Generic Fully Simulatable Adaptive Oblivious Transfer*

Kaoru Kurosawa[1], Ryo Nojima[2], and Le Trieu Phong[2]

[1] Ibaraki University, Ibaraki, Japan, kurosawa@mx.ibaraki.ac.jp
[2] National Institute of Information and Communications Technology (NICT), Tokyo, Japan
{ryo-no,phong}@nict.go.jp

Abstract. We aim at constructing adaptive oblivious transfer protocols, enjoying fully simulatable security, from various well-known assumptions such as DDH, d-Linear, QR, and DCR. To this end, we present two generic constructions of adaptive OT, one of which utilizes verifiable shuffles together with threshold decryption schemes, while the other uses permutation networks together with what we call *loosely-homomorphic* key encapsulation schemes. The constructions follow a novel designing approach called "blind permutation", which completely differs from existing ones. We then show that specific choices of the building blocks lead to concrete adaptive OT protocols with fully simulatable security in the standard model under the targeted assumptions. Our generic methods can be extended to build universally composable (UC) secure, and leakage-resilient OT protocols.

Keywords: adaptive OT, fully-simulatable, verifiable shuffles, permutation networks, loose homomorphism, leakage resilience

1 Introduction

1.1 Background

Oblivious transfer (OT) with adaptive queries, or adaptive OT for short, was first examined by Naor and Pinkas in [27], in which there are a sender and a receiver. The sender holds n messages, and the receiver would like to retrieve k of them, one after the other, so that: (1) the sender does not know what the receiver obtains, and (2) the receiver gets nothing more beside the k messages. The key applications of this type of OT are in patent searches, oblivious search, medical databases etc.

The security notion capturing the above requirements has evolved in the literature. The notion of full simulatability was introduced by Camenisch, Neven, and Shelat in [3], following the real-world, ideal-world paradigm. In the ideal world, there exists a trusted third party (TTP), to which the sender gives all of his messages. When a receiver wants to obtain a message, he simply sends the corresponding index to the TTP. On the other hand, in the real world, there is no TTP at all, and the protocol of adaptive OT is run by the sender and the receiver. The intuition of full simulatability is that the real world is indistinguishable from the ideal world, with respect to any poly-time adversary.

There are a few approaches in building fully simulatable adaptive OT protocols. Let us have a look at them below.

The "assisted decryption" approach: Beginning with the work of Ogata and Kurosawa [30], at the core of this approach is a blind signature scheme. The blind property helps hiding the indexes of

* A preliminary version of this paper was presented at the 9th International Conference on Applied Cryptography and Network Security (ACNS '11) [23], improving and generating upon papers by the same authors at Asiacrypt '09 [21] and SCN '10 [22].

the receiver. The signatures assist the decryption at the receiver side, so that the desired message is obtained. The unforgeability of the blind signature scheme ensures that the receiver cannot obtain more signatures (and hence messages) than requested.

Ogata and Kurosawa [30] used the RSA blind signature scheme in the random oracle model, so their corresponding OT scheme was in that model. Also in ROM, Chu and Tzeng [4] presented an OT scheme using the Boldyreva's blind signature scheme [2].

Camenisch et al. [3] generalized and refined the schemes given in [4,30]. They furthermore gave a construction in the standard model, using $\mathfrak{q}$-based assumptions (in which $\mathfrak{q}$ depends on n) in pairing groups.

Efforts have been devoted to further extending the approach. In ROM, Green and Hohenberger [11] showed a protocol under the decisional bilinear Diffie-Hellman assumption. In [12], Green and Hohenberger constructed a universally-composable, so fully-simulatable, scheme under the $\mathfrak{q}$-hidden LRSW assumption in the standard model. The same authors in [13] gave a construction under the decision 3-party DDH (3DDH) assumption in pairing groups.

The "oblivious PRF" approach: Jarecki and Liu [19] joined the research line with a scheme based on the $\mathfrak{q}$-DHI assumption yet in RSA groups. The scheme is based on an oblivious pseudo-random function, in which the receiver with index σ can obtain, without revealing σ, the output $\mathrm{PRF}_K(\sigma)$ where PRF's key K is held by the sender.

The above approaches, up to now, yield adaptive OT schemes based on dynamic, or $\mathfrak{q}$-based assumptions in the standard model, with only an exception in [13] under the 3DDH assumption.

Our "blind permutation" approach: With respect to assumptions which are not $\mathfrak{q}$-based, we in [21] showed a simple scheme fully simulatable under the DDH assumption. However, the scheme suffered from a large communication cost of $O(n)$ in each transfer, as pointed out by Green and Hohenberger in [13]. Soon afterwards, we in [22], using a verifiable shuffle protocol, overcome the demerit in [21] by reducing the cost to $O(1)$, while still maintaining the DDH assumption for security. Specifically, the work [22] used the verifiable shuffle protocol of Neff [29] which is a 7-move honest verifier zero-knowledge protocol for proving the relation between $(g, X_1, \ldots, X_n)$ and $(g^c, X_{\pi(1)}^c, \ldots, X_{\pi(n)}^c)$, where π is a random permutation and c is random. Note that Neff's shuffle protocol is computationally zero-knowledge under the DDH assumption, so that it seems impossible to utilize the shuffle beyond the DDH case.

This paper continues and refines the blind permutation approach. This approach is unique since, so far, it is the only one effectively yielding adaptive OT schemes fully simulatable under standard, well-known assumptions.

1.2 Our contribution

We present two generic methods for constructing fully simulatable adaptive OT in the standard model. They yield numerous protocols from various assumptions, including the DDH, d-linear ($d \geq 2$), quadratic residuosity (QR), and decisional composite residuosity (DCR) assumptions. A comparison with previous works is given in Table 1, in which our DDH-based OT protocol has less number of moves in the initialization phase than that of [22]. Note that our schemes based on the QR and DCR assumptions induce a bit higher communication cost for initialization.

Our first method can be applied to any public-key encryption scheme E which satisfies two conditions: (1) It must be a homomorphic encryption scheme such that the message space is a group of prime public order; and (2) It can be used as a 2-out-of-2 threshold decryption scheme.

Table 1. Fully simulatable adaptive OT schemes without random oracles. The O also hides the message length in the QR case, which is assumed small compared to n.

Scheme	Assumption	Communication Cost (each transfer)	Initialization Cost
CNS [3]	$\mathfrak{q}$-strong DH and $\mathfrak{q}$-PDDH	$O(1)$	$O(n)$
GH [12]	$\mathfrak{q}$-hidden LRSW (UC secure)	$O(1)$	$O(n)$
JL [19]	$\mathfrak{q}$-DHI (RSA group)	$O(1)$	$O(n)$
KN [21]	DDH	$O(n)$	$O(n)$
GH [13]	decision 3-party DH (3DDH)	O(1)	$O(n)$
KNP [22]	DDH	$O(1)$	$O(n)$ (more moves)
This work	DDH	$O(1)$	$O(n)$ (less moves)
	d-Linear		$O(n)$
	DCR		$O(n \log n)$
	QR		$O(n \log n)$

The first condition allows us to use the verifiable shuffle protocol of Groth and Lu [14] which is a statistical zero-knowledge shuffle protocol for proving the relation between $\big(E(m_1), \ldots, E(m_n)\big)$ and $\big(E(m_{\pi(1)}), \ \ldots, \ E(m_{\pi(n)})\big)$, where π is a random permutation. However, we cannot obtain any adaptive OT even if we directly replace Neff's shuffle protocol by Groth-Lu's shuffle protocol into [22]. This is because the sender can compute π from $\big(E(m_{\pi(1)}), \ldots, E(m_{\pi(n)})\big)$. To overcome this problem, we use 2-out-of-2 threshold decryption. From this method, new adaptive OTs are obtained under the DDH assumption and the d-linear ($d \geq 2$) assumption, respectively.

Our second method can be applied to any key encapsulation mechanisms (KEM) satisfying what we call *loosely-homomorphic* property. We use permutation networks for this case while we do not use threshold decryption. From this method, new adaptive OTs are respectively obtained from the QR and DCR assumptions.

Theoretically, the generic constructions show that encryption, with some homomorphic property, implies adaptive OT.

Technically, we will later assume that the receiver never repeats its requests, which is not quite a strict restriction. For applications sensitive to this, by adding dummy messages as elaborated in [31], one can easily overcome the restriction.

Our generic methods enjoy further extensions. In Sect.5.1, we show how to obtain UC-secure adaptive OT protocols, with a little loss in efficiency. Specifically, we use a transformation of Σ-protocols to UC-secure ones [17, 26], with the help of a recent UC-secure commitment scheme [25] by Lindell. The communication cost becomes $O(L)$ for soundness error 2^{-L}.

Furthermore, in Sect.5.2, we show an adaptive (and hence 1-out-of-2) OT protocol which is resilient to the randomness leakage of the sender's first step. As an independent work, Damgard, Hazay and Patra [9] recently considered a framework for leakage resilient two party protocols. However, they were unable to construct such a 1-out-of-2 OT (see Sect.6 of [9]).

1.3 Intuition behind our protocols

For the illustration, let us consider ElGamal encryption, over cyclic group G with generator g of prime order q. The secret key is $x = x_{\mathsf{S}} + x_{\mathsf{R}} \in Z_q$ in which the sender S holds x_{S}, and the receiver R

gets x_{R}. This will prevent full decryption of a ciphertext without the cooperation from either party. Let $h_{\mathsf{S}} = g^{x_{\mathsf{S}}}$, $h_{\mathsf{R}} = g^{x_{\mathsf{R}}}$, and $h = h_{\mathsf{S}} h_{\mathsf{R}}$ be public. The sender with private inputs $(x_{\mathsf{S}}, M_1, \ldots, M_n)$, and the receiver R with private inputs (σ, x_{R}) acts as follows for R to obtain M_σ.

