Knowledge Encryption and Its Applications to Simulatable Protocols With Low Round-Complexity

Yi Deng[1,2] and Xinxuan Zhang[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{deng, zhangxinxuan}@iie.ac.cn

Abstract. We introduce a new notion of public key encryption, *knowledge encryption*, for which its ciphertexts can be reduced to the public-key, i.e., any algorithm that can break the ciphertext indistinguishability can be used to extract the (partial) secret key. We show that knowledge encryption can be built solely on any two-round oblivious transfer with game-based security, which are known based on various standard (polynomial-hardness) assumptions, such as the DDH, the Quadratic(N^{th}) Residuosity or the LWE assumption.

We use knowledge encryption to construct the *first three-round* (weakly) simulatable oblivious transfer. This protocol satisfies (fully) simulatable security for the receiver, and weakly simulatable security ((T, ϵ)-simulatability) for the sender in the following sense: for any polynomial T and any inverse polynomial ϵ, there exists an efficient simulator such that the distinguishing gap of any distinguisher of size less than T is at most ϵ.

Equipped with these tools, we construct a variety of fundamental cryptographic protocols with low round-complexity, *assuming only the existence of two-round oblivious transfer with game-based security*. These protocols include three-round delayed-input weak zero knowledge argument, three-round weakly secure two-party computation, three-round concurrent weak zero knowledge in the BPK model, and a *two-round* commitment with weak security under selective opening attack. These results improve upon the assumptions required by the previous constructions. Furthermore, all our protocols enjoy the above (T, ϵ)-simulatability (stronger than the distinguisher-dependent simulatability), and are quasi-polynomial time simulatable under the same (polynomial hardness) assumption.

1 Introduction

We study the problem of constructing generic public-key encryption with a natural property that the public key can be reduced to its ciphertexts, i.e., any algorithm that breaks the ciphertext indistinguishability can be used to extract the (partial) secret key. We call such a public-key encryption scheme *knowledge encryption*. Although we often have the impression of public key encryption that only the one holding the secret key can decrypt/distinguish a ciphertext, almost none of known constructions *provably* achieves this property. Instead, they only guarantee that, if an algorithm can break the ciphertext indistinguishability, then we can use it to find a solution to a random instance

of certain hard problem (rather than finding the corresponding secret key). The only exception we aware of is the public-key encryption based on Rabin's trapdoor permutations, for which one can establish the equivalence between breaking the ciphertext indistinguishability and finding a secret key.

Essentially, the decryption of a knowledge encryption scheme can be viewed as a *proof of knowledge* of the (partial) secret key. From this prospective, the concepts of conditional disclosure of secret (CDS) [GIKM98, AIR01, AJ17] and witness encryption (WE) [GGSW13] in the literature are close to our knowledge encryption. Specifically, a public key of a CDS (WE) scheme is generated from a publicly known instance x (for WE, x serves as the pubic key) of an NP language L, and guarantees that if $x \notin L$, then the receiver obtains nothing about the encrypted message.

But the decryption of CDS/WE schemes provides *only* a *sound* proof that the corresponding public key is valid (i.e., $x \in L$), rather than *proof of knowledge* (or, *extractability*) of the witness of $x \in L$. Goldwasser et al. [GKP+13] put forward the notion of *extractable* witness encryption, which, similar in spirit to our knowledge encryption, requires that any algorithm that breaks the ciphertext indistinguishability can be used to extract the witness for the instance x. However, their scheme requires rather strong (unfalsifiable) knowledge assumptions.

Motivation. Our study is motivated by the recent breakthrough [JKKR17, BKP19, Den20] on cryptographic protocols with low round-complexity beyond the known black-box barriers. At a very high level, the idea of behind these constructions is to design a protocol in such a way that any distinguisher with relatively large distinguishing advantage (inverse polynomial) ϵ can be used to extract certain secret of the adversary, which can be used for a successful simulation (except with probability ϵ). Thus, for a given distinguisher, the simulator now can first exploit the power of it to extract some secret information from the adversary and then simulate in a straightforward manner. This distinguisher-dependent simulation technique was introduced by Jain et al. in [JKKR17] and used to achieve delayed-input weak zero knowledge argument and weakly secure two-party computation for certain functionalities in three round, which bypass the well-known lower bounds on the round-complexity [GK96b] and are round-optimal under polynomially hard falsifiable assumptions while black-box reduction/simulation are used to prove the soundness/security for receiver [Kiy21]. Bitansky et al. [BKP19] introduced an ingenious homomorphic trapdoor simulation paradigm and presented a three-round weak zero knowledge argument, without requiring "delayed-input" or the simulator to work in distributional setting. Latter, the distinguisher-dependent simulation was also used to achieve oblivious transfer (OT) in three round with distinguisher-dependent simulatable security for the sender [GJJM20].

Deng [Den20] introduced an individual simulation technique and exploited a variant of Rabin encryption (the only known "knowledge encryption") to realize the above-mentioned design idea. The work of [Den20] proposed a two-round commitment satisfying (T, ϵ)-simulatable security under selective opening attack and a three-round concurrent (T, ϵ)-zero knowledge argument in the bare public-key model (both bypassing the black-box lowerbounds [Xia11, Xia13, APV05]), where the (T, ϵ)-simulatability is defined as follows: For any polynomial T and any inverse polynomial ϵ, there exists a simulator such that the distinguishing gap of any distinguisher of size less than

T is at most ϵ. Note that the (T, ϵ)-simulatability is stronger[3] than the distinguisher-dependent simulatability since it depends only on *the size of the distinguisher* (*not* on the distinguisher per se).

All above protocols require specific number-theoretic assumptions. This state of the art leaves the several intriguing questions:

> *Can we construct oblivious transfer in three-round that achieves simulatable security for both sides? Can we base the above protocols on more general assumptions?*

1.1 Our Contribution

We introduce the notion of knowledge encryption. Like CDS, a knowledge encryption scheme is associated with an NP language L, and the public/secret key pair $(\mathsf{pk}, \mathsf{sk})$ is generated from an instance $x \in L$ and its witness w. We let the public key (secret key) contain the instance x (witness w, respectively). We require the following properties from a knowledge encryption scheme:

1. Indistinguishability: ciphertext indistinguishability holds for any $(x, w) \in R_L$;
2. Witness extractability: for any algorithm that can break the ciphertext indistinguishability can be used to extract the witness w (part of the secret key). This holds even when the public key is maliciously generated.
3. Public key simulation: for any $(x, w) \in R_L$, there is a simulator that, taking only x as input, can output a public key that is indistinguishable from the honestly generated one.

We show that knowledge encryption can be built solely on any two-round OT with game-based security, which are known based on various standard (polynomial-hardness) assumptions, such as the DDH [NP01], the Quadratic(Nth) Residuosity [HK12] or the LWE assumption [BD18].

Equipped with knowledge encryption, we obtain the following results assuming *only the existence of two-round OT with game-based security (against polynomial-time adversaries)*:

- **The *first* three-round (T, ϵ)-simulatable OT** with fully simulatable security for the receiver and (T, ϵ)-simulatable security for the sender.
  Achieving polynomially simulatable security (of any kind) for *both parties* of OT in three rounds has been an elusive. Previous work on three-round OT achieves either *one-sided* (distinguisher-dependent) simulatability for the sender [GJJM20], or *game-based* security for both parties [CCG+21].
- **A variety of protocols achieving (T, ϵ)-simulatable security**, including three-round delayed-input (T, ϵ)-zero knowledge argument, three-round (T, ϵ)-secure two-party computation for independent-input functionalities, three-round concurrent (T, ϵ)-zero knowledge in the BPK model and *two-round* commitment with (T, ϵ)-security under selective opening attack.

[3] Note that the result of [CLP15] that distinguisher-dependent simulatability can be upgraded to (T, ϵ)-simulatability holds only for *zero knowledge* protocols.

Prior works on these protocols either require an additional assumption–the existence of dense encryption, or are only known based on the Factoring assumption [Den20]. The three-round protocol of secure two-party computation in [AJ17] is built on a rather strong assumptions of the existence of succinct randomized encodings scheme, which are only known based on indistinguishable obfuscation. Furthermore, as mentioned before, the (T, ϵ)-simulatability we achieve is stronger than the notion of distinguisher-dependent simulatability achieved by the work of [JKKR17].

Our result on weak zero knowledge is incomparable to the work of [BKP19]: The protocol in [BKP19] requires both LWE and Factoring (or standard Bilinear-Group) assumptions, but the common input need not to be delayed to the last round.

- **Quasi-polynomial time simulatable under polynomial hardness assumption**: All above protocols are quasi-polynomial time simulatable under the same (polynomial hardness) assumption.

  Previous results achieving quasi-polynomial time simulatable security (e.g., see [Pas03] and [KKS18]) usually require quasipolynomial/exponential hardness assumption.

1.2 Technique Overview

Knowledge encryption. Before describing our construction, we briefly recall the idea behind a CDS scheme for an NP relation R_L. Given input $(x, w) \in R_L$ of length $\lambda + \ell$, the receiver uses the algorithm OT_1 to encode w bit-by-bit, and publishes his public key $(x, \mathsf{OT}_1(w_1), \mathsf{OT}_1(w_2) \cdots, \mathsf{OT}_1(w_\ell))$; to encrypt a bit $m \in \{0, 1\}$, the sender first garbles the following circuit C: on input (x, w, m), C checks if $(x, w) \in R_L$, if so, outputs m; otherwise outputs $\perp$. After obtaining a garbled circuit $\hat{C}$ and the associated labels $\{\mathsf{lab}_{i,b}\}_{i\in[\lambda+\ell+1],b\in\{0,1\}}$, the sender sends the ciphertext $c := (\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[\lambda]}, \{\mathsf{OT}_2(\mathsf{lab}^w_{i,0}, \mathsf{lab}^w_{i,1})\}_{i\in[\ell]}, \mathsf{lab}^m_m)$ to the receiver, which retrieves the labels $\{\mathsf{lab}^w_{i,w_i}\}_{i\in[\ell]}$ and then decrypts c using the evaluating algorithm of the garbling scheme.

To achieve the *witness extractability* property, our key idea is to embed a simple decoding mechanism in the above circuit C, which enables us to reduce the instance x to random ciphertexts. Specifically, we let C to take an extra input y of length ℓ and define it as follows: on input $((x, w, y, m)$, if $(x, w) \in R_L$ and $y = 0^\ell$, output m; if $(x, w) \in R_L$ and the Hamming weight of $\|y\|_1 \geq 1$, output $\Sigma^\ell_{i=1} y_i w_i \bmod 2$; if $(x, w) \notin R_L$, output $\perp$. With this modification, when encrypting a bit m, the honest sender always chooses $y = 0^\ell$, garbles the above circuit C and then sets the ciphertext to be $c := (\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[\lambda]}, \{\mathsf{OT}_2(\mathsf{lab}^w_{i,0}, \mathsf{lab}^w_{i,1})\}_{i\in[\lambda]}, \{\mathsf{lab}^y_{i,0}\}_{i\in[\ell]}, \mathsf{lab}^m_m)$.

It is not hard to see that this modification does not affect the *indistinguishability* of the scheme. On the other hand, the *witness extractability* property follows from the following observations. Note first that, for every $i \in [\ell]$, one can always choose a bad y which has 1 on the i-th coordinate and zero on all others, and compute a ciphertext with such a y. Due to the security of the underlying garbling scheme, no polynomial size circuit can distinguish these bad ciphertexts from the honestly-generated ones. Thus, for any polynomial size circuit that decrypts honestly-generated ciphertexts correctly with high probability, when given a bad ciphertext as input, it would output $\Sigma^\ell_{i=1} y_i w_i \bmod 2 = w_i$ correctly with almost the same probability. One can apply this reasoning to ciphertext distinguishers and prove the witness extractability property.

Nearly optimal (T, ϵ)-extractor for knowledge encryption. Applying the result of [Den20], we will have a nearly optimal (T, ϵ)-extractor for any (possibly malicious) key generation algorithm of knowledge encryption in the following sense: for any polynomial T and any inverse polynomial ϵ, the extractor outperforms any circuits of size T in extracting the witness for x in the public key except for probability ϵ.

Looking ahead, the (T, ϵ)-simulatability of all our protocols relies on this nearly optimal extractor. When receiving the public key(s) of knowledge encryption from an adversary, the corresponding simulator will run this extractor to extract the witness for x, and if it succeeds, then the simulation can be done; if it fails, then the optimality of the extractor guarantees that no other circuits (distinguishers) of size T can extract the witness either (except for small probability ϵ) , and thus the simulator can encrypt a dummy message in its last round, which cannot be told apart from an real execution by any distinguishers of size T except for probability ϵ (by the witness extractability of knowledge encryption.)

Three-round OT with (T, ϵ)-simulatability for both parties. A natural idea here is to have the receiver generate a pair of public keys $\mathsf{pk}_0, \mathsf{pk}_1$ of knowledge encryption from two NP instances x_0 and x_1, for one of which it knows a valid witness so that it can receive one message encrypted by the sender. However, there are two challenges that arise from this approach:

1 We need to make sure that the receiver knows a witness for *only one* of these two instances (to achieve the sender security), while at the same time one needs to know both witnesses for x_0 and x_1 to extract the two messages from the sender in the proof of receiver security.
2 There is no way for the receiver to tell honest ciphertexts from "bad" ones.

One may think of the following solution to the first challenge: the sender generates some hard instance y (and prove to the receiver that it knows a witness for y in three rounds), and then the receiver proves that it knows either a witness for y or only one of x_0 and x_1 is in the language L (for some suitable language) in a two-round WI protocol. However, among other issues, there is no known two-round WI protocol based on two-round OT.

To this end, we have the sender generate two images y_0 and y_1 of a one-way function f and prove to the receiver that it knows one pre-image of y_0 or y_1 via a three-round WI protocol[4]. Given the pair (y_0, y_1) and input b, the receiver prepares two instances x_0 and x_1 in the following way: it runs the HVZK simulator of the Σ-protocol to obtain an acceptable proof (a, b, z) of knowledge of one preimage of y_0 or y_1, and sets $x_b = (y_0, y_1, a, b)$ and $x_{1-b} = (y_0, y_1, a, 1-b)$, where $x_i = (y_0, y_1, a, i)$ is said to be a YES instance if and only if there exists a z such that (a, i, z) is acceptable. The receiver now generates pk_b honestly using the valid witness z for $x_b = (y_0, y_1, a, b)$, and runs the key simulator of knowledge encryption to obtain the other public key pk_{1-b}. In the

[4] Note that the three-round WI and the Σ-protocol used in our construction can be based on non-interactive commitment. As noted in [CCG+20], combing the recent work of [LS19] with the work [GKM+00], one can build non-interactive commitment from two-round (perfectly correct) OT with game-based security as defined in Definition 6. Thus, two-round OT with game-based security as we define is sufficient for constructing all primitives used in our protocol.

third round, the sender encrypt its two message under the two public keys respectively and send the two ciphertexts to the receiver.

Notice that the receiver does not know a witness for the instance x_{1-b} on the public key pk_{1-b}, since otherwise it would be able to compute a preimage of y_0 or y_1 generated by the sender at random (which is infeasible due to the fact that the WI proof actually hides the two preimages of y_0 or y_1.) This observation, together with the existence of nearly optimal extractor (as mentioned above) that outperforms any other circuits of a-priori bounded size for extracting a witness of x_0 or x_1, one can prove the (T, ϵ)-simulatable security for the sender.

Our proof of the (fully) simulatable security for the receiver departs from the traditional proof strategy that is usually done by extracting the sender's two messages from a WI proof of knowledge. Our simulator extracts the sender's two messages by decryption. Using rewinding strategy[5] the simulator extracts a preimage of y_0 and y_1, then generates two Yes instance x_0 and x_1 and two valid public keys. When receiving the two ciphertexts from the sender, it can decrypt to obtain both messages[6] and send them to the functionality. Note that, although these ciphertexts from the sender may be generated maliciously (as mentioned in the above second challenge) and adaptively (depending on the receiver's public keys), we can still prove the simulatable security for the receiver since the public keys of the receiver in the real model execution and the ones in the ideal model execution are indistinguishable.

(T, ϵ)-zero knowledge and (T, ϵ)-secure two-party computation. At a high level, our construction of (T, ϵ)-zero knowledge protocol follows the paradigm of [ABOR00, KR09]. The prover and the verifier execute a three-round OT as constructed above (denoted by $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ the three OT step algorithms respectively), where the verifier plays the role of the receiver and chooses a random bit $\beta \leftarrow \{0, 1\}$ as the receiver's input in the second round. In the last round of OT, the prover prepares two acceptable Σ-proofs $(\alpha, 0, \gamma_0), (\alpha, 1, \gamma_1)$ for the statement $x \in L$, and sends x and $(\alpha, \mathsf{OT}_3(\gamma_0, \gamma_1))$ to the verifier. Finally, the verifier recovers γ_β from OT and checks whether $(\alpha, \beta, \gamma_\beta)$ is an acceptable proof. In order to reduce the soundness error, we have the prover and the verifier run this protocol λ times in parallel. The (T, ϵ)-zero knowledge of the protocol essentially follows from the (T, ϵ)-simulatable security for sender of the underlying OT and the fact that the nearly optimal extractor guaranteed by Lemma 2 works well for (possibly malicious) parallelized key generator of knowledge encryption.

One can also prove a sort of soundness of the above protocol due to the simulatable security for receiver of the underlying OT. However, we do not know how to show it satisfies *adaptive* soundness/argument of knowledge, which is naturally required in settings where the prover can choose statements to be proven adaptively. Inspired by [JKKR17], we use additional knowledge encryption schemes to achieve *adaptive* argument of knowledge. In addition to executing the above protocol, the prover generates two public keys of knowledge encryption and proves to the verifier that one of them is generated honestly in a three-round WI protocol. In the last round, it encrypts each of γ_0 and γ_1 twice under the two public keys, and sends these encryptions along with

[5] Here we actually need Goldriech-Kahan technique to bound the running time of the extractor, see the detailed proof in Section 4.

[6] If the simulator fails to decrypt a ciphertext, it sets the corresponding "plaintext" to be $\perp$.

the third OT messages (which now encode both (γ_0, γ_1) and the *randomnesses* used in these encryptions). We observe that these additional encryptions does not harm zero knowledge property of the above protocol since the WI proof for the sender's two public keys actually hides both secret keys. On the other hand, it does help us achieve *adaptive* argument of knowledge: One can extract a secret key by rewinding the prover and decrypt those encryptions in the original transcript obtained before rewinding, which will reveal a witness for the statement in that transcript.

Equipped with the above three-round OT and weak zero knowledge argument, we follow the GMW paradigm [GMW87] to give a three-round protocol for (T, ϵ)-secure two-party computation for independent-input functionalities. We stress that the (T, ϵ)-simulatable security against malicious receiver of our two-party computation protocol only holds for *independent-input functionalities*, since for the proof of (T, ϵ)-simulatability against malicious receiver to go through, we need to make sure that one can freely sample the sender's input x even when the malicious receiver's input y is fixed. This is roughly also the reason that we achieve (T, ϵ)-zero knowledge only for *delayed-input* argument.

Our protocols of commitment with weak security under selective opening attack and concurrent weak zero knowledge argument (in the BPK model) simply follows by replacing the corresponding encryption scheme in the constructions of [Den20] with our knowledge encryption (and revising their protocol accordingly so that the simulation can go through with a witness for the instance on the public key of knowledge encryption). Furthermore, when using our construction of (T, ϵ)-zero knowledge argument of knowledge in the extractable commitment of [JKKR17], we obtain a three-round extractable commitment from two-round OT with game-based security.

1.3 More Related Work

Related work on simulatable Oblivious transfer. The work of [ORS15, FMV19, CCG+21] achieved fully-simulatable black-box construction of OT in four-round from certified/full domain trapdoor permutations or strongly uniform key agreement protocol, which are also round optimal for black-box constructions [KO04]. In the common reference string model, fully-simulatable secure (even UC-secure) OT can be achieved in two rounds from various assumptions [PVW08, DGH+20], such as DDH, LWE, CDH or LPN assumptions.

Related work on two/multi-party computation. Katz and Ostrovsky [KO04] showed that four-round is necessary for black-box two-party computation for general functionalities where only one party receives the output. The construction of four-round black-box two-party computation was constructed in [ORS15, COSV17]. Garg et. al [GMPP16] study two-party computations with simultaneous message transmission and give a four-round construction for general functionalities where both parties receive the output. Four-round secure multi-party computation can be constructed from various assumptions [BGJ+18, HHPV18]. Recently, Choudhuri et. al [CCG+20] constructed a four-round construction only from four-round fully-simulatable OT. In the CRS model, Benhamouda and Lin [BL18] and Garg and Srinivasan [GS18] presented

the two-round constructions from two-round semi-malicious OT protocol and NIZK or two-round fully-simulatable OT respectively.

2 Preliminaries

Throughout this paper, we let λ denote the security parameter. Given a positive integer m, a and b, we denote by $[m]$ the set $\{1, 2, \cdots, m\}$, and by $[a, b]$ the set $\{a, a+1, \cdots, b\}$. We often write a string x as a concatenation of its bits, $x = x_1 \| x_2 \| \cdots \| x_n$, where x_i is the i-th bit of x. For a given y, we denote by $\|y\|_1$ the Hamming weight of y. We use the standard abbreviation PPT to denote probabilistic polynomial time. We will use the terms (non-uniform) PPT algorithm and polynomial-size circuits interchangeably. When writing a polynomial-size circuit C, we mean a polynomial-size family of circuits $C = \{C_\lambda\}_{\lambda \in \mathbb{N}}$. For two random ensembles $\mathcal{X} := \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{Y} := \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$, we write $\mathcal{X} \overset{c}{\approx} \mathcal{Y}$ to mean $\mathcal{X} := \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{Y} := \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$ are indistinguishable against all polynomial-size circuits.

