Magic Adversaries Versus Individual Reduction: Science Wins Either Way [⋆]

Yi Deng[1,2]

[1] SKLOIS, Institute of Information Engineering, CAS, Beijing, P.R.China
[2] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing ,100878,China
deng@iie.ac.cn

Abstract. We prove that, assuming there exists an injective one-way function f, *at least* one of the following statements is true:

- (Infinitely-often) Non-uniform public-key encryption and key agreement exist;
- The Feige-Shamir protocol instantiated with f is distributional concurrent zero knowledge for a large class of distributions over any OR NP-relations with small distinguishability gap.

The questions of whether we can achieve these goals are known to be subject to black-box limitations. Our win-win result also establishes an unexpected connection between the complexity of public-key encryption and the round-complexity of concurrent zero knowledge.

As the main technical contribution, we introduce a dissection procedure for concurrent adversaries, which enables us to transform a magic concurrent adversary that breaks the distributional concurrent zero knowledge of the Feige-Shamir protocol into non-black-box constructions of (infinitely-often) public-key encryption and key agreement.

This dissection of complex algorithms gives insight into the fundamental gap between the known *universal* security reductions/simulations, in which a single reduction algorithm or simulator works for *all* adversaries, and the natural security definitions (that are sufficient for almost all cryptographic primitives/protocols), which switch the order of qualifiers and only require that for every adversary there *exists* an *individual* reduction or simulator.

1 Introduction

The seminal work of Impagliazzo and Rudich [IR89] provides a methodology for studying the limitations of black-box reductions. Following this methodology, plenty of black-box barriers, towards building cryptographic systems on simpler primitives/assumptions and achieving more efficient constructions, have been found in the last three decades. These findings have long challenged us to

[⋆] A full version of this work appeared in [Den16]. “Science wins either way” is credited to Silvio Micali. This work was supported by the National Natural Science Foundation of China (Grant No. 61379141), and the Open Project Program of the State Key Laboratory of Cryptology.

develop new reduction methods and get around the limitations of black-box reduction, however, the progress towards this goal is quite slow, and for most of the known black-box barriers, it is still unclear whether they even hold for arbitrary reductions.

We revisit two seemingly unrelated fundamental problems, for both of which the black-box impossibility results are well known.

The first problem is to identify the weakest complexity assumptions required for public-key encryption. Ever since the invention of public key cryptography by Diffie and Hellman [DH76], the complexity of public-key cryptography, i.e., lowering the underlying complexity assumptions for cryptographic primitives/protocols, is one of the most basic problems. In the past four decades, for some primitives, including pseudorandom generators, signatures and statistically-hiding commitments, we witnessed huge success on this line of research and can now base them on the existence of one-way functions [Rom90, HILL99, HR07], which is the minimum assumption in the sense that, as showed by [IL89], almost all cryptographic primitives/protocols imply the existence of one-way functions.

But for public-key encryption and key agreement– the concepts that were conceived in the original paper of Diffie and Hellman, we did not make that successful progress yet. Impagliazzo and Rudich proved in their seminal work [IR89] that there is no black-box reduction of one-way permutations to key agreement, and since public-key encryption implies key agreement, their result also separates one-way permutations from public-key encryption with respect to black-box reduction.

In [Imp95] Impagliazzo describes five possible worlds of complexity theory. The top two worlds among them are Cryptomania, where public-key crypggraphy exists, and Minicrypt where there are one-way functions but no public-key cryptography. Though the above black-box separation provides some strong negative evidences, they do not rule out the possibility of constructing public-key encryption from one-way functions, i.e., do not prove that we live in Minicrypt.

The other fundamental problem we consider is that of the round-complexity of concurrent zero knowledge. The notion of concurrent zero-knowledge, put forward by Dwork, Naor and Sahai [DNS98], extends the standard-alone zero-knowledge security notion [GMR89] to the case where multiple concurrent executions of the same protocol take place and an adversarial verifier may corrupt multiple verifiers and control the scheduling of the messages.

As observed in [DNS98], the traditional black-box simulator does not work for the classic constant-round protocols (including the Feige-Shamir type protocol [FS89] and the Goldreich-Kahan type protocol [GK96]) in the concurrent setting. Indeed, Canetti et al. [CKPR01] proved that concurrent zero-knowledge with black-box simulation requires a logarithmic number of rounds for languages outside BPP. Prabhakaran et al. [PRS02] later refined the analysis of the Kilian and Petrank's [KP01] recursive simulator and gave an (almost) logarithmic round concurrent zero knowledge protocol for NP.

