (x\beta_k^d)s_k$ using the values $s_1, s_2, \ldots, s_k$ and denotes them as set B, and generates n random values as ciphertext set C.
4. Compute $\sigma_i = Pindex(\beta_i)$, where the function $Pindex(\beta_i)$ returns the position of β_i within the sorted (ascending) list of all β values, and then construct $\Omega = \{\sigma_1, \ldots, \sigma_k\}$
5. Constructs the transcript $T' = \{PP, \Omega, S, A, B, C\}$ in the simulated world, which is indistinguishable from the real-world transcript T.
6. Executes $\mathcal{R}_2$ as a subroutine, with inputs T' and $Pindex(\beta_{k+1})$, and obtains $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ with the extra plaintext m_j, where $j = Pindex(\beta_{k+1})$.
7. Computes $m_j^{-1} \cdot c_j = x\alpha_j^d = x\beta_{k+1}^d$ and outputs the solution to the GBRSA problem with input $(N, e, x^e, \beta_1, \beta_2, \ldots, \beta_k, \beta_{k+1}, x\beta_1^d, x\beta_2^d, \ldots, x\beta_k^d)$.

If the receiver could recover more than k chosen plaintexts, a solver $\mathcal{A}_5$ could use Scheme B.1 to find a solution for the GBRSA for the special case, where $\beta_1, \ldots, \beta_{k+1}$ are prime numbers, which was proved to be at least as hard as the RSA problem. Therefore, the receiver cannot recover additional data ,and the sender's security is conditional. □

6 Performance Analysis

6.1 Comparison

In this paper, we have proposed three efficient two-round k-out-of-n Oblivious Transfer protocols. In these protocols, the receiver first transfers k data to the sender, followed by the sender transferring n+k data to the receiver. This communication pattern achieves the lowest data transmission among the existing pairing-free k-out-of-n oblivious transfer schemes. An additional significant feature of the three proposed schemes is their support for adaptivity, enabling the receiver to retrieve one of the k selected data by sequentially executing the protocol to recover one data at a time (k=1). Furthermore, our constructions enable offline encryption of the n messages by the sender, independent of the receiver's choices and random variables, before executing the protocol. This property offers significant performance advantages in scenarios where the sender needs to prepare encrypted data for multiple receivers,which was first introduced in [10]. Table 2 provides a comprehensive comparison of our proposed schemes with other existing pairing-free k-out-of-n oblivious transfer protocols that support adaptivity, focusing on the computational complexity for the sender and the receiver during the protocol execution[17, 19, 24–27].

6.2 Performance Evaluation

We have implemented and evaluated our proposed schemes using Python, leveraging the gmpy2 library for efficient arbitrary-precision arithmetic and SageMath for advanced cryptographic operations. The simulations were conducted as follows:

- Scheme A.1, employing multiplicative group arithmetic, utilizes a 2048-bit modulus.
- Scheme A.2, based on elliptic curve cryptography, is implemented with a 224-bit curve, providing security comparable to Scheme A.1.
- Scheme B.1, based on RSA, also employed a 2048-bit modulus, ensuring equivalent security to Schemes A.1 and A.2.

This setup ensures consistent performance metrics across all three protocols, allowing for accurate comparison of their computational efficiency. The source code for our implementations, along with additional execution time comparisons, is publicly available at GitHub [1] . We have performed the simulations using a desktop system featuring an Intel Core i7-6500U processor operating at 2.5 GHz, supported by 8 GB of RAM.

In the remainder of this section, we analyze the impact of parameters n and k on the execution time of our protocols, both with and without precomputation. For clarity, we present representative results using Scheme A.1; however, similar trends are observed across all schemes, as documented in our supplementary GitHub repository.

Figure 5 illustrates the impact of increasing n on the execution time of 7-out-of-n Oblivious Transfer in Scheme A.1. The comparison between scenarios with and without precomputation reveals that the precomputation performance of the model remains constant regardless of n, indicating its independence from this parameter. Figure 6 demonstrates how increasing k affects the execution time of k-out-of-45 Oblivious Transfer in Scheme A.1, again comparing precomputation and non-precomputation scenarios.