2.1 Interactive Argument

Let L be an NP language and R_L be its associated relation. For a given $x \in L$, we use $R_L(x)$ to denote the set of valid witnesses to x. An interactive argument (P, V) for L is a pair of PPT algorithms (called the prover and the verifier), in which the prover P wants to convince the verifier V of a statement $x \in L$. For a given $(x, w) \in R_L$, we denote by $\mathsf{Out}_V(P(w), V)(x)$ the output of V at the end of an execution of (P, V), and by $\mathsf{View}_V^{P(w)}(x)$ the view of V in an interaction.

Definition 1. (Argument) *A protocol (P, V) for an NP language L is an argument if the following two conditions hold:*

- **Completeness***: For any $x \in L$ and $w \in R_L(x)$, $\mathsf{Out}_V(P(w), V)(x) = 1$.*
- **Computational soundness***: For any polynomial-size prover P^*, there exists a negligible function $negl(\cdot)$ such that for any $x \notin L$ of length λ,*

$$\Pr[\mathsf{Out}_V(P^*, V)(x) = 1] < negl(\lambda).$$

Additionally, an interactive argument system is called public-coin if at every verifier step, the verifier sends only truly random messages.

Delayed-input and adaptive computational soundness. We call an argument is *delayed-input* if the statement x is sent to verifier only in the last round. Note that delayed-input argument system would enable a cheating prover to choose a false statement adaptively (depending on the interaction history) to fool the verifier. We consider such an adaptive cheating prover and define adaptive computational soundness in a natural way: A delayed-input argument is called *adaptive computational sound* if its computational soundness condition holds even against adaptive cheating prover.

Argument of knowledge and adaptive argument of knowledge. The adaptive argument of knowledge property is defined in similar way to the argument of knowledge,

except that here we need to deal with the issue that the statement may be chosen adaptively. We follow the definition in [BCPR14, BBK+16] to define three-round adaptive argument of knowledge.

Definition 2. *A three-round delayed-input argument system with message (a_1, a_2, a_3) for NP language L is called an adaptive argument of knowledge if there exists an oracle extractor E and a polynomial poly such that for any PPT malicious prover P^*, any noticeable function ϵ and any security parameter $\lambda \in \mathbb{N}$:*

$$\textit{if} \quad \Pr\left[V(x,(a_1,a_2,a_3))=1 \;\middle|\; \begin{array}{r} a_1 \leftarrow P^* \\ a_2 \leftarrow V(\lambda,a_1) \\ x, a_3 \leftarrow P^*(a_1,a_2) \end{array}\right] \geq \epsilon(\lambda),$$

$$\textit{then} \quad \Pr\left[\begin{array}{r} V(x,(a_1,a_2,a_3))=1 \wedge \\ E^{P^*}(x,(a_1,a_2,a_3)) \notin R_L(x) \end{array} \;\middle|\; \begin{array}{r} a_1 \leftarrow P^* \\ a_2 \leftarrow V(\lambda,a_1) \\ x, a_3 \leftarrow P^*(a_1,a_2) \end{array}\right] \leq negl(\lambda),$$

where E runs in expected time bounded by $poly(\lambda)/\epsilon$.

An argument system is zero knowledge [GMR89] if the view of the (even malicious) verifier in an interaction can be efficiently reconstructed. We consider a weak version of zero-knowledge as defined in [Den20, CLP15], (T,ϵ)-zero-knowledge, which relaxes the definition of zero-knowledge and requires that, for any polynomial T and inverse polynomial ϵ, there exists an efficient simulator such that the distinguishing gap of any T-size distinguisher is at most ϵ.

Definition 3. **((T,ϵ)-Zero-Knowledge)** *An argument (P,V) is (T,ϵ)-zero-knowledge if for any polynomial-size malicious verifier V^*, any polynomial T and any inverse polynomial ϵ, there exists a polynomial-size simulator $S = \{S_\lambda\}_{\lambda\in\mathbb{N}}$ such that for any T-size distinguisher $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$, and any statement $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(x)$:*

$$\left|\Pr\left[D_\lambda(\mathsf{View}_{V^*}^{P(w)}(x))=1\right] - \Pr\left[D_\lambda(S_\lambda(x))=1\right]\right| < \epsilon(\lambda).$$

Definition 4. **(Witness Indistinguishability)** *An argument (P,V) for an NP language L is witness indistinguishable if for any polynomial-size V^*, any $\{(x,w_0,w_1)\}_{x\in L}$ such that both (x,w_0) and $(x,w_1) \in R_L$, it holds that*

$$\mathsf{View}_{V^*}^{P(w_0)}(x) \overset{c}{\approx} \mathsf{View}_{V^*}^{P(w_1)}(x).$$

Special soundness. We will use three-round public-coin witness indistinguishable argument of the form $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ as a building block with *special soundness*: There exists a PPT algorithm, such that on input two accepting proofs $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ and $(\mathsf{WI}_1, \mathsf{WI}_2', \mathsf{WI}_3')$ for x with $(\mathsf{WI}_2 \neq \mathsf{WI}_2')$, it outputs $w \in R_L(x)$.

Definition 5. **(Σ-protocols)** *A three-round public-coin protocol (P,V) for an NP language L is called a Σ- protocol if the following conditions hold:*

- **Completeness**: *For any $x \in L$ and $w \in R_L(x)$, $\mathsf{Out}_V(P(w), V)(x) = 1$.*
- **Special soundness**: *There exists a PPT algorithm which, given any instance $x \in L$ and two acceptable transcripts (a, e, z) and (a, e', z') with $e \neq e'$, computes a witness w s.t. $(x, w) \in R$.*
- **Special honest verifier zero knowledge (HVZK)**: *There exists a PPT algorithm Sim which, taking $x \in L$ and a random challenge e as inputs, outputs (a, z) such that the tuple (a, e, z) is indistinguishable from an acceptable transcript generated by a real protocol run between the honest prover and verifier.*

Constructions. Three-round public-coin WI arguments with special soundness and three-round Σ-protocols can be constructed from non-interactive commitment [Blu86].

2.2 Oblivious Transfer

A 1-out-of-2 oblivious transfer protocol (OT) (S, R) is a two-party protocol between a sender S and a receiver R. The sender S has input of two strings (m_0, m_1) and the receiver R has input a bit b. At the end of the protocol, the receiver R learns m_b (and nothing beyond that), whereas the sender S learns nothing about b. We denote the output of receiver $\mathsf{Out}_R(S(m_0, m_1), R(b))(1^\lambda)$.

There are two notable security definitions in the literature, the game-based security [NP01, AIR01] and the simulation-based security [Gol04].

Game-based security. Following [HK12, AJ17, BD18], we give a formal game-based security definition for two-round OT.

Definition 6. (Oblivious Transfer with Game-based Security) *A game-based secure two-round oblivious transfer (S, R) satisfies the following properties:*

- **Perfect Correctness**: *For any $\lambda \in \mathbb{N}$ and $m_0, m_1 \in \{0,1\}^n, b \in \{0,1\}$,*

$$\Pr[m_b = \mathsf{Out}_R\langle S(m_0, m_1), R(b)\rangle(1^\lambda)] = 1$$

- **Receiver Security**: *Denote by $R(1^\lambda, b)$ the distribution over the message sent by the honest receiver on input $(1^\lambda, b)$. Then we have:*

$$\{(R(1^\lambda, 0))\} \stackrel{c}{\approx} \{(R(1^\lambda, 1))\}$$

- **Sender Security**: *Denote by $S(1^\lambda, m_0, m_1, \mathsf{ot}_1)$ the distribution over the response of the honest sender on input $(1^\lambda, m_0, m_1)$ and the (possibly malicious) receiver's first message ot_1. Then at least one of the following conditions holds:*
  1. *For any $m_0, m_1, m' \in \{0,1\}^n$:*

$$\{(S(1^\lambda, m_0, m_1, \mathsf{ot}_1))\} \stackrel{c}{\approx} \{(S(1^\lambda, m_0, m', \mathsf{ot}_1))\}.$$

  2. *For any $m_0, m_1, m' \in \{0,1\}^n$:*

$$\{(S(1^\lambda, m_0, m_1, \mathsf{ot}_1))\} \stackrel{c}{\approx} \{(S(1^\lambda, m', m_1, \mathsf{ot}_1))\}.$$

Constructions. Two-round OT satisfying above security can be constructed based on the DDH Assumption [NP01], or the Quadratic(N^{th}) Residuosity Assumption [HK12] or the LWE Assumption [BD18].

Simulation-based security. We follow the standard real/ideal paradigm and define the simulation-based security of OT. Roughly, to prove security in the real/ideal paradigm, one first defines an ideal functionality $\mathcal{F}$ executed by a trusted party, then constructs a simulator Sim that interacts with $\mathcal{F}$ and the adversary, and then shows that the output of Sim is indistinguishable from the real execution.

We let the message space $\mathcal{M}$ to include the special symbol $\perp$, i.e., $\mathcal{M} := \{0,1\}^n \cup \perp$.[7] The ideal functionality of OT is provided in Fig.1.

Functionality $\mathcal{F}_{OT}$

Security parameter: λ

$\mathcal{F}_{OT}$ interacts with a sender S and a receiver R.

- Upon receiving $(\mathsf{send}, m_0, m_1)$ from S, where $m_0, m_1 \in \mathcal{M}$, record m_0, m_1 and then send send to R.
- Upon receiving $(\mathsf{receive}, b)$ from R, send m_b to R and $\mathsf{receive}$ to S and halt.

Fig. 1: The Oblivious Transfer Functionality $\mathcal{F}_{OT}$

We denote by $\mathsf{REAL}_{\Pi,R^*(\tau)}(1^\lambda, m_0, m_1, b)$(resp., $\mathsf{REAL}_{\Pi,S^*(\tau)}(1^\lambda, m_0, m_1, b)$) the distribution of the output of the malicious receiver (resp., the malicious sender and the honest receiver) during a real execution of the protocol Π (with m_0, m_1 as inputs of the sender, b as choice bit of the receiver), and by $\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}^{R^*(\tau)}}(1^\lambda, m_0, m_1, b)$ (resp., $\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}^{S^*(\tau)}}(1^\lambda, m_0, m_1, b)$) the distribution of the output of the malicious receiver (resp., the malicious sender and the honest receiver) during a ideal execution where τ is the auxiliary input.

Definition 7. (Oblivious Transfer with Simulation-based Security) *A protocol $\Pi = (S, R)$ securely computing $\mathcal{F}_{OT}$ if it satisfies the following properties:*

- **Simulatable Security for Receiver***: For any polynomial-size malicious sender S^*, there exists a polynomial-size simulator Sim such that for any auxiliary input $\tau \in \{0,1\}^*$, any $m_0, m_1 \in \{0,1\}^n, b \in \{0,1\}$,*

$$\{\mathsf{REAL}_{\Pi,S^*(\tau)}(1^\lambda, m_0, m_1, b)\} \overset{c}{\approx} \{\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}^{S^*(\tau)}}(1^\lambda, m_0, m_1, b)\}.$$

- **Simulatable Security for Sender***: For any polynomial-size malicious receiver R^*, there exists a polynomial-size simulator Sim such that for any auxiliary input $\tau \in \{0,1\}^*$, any $m_0, m_1 \in \{0,1\}^n, b \in \{0,1\}$,*

$$\{\mathsf{REAL}_{\Pi,R^*(\tau)}(1^\lambda, m_0, m_1, b)\} \overset{c}{\approx} \{\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}^{R^*(\tau)}}(1^\lambda, m_0, m_1, b)\}.$$

[7] Jumping ahead, in the proof of receiver's security of our construction, the simulator may extract (by decryption) two messages like $(m, \perp)$ or $(\perp, \perp)$ from a corrupted sender. In this case, the simulator will not abort, instead, it views $\perp$ as a message and send these two messages to the functionality.

In this paper, we follow the definition of weak simulatability in [Den20, CLP15] and give a definition of simulatable (T, ϵ)-security for sender of an OT protocol (S, R).

Definition 8. **((T, ϵ)-Simulatable Security for Sender)** *For any polynomial-size malicious receiver R^*, any polynomial T, any inverse polynomial ϵ, any auxiliary input distribution $\mathcal{Z}$ and $\tau \leftarrow \mathcal{Z}$, there exists a polynomial-size simulator* Sim *such that for any T-size distinguisher $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$, any $m_0, m_1 \in \{0,1\}^n, b \in \{0,1\}$:*

$$\begin{aligned}\big|\Pr[D_\lambda(\mathsf{REAL}_{\Pi,R^*(\tau)}(1^\lambda, m_0, m_1, b))] = 1 \\ - \Pr[D_\lambda(\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}(\tau)}(1^\lambda, m_0, m_1, b))] = 1\big| \leq \epsilon(\lambda).\end{aligned} \tag{1}$$

Remark 1. Notice that traditional security definitions (such as the definition of sender's security above) require that the *black-box* simulator can deal with *any* auxiliary input τ, while, in our definition of (T, ϵ)-sender's security, we weaken this requirement by switching the order of the qualifiers and require only that for any auxiliary input τ drawn from a (known) distribution, there is a desired *individual* simulator. We make this change for the reason that, in the proof of (T, ϵ)-simulatability for the sender of our OT protocol, the simulator will apply the nearly-optimal extractor (similar to the one in [Den20]) for extracting some secret keys from the malicious receiver, and such an extractor is really sensitive and works well only when *all input distributions* (including the auxiliary input distribution) of the malicious receiver are well defined.

Still, as we will see, this weaker notion also has wide applications in protocol composition. We can plug a protocol Π_i satisfying this weaker security into a global protocol Π composed from a series of subprotocols $\Pi_1, \Pi_2, ..., \Pi_n$, and achieve (T, ϵ)-simulation security of Π, *as long as* all these subprotocols are simulatable and specified in advance[8]. One can view all messages from subprotocols $\Pi_{j\neq i}$ as auxiliary input drawn from the distributions over the transcripts of these subprotocols, which are well defined when we simulate the subprotocol Π_i in the proof of (T, ϵ)-simulatability of Π.

2.3 Secure Two-Party Computation

In this subsection we present the definition of secure two-party computation, independent-input functionalities and the (T, ϵ)-security. Parts of the definition of secure two-party computation are taken verbatim from [AJ17]. In this paper, we only consider the case where only one party (a.k.a receiver R) learns the output. The other party is referred to as the sender S. Sender S has input x and receiver R has input y. For a given deterministic functionality F, they execute a protocol to jointly compute $F(x, y)$, and R obtains $F(x, y)$ at the end of execution. As observed in [KO04], a two-party computation protocol which only one party learns the output can be easily transformed into the one where both parties receive the output by computing a modified functionality that outputs signed values.

We follow the real/ideal paradigm to define the simulation-based security of two-party computation. The ideal model execution proceeds as follows:

Ideal model execution. Ideal model execution is defined as follows.

[8] One exceptional case is the UC composition [Can01], where Π may be composed with arbitrarily unknown protocols.

- *Input*: Each party obtains an input, denoted u ($u = x$ for S and $u = y$ for R).
- *Send inputs to trusted party*: The parties now send their inputs to the trusted party. The honest party always sends u to the trusted party. A malicious party may, however, can send a different input to the trusted party.
- *Aborting Adversaries*: An adversarial party can then send a message $\perp$ to the trusted party to abort the execution. Upon receiving this, the trusted party terminates the ideal world execution. Otherwise, the following steps are executed.
- *Trusted party answers receiver R*: Suppose the trusted party receives inputs (x', y') from S and R respectively. It sends the output $\mathsf{out} = F(x', y')$ to receiver.
- *Outputs*: If the receiver R is honest, then it outputs out. The adversarial party (S or R) outputs its entire view.

We denote the adversary participating in the above protocol to be $\mathcal{B}$ and the auxiliary input to $\mathcal{B}$ is denoted by τ. We define $\mathsf{IDEAL}_{\mathcal{F}_{2pc},\mathcal{B}}$ to be the joint distribution over the outputs of the adversary and the honest party from above ideal execution.

Real model execution. We next consider the real model in which a real two-party protocol is executed (and there exists no trusted third party). In this case, a malicious party may follow an arbitrary feasible strategy. In particular, the malicious party may abort the execution at any time (and when this happens prematurely, the other party is left with no output).

Let Π be a two-party protocol for computing F. Note that in the two-party case at most one of S, R is controlled by an adversary. We denote the adversarial party to be $\mathcal{A}$ and the auxiliary input to $\mathcal{A}$ is denoted by τ. We define $\mathsf{REAL}_{\Pi,\mathcal{A}}$ to be the joint distribution over the outputs of the adversary and the honest party from the real execution.

Definition 9. (Security) *Let F and Π be described above. We say that Π securely computes F if for every polynomial-size malicious adversary $\mathcal{A}$ in the real world, there exists a polynomial-size adversary $\mathcal{B}$ for the ideal model, such that for any auxiliary input $\tau \in \{0,1\}^*$.*

$$\{\mathsf{REAL}_{\Pi,A(\tau)}(1^\lambda, x, y)\} \overset{c}{\approx} \{\mathsf{IDEAL}_{\mathcal{F}_{2pc},B(\tau)}(1^\lambda, x, y)\}.$$

In this paper, we only consider independent-input functionalities, as defined [JKKR17].

Definition 10. (Independent-Input Functionalities) *An independent-input functionality is defined as a functionality between two parties, Alice and Bob. Let $(\mathcal{Q}, \mathcal{R}, \mathcal{U})$ denote the joint distribution over inputs of both parties, where Alice's input is sampled efficiently from $\mathcal{Q}$ and Bob's input is sampled efficiently from distribution $\mathcal{R}$, and $\mathcal{U}$ denotes their common public input. Then, a functionality F over $(\mathcal{X} = (\mathcal{Q}, \mathcal{U}) \times \mathcal{Y} = (\mathcal{R}, \mathcal{U}))$ is independent-input for Alice if $\mathcal{Q}$ is independent of $(\mathcal{R}, \mathcal{U})$.*

Similar to (T, ϵ)-zero knowledge, we define (T, ϵ)-security for a protocol of two-party computation as follows.

Definition 11. ((T, ϵ)-Security) *Let F and Π be described above. We say Π computes F with (T, ϵ)-security if for any polynomial-size malicious adversary $\mathcal{A}$ in the real*

model, any polynomial T, any inverse polynomial ϵ, and any auxiliary input distribution $\mathcal{Z}$, there exists a polynomial-size adversary $\mathcal{B}$ in the ideal model, such that for any T-size distinguisher $D := \{D_\lambda\}_{\lambda\in\mathbb{N}}$,

$$\Big|\Pr[D_\lambda(\mathsf{REAL}_{\Pi,A(\tau)}(1^\lambda,x,y))]=1 \\ -\Pr[D_\lambda(\mathsf{IDEAL}_{\mathcal{F}_{2pc},B(\tau)}(1^\lambda,x,y))]=1\Big| \le \epsilon(\lambda).$$

where the probabilities is over the coin of joining parties and $\tau \leftarrow \mathcal{Z}$.

2.4 Garbled Circuits

Garbled circuits was introduced by Yao [Yao86] as a key tool for two-party computation. We follow the definition of [AJ17, GMPP16] and refer the reader to [BHR12] for a comprehensive treatment.

Definition 12. (Garbling Scheme) *A garbling scheme for circuits is a tuple of PPT algorithms* $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$*:*

- $\mathsf{Garble}(1^\lambda, C)$*: On input a security parameter λ and a circuit $C : \{0,1\}^n \to \{0,1\}^m$,* Garble *outputs a garbled circuit $\hat{C}$ along with labels $\{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}$.*
- $\mathsf{Eval}(\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]})$*: On input the garbled circuit $\hat{C}$ and the labels $\{\mathsf{lab}_{i,x_i}\}_{i\in[n]}$,* Eval *outputs $y \in \{0,1\}^m$.*

It satisfies the following two properties:

- **Correctness***: For any security parameter $\lambda \in \mathbb{N}$, any circuit C and any input $x = x_1||\cdots||x_n$, we have that*

  $$\Pr[\mathsf{Eval}(\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}) = C(x)] = 1$$

  where $(\hat{C}, \{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C)$.
- **Security***: There exists a PPT simulator* Sim *such that for any circuit C and any input $x = x_1||\cdots||x_n$:*

  $$\{(\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]})\} \overset{c}{\approx} \{\mathsf{Sim}(1^\lambda, \phi(C), C(x))\}$$

  where $(\hat{C}, \{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C)$ and $\phi(C)$ is the topology of C.

Constructions. A secure garbling scheme can be constructed from any one-way functions[Yao86, LP09].

2.5 Random Self-Reducible Encryption

Random self-reducible encryption was defined in [BKP19]. Loosely speaking, one can rerandomize a ciphertext and obtain a random ciphertext of the same message under the same public key, and furthermore, given an oracle access to any good (i.e., with noticeable distinguishing advantage) distinguisher, one can decrypt with high probability.

Definition 13. (Random Self-Reducible Encryption Scheme) *A random self-reducible encryption scheme consists of four PPT algorithms* $(\mathsf{RSR.Gen}, \mathsf{RSR.Enc}, \mathsf{RSR.Dec}, \mathsf{RSR.Dec}^*)$ *and satisfies following properties:*

- **Correctness***: For any* $b \in \{0,1\}, \lambda \in \mathbb{N}$

$$\Pr\left[\mathsf{RSR.Dec}(sk, ct) = b \,\middle|\, \begin{array}{r}(pk, sk) \leftarrow \mathsf{RSR.Gen}(1^\lambda) \\ ct \leftarrow \mathsf{RSR.Enc}(pk, b)\end{array}\right] = 1$$

- **Indistinguishability***: For any polynomial-size distinguisher* $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $negl$*,*

$$\Pr\left[D_\lambda(pk, c_b) = b \,\middle|\, \begin{array}{r}(pk, sk) \leftarrow \mathsf{RSR.Gen}(1^\lambda) \\ b \leftarrow \{0,1\};\ c_b \leftarrow \mathsf{RSR.Enc}(pk, b)\end{array}\right] < \frac{1}{2} + negl(\lambda)$$

- **Random self-reducible***: For any public key* $pk \in \mathsf{RSR.Gen}(1^\lambda)$*, any distinguisher* $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$ *and any inverse polynomial* ϵ*, if*

$$|\Pr[D_\lambda(\mathsf{RSR.Enc}(pk, 0)) = 1] - \Pr[D_\lambda(\mathsf{RSR.Enc}(pk, 1)) = 1]| \geq \epsilon$$

then for any $b \in \{0,1\}$*,*$ct \in \mathsf{RSR.Enc}(pk, b)$

$$\Pr[\mathsf{RSR.Dec}^{*D_\lambda}(ct, pk, 1^{1/\epsilon}) = b] = 1 - negl(\lambda)$$

where the size of $\mathsf{RSR.Dec}^*$ *is polynomial in* ϵ *and* λ*.*

Relaxed RSR. Bitansky et al. [BKP19] also introduced a relaxed version of RSR encryption, which relaxes the third property of RSR encryption and requires that the ciphertexts $ct \in \mathsf{RSR.Enc}(pk, b)$ can be reduced to ciphertexts with respect to a different encryption algorithm $\mathsf{RSR.Enc}^*$.

