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) [23,1,4] and witness encryption (WE) [20] 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. [29] 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 works [34,9,16] 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 [34] 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 [27] and are round-optimal under polynomially hard falsifiable assumptions while black-box reduction/simulation are used to prove the soundness/security for receiver [38]. Bitansky et al. [9] 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 [31].

Deng [16] 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 [16] 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 [44,45,3]), 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 [40], the Quadratic(Nth) Residuosity [33] or the LWE assumption [10].

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.

[3] Note that the result of [14] that distinguisher-dependent simulatability can be upgraded to (T, ϵ)-simulatability holds only for *zero knowledge* protocols.

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 [31], or *game-based* security for both parties [13].

- **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.
  Prior works on these protocols either require an additional assumption–the existence of dense encryption, or are only known based on the Factoring assumption [16]. The three-round protocol of secure two-party computation in [4] 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 [34].
  Our result on weak zero knowledge is incomparable to the work of [9]: The protocol in [9] 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 [42] and [35]) 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

$\Sigma_{i=1}^{\ell} y_i w_i \bmod 2$; 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 $\Sigma_{i=1}^{\ell} y_i w_i \bmod 2 = w_i$ correctly with almost the same probability. One can apply this reasoning to ciphertext distinguishers and prove the witness extractability property.

An alternative construction from CDS and random-self-reducible encryptions is presented in the full version of this paper [17].

Nearly optimal (T, ϵ)-extractor for knowledge encryption. Applying the result of [16], 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 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.

[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 [12], combing the recent work of [39] with the work [24], one can build non-interactive commitment from two-round (perfectly correct) OT with game-based security. Thus, two-round OT with game-based security as we define is sufficient for constructing all primitives used in our protocol.

[5] Here we actually need Goldriech-Kahan technique to bound the running time of the extractor, see the detailed proof in the full version of this paper [17].

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

(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 [2,36]. 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 [34], 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 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 [28] 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 [16] 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 [34], 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 [41,19,13] 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 [37]. In the common reference string model, fully-simulatable secure (even UC-secure) OT can be achieved in two rounds from various assumptions [43,18], such as DDH, LWE, CDH or LPN assumptions.

Related work on two/multi-party computation. Katz and Ostrovsky [37] 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 [41,15]. Garg et. al [21] 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 [5,32]. Recently, Choudhuri et. al [12] constructed a four-round construction only from four-round fully-simulatable OT. In the CRS model, Benhamouda and Lin [6] and Garg and Srinivasan [22] 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.

Due to space limitations, most of standard definitions (e.g., commitment schemes, Σ-protocol, game-based secure OT, garbled circuits etc.) are deferred to the full version of this paper [17].

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 [8,7] 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 [30] 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 [16,14], (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 $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(x)$:*

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

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 [40,1] and the simulation-based security [25]. In this paper our goal is to achieve simulation-based security, which is defined as follows..

Message Space. We let the message space $\mathcal{M}$ to include the special symbol $\perp$, i.e., $\mathcal{M} := \{0,1\}^n \cup \perp$. 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.

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.

The ideal functionality of OT is provided in Fig.1.

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.

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 $\mathcal{F}_{OT}$

Definition 4. (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 $m_0, m_1 \in \{0,1\}^n, b \in \{0,1\}$,*

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

In this paper, we follow the definition of weak simulatability in [16,14] and give a definition of simulatable (T, ϵ)-security for sender of an OT protocol (S, R).

Definition 5. ((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 [16]) 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[7]. 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 [4]. 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 [37], 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.

- *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

[7] One exceptional case is the UC composition [11], where Π may be composed with arbitrarily unknown protocols.

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 6. (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 [34].

Definition 7. (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 8. ((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 $D := \{D_\lambda\}_{\lambda\in\mathbb{N}}$,*

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

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

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 9 (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 $m \in \{0,1\}$,* KE.Enc *outputs a ciphertext c.*
- $\mathsf{KE.Dec}(\mathsf{sk}, c)$: *On input the secret key* sk *and ciphertext c,* 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 2. 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 [16] 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:

$$C(x, w, y, m) = \begin{cases} m & \text{if } (x, w) \in R_L \text{ and } y = 0^\ell, \\ \Sigma_{i=1}^{\ell} y_i w_i \bmod 2 & \text{if } (x, w) \in R_L \text{ and } \|y\|_1 \geq 1^{8}, \\ \perp & \text{if } (x, w) \notin R_L. \end{cases} \quad (2)$$

[8] 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.