7 Conclusions

In this paper, we have presented three efficient two-round pairing-free k-out-of-n oblivious transfer protocols with standard security in the semi-honest model. These protocols can also be used as adaptive oblivious transfer schemes. Our schemes offer comparable performance in terms of communication rounds, computational complexity for both parties, and the size of transmitted messages. Furthermore, they provide provable security under the well-studied Computa-

[1] https://github.com/keykhosro/k-n-Oblivious-Transfer.git

Table 2. Comparison of pairing-free k-out-of-n OT schemes

k-n OT	rounds	Sender Comp.	Receiver Comp.	Comm.	Security Proof	Adversary	Pros & Cons
Scheme A.1	2	$(k+n)M_E$ $+nM_M$	$2kM_E+k\text{Inv}$ $+kM_M$	$n+2k$	CDH-standard	Semi-honest	NA
Scheme A.2	2	$(k+n)A_M$ $+nA_A$	$2kA_M+k\text{inv}$ $+kA_A$	$n+2k$	CDH-standard	Semi-honest	NA
Scheme B.1	2	$(k+n)M_E$ $+nM_M$	$kM_E+k\text{Inv}$ $+2kM_M$	$n+2k$	RSA-standard	Semi-honest	NA
[25]	3	$(n+k)M_E$	$2kM_E$	$n+2k$	RSA-standard	Semi-honest	The message itself is exponentiated
[19]	2	$(k+n)M_E+$ $2nH+n\text{Enc}$	$2kM_E+k\text{Inv}$ $+k\text{dec}+2kH$	$n+2k$	CDH-ROM	Malicious	NA
[24]	3	$(n+k)M_E$ $+1\text{Inv}$	$2kM_E+k\text{Inv}$	$n+2k$	DDH-standard	Semi-honest	The message itself is exponentiated
[17]	3	$(2k+1)M_E+2kM_M$ $+2kH+k\text{XOR}$ $+k\text{Inv}$	$nH+(n+2)M_E$ $+nM_M+n\text{XOR}$	$n+2k+3$	CDH-ROM	Malicious	needs a third party for computing $(k+1)M_E$ operations
[26]	2	$1F+1H$ $+1\text{XOR}+2M_E$	$nF+nH$ $+n\text{XOR}+(n+1)M_E$	$n+3$	CDH-ROM	Semi-honest	only secure as 1-out-of-n OT
[27]	2	$nH+n\text{XOR}$ $(20n+20k+$ $20log(s_S))A_A$ $+(40n+40k+$ $40log(s_S))A_M$	$kH+k\text{XOR}$ $(36+40k+$ $20log(s_R))A_A$ $+(70+80k+$ $40log(s_R))A_M$	$n+20k$	GFP-standard	Semi-honest	consist of $n+2$ matrix as Public parameters

M_E: Exponentiation in multiplicative group, M_M: Multiplication in multiplicative group, A_A: Addition in additive group, A_M: Scalar multiplication in additive group, H: Hash function, Enc/Dec: Symmetric Encryption/Decryption, Inv: Inversion, F: Permutable Function, s_S, s_R: Private keys of Sender and Receiver, GFP: Group Factorization Problem

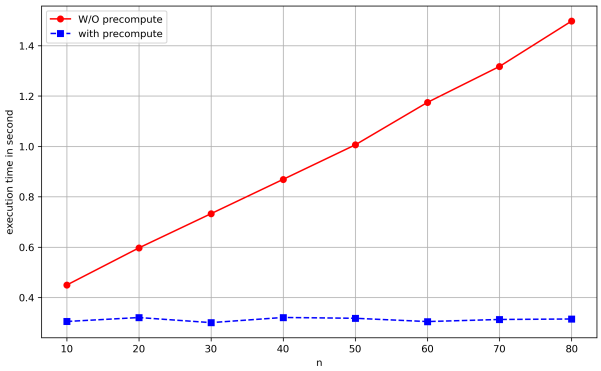


Figure 5. Comparing Scheme A.1 With and Without Precomputation, Impact of Increasing n on Execution Time for 7-out-of-n OT.