1. S sends ElGamal ciphertexts $(A_i, B_i) = (g^{r_i}, M_i h^{r_i})$ to R for all $1 \leq i \leq n$.
2. R with index σ sends back $(C_1, C_2) = \mathbf{Rand}(A_\sigma, B_\sigma) = (g^{r_\sigma + r'}, M_\sigma h^{r_\sigma + r'})$ where $r' \stackrel{\$}{\leftarrow} Z_q$ is chosen by R.
3. S sends the partial decryption $\mu_{\mathsf{S}} = C_1^{x_{\mathsf{S}}}$ to R, who computes $\mu_{\mathsf{R}} = C_1^{x_{\mathsf{R}}}$ and $\mu = \mu_{\mathsf{S}} \mu_{\mathsf{R}}$ to obtain $M_\sigma = C_2 / \mu$.

If both parties are honest-but-curious, S cannot know σ due to the re-randomization at the second step. Also, R cannot obtain other messages due to the encryption at the first step.

To attain full simulability, we add zero-knowledge proofs to each step. For the first and third steps, the proofs are Schnorr-type ones, and are efficient. The second one is difficult, in which we need to ensure that the re-randomized ciphetext (C_1, C_2) is either

$$\mathbf{Rand}(A_1, B_1) \vee \cdots \vee \mathbf{Rand}(A_n, B_n)$$

which can be implemented by an OR zero-knowledge proof. Naive and direct implementation of this zero-knowledge proof requires $O(n)$ communication cost for *each* receiver's index. Quite surprisingly, in Sect.3, we are able to reduce the cost to $O(1)$ by utilizing shuffle protocols.

The above ideas work well with the DDH and its weaker variants. For the DCR case, it is hard to share the secret key between the parties. For the QR case, we are unable to find a proper shuffle protocol to work with since the message space is of order 2 (or 2^ℓ). We overcome these difficulties in Sect.4, employing permutation networks for shuffling. The communication cost for each receiver's index is still $O(1)$, but the initialization cost is slightly increased as shown in Table 1.

2 Preliminaries

2.1 Notations

Throughout the paper, $\mathrm{OT}^n_{k\times 1}$ denote the adaptive OT with n messages of the sender and k choices of the receiver. ZKPK stands for zero-knowledge proof of knowledge, while ZKPM for zero-knowledge proof of membership. WIPK means witness-indistinguishable proof of knowledge. Furthermore, $\mathsf{ZKPK}\{(x) : X = g^x\}$ means a ZKPK protocol showing the knowledge of secret x satisfying the equation; and similar notations for more complex ZKPK, ZKPM, WIPK protocols will be used.

Taking an element a randomly from a set A is denoted by $a \stackrel{\$}{\leftarrow} A$. We use $a[i]$ to indicate the i-th component of a. For example, when a is a bit string, $a[i]$ is the i-th bit; when a is a tuple of elements, $a[i]$ becomes the i-th element.

2.2 Fully-simulatable $\mathrm{OT}^n_{k\times 1}$

We use almost the same presentation as [21], and consider a weak model of universally composable (UC) framework as follows.

– At the beginning of the game, an adversary $\mathcal{A}$ can corrupt either a sender S or a receiver R, but not both of them.

– $\mathcal{A}$ can send a message, denoted by $\mathcal{A}_{\text{out}}$, to an environment $\mathcal{Z}$ after the end of the protocol. However, $\mathcal{A}$ cannot communicate with $\mathcal{Z}$ during the protocol execution. (This property makes the definitions weaker than standard UC security.)

The ideal functionality of $\text{OT}^n_{k\times 1}$ will be shown below. For a protocol $\Pi = (\mathsf{S}, \mathsf{R})$, define the advantage of $\mathcal{Z}$ as

$$\mathbf{Adv}(\mathcal{Z}) \stackrel{\text{def}}{=} \Big| \Pr(\mathcal{Z} = 1 \text{ in the real world}) - \Pr(\mathcal{Z} = 1 \text{ in the ideal world}) \Big|$$

where the real and ideal worlds are defined below.

The ideal world: there are a few parties consisting of the ideal functionality $\mathcal{F}_{\text{adapt}}$, an ideal world adversary $\mathcal{A}'$, and the environment $\mathcal{Z}$. Also we have dummy sender S' and receiver R'. The parties behave as follows.

Initialization phase

1. The environment $\mathcal{Z}$ sends $(M_1, \ldots, M_n)$ to the dummy sender S'.
2. S' sends $(M_1^*, \ldots, M_n^*)$ to $\mathcal{F}_{\text{adapt}}$, where $(M_1^*, \ldots, M_n^*) = (M_1, \ldots, M_n)$ if S' is not corrupted.

Transfer phase $i = 1, \ldots, k$

1. $\mathcal{Z}$ sends σ_i to the dummy receiver R', where $1 \le \sigma_i \le n$.
2. R' sends σ_i^* to $\mathcal{F}_{\text{adapt}}$, where $\sigma_i^* = \sigma_i$ if R' is not corrupted.
3. $\mathcal{F}_{\text{adapt}}$ sends `received` to $\mathcal{A}'$.
4. $\mathcal{A}'$ sends $b = 1$ or 0 to $\mathcal{F}_{\text{adapt}}$, where $b = 1$ if S' is not corrupted.
5. $\mathcal{F}_{\text{adapt}}$ sends E_i to R', where
$$E_i = \begin{cases} M^*_{\sigma_i^*} \text{ if } b = 1 \\ \perp \text{ if } b = 0 \end{cases}$$
6. R' sends E_i to $\mathcal{Z}$.

After the end of the protocol, $\mathcal{A}'$ sends a message $\mathcal{A}'_{\text{out}}$ to $\mathcal{Z}$. Finally $\mathcal{Z}$ outputs 1 or 0.

The real world: Simply in this world, the protocol $\Pi = (\mathsf{S}, \mathsf{R})$ is executed as specified by its construction (thus without $\mathcal{F}_{\text{adapt}}$). The environment $\mathcal{Z}$ and the real world adversary $\mathcal{A}$ behave in the same way as above.

Definition 1. *Protocol $\Pi = (\mathsf{S}, \mathsf{R})$ is secure against the sender (resp, receiver) corruption if for any real world adversary $\mathcal{A}$ who corrupts the sender S (resp, receiver R), there exists an ideal world adversary $\mathcal{A}'$ who corrupts the dummy sender S' (resp, dummy receiver R') such that for any poly-time environment $\mathcal{Z}$, the advantage $\mathbf{Adv}(\mathcal{Z})$ is negligible.*

Definition 2. *Protocol $\Pi = (\mathsf{S}, \mathsf{R})$ is a fully simulatable $OT^n_{k\times 1}$ if it is secure against the sender corruption and the receiver corruption.*

3 Generic adaptive OT from verifiable shuffles

3.1 Building blocks

Threshold PKE We need an 2-out-of-2 threshold PKE scheme TPKE, which consists of the following algorithms.

- TGen: Two parties S and R run a protocol so that they respectively obtain (pk, sk_{S}) and (pk, sk_{R}) where $pk = (pk_{\mathsf{S}}, pk_{\mathsf{R}})$ is the agreed public key and $sk_{\mathsf{S}}, sk_{\mathsf{R}}$ are the shares of secret key. (The public key is needed for all algorithms below, and we omit writing it for clarity.)
- $\mathsf{TEnc}(M; r)$: output a ciphertext C for a plaintext M and a random coin r.
- $\mathsf{TDec}(sk_{\mathsf{P}}, C)$: for $\mathsf{P} \in \{\mathsf{S}, \mathsf{R}\}$, output μ_{P} which is the decryption share of the ciphertext C under secret key sk_{P}.
- $\mathsf{TComb}(C, \mu_{\mathsf{S}}, \mu_{\mathsf{R}})$: output a plaintext M by combining the input $C, \mu_{\mathsf{S}}, \mu_{\mathsf{R}}$.

We require the following properties on the TPKE scheme.

Homomorphism: Namely,

$$\mathsf{TEnc}(M; r) \otimes \mathsf{TEnc}(M'; r') = \mathsf{TEnc}(M \oplus M'; r \odot r'),$$

where $\otimes, \oplus, \odot$ are the operators on the corresponding spaces.

Semantic security: Assuming either S or R is honest, we require that for all M, the ciphertext $\mathsf{Enc}(M; r)$ for random r is computationally indistinguishable from random elements over some group.

Verifiable shuffles Consider a set of ciphertexts $C_i = \mathsf{TEnc}(M_i; r_i)$ for $1 \le i \le n$ of the TPKE scheme forming by S. Let I be the identity element of the message space. It is easy enough for R to choose a permutation π on $\{1, \ldots, n\}$, and random s_i to form the set of $C'_i = C_{\pi(i)} \otimes \mathsf{TEnc}(I; s_i)$ for $1 \le i \le n$, so that both sets of ciphertexts contain the same plaintexts. The set of $C'_i (1 \le i \le n)$ is called a shuffle of the original one. If the scheme TPKE is semantically secure, publishing the shuffle $C'_i (1 \le i \le n)$ reveals nothing on the permutation π to S. Correctness of the shuffle is verified via the following protocol

$$\mathsf{ZKPK}\big\{(\pi, s_i) : C'_i = C_{\pi(i)} \otimes \mathsf{TEnc}(I; s_i) \forall 1 \le i \le n\big\},$$

which has efficient implementations for homomorphic encryption schemes such as ElGamal or Paillier[3] as shown in the work of Groth and Lu [14]. More generally, the results of Groth and Lu apply for homomorphic encryption schemes with the following properties:

Proper message space: the order of the message space does not have any small prime factor (say less than 2^{80}).

Root extraction: from $C^e = \mathsf{TEnc}(M; R)$, it is possible to efficiently extract (m, r) such that $C = \mathsf{TEnc}(m; r)$ for every e co-prime with the order of the message space.

The protocols for verifiable shuffles given in [14] are statistical strong HVZK arguments of three rounds, and can be turned into fully zero-knowledge by standard techniques.

The additional property below will be needed in proving sender security.

[3] However, the Paillier encryption scheme with *threshold decryption* needs a setup assumption for the secret keys.