The formal description of knowledge encryption for R_L[9] from two-round OT is shown in Fig.2.

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 9.)

$\mathcal{D}_{2,m}$: 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., $(\hat{C}, \{\mathsf{lab}_{i,b_i}\}) \leftarrow \mathsf{Sim}(1^\lambda, \phi(C), \perp)$.

[9] 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:

$$R'_L := (x, w') \in \{0,1\}^\lambda \times \{0,1\}^{\ell+1} : w' = w\|1 \text{ and } (x, w) \in R_L,$$

for which $w\|0$ is not a valid witness (for any instance x).

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 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[10] 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

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

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.

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

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, 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,

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

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

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

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. □

An alternative construction based on RSR encryption and CDS scheme appears in the full version of this paper [17].

3.2 Nearly-optimal Extractor for Knowledge Encryption

Following [16], 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 [16].

Lemma 1 (Nearly-Optimal (T, ϵ)-Extractor for t-Instance Sampler [16]). *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 $\lambda \in \mathbb{N}$,*

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

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 [16] 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 [16] (which holds only with respect to the Rabin's encryption based on factoring). For the sake of completeness, we provide its proof in the full version [17].

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 $\lambda \in \mathbb{N}$,*

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

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.

Remark 3. The proof strategy of [16] 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.3, 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:

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

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. 3: Three-round Oblivious Transfer Protocol

Note that non-interactive commitment can be built from two-round (perfectly correct) OT with game-based security (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 2. *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.*

Due to space limitation, we defer the detailed proof to the full version of this paper [17]. Here we only provide a sketch of proof.

proof sketch. The simulatable security for the receiver can be proven using rewinding simulation strategy (once a preimage is obtained by rewinding, the simulator can generate two valid public keys and decrypt both ciphertexts[11] from the sender), but one must be careful in the analysis of the running time of the rewinding simulator, which actually requires the Goldreich-Kahan technique [26] to make sure that the simulator will run in expected polynomial time.

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

The (T, ϵ)-simulatable security for the sender can be proven by constructing the following simulator. 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.

When replacing (T, ϵ)-extractor with a quasi-polynomial extractor (guaranteed by Lemma 2) in the simulation of the receiver's view, the second part of Theorem 2 follows.

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.3.
- 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.4. Due to space limitation, we give only the statement of our result in this section. The proof can be found in the full version [17].

Theorem 3. *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.*

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 $\delta_0,\delta_1 \leftarrow \{0,1\}^\lambda$ 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. 4: 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 [28] 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.3).
- A 3-round delayed-input weak zero knowledge argument $(\mathsf{ZK}_1, \mathsf{ZK}_2, \mathsf{ZK}_3)$ (presented in Fig.4) 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.5.

Theorem 4. *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*

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. 5: 3-round Two-party Weak Secure Computation

(T, ϵ)-security against 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 4 in the full version of this paper [17].

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 [34,16], we will not repeat these proofs here.

The work [34] 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 5. *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 [16] 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 6. *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

1. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: How to sell digital goods. In: Advances in Cryptology - EUROCRYPT'01. pp. 119–135. LNCS 2045, Springer, Springer (2001). https://doi.org/10.1007/3-540-44987-6_8
2. Aiello, W., Bhatt, S.N., Ostrovsky, R., Rajagopalan, S.: Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In: Automata, Languages and Programming - ICALP'00. pp. 463–474. LNCS 1853, Springer (2000). https://doi.org/10.1007/3-540-45022-X_39
3. Alwen, J., Persiano, G., Visconti, I.: Impossibility and feasibility results for zero knowledge with public keys. In: Advances in Cryptology – CRYPTO'05. pp. 135–151. LNCS 3621, Springer (2005). https://doi.org/10.1007/11535218_9
4. Ananth, P., Jain, A.: On secure two-party computation in three rounds. In: Theory of Cryptography - TCC'17. pp. 612–644. LNCS 10677, Springer (2017). https://doi.org/10.1007/978-3-319-70500-2_21
5. Badrinarayanan, S., Goyal, V., Jain, A., Kalai, Y.T., Khurana, D., Sahai, A.: Promise zero knowledge and its applications to round optimal MPC. In: Advances in Cryptology - CRYPTO'18. pp. 459–487. LNCS 10992, Springer (2018). https://doi.org/10.1007/978-3-319-96881-0_16
6. Benhamouda, F., Lin, H.: k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In: Advances in Cryptology - EUROCRYPT'18. pp. 500–532. LNCS 10821, Springer (2018). https://doi.org/10.1007/978-3-319-78375-8_17

