Efficient Pairing-Free Adaptable k-out-of-N Oblivious Transfer Protocols

Keykhosro Khosravani *Department of Electrical Engineering Sharif University of Technology* Tehran, Iran keykhosro khosravani@ee.sharif.edu

Taraneh Eghlidos *Electronics Research institute Sharif University of Technology* Tehran, Iran teghlidos@sharif.edu

Mohammad reza Aref *Department of Electrical Engineering Sharif University of Technology* Tehran, Iran aref@sharif.edu

Abstract—Oblivious Transfer (OT) is one of the fundamental building blocks in cryptography that enables various privacypreserving 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 also, 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. Furthermore, the first two proposed schemes require only one operation, making them ideal for resourceconstrained devices.

Index Terms—Oblivious Transfer (OT), adaptable Oblivious Transfer, privacy-preserving, secure multiparty computation,offline precomputation

I. 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 gains no information about M_{1-b} , and the sender remains oblivious to the receiver's choice bit b. Despite its seeming simplicity, OT is a powerful building block for constructing secure multiparty computation (SMPC) protocols [\[1\]](#page-6-0) and various other privacy-preserving applications, such as private set intersection [\[2\]](#page-7-0), [\[3\]](#page-7-1), and locationbased services [\[4\]](#page-7-2). Since its introduction by Rabin in 1981 [\[5\]](#page-7-3), Oblivious Transfer (OT) has been widely researched [\[6\]](#page-7-4), [\[7\]](#page-7-5) 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) emerged as a generalization of the fundamental 1-out-of-2 OT primitive introduced by Rabin in 1981 [\[5\]](#page-7-3). This generalization, first proposed by Brassard, Crépeau, and Robert $[8]$ $[8]$. Over the years, numerous efficient constructions and security models for k-out-of-N OT have been explored $[9]-[13]$ $[9]-[13]$ $[9]-[13]$, driven by its potential applications in areas such as secure database querying [\[14\]](#page-7-9), private information retrieval [\[15\]](#page-7-10), privacy-preserving data mining $[16]$ and data transmission $[17]$. In the pairing model, Lai et al. [\[18\]](#page-7-13) 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 [\[19\]](#page-7-14), [\[20\]](#page-7-15).While these constructions optimize communication complexity, designing efficient kout-of-N OT protocols under stronger security assumptions remains an active research direction. [\[21\]](#page-7-16), [\[22\]](#page-7-17)

In this paper, we construct three pairing-free k-out-of-N oblivious transfer schemes with the ability to use them as adaptive oblivious transfer protocols. Our schemes also support offline encryption of messages, enabling their application in scenarios involving one sender and multiple receivers. Our proposed schemes are accompanied by security proofs under the standard model, providing provable security guarantees 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-sendermultiple-receiver scenarios, but also enhances the efficiency of online execution of the protocol. In Section V, we provide a comprehensive comparison of our protocols with other kout-of-n oblivious transfer schemes that share similar features, such as adaptivity and offline precomputation capabilities [\[19\]](#page-7-14), [\[24\]](#page-7-19), [\[25\]](#page-7-20).

The organization of this paper is as follows: In Section II, we review some preliminary concepts, hard assumptions, and their respective proofs. Section III 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 IV. In Section V, we compare our proposed schemes with other existing pairing-free k-out-of-N oblivious transfer protocols. Finally, we conclude the paper in Section VI.

II. 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 4. We also define the notations used in the paper.

A. Notations

Let E be an elliptic curve defined over a finite field \mathbb{F}_p , and let $N = \#E(\mathbb{F}_p)$ denote the number of points on 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.,

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

B. Oblivious Transfer

Oblivious Transfer (OT) is a two-party cryptographic protocol, first introduced by Rabin [\[5\]](#page-7-3) 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 two messages, M_0 and M_1 , while the receiver holds a bit b. At the end of the protocol, the receiver learns M_b without gaining any information about M_{1-b} , and the sender remains oblivious to the value of b.
- 1-out-of-n OT: This is a generalization of the 1-out-of-2 OT, where the sender has 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 $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$. At the end of the protocol, the receiver learns $M_{\sigma_1}, M_{\sigma_2}, \ldots, M_{\sigma_k}$ 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.

C. Computational assumptions

Computational Diffie-Hellman (CDH) assumption [\[26\]](#page-7-21): Let g be a randomly chosen generator of a cyclic group G of prime order p. 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} .