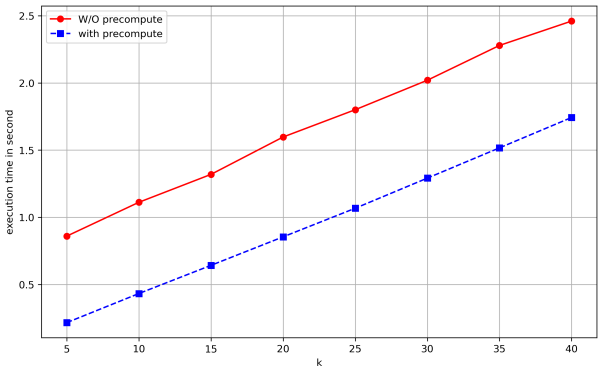


Figure 6. Comparing Scheme A.1 With and Without Precomputation, Impact of Increasing k on Execution Time for k-out-of-45 OT.

tional Diffie-Hellman (CDH) and RSA assumptions, without relying on the Random Oracle Model (ROM).

It is crucial to recognize that the emergence of quantum computers poses a significant threat to traditional cryptographic systems based on the hardness of integer factorization and discrete logarithms. These systems are no longer considered secure in the face of quantum computing capabilities. To address this challenge, NIST recommends the use of hybrid cryptography during the transition period from classical to post-quantum cryptography. Building on this recommendation, a promising direction for further research is the design of an Oblivious Transfer (OT) scheme that incorporates either hybrid or post-quantum encryption methods. Such a scheme should be optimized for implementation in Internet of Things (IoT) systems, addressing both the security concerns of the post-quantum era and the practical constraints of IoT devices.

References

[1] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation:

theory, practice and applications. *Information Sciences*, 476:357–372, 2019.

[2] Benny Pinkas, Thomas Schneider, and Michael Zohner. Scalable private set intersection based on ot extension. *ACM Transactions on Privacy and Security (TOPS)*, 21(2):1–35, 2018.

[3] Daniel Morales, Isaac Agudo, and Javier Lopez. Private set intersection: A systematic literature review. *Computer Science Review*, 49:100567, 2023.

[4] Bo Bi, Darong Huang, Bo Mi, Zhenping Deng, and Hongyang Pan. Efficient lbs security-preserving based on ntru oblivious transfer. *Wireless Personal Communications*, 108(4):2663–2674, 2019.

[5] Michael O Rabin. How to exchange secrets with oblivious transfer. *Cryptology ePrint Archive*, 2005.

[6] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Oblivious transfer with constant computational overhead. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 271–302. Springer, 2023.

[7] Vanessa Vitse. Simple oblivious transfer protocols compatible with supersingular isogenies. In *International Conference on Cryptology in Africa*, pages 56–78. Springer, 2019.

[8] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 234–238. Springer, 1986.

[9] Vijay Kumar Yadav, Nitish Andola, Shekhar Verma, and S Venkatesan. A survey of oblivious transfer protocol. *ACM Computing Surveys (CSUR)*, 54(10s):1–37, 2022.

[10] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *SODA*, volume 1, pages 448–457, 2001.

[11] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In *Annual International Cryptology Conference*, pages 573–590. Springer, 1999.

[12] Stanisław Jarecki and Xiaomin Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In *Theory of Cryptography Conference*, pages 577–594. Springer, 2009.

[13] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18(1):1–35, 2005.

[14] Jan Camenisch, Maria Dubovitskaya, Robert R Enderlein, and Gregory Neven. Oblivious transfer with hidden access control from attribute-based encryption. In *International Conference on Security and Cryptography for Networks*, pages 559–579. Springer, 2012.

[15] Yizhou Huang and Ian Goldberg. Outsourced private information retrieval. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, pages 119–130, 2013.

[16] Yi Li and Wei Xu. Privpy: General and scalable privacy-preserving data mining. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1299–1307, 2019.

[17] Huijie Yang, Jian Shen, Junqing Lu, Tianqi Zhou, Xueya Xia, and Sai Ji. A privacy-preserving data transmission scheme based on oblivious transfer and blockchain technology in the smart healthcare. *Security and Communication Networks*, 2021(1):5781354, 2021.