7. Bitansky, N., Brakerski, Z., Kalai, Y., Paneth, O., Vaikuntanathan, V.: 3-message zero knowledge against human ignorance. In: Theory of Cryptography - TCC'16. pp. 57–83. LNCS 9985, Springer (2016). https://doi.org/10.1007/978-3-662-53641-4_3
8. Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Proceedings of the 45th Annual ACM Symposium on the Theory of Computing - STOC'14. pp. 505–514. ACM Press (2014). https://doi.org/10.1145/2591796.2591859
9. Bitansky, N., Khurana, D., Paneth, O.: Weak zero-knowledge beyond the black-box barrier. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC'19. pp. 1091–1102. ACM press (2019). https://doi.org/10.1145/3313276.3316382
10. Brakerski, Z., Döttling, N.: Two-message statistically sender-private ot from lwe. In: Theory of Cryptography - TCC'18,. pp. 370–390. LNCS 11240, Springer (2018). https://doi.org/10.1007/978-3-030-03810-6_14
11. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings of the 42nd Annual Symposium on Foundations of Computer Science - FOCS'01. pp. 136–145. IEEE Computer Society (2001). https://doi.org/10.1109/SFCS.2001.959888
12. Choudhuri, A.R., Ciampi, M., Goyal, V., Jain, A., Ostrovsky, R.: Round optimal secure multiparty computation from minimal assumptions. In: Theory of Cryptography - TCC'20. pp. 291–319. LNCS 12551, Springer (2020). https://doi.org/10.1007/978-3-030-64378-2_11
13. Choudhuri, A.R., Ciampi, M., Goyal, V., Jain, A., Ostrovsky, R.: Oblivious transfer from trapdoor permutations in minimal rounds. In: Theory of Cryptography - TCC'21. pp. 518–549. LNCS 13043, Springer (2021). https://doi.org/10.1007/978-3-030-90453-1_18
14. Chung, K.M., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Theory of Cryptography - TCC'15. pp. 66–92. LNCS 9014, Springer (2015). https://doi.org/10.1007/978-3-662-46494-6_4
15. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Round-optimal secure two-party computation from trapdoor permutations. In: Theory of Cryptography - TCC'17. pp. 678–710. LNCS 10677, Springer (2017). https://doi.org/10.1007/978-3-319-70500-2_23
16. Deng, Y.: Individual simulations. In: Advances in Cryptology – ASIACRYPT'20. pp. 805–836. LNCS 12493, Springer (2020). https://doi.org/10.1007/978-3-030-64840-4_27
17. Deng, Y., Zhang, X.: Knowledge encryption and its applications to simulatable protocols with low round-complexity. Cryptology ePrint Archive, Paper 2022/1193 (2022), https://eprint.iacr.org/2022/1193
18. Döttling, N., Garg, S., Hajiabadi, M., Masny, D., Wichs, D.: Two-round oblivious transfer from CDH or LPN. In: Advances in Cryptology - EUROCRYPT'20. pp. 768–797. LNCS 12106, Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_26
19. Friolo, D., Masny, D., Venturi, D.: A black-box construction of fully-simulatable, round-optimal oblivious transfer from strongly uniform key agreement. In: Theory of Cryptography - TCC'19. pp. 111–130. LNCS 11891, Springer (2019). https://doi.org/10.1007/978-3-030-36030-6_5
20. Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness encryption and its applications. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of