Our constructions 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, defined as follows:

Alternative Generalized Computational Diffie-Hellman problem (AGCDH): Let g be a randomly chosen generator of a cyclic group G of prime order p . Given $(g, g^{\alpha_1}, g^{\alpha_2}, \dots, g^{\alpha_k}, g^{\alpha_{k+1}}, g^{r\alpha_1}, g^{r\alpha_2}, \dots, g^{r\alpha_k})$ 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: We show that if there exists a polynomial-time algorithm for the AGCDH, then we can use it to solve the CDH in polynomial time, which means $CDH \preceq AGCDH$. We assume that there exists an efficient algorithm 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 $\beta_1, \beta_2, \dots, \beta_k$ from \mathbb{Z}_p^* .
- 2. S calls 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. 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 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. $□$

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.

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 ∗ N , it is computationally hard to compute $x\beta_{k+1}^d$ given $(N, e, x^e, \beta_1, \beta_2, \dots, \beta_k, \beta_{k+1}, x\beta_1^d, x\beta_2^d, \dots, x\beta_k^d).$

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 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^1 calls 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. 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 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. $□$

Lemma 1: Let 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 G .

Lemma 2: If G is a finite cyclic group of prime order p , then every non-identity element of G is a generator.

Remark: To creat 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 G of prime order q .

III. 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 $\Omega = {\sigma_1, \sigma_2, \dots, \sigma_k}$ are the k indices chosen by the receiver. The source codes for our implementations are available at g ithub^{[1](#page-2-0)}.

A. Scheme 1

Let q and $p = 2q + 1$ be fixed prime numbers, and let g be a generator of a cyclic multiplicative group 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 1 is depicted in Figure [1.](#page-2-1)

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 $SP = \{q, g, g^{\alpha_1}, g^{\alpha_2}, \dots, g^{\alpha_n}\}.$ 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_i modulo q. Then, using the Extended Euclidean algorithm, the receiver computes

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

scheme 1
\n**Sender**
$$
(m_1, ..., m_n)
$$
 Receiveder $(\sigma_1, ..., \sigma_k)$
\n $q, p = 2q + 1$ are prime,
\n $\langle g \rangle$ is a subgroup of Z_p^* of order q
\n $r, \alpha_1, \alpha_2, ..., \alpha_n \in_R Z_q^*$
\n
$$
SP = \{q, g, g^{\alpha_1}, g^{\alpha_2}, ..., g^{\alpha_n}\}
$$
\n
$$
s_1, s_2, ..., s_k \in_R Z_q^*
$$
\nfor $i = 1, 2, ..., n$
\n
$$
c_i \stackrel{d}{=} m_i \cdot g^{r\alpha_i}
$$
\n
$$
c_1, ..., c_n, (g^{s_1\alpha_{\sigma_1}})^r, ..., (g^{s_k\alpha_{\sigma_k}})^r
$$
\nfor $j = 1, 2, ..., k$
\n
$$
(g^{rs_j\alpha_{\sigma_j}})^{-s_j-1} \stackrel{d}{=} g^{-r\alpha_{\sigma_j}}
$$
\n
$$
m_{\sigma_j} \stackrel{p}{=} c_{\sigma_j} \cdot g^{-r\alpha_{\sigma_j}}
$$

Fig. 1: Scheme 1: The construction of k-N OT based on Discrete logarithm in mult. group

 $g^{-r\alpha_{\sigma_j}}$. Finally, the receiver can recover m_{σ_j} as follows: $m_{\sigma_j} = c_{\sigma_j} \cdot g^{-r \alpha_{\sigma_j}}.$

B. Scheme 2

Let p be a fixed prime number, and G be a generator of a prime order elliptic curve E of order $N = \#E(\mathbb{F}_p)$ defined over finite field \mathbb{F}_p . The sender generates the parameters p, N and G and shares these values with the receiver. All arithmetic operations mentioned hereafter are performed in the elliptic curve group $E(\mathbb{F}_p)$. Scheme 2 is depicted in Figure [2.](#page-3-0) The sender has n messages, m_1, m_2, \ldots , and m_n , which are points on elliptic curve $E(\mathbb{F}_p)$. 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 denoted 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 $[r s_1 \alpha_{\sigma_1}] G, [r s_2 \alpha_{\sigma_2}] G, \ldots, [r s_k \alpha_{\sigma_k}] G$ 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 $j = 1, 2, ..., k$ as follows: $m_{\sigma_j} = c_{\sigma_j} - [r \alpha_{\sigma_j}] G.$

Fig. 2: Scheme 2: The construction of k-N OT based on Discrete logarithm in additive group

C. Scheme 3

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 $\alpha_1 = 3$. All arithmetic operations mentioned hereafter are performed modulo N. Scheme 3 is depicted in Figure [3.](#page-3-1)