Computing μ_{S} without sk_{S}: Let $\mu_{\mathsf{S}} = \mathsf{TDec}(sk_{\mathsf{S}}, C')$ where $C' = C'_{\pi^{-1}(\sigma)}$ as above for some $1 \leq \sigma \leq n$. Note that $C' = C_\sigma \otimes \mathsf{TEnc}(I; s_{\pi^{-1}(\sigma)})$ for $C_\sigma = \mathsf{TEnc}(M_\sigma; r_\sigma)$. We require that μ_{S} can be alternatively expressed as a function of $pk, C_\sigma, M_\sigma, s_{\pi^{-1}(\sigma)}$, and sk_{R}. Namely there exists an efficiently-computable function f such that we have $\mu_{\mathsf{S}} = f(pk, C_\sigma, M_\sigma, s_{\pi^{-1}(\sigma)}, sk_{\mathsf{R}})$.

3.2 The OT protocol

Initialization:

1. The sender S and the receiver R run the protocol TGen so that they obtain a common public key $pk = (pk_{\mathsf{S}}, pk_{\mathsf{R}})$; and S gets secret key sk_{S}, R gets secret key sk_{R}.The receiver R proves in ZKPK that he knows sk_{R} corresponding to pk_{R}.
2. For $1 \leq i \leq n$, S computes and sends
$$C_i = \mathsf{TEnc}(M_i; r_i)$$
to R where r_i are randomness used by TEnc.
3. The sender S then proves to R by ZKPK that he knows M_i for all i. (This is equivalent to proving the knowledge of r_i in our below instantiations.)
4. (**Shuffling**) The receiver R chooses a permutation π on $\{1, \ldots, n\}$ and randomness s_i for $1 \leq i \leq n$, and computes then sends to S for all i
$$C'_i = C_{\pi(i)} \otimes \mathsf{TEnc}(I; s_i),$$
where I is the unit element of the message space.
5. The receiver R proves to S in ZKPK that he knows π and $s_i (1 \leq i \leq n)$ satisfying the equation at Step 4.

The j-th transfer:

6. The receiver R obtains an index σ as input, and sends $\pi^{-1}(\sigma)$ to S.
7. The sender S checks $\pi^{-1}(\sigma) \in \{1, \ldots, n\}$, then computes and sends $\mu_{\mathsf{S}} = \mathsf{TDec}(sk_{\mathsf{S}}, C'_{\pi^{-1}(\sigma)})$ to R.
8. The sender S then proves in ZKPM that he did the right (partial) decryption, with sk_{S} corresponding to pk_{S}, in the above step.
9. The receiver R himself computes $\mu_{\mathsf{R}} = \mathsf{TDec}(sk_{\mathsf{R}}, C'_{\pi^{-1}(\sigma)})$, and then obtaining
$$M_\sigma = \mathsf{TComb}(pk, C'_{\pi^{-1}(\sigma)}, \mu_{\mathsf{S}}, \mu_{\mathsf{R}}).$$

To prove correctness of the OT, note that

$$C' = C'_{\pi^{-1}(\sigma)} = C_\sigma \otimes \mathsf{TEnc}(I; s_{\pi^{-1}(\sigma)}) = \mathsf{TEnc}(M_\sigma, r_\sigma) \otimes \mathsf{TEnc}(I; s_{\pi^{-1}(\sigma)})$$

which means C' encrypts the plaintext M_σ thanks to the homomorphic property of the threshold PKE scheme. Now, by the correctness of the threshold PKE scheme, $\mathsf{TComb}(pk, C', \mu_{\mathsf{S}}, \mu_{\mathsf{R}})$ is exactly M_σ as required.

Theorem 1 *The generic* $\mathrm{OT}^n_{k\times 1}$ *from verifiable shuffles above is fully simulatable, if the* TPKE *scheme has semantic security.*

The proof is postponed in Appendix A.

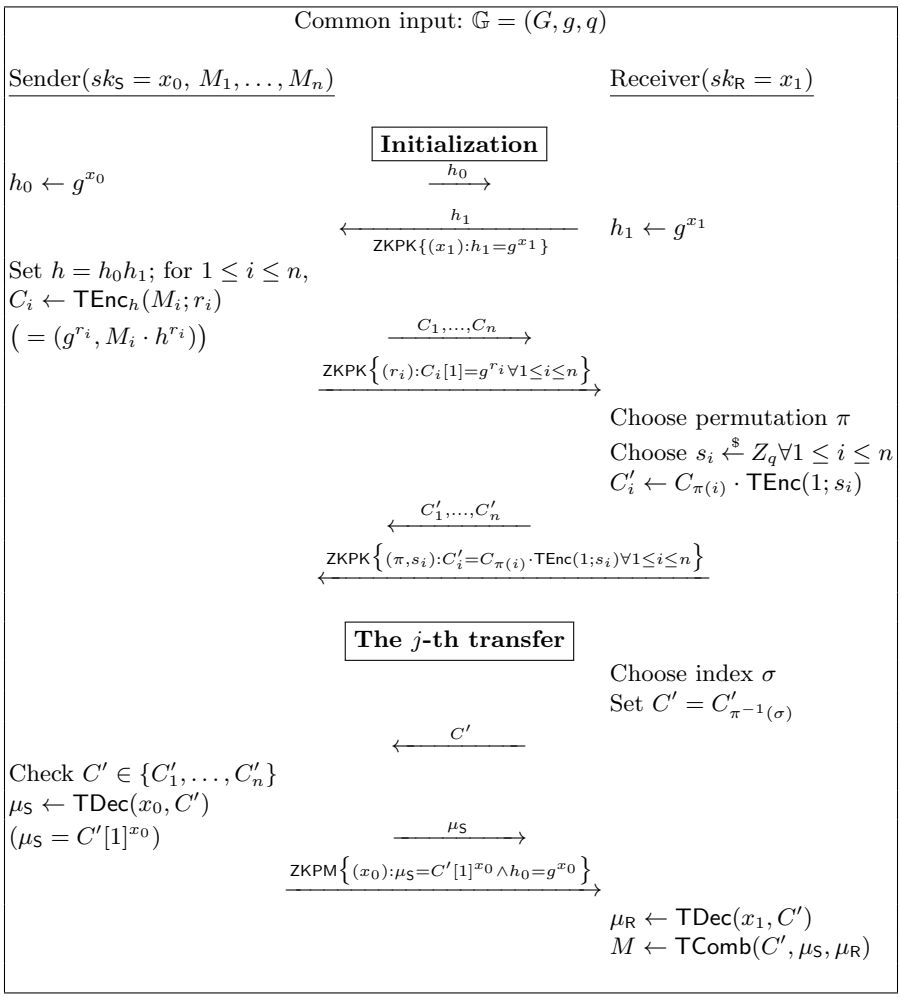

Fig. 1. The $\mathrm{OT}^n_{k\times 1}$ secure under the DDH assumption.

3.3 Instantiations from DDH and linear assumptions

$\mathbf{OT}^{\boldsymbol{n}}_{\boldsymbol{k}\times\mathbf{1}}$ from the DDH assumption We will use the threshold ElGamal encryption scheme. The scheme works on a cyclic group $\mathbb{G} = (G, g, q)$ where g is the generator of prime order q, and has semantic security under the DDH assumption on $\mathbb{G}$.

- TGen: S chooses $sk_{\mathsf{S}} = x_0 \xleftarrow{\$} Z_q$, computes and sends $h_0 \leftarrow g^{x_0}$ to R. Similarly, R chooses $sk_{\mathsf{R}} = x_1 \xleftarrow{\$} Z_q$ and sends $h_1 \leftarrow g^{x_1}$ to S. The agreed public key is then $h = h_0 h_1$.
- $\mathsf{TEnc}(M; r)$: Output $C = (C[1], C[2]) = (g^r, M \cdot h^r)$ for $r \xleftarrow{\$} Z_q$ and $M \in G$.
- $\mathsf{TDec}(sk_{\mathsf{P}}, C)$: Output $\mu_{\mathsf{P}} = C[1]^{sk_{\mathsf{P}}}$ for P is either S or R.
- $\mathsf{TComb}(C, \mu_{\mathsf{S}}, \mu_{\mathsf{R}})$: Output $C[2]/(\mu_{\mathsf{S}}\mu_{\mathsf{R}})$.

The TPKE scheme satisfies all requirements described in Sect.3.1. Our $\mathrm{OT}^n_{k\times 1}$ instantiation from the threshold ElGamal encryption scheme is depicted in Fig.1. In the figure, the element $\mu_{\mathsf{S}} = C'[1]^{x_0}$

can be alternatively expressed as[4]

$$\mu_{\mathsf{S}} = f(pk, C_\sigma, M_\sigma, s_{\pi^{-1}(\sigma)}, sk_{\mathsf{R}}) \stackrel{\text{def}}{=} C_\sigma[2] M_\sigma^{-1} C_\sigma[1]^{-x_1} h_0^{s_{\pi^{-1}(\sigma)}},$$

which is the formula needed when proving sender security.

Since the threshold ElGamal encryption scheme has semantic security under the DDH assumption, thanks to Theorem 1, the $\mathrm{OT}^n_{k\times 1}$ in Fig.1 is fully-simulatable under the same assumption.

$\mathbf{OT}^{\boldsymbol{n}}_{\boldsymbol{k}\times\mathbf{1}}$ from the d-linear assumptions We also works on $\mathbb{G} = (G, g, q)$, and let us introduce some more notations. For two vectors $v = (v[1], \ldots, v[l]) \in G^{1\times l}$, $u = (u[1], \ldots, u[l]) \in Z_q^{1\times l}$ define

$$v \cdot u^{\mathsf{T}} = u \cdot v^{\mathsf{T}} = \prod_{i=1}^{l} v[i]^{u[i]} \in G.$$

Matrix-matrix and matrix-vector multiplications are defined in the same manner. Sometimes, the $\cdot$ operators are implicitly understood. Also recall that for $u, u' \in Z_q^{1\times l}$, we have $u + u' = (u[1] + u'[1], \ldots, u[l] + u'[l])$ as normal. It is easy to check that $(u + u') \cdot v^{\mathsf{T}} = (u \cdot v^{\mathsf{T}})(u' \cdot v^{\mathsf{T}}) \in G$, and $v \cdot (u + u')^{\mathsf{T}} = (v \cdot u^{\mathsf{T}})(v \cdot u'^{\mathsf{T}}) \in G$.