Computing - STOC'13. p. 467–476. ACM press (2013). https://doi.org/10.1145/2488608.2488667
21. Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: Advances in Cryptology - EUROCRYPT'16. pp. 448–476. LNCS 9666, Springer (2016). https://doi.org/10.1007/978-3-662-49896-5_16
22. Garg, S., Srinivasan, A.: Two-round multiparty secure computation from minimal assumptions. In: Advances in Cryptology - EUROCRYPT'18. pp. 468–499. LNCS 10821, Springer (2018). https://doi.org/10.1007/978-3-319-78375-8_16
23. Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing - STOC'98. p. 151–160. ACM press (1998). https://doi.org/10.1145/276698.276723
24. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science - FOCS'00. pp. 325–335. IEEE Computer Society (2000). https://doi.org/10.1109/SFCS.2000.892121
25. Goldreich, O.: Foundations of Cryptography, vol. Basic Applications. Cambridge University Press (2004). https://doi.org/10.1017/CBO9780511721656
26. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology **9**(3), 167–190 (1996). https://doi.org/10.1007/BF00208001
27. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM Journal on Computing **25**(1), 169–192 (1996). https://doi.org/10.1137/S0097539791220688
28. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the nineteenth annual ACM symposium on Theory of computing - STOC'87. pp. 218–229. ACM press (1987). https://doi.org/10.1145/28395.28420
29. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to run turing machines on encrypted data. In: Advances in Cryptology – CRYPTO'13. pp. 536–553. LNCS 8043, Springer (2013). https://doi.org/10.1007/978-3-642-40084-1_30
30. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing **18**(1), 186–208 (1989). https://doi.org/10.1137/0218012
31. Goyal, V., Jain, A., Jin, Z., Malavolta, G.: Statistical zaps and new oblivious transfer protocols. In: Advances in Cryptology – EUROCRYPT'20. pp. 668–699. LNCS 12107, Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_23
32. Halevi, S., Hazay, C., Polychroniadou, A., Venkitasubramaniam, M.: Round-optimal secure multi-party computation. In: Advances in Cryptology - CRYPTO'18. pp. 488–520. LNCS 10992, Springer (2018). https://doi.org/10.1007/978-3-319-96881-0_17
33. Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. Journal of Cryptology **25**(1), 158–193 (2012). https://doi.org/10.1007/s00145-010-9092-8
34. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Advances in Cryptology - CRYPTO'17. pp. 158–189. LNCS 10402, Springer (2017). https://doi.org/10.1007/978-3-319-63715-0_6

35. Kalai, Y.T., Khurana, D., Sahai, A.: Statistical witness indistinguishability (and more) in two messages. In: Advances in Cryptology - EUROCRYPT'18. pp. 34–65. LNCS 10822, Springer (2018). https://doi.org/10.1007/978-3-319-78372-7_2
36. Kalai, Y.T., Raz, R.: Probabilistically checkable arguments. In: Advances in Cryptology - CRYPTO'09. pp. 143–159. LNCS 5677, Springer (2009). https://doi.org/10.1007/978-3-642-03356-8_9
37. Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: Advances in Cryptology - CRYPTO'04. pp. 335–354. LNCS 3152, Springer (2004). https://doi.org/10.1007/978-3-540-28628-8_21
38. Kiyoshima, S.: Black-box impossibilities of obtaining 2-round weak ZK and strong WI from polynomial hardness. In: Theory of Cryptography - TCC'21. LNCS, vol. 13042, pp. 369–400. Springer (2021). https://doi.org/10.1007/978-3-030-90459-3_13
39. Lombardi, A., Schaeffer, L.: A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Paper 2019/279 (2019), https://eprint.iacr.org/2019/279
40. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Proceedings of the Twelfth Annual Symposium on Discrete Algorithms - SODA'01. pp. 448–457. Society for Industrial and Applied Mathematics (2001)
41. Ostrovsky, R., Richelson, S., Scafuro, A.: Round-optimal black-box two-party computation. In: Advances in Cryptology – CRYPTO'15. pp. 339–358. LNCS 9216, Springer (2015). https://doi.org/10.1007/978-3-662-48000-7_17
42. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Advances in Cryptology - EUROCRYPT'03. pp. 160–176. LNCS 2656, Springer (2003). https://doi.org/10.1007/3-540-39200-9_10
43. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Advances in Cryptology - CRYPTO'08. pp. 554–571. LNCS 5157, Springer (2008). https://doi.org/10.1007/978-3-540-85174-5_31
44. Xiao, D.: (nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In: Theory of Cryptography - TCC'11. pp. 541–558. LNCS 6597, Springer (2011). https://doi.org/10.1007/978-3-642-19571-6_33
45. Xiao, D.: Errata to (nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In: Theory of Cryptography - TCC'13. pp. 721–722. LNCS 7785, Springer (2013). https://doi.org/10.1007/978-3-642-36594-2_40