In his breakthrough work, Barak [Bar01] introduced a non-black-box simulation technique based on PCP mechanism and constructed a constant-round public-coin *bounded-concurrent* zero knowledge protocol for NP, which breaks several known lower bounds for black-box zero knowledge. There has been a vast body of work (see Section 1.4) since then on developing new non-black-box techniques and reducing the round-complexity of zero knowledge protocol in the concurrent setting. However, The problem of whether we can achieve constant-round concurrent zero knowledge based on standard assumptions is still left open.

Note also that the known constructions that beat the lower bound on the black-box round-complexity are rather complicated and therefore impractical. Given the current state of the art, a more ambitious question is whether we can prove the concurrent zero knowledge property of the classic 4-round protocols (such as the Feige-Shamir protocol), although it is known to be impossible to give such a proof for these simple and elegant constructions via black-box simulations.

1.1 Universal Simulator "$\exists S \forall A$" Versus Individual Simulator "$\forall A \exists S$"

We observe that almost all known reduction and simulation techniques are *universal* in the sense that, in the security proof of a protocol/premitive, the reduction R (or simulator S) works for all possible efficient adversaries and turn the power of a given adversary A into the power of breaking the underlying assumptions (i.e., "$\exists R$ or S $\forall A$"). However, for most natural security definitions, it is only required that for every adversary A there *exists* an *individual* reduction R (or a simulator S) that works for A (i.e., "$\forall A \exists R$ or S").

This motivates us to step back and look at the concurrent security of the simplest Feige-Shamir protocol. We will show that there is an *individual* simulator for the specific adversarial verifier (and thus it is not a concrete "attacker") constructed by Canetti et al. [CKPR01], though it was shown that for such a adversary the known black-box simulator fails. Sure, showing the existence of a simulator for a specific verifier does not mean that the Feige-Shamir protocol is concurrent zero knowledge, but this example does reveal a gap between the universal simulation "$\exists S \forall A$" and the individual simulation "$\forall A \exists S$" .

The Feige-Shamir protocol for proving $x \in L$ proceeds as follows. In the first phase, the verifier picks two random strings α_1 and α_2, computes two images, $\beta_1 = f(\alpha_1)$, $\beta_2 = f(\alpha_2)$, of a one-way function f, and then proves to the prover via a constant-round witness indistinguishability protocol that he knows either α_1 or α_2; in the second phase, the prover proves that either $x \in L$ or he knows one of α_1, α_2. The adversary V^* constructed in [CKPR01] adopts a delicate scheduling strategy, and when computing a verifier message, it applies a hash function h with high independence to the history hist sofar and generates the randomness $r = h(\mathsf{hist})$ for computing the current message. In our case, the randomness for the first verifier step of a session includes the two pre-images α_1 and α_2.

Canetti et al. showed that it is impossible for an efficient simulator to simulate V^*'s view when treating it as a black-box[1]. However, as mentioned before, the natural concurrent zero knowledge condition does not require a universal (or black-box) simulator that works for all adversarial verifiers, but just requires that for every specific V^* there *exists* an *individual* simulator.

Note that the individual simulator may depends on the specific verifier, and more importantly, since we are only required to show the *mere existence* of such a simulator, we can assume that the individual simulator knows (or equivalently, takes as input) the verifier's *functionality*, *randomness*, etc.

Indeed, for the adversary V^* of [CKPR01], there *exists*, *albeit* probably not efficiently constructible from a given (possibly obfuscated) code of V^*, a simple simulator for the above specific V^*: Note that there *exists* an adversary V' that acts exactly in the same way as V^* except that at each step V' outputs $r = h(\mathsf{hist})$ together with the current message, and thus a trivial simulator $\mathsf{Sim}(V')$, incorporating V' and using the fake witness (one of α_1 and α_2[2]) output by V' at the first verifier step of each session, can easily generate a transcript that is indistinguishable from the real interaction between V^* and honest provers .

1.2 Our Work

We prove an unexpected connection between the complexity of public-key encryption and the round-complexity of concurrent zero knowledge. Specifically, we show how to transform an attacker that can break a weak version of distributional concurrent zero knowledge of the Feige-Shamir protocol instantiated with injective one-way functions into (infinitely-often) constructions of public-key encryption and key agreement. This means at least one of the two problems (with respect to infinitely-often version and distributional version respectively) mentioned above has a *positive* answer.

[1] I.e., the simulator is given only oracle access to V^*, and does not have knowledge about its code, running time, etc.

[2] Note that α_1 and α_2 are part of the randomness r used in the first verifier message of a session.

A formal statement of our result. Let L and R_L be an arbitrary NP language and its associated NP relation respectively. The OR language $L \vee L$[3] and the corresponding relation $R_{L_{OR}}$ are defined in a natural way.

Given an arbitrary efficiently samplable distribution ensemble $D = \{D_n\}_{n \in N}$ over R_L (each D_n is over $R_L^n := \{(x, w) : (x, w) \in R_L \wedge |x| = n\}$), and an arbitrary efficiently samplable distribution Z_