[18] Jianchang Lai, Yi Mu, Fuchun Guo, Rongmao Chen, and Sha Ma. Efficient k-out-of-n oblivious transfer scheme with the ideal communication cost. *Theoretical Computer Science*, 714:15–26, 2018.

[19] Cheng-Kang Chu, Wen-Guey Tzeng, et al. Efficient k-out-of-n oblivious transfer schemes. *J. Univers. Comput. Sci.*, 14(3):397–415, 2008.

[20] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *International Workshop on Public Key Cryptography*, pages 172–183. Springer, 2005.

[21] Bing Zeng, Christophe Tartary, Peng Xu, Jiandu Jing, and Xueming Tang. A practical framework for t-out-of-n oblivious transfer with security against covert adversaries. *IEEE Transactions on Information Forensics and Security*, 7(2):465–479, 2012.

[22] Ran Canetti, Pratik Sarkar, and Xiao Wang. Efficient and round-optimal oblivious transfer and commitment with adaptive security. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 277–308. Springer, 2020.

[23] Fuchun Guo, Yi Mu, and Willy Susilo. Subset membership encryption and its applications to oblivious transfer. *IEEE transactions on information forensics and security*, 9(7):1098–1107, 2014.

[24] Qian-Hong Wu, Jian-Hong Zhang, and Yu-Min Wang. Practical t-out-n oblivious transfer and its applications. In *International Conference on Information and Communications Security*, pages 226–237. Springer, 2003.

[25] Jen-Chieh Hsu, Raylin Tso, Yu-Chi Chen, and Mu-En Wu. Oblivious transfer protocols based

on commutative encryption. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2018.

[26] Xiaopeng Zhu, Yong Wu, Xiaodong Li, and Jianyi Zhang. A new ciphertext based ot protocol in cloud computing. In *2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 408–413. IEEE, 2023.

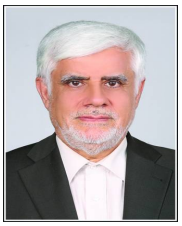
[27] Xianmin Wang, Xiaohui Kuang, Jin Li, Jing Li, Xiaofeng Chen, and Zheli Liu. Oblivious transfer for privacy-preserving in vanet's feature matching. *IEEE transactions on intelligent transportation systems*, 22(7):4359–4366, 2020.

[28] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*, volume 2. Cambridge university press, 2001.

[29] Whitfield Diffie and Martin E Hellman. New directions in cryptography. In *Democratizing cryptography: the work of Whitfield Diffie and Martin Hellman*, pages 365–390. 2022.

Keykhosro Khosravani received the B.Sc. degree in Electrical Engineering and the M.Sc. degree in Cryptography and Secure Communication from Sharif University of Technology (SUT), Tehran, Iran, in 2021 and 2025, respectively. His research interests include PQC, blockchain, federated learning and Data security and Privacy.

Taraneh Eghlidos received the B.Sc. degree in Mathematics from the University of Shahid Beheshti, Tehran, Iran, in 1986, and the M.Sc. degree in Industrial Mathematics from the University of Kaiserslautern, Germany, in 1991 and the Ph.D. degree in Mathematics from the University of Giessen, Germany, in 2000. She joined Sharif University of Technology (SUT) in 2002 as the faculty member, and is currently an Associate Professor with the Electronics Research Institute at SUT. Her research interests include interdisciplinary research areas, such as symmetric and asymmetric cryptography, applications of coding theory in cryptography, and mathematical modeling for solving real world problems. Her current fields of research include Lattice-based and Code-based Cryptography.



Mohammad reza Aref received his B.Sc. in 1975 from University of Tehran, his M.Sc. and Ph.D. in 1976 and 1980, respectively, from Stanford University, all in Electrical Engineering. He was a faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of Electrical Engineering at Sharif University of Technology since 1995. His current research interests include communication theory, information theory, and cryptography.

Appendix: Related proofs of Lemma 7

In this section, we prove that Step 1 of Lemma 7 runs in polynomial time.

Claim A.1. Step 1 of Lemma 7 is executed in polynomial-time.