Formally, the **relaxed random self-reducibility** is defined as follows. There exists an additional PPT algorithm $\mathsf{RSR.Enc}^*$ such that for any public key $pk \in \mathsf{RSR.Gen}(1^\lambda)$ it holds that for any (possibly probabilistic) distinguisher D and inverse polynomial ϵ, if

$$\left|\Pr[D_\lambda(\mathsf{RSR.Enc}^*(pk, 0)) = 1] - \Pr[D_\lambda(\mathsf{RSR.Enc}^*(pk, 1)) = 1]\right| \geq \epsilon,$$

then for any $b \in \{0,1\}$ and $ct \in \mathsf{RSR.Enc}(pk, b)$,

$$\Pr[\mathsf{RSR.Dec}^{*D_\lambda}(ct, pk, 1^{1/\epsilon}) = b] = 1 - negl(\lambda).$$

Constructions. Random self-reducible encryption scheme can be constructed based on several standard algebraic assumptions [GM84, Gam85, Pai99]. The following relaxed RSR encryption scheme can be constructed from LWE[BKP19], which is already sufficient for our results.

2.6 Conditional Disclosure of Secrets

Conditional disclosure of secrets (CDS)[AIR01, BP12, AJ17] can be seen as an interactive version of witness encryption [GGSW13]. For an NP langauge L, the receiver of CDS scheme holds $(x, w) \in R_L$ and generates a key pair (pk, sk), and the sender encrypts a message m under the public key pk and x and produces a ciphertext c. The receiver can decrypt this ciphertext only if it holds a valid witness w; if $x \notin L$ then no one can tell apart ciphertexts of any two equal length messages.

Definition 14. (Conditional Disclosure of Secrets) *A conditional disclosure of secrets scheme* $(\mathsf{CDS.Gen}, \mathsf{CDS.Enc}, \mathsf{CDS.Dec})$ *for an NP language* L *satisfies following properties:*

- **Correctness***: For any security parameter* $\lambda \in \mathbb{N}$*, statement* $x \in L \cap \{0,1\}^\lambda, w \in R_L(x)$ *and* $m \in \{0,1\}^*$

$$\Pr\left[\mathsf{CDS.Dec}(sk, c) = m \;\middle|\; \begin{array}{r}(pk, sk) \leftarrow \mathsf{CDS.Gen}(1^\lambda, x, w)\\ c \leftarrow \mathsf{CDS.Enc}(pk, x, m)\end{array}\right] = 1$$

- **Message indistinguishability***: For any polynomial-sized distinguisher* $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $negl$ *such that for any* $x \in \{0,1\}^\lambda \backslash L$*,* pk^* *and two equal-length* m_0, m_1*,*

$$\Pr\left[D_\lambda(pk^*, x, c_b) = b \;\middle|\; \begin{array}{r}b \leftarrow \{0,1\}\\ c_b \leftarrow \mathsf{CDS.Enc}(pk^*, x, m_b)\end{array}\right] \leq \frac{1}{2} + negl(\lambda)$$

- **Receiver simulation***: There exists a PPT simulator* $\mathsf{CDS.Sim}$*, such that for any* $x \in L \cap \{0,1\}^\lambda$ *and any* $w \in R_L(x)$*,*

$$\{\mathsf{CDS.Gen}(1^\lambda, x, w)\} \overset{c}{\approx} \{\mathsf{CDS.Sim}(1^\lambda, x))\}.$$

Constructions. Conditional disclosure of secrets schemes can be constructed from any two-round OT with game-based security (seeing Definition 6)[AJ17, BKP19].

2.7 Commitment

A commitment scheme (C, R) is a two-phase protocol between a committer C and a receiver R, the commitment phase and the opening phase. In the commitment phase, $C(b)$ generates a commitment $\mathsf{Com}(b)$ of $b \in \{0,1\}$ by interacting with R (We define the round of a commitment scheme as the round of its commitment phase.); In the opening phase, in order to decommit $\mathsf{Com}(b)$, C outputs a decommitment (b, dec), and R outputs 1 iff the decommitment is valid.

Definition 15. (Commitment Scheme) *A two-phase protocol* (C, R) *is called a commitment scheme if it satisfies the following two properties:*

1. *Binding: For every polynomial-size committer $C^* = \{C^*_\lambda\}_{\lambda\in\mathbb{N}}$, the probability of the following event is negligible: C^* interacts with R and generates a commitment $\mathsf{Com}(b)$ in the committing phase, and then produces two decommitments (b, dec) and (b', dec') with $b' \neq b$ in two executions of the opening phase.*
2. *Hiding: For every polynomial-size receiver $R^* = \{R^*_\lambda\}_{\lambda\in\mathbb{N}}$, the commitments $\mathsf{Com}(0)$ and $\mathsf{Com}(1)$ are computational indistinguishable.*

Definition 16. (Extractable Commitment) [JKKR17] *In addition to the standard properties of binding and hiding, a commitment is extractable if additionally, for any committer C that generates a commitment transcript Com, there exists an efficient algorithm, called an extractor, which extracts m such that with probability $1 - negl$ over the randomness of the extractor and the transcript, Com could be opened to m.*

Remark 2. In above definition, the extractor is required to extract the committed value no matter whether the commitment is "well-formed" (i.e. computed honestly). In other words, if the commitment is not well-formed, then the extractor must output $\perp$; if the commitment is well-formed, then the extractor must output the correct committed value. Note that the well-known construction of extractable commitments (i.e. [PRS02]) only require the correctness of the extracted value when the commitment is well-formed.

Another notable notion of security is the selective opening security [DNRS03]. Consider a k-parallel composition of a commitment scheme (C, R). We denote by ζ_i the commitment of b_i. In a selective opening attack, malicious R^* can choose a set $I \in \mathcal{I}$ (depended on the commitments received) and ask the committer $\{C_i\}_{i\in I}$ to open the commitments $\{\zeta_i\}_{i\in I}$, where $\mathcal{I}$ is the family of subset of $[k]$. Informally, the commitment scheme (C, R) is said to be secure under selective opening attacks if the remaining unopened commitments still stay secret.

Definition 17. ((T, ϵ)-secure under selective opening attacks) [Den20] *let k be a polynomial in λ, and $\mathcal{B}$ be a distribution on $\{0, 1\}^k$. We denote by $\mathcal{I}$ the family of subset of $[k]$. A commitment scheme (C, R) is (T, ϵ)-secure under selective opening attacks if for any k, any $\mathcal{B}$, any polynomial-size $R^* = \{R^*_\lambda\}_{\lambda\in\mathbb{N}}$, and any polynomial T, any polynomial inverse ϵ, there exists a polynomial-size Sim such that no T-size distinguisher can tell apart the following two distributions with probability greater than ϵ:*

Real distribution:
$\{(\{b_i\}_{i\in[k]}, I, \mathsf{Out}_{R^*})\}$ *where:* $\{b_i\}_{i\in[k]} \leftarrow \mathcal{B}; \{\zeta_i\}_{i\in[k]} \leftarrow (\{C_i(b_i)\}_{i\in[k]}, R^*_\lambda)_{\mathsf{Com}}$;
$I \leftarrow R^*(\{\zeta^i\}_{i\in[k]}); \{b_i, dec_i\}_{i\in I} \leftarrow (\{C_i\}_{i\in[k]}, R^*)_{\mathsf{Open}}; \mathsf{Out}_{R^*} \leftarrow R^*(\{b_i, dec_i\}_{i\in I})$.
Simulation distribution:
$\{(\{b_i\}_{i\in[k]}, I, \mathsf{Out}_{\mathsf{Sim}})$ *where:* $\{b_i\}_{i\in[k]} \leftarrow \mathcal{B};\ I \leftarrow \mathsf{Sim}^{R^*}; \mathsf{Out}_{\mathsf{Sim}} \leftarrow \mathsf{Sim}^{R^*}(\{b_i\}_{i\in I})$.

2.8 Concurrent (T, ϵ)-zero-knowledge Argument in the BPK model and Witness Hiding Argument

In the BPK model, there exists an extra phase, the key-registration phase. In this phase, the verifier needs to register a public key pk on a public-file F before the proof phase. And in the proof phase, prover interacts with verifier under the verifier's public key registered on the public-file F. Parts of the definitions in this section are taken verbatim from [Den20].

Concurrent soundness in the BPK model: For a malicious concurrent prover P^*, it is allowed to launch the following attack: In the proof phase, given a public key pk, P^* initiates polynomially many sessions. In every session, it chooses a statement x adaptively (depending on the interaction history), and fully controls the message scheduling in the entire interaction with V.

Definition 18. (Concurrent Soundness in the BPK model) *An interactive argument (P, V) for a language L in the BPK model is called concurrent sound if for all concurrent malicious prover $P^* = \{P^*_\lambda\}_{\lambda\in\mathbb{N}}$, the probability that in an execution of concurrent attack P^* makes V accept a false statement $x \in \{0,1\}^\lambda \backslash L$ in one section is negligible.*

Concurrent (T, ϵ)-zero-knowledge in the BPK model: A concurrent attack launched by a t-concurrent malicious polynomial-sized verifier V^*, for any polynomial t, is defined as following:

1. In the key-registration stage, V^* registers t public keys $\{pk_i\}_{i\in[t]}$ on the public file F.
2. Upon receiving $\{x_i\}_{i\in[t]}$, V^* interacts with $\{P(x_i, w_i, pk_j, F)\}_{1\le i,j\le t}$ and fully controls the message scheduling in the entire interaction.
3. V^* finally outputs its entire view of the interaction (i.e., its random tape and the messages received from the provers). We write it $\mathsf{View}^{P(F)}_{V^*}(\{x_i\}_{i\in[t]})$.

Definition 19. (Concurrent (T, ϵ)-Zero-Knowledge in the BPK model) *An interactive argument (P, V) for an NP language L in the BPK model is called concurrent (T, ϵ)-zero-knowledge if for any polynomials t, T, any polynomial inverse ϵ and any t-concurrent malicious verifier $V^* = \{V^*_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a polynomial-sized simulator $S = \{S_\lambda\}_{\lambda\in\mathbb{N}}$ such that for any T-size distinguisher $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$, any $\{(x_i, w_i)\}_{i\in[t]}$ where $x_i \in \{0,1\}^\lambda, w_i \in R_L(x_i)$:*

$$\left|\Pr\left[D(\mathsf{View}^{P(F)}_{V^*}(\{x_i\}_{i\in[t]})) = 1\right] - \Pr[D(S(\{x_i\}_{i\in[t]})) = 1]\right| < \epsilon(\lambda)$$

The probability is over the coins of D, V^*, P, S.

Witness hiding is a weaker notion than zero-knowledge, which only requires that for any random instance (statement) x sampled from a hard distribution, no verifier can output a witness at the end of interaction with the prover with noticeable probability.

Definition 20. (Distribution of Hard Instances) *Let L be an NP language. We say a distribution ensemble $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ is hard for relation R_L if for any polynomial-size $\{M_\lambda\}_{\lambda\in\mathbb{N}}$,*

$$\Pr[M_\lambda(X_\lambda) \in R_L(X_\lambda)] \le negl(\lambda)$$

Definition 21. (Delayed-input Witness Hiding Argument) *A delayed-input interactive argument (P, V) for an NP language L is witness hiding if for any polynomial-size circuit V^*, and any hard distribution ensemble $\{X_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\Pr[\mathsf{Out}_{V^*}(P(W_\lambda), V^*)(X_\lambda)) \in R_L(X_\lambda)] \le negl(\lambda),$$

where W_λ is arbitrarily distributed over $R_L(X_\lambda)$.

3 Knowledge Encryption and the Nearly Optimal Extractor for Key Generation

We now introduce a new concept of encryption– *knowledge encryption*. Roughly, a knowledge encryption is a public-key encryption scheme for which ciphertexts can be reduced to the public-key, i.e., any algorithm with large (ciphertexts) distinguishing advantage can be used to extract the (partial) secret key. Like CDS/WE schemes, a public-key of a knowledge encryption scheme is generated from a (publicly known) instance x of an NP language L, but it provides stronger security guarantee in that the decryption of knowledge encryption actually constitutes a *proof of knowledge* of the corresponding (partial) secret key: While CDS/WE schemes guarantee that the receiver obtains nothing about the encrypted message when $x \notin L$, knowledge encryption ensures that any receiver that can decrypt ciphertexts must *know* a valid witness of x (and hence $x \in L$). The semantic security of knowledge encryption is required to hold when $(x, w) \in R_L$ and the public key is honestly generated. This is in contrast to that of CDS/WE schemes, which only consider semantic security for false statements.

Definition 22 (Knowledge Encryption). *A knowledge encryption scheme with respect to an NP relation R_L is a triple of PPT algorithms* $(\mathsf{KE.Gen}, \mathsf{KE.Enc}, \mathsf{KE.Dec})$*:*

- $\mathsf{KE.Gen}(1^\lambda, x, w)$: *On input the security parameter $\lambda \in \mathbb{N}$ and statement $x \in L \cap \{0,1\}^\lambda, w \in R_L(x)$,* Gen *outputs a key pair* $(\mathsf{pk},\mathsf{sk})$*, where the public key is of the form* $\mathsf{pk} = (\mathsf{k}, x)$*.*
- $\mathsf{KE.Enc}(\mathsf{pk}, m)$: *On input the public key* pk *and a message $m \in \{0,1\}$,* $\mathsf{KE.Enc}$ *outputs a ciphertext c.*
- $\mathsf{KE.Dec}(\mathsf{sk}, c)$: *On input the secret key* sk *and ciphertext c,* $\mathsf{KE.Dec}$ *outputs a message m (if c is undecryptable, we set m to be " $\perp$ ").*

We require the following properties from above scheme:

- **Completeness***: For any $\lambda \in \mathbb{N}$, $m \in \{0,1\}$ and $x \in L \cap \{0,1\}^\lambda, w \in R_L(x)$:*

$$\Pr\left[\mathsf{KE.Dec}(\mathsf{sk}, c) = m \;\middle|\; \begin{array}{r}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KE.Gen}(1^\lambda, x, w)\\ c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}, m)\end{array}\right] = 1.$$

- **Indistinguishability***: For any polynomial-size distinguisher $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function $negl$ such that for any security parameter $\lambda \in \mathbb{N}$ and $x \in L \cap \{0,1\}^\lambda, w \in R_L(x)$:*

$$\Pr\left[D_\lambda(\mathsf{pk}, c) = m \;\middle|\; \begin{array}{r}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KE.Gen}(1^\lambda, x, w)\\ m \leftarrow \{0,1\};\ c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}, m)\end{array}\right] < \frac{1}{2} + negl(\lambda).$$

- **Witness Extractability***: There exists a PPT extractor E satisfying that, for any public key $\mathsf{pk}^* = (\mathsf{k}^*, x)$, polynomial-size distinguisher $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$ and inverse polynomial ϵ, if*

$$|\Pr[D_\lambda(\mathsf{KE.Enc}(\mathsf{pk}^*, 0)) = 1] - \Pr[D_\lambda(\mathsf{KE.Enc}(\mathsf{pk}^*, 1)) = 1]| \geq \epsilon,$$

then

$$\Pr[E^{D_\lambda}(\mathsf{pk}^*, 1^{1/\epsilon}) = w \wedge (x, w) \in R_L] \geq 1 - negl(\lambda),$$

where E runs in time polynomial in ϵ^{-1} and λ.

- **Public Key Simulation**: *There exists a PPT simulator* $\mathsf{KE.KeySim}$ *such that for any (x, w) where $x \in L \cap \{0,1\}^\lambda, w \in R_L(x)$:*

$$\{\mathsf{KE.Gen}(1^\lambda, x, w)\} \overset{c}{\approx} \{\mathsf{KE.KeySim}(1^\lambda, x)\}.$$

Remark 3. One can also define the security properties of knowledge encryption over a randomly chosen (according to certain distribution) instance x. We choose our definition because it gives great flexibility in applications, especially in the applications where several parties *jointly* compute the instance x for some public key of knowledge encryption, like our construction of three-round OT. However, we note that the distributional version of our definition may admit more instantiations, for example, the public-key encryption based on Rabin's one-way permutation is also a distributional knowledge encryption scheme.

In the rest of this section, we first present how to construct knowledge encryption from two-round OT, and then we will apply techniques of [Den20] and prove that, for any key generator of knowledge encryption, there exists a nearly optimal extractor for the witness of x such that when it fails, no circuit of a-priori bounded size can distinguish ciphertexts except with small probability.

3.1 Knowledge Encryption from Two-round OT

In this section, we give a construction of knowledge encryption from two-round OT. At a high level, this construction follows the two-party-function-evaluation approach used in CDS scheme, and relies on the following two ingredients:

- A two-round OT $(\mathsf{OT}_1, \mathsf{OT}_2)$ with game-based security, and,
- A garbling circuit scheme $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$.

Note that the garbling circuit scheme can be based on any one-way function, which is already implied by the existence of two-round OT with game-based security.

The main idea behind our construction is to modify the circuit C to be garbled in a CDS scheme and embed a simple decoding mechanism in C, which enables us to reduce the instance x to random ciphertexts. Specifically, we let C take an extra input y of length ℓ and define it as follows:

$$C(x, w, y, m) = \begin{cases} m & \text{if } (x, w) \in R_L \text{ and } y = 0^\ell, \\ \Sigma_{i=1}^{\ell} y_i w_i \bmod 2 & \text{if } (x, w) \in R_L \text{ and } \|y\|_1 \geq 1^{[9]}, \\ \perp & \text{if } (x, w) \notin R_L. \end{cases} \tag{2}$$

The formal description of knowledge encryption for R_L[10] from two-round OT is shown in Fig.2.

[9] In the following proofs, we only consider the case that $\|y\|_1 = 1$. In this case, C will output a coordinate of w, and the extractor will extract the witness bit-by-bit.

[10] For ease of presentation, we assume that for every $x \in L \cap \{0,1\}^\lambda$ there is a string $w^* \in \{0,1\}^\ell$ such that $(x, w^*) \notin R_L$. For any NP relation R_L that does not satisfy this condition, one can easily extend it to a new relation:

Knowledge Encryption from Two-Round OT

$\mathsf{KE.Gen}(1^\lambda, x, w)$: Parse $w = w_1 \| w_2 \| \cdots \| w_\ell$, and choose random coins $\{r_i\}_{i\in[\ell]}$, then run the 2-round OT scheme in parallel to generate $\mathsf{k} := (\mathsf{OT}_1(1^\lambda, w_1; r_1), \cdots, \mathsf{OT}_1(1^\lambda, w_\ell; r_\ell))$. Output the public-key $\mathsf{pk} = (\mathsf{k}, x)$ and the secret key $\mathsf{sk} = (w, r_1, \cdots, r_\ell)$.

$\mathsf{KE.Enc}(\mathsf{pk}, m)$: Set $y = 0^\ell$, and run the GC scheme to generate a garbled circuit $\hat{C}$ with labels $\{\mathsf{lab}^x_{i,b}\}_{i\in[\lambda],b\in\{0,1\}}$, $\{\mathsf{lab}^w_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \{\mathsf{lab}^y_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \{\mathsf{lab}^m_b\}_{b\in\{0,1\}}$ for circuit C defined in (2). Output ciphertext

$$c := (\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[\lambda]}, \{\mathsf{OT}_2(\mathsf{lab}^w_{i,0}, \mathsf{lab}^w_{i,1})\}_{i\in[\ell]}, \{\mathsf{lab}^y_{i,0}\}_{i\in[\ell]}, \mathsf{lab}^m_m) \quad .$$

$\mathsf{KE.Dec}(\mathsf{sk}, c)$: Use sk to retrieve $\{\mathsf{lab}^w_{i,w_i}\}_{i\in[\ell]}$ from $\{\mathsf{OT}_2(\mathsf{lab}^w_{i,0}, \mathsf{lab}^w_{i,1})\}_{i\in[\ell]}$, and compute $m \leftarrow \mathsf{Eval}(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[\lambda]}, \{\mathsf{lab}^w_{i,w_i}\}_{i\in[\ell]}, \{\mathsf{lab}^y_{i,0}\}_{i\in[\ell]}, \mathsf{lab}^m_m)$.

Fig. 2: The construction of Knowledge Encryption from two-round OT

Theorem 1. *Assuming the existence of two-round OT protocol with computational game-based security, there exists a knowledge encryption scheme.*

Proof. We prove that the construction presented in Fig 2 is a knowledge encryption scheme. Since the two-round OT with game-based security implies the existence of garbling scheme, our construction can be based solely on the two-round OT with game-based security. Note first that it is easy to verify the completeness property.

Indistinguishability. For a given pair $(x, w) \in R_L$, denote by $\mathcal{D}_m$ the distribution $\{\mathsf{pk}, c | \mathsf{pk} \leftarrow \mathsf{KE.Gen}(1^\lambda, x, w), c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}, m)\}$ for $m = \{0, 1\}$. We prove $\mathcal{D}_0 \overset{c}{\approx} \mathcal{D}_1$ by a standard hybrid argument. Consider the following distributions.