The sender randomly selects an integer e from the set $\mathbb{Z}_{\phi(N)}^*$, which is the set of positive integers less than $\phi(N)$ and relatively prime to $\phi(N)$. 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 $x^4 \neq 1$ (mod N) and computes x^e mod 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 $s_1, s_2, \ldots, s_k \in_R \mathbb{Z}_N^*$, 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

scheme 3	Receiver $(\sigma_1, \ldots, \sigma_k)$
p, q are prime,	$N = pq, \phi(N) = (p-1)(q-1)$
$e \in \mathbb{Z}_{\phi(N)}^*$, $ed \equiv 1 \pmod{\phi(N)}$	
$x \in \mathbb{Z}_N^*$, $SP = \{N, e, x^e \mod N\}$	$s_1, s_2, \ldots, s_k \in_R \mathbb{Z}_N^*$
$\alpha_{\sigma_1}(x^e) s_1^e, \ldots, \alpha_{\sigma_k}(x^e) s_k^e$	
$\text{for } i = 1, 2, \ldots, n$	$c_i = m_i \cdot x \alpha_i^d$
$c_1, \ldots, c_n, \alpha_{\sigma_1}^d x s_1, \ldots, \alpha_{\sigma_k}^d x s_k$	
$\text{for } j = 1, 2, \ldots, k$	
$\alpha_{\sigma_j}^d x s_j s_j^{-1} = \alpha_{\sigma_j}^d x$	
$m_{\sigma_j} = c_{\sigma_j} \cdot (\alpha_{\sigma_j}^d x)^{-1}$	

Fig. 3: Scheme 3: The construction of k-N OT based on RSA

multiplicative inverse of $\alpha_{\sigma_j}^d x$ modulo N, which is also computed using the Extended Euclidean algorithm.

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.
- *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 $F : \{1, \ldots, n\} \rightarrow G$, 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.

IV. SECURITY PROOFS

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 [\[19\]](#page-7-14).
- Sender's Security: The receiver cannot recover any message m_j for $j \notin {\sigma_1, \sigma_2, \ldots, \sigma_k}$

in this section we are going to provide security proofs for our schems:

A. Security of Scheme 1

Lemma 3: In scheme 1, 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 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}, \dots, g^{\alpha_n}\}\$, because all elements in B are generators of G. As g^{α_i} and g^{α_j} are generators of the group 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. \square

Lemma 4: In scheme 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_i from executing the protocol, defined in Scheme 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 $T = \{g^{\alpha_1}, \dots, g^{\alpha_n}, \sigma_1, \dots, \sigma_k, s_1, \dots, s_k, \}$ $g^{s_1\alpha_{\sigma_1}}, \ldots, g^{s_k\alpha_{\sigma_k}}, g^{rs_1\alpha_{\sigma_1}}, \ldots, g^{rs_k\alpha_{\sigma_k}}, c_1, \ldots, c_n\},$ 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_i , it means there exists a polynomial-time algorithm \mathcal{R}_1 that the receiver executes to recover m_i . 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_i .

Suppose there exists a PPT algorithm \mathcal{R}_1 that can recover m_i 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 $(g, A_1, A_2, \ldots, A_k, x, A_1^r, A_2^r, \ldots, A_k^r)$ 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, A_3 selects random non-identity elements from the cyclic group 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 realworld 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_i for $j \notin \mathcal{L}$ Ω in polynomial time, then the algorithm \mathcal{A}_3 could use Scheme 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 1 is computationally secure for the sender's security. \Box

B. Security of Scheme 2

Lemma 5: In Scheme 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 2 operates on elliptic curve points. Therefore, we omit the proofs.

C. Security of Scheme 3

Lemma 6: In Scheme 3, 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 $s' \equiv \alpha_i^{\overline{d}} \alpha_j^{-d} s$ (mod N), according to Scheme 3. Therefore, by observing $(w, w^d) := (\alpha_i s^e, \alpha_i^d s)$ 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_i , where

$$
(w, wd) = (\alpha_j te, \alpha_jd t)
$$

$$
t = wd \alpha_j-d \text{ mod } N
$$

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. \Box

To establish the sender's security for Scheme 3, 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 $RSA \leq GBRSA$ -prime (RSA reduces to GBRSA in the prime setup).

Suppose there exists an efficient algorithm 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 be prime.
- 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 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. A_4 returns $xy^d \theta^{ed} \equiv x(m^e)^d \theta \equiv x m \theta \pmod{N}$.
- 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 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 3, sender's security is conditional, according to lemma 7.