For $d \geq 2$, the following PKE scheme, introduced by Naor and Segev [28], has semantic security under the d-linear assumption.

- Gen: $sk \stackrel{\$}{\leftarrow} Z_q^{(d+1)\times 1}$, $\phi \stackrel{\$}{\leftarrow} G^{d\times(d+1)}$. The secret key is sk, and the public key is $pk = (\phi, \psi)$ for $\psi = \phi \cdot sk \in G^{d\times 1}$.
- $\mathsf{Enc}(M; R)$: On message $M \in G$ and random $R \in Z_q^{1\times d}$ as input, output the ciphertext $C = (R\phi, (R\psi)M) \in G^{1\times(d+1)} \times G$.
- $\mathsf{Dec}(sk, C)$: On input C and sk, output $C[2]/(C[1] \cdot sk)$.

The correctness of the PKE scheme comes from the equation $(R \cdot \phi) \cdot sk = R \cdot (\phi \cdot sk)$.

The semantic security of the PKE scheme implies that, given ϕ, ψ, the pair $\mathsf{Enc}(1; R) = (R\phi, R\psi)$ is indistinguishable from random over $G^{1\times(d+1)} \times G$.

We now present the 2-out-of-2 threshold variant of the above PKE, whose necessary properties are checked in Appendix B. The resulting OT scheme is given in Fig.2.

- TGen: The parties S and R, using $\mathbb{G}$, agree on the matrix $\phi \in G^{d\times(d+1)}$. They then choose secrets sk_{S} and sk_{R} respectively in $Z_q^{(d+1)\times 1}$; S publishes $\psi_{\mathsf{S}} = \phi \cdot sk_{\mathsf{S}} \in G^{d\times 1}$ while R does the same with $\psi_{\mathsf{R}} = \phi \cdot sk_{\mathsf{R}} \in G^{d\times 1}$. The agreed common public key is ϕ, $\psi_{\mathsf{S}}, \psi_{\mathsf{R}}$ in which $\psi = \psi_{\mathsf{S}}\psi_{\mathsf{R}} = (\psi_{\mathsf{S}}[1]\psi_{\mathsf{R}}[1], \ldots, \psi_{\mathsf{S}}[d]\psi_{\mathsf{R}}[d])^{\mathsf{T}} \in G^{d\times 1}$ will be used in encryption. Note that $\psi = \phi \cdot (sk_{\mathsf{S}} + sk_{\mathsf{R}})$.
- $\mathsf{TEnc}(M; R)$: Output $C = \mathsf{Enc}(M; R) = (R\phi, (R\psi)M) \in G^{1\times(d+1)} \times G$ as above.
- $\mathsf{TDec}(sk_{\mathsf{P}}, C)$: Output $\mu_{\mathsf{P}} = C[1] \cdot sk_{\mathsf{P}} \in G$ for $\mathsf{P} \in \{\mathsf{S}, \mathsf{R}\}$.
- $\mathsf{TComb}(C, \mu_{\mathsf{S}}, \mu_{\mathsf{R}})$: Output $C[2]/(\mu_{\mathsf{S}}\mu_{\mathsf{R}})$.

4 Generic adaptive OT from permutation networks

We present $\mathrm{OT}^n_{k\times 1}$ with $O(1)$ communication cost for the transfer phase, while with $O(n \log n)$ for the initialization phase. The assumptions used for security will be DCR or QR.

[4] Let us elaborate a bit on the formula. We have $C_\sigma[2] M_\sigma^{-1} C_\sigma[1]^{-x_1} h_0^{s_{\pi^{-1}(\sigma)}} = (M_\sigma h^{r_\sigma}) M_\sigma^{-1} C_\sigma[1]^{-x_1} h_0^{s_{\pi^{-1}(\sigma)}} = h^{r_\sigma} (g^{r_\sigma})^{-x_1} h_0^{s_{\pi^{-1}(\sigma)}} = (h_0 h_1)^{r_\sigma} (h_1^{-r_\sigma}) h_0^{s_{\pi^{-1}(\sigma)}} = h_0^{r_\sigma + s_{\pi^{-1}(\sigma)}} = g^{(r_\sigma + s_{\pi^{-1}(\sigma)})x_0} = C'[1]^{x_0} = \mu_{\mathsf{S}}$, as required.

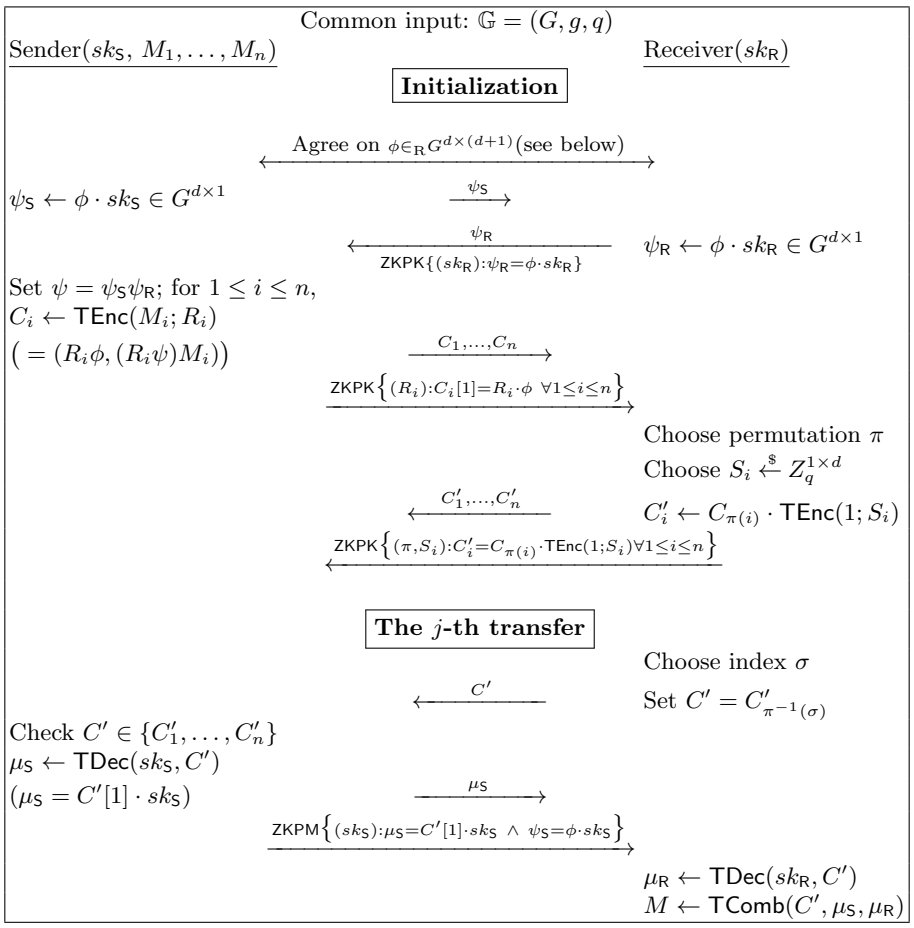

Fig. 2. The $\mathrm{OT}^n_{k\times 1}$ secure under the d-linear assumption. The matrix $\phi \in G^{d\times(d+1)}$ can be agreed by S and R as follows. S chooses and sends $\phi_\mathsf{S} = (g^{t_{i,j}})_{(i,j)} \xleftarrow{\$} G^{d\times(d+1)}$ to R, and then proves in ZK the knowledge of all $t_{i,j}$. R does the same with $\phi_\mathsf{R} = (g^{t'_{i,j}})$. The matrix ϕ is set as $(\phi_\mathsf{S}[i,j]\phi_\mathsf{R}[i,j])_{(i,j)}$, which remains uniformly distributed even if either S or R is corrupted.

4.1 Loosely homomorphic KEM

A key encapsulation mechanism KEM consists of algorithms Gen, Encap, Decap as follows: Gen produces keys (pk, sk); $\mathsf{Encap}(pk)$ outputs (ψ, K) where ψ is the encapsulation of the key K; $\mathsf{Decap}_{sk}(\psi)$ returns K as the decapsulation of ψ. We write $\mathsf{Encap}(pk; r)$ to emphasize the random coin r. We need the following conditions on KEM.

Semantic security: Suppose $\mathsf{Encap}(pk) = (\psi, K)$. Given pk, ψ, the key K is indistinguishable from random.

Uniformity of encapsulation: Let $\mathsf{Encap}(pk; r) = (\psi(r), K)$ for uniformly distributed randomness r. Then, $\psi(r)$ is also uniformly distributed over some group.

Loose homomorphism: Given (ψ, K) and (ψ', K'), there are efficiently computable functions f_1, f_2 such that

$$\mathsf{Decap}_{sk}(\psi \cdot \psi') = f_1(\psi, \psi', K, K') \text{ and } K' = f_2\left(\psi, \psi', K, \mathsf{Decap}_{sk}\left(\psi \cdot \psi'\right)\right).$$

The former equation is used in proving sender security, while the latter is for the OT's correctness. It is clear that a KEM is loosely homomorphic if it is homomorphic (namely, satisfying $\mathsf{Decap}_{sk}(\psi \cdot \psi') = K \oplus K'$).

Let us now show some concrete examples of loosely homomorphic KEM.

First example $\mathsf{KEM}_{\mathrm{DCR}}$: Gen generates primes p, q, setting $pk = N = pq$, and $sk = (p, q)$. Encap takes $r \stackrel{\$}{\leftarrow} Z_N$ and computes $(\psi, K) \in Z_N^2$ satisfying $r^N = \psi + K \cdot N \bmod N^2$. Note that $\psi = [r^N \bmod N] \in Z_N$. Using sk, $\mathsf{Decap}_{sk}(\psi)$ first computes r satisfying $r^N = \psi \bmod N$, an then outputs $K = (r^N - \psi \bmod N^2)/N$. The computation $\psi \cdot \psi'$ is normally defined over Z_N.