$\mathcal{D}_{1,m}$: the same as $\mathcal{D}_m$ except that the public key is generated by using $(x, w^*) \notin R_L$, i.e., $\mathsf{pk} \leftarrow \mathsf{KE.Gen}(1^\lambda, x, w^*)$ (w.o.l.g.,we assume that such a w^* exists, see footnote 10.)

$\mathcal{D}_{2,m}$: the same as $\mathcal{D}_{1,m}$ except that it computes $\{\mathsf{OT}_2(\mathsf{lab}^{w^*}_{i,w^*_i}, \mathsf{lab}^{w^*}_{i,w^*_i})\}_{i\in[\ell]}$ in the key generation, rather than $\{\mathsf{OT}_2(\mathsf{lab}^{w^*}_{i,0}, \mathsf{lab}^{w^*}_{i,1})\}_{i\in[\ell]}$.

$\mathcal{D}_{3,m}$: the same as $\mathcal{D}_{2,m}$ except that it generates the labels and garbled circuit using the simulator of GC, i.e., $(\hat{C}, \{\mathsf{lab}_{i,b_i}\}) \leftarrow \mathsf{Sim}(1^\lambda, \phi(C), \perp)$.

Note that the only difference between $\mathcal{D}_m$ and $\mathcal{D}_{1,m}$ is the first OT messages on those positions i where $w_i \neq w^*_i$. Due to the receiver's security of the underlying two-round OT, one can prove that $\mathcal{D}_m \overset{c}{\approx} \mathcal{D}_{1,m}$ by a standard hybrid argument. From the sender's security of the underlying two-round OT, it follows $\mathcal{D}_{1,m} \overset{c}{\approx} \mathcal{D}_{2,m}$. Furthermore, we have $\mathcal{D}_{2,m} \overset{c}{\approx} \mathcal{D}_{3,m}$, since for $(x, w^*) \notin R_L$, the circuit garbled in the distribution $\mathcal{D}_{2,m}$ on input (x, w^*, y, m) always outputs $\perp$. Observing that both $\mathcal{D}_{3,0}$ and $\mathcal{D}_{3,1}$ are generated by the simulator of the garbling scheme and are independent

$R'_L := (x, w') \in \{0,1\}^\lambda \times \{0,1\}^{\ell+1} : w' = w\|1$ and $(x, w) \in R_L$, for which $w\|0$ is not a valid witness (for any instance x).

of the message m, one can see that $\mathcal{D}_{3,0} \equiv \mathcal{D}_{3,1}$. This concludes the proof of indistinguishability of our knowledge encryption scheme.

Public Key Simulation. One can easily construct a simulator for simulating the public key: On input x, the simulator chooses $\{r_i\}_{i\in[\ell]}$ at random and outputs $\mathsf{pk} = (\{\mathsf{OT}_1(1^\lambda, 0; r_i)\}_{i\in[\ell]}, x)$. This simulated public key is indistinguishable from the honestly-generated one due simply to the receiver's security of the underlying two-round OT.

Witness Extractability: Here our basic goal is to build an efficient extractor such that for any $\mathsf{pk}^* = (\mathsf{k}^*, x)$ and any distinguisher D[11] with high distinguishing advantage, the extractor, with oracle access to D, can extract a witness for x except for negligible probability.

Fix an arbitrary public key $\mathsf{pk}^* = ((\mathsf{k}^* = (\mathsf{ot}^*_{1,1}, \cdots, \mathsf{ot}^*_{1,\ell})), x)$. We use the sender's security property (which is against unbounded receiver) of the two-round OT to define $w^* \in \{0,1\}^\ell$ as follows: For each $i \in [\ell]$, if for any (δ_0, δ_1), $\mathsf{OT}_{2,i}(\delta_0, \delta_1)$ is indistinguishable from $\mathsf{OT}_{2,i}(\delta_0, \delta_0)$ against any polynomial-size adversary, $w^*_i = 0$, otherwise $w^*_i = 1$.

Suppose that D is a polynomial-size distinguisher and ϵ is an inverse polynomial such that

$$|\Pr[D(\mathsf{KE.Enc}(\mathsf{pk}^*, 0)) = 1] - \Pr[D(\mathsf{KE.Enc}(\mathsf{pk}^*, 1)) = 1]| \geq \epsilon(\lambda), \qquad (3)$$

we construct a desirable oracle machine E^D to complete the proof of the witness extractability property.

We first argue that the definition of w^*, together with the inequality (3), implies $(x, w^*) \in R_L$. Suppose otherwise $(x, w^*) \notin R_L$. Let $\{\mathcal{D}_{j,m}\}_{j\in[3],m\in\{0,1\}}$ be as above. For every $j \in [3]$ and $m \in \{0,1\}$, Denote by $\mathcal{D}_{j,m}|\mathsf{pk}^*$ the distribution conditioned on pk^*. Then, for each $m \in \{0,1\}$, we have $\mathsf{KE.Enc}(\mathsf{pk}^*, m) \equiv \mathcal{D}_{1,m}|\mathsf{pk}^*$ and $\mathcal{D}_{1,m}|\mathsf{pk}^* \overset{c}{\approx} \mathcal{D}_{2,m}|\mathsf{pk}^*$ (by definition of w^*). Furthermore, applying the same reasoning as in the proof of the indistinguishability property, we also have $\mathcal{D}_{2,m}|\mathsf{pk}^* \overset{c}{\approx} \mathcal{D}_{3,m}|\mathsf{pk}^*$ (for each $m \in \{0,1\}$) and $\mathcal{D}_{3,0}|\mathsf{pk}^* \equiv \mathcal{D}_{3,1}|\mathsf{pk}^*$. Putting together, we conclude that $\mathsf{KE.Enc}(\mathsf{pk}^*, 0)$ and $\mathsf{KE.Enc}(\mathsf{pk}^*, 1)$ are indistinguishable, which contradicts the inequality (3).

We now turn to the construction of the oracle machine E^D assuming the distinguisher D satisfies the inequality (3). Our main idea is to run D on *fake* ciphertexts by manipulating the input y and use its distinguishing advantage to compute the witness w^* bit-by-bit.

Denote by $\vec{y}(j)$ the string with the j-th coordinate being 1 and all others being 0. Observe that, by the definition of circuit C, when choosing $\vec{y}(j)$ to compute a ciphertext, it will be decrypted to w^*_j. We formally define such an encryption algorithm $\mathsf{KE.Enc}'(\mathsf{pk}^*, 0)$ as follows: $\mathsf{KE.Enc}'(\mathsf{pk}^*, 0)$ acts exactly the same as $\mathsf{KE.Enc}(\mathsf{pk}^*, 0)$ except that it chooses $y' = \vec{y}(j) = y'_1 \| y'_2 \| \cdots \| y'_\ell$ (as a result, the i-th label with respect to y generated by $\mathsf{KE.Enc}'(\mathsf{pk}^*, 0)$ is $\mathsf{lab}^y_{j,1}$, rather than $\mathsf{lab}^y_{j,0}$). A ciphertext generated by $\mathsf{KE.Enc}'(\mathsf{pk}^*, 0)$ can be viewed as a ciphertext of w^*_j, and furthermore, the distribution $\mathsf{KE.Enc}'(\mathsf{pk}^*, 0)$ is actually indistinguishable from $\mathsf{KE.Enc}(\mathsf{pk}^*, w^*_j)$. To see this,

[11] D might know of the random coins used to sample pk^*.

consider the following distribution $\mathcal{D}_S$: run the simulator Sim for garbling scheme and obtain $(\hat{C},\{\mathsf{lab}^x_{i,x_i}\}_{i\in[\lambda]},\{\mathsf{lab}^w_{w^*_i}\}_{i\in[\ell]},\{\mathsf{lab}^y_{y'_i}\}_{i\in[\ell]},\mathsf{lab}^m_m) \leftarrow \mathsf{Sim}(1^\lambda,\phi(C),w^*_j)$, and output ciphertext $c=(\hat{C},\{\mathsf{lab}^x_{i,x_i}\}_{i\in[\lambda]},\{\mathsf{OT}_2(\mathsf{lab}^w_{i,w^*_i},\mathsf{lab}^w_{i,w^*_i})\}_{i\in[\ell]},\{\mathsf{lab}^y_{i,y'_i}\}_{i\in[\ell]},\mathsf{lab}^m_m)$.

Note that $w^*_j = C(x,w^*,y'=\vec{y}(j),0) = C(x,w^*,y=0^\ell,w^*_j)$, and for this reason, the above ciphertext simulator can be viewed as a simulator for both $\mathsf{KE.Enc}'(\mathsf{pk}^*,0)$, which garbles C on input $(x,y'=\vec{y}(j),0)$, and $\mathsf{KE.Enc}(\mathsf{pk}^*,w^*_j)$, which garbles C on input $(x,y=0^\ell,w^*_j)$. Similarly to the proof of the indistinguishability property, due to the sender's security of the two-round OT and the security of the garbling scheme, one can prove that both $\mathsf{KE.Enc}'(\mathsf{pk}^*,0)$ and $\mathsf{Enc}(\mathsf{pk}^*,w^*_j)$ are indistinguishable from $\mathcal{D}_S$. Thus,

$$\mathsf{KE.Enc}'(\mathsf{pk}^*,0)) \stackrel{c}{\approx} \mathsf{KE.Enc}(\mathsf{pk}^*,w^*_j)). \tag{4}$$

This means the distinguisher D can tell apart $\mathsf{KE.Enc}'(\mathsf{pk}^*,0))$ from $\mathsf{KE.Enc}(\mathsf{pk}^*,1-w^*_j))$, which gives rise to the following oracle extraction machine E^D.

$E^D(\mathsf{pk}^*,1^{1/\epsilon})$

1. For each $j\in[\lambda]$:
   (a) Run D on input $\mathsf{KE.Enc}(\mathsf{pk}^*,0)$ $\lambda\epsilon^{-2}$ times with fresh randomness (for both D and $\mathsf{KE.Enc}$) each time. Denote by $d_{0,k}$ the output of $D(\mathsf{KE.Enc}(\mathsf{pk}^*,0))$ in the k-th repetition. Compute $d_0 = \lambda^{-1}\epsilon^2\Sigma_{k\in[p]}d_{0,k}$.
   (b) Run D on input $\mathsf{KE.Enc}(\mathsf{pk}^*,1)$ $\lambda\epsilon^{-2}$ times with fresh randomness (for both D and $\mathsf{KE.Enc}$) each time. Denote by $d_{1,k}$ the output of $D(\mathsf{KE.Enc}(\mathsf{pk}^*,1))$ in the k-th repetition. Compute $d_1 = \lambda^{-1}\epsilon^2\Sigma_{k\in[p]}d_{1,k}$.
   (c) Run D on input $\mathsf{KE.Enc}'(\mathsf{pk}^*,0)$ $\lambda\epsilon^{-2}$ times with fresh randomness (for both D and $\mathsf{KE.Enc}$) each time. Denote by $\hat{d}_k$ the output of $D(\mathsf{KE.Enc}'(\mathsf{pk}^*,0))$ in the k-th repetition. Compute $\hat{d} = \lambda^{-1}\epsilon^2\Sigma_{k\in[p]}d_{0,k}$.
   (d) If $|d_0-\hat{d}| > |d_1-\hat{d}|$, then set $\hat{w}_j=1$, if else, set $\hat{w}_j=0$.
2. Output $\hat{w}=\hat{w}_1\|\hat{w}_2\|\cdots\|\hat{w}_\ell$.

We denote by u_0 the probability $\Pr[D(\mathsf{KE.Enc}(\mathsf{pk}^*,0))=1]$, by u_1 the probability $\Pr[D(\mathsf{KE.Enc}(\mathsf{pk}^*,1))=1]$ and by $\hat{u}$ the probability $\Pr[D(\mathsf{KE.Enc}'(\mathsf{pk}^*,0))=1]$. By Chernoff bound, we have

$$\Pr[|d_0-u_0|\geq\delta u_0]\leq 2e^{-\delta^2u_0p/3}.$$

Set $\delta u_0=\epsilon/8$. Due to that $u_0\leq 1$, we have that $\delta\geq\epsilon/8$. Therefore,

$$\Pr[|d_0-u_0|\geq\epsilon/8]\leq 2e^{-\lambda/2^6\cdot 3}. \tag{5}$$

Similarly,

$$\Pr[|d_1-u_1|\geq\epsilon/8]\leq 2e^{-\lambda/2^6\cdot 3}, \text{ and} \tag{6}$$

$$\Pr[|\hat{d}-\hat{u}|\geq\epsilon/8]\leq 2e^{-\lambda/2^6\cdot 3}. \tag{7}$$

From the (in)equalities (3) and (4), we also have $|u_0 - u_1| \geq \epsilon$ and $|\hat{u} - u_{w_j^*}| \leq negl$. Putting together with the inequalities (5),(6),(7), it follows

$$\Pr[|d_{1-w_j^*} - \hat{d}| > |d_{w_j^*} - \hat{d}|] \geq 1 - negl,$$

which implies that,

$$\Pr[w_j^* \neq \hat{w}_j | \hat{w} \leftarrow E^D(\mathsf{pk}^*, 1^{1/\epsilon})] \leq negl(\lambda).$$

Note also that $(x, w^*) \in R_L$, we have

$$\Pr[\hat{w} \leftarrow E^D(\mathsf{pk}^*, 1^{1/\epsilon}) \wedge (x, \hat{w}) \in R_L] \geq 1 - negl(\lambda),$$

as desired. □

3.2 Knowledge Encryption from RSR Encryption and CDS Scheme

In this section, we present an alternative construction of knowledge encryption for an NP language L assuming the following ingredients:

- An RSR encryption scheme ($\mathsf{RSR.Gen}$,$\mathsf{RSR.Enc}$,$\mathsf{RSR.Dec}$), and,
- A CDS scheme ($\mathsf{CDS.Gen}$,$\mathsf{CDS.Enc}$,$\mathsf{CDS.Dec}$) for the following language L':

$L' = \{(x, pk_{\mathsf{RSR}}, c_w) | \exists w \in R_L(x)\ s.t.\ pk_{\mathsf{RSR}} \in \mathsf{RSR.Gen}(1^\lambda), c_w \in \mathsf{RSR.Enc}(pk_{\mathsf{RSR}}, w)\}$.

Inspired by work of [BKP19], our construction critically relies on the random self-reducibility of the RSR encryption, which says that the ciphertext c_w of the witness w (in the public key) reduces to random ciphertexts. The purpose of the CDS scheme is to make sure that the witness extractability property holds even for a maliciously-generated public key.

Given $(x, w) \in R_L$, the public key consists of two public keys, one for the RSR encryption and one for the CDS scheme, the instance x and a ciphertext of w under the public key of RSR encryption. The corresponding secret key includes the secret keys for both the RSR encryption and the CDS scheme. When encrypting a message, one executes double encryptions: First encrypts it under the public key of the RSR encryption and then encrypts the RSR ciphertext under the public key of the CDS scheme. Decryption follows naturally. The formal description of this construction is depicted in Fig.3.

Theorem 2. *Assuming the existence of RSR encryption and two-round OT with game-based security, there exists a knowledge encryption scheme.*

Proof. We give a sketch of proof that the construction in Fig.3 is a knowledge encryption scheme. Note that two-round OT with game-based security implies the existence of the CDS scheme.

The **Completeness** property is obvious.

The **Indistinguishability** property follows from the indistinguishability of the RSR encryption and receiver simulation of CDS scheme. To see this, we denote by $\mathcal{D}_m$ the distribution $\{\mathsf{pk}, c | \mathsf{pk} \leftarrow \mathsf{KE.Gen}(1^\lambda, x, w), c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}, m)\}$ for $m = \{0, 1\}$,

An Alternative Construction of Knowledge Encryption

$\mathsf{KE.Gen}(1^\lambda, x, w)$: Generate $(pk_{\mathsf{RSR}}, sk_{\mathsf{RSR}}) \leftarrow \mathsf{RSR.Gen}(1^\lambda)$, $c_w \leftarrow \mathsf{RSR.Enc}(w; r)$ and, $(pk_{\mathsf{CDS}}, sk_{\mathsf{CDS}}) \leftarrow \mathsf{CDS.Gen}(1^\lambda, (x, pk_{\mathsf{RSR}}, c_w), (w, r))$ for $((x, pk_{\mathsf{RSR}}, c_w), (w, r)) \in R_{L'}$. Output $\mathsf{pk} = ((pk_{\mathsf{RSR}}, c_w, pk_{\mathsf{CDS}}), x)$ and $\mathsf{sk} = (sk_{\mathsf{RSR}}, sk_{\mathsf{CDS}})$.

$\mathsf{KE.Enc}(\mathsf{pk}, m)$: Compute and output
$c \leftarrow \mathsf{CDS.Enc}(pk_{\mathsf{CDS}}, (x, pk_{\mathsf{RSR}}, c_w), \mathsf{RSR.Enc}(pk_{\mathsf{RSR}}, m))$.

$\mathsf{KE.Dec}(\mathsf{sk}, c)$: Run $c' \leftarrow \mathsf{CDS.Dec}(sk_{\mathsf{CDS}}, c)$ and then output $m \leftarrow \mathsf{RSR.Dec}(sk_{\mathsf{RSR}}, c')$.

Fig. 3: The construction of Knowledge Encryption from RSR encryption and CDS

by $\mathcal{D}'_m$ the distribution identically to $\mathcal{D}_m$ except that the public key of CDS scheme is generated using the receiver simulator. From the receiver simulation of CDS scheme, we have that $\mathcal{D}'_m \overset{c}{\approx} \mathcal{D}_m$. From the indistinguishability of the RSR encryption, we have that $\mathcal{D}'_0 \overset{c}{\approx} \mathcal{D}'_1$, therefore, $\mathcal{D}_0 \overset{c}{\approx} \mathcal{D}_1$.

We construct a simulator for **Public Key Simulation** as follows: On input the statement $x \in L$, the simulator $\mathsf{KE.KeySim}(1^\lambda, x)$ generates $(pk_{\mathsf{RSR}}, sk_{\mathsf{RSR}}) \leftarrow \mathsf{RSR.Gen}(1^\lambda)$ and $c_w \leftarrow \mathsf{RSR.Enc}(0^\lambda; r)$, $pk_{\mathsf{CDS}} \leftarrow \mathsf{CDS.Sim}(1^\lambda, (x, pk_{\mathsf{RSR}}, c_w))$. Output $\mathsf{pk} = ((pk_{\mathsf{RSR}}, c_w, pk_{\mathsf{CDS}}), x)$. This simulation is indistinguishable from the honest key generation due to the indistinguishability of the RSR encryption and the receiver (key) simulation property of the CDS scheme.

We can prove the **Witness Extractability** property similarly to the proof of our first construction, except that here we use the random self-reducibility of the underlying RSR encryption to extract the witness w.

Fix an arbitrary public key $\mathsf{pk}^* = ((pk_{\mathsf{RSR}}, c_w, pk_{\mathsf{CDS}}), x)$ and suppose that we have a distinguisher D with advantage greater than an inverse polynomial ϵ:

$$|\Pr[D(\mathsf{KE.Enc}(\mathsf{pk}^*, 0)) = 1] - \Pr[D(\mathsf{KE.Enc}(\mathsf{pk}^*, 1)) = 1]| \geq \epsilon(\lambda).$$

Similarly, assuming the existence of the above distinguisher D, one can prove $(pk_{\mathsf{RSR}}, c_w, x) \in L'$. Otherwise, from the message indistinguishability of CDS scheme on a false instance $(pk_{\mathsf{RSR}}, c_w, x) \notin L'$, we would have $\mathsf{KE.Enc}(\mathsf{pk}^*, 0) \overset{c}{\approx} \mathsf{KE.Enc}(\mathsf{pk}^*, 1)$.

Our oracle machine $E^D(\mathsf{pk})$ for extracting the encrypted w proceeds as follows.

$E^D(\mathsf{pk}, 1^{1/\epsilon})$:

1. Construct a RSR ciphertext distinguisher D' as follows: On input a RSR ciphertext challenge c, D' uses the pk_{CDS} to encrypt it to obtain c', then outputs $D(c')$.
2. Run the extractor $\mathsf{RSR.Dec}^{*D'}$ guaranteed by the random self-reducible property of RSR encryption to extract the witness, i.e. $w \leftarrow \mathsf{RSR.Dec}^{*D'}(c_w, pk_{\mathsf{RSR}}, 1^{1/\epsilon})$.

By the construction of D' and the assumption on the distinguishing advantage of D, we have

$$|\Pr[D'(\mathsf{RSR.Enc}(pk_{\mathsf{RSR}}, 0)) = 1] - \Pr[D'(\mathsf{RSR.Enc}(pk_{\mathsf{RSR}}, 1)) = 1]| \geq \epsilon.$$

It follows from the random self-reducible property of RSR encryption that, in its second step, E^D will output a valid witness w such that $(x, w) \in R_L$ with probability negligibly close to 1. □

3.3 Nearly-optimal Extractor for Knowledge Encryption

Following [Den20], we show the existence of the nearly optimal (T, ϵ)-extractor for any (malicious) key generation algorithm of knowledge encryption, which essentially states that, for any ciphertext distinguisher of size T, the probability that the extractor fails to extract a valid witness for the instance x on the public key whereas the ciphertext distinguisher succeeds is less than ϵ. For any (malicious) key generator that generates multiple public keys simultaneously, this property holds for each one of them, even if the distinguisher takes the output of the nearly optimal extractor as input.

For a given polynomial t, denote by $\overline{x}_{[t]}$ the set of t strings $\{x_k\}_{k\in[t]}$. We first recall the lemma on the existence of nearly-optimal (T, ϵ)-extractor for any hard distributions in [Den20].