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_i from the received ciphertext $c_j = m_j \cdot x \alpha_j^d$, where α_j is j-th prime starting from 3, 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 $T = \{N, e, x^e, \sigma_1, \ldots, \sigma_k, s_1, \ldots, s_k, \alpha_{\sigma_1} x^e s_1^e, \ldots, s_k, s_k, \sigma_{\sigma_k} x^e s_1^e, \ldots, s_k, s_k, \sigma_{\sigma_k} x^e s_1^e, \ldots, s_k, s_k, \sigma_{\sigma_k} x^e s_1^e, \ldots, s_k, \sigma_{\sigma$ $\alpha_{\sigma_k} x^e s_k^e, \alpha_{\sigma_1}^d x s_1, \ldots, \alpha_{\sigma_k}^d x s_1, c_1, \ldots, c_n$, 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 A_5 that uses \mathcal{R}_2 as a subroutine to solve the GBRSA problem in the prime setup. Given a GBRSA prime instance $(N, e, x^e, \beta_1, \beta_2, \dots, \beta_k, \beta_{k+1}, x\beta_1^d, x\beta_2^d, \dots, x\beta_k^d)$ \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 $\{\beta_1 x^e s_1^e, \dots, \beta_k x^e s_k^e\}$ 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 σ_i = $Pindex(\beta_i)$, where The function $Pindex(x)$ returns t such that x is the $t - th$ prime number in the sequence of prime numbers starting from 3, 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} $c_j = x\alpha_j^d = x\beta_{k+1}^d$ and outputs the solution to the GBRSA problem with input $(N, e, x^e, \beta_1, \beta_2, \dots, \beta_k, \beta_{k+1}, x\beta_1^d, x\beta_2^d, \dots, x\beta_k^d).$

If the receiver could recover more than k chosen plaintexts, a solver A_5 could use Scheme 3 to find a solution for the GBRSA for 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. \Box

V. PERFORMANCE ANALYSIS

A. Comparison

In this paper we have proposed three efficient 2-round k-outof-N Oblivious Transfer schemes, in which k data transferred from the receiver to the sender, followed by n+k data from the sender to the receiver. This communication pattern achieves the lowest data transmission among the existing pairing-free kout-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 [\[10\]](#page-7-22). Table [I](#page-7-23) 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 $[19]$, $[24]$, $[25]$.

B. 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 1, employing multiplicative group arithmetic, utilizes a 2048-bit modulus.

- Scheme 2, based on elliptic curve cryptography, is implemented with a 224-bit curve, providing security comparable to Scheme 1.
- Scheme 3, based on RSA, also employed a 2048-bit modulus, ensuring equivalent security to Schemes 1 and 2.

This setup ensures consistent performance metrics across all three schemes, allowing for accurate comparison of their computational efficiency. All simulations were executed on a desktop computer equipped with an Intel Core i7-6500U multi-core processor running at 2.5 GHz and 8 GB of RAM. Table $\,$ [II](#page-7-24) presents a comparative analysis of average execution times for 1,000 7-out-of-45 Oblivious Transfer operations across our proposed schemes, both with and without precomputation, where $k = 7$ and $n = 45$ are chosen arbitrarily. Figure [4](#page-6-1) illustrates the impact of increasing n on the execution time of 7-out-of-n Oblivious Transfer in Scheme 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 [5](#page-6-2) demonstrates how increasing k affects the execution time of k-out-of-45 Oblivious Transfer in Scheme 1, again comparing precomputation and nonprecomputation scenarios. Figure [6](#page-6-3) showcases the effect of increasing k on the execution time of k-out-of-45 Oblivious Transfer in our proposed scheme with precomputation.

Fig. 4: Impact of Increasing n on Execution Time for 7-out-of-n Oblivious Transfer: Comparing Scheme 1 With and Without Precomputation

VI. 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 compairable performance in terms of communication rounds, computational complexity for both parties, and the size of transmitted messages. Furthermore, they provide

Fig. 5: Impact of Increasing k on Execution Time for k-out-of-45 Oblivious Transfer: Comparing Scheme 1 With and Without Precomputation

Fig. 6: Impact of Increasing k on Execution Time for k-out-of-45 Oblivious Transfer: Comparing proposed schemes with precomputation

provable security under the well-studied Computational 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 postquantum era and the practical constraints of IoT devices.

REFERENCES

[1] Zhao, Chuan, 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).