The semantic security of $\mathsf{KEM}_{\mathrm{DCR}}$ comes from the DCR assumption. To show that it is loosely homomorphic, consider (ψ, K) and (ψ', K') satisfying $r^N = \psi + K \cdot N \bmod N^2$, and $r'^N = \psi' + K' \cdot N \bmod N^2$. Writing $\psi\psi' = S + TN \bmod N^2$, we have

$$(rr')^N = [(\psi + KN)(\psi' + K'N) \bmod N^2] = [S + (T + K\psi' + K'\psi)N \bmod N^2],$$

so that $\hat{K} = \mathsf{Decap}_{sk}(\psi\psi' \in Z_N) = T + K\psi' + K'\psi \bmod N$, which is the function f_1. Moreover, since $(\psi + KN)(\psi' + K'N) = S + \hat{K}N \bmod N^2$, the key K' can be computed as

$$K' = \frac{[(S + \hat{K}N)(\psi + KN)^{-1} - \psi'] \bmod N^2}{N},$$

which expresses the function f_2.

Second example $\mathsf{KEM}_{\mathrm{QR}}$: To apply the recent 3-move ZKPK of Cramer and Damgård [6], we will use an expanded version of the Goldwasser-Micali encryption scheme. In particular, Gen is the same as above, except that a quadratic non-residue $y \in \mathcal{QNR}_N^{+1}$ is added to pk. The algorithm Encap takes $K \stackrel{\$}{\leftarrow} \{0,1\}^\ell$ and $r \stackrel{\$}{\leftarrow} Z_N^\ell$, returning the key K and its encapsulation

$$\psi = \left(y^{K[1]} r[1]^2 \bmod N, \ldots, y^{K[\ell]} r[\ell]^2 \bmod N\right).$$

The algorithm $\mathsf{Decap}_{sk}(\psi)$, for $1 \le i \le \ell$, returns $K[i] = 0$ if $\psi[i]$ is a quadratic residue modulo N; otherwise returns $K[i] = 1$. The scheme $\mathsf{KEM}_{\mathrm{QR}}$ is homomorphic, and has semantic security under the QR assumption.

In [6], the protocol $\mathsf{WIPK}\{(K, r) : \psi = \mathsf{Encap}\big(N; (K, r)\big)\}$, is realized by a Σ-protocol, with soundness error $2^{-\ell}$ and communication cost $O(\ell)$ (instead of $O(\ell^2)$ via the cut-and-choose technique). Turning the Σ-protocol into a fully zero-knowledge one with 4 moves can be done by standard techniques (e.g., see [8]).

4.2 The OT protocol

We show that an adaptive $\mathrm{OT}_{k\times 1}^n$ can be constructed from a loosely homomorphic $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encap}, \mathsf{Decap})$.

Initialization Phase

1. The sender S generates $(pk, sk) \leftarrow \mathsf{Gen}$ and sends pk to R. The sender proves that pk is a valid public-key by ZKPM.

2. For $i = 1, \ldots, n$, the sender S generates $(\psi(r_i), K_i) = \mathsf{Encap}(pk; r_i)$ by choosing r_i randomly and sends to R
$$C_i = (A_i, B_i) = (\psi(r_i), K_i M_i),$$
where r_i is a random string used by Encap.
3. The sender proves by ZKPK that he knows r_i of $\psi(r_i)$ for every $1 \leq i \leq n$. Alternatively, he proves that he knows sk by ZKPK.
4. (**Permuting and Blinding**) The receiver chooses u_i randomly for $1 \leq i \leq n$, and generates
$$\mathsf{Encap}(pk; u_i) = (\varphi(u_i), K'_i).$$
He then randomly picks a permutation π on $\{1, \ldots, n\}$, computes $U_i = A_{\pi(i)} \cdot \varphi(u_i)$, and sends $U_1, \ldots, U_n$ to the sender. The receiver, equipped with secrets $(u_1, \ldots, u_n)$ and π, proves in ZKPK that
$$\Big[U_1 = A_{\pi(1)} \cdot \varphi(u_1)\Big] \wedge \cdots \wedge \Big[U_n = A_{\pi(n)} \cdot \varphi(u_n)\Big].$$
We will describe in Sect.4.3 how to perform the ZKPK with $O(n \log n)$ communication cost.

The jth Transfer Phase

5. The receiver chooses an index $1 \leq \sigma \leq n$, then sends $U = U_{\pi^{-1}(\sigma)}$.
6. The sender checks $U \in \{U_1, \ldots, U_n\}$, computes $\hat{K} = \mathsf{Decap}_{sk}(U)$ and sends $\hat{K}$ to the receiver.
7. The sender proves that $\hat{K} = \mathsf{Decap}_{sk}(U)$ by ZKPM.
8. Note that $U = A_\sigma \cdot \varphi(u_{\pi^{-1}(\sigma)})$. The receiver computes
$$K_\sigma = f_2\left(A_\sigma, \varphi(u_{\pi^{-1}(\sigma)}), K'_{\pi^{-1}(\sigma)}, \hat{K}\right),$$
and then obtains M_σ by computing $B_\sigma K_\sigma^{-1}$. (In additive groups, this becomes $B_\sigma - K_\sigma$, and all B_i as above are $K_i + M_i$.)

Theorem 2 *The generic* $\mathrm{OT}^n_{k\times 1}$ *from permutation networks above is fully simulatable, if the* KEM *scheme has semantic security. In other words, loosely-homomorphic KEM implies adaptive OT.*

The proof is postponed in Appendix C. Below we will show how to obtain efficient instantiations based on specific complexity assumptions.

4.3 How to execute the ZKPK at Step 4

The case $n = 2$: First, let us focus on $n = 2$, proving the knowledge of u_1, u_2 such that
$$U_1 = A_{\pi(1)} \cdot \varphi(u_1) \wedge U_2 = A_{\pi(2)} \cdot \varphi(u_2),$$
for *some* permutation π on $\{1, 2\}$. The task is equivalent to proving
$$\Big(U_1 = A_1 \cdot \varphi(u_1) \wedge U_2 = A_2 \cdot \varphi(u_2)\Big) \vee \Big(U_1 = A_2 \cdot \varphi(u_1) \wedge U_2 = A_1 \cdot \varphi(u_2)\Big),$$
depending on whether $\big(\pi(1), \pi(2)\big) = (1, 2)$ or $(2, 1)$. Expanding further, what is proved becomes
$$\begin{aligned}&\Big(U_1 = A_1 \cdot \varphi(u_1) \vee U_1 = A_2 \cdot \varphi(u_1)\Big) \wedge \Big(U_1 = A_1 \cdot \varphi(u_1) \vee U_2 = A_1 \cdot \varphi(u_2)\Big)\\ \wedge &\Big(U_2 = A_2 \cdot \varphi(u_2) \vee U_1 = A_2 \cdot \varphi(u_1)\Big) \wedge \Big(U_2 = A_2 \cdot \varphi(u_2) \vee U_2 = A_1 \cdot \varphi(u_2)\Big).\end{aligned}$$

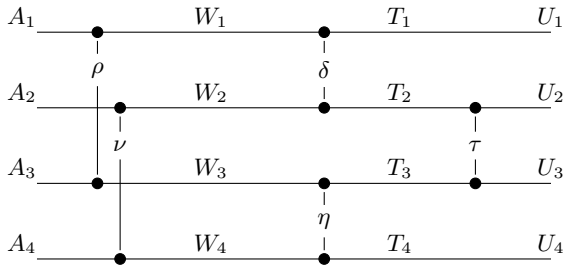

Fig. 3. From $n = 2$ to $n = 4$ with a permutation network of five switches.

The above are exactly four OR-proofs. If one can implement the interactive proof WIPK$\{(u) : U = A \cdot \varphi(u)\}$ by a Σ-protocol, then it is well-known that one can efficiently realize the OR-proofs also with Σ-protocols. Note that if u_1, u_2 are known, then the permutation π can be extracted as well. Transforming Σ-protocols to ZKPK ones can be done by well-known techniques [7]. Therefore, the ZKPK for $n = 2$ in consideration can be implemented in four rounds, and we count its communication cost asymptotically as $O(1)$.

From 2 to general n: We will use the idea of n-permutation networks, which turn n inputs to n outputs, and the outputs are a permutation of the inputs. It is known that n-permutation networks can be built from 2-ones, which are called switches. There are constructions of n-permutation networks with $O(n \log^2 n)$ [5] or even $O(n \log n)$ switches [1,10]. A comprehensive treatment on the topic can be found in [5, Chapter 28].

The idea is now we replace the switches by the WIPK protocol for $n = 2$ described above. We need $O(n \log n)$ protocols as switches, and each protocol requires $O(1)$ communication cost, so that the total communication cost becomes $O(n \log n)$.

Let us concretely illustrate how one proceeds from $n = 2$ to $n = 4$, using the permutation network depicted in Fig.3 of five switches. The elements W_i, T_i, U_i are sent to the sender by the receiver[5]. The first two switches ρ and ν prove that

$$W_1 = A_{\rho(1)} \cdot \varphi(w_1), W_3 = A_{\rho(3)} \cdot \varphi(w_3), W_2 = A_{\nu(2)} \cdot \varphi(w_2), W_4 = A_{\nu(4)} \cdot \varphi(w_4).$$

Consequently, the second two switches δ and η ensure

$$T_1 = W_{\delta(1)} \cdot \varphi(t_1), T_2 = W_{\delta(2)} \cdot \varphi(t_2), T_3 = W_{\eta(3)} \cdot \varphi(t_3), T_4 = W_{\eta(4)} \cdot \varphi(t_4).$$

The final switch τ is between T_2 and T_3, showing

$$U_2 = T_{\tau(2)} \cdot \varphi(v_2) \wedge U_3 = T_{\tau(3)} \cdot \varphi(v_3).$$

To ease the illustration, let us take concrete switches $\tau = (2\ 3)$ (namely 2 to 3 and vice versa), $\delta = (1\ 2)$, $\nu = (2\ 4)$, and the others are identity switches. Denote $U \sim A$ if there is u such that $U = A \cdot \varphi(u)$, so that

$$U_1 \sim T_1 \sim W_2 \sim A_4$$
$$U_2 \sim T_3 \sim W_3 \sim A_3$$
$$U_3 \sim T_2 \sim W_1 \sim A_1$$
$$U_4 \sim T_4 \sim W_4 \sim A_2$$

[5] In general, the receiver needs to send n ($= 4$ in Fig.3) elements at $O(\log n)$ ($= 3$ in Fig.3) steps.

which means (U_1, U_2, U_3, U_4) blinds and permutes (A_1, A_2, A_3, A_4) as expected.