Lemma 1 (Nearly-Optimal (T, ϵ)-Extractor for t-Instance Sampler [Den20]). *Let L be an NP language and poly be the size of the circuits for deciding the NP-language R_L. Let Samp be an arbitrarily t-instance sampling algorithm over L with input distribution ensemble $\mathcal{R} := \{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$. Let $F := \{F_\lambda\}_{\lambda\in\mathbb{N}}$ be a probabilistic (not necessarily efficient-computable) machine.*

1. *For every polynomial T, ϵ^{-1}, there exists a probabilistic circuit family $\mathsf{Ext} := \{\mathsf{Ext}_\lambda\}_{\lambda\in\mathbb{N}}$ of size $O(\frac{t}{\epsilon}(T + poly))$ such that for every $j \in [t]$, every probabilistic circuit family $C := \{C_\lambda\}_{\lambda\in\mathbb{N}}$ of size T and every security parameter $\lambda \in \mathbb{N}$,*

$$\Pr\left[\begin{array}{l|l}(x_j, w_j^*) \in R_L \wedge & r \leftarrow \mathcal{R}; \overline{x}_{[t]} \leftarrow \mathsf{Samp}(1^\lambda, r); \\ (x_j, w_j') \notin R_L & \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{x}_{[t]}, r, F(r)); \\ & w_j^* \leftarrow C(\overline{x}_{[t]}, r, F(r), \overline{w}_{[t]});\end{array}\right] < \epsilon(\lambda).$$

2. *There exists a probabilistic circuit family $\mathsf{Ext} := \{\mathsf{Ext}_\lambda\}_{\lambda\in\mathbb{N}}$ of quasi-polynomial size such that for every probabilistic circuit family $C := \{C_\lambda\}_{\lambda\in\mathbb{N}}$ of polynomial size, the above probability is negligible.*

The original version of this lemma in [Den20] considers only a deterministic function F, however, it is easy to verify that the same proof also yields the above lemma with respect to a probabilistic (possibly unbounded) function F.

We consider an arbitrary key generator $\mathsf{KE.Gen}^*$ that outputs t public keys simultaneously. We write its input as r (including possibly its random coins, NP instances and the corresponding witnesses), and assume that r are drawn from certain distribution ensemble $\mathcal{R} := \{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$.

The following lemma can be viewed as a knowledge encryption version of Lemma 4 in [Den20] (which holds only with respect to the Rabin's encryption based on factoring).

Lemma 2. *Let t be a polynomial. Let $\mathsf{KE.Gen}^*$ be any t-public-key generator of knowledge encryption with respect to an NP language L, whose output is of the form $\overline{\mathsf{pk}}^*_{[t]} = \{(\mathsf{k}^*_k, x_k)\}_{k\in[t]}$, and let the input distribution ensemble be $\mathcal{R} := \{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$. Let $F := \{F_\lambda\}_{\lambda\in\mathbb{N}}$ be a probabilistic (not necessarily efficient-computable) machine.*

1. *For every polynomial T and every inverse polynomial ϵ, there exists a probabilistic circuit family $\mathsf{Ext} := \{\mathsf{Ext}_\lambda\}_{\lambda\in\mathbb{N}}$ of polynomial size such that for every $j \in [t]$, every probabilistic distinguisher $D := \{D_\lambda\}_{\lambda\in\mathbb{N}}$ of size T and any security parameter $\lambda \in \mathbb{N}$,*

$$\left|\Pr\left[\begin{array}{r|l} D(\overline{\mathsf{pk}}^*_{[t]}, c, r, F(r), \overline{w}'_{[t]}) = 1 \wedge & r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(1^\lambda, r) \\ (x_j, w'_j) \notin R_L & \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \\ & c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}^*_j, 0); \end{array}\right] - \Pr\left[\begin{array}{r|l} D(\overline{\mathsf{pk}}^*_{[t]}, c, r, F(r), \overline{w}'_{[t]}) = 1 \wedge & r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(1^\lambda, r) \\ (x_j, w'_j) \notin R_L & \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \\ & c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}^*_j, 1); \end{array}\right]\right| < \epsilon(\lambda).$$

2. *There exists a probabilistic circuit family $\mathsf{Ext} := \{\mathsf{Ext}_\lambda\}_{\lambda\in\mathbb{N}}$ of quasi-polynomial size such that for every probabilistic distinguisher $D := \{D_\lambda\}_{\lambda\in\mathbb{N}}$ of polynomial size, the above holds with respect to a negligible function ϵ.*

Proof. We only prove the first property of this lemma. Since the size of nearly-optimal extractor is actually a polynomial in $1/\epsilon$ and the size of C, the second property follows when we set ϵ to be negligible.

Let T'' be the size of witness extractor of knowledge encryption, working for any T'-size distinguisher with distinguishing gap ϵ, as guaranteed by the witness extractability property of knowledge encryption. Let T_R be size of the circuit that decides whether $(x, w) \in R_L$.

From Lemma 1, there exists a nearly-optimal $(T'', \epsilon/2)$-extractor Ext satisfying that for any T''-size C:

$$\Pr\left[\begin{array}{l|l} (x_j, w^*_j) \in R_L \wedge & r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(r) \\ (x_j, w'_j) \notin R_L & \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \\ & w^*_j \leftarrow C(\overline{\mathsf{pk}}^*_{[t]}, r, F(r), \overline{w}'_{[t]}); \end{array}\right] < \epsilon(\lambda)/2. \tag{8}$$

Now suppose that the first property of Lemma 2 does not hold, i.e., there exists a distinguisher D of size T such that the left-hand side of the corresponding inequality greater than ϵ. Consider the following distinguisher D' of size $T' = T + T_R$: On the same input of D, D' checks if $(x_j, w'_j) \in R_L$, if not, outputs what D outputs; otherwise, outputs 0. Thus, we have (notice that the distinguishing advantage of D' is 0 when $(x_j, w'_j) \in R_L$):

$$\left|\Pr\left[D'(\overline{\mathsf{pk}}^*_{[t]}, c, r, F(r), \overline{w}'_{[t]}) = 1 \;\middle|\; \begin{array}{r} r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(1^\lambda, r) \\ \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \\ c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}^*_j, 0); \end{array}\right] - \Pr\left[D'(\overline{\mathsf{pk}}^*_{[t]}, c, r, F(r), \overline{w}'_{[t]}) = 1 \;\middle|\; \begin{array}{r} r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(1^\lambda, r) \\ \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \\ c \leftarrow \mathsf{KE.Enc}(\mathsf{pk}^*_j, 1); \end{array}\right]\right| > \epsilon(\lambda).$$

By the witness extractability of knowledge encryption, we have a circuit $C := E^{D'}$ of size T'' with respect to such a D' of size T', which guarantees that, given the same input of D' with $(x_j, w'_j) \notin R_L$, C will extract a valid witness for x_j with probability negligibly close to 1. Thus, it follows

$$\begin{aligned} &\Pr\left[\begin{array}{l}(x_j, w^*_j) \in R_L \wedge \\ (x_j, w'_j) \notin R_L\end{array} \;\middle|\; \begin{array}{r} r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(r) \\ \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \\ w^*_j \leftarrow C(\overline{\mathsf{pk}}^*_{[t]}, r, F(r), \overline{w}'_{[t]}); \end{array}\right] \\ \geq &\Pr\left[(x_j, w'_j) \notin R_L \;\middle|\; \begin{array}{r} r \leftarrow \mathcal{R}; \overline{\mathsf{pk}}^*_{[t]} \leftarrow \mathsf{KE.Gen}^*(r) \\ \overline{w}'_{[t]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}^*_{[t]}, r, F(r)); \end{array}\right] - negl \\ \geq &\epsilon - negl, \end{aligned}$$

where the last inequality simply follows from our assumption that the left-hand side of the inequality Lemma 2 is greater than ϵ. We arrive at a contradiction with inequality (8). □

Remark 4. The proof strategy of [Den20] for this kind of lemma only works if the algorithms Ext and D take the same input (except that D is also given the output of Ext as input). However, in the security reduction, D usually sees a complete session transcript, but the simulator has only a partial transcript when it applies Ext to extract some secrets from the adversary. This is the reason why we have both Ext and D take an extra input $F(r)$, which represents some messages in a session generated after the point that the simulator did extraction. Although $F(r)$ may not be efficiently computable from the input of Ext, but in our cases, the simulator is able to compute it efficiently with the randomness used in generating certain transcript prefix.

4 Three-round Simulatable Oblivious Transfer

In this section, we show how to use the knowledge encryption scheme to construct a three-round OT scheme with simulatable security for the receiver and (T, ϵ)-simulatable security for the sender.

Our protocol proceeds as follows. The sender generates two images y_0 and y_1 of a one-way function f and prove to the receiver that it knows one pre-image of y_0 or y_1 via a three-round WI protocol. Given the pair (y_0, y_1) and input b, the receiver prepares two instances x_0 and x_1 in the following way: it runs the HVZK simulator of the Σ-protocol to obtain an acceptable proof (a, b, z) of knowledge of one preimage of y_0 or y_1, and sets $x_b = (y_0, y_1, a, b)$ and $x_{1-b} = (y_0, y_1, a, 1-b)$, where $x_i = (y_0, y_1, a, i)$ is said to be a YES instance if and only if there exists a z such that (a, i, z) is acceptable. The receiver now generates pk_b honestly using the valid witness z for $x_b = (y_0, y_1, a, b)$, and runs the key simulator of knowledge encryption to obtain the other public key pk_{1-b}. In the third round, the sender encrypts its two message under the two public keys respectively and sends the two ciphertexts to the receiver.

We give a formal description of our construction in Fig.4, which is based on the following ingredients:

- A one-way function f.
- A three-round public-coin witness indistinguishable argument $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ with special soundness and negligible soundness error for language L_f.
- A Σ-protocol (a, e, z) with 1-bit challenge for language L_f.
- A knowledge encryption scheme $(\mathsf{KE.Gen}, \mathsf{KE.Enc}, \mathsf{KE.Dec})$ for language L_Σ.

where L_f, L_Σ are defined as follows:

$$L_f := \{(y_0, y_1) | \exists x \ s.t. \ f(x) = y_0 \vee f(x) = y_1\}$$
$$L_\Sigma := \{(y_0, y_1, a, e) | \exists z \ s.t. \ (a, e, z) \text{ is an acceptable proof for } (y_0, y_1) \in L\}$$

Note that non-interactive commitment can be built from two-round (perfectly correct) OT with game-based security as defined in Definition 6(see footnote 4). Thus, two-round OT with game-based security as we define is sufficient for constructing all primitives used in our protocol.

Theorem 3. *Assuming the existence of two-round OT with game-based security (against polynomial-time adversaries), there exists a three-round OT protocol with fully simulatable security for the receiver and (T, ϵ)-simulatable security for the sender. Furthermore, the same protocol also achieves quasi-polynomial simulatable security for the sender under the same assumption.*

Proof. In the following, we prove that the protocol presented in Fig.4 is a three-round OT protocol with fully simulatable security for the receiver and (T, ϵ)-simulatable security for the sender. By replacing (T, ϵ)-extractor with a quasi-polynomial extractor (guaranteed by Lemma 2) in the simulation of the malicious receiver's view, the second part of Theorem 3 follows.

Fully Simulatable Security for the Receiver.The basic simulation strategy for the receiver security is to rewind the malicious sender, and once a preimage of one of the two images generated by the sender is extracted out, it could generate two public keys using the honest key generation algorithm Gen, which allows it to decrypt both ciphertexts[12] from the sender.

[12] Like the honest receiver, the simulator sets the "plaintext" of an undecryptable ciphertext to be $\perp$

Three-round Oblivious Transfer Protocol

Sender Input: Security parameter 1^λ and messages $m_0, m_1 \in \{0,1\}^n$.
Receiver Input: Security parameter 1^λ and bit $b \in \{0,1\}$.

- **Sender Message**: Sample $\delta_0, \delta_1 \leftarrow \{0,1\}^\lambda$ at random, compute $y_0 = f(\delta_0)$, $y_1 = f(\delta_1)$ and generate WI_1 as the first message of WI for $(y_0, y_1) \in L_f$. Send $(y_0, y_1, \mathsf{WI}_1)$.
- **Receiver Message**: Generate the second WI message WI_2. Use the HVZK simulator of the Σ-protocol to generate an acceptable Σ-proof (a, b, z) for $(y_0, y_1) \in L_f$ (where b is the receiver's input). Generate $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{KE.Gen}(1^\lambda, (y_0, y_1, a, b), z)$ (where $((y_0, y_1, a, b), z) \in R_{L_\Sigma}$) and $\mathsf{pk}_{1-b} \leftarrow \mathsf{KE.KeySim}(1^\lambda, (y_0, y_1, a, 1-b))$. Send $(\mathsf{WI}_2, \mathsf{pk}_0, \mathsf{pk}_1)$.
- **Sender Message**: Write $\mathsf{pk}_i = (\mathsf{k}_i, x_i = ((y_0, y_1, a, i)))$ for $i \in \{0,1\}$, and check if both x_i share the same (y_0, y_1, a). If not, abort; Otherwise, generate the third WI message WI_3 using a random witness and encrypt messages m_i under public key pk_i in bitwise manner: $c_0 \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_0, m_0)$, $c_1 \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_1, m_1)$. Send $(\mathsf{WI}_3, c_0, c_1)$.
- **Receiver's Output**: Check if $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ is acceptable. If not, output $\perp$; otherwise, output $m_b \leftarrow \mathsf{KE.Dec}(\mathsf{sk}_b, c_b)$(if c_b is not decryptable, set m_b to be $\perp$).

Fig. 4: Three-round Oblivious Transfer Protocol

One subtle issue arises in this rewinding strategy. Note that there is a gap between the probability that the sender answer the receiver message before rewinding and the one after, since the public keys are generated in different ways in these two cases. As noted in [GK96a], this gap, albeit being negligible, may cause the simulator to run in exponential time. Goldreich and Kahan introduced an estimation technique to bound the running time of the simulator. Here we use the their technique in our simulation to solve the same issue.

$\mathsf{Sim}^{S^*(\alpha)}$:

1. Run $(y_0, y_1, \mathsf{WI}_1) \leftarrow S^*(1^\lambda, m_0, m_1, r, \alpha)$ with random randomness r and auxiliary input α.
2. Generate a and WI_2 honestly, and define $x_i := (y_0, y_1, a, i)$ for $i \in \{0,1\}$. Generate $\mathsf{pk}_i \leftarrow \mathsf{KE.KeySim}(1^\lambda, x_i)$ for $i \in \{0,1\}$ using the public key simulator of knowledge encryption. Send $(\mathsf{WI}_2, \mathsf{pk}_0, \mathsf{pk}_1)$ to S^*, and obtain the third message $(\mathsf{WI}_3, c_0, c_1)$ from S^*.
3. Check if $\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3$ an acceptable WI proof. If not, send $(\perp, \perp)$ to $\mathcal{F}_{OT}$ and output the view of S^*. Otherwise, go to the next step.
4. *Estimation*: Rewind S^* to the point when it just sent out the first sender message, and repeat the step 2 until the n^2-th acceptable WI proof from S^* is obtained. Denote by X the total number of repetitions of the step 2.
5. Use two acceptable WI proofs (generated above) $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ and $(\mathsf{WI}_1, \mathsf{WI}'_2, \mathsf{WI}'_3)$ to extract a pre-image β of y_0 or y_1.

6. Repeat the following until an acceptable WI proof from S^* is obtained or the total number of repetitions reaches X: Rewind S^* to the point when it just sent out the first sender message. Generate WI_2^* honestly, and use β to Obtain two acceptable Σ-proofs $(a^*, 0, z_0^*)$ and $(a^*, 1, z_1^*)$. Generate $(\mathsf{pk}_i^*, \mathsf{sk}_i^*) \leftarrow \mathsf{KE.Gen}(1^\lambda, a^*, i, z_i^*)$ for $i \in \{0, 1\}$. Send $(\mathsf{WI}_2^*, \mathsf{pk}_0^*, \mathsf{pk}_1^*)$ to S^* and obtain $(\mathsf{WI}_3^*, c_0^*, c_1^*)$ from S^*.
7. If no acceptable WI proof from S^* is obtained in step 6, send $(\perp, \perp)$ to $\mathcal{F}_{OT}$ and output the view of S^* in the last repetition. Otherwise, let $(\mathsf{WI}_3^*, c_0^*, c_1^*)$ be the acceptable message from S^* obtained in step 6, and decrypt $m_i^* \leftarrow \mathsf{KE.Dec}(\mathsf{sk}_i^*, c_i^*)$ (if c_i^* is not decryptable, set m_i^* to be $\perp$) for $i \in \{0, 1\}$. Send (m_0^*, m_1^*) to $\mathcal{F}_{OT}$ and output the view of S^* in the last repetition of step 6.

As showed in [GK96a], the estimation step guarantees that, with probability negligibly close to 1, the probability that the sender answering the receiver message is approximately n^2/X (up to a constant factor) and the simulator runs in expected polynomial time.

Next, we prove

$$\{\mathsf{REAL}_{\Pi, S_\lambda^*(\alpha)}(1^\lambda, m_0, m_1, b)\} \stackrel{c}{\approx} \{\mathsf{IDEAL}_{\mathcal{F}_{OT}, \mathsf{Sim}^{S_\lambda^*(\alpha)}}(1^\lambda, m_0, m_1, b)\}$$

by hybrid argument. Denote by $\mathsf{HYB}_0(\lambda)$ the real world experiment.

$\mathsf{HYB}_1(\lambda)$: This is the same as $\mathsf{Sim}^{S^*(\alpha)}$ except that, in step 2 (and hence in each repetition in step 4) and each repetition of step 6, it acts exactly as an honest receiver.

It is easy to verify that, conditioned on the event that an acceptable WI proof from the sender is obtained in step 6, $\mathsf{HYB}_0(\lambda)$ is identical to $\mathsf{HYB}_1(\lambda)$. As showed in [GK96a], this event occurs with probability negligibly close to 1. Thus, we conclude that $\mathsf{HYB}_0(\lambda)$ is statistically close to $\mathsf{HYB}_1(\lambda)$.

$\mathsf{HYB}_2(\lambda)$: This is the same as $\mathsf{HYB}_1(\lambda)$ except that it generates both public keys $\mathsf{pk}_0, \mathsf{pk}_1$ in step 2 (and hence in each repetition in step 4) by running the key simulator $\mathsf{KE.KeySim}$.

It follows from the public key simulation property that $\mathsf{HYB}_2(\lambda)$ is indistinguishable from $\mathsf{HYB}_1(\lambda)$.

$\mathsf{HYB}_3(\lambda)$: This is the same as $\mathsf{HYB}_2(\lambda)$ except that it generates the first message a of the Σ-protocol in step 2 (and hence in each repetition in step 4) by following the honest prover strategy (rather than the HVZK simulator).

$\mathsf{HYB}_4(\lambda)$: This is the same as $\mathsf{HYB}_3(\lambda)$ except that it uses the extracted witness and generates (a^*, b, z_b^*) in each repetition of step 6 by following the honest prover strategy.

From the HVZK property of the Σ-protocol, it follows that $\mathsf{HYB}_2(\lambda)$, $\mathsf{HYB}_3(\lambda)$, and $\mathsf{HYB}_4(\lambda)$ are indistinguishable.

$\mathsf{HYB}_5(\lambda)$: This is the same as $\mathsf{HYB}_4(\lambda)$ except that it uses the extracted witness and generates both (a^*, b, z_b^*) and $(a^*, 1-b, z_{1-b}^*)$ in each repetition of step 6 by following the honest prover strategy, and then generates pk_{1-b}^* by running $\mathsf{KE.Gen}$ (rather than $\mathsf{KE.KeySim}$).

Again, from the public key simulation property of knowledge encryption, it follows that $\mathsf{HYB}_4(\lambda)$ and $\mathsf{HYB}_5(\lambda)$ are indistinguishable.

Observe that $\mathsf{HYB}_5(\lambda)$ is identical to the ideal experiment $\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}^{S^*_\lambda(\alpha)}}(1^\lambda, m_0, m_1, b)$. This concludes the proof of receiver's security.

(T,ϵ)-**Simulatable Security for the Sender**. We start with a high level description of the simulator for the sender. The simulator generates the first message by following the honest sender strategy. Upon receiving two public keys $\mathsf{pk}_0 = (\mathsf{k}_0, x_0), \mathsf{pk}_1 = (\mathsf{k}_1, x_1)$ of knowledge encryption from the malicious receiver, it applies the nearly optimal extractor for the receiver and tries to extract one witness of x_i. For the case that the simulator extracts two witnesses, it aborts the simulation; For the case that the simulator extracts at most one valid witness, it sets $b' = 0$ if a valid z_0 is extracted s.t. $(x_0 = (y_0, y_1, a, 0), z_0) \in R_{L_\Sigma}$ and sets $b' = 1$ if else. Then it sends b' to $\mathcal{F}_{OT}$ and encrypts the message $m_{b'}$ received from $\mathcal{F}_{OT}$ under both public keys $\mathsf{pk}_{b'}$ and $\mathsf{pk}_{1-b'}$. For the first case, we prove that it happens only with negligible probability. For the second case, we will use the (near) optimality of the extractor to prove that the simulation and the real execution are indistinguishable against distinguishers of certain size except for small probability.

We fix the security parameter λ, the polynomial T and the inverse polynomial ϵ. Fix an arbitrary messages $m_0, m_1 \in \{0,1\}^n$ and $b \in \{0,1\}$. For a given (malicious) receiver R^* and an auxiliary input distribution $\mathcal{Z}$, we define:

- $\mathcal{R}$: The distribution over the *entire* input of R^*. This is a joint distribution of four random variables: the private input b, the receiver's randomness r, the first message of the sender S and the auxiliary input τ (drawn from $\mathcal{Z}$).
- $\mathsf{KE.Gen}^*$: On input $(b, r, \mathsf{WI}_1, y_0, y_1, \tau) \leftarrow \mathcal{R}$, compute $(\mathsf{WI}_2, \mathsf{pk}_0, \mathsf{pk}_1) \leftarrow R^*(b, r, \mathsf{WI}_1, y_0, y_1, \tau)$ and output $(\mathsf{pk}_0, \mathsf{pk}_1)$.
- F: On input $(b, r, \mathsf{WI}_1, y_0, y_1, \tau)$, run $R^*(b, r, \mathsf{WI}_1, y_0, y_1, \tau)$ to obtain $(\mathsf{WI}_2, \mathsf{pk}_0, \mathsf{pk}_1)$, compute all possible witnesses and the prover's randomness $\{(\delta, r_p)\}$ that are consistent with the first two messages $(\mathsf{WI}_1, \mathsf{WI}_2)$ for statement $(y_0, y_1) \in L_f$, and pick a random (δ', r'_p) from the set $\{(\delta, r_p)\}$ to compute WI_3 and output it. Notice that F is identical to an honest prover of the WI protocol.
- T' and ϵ': We set $T' = T + T_{R^*} + T_S$ and $\epsilon' = \epsilon/3n$, where T is the size of the distinguisher, and T_{R^*} and T_S are the size of R^* and the sender S respectively.