 M_E : Exponentiation in mult. group, M_M : Multiplication in mult. group, A_A : Addition in add. group, A_M : Scaler product in add. group

TABLE II: Comparision of execution times of our proposed schemes

proposed scheme	without precomputation	with precomputation
scheme 1	0.9529	0.3096
scheme 2	1.2309	0.4092
scheme 3	0.7922	0.1018

Average Execution Time in Seconds for 1,000 Iterations of 7-out-of-45

Oblivious Transfer

- [2] Pinkas, Benny, Thomas Schneider, and Michael Zohner. "Scalable private set intersection based on OT extension." ACM Transactions on Privacy and Security (TOPS) 21, no. 2: 1-35 (2018).
- [3] Morales, Daniel, Isaac Agudo, and Javier Lopez. "Private set intersection: A systematic literature review." Computer Science Review 49 : 100567 (2023).
- [4] Bi, Bo, Darong Huang, Bo Mi, Zhenping Deng, and Hongyang Pan. "Efficient LBS security-preserving based on NTRU oblivious transfer." Wireless Personal Communications 108, no. 4 : 2663-2674 (2019).
- [5] Rabin, M. O. How to exchange secrets by oblivious transfer. Aiken Computation Laboratory, Harvard University, Cambridge, MA. USA, Tech. Rep. TR-81 [Online]. Available: available at http://eprint. iacr. org/2005/187, 1981.
- [6] Boyle, Elette, 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, pp. 271-302. Cham: Springer Nature Switzerland, 2023.
- [7] Vitse, Vanessa. "Simple oblivious transfer protocols compatible with supersingular isogenies." In Progress in Cryptology–AFRICACRYPT 2019: 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9–11, 2019, Proceedings 11, pp. 56-78. Springer International Publishing, 2019.
- [8] Brassard, Gilles, Claude Crépeau, and Jean-Marc Robert. "All-ornothing disclosure of secrets." In Conference on the Theory and Application of Cryptographic Techniques, pp. 234-238. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986.
- [9] Yadav, Vijay Kumar, Nitish Andola, Shekhar Verma, and S. Venkatesan. "A survey of oblivious transfer protocol." ACM Computing Surveys (CSUR) 54, no. 10s : 1-37 (2022).
- [10] Naor, Moni, and Benny Pinkas. "Efficient oblivious transfer protocols." In SODA, vol. 1, pp. 448-457. 2001.
- [11] Naor, Moni, and Benny Pinkas. "Oblivious transfer with adaptive queries." In Annual International Cryptology Conference, pp. 573-590. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.
- [12] Jarecki, Stanisław, and Xiaomin Liu. "Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection." In Theory of Cryptography: 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings 6, pp. 577-594. Springer Berlin Heidelberg, 2009.
- [13] Naor, Moni, and Benny Pinkas. "Computationally secure oblivious transfer." Journal of Cryptology 18: 1-35 (2005).
- [14] Camenisch, Jan, Maria Dubovitskaya, Robert R. Enderlein, and Gregory Neven. "Oblivious transfer with hidden access control from attributebased encryption." In Security and Cryptography for Networks: 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings 8, pp. 559-579. Springer Berlin Heidelberg, 2012.
- [15] Huang, Yizhou, and Ian Goldberg. "Outsourced private information retrieval." In Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, pp. 119-130. 2013.
- [16] Li, Yi, 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, pp. 1299-1307. 2019.

- [17] Yang, Huijie, 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 : 1-12 (2021).
- [18] Lai, Jianchang, 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] Chu, Cheng-Kang, and Wen-Guey Tzeng. "Efficient k-out-of-n Oblivious Transfer Schemes." J. Univers. Comput. Sci. 14, no. 3 : 397-415 (2008).
- [20] Chu, Cheng-Kang, and Wen-Guey Tzeng. "Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries." In Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005. Proceedings 8, pp. 172-183. Springer Berlin Heidelberg, 2005.
- [21] Zeng, Bing, 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, no. 2 : 465-479 (2012).
- [22] Canetti, Ran, Pratik Sarkar, and Xiao Wang. "Efficient and roundoptimal oblivious transfer and commitment with adaptive security." In International Conference on the Theory and Application of Cryptology and Information Security, pp. 277-308. Cham: Springer International Publishing, 2020.
- [23] Guo, Fuchun, Yi Mu, and Willy Susilo. "Subset membership encryption and its applications to oblivious transfer." IEEE transactions on information forensics and security 9, no. 7 : 1098-1107 (2014).
- [24] Wu, Qian-Hong, Jian-Hong Zhang, and Yu-Min Wang. "Practical t-outn oblivious transfer and its applications." In International Conference on Information and Communications Security, pp. 226-237. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.
- [25] Hsu, Jen-Chieh, 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), pp. 1-5. IEEE, 2018.
- [26] Diffie, Whitfield, and Martin E. Hellman. "New Directions in Cryptography." IEEE Transactions on Information Theory 22, no. 6: 644-54 (1976).