Instantiations: As shown above, we just need to implement the atomic WIPK$\{(u) : U = A \cdot \varphi(u)\}$ by a Σ-protocol.

DCR assumption: Set $\varphi(u) = u^N \bmod N$ for $u \in Z_N$, so that the atomic WIPK is similar to the GQ proof [15].

QR assumption: Set

$$\varphi\left(u = (K, r)\right) = \left(y^{K[1]} r[1]^2 \bmod N, \ldots, y^{K[\ell]} r[\ell]^2 \bmod N\right)$$

for $u = (K, r) \in Z_2^\ell \times Z_N^\ell$. The elegant result of Cramer and Damgård [6] gives us the desired 3-move WIPK with soundness error $2^{-\ell}$.

4.4 How to execute other zero-knowledge protocols

The ZKPM at step 7, in the case of $\mathsf{KEM}_{\mathrm{DCR}}$, is equivalent to proving $r^N = U + \hat{K}N \bmod N^2$ for some r, for which the 4-move ZK protocol can be found in [24]. For $\mathsf{KEM}_{\mathrm{QR}}$, proving $U = (y^{\hat{K}[1]} r[1]^2 \bmod N, \ldots, y^{\hat{K}[\ell]} r[\ell]^2 \bmod N)$ for some $r \in Z_N^\ell$ is needed, which can be accomplished by the 4-move ZK protocol for the knowledge of ℓ square roots in [6].

We now turn to the necessary protocols for the validity of the public key. Proving y is a quadratic non-residue can be done in 4 moves as in [6]. What is left is how to prove the validity of N, namely $N = pq$ for some distinct primes p, q. We proceed in two steps: first proving $\gcd(N, \phi(N)) = 1$ in 4 moves (see Appendix D), and then showing $N = pq$ as required. Merging the moves of the former and the latter gives us the 4-move protocol for the validity of N. The latter protocol is accomplished as follows.

Proving $N = pq$ in four moves: Suppose $\gcd(N, \phi(N)) = 1$ and N is not a prime, so that $N = \prod_{i=1}^{\nu} p_i$ for $\nu \geq 2$, and let $y \in \mathcal{J}_N^{+1} \setminus (Z_N^*)^2$. Improving a cut-and-choose protocol in [20], the following protocol proves that $\nu = 2$.

1. The verifier sends random $z_1, \ldots, z_\ell \in \mathcal{J}_N^{+1}$ to the prover.
2. The prover shows that there are $(m_i, r_i) \in Z_2 \times Z_N^*$ satisfying

$$z_1 = y^{m_1} r_1^2 \wedge \cdots \wedge z_\ell = y^{m_\ell} r_\ell^2,$$

by the 4-move ZK protocol of Cramer and Damgård [6], whose communication cost is $O(\ell)$ elements in Z_N^* with soundness error $2^{-\ell}$.

The completeness, soundness, and zero-knowledge properties are checked in Appendix E.

5 Extensions

5.1 UC-secure OT under the DDH assumption

Although being fully-simulatable, the schemes in previous sections is not UC-secure. The reason is that rewinding is used in the zero-knowledge proofs. Therefore, to obtain UC-secure OT protocols, it suffices to use UC-secure zero-knowledge proofs in our constructions.

Observe that all zero-knowledge proofs used in Sect.3 can be effectively realized from Σ-protocols. Therefore, if we can turn a Σ-protocol into a UC-secure zero-knowledge one, then we are done. Fortunately, such a transformation is presented by Hazay and Nissim [17] (see also [26]) with the help of a UC-secure commitment scheme (e.g., by Lindell [25]), with a bit sacrifice in efficiency. Let $(a, c \in \{0,1\}, z)$ be transcripts of the Σ-protocol, we roughly describe the transformation here for completeness.

1. The prover generates $(a, 0, z_0)$ and $(a, 1, z_1)$. The prover then commits a, z_0, z_1 to the verifier via a UC-secure commitment scheme (e.g. [25]).
2. The verifier sends back a challenge $c \stackrel{\$}{\leftarrow} \{0,1\}$.
3. The prover decommits a, z_c. The verifier then checks whether (a, c, z_c) is correct as in the basic Σ-protocol.

One needs to repeat the above L times to obtain soundness error 2^{-L}. Plugging this transformation into the instantiation in Sect.3, we obtain a UC-secure OT protocol under the DDH assumption. (Note that the UC-secure commitment scheme of Lindell [25] is under the DDH assumption.) The trade-off for this higher security is the efficiency loss by a factor L.

5.2 Leakage-resilient adaptive OT under the DDH assumption

Let us re-consider the generic construction given in Sect.4, yet further assume that the randomness of the sender at the first step is somehow leaked. Namely, the randomness used to generate the secret sk is leaked. We show that if using leakage-resilient encryption, the resulting OT instantiation remains secure. Specifically, under the DDH assumption, consider the following KEM derived from Naor and Segev [28, Eprint, Sect.5.2]. The scheme is proved leakage-resilient with rate $1 - o(1)$.

- Gen: $s_1, \ldots, s_\ell \stackrel{\$}{\leftarrow} Z_q$, $g_1, \ldots, g_\ell \stackrel{\$}{\leftarrow} G$, $h = \prod_{i=1}^{\ell} g_i^{s_i}$. Set the public key $pk = (g_1, \ldots, g_\ell, h)$ and the secret key $s = (s_1, \ldots, s_\ell)$.
- Encap: Take $r \stackrel{\$}{\leftarrow} Z_q$, the encapsulation of $K = h^r$ is $(g_1^r, \ldots, g_\ell^r)$.
- Decap: To decapsulate $(g_1^r, \ldots, g_\ell^r)$, compute $\prod_{i=1}^{\ell} (g_i^r)^{s_i}$, which equals h^r.

The KEM is homomorphic and leakage-resilient, so we can apply the result in Sect.4 to obtain a leakage-resilient OT protocol under the DDH assumption. The atomic WIPK used in the permutation network is to essentially prove the knowledge of $r \in Z_q$ satisfying $C = (g_1^r, \ldots, g_\ell^r)$. This WIPK and other required zero-knowledge proofs can be realized efficiently.

References

1. M. Ajtai, J. Komlós, and E. Szemerédi. An $O(n \log n)$ sorting network. In *STOC*, pages 1–9. ACM, 1983.
2. A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Y. Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003.
3. J. Camenisch, G. Neven, and A. Shelat. Simulatable adaptive oblivious transfer. In M. Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 2007.
4. C.-K. Chu and W.-G. Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In S. Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2005.
5. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms, second edition, 2001.

6. R. Cramer and I. Damgård. On the amortized complexity of zero-knowledge protocols. In Halevi [16], pages 177–191.
7. R. Cramer, I. Damgård, and P. D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In H. Imai and Y. Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–373. Springer, 2000.
8. I. Damgård. On Σ-protocols, 2010. Course notes in Cryptologic Protocol Theory, `http://www.daimi.au.dk/~ivan/CPT.html`.
9. I. Damgård, C. Hazay, and A. Patra. Leakage resilient secure two-party computation. Cryptology ePrint Archive, Report 2011/256, 2011. `http://eprint.iacr.org/`.
10. M. T. Goodrich. Randomized shellsort: A simple oblivious sorting algorithm. In *Proceedings 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.
11. M. Green and S. Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In K. Kurosawa, editor, *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 265–282. Springer, 2007.
12. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In J. Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 179–197. Springer, 2008.
13. M. Green and S. Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In Ishai [18], pages 347–363.
14. J. Groth and S. Lu. Verifiable shuffle of large size ciphertexts. In T. Okamoto and X. Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 377–392. Springer, 2007.
15. L. C. Guillou and J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, editor, *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 1988.
16. S. Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.
17. C. Hazay and K. Nissim. Efficient set operations in the presence of malicious adversaries. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 312–331. Springer, 2010.
18. Y. Ishai, editor. *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*. Springer, 2011.
19. S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.
20. K. Kurosawa, Y. Katayama, W. Ogata, and S. Tsujii. General public key residue cryptosystems and mental poker protocols. In *EUROCRYPT*, pages 374–388, 1990.
21. K. Kurosawa and R. Nojima. Simple adaptive oblivious transfer without random oracle. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 334–346. Springer, 2009.
22. K. Kurosawa, R. Nojima, and L. T. Phong. Efficiency-improved fully simulatable adaptive OT under the DDH assumption. In J. A. Garay and R. D. Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 172–181. Springer, 2010.
23. K. Kurosawa, R. Nojima, and L. T. Phong. Generic fully simulatable adaptive oblivious transfer. In J. Lopez and G. Tsudik, editors, *ACNS*, volume 6715 of *Lecture Notes in Computer Science*, pages 274–291, 2011.
24. K. Kurosawa and T. Takagi. New approach for selectively convertible undeniable signature schemes. In X. Lai and K. Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 428–443. Springer, 2006.
25. Y. Lindell. Highly-efficient universally-composable commitments based on the DDH assumption. In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 446–466. Springer, 2011.
26. Y. Lindell and B. Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In Ishai [18], pages 329–346.
27. M. Naor and B. Pinkas. Oblivious transfer with adaptive queries. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 1999.
28. M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In Halevi [16], pages 18–35. Full version available at `http://eprint.iacr.org/2009/105.pdf`.
29. C. A. Neff. A verifiable secret shuffle and its application to e-voting. In *ACM Conference on Computer and Communications Security*, pages 116–125, 2001.

30. W. Ogata and K. Kurosawa. Oblivious keyword search. *J. Complexity*, 20(2-3):356–371, 2004. Also available at `http://eprint.iacr.org/2002/182`.
31. B. Zhang. Simulatable adaptive oblivious transfer with statistical receiver's privacy. In X. Boyen and X. Chen, editors, *ProvSec*, volume 6980 of *Lecture Notes in Computer Science*, pages 52–67. Springer, 2011.