By Lemma 2, we have a nearly-optimal (T', ϵ')-extractor Ext (with respect to $t = 2$) against circuits of size T'. Using this extractor, the simulator proceeds as follows.

$\mathsf{Sim}(R^*)$:

1. Sample the randomness r for R^* and generate $(y_0, y_1, \mathsf{WI}_1)$ by following the honest sender strategy.
2. Upon obtaining $(\mathsf{WI}_2, \mathsf{pk}_0, \mathsf{pk}_1) \leftarrow R^*(b, r, \tau, (\mathsf{WI}_1, y_0, y_1))$, write $\mathsf{pk}_i = (\mathsf{k}_i, x_i)$ for $i \in \{0,1\}$, and check if x_0 and x_1 share the same (y_0, y_1, a). If not, send $\perp$ to R^*; otherwise, compute WI_3 by following the honest sender strategy.
3. Compute $(z_0, z_1) \leftarrow \mathsf{Ext}(\mathsf{pk}_0, \mathsf{pk}_1, (\mathsf{WI}_1, y_0, y_1, r, \tau), \mathsf{WI}_3)$, and do the following:
   (a) If for both $i = 0, 1$, $(x_i = (y_0, y_1, a, i), z_i) \in R_{L_\Sigma}$, i.e., (a, i, z_i) is acceptable proof for $(y_0, y_1) \in L_f$, then send $\perp$ to R^*.

(b) If *at most* one witness of z_0 and z_1 is valid, then do: If z_0 is valid, i.e., $(x_0 = (y_0, y_1, a, b'), z_0) \in R_{L_\Sigma}$, set $b' = 0$; otherwise (z_1 is valid or neither is valid) set $b' = 1$. Send b' to $\mathcal{F}_{OT}$ and receive $m_{b'}$. Compute $c_0 \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_0, m_{b'})$ and $c_1 \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_1, m_{b'})$ in bitwise manner. Send WI_3, c_0, c_1 to R^*.

4. Output the view of R^*.

We are now ready to prove the (T, ϵ)-sender's security using hybrid argument. In the following, all hybrid experiments are further parameterized with the sender's input (m_0, m_1), the receiver's input b and the auxiliary input τ. Let $\mathsf{HYB}_0(\lambda)$ the ideal world experiment $\mathsf{IDEAL}_{\mathcal{F}_{OT}, \mathsf{Sim}^{R^*(\tau)}}(1^\lambda, m_0, m_1, b)$).

$\mathsf{HYB}_1(\lambda)$: It proceeds identically to the ideal world experiment except that the simulator encrypts $c_0 \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_0, m_0)$ and $c_1 \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_1, m_1)$ honestly in step 3(a).

Lemma 3. $\mathsf{HYB}_1(\lambda)$ *is statistically close to* $\mathsf{HYB}_0(\lambda)$.

Proof. It is sufficient to prove that the "if" condition in step 3(a), i.e., $(x_i = (y_0, y_1, a, i), w_i) \in R_{L_\Sigma}$ holds for both $i = 0$ and 1, occurs only with negligible probability.

To see this, suppose, toward a contradiction, that the simulator in its step 3 extracts (z_0, z_1) such that both $((y_0, y_1, a, 0), z_0) \in R_{L_\Sigma}$ and $((y_0, y_1, a, 1), z_1) \in R_{L_\Sigma}$ with probability p for some inverse polynomial p. Notice that by the special soundness of the underlying Σ-protocol, one can extract a witness δ for $(y_0, y_1) \in L_f$ from $(a, 0, z_0)$ and $(a, 1, z_1)$. This leads to the following cheating verifier V' of the underlying WI protocol: V' *externally* interacts with an honest prover P, and *internally* interacts with R^* in the same way as Sim except that the prover messages WI_1 and WI_3 are generated by the external prover P. When V' extracts two valid witnesses (z_0, z_1), it computes a witness δ for $(y_0, y_1) \in L_f$ and outputs δ at the end of the external execution. It is easy to verify that V' outputs δ with the same (non-negligible) probability p, breaking either the witness indistinguishability of the underlying WI protocol or the one-wayness of function f. □

According to the order of the $2n$ bit-wise encryptions in step 3(b) of Sim, we continue to construct $2n$ hybrid distributions as follows:

$\mathsf{HYB}_{1+i}(\lambda)$ ($i \in [n]$): This hybrid proceeds identically to the previous $\mathsf{HYB}_i(\lambda)$ except that, in the step 3(b) of Sim, the simulator encrypts $m'_i = m_{0,1} \| \cdots \| m_{0,i} \| m_{b',i+1} \| \cdots || m_{b',n}$ in bitwise manner under public key pk_0.

$\mathsf{HYB}_{n+1+i}(\lambda)$ ($i \in [n]$): This hybrid proceeds identically to the previous $\mathsf{HYB}_{n+i}(\lambda)$ except that, in the step 3(b) of Sim, the simulator encrypts $m'_i = m_{0,1} \| \cdots \| m_{0,i} \| m_{b',i+1} \| \cdots \| m_{b',n}$ in bitwise manner under public key pk_1.

It is easy to see that $\mathsf{HYB}_{2n+1}(\lambda) \equiv \mathsf{REAL}_{\Pi, R^*_\lambda(z)}(1^\lambda, m_0, m_1, b)$. The remaining task is to prove the following lemma.

Lemma 4. *For any $j \in [0, 1]$ and $i \in [n]$, and any T-sized distinguisher D, we have that:*

$$|\Pr[D(\mathsf{HYB}_{jn+1+i}(\lambda))] = 1 - \Pr[D(\mathsf{HYB}_{jn+i}(\lambda))] = 1| \leq \epsilon/3n.$$

Proof. We prove this lemma by contradiction. Assume there exists $j \in [0,1]$, $i \in [n]$ and a T-sized distinguisher D such that

$$|\Pr[D(\mathsf{HYB}_{jn+1+i}(\lambda))] = 1 - \Pr[D(\mathsf{HYB}_{jn+i}(\lambda))] = 1| > \epsilon/3n. \tag{9}$$

We now use such a distinguisher D to construct a circuit D' that breaks the (near) optimality property of the extractor Ext guaranteed by Lemma 2.

We start with a proof for the case $j = 0$ (and assume that the inequality (9) holds for $j = 0$). At a high level, D' simulates the view of R^* and then invokes D to distinguish a random ciphertext under the public key pk_0.

To put D' in the context of Lemma 2, we rewrite the simulator Sim using notations and algorithms $\mathcal{R}$, Gen^* and F as defined in the beginning of this subsection: Let $\bar{\mathsf{r}} = (b, r, \mathsf{WI}_1, y_0, y_1, \tau) \leftarrow \mathcal{R}$, where, by the definition of $\mathcal{R}$, (b, r, τ) and $(\mathsf{WI}_1, y_0, y_1)$ are generated in the same way as Sim; $\mathsf{KE.Gen}^*(\bar{\mathsf{r}})$ generates $\overline{\mathsf{pk}}_{[2]} = (\mathsf{pk}_0, \mathsf{pk}_1)$ (partial output of $R^*(\bar{\mathsf{r}})$); $F(\bar{\mathsf{r}})$ generates WI_3 as the honest third prover message of the WI protocol, and $\mathsf{Ext}(\mathsf{pk}_0, \mathsf{pk}_1, \bar{\mathsf{r}}, F(\bar{\mathsf{r}}))$ obtains two witnesses $\overline{z}_{[2]} = (z_0, z_1)$ for the instances on the two public keys. Given a ciphertext c' under the public key pk_0, D' (having m_0, m_1 hardwired and incorporating R^* and D) takes as input $(\overline{\mathsf{pk}}_{[2]}, c', \bar{\mathsf{r}}, F(\bar{\mathsf{r}}), \overline{z}_{[2]})$, and guesses the plaintext bit of c' in the following way:

$D'(\overline{\mathsf{pk}}_{[2]}, c', \bar{\mathsf{r}}, F(\bar{\mathsf{r}}), \overline{z}_{[2]})$:

1. Write $\bar{\mathsf{r}}$ as $(b, r, \mathsf{WI}_1, y_0, y_1, \tau)$ and compute $(\mathsf{WI}_2, \mathsf{pk}_0, \mathsf{pk}_1) \leftarrow R^*(b, r, \mathsf{WI}_1, y_0, y_1, \tau)$. Check if the two instances x_0 and x_1 on the two public keys are well-formed like Sim. If not, Send $\perp$ to R^*; otherwise, continue.
2. With the input (notice that $F(\bar{\mathsf{r}}) = \mathsf{WI}_3$) and the message WI_2 generated by R^*, compute two ciphertexts c_0 and c_1 to complete a session with R^* as follows: It first acts as $\mathsf{HYB}_{1+i}(\lambda)$ to compute c_0 and c_1 in bitwise manner, and then updates them by replacing the i-th ciphertext under pk_0 with c'.
3. Feed D with the view of R^*, and output whatever D outputs.

It's easy to check that the size of D' is less than $T + T_{R^*} + T_S$. Observe that if c' is the ciphertext of $m_{1,i}$ (the i-th bit of m_1) under pk_0, then the view of R^* generated by D' is identical to $\mathsf{HYB}_i(\lambda)$; if c' is the ciphertext of $m_{0,i}$ (the i-th bit of m_0) under pk_0, then the view of R^* generated by D' is identical to $\mathsf{HYB}_{i+1}(\lambda)$.

For $i \in [n]$, the only difference between $\mathsf{HYB}_{i+1}(\lambda)$ and $\mathsf{HYB}_i(\lambda)$ is that they encrypt different messages under pk_0 when $(x_0, z_0) \notin R_{L_\Sigma}$ (and they are identical when $(x_0, z_0) \in R_{L_\Sigma}$). Therefore, from inequality (9) as well as the construction of D', it follows

$$\left| \Pr\left[D'(\overline{\mathsf{pk}}_{[2]}, c', \bar{\mathsf{r}}, F(\bar{\mathsf{r}}), \overline{z}_{[2]}) = 1 \wedge (x_0, z_0) \notin R_L \;\middle|\; \begin{array}{r} \bar{\mathsf{r}} \leftarrow \mathcal{R}; \mathsf{pk}_{[2]} \leftarrow \mathsf{KE.Gen}^*(\bar{\mathsf{r}}) \\ \overline{z}_{[2]} \leftarrow \mathsf{Ext}(\mathsf{pk}_{[2]}, \bar{\mathsf{r}}, F(\bar{\mathsf{r}})); \\ c' \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_0, 0); \end{array} \right] - \Pr\left[D(\overline{\mathsf{pk}}_{[2]}, c', \bar{\mathsf{r}}, F(\bar{\mathsf{r}}, \overline{z}_{[2]}) = 1 \wedge (\mathsf{x}_0, w_0') \notin R_L \;\middle|\; \begin{array}{r} \bar{\mathsf{r}} \leftarrow \mathcal{R}; \overline{\mathsf{pk}}_{[2]} \leftarrow \mathsf{KE.Gen}^*(\bar{\mathsf{r}}) \\ \overline{z}_{[2]} \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}_{[2]}, \bar{\mathsf{r}}, F(\bar{\mathsf{r}})); \\ c' \leftarrow \mathsf{KE.Enc}(\mathsf{pk}_0, 1); \end{array} \right] \right| > \epsilon/3n.$$

We thus arrive at a contradiction with lemma 2. For $j = 1$, one can prove this lemma in a similar way. □

In sum, we have that, for any T-size distinguisher $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$, any $m_0, m_1 \in \{0,1\}^n, b \in \{0,1\}$,

$$\Big|\Pr[D(\mathsf{REAL}_{\Pi,R^*_\lambda(\tau)}(1^\lambda, m_0, m_1, b))] = 1 - \Pr[D(\mathsf{IDEAL}_{\mathcal{F}_{OT},\mathsf{Sim}^{R^*_\lambda(\tau)}}(1^\lambda, m_0, m_1, b))] = 1\Big| \leq 2\epsilon/3 + negl.$$

which concludes the proof of (T, ϵ)-simulatable security for the sender. □

5 Three-round weak zero-knowledge argument of knowledge

In this section, we construct a delayed-input (T, ϵ)-zero-knowledge argument satisfying adaptive argument of knowledge, which is based on the following ingredients:

- A 3-round OT $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ presented in Fig.4.
- A one-way function f.
- A knowledge encryption scheme $(\mathsf{KE.Gen}, \mathsf{KE.Enc}, \mathsf{KE.Dec})$ for language L'_f.
- A 3-round public-coin WI protocol $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ with special-soundness property for language L_{pk}.
- A Σ-protocol (α, β, γ) with 1-bit challenge space for an NP language L.

where L'_f, L_{pk} are defined as follows:

$$L'_f : \{y | \exists \delta \ s.t. \ f(\delta) = y\}$$
$$L_{pk} : \{\mathsf{pk}_0, \mathsf{pk}_1 | \exists b, \mathsf{sk}_b, r_{\mathsf{KE}}, (y_b, \delta_b) \in L'_f \ s.t. \ (\mathsf{pk}_b, \mathsf{sk}_b) = \mathsf{KE.Gen}(1^\lambda, y_b, \delta_b; r_{\mathsf{KE}})\}$$

We formally present our construction in Fig.5.

Theorem 4. *Assuming the existence of two-round OT protocol with game-based security (against polynomial-time adversaries), there exists a three-round delayed-input (T, ϵ)-zero-knowledge adaptive argument of knowledge. Furthermore, the same protocol also satisfies witness hiding and quasi-polynomial simulatable zero knowledge under the same assumption.*

We provide the proof of Theorem 4 in Appendix A.

Delayed-input (T, ϵ)-Zero-knowledge Argument of Knowledge

Prover Input: $(x, w) \in R_L$.

- **Prover Message:** Run OT_1 λ times in parallel and obtain $\{\mathsf{ot}_i\}_{i\in[\lambda]}$. Sample δ_0,δ_1 $\leftarrow \{0,1\}^\lambda$ and compute $y_0 = f(\delta_0)$, $y_1 = f(\delta_1)$. Generate two knowledge encryption public keys $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KE.Gen}(1^\lambda, y_0, \delta_0)$, $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KE.Gen}(1^\lambda, y_1, \delta_1)$ and the first message WI_1 of WI for statement $(\mathsf{pk}_0, \mathsf{pk}_1) \in L_{pk}$.
  Send $(\{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1)$.
- **Verifier Message:** For each $i \in [\lambda]$, sample $\beta_i \leftarrow \{0,1\}$ and compute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_{2,i}(\beta_i)$ independently. Generate the second message WI_2 of WI.
  Send $(\{\mathsf{ot}_{2,i}\}_{i\in[\lambda]}, \mathsf{WI}_2)$.
- **Prover Message:** For each $i \in [\lambda]$, generate two Σ-proofs with the same first message (i.e. $(\alpha_i, 0, \gamma_{i,0}), (\alpha_i, 1, \gamma_{i,1})$). For $b = 0, 1$, encrypt $\gamma_{i,b}$ using both of $\mathsf{pk}_0, \mathsf{pk}_1$ separately to obtain $C_{i,b}$, i.e. $C_{i,b} = (\mathsf{KE.Enc}(\mathsf{pk}_0, \gamma_{i,b}), \mathsf{KE.Enc}(\mathsf{pk}_1, \gamma_{i,b}))$. Let $\gamma'_{i,b}$ be the message consisting of $\gamma_{i,b}$ and the randomness used in computing $C_{i,b}$. Compute $\mathsf{ot}_{3,i} \leftarrow \mathsf{OT}_{3,i}(\gamma'_{i,0}, \gamma'_{i,1})$. Generate the third message WI_3 of WI.
  Send $(x, \{\alpha_i, C_{i,0}, C_{i,1}, \mathsf{ot}_{3,i}\}_{i\in[\lambda]}, \mathsf{WI}_3)$.
- **Verifier's Output:** Recover γ'_{i,β_i} from OT, output 1 if for all $i \in [\lambda]$, $(\alpha_i, \beta_i, \gamma_{i,\beta_i})$ and $\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3$ are acceptable proofs and C_{i,β_i} is indeed the encryptions of γ_{i,β_i} (using the randomness contained in γ'_{i,β_i}).

Fig. 5: Three-round Argument System for NP

6 Two-party Secure Computation

Equipped with the three-round OT and zero knowledge argument constructed in previous sections, we now follow the GMW paradigm [GMW87] to give a three-round protocol for weakly secure two-party computation for independent-input functionalities. We use the following ingredients in our construction:

- A 3-round OT $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ (presented in Fig.4).
- A 3-round delayed-input weak zero knowledge argument $(\mathsf{ZK}_1, \mathsf{ZK}_2, \mathsf{ZK}_3)$ (presented in Fig.5) for language L_{2pc}.
- A garbling circuit scheme $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$,

where L_{2pc} is defined as follows: $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{1,i}, \mathsf{ot}_{2,i}, \mathsf{ot}_{3,i}\}_{i\in[n]}) \in L_{2pc}$ if and only if there exists a random tape for the honest sender (on input $\mathsf{ot}_{2,i}$) to generate messages $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{c_{i,b} = \mathsf{KE.Enc}(\mathsf{pk}^1_{i,b}, \mathsf{lab}^y_{i,b})\}_{i\in[n],b\in\{0,1\}})$($c_{i,b}$ is the ciphertexts in $\mathsf{ot}_{3,i}$ under the public key $\mathsf{pk}^1_{i,b}$ contained in $\mathsf{ot}_{2,i}$).

We assume that the independent-input functionality C maps (x, y) of length $2n$ to a string of length n. The protocol is formally presented in Fig.6.

Theorem 5. *Assuming the existence of two-round OT protocol with game-based security (against polynomial-time adversaries), there exists a three-round two-party computation protocol for independent-input functionalities that achieves (T, ϵ)-security against*

3-round Two-party Weak Secure Computation

Sender Input: $x \in \{0,1\}^n$
Receiver Input: $y \in \{0,1\}^n$

- **Sender Message:** Run OT_1 λ times in parallel and obtain $\{\mathsf{ot}_i\}_{i\in[\lambda]}$. Generate the first message ZK_1.
  Send $(\{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1)$.
- **Receiver Message:** Generate the second message ZK_2. For each $i \in [n]$,compute $\mathsf{ot}_{2,i} \leftarrow \{\mathsf{OT}_{2,i}(y_i)\}_{i\in[n]}$ independently where y_i is the i-th bit of y.
  Send $(\{\mathsf{ot}_{2,i}\}_{i\in[n]}, \mathsf{ZK}_2)$.
- **Sender Message:** Use GC to generate the garbled circuit $\hat{C}$ along with labels $\{\mathsf{lab}^x_{i,b}\}_{i\in[n],b\in\{0,1\}}, \{\mathsf{lab}^y_{i,b}\}_{i\in[n],b\in\{0,1\}}$ for functionality C. Compute $\mathsf{ot}_{3,i} \leftarrow \mathsf{OT}_{3,i}(\mathsf{lab}^y_{i,0}, \mathsf{lab}^y_{i,1})$. Compute ZK_3 for $(\hat{C},\{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]},\{\mathsf{ot}_{1,i},\mathsf{ot}_{2,i},\mathsf{ot}_{3,i}\}_{i\in[n]}) \in L_{2pc}$.
  Send $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{3,i}\}_{i\in[n]}, \mathsf{ZK}_3)$.
- **Receiver's Output:** Recover lab^y_{i,y_i} from OT, and check if $(\mathsf{ZK}_1, \mathsf{ZK}_2, \mathsf{ZK}_3)$ is acceptable. If not, output $\perp$; otherwise, output $\hat{C}(\{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{lab}^y_{i,y_i}\}_{i\in[n]})$.

Fig. 6: 3-round Two-party Weak Secure Computation

malicious receiver and standard security against malicious sender. Furthermore, the same protocol also achieves quasi-polynomial simulatable security against malicious receiver under the same assumption.

We provide the proof of Theorem 5 in Appendix B.

7 More Applications

In this section we present direct applications of our results in previous sections to various protocols, including extractable commitment, selective opening secure commitment and concurrent zero knowledge argument in the BPK model. Compared with existing protocols, all our new constructions only rely on two-round OT with game-based security. Since one can prove the security of these new constructions using essentially the same security proof strategies in [JKKR17, Den20], we will not repeat these proofs here.

The work [JKKR17] provides a transformation of non-interactive commitment into a three-round extractable commitment via three-round weak zero knowledge argument of knowledge. When using our construction of (T, ϵ)-zero knowledge argument of knowledge in their transformation, we have the following result.

Theorem 6. *Assuming the existence of two-round OT with game-based security (against polynomial-time adversaries), there exists a three-round extractable commitment scheme.*

The commitment with (T, ϵ)-security under selective opening attack and concurrent (T, ϵ)-zero knowledge argument (in the BPK model) in [Den20] are constructed from Rabin encryption scheme (based on hardness of Factoring). We can also replace the Rabin encryption scheme with our knowledge encryption (and revise their protocol accordingly so that the simulation can go through with a witness for the instance on the public key of knowledge encryption), and obtain the following result.

Theorem 7. *Assuming the existence of two-round OT with game-based security (against polynomial-time adversaries), there exist:*

1. *Two-round commitment scheme with (T, ϵ)-security under selective opening attacks.*
2. *Three-round concurrent (T, ϵ)-zero knowledge argument with concurrent soundness in the BPK model, which also satisfies concurrent witness hiding in the same model.*
3. *All above protocols satisfy (fully) quasi-polynomial simulatable security.*

Acknowledgments. We would like to thank the anonymous reviewers for their valuable suggestions. We are supported by the National Natural Science Foundation of China (Grant No. 61932019 and No. 61772522), the Key Research Program of Frontier Sciences, CAS (Grant No. QYZDB-SSW-SYS035) and Beijing Natural Science Foundation (Grant No. M22003).

References

[ABOR00] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In *Automata, Languages and Programming - ICALP'00*, LNCS 1853, pages 463–474. Springer, 2000.

[AIR01] Bill Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology - EUROCRYPT'01*, LNCS 2045, pages 119–135. Springer, 2001.

[AJ17] Prabhanjan Ananth and Abhishek Jain. On secure two-party computation in three rounds. In *Theory of Cryptography - TCC'17*, LNCS 10677, pages 612–644. Springer, 2017.

[APV05] Joël Alwen, Giuseppe Persiano, and Ivan Visconti. Impossibility and feasibility results for zero knowledge with public keys. In *Advances in Cryptology – CRYPTO'05*, LNCS 3621, pages 135–151. Springer, 2005.

[BBK+16] Nir Bitansky, Zvika Brakerski, Yael Kalai, Omer Paneth, and Vinod Vaikuntanathan. 3-message zero knowledge against human ignorance. In *Theory of Cryptography - TCC'16*, LNCS 9985, pages 57–83. Springer, 2016.

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In *Proceedings of the 45th Annual ACM Symposium on the Theory of Computing - STOC'14*, pages 505–514. ACM Press, 2014.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private ot from lwe. In *Theory of Cryptography - TCC'18,*, LNCS 11240, pages 370–390. Springer, 2018.

[BGJ+18] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal MPC. In *Advances in Cryptology - CRYPTO'18*, LNCS 10992, pages 459–487. Springer, 2018.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security - CCS'12*, pages 784–796. ACM press, 2012.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC'19*, pages 1091–1102. ACM press, 2019.

[BL18] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In *Advances in Cryptology - EUROCRYPT'18*, LNCS 10821, pages 500–532. Springer, 2018.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of international congress of mathematicians - ICM'86*, 1986.

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In *Theory of Cryptography Conference - TCC'12*, LNCS 7194, pages 190–208. Springer, 2012.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science - FOCS'01*, pages 136–145. IEEE Computer Society, 2001.

[CCG+20] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Round optimal secure multiparty computation from minimal assumptions. In *Theory of Cryptography - TCC'20*, LNCS 12551, pages 291–319. Springer, 2020.

[CCG+21] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Oblivious transfer from trapdoor permutations in minimal rounds. In *Theory of Cryptography - TCC'21*, LNCS 13043, pages 518–549. Springer, 2021.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In *Theory of Cryptography - TCC'15*, LNCS 9014, pages 66–92. Springer, 2015.

[COSV17] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Round-optimal secure two-party computation from trapdoor permutations. In *Theory of Cryptography - TCC'17*, LNCS 10677, pages 678–710. Springer, 2017.

[Den20] Yi Deng. Individual simulations. In *Advances in Cryptology – ASIACRYPT'20*, LNCS 12493, pages 805–836. Springer, 2020.

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In *Advances in Cryptology - EUROCRYPT'20*, LNCS 12106, pages 768–797. Springer, 2020.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *Journal of the ACM*, 50(6):852–921, 2003.

[FMV19] Daniele Friolo, Daniel Masny, and Daniele Venturi. A black-box construction of fully-simulatable, round-optimal oblivious transfer from strongly uniform key agreement. In *Theory of Cryptography - TCC'19*, LNCS 11891, pages 111–130. Springer, 2019.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology - CRYPTO'84*, LNCS 196, pages 10–18. Springer, 1985.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing - STOC'13*, page 467–476. ACM press, 2013.

[GIKM98] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing - STOC'98*, page 151–160. ACM press, 1998.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In *Advances in Cryptology – EUROCRYPT'20*, LNCS 12107, pages 668–699. Springer, 2020.

[GK96a] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[GK96b] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science - FOCS'00*, pages 325–335. IEEE Computer Society, 2000.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run turing machines on encrypted data. In *Advances in Cryptology – CRYPTO'13*, LNCS 8043, pages 536–553. Springer, 2013.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In *Advances in Cryptology - EUROCRYPT'16*, LNCS 9666, pages 448–476. Springer, 2016.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing - STOC'87*, pages 218–229. ACM press, 1987.

[Gol04] O. Goldreich. *Foundations of Cryptography*, volume Basic Applications. Cambridge University Press, 2004.

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In *Advances in Cryptology - EUROCRYPT'18*, LNCS 10821, pages 468–499. Springer, 2018.

[HHPV18] Shai Halevi, Carmit Hazay, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Round-optimal secure multi-party computation. In *Advances in Cryptology - CRYPTO'18*, LNCS 10992, pages 488–520. Springer, 2018.

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, 2012.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In *Advances in Cryptology - CRYPTO'17*, LNCS 10402, pages 158–189. Springer, 2017.

[Kiy21] Susumu Kiyoshima. Black-box impossibilities of obtaining 2-round weak ZK and strong WI from polynomial hardness. In *Theory of Cryptography - TCC'21*, volume

13042 of *LNCS*, pages 369–400. Springer, 2021.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In *Advances in Cryptology - EUROCRYPT'18*, LNCS 10822, pages 34–65. Springer, 2018.

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *Advances in Cryptology - CRYPTO'04*, LNCS 3152, pages 335–354. Springer, 2004.

[KR09] Yael Tauman Kalai and Ran Raz. Probabilistically checkable arguments. In *Advances in Cryptology - CRYPTO'09*, LNCS 5677, pages 143–159. Springer, 2009.

[LP09] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. *J. Cryptol.*, 22(2):161–188, apr 2009.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms - SODA'01*, pages 448–457. Society for Industrial and Applied Mathematics, 2001.

[ORS15] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In *Advances in Cryptology – CRYPTO'15*, LNCS 9216, pages 339–358. Springer, 2015.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — EUROCRYPT '99*, LNCS 1592, pages 223–238. Springer, 1999.

[Pas03] R. Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *Advances in Cryptology - EUROCRYPT'03*, LNCS 2656, pages 160–176. Springer, 2003.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *Proceedings of the 43th Annual IEEE Symposium on Foundations of Computer Science - FOCS'02*, pages 366–375. IEEE Computer Society, 2002.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Advances in Cryptology - CRYPTO'08*, LNCS 5157, pages 554–571. Springer, 2008.

[Rot13] Ron Rothblum. On the circular security of bit-encryption. In *Theory of Cryptography - TCC'13*, volume 7785 of *LNCS*, pages 579–598. Springer, 2013.

[Xia11] David Xiao. (nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In *Theory of Cryptography - TCC'11*, LNCS 6597, pages 541–558. Springer, 2011.

[Xia13] David Xiao. Errata to (nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In *Theory of Cryptography - TCC'13*, LNCS 7785, pages 721–722. Springer, 2013.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science - FOCS'86*, pages 162–167. IEEE, 1986.

Appendix

A Proof of Theorem 4

In this subsection, we prove that the protocol presented in Fig.5 is a delayed-input (T, ϵ)-zero-knowledge adaptive argument of knowledge. The furthermore part of Theorem 4 follows from the fact that weak zero knowledge implies witness hiding [JKKR17], and the second part of Lemma 2.

Again, note that all primitives used in Fig 5 can be built from two-round OT protocol with game-based security (as defined in Definition 6).
The completeness of the protocol is obvious.

Adaptive argument of knowledge. Suppose that a cheating prover P^* interacts with the verifier and generates an acceptable transcript with some non-negligible probability. Consider the following extractor E^{P^*}. It plays the role of the verifier and interacts with P^*. If in the first run of the protocol E obtains an acceptable transcript tr for a statement x (denote by pk_0 and pk_1 the two public keys generated by P^*), then it rewinds P^* to the point where P^* just sent out its first message, and repeats to compute the second verifier message (always following the honest verifier strategy) with fresh randomness and send it to P^* until another acceptable transcript tr' is obtained. E computes a valid secret key sk_j for some $j \in \{0, 1\}$ from the two acceptable $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ and $(\mathsf{WI}_1, \mathsf{WI}'_2, \mathsf{WI}'_3)$ contained in tr and tr' respectively (except for exponentially small probability, $\mathsf{WI}_2 \neq \mathsf{WI}'_2$), and then, for $i \in [\lambda], b \in \{0, 1\}$, use sk_j to decrypt the corresponding ciphertexts $C_{i,b}$ under the public key pk_j in the transcript tr to obtain $\gamma_{i,b}$. If there exists $i' \in [\lambda]$ such that $(\alpha_{i'}, 0, \gamma_{i',0})$, $(\alpha_{i'}, 1, \gamma_{i',1})$ are both acceptable proofs for statement $x \in L$, then E extracts $w \in R_L(x)$ from them and outputs (x, w); otherwise, E outputs $\perp$.

Note that, from the transcript tr, for each $i \in [\lambda]$, E already obtained an acceptable $(\alpha_i, \beta_i, \gamma_{i,\beta_i})$ via the underlying three-round OT, where β_i is the bit encoded in the i-th OT receiver message. Furthermore, since the two ciphertexts of γ_{i,β_i} in tr are correct[13], E can also obtain the acceptable $(\alpha_i, \beta_i, \gamma_{i,\beta_i})$ by decryption using sk_j. Hence, if E^{P^*} finally outputs $\perp$ (i.e., fails to obtain $\gamma_{i,1-\beta_i}$ by decryption) with non-negligible probability, then we have the following algorithm E'^{P^*} that can break the receiver security of the underlying three-round OT: E'^{P^*} proceeds the same as E^{P^*} except that, in the first run of the protocol, all the OT receiver message $\{\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_{2,i}(\beta_i)\}$ are generated by an external OT challenger. After obtaining all $\gamma_{i,b}$ for each $i \in [\lambda], b \in \{0, 1\}$ by decryption, if $(\alpha_i, 0, \gamma_{i,0})$ is an acceptable proof, set $\beta_i = 0$; otherwise, set $\beta_i = 1$. Thus, we conclude that E^{P^*} outputs $\perp$ with negligible probability and our protocol satisfies adaptive argument of knowledge property.

Delayed-input (T, ϵ)-zero-knowledge. At a high level, our simulator is similar to the one presented in the proof of (T, ϵ)-simulatable security for the OT sender. It first tries to apply the nearly optimal extractor to extract all β_i encoded in the second OT message $\{\mathsf{ot}_{2,i}\}$ from a cheating verifier V^*. Once β_i is obtained, the simulator can generate an

[13] Notice that, the honest verifier is supposed to retrieve the corresponding randomness used in these ciphertexts from the last OT message and check if they are correct

acceptable $\alpha_i, \beta_i, \gamma_{i,\beta_i}$ for $x \in L$ and compute the third prover message accordingly; If it fails to extract anything for $\{\mathsf{ot}_{2,i}\}$, then encrypts dummy messages (under both the prover's public keys and the public keys contained in $\{\mathsf{ot}_{2,i}\}$) in the third prover message.

We fix the security parameter λ, the polynomial T and the inverse polynomial ϵ. For a given (malicious) verifer V^*, we formally define:

- $\mathcal{R}$: The distribution over the input of V^*. This is a joint distribution of two random variables: the verifier's randomness r and the first message of the prover P.
- $\mathsf{KE.Gen}^*$: On input $(r, \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1) \leftarrow \mathcal{R}$, compute $(\mathsf{WI}_2, \{\mathsf{ot}_{2,i}\}_{i\in\lambda}) \leftarrow V^*(r, \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1)$. Write $\mathsf{ot}_{2,i} = (\mathsf{WI}_2^i, \mathsf{pk}_0^i, \mathsf{pk}_1^i)$ and output $\{\mathsf{pk}_0^i, \mathsf{pk}_1^i\}_{i\in[\lambda]}$.
- F: On input $(r, \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1) \leftarrow \mathcal{R}$, compute $(\mathsf{WI}_2, \{\mathsf{ot}_{2,i}\}_{i\in\lambda}) \leftarrow V^*(r, \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1)$. Compute all possible witnesses and the prover's randomness that are consistent with the first two messages $(\mathsf{WI}_1, \mathsf{WI}_2)$ for statement $(\mathsf{pk}_0, \mathsf{pk}_1) \in L_{pk}$, and pick a random one to compute WI_3 and output it. For each $i \in [\lambda]$, F computes and outputs WI_3^i in OT in the same way as section 4. Output WI_3 and $\{\mathsf{WI}_3^i\}_{i\in\lambda}$
- T' and ϵ': We set $T' = T + T_{V^*} + T_P$ and $\epsilon' = \epsilon/3n\lambda$, where T is the size of the distinguisher, T_{V^*} and T_P are the size of V^* and the sender P respectively and n is the length of message transferred by a single run of the underlying three-round OT.

By Lemma 2, we have a nearly-optimal (T', ϵ')-extractor Ext (with respect to $t = 2\lambda$) against circuits of size T'. Denote by l the length of the third message γ of the Σ-protocol for proving $x \in L$. The simulator proceeds as follows.

$\mathsf{Sim}(x)$:

1. Sample the randomness r for V^* and generate $(\{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1)$ following the honest sender strategy.
2. Upon receiving $(\mathsf{WI}_2, \{\mathsf{ot}_{2,i}\}_{i\in[\lambda]}) \leftarrow V^*(r, \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1)$, Check if $\{\mathsf{ot}_{2,i}\}_{i\in[\lambda]}$ are well-formed. If not, abort and output the view of V^*; otherwise, write $\mathsf{ot}_{1,i} = (y_0^i, y_1^i, \mathsf{WI}_1^i)$ and $\mathsf{ot}_{2,i} = (\mathsf{WI}_2^i, \mathsf{pk}_0^i, \mathsf{pk}_1^i)$, and compute WI_3 and $\{\mathsf{WI}_3^i\}_{i\in[\lambda]}$ following the honest prover strategy.
3. Compute $\{z_0^i, z_1^i\}_{i\in[\lambda]} \leftarrow \mathsf{Ext}(\{\mathsf{pk}_0^i, \mathsf{pk}_1^i\}_{i\in[\lambda]}, (r, \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1), \mathsf{WI}_3, \{\mathsf{WI}_3^i\})$. Write $\mathsf{pk}_0^i = (\mathsf{k}_0^i, x_0^i), \mathsf{pk}_1^i = (\mathsf{k}_1^i, x_1^i)$ and do the follows:
   (a) If there exists $i \in [\lambda]$ such that for both $b = 0, 1$, $(x_b^i = (y_0^i, y_1^i, a^i), z_b^i) \in R_{L_\Sigma}$, i.e. (a^i, b, z_b^i) is an acceptable proof for $(y_0^i, y_1^i) \in L_f$, abort. If else, continue.
   (b) For each $i \in [\lambda]$, if $(x_0^i = (y_0^i, y, 1_1^i, a^i, 0), z_0^i) \in R_{L_\Sigma}$, then set $\beta_i = 0$; otherwise (z_0^i is acceptable or neither is acceptable), set $\beta_i = 1$. Use the HVZK simulator of the Σ-protocol to generate an acceptable $(\alpha_i, \beta_i, \gamma_{i,\beta_i})$ for $x \in L$. Compute $C_{i,\beta_i} = (\mathsf{KE.Enc}(\mathsf{pk}_0, \gamma_{i,\beta_i}), \mathsf{KE.Enc}(\mathsf{pk}_1, \gamma_{i,\beta_i}))$ and $C_{i,1-\beta_i} = (\mathsf{KE.Enc}(\mathsf{pk}_0, 0^l), \mathsf{KE.Enc}(\mathsf{pk}_1, 0^l))$. Let γ'_{i,β_i} be the message consisting of γ_{i,β_i} and the randomness used in computing C_{i,β_i} and $\gamma'_{i,1-\beta_i} := 0^n$. Compute and set $\mathsf{ot}_{3,i} = (\mathsf{WI}_3^i, \mathsf{KE.Enc}(\mathsf{pk}_0^i, \gamma'_{i,0}), \mathsf{KE.Enc}(\mathsf{pk}_1^i, \gamma'_{i,1}))$.

4. Output the view of V^*.

To prove the delayed-input (T, ϵ)-zero-knowledge of our protocol, we construct a sequence of hybrid simulators and show that any two neighboring hybrid simulators are indistinguishable.

.

$\mathsf{HSim}_1(x, w)$ is identical to Sim except that for each $i \in [\lambda]$, it generates $(\alpha_i, \beta_i, \gamma_{i,\beta_i})$ for $x \in L$ using witness w in step 3(b). From the HVZK property of Σ-protocol, we have $\mathsf{HSim}_1(x, w) \stackrel{c}{\approx} \mathsf{Sim}(x)$.

$\mathsf{HSim}_2(x, w)$ is identical to $\mathsf{HSim}_1(x, w)$ except that for each $i \in [\lambda]$, in step 3(b), it honestly generates $\gamma_{i,1-\beta_i}$ using witness w and computes $C_{i,1-\beta_i} = (\mathsf{KE.Enc}(\mathsf{pk}_0, \gamma_{i,1-\beta_i}), \mathsf{KE.Enc}(\mathsf{pk}_1, \gamma_{i,1-\beta_i}))$.

Lemma 5. $\mathsf{HSim}_2(x, w) \stackrel{c}{\approx} \mathsf{HSim}_1(x, w)$

Proof. Note that the only difference between $\mathsf{HSim}_1(x, w)$ and $\mathsf{HSim}_2(x, w)$ is the plaintext encrypted in $C_{i,1-\beta_i}$ in step 3(b). Again, we prove this lemma by hybrid argument. Consider the following sub-hybrid simulators.

The first sub-hybrid proceeds the same as $\mathsf{HSim}_1(x, w)$ except that it generates pk_0 honestly and uses the corresponding witness $(\mathsf{pk}_0, \mathsf{pk}_1) \in L_{pk}$ to compute WI. This sub-hybrid simulator is indistinguishable from $\mathsf{HSim}_1(x, w)$ because of the witness indistinguishability of WI.

The next sub-hybrid proceeds the same as the above sub-hybrid except that it generates the ciphertexts $C_{i,1-\beta_i}$ under public-key pk_1 in step 3(b) in the same way as the above $\mathsf{HSim}_2(x, w)$. This sub-hybrid simulator is indistinguishable from the above sub-hybrid due to the indistinguishability of knowledge encryption.

Next we consider a sub-hybrid that proceeds the same as the above sub-hybrid except that it generates pk_1 honestly and uses the corresponding witness $(\mathsf{pk}_0, \mathsf{pk}_1) \in L_{pk}$ to compute WI. Again, this sub-hybrid simulator is indistinguishable from the above sub-hybrid due to the witness indistinguishability of WI.

The final sub-hybrid proceeds the same as the above sub-hybrid except that it generates the ciphertext $C_{i,1-\beta_i}$ under public-key pk_0 in step 3(b) in the same way as $\mathsf{HSim}_2(x, w)$. Again, this sub-hybrid simulator is indistinguishable from the above sub-hybrid due to the indistinguishability of knowledge encryption.

Due to the witness indistinguishability of WI, the above (final) sub-hybrid is indistinguishable from $\mathsf{HSim}_2(x, w)$, which concludes this lemma. □

$\mathsf{HSim}_3(x, w)$ is identical to $\mathsf{HSim}_2(x, w)$ except that in step 3(a), if there exists $i \in [\lambda]$ such that for both $b = 0, 1$, $(x_b^i = (y_0^i, y_1^i, a^i), z_b^i) \in R_{L_\Sigma}$, i.e. (a^i, b, z_b^i) is an acceptable proof for $(y_0^i, y_1^i) \in L_f$, it generates the last round message honestly.

Lemma 6. $\mathsf{HSim}_3(x, w)$ *is statistically close to* $\mathsf{HSim}_2(x, w)$.

Proof. The proof of this lemma is similar to lemma 3. One can prove that the "if" condition in step 3(a), i.e., there exists an $i \in [\lambda]$ s.t. $(x_b^i = (y_0^i, y_1^i, a^i, b), z_b^i) \in R_{L_\Sigma}$ holds for both $b = 0$ and 1, occurs only with negligible probability, since otherwise we can construct a verifier of the WI protocol from V^* that breaks either the witness indistinguishability of the WI protocol or the one-wayness of the function f. □

$\mathsf{HSim}_{3+k}(x,w)(i \in [\lambda])$ is identical to $\mathsf{HSim}_{2+k}(x,w)$ except that it generates $\gamma'_{k,1-\beta_k}$ and $\mathsf{ot}_{3,i}$ in step 3(b) in the same way as honest prover.

Observe that $\mathsf{HSim}_{3+\lambda}(x,w)$ is identical to the real execution between $P(x,w)$ and V^*. To complete the proof we prove following lemma.

Lemma 7. *For any $k \in [\lambda]$, and any T-size distinguisher D, we have that:*

$$|\Pr[D(\mathsf{HSim}_{3+k}(x,w))] = 1 - \Pr[D(\mathsf{HSim}_{2+k}(x,w))] = 1| \leq 2\epsilon/3\lambda$$

One can use the same proof strategy for Lemma 4 to prove this lemma, and here we just present a proof sketch. Observe that for each $k \in [\lambda]$, the difference between $\mathsf{HSim}_{3+k}(x,w)$ and $\mathsf{HSim}_{2+k}(x,w)$ is that, in the k-th run of the underlying three-round OT, the $2n$ ciphertexts in $\mathsf{ot}_{3,k}$ (under the public keys contained in $\mathsf{ot}_{2,k}$) are encryptions of different plaintexts.

We first construct $2n$ hybrids gradually moving from $\mathsf{HSim}_{2+k}(x,w)$ to $\mathsf{HSim}_{3+k}(x,w)$, each of them acting as the previous one but making a change on a single plaintext. One can prove that for any two neighboring hybrids, any distinguisher D of size T, the distinguishing advantage of D is less than $\epsilon/3\lambda n$. This follows from similar reasoning underlying the proof of Lemma 4: Otherwise, for any (x,w), any distinguisher D of size T that can tell (any) two neighboring hybrids apart, we can construct a distinguisher D' of size $T' = T + T_{V^*} + T_P$ (having (x,w) hardwired and incorporating D and V^*) that contradicts with Lemma 2 with parameter T' and $\epsilon' = \epsilon/3\lambda n$. Since there are $2n$ hybrids in total, we conclude Lemma 7.