Proof. Let N be an RSA modulus (so $N = pq$ with distinct primes p, q), let

$$\varphi(N) = (p-1)(q-1),$$

and let integers e, d satisfy $\gcd(e, \varphi(N)) = 1$ and $ed \equiv 1 \pmod{\varphi(N)}$. Denote by $n' := \lceil \log_2 N \rceil$ the bit-length of N. Complexity statements are measured as functions of n.

a) algorithm

The randomized algorithm is as follows:

1. Choose A uniformly at random from $\{1, \ldots, N-1\}$. If $\gcd(A, N) \neq 1$ then reject and repeat Step 1 (this step ensures $A \in \mathbb{Z}_N^*$).
2. Compute $y \leftarrow A^e \bmod N$.
3. Test whether y is prime. If yes, output A; otherwise, repeat from Step 1.

b) Expected running time.

We prove the algorithm succeeds in expected polynomial-time by three facts:

1. Exponentiation by e is a permutation of $\mathbb{Z}_N^*$.
2. A uniformly random element of $\mathbb{Z}_N^*$ is prime with probability $\Omega(1/\log N)$.
3. Each trial (one iteration) takes polynomial-time in terms of n.

c) The map $x \mapsto x^e \bmod N$ is a bijection on $\mathbb{Z}_N^*$.

Since $\gcd(e, \varphi(N)) = 1$, there exists an integer d with $ed \equiv 1 \pmod{\varphi(N)}$. For any $x \in \mathbb{Z}_N^*$, we have

$$(x^e)^d = x^{ed} = x^{1+k\varphi(N)} = x \cdot (x^{\varphi(N)})^k \equiv x \pmod{N},$$

where $ed = 1 + k\varphi(N)$ for some integer k, and we use Euler's theorem $x^{\varphi(N)} \equiv 1 \pmod{N}$. This shows that the inverse map is $y \mapsto y^d \bmod N$. Therefore, the map $f : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ defined by $f(x) = x^e \bmod N$ is a permutation. In particular, if A is uniform over $\mathbb{Z}_N^*$, then $y = A^e \bmod N$ is uniform over $\mathbb{Z}_N^*$.

d) Probability that a random element of $\mathbb{Z}_N^*$ is prime.

Let $\pi(x)$ denote the number of primes $\leq x$. The number of primes in $\{2, \ldots, N-1\}$ that are co-Prime to N is Asymptotically $\pi(N-1) - 2$ (p, q are primes but not co-Prime to N). The size of $\mathbb{Z}_N^*$ is $\varphi(N) \leq N$. Thus, for a random y drawn uniformly from $\mathbb{Z}_N^*$,

$$\Pr[y \text{ is prime}] \simeq \frac{\pi(N-1) - 2}{\varphi(N)} \geq \frac{\pi(N-1) - 2}{N}.$$

By the Prime Number Theorem, we have $\pi(N-1) = \Theta(\frac{N}{\log N})$. Hence, for sufficiently large N,

$$\Pr[y \text{ is prime}] = \Omega\left(\frac{1}{\log N}\right).$$

Consequently, the expected number of independent trials until success is

$$\mathbb{E}[\#\text{trials}] = O(\log N).$$

e) Time per trial is polynomial in n.

We show that each trial is performed using the following polynomial-time operations:

- Compute $\gcd(A, N)$ using the Euclidean algorithm, which runs in polynomial-time in terms of n'.
- Compute $A^e \bmod N$ using binary exponentiation (square-and-multiply). This requires $O(\log e)$ modular multiplications; since $e < \varphi(N) < N$, we have $\log e = O(n')$, and each multiplication/reduction is polynomial-time in n'.
- Test the primality of y using a polynomial-time primality test, which runs in polynomial-time.

Therefore, the time per trial is polynomial-time in n', denoted $\text{poly}(n')$.

f) Final computation

The expected running time of the algorithm is

$$\begin{aligned}\mathbb{E}[\text{total time}] &= \mathbb{E}[\#\text{trials}] \cdot (\text{time per trial}) \\ &= O(\log N) \cdot \text{poly}(n')\end{aligned}$$

Since $n' := \lceil \log_2 N \rceil$ and $n' \cdot \text{poly}(n')$ is still polynomial in terms of n', the expected total time is polynomial in terms of n'. Thus, the randomized procedure runs in expected polynomial-time. □