A Proof of Theorem 1

Lemma 3 (Receiver security) *The* $\mathrm{OT}^n_{k\times 1}$ *protocol in Sect.3 is secure against sender corruption.*

Proof. For every real-world adversary $\mathcal{A}$ who corrupts the sender, we construct an ideal-world adversary $\mathcal{A}'$ such that the advantage $\mathbf{Adv}(\mathcal{Z})$ is negligible.

We will consider a sequence of games beginning from game G_0, which is the real world experiment, and proceed to the final game, which is the ideal world experiment as in Sec.2.2. For each integer i, let

$$\Pr(G_i) = \Pr\left(\mathcal{Z} = 1 \text{ in game } G_i\right),$$

and denote $\Pr(G_i) \approx \Pr(G_j)$ when the two values are negligibly close.

Game G_0: This is the real world experiment such that the sender is controlled by the adversary $\mathcal{A}$. By definition $\Pr(G_0) = \Pr(\mathcal{Z} = 1 \text{ in the real world})$.

Game G_1: This game is the same as the previous one except the following. In the initialization phase, the game extracts $M_i^* (1 \le i \le n)$ from $\mathcal{A}$ by using the knowledge extractor of the ZKPK.

If it fails, then the protocol stops. Since the failure occurs with negligible probability, we have $\Pr(G_0) \approx \Pr(G_1)$.

Game G_2: This game is the same as game G_1 except that, in the initialization phase, the game uses two zero-knowledge simulators of the ZKPKs for proving the knowledge of sk_{R}, and (π, s_i) respectively. Since ZKPK protocols are zero-knowledge, we have $\Pr(G_1) \approx \Pr(G_2)$.

Game G_3: This game is the same as the previous one, except that $C'_i (1 \le i \le n)$ are randomly chosen from the ciphertext space. We have $\Pr(G_3) \approx \Pr(G_2)$, thanks to the semantic security of the threshold encryption scheme. (Namely, $\mathsf{TEnc}(I; s_i)$ are almost random, so are $C'_i = C_{\pi(i)}\mathsf{TEnc}(1; s_i)$.)

Game G_4: This game is the same as the previous one except the following. In each transfer phases, the receiver chooses C' randomly and distinctly from the set $\{C'_1, \ldots, C'_n\}$. Since the view of $\mathcal{A}$ is unchanged, we have $\Pr(G_4) = \Pr(G_3)$.

Game G_5: This game is the ideal world experiment in which an ideal-world adversary $\mathcal{A}'$ uses $\mathcal{A}$ as a black-box as follows.

1. $\mathcal{A}'$ receives $(M_1, \ldots, M_n)$ from $\mathcal{Z}$, and sends $(M_1, \ldots, M_n)$ to $\mathcal{A}$.
2. $\mathcal{A}'$ runs game G_4 with $\mathcal{A}$, where $\mathcal{A}'$ plays the role of the receiver, which can be accomplished since σ, the secret index of the receiver, is not used in game G_4.
3. $\mathcal{A}'$ sends the extracted $(M_1^*, \ldots, M_n^*)$ as in game G_1 to $\mathcal{F}_{\text{adapt}}$.
4. In each transfer phase, if $\mathcal{A}$ behaved in an acceptable way, then $\mathcal{A}'$ sends $b = 1$ to $\mathcal{F}_{\text{adapt}}$. Otherwise $\mathcal{A}'$ sends $b = 0$ to $\mathcal{F}_{\text{adapt}}$.
5. Suppose that $\mathcal{A}$ sends $\mathcal{A}_{\text{out}}$ to $\mathcal{Z}$ at the end of the game. Then $\mathcal{A}'$ sends $\mathcal{A}'_{\text{out}} = \mathcal{A}_{\text{out}}$ to $\mathcal{Z}$.

We have $\Pr(G_4) = \Pr(G_5)$, and by definition $\Pr(\mathcal{Z} = 1 \text{ in the ideal world}) = \Pr(G_5)$. Summing up all above, we have $\mathbf{Adv}(\mathcal{Z}) = |\Pr(G_0) - \Pr(G_5)|$ is negligible as required. □

Lemma 4 (Sender security) *The* $\mathrm{OT}^n_{k\times 1}$ *protocol in Sect.3 is secure against receiver corruption.*

Proof. For every real-world adversary $\mathcal{A}$ who corrupts the receiver, we construct an ideal-world adversary $\mathcal{A}'$ such that the advantage of the environment $\mathbf{Adv}(\mathcal{Z})$ is negligible.

We again consider a sequence of games in which the first is the real world experiment of Sec.2.2, while the final is the ideal world experiment. Again, let $\Pr(G_i) = \Pr(\mathcal{Z} = 1 \text{ in game } G_i)$.

Game G_0: In this game the receiver is controlled by the adversary $\mathcal{A}$, and by definition $\Pr(G_0) = \Pr(\mathcal{Z} = 1 \text{ in the real world})$.

Game G_1: This game is the same as game G_0 except the following. In the initialization phase, the game extracts sk_{R}, and (π, s_i) by using the extractors of the corresponding ZKPKs respectively.

Unless the extractors fail, which occurs with negligible probability, games G_1 and G_0 are identical, so that $\Pr(G_1) \approx \Pr(G_0)$.

Game G_2: In this game the index σ used in the transfer phase is extracted as follows. Since $\mathcal{A}$ sends C' such that $C' \in \{C'_1, \ldots, C'_n\}$, the sender searches the index $1 \le \rho \le n$ satisfying $C' = C'_\rho$. Recall $C' = C'_{\pi^{-1}(\sigma)}$, so $\pi^{-1}(\sigma) = \rho$, and hence $\sigma = \pi(\rho)$. Since the change is syntactic, we have $\Pr(G_2) = \Pr(G_1)$.

Game G_3: This game is the same as the previous one except the following. In each transfer phase, the game computes μ_{S} as

$$\mu_{\mathsf{S}} = f(pk, C_\sigma, M_\sigma, s_{\pi^{-1}(\sigma)}, sk_{\mathsf{R}}).$$

Since the change is syntactic, we have $\Pr(G_3) = \Pr(G_2)$.

Game G_4: This game is the same as the previous one except the following. In each transfer phase, instead of running the ZKPK proving the correct decryption of C' under sk_{S}, the zero-knowledge simulator of the ZKPK is run so that $\Pr(G_4) = \Pr(G_3)$.

Game G_5: This game is the same as the previous one except the following. In the initialization phase, each C_i is randomly chosen. It is easy to see that $\Pr(G_5) \approx \Pr(G_4)$ thanks to the semantic security of the TPKE scheme.

Game G_6: This game is the ideal world experiment in which an ideal-world adversary $\mathcal{A}'$ uses $\mathcal{A}$ as a black-box as follows.

1. $\mathcal{A}'$ runs game G_5 with $\mathcal{A}$, where $\mathcal{A}'$ plays the role of the sender.
2. In each transfer phase, $\mathcal{A}'$ sends σ which is extracted as in game G_2 to $\mathcal{F}_{\mathrm{adapt}}$, and obtains M_σ. Then $\mathcal{A}'$ computes μ_{S} as in game G_3.
3. Suppose that $\mathcal{A}$ sends $\mathcal{A}_{\mathrm{out}}$ to $\mathcal{Z}$ at the end of the game. Then $\mathcal{A}'$ sends $\mathcal{A}'_{\mathrm{out}} = \mathcal{A}_{\mathrm{out}}$ to $\mathcal{Z}$.

We have by definition $\Pr(G_6) = \Pr(\mathcal{Z} = 1 \text{ in the ideal world})$. Summing up all above, we have $\mathbf{Adv}(\mathcal{Z}) = |\Pr(G_0) - \Pr(G_6)|$ is negligible as required. □

B The properties of the *d*-linear assumption-based threshold PKE

Correctness: The correctness of the threshold variant comes from the following equations $R\psi = R(\phi(sk_{\mathsf{S}} + sk_{\mathsf{R}})) = (R\phi)(sk_{\mathsf{S}} + sk_{\mathsf{R}}) = C[1](sk_{\mathsf{S}} + sk_{\mathsf{R}}) = (C[1]sk_{\mathsf{S}})(C[1]sk_{\mathsf{R}}) = \mu_{\mathsf{S}}\mu_{\mathsf{R}} \in G$.

Homomorphism: Following from below equations

$$\begin{aligned}\mathsf{TEnc}(M;R)\cdot\mathsf{TEnc}(M';R') &= \big(R\phi,(R\psi)M\big)\cdot\big(R'\phi,(R'\psi)M'\big)\\ &= \big((R\phi)(R'\phi),(R\psi)(R'\psi)MM'\big)\\ &= \big((R+R')\phi,(R+R')\psi MM'\big)\\ &= \mathsf{TEnc}(MM';R+R').\end{aligned}$$

Semantic security: The security of the threshold variant can be reduced to its original PKE as follows. Note that $\mathsf{TEnc}(1;R) = \big(R\phi, R\phi\cdot(sk_{\mathsf{S}} + sk_{\mathsf{R}})\big) = \big(R\phi,(R\phi\cdot sk_{\mathsf{S}})(R\phi\cdot sk_{\mathsf{R}})\big) = \big(R\phi,(R\psi_{\mathsf{S}})(R\psi_{\mathsf{R}})\big)$.

Suppose S is corrupted. In that case we have $(R\phi, R\psi_{\mathsf{R}})$ is indistinguishable from random from the view of S, so is $\mathsf{TEnc}(1;R)$. Similarly, $\mathsf{TEnc}(1;R)$ is still random-like if R is corrupted. Therefore, even either S or R is corrupted, $\mathsf{TEnc}(1;R)$ still looks random.

Proper message space: This (and the right below property) is for the usage the shuffle protocols of [14]. The message space is G of prime order q, which does not have small prime factors if q is big enough.

Root extraction: Given $C^e = \mathsf{TEnc}(M;R) = \big(R\phi,(R\psi)M\big)$ with $(e,q)=1$, we want to extract (m,r) satisfying $C = \mathsf{TEnc}(m;r)$. This is done by just putting $m = M^{[e^{-1} \bmod q]}$, and $r = R\cdot[e^{-1} \bmod q] = \big(R[1](e^{-1} \bmod q),\ldots,R[d](e^{-1} \bmod q)\big)$.