From Lemma 7, it follows that, for any T-size distinguisher D and $(x,w) \in R_L$,

$$\left|\Pr[D(\mathsf{Sim}(x,w))] = 1 - \Pr[D(\mathsf{View}_{V^*}^{P(x,w)})] = 1\right| \leq 2\epsilon/3 + negl \leq \epsilon.$$

This completes the proof of the delayed-input (T,ϵ)-zero-knowledge property.

B Proof of Theorem 5

In this subsection, we show that the protocol presented in Fig 6 is a three-round two-party computation protocol for *independent-input functionalities* that achieves (T,ϵ)-security against malicious receiver and standard security against malicious sender. The furthermore part of Theorem 5 follows from the second part of Lemma 2.

Again, note that all primitives used in Fig 6 could be constructed by two-round OT protocol with computational game-based security (seeing Definition 6).

Security against malicious sender. At a high level, the simulator acts as honest receiver R except that it generates $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_{2,i}(0)$ to interact with the cheating sender S^*. It extracts the witness from the (weak) ZK proof and retrieves x' from it, and sends x' to functionality $\mathcal{F}_{2pc}$ to finish the simulation. The simulator is constructed as follows:

$\mathsf{Sim}^{S^*(\tau)}$

1. Run $S^*(x,r,\tau)$ to obtain the first round message $(\{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1)$ with randomness r and auxiliary input τ.

2. Generate the second message ZK_2 of the ZK protocol. For each $i \in [n]$, compute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_{2,i}(0)$ independently. Send $(\{\mathsf{ot}_{2,i}\}_{i\in[n]}, \mathsf{ZK}_2)$ to S^*.
3. Upon receiving $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{3,i}\}_{i\in[n]}, \mathsf{ZK}_3)$, check if $(\mathsf{ZK}_1, \mathsf{ZK}_2, \mathsf{ZK}_3)$ is an acceptable proof and $\{\mathsf{ot}_{1,i}, \mathsf{ot}_{2,i}, \mathsf{ot}_{3,i}\}$ are well-formed (i.e. the WI proofs in OT are acceptable). If not, send $\perp$ to ideal functionality $\mathcal{F}_{2pc}$ and output the view of S^*. Otherwise, go to next step.
4. Use the extractor of the ZK protocol (by generating a fresh second ZK message along with a fresh second OT message in each invocation of S^*) to extract the witness for $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{1,i}, \mathsf{ot}_{2,i}, \mathsf{ot}_{3,i}\}_{i\in[n]}) \in L_{2pc}$. Retrieve x' from the extracted witness and send it to ideal functionality $\mathcal{F}_{2pc}$. Output the view of S^*.

We now prove the receiver security by hybrid argument. In the following, all hybrid experiments are further parameterized by the sender's input x, the receiver's input y and the auxiliary input τ. Let $\mathsf{HYB}_0(\lambda)$ the ideal world experiment. Consider the following hybrid $\mathsf{HYB}_1(\lambda)$.

$\mathsf{HYB}_1(\lambda)$ is identical to the ideal world experiment except that the simulator always generates $\mathsf{ot}_{2,i} \leftarrow \{\mathsf{OT}_{2,i}(y_i)\}_{i\in[n]}$ independently, where y_i is the i-th bit of y. From the receiver security of the 3-round OT, we have that $\mathsf{HYB}_1(\lambda) \overset{c}{\approx} \mathsf{HYB}_0(\lambda)$.

Denote by $\mathsf{HYB}_2(\lambda)$ the real world experiment. Then it is easy to verify that the view of S^* in its first execution (before rewinding) in $\mathsf{HYB}_1(\lambda)$ is identical to the one in real world. By the adaptive argument of knowledge property, if E extracts the witness successfully, then the output of R in $\mathsf{HYB}_1(\lambda)$ is identical to the one in real world. Note that the probability E fails to extract a valid witness whereas the ZK proof is acceptable is negligible, we have that $\mathsf{HYB}_2(\lambda) \overset{c}{\approx} \mathsf{HYB}_1(\lambda)$, which means

$$\{\mathsf{REAL}_{\Pi,S^*(\tau)}(1^\lambda, x, y)\} \overset{c}{\approx} \{\mathsf{IDEAL}_{\mathcal{F}_{2pc},\mathsf{Sim}(\tau)}(1^\lambda, x, y)\}.$$

(T, ϵ)-security against malicious receiver. At a high level, the simulator acts as honest sender to interact with malicious receiver R^* in the first two round. Then it uses the nearly optimal extractor and tries to extract all witnesses for instances on the public keys of knowledge encryption appeared in the second message. Note that the witnesses for instances on knowledge encryption public keys appeared in $\{\mathsf{ot}_{2,i}\}$ reveal all the y' encoded in the second OT message. Simulator retrieves y' and sends it to functionality $\mathcal{F}_{2pc}$. After obtaining $f(x, y')$ from $\mathcal{F}_{2pc}$, it uses the simulator of GC scheme to "garble" the circuit. Finally, it uses the witnesses for instances on public keys of knowledge encryption in $\{\mathsf{ZK}_2\}$ to simulate the ZK_3 to complete the session.

We fix the security parameter λ, the polynomial T and the inverse polynomial ϵ. To avoid misunderstanding, we use $\mathsf{pk}^1_{i,0}, \mathsf{pk}^1_{i,1}$ to denote the public keys in $\mathsf{ot}_{2,i}$ and $\mathsf{pk}^2_{j,0}, \mathsf{pk}^2_{j,1}$ to denote the public keys in ZK_2. For a given (malicious) receiver R^*, we define:

- $\mathcal{R}$: The distribution over the *entire* input of R^*. This is a joint distribution of four random variables: the private input y, the receiver's randomness r, the first message of the sender S and the auxiliary input τ (drawn from $\mathcal{Z}$).

- Gen^*: On input $(y, r, \tau, \{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1) \leftarrow \mathcal{R}$, compute $(\{\mathsf{ot}_{2,i}\}_{i\in[n]}, \mathsf{ZK}_2) \leftarrow R^*(y, r, \tau, (\{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1))$. Output $\overline{\mathsf{pk}}_{[2n+2\lambda]} = (\{\mathsf{pk}^1_{i,0}, \mathsf{pk}^1_{i,1}\}_{i\in[n]}, \{\mathsf{pk}^2_{j,0}, \mathsf{pk}^2_{j,1}\}_{j\in[\lambda]})$, which appears in $(\{\mathsf{ot}_{2,i}\}_{i\in[n]}, \mathsf{ZK}_2)$.
- F: On input $(y, r, \tau, \{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1) \leftarrow \mathcal{R}$, compute $(\{\mathsf{ot}_{2,i}\}_{i\in[n]}, \mathsf{ZK}_2) \leftarrow R^*(y, r, \tau, (\{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1))$. Compute and output all third round messages of the WI proof in the same way as in the proof of our OT and ZK protocols.
- T' and ϵ': We set $T' = T + T_{R^*} + T_S$ and $\epsilon' = \epsilon/3(nl_1 + \lambda l_2)$, where T is the size of the distinguisher, T_{R^*} and T_S are the size of R^* and the sender S respectively, l_1 is the length of labels (which are supposed to be encrypted by $\mathsf{pk}^1_{i,0}$ or $\mathsf{pk}^1_{i,1}$) and l_2 is the length of messages which are supposed to be encrypted by $\mathsf{pk}^2_{j,0}$ or $\mathsf{pk}^2_{j,1}$ in ZK_3.

By Lemma 2, we have a nearly-optimal (T', ϵ')-extractor Ext (with respect to $t = 2n + 2\lambda$) against circuits of size T'. Using this extractor, the simulator proceeds as follows.

$\mathsf{Sim}(R^*)$:

1. Sample the randomness r for R^* and generate $\{\mathsf{OT}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1$ following the honest sender strategy.
2. Upon receiving $(\{\mathsf{OT}_{2,i}\}_{i\in[n]}, \mathsf{ZK}_2) \leftarrow R^*(r, y, \tau, \{\mathsf{OT}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1)$. Check if $\mathsf{ot}_{2,i}$ and the OT messages in ZK_2 are well-formed. If not, send $\perp$ to $\mathcal{F}_{2pc}$ and output the view of R^*; otherwise, compute all third round messages $\overline{\mathsf{WI}}_3$ of WI contained in $\mathsf{ot}_{3,i}$ and ZK_3 following the honest sender strategy.
3. Compute $(\{z^1_{i.0}, z^1_{i,1}\}_{i\in[n]}, \{z^2_{j,0}, z^2_{j,1}\}_{j\in[\lambda]}) \leftarrow \mathsf{Ext}(\overline{\mathsf{pk}}_{[2n+2\lambda]}, (y, r, \tau, \{\mathsf{ot}_{1,i}\}_{i\in[n]}, \mathsf{ZK}_1), \overline{\mathsf{WI}}_3)$. Write $\mathsf{pk}^1_{i,0} = (\mathsf{k}^1_{i,0}, x^1_{i,0})$, $\mathsf{pk}^1_{i,1} = (\mathsf{k}^1_{i,1}, x^1_{i,1})$ and $\mathsf{pk}^2_{j,0} = (\mathsf{k}^2_{j,0}, x^2_{j,0})$, $\mathsf{pk}^2_{j,1} = (\mathsf{k}^2_{j,1}, x^2_{j,1})$.
   (a) If there exists $i \in [n]$ or $j \in [\lambda]$ such that $z^1_{i,0}, z^1_{i,1}$ or $z^2_{j,0}, z^2_{j,1}$ are both valid witnesses for $x^1_{i,0}, x^1_{i,1}$ or $x^2_{j,0}, x^2_{j,1}$, then send $\perp$ to $\mathcal{F}_{2pc}$ and go to step 4. If else, do as follows.
   (b) For each $i \in [n]$, if $z^1_{i,0}$ is a valid witness for $x^1_{i,0} \in L_\Sigma$, then set the i^{th} bit of y' as 0, i.e. $y'_i = 0$. If else, then set $y'_i = 1$ directly.
   (c) Send y' to functionality $\mathcal{F}_{2pc}$. Upon receiving the output out, using the GC simulator to generate the labels, i.e. $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}, \mathsf{lab}^y_{i,y'_i}\}) \leftarrow \mathsf{GC.Sim}(1^\lambda, \phi(f), \mathsf{out})$. For each $i \in [n]$, compute and set $\mathsf{ot}_{3,i} = (\mathsf{WI}^1_{3,i}, \mathsf{KE.Enc}(\mathsf{pk}^1_{i,0}, \mathsf{lab}^y_{i,y'_i}), \mathsf{KE.Enc}(\mathsf{pk}^1_{i,1}, \mathsf{lab}^y_{i,y'_i}))$.
   (d) For the statement $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{1,i}, \mathsf{ot}_{2,i}, \mathsf{ot}_{3,i}\}_{i\in[n]})$, we simulate ZK_3 for it by running the step 3(b) of our ZK simulator. Specifically, write $\mathsf{ZK}_1 = (\{\mathsf{ot}'_{1,i}\}_{i\in[\lambda]}, \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{WI}_1)$, $\mathsf{ZK}_2 = (\{\mathsf{ot}'_{2,i}\}_{i\in[\lambda]}, \mathsf{WI}_2)$. For each $j \in [\lambda]$, if $z^2_{j,0}$ is a valid witness for $x^2_{j,0} \in L_\Sigma$, then set $\beta_j = 0$, otherwise, set $\beta_j = 1$. Use the HVZK simulator of the Σ-protocol to generate an acceptable proofs $(\alpha_j, \beta_j, \gamma_{j,\beta_j})$ for $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{1,i}, \mathsf{ot}_{2,i}, \mathsf{ot}_{3,i}\}_{i\in[n]}) \in L_{pk}$. Compute $C_{j,\beta_j} = (\mathsf{KE.Enc}(\mathsf{pk}_0, \gamma_{j,\beta_j}), \mathsf{KE.Enc}(\mathsf{pk}_1, \gamma_{j,\beta_j})$ and $C_{j,1-\beta_j} = (\mathsf{KE.Enc}(\mathsf{pk}_0, 0^{|\gamma|}), \mathsf{KE.Enc}(\mathsf{pk}_1, 0^{|\gamma|}))$. Let γ'_{j,β_j} be the message consisting

of γ_{j,β_j} and the randomness used in computing C_{j,β_j} and $\gamma'_{j,1-\beta_j} := 0^{l_2}$. Compute and set $\mathsf{ot}'_{3,j} = (\mathsf{WI}^2_{3,i}, \mathsf{KE.Enc}(\mathsf{pk}^2_{j,0}, \gamma'_{j,0}), \mathsf{KE.Enc}(\mathsf{pk}^2_{j,1}, \gamma'_{j,1}))$. Set $\mathsf{ZK}_3 = (\{\alpha_j, C_{j,0}, C_{j,1}, \mathsf{ot}'_{3,j}\}_{j\in[\lambda]}, \mathsf{WI}_3)$.

(e) Send $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{3,i}\}_{i\in[n]}, \mathsf{ZK}_3)$ to R^*.

4. Output the view of R^*.

We now prove the (T, ϵ)-security against malicious receiver. In the following, the distributions of all hybrid experiments are over the randomness of both parties and $\tau \leftarrow \mathcal{Z}, (x, y) \leftarrow (\mathcal{X}, \mathcal{Y})$. Let $\mathsf{HYB}_0(\lambda)$ be the ideal world experiment $\mathsf{IDEAL}_{\mathcal{F}_{2pc},\mathsf{Sim}(\tau)}(1^\lambda, x, y))$.

$\mathsf{HYB}_1(\lambda)$ is identical to the ideal world experiment except that it uses the honest GC algorithm to generate the needed labels $\{\mathsf{lab}^x_{i,x_i}, \mathsf{lab}^y_{i,y'_i}\}$. From the security of GC scheme, we have that $\mathsf{HYB}_1(\lambda) \overset{c}{\approx} \mathsf{HYB}_0(\lambda)$.

$\mathsf{HYB}_2(\lambda)$ is identical to $\mathsf{HYB}_1(\lambda)$ except that in step 3(a), if there exists $i \in [n]$ or $j \in [\lambda]$ such that $z^1_{i,0}, z^1_{i,1}$ (or $z^2_{j,0}, z^2_{j,1}$) are both valid witness for $x^1_{i,0}, x^1_{i,1}$ ($x^2_{j,0}, x^2_{j,1}$, repectively), then it generates the third round message honestly and go to step 4 directly.

Lemma 8. $\mathsf{HYB}_2(\lambda)$ *is statistically close to* $\mathsf{HYB}_1(\lambda)$

Proof. The proof of this lemma is similar to lemma 3 and 6. One can prove that the "if" condition in step 3(a) holds for both $b = 0$ and 1, occurs only with negligible probability, since otherwise we can construct a verifier of the WI protocol from R^* that breaks either the witness indistinguishability of the WI protocol or the one-wayness of the function f. □

$\mathsf{HYB}_{2+i}(\lambda)(i \in [n])$ is identical to $\mathsf{HYB}_{1+i}(\lambda)$ except that the simulator generates $\mathsf{ot}_{3,i} = (\mathsf{WI}^1_{3,i}, \mathsf{KE.Enc}(\mathsf{pk}^1_{i,0}, \mathsf{lab}^y_{i,0}), \mathsf{KE.Enc}(\mathsf{pk}^1_{i,1}, \mathsf{lab}^y_{i,1}))$, rather than $\mathsf{ot}_{3,i} = (\mathsf{WI}^1_{3,i}, \mathsf{KE.Enc}(\mathsf{pk}^1_{i,0}, \mathsf{lab}^y_{i,y'_i}), \mathsf{KE.Enc}(\mathsf{pk}^1_{i,1}, \mathsf{lab}^y_{i,y'_i}))$.

Lemma 9. *For any $i \in [n]$, and any T-size distinguisher D, we have that:*

$$|\Pr[D(\mathsf{HYB}_{2+i}(\lambda))] = 1 - \Pr[D(\mathsf{HYB}_{1+i}(\lambda))] = 1| \leq 2\epsilon l_1/3(nl_1 + \lambda l_2)$$

Again, one can use the same proof strategy for Lemma 4 to prove this lemma. Observe that for each $k \in [n]$, the difference between $\mathsf{HYB}_{2+k}(\lambda)$ and $\mathsf{HYB}_{1+k}(\lambda)$ is that, in the k-th run of the underlying three-round OT, the $2l_1$ ciphertexts in $\mathsf{ot}_{3,k}$ (under the public keys contained in $\mathsf{ot}_{2,k}$) are encryptions of different plaintexts.

Similarly, we can construct $2l_1$ hybrids moving from $\mathsf{HYB}_{1+k}(\lambda)$ to $\mathsf{HYB}_{2+k}(\lambda)$, each of them acting as the previous one but making a change on a single plaintext. One can prove that for any two neighboring hybrids, any distinguisher D of size T, the distinguishing advantage of D is less than $\epsilon/3(nl_1 + \lambda l_2)$, since otherwise we can construct a distinguisher D' of size $T' = T + T_{R^*} + T_S$ (incorporating D and R^*) that contradicts with Lemma 2 with parameter T' and $\epsilon' = \epsilon/3(nl_1 + \lambda l_2)$. Note that here the input x of the sender is not given to D', and therefore, to simulate hybrid

experiments, D' has to sample x in the first place. This can be done for *independent-input functionalities*, for which one can sample x conditioned on a given $y \leftarrow \mathcal{Y}$. Since there are $2l_1$ hybrids in total, we conclude Lemma 9.

Note that in hybrid $\mathsf{HYB}_{n+2}(\lambda)$, all GC circuit $\hat{C}$, labels $\{\mathsf{lab}^x_{i,b}\}, \{\mathsf{lab}^y_{i,b}\}$ and (the ciphertexts in) $\{\mathsf{ot}_{3,i}\}$ are generated honestly. We now construct $\mathsf{HYB}_{n+3}(\lambda)$ as follows.

$\mathsf{HYB}_{n+3}(\lambda)$ is identical to $\mathsf{HYB}_{n+2}(\lambda)$ except that for each $j \in [\lambda]$, the simulator generates Σ-proof $(\alpha_j, \beta_j, \gamma_{j,\beta_j})$ for $(\hat{C}, \{\mathsf{lab}^x_{i,x_i}\}_{i\in[n]}, \{\mathsf{ot}_{1,i}, \mathsf{ot}_{2,i}, \mathsf{ot}_{3,i}\}_{i\in[n]}) \in L_{pk}$ honestly in step 3(d) rather than using the HVZK simulator. From the HVZK property of Σ-protocol, we have $\mathsf{HYB}_{n+3}(\lambda) \overset{c}{\approx} \mathsf{HYB}_{n+2}(\lambda)$

$\mathsf{HYB}_{n+4}(\lambda)$ is identical to $\mathsf{HYB}_{n+3}(\lambda)$ except that for each $j \in [\lambda]$, in step 3(d), the simulator honestly generates $\gamma_{j,1-\beta_j}$ using witness and computes $C_{j,1-\beta_j} = (\mathsf{KE.Enc}(\mathsf{pk}_0, \gamma_{j,1-\beta_j}), \mathsf{KE.Enc}(\mathsf{pk}_1, \gamma_{j,1-\beta_j}))$.

Lemma 10. $\mathsf{HYB}_{n+4}(\lambda) \overset{c}{\approx} \mathsf{HYB}_{n+3}(\lambda)$

One can use the same proof strategy for Lemma 5 to prove this lemma.

$\mathsf{HYB}_{n+4+j}(\lambda)(j \in [\lambda])$ is identical to $\mathsf{HYB}_{n+3+j}(\lambda)$ except that in step 3(d), the simulator generates $\gamma'_{j,1-\beta_j}$ and the third round message $\mathsf{ot}'_{3,j}$ of OT *in* ZK_3 in the same way as honest sender.

It is easy to verify that $\mathsf{HYB}_{n+\lambda+4}(\lambda)$ is identical to the real execution between S and R^*. To conclude the proof of the (T, ϵ)-simulatability for the sender, we prove the following lemma.

Lemma 11. *For any $j \in [\lambda]$ and any T-size distinguisher D, we have that:*

$$|\Pr[D(\mathsf{HYB}_{n+4+j}(\lambda))] = 1 - \Pr[D(\mathsf{HYB}_{n+3+j}(\lambda))] = 1| \leq 2\epsilon l_2/3(nl_1 + \lambda l_2)$$

One can prove this lemma using the same reasoning underlying the proof of Lemma 4. Observe that for each $k \in [\lambda]$, the difference between $\mathsf{HYB}_{n+4+k}(\lambda)$ and $\mathsf{HYB}_{n+3+k}(\lambda)$ is that, in the k-th run of the underlying three-round OT contained in ZK, the $2l_2$ ciphertexts in $\mathsf{ot}'_{3,k}$ (under the public keys contained in $\mathsf{ot}'_{2,k}$) are encryptions of different plaintexts. We can show that for any two neighboring hybrids, any distinguisher D of size T, the distinguishing advantage of D is less than $\epsilon/3(nl_1 + \lambda l_2)$, otherwise, we can construct a distinguisher D' of size $T' = T + T_{R^*} + T_S$ (incorporating D and R^*) that contradicts with Lemma 2 with parameter T' and $\epsilon' = \epsilon/3(nl_1 + \lambda l_2)$. Thus, by a standard hybrid argument, we conclude Lemma 11.

In sum, we have that, for any T-size distinguisher D,

$$\Big|\Pr[D(\mathsf{REAL}_{\Pi,R^*(\tau)}(1^\lambda, x, y))] = 1 - \Pr[D(\mathsf{IDEAL}_{\mathcal{F}_{2pc},\mathsf{Sim}(\tau)}(1^\lambda, x, y))] = 1\Big| \leq 2\epsilon/3 + negl < \epsilon,$$

which concludes the proof of (T, ϵ)-security against malicious receiver.