Computing μ_{S} without sk_{S}: Referring to Fig.2, we show that the element $\mu_{\mathsf{S}} = \mathsf{TDec}(sk_{\mathsf{S}}, C') = C'[1]\cdot sk_{\mathsf{S}}$ can be computed by R in case the receiver already knew M_σ. Note that

$$C' = C'_{\pi^{-1}(\sigma)} = C_\sigma\cdot\mathsf{TEnc}(1;S_{\pi^{-1}(\sigma)}) = \mathsf{TEnc}(M_\sigma;R_\sigma + S_{\pi^{-1}(\sigma)})$$

so that $C'[1] = (R_\sigma + S_{\pi^{-1}(\sigma)})\phi$, and hence

$$\begin{aligned}\mu_{\mathsf{S}} &= (R_\sigma + S_{\pi^{-1}(\sigma)})\phi\cdot sk_{\mathsf{S}} = (R_\sigma\phi\cdot sk_{\mathsf{S}})(S_{\pi^{-1}(\sigma)}\phi\cdot sk_{\mathsf{S}})\\ &= \big(R_\sigma\phi\cdot(sk - sk_{\mathsf{R}})\big)\big(S_{\pi^{-1}(\sigma)}\phi\cdot(sk - sk_{\mathsf{R}})\big) \text{ for } sk = sk_{\mathsf{S}} + sk_{\mathsf{R}}\\ &= (R_\sigma\phi sk)\cdot(S_{\pi^{-1}(\sigma)}\phi sk)\cdot\big\{R_\sigma\phi(-sk_{\mathsf{R}})\big\}\cdot\big\{S_{\pi^{-1}(\sigma)}\phi(-sk_{\mathsf{R}})\big\}\\ &= (C_\sigma[2]M_\sigma^{-1})\cdot(S_{\pi^{-1}(\sigma)}\psi)\cdot\big\{C_\sigma[1](-sk_{\mathsf{R}})\big\}\cdot\big\{S_{\pi^{-1}(\sigma)}\phi(-sk_{\mathsf{R}})\big\}\\ &\stackrel{\text{def}}{=} f(\phi,\psi,C_\sigma,M_\sigma,S_{\pi^{-1}(\sigma)},sk_{\mathsf{R}}),\end{aligned}$$

which is a function of what R has, as desired.

C Proof of Theorem 2

Lemma 5 (Receiver security) *The* $\mathrm{OT}^n_{k\times 1}$ *protocol in Sect.4 is secure against sender corruption.*

Proof. For every real-world adversary $\mathcal{A}$ who corrupts the sender, we construct an ideal-world adversary $\mathcal{A}'$ such that the advantage $\mathbf{Adv}(\mathcal{Z})$ is negligible. We consider a series of games as follows.

Game G_0: This is exactly the real-world experiment where the sender is corrupted.

Game G_1: This game is the same as the above game except that it extracts the secret key sk from the corrupted sender. It is easy to see that $\Pr(G_1) \approx \Pr(G_0)$.

Game G_2: The difference in this game is that $U_1, \ldots, U_n$ is chosen randomly and sent to the sender. Then the simulator for the corresponding ZKPK (at step 4) is run. It is clear that $\Pr(G_2) \approx \Pr(G_1)$.

Game G_3: This is the ideal experiment in which $\mathcal{A}'$ runs game G_2 with $\mathcal{A}$. Since $\mathcal{A}'$ extracts sk from $\mathcal{A}$, it obtains M_i^* from C_i for $1 \le i \le n$ and sends all the messages to $\mathcal{F}_{\text{adapt}}$. In each transfer, $\mathcal{A}'$ chooses U randomly and distinctly from $\{U_1, \ldots, U_n\}$. Moreover, if the ZKPM (at step 7) passes, $\mathcal{A}'$ sends 1, otherwise sends 0 to $\mathcal{F}_{\text{adapt}}$.

We thus have $\Pr(G_3) = \Pr(G_2)$ and hence $\Pr(G_3) \approx \Pr(G_0)$, meaning the ideal and real worlds are indistinguishable, so that $\mathbf{Adv}(\mathcal{Z})$ must be negligible as required. □

Lemma 6 (Sender security) *The* $\mathrm{OT}^n_{k\times 1}$ *protocol in Sect.4 is secure against receiver corruption.*

Proof. For every real-world adversary $\mathcal{A}$ who corrupts the receiver, we construct an ideal-world adversary $\mathcal{A}'$ such that the advantage of the environment $\mathbf{Adv}(\mathcal{Z})$ is negligible. We consider a series of games as follows. First, game G_0 is the real-world experiment.

Game G_1: This game is identical to the above game, except that it extracts the secrets $u_1, \ldots, u_n$ and π from the corrupted receiver. We have $\Pr(G_1) \approx \Pr(G_0)$.

Game G_2: In this game, the index σ is extracted as follows. In the transfer phase, when the receiver sends U, the game searches for an index $1 \le \rho \le n$ such that $U = U_\rho$. By the construction $U = U_{\pi^{-1}(\sigma)}$ so that $\rho = \pi^{-1}(\sigma)$ and hence $\sigma = \pi(\rho)$. We have $\Pr(G_2) \approx \Pr(G_1)$.

Game G_3: In this game, $\hat{K} = \mathsf{Decap}_{sk}(U_{\pi^{-1}(\sigma)}) = \mathsf{Decap}_{sk}\big(A_\sigma \cdot \varphi(u_{\pi^{-1}(\sigma)})\big)$ is alternatively computed as $\hat{K} = f_1\big(A_\sigma, \varphi(u_{\pi^{-1}(\sigma)}), B_\sigma M_\sigma^{-1}, K'_{\pi^{-1}(\sigma)}\big)$. We have $\Pr(G_3) \approx \Pr(G_2)$.

Game G_4: In this game, all $C_i = (A_i, B_i)$ are randomly chosen. By the semantic security of KEM, we have $\Pr(G_4) \approx \Pr(G_3)$.

Game G_5: This is the ideal world in which $\mathcal{A}'$ runs $\mathcal{A}$ as in game G_4. The adversary $\mathcal{A}'$ extracts σ as in game G_2, and the index is sent to $\mathcal{F}_{\text{adapt}}$ to obtain M_σ. Then the key $\hat{K}$ is computed as in game G_3. All the zero-knowledge proofs to the corrupted receiver are replaced by the simulated ones. It is clear that $\Pr(G_5) \approx \Pr(G_4)$ so that $\Pr(G_5) \approx \Pr(G_0)$, and hence $\mathbf{Adv}(\mathcal{Z})$ is negligible as required. □

D Proving $\gcd(N, \phi(N)) = 1$

Suppose that N is odd and not a prime. The factorization of N is

$$N = \prod_{i=1}^{\nu} p_i^{r_i},$$

for odd, distinct primes p_i, integers $r_i \ge 1$, and $\nu \ge 1$. We suggest the following 4-move protocol showing that $\gcd\big(N, \phi(N)\big) = 1$.

1. The verifier sends random $y_1, \ldots, y_\ell \in Z_N^*$ to the prover.
2. The prover shows in ZKPM that there are $x_i \in Z_N^*$ satisfying $y_i = x_i^N$ for all $1 \le i \le \ell$.

Completeness: if $\gcd\big(N, \phi(N)\big) = 1$ then $x \mapsto x^N$ is a bijection over Z_N^*. Given $y_1, \ldots, y_\ell \in Z_N^*$, the prover then can compute the N-th roots x_i to complete the protocol.

Soundness: Suppose $\gcd\big(N, \phi(N)\big) > 1$. We have the set $(Z_N^*)^N = \{x^N : x \in Z_N^*\}$ is a subgroup of (but does not equal) Z_N^*, so that $|(Z_N^*)^N| \le \phi(N)/2$. Thus for a random $y \in Z_N^*$, $\Pr[y \in (Z_N^*)^N] \le 1/2$, so that

$$\Pr[\exists x_i \in Z_N^* : y_i = x_i^N \forall 1 \le i \le \ell] = \Pr[y_i \in (Z_N^*)^N \forall 1 \le i \le \ell] \le 2^{-\ell},$$

which is the soundness error.

Zero-knowledge: coming directly from the sub-protocol.

If $\gcd\big(N, \phi(N)\big) = 1$, then $r_i = 1$ for all i, which is the fact we use in proving $N = pq$. To see why, note that $\phi(N) = \prod_{i=1}^{\nu} p_i^{r_i - 1}(p_i - 1)$, so that $1 = \gcd\big(N, \phi(N)\big) \ge \prod_{i=1}^{\nu} p_i^{r_i - 1}$, and hence $r_i = 1$ for all i.

E Proofs for ZKPM of $N = pq$

Completeness: If $\nu = 2$, namely $N = p_1 p_2$, then for all $z \in \mathcal{J}_N^{+1} = \mathcal{QR}_N \cup \mathcal{QNR}_N^{+1}$, there always exists $(m, r) \in Z_2 \times Z_N^*$ meeting $z = y^m r^2 \bmod N$.

Soundness: Suppose $\nu \ge 3$. Since N is not a square of an integer, $|\mathcal{J}_N^{+1}| = \phi(N)/2$ and $|(Z_N^*)^2| = \phi(N)/2^\nu$. Thus $[\mathcal{J}_N^{+1} : (Z_N^*)^2] = 2^{\nu-1} \ge 4$. For a random $z \in \mathcal{J}_N^{+1}$,

$$\begin{aligned}
&\Pr[z = y^m r^2 \text{ for some } m, r \in Z_2 \times Z_N^*] \\
&= \Pr[z = r^2 \text{ for some } r] + \Pr[zy^{-1} = r^2 \text{ for some } r] \\
&= \Pr[z \in (Z_N^*)^2] + \Pr[zy^{-1} \in (Z_N^*)^2] \\
&\le 2 \cdot \frac{1}{4} = \frac{1}{2},
\end{aligned}$$

so that $\Pr[z_1 = y^{m_1} r_1^2 \wedge \cdots \wedge z_\ell = y^{m_\ell} r_\ell^2] \le 2^{-\ell}$ provided that $\nu \ge 3$.

Zero-knowledge: coming directly from the protocol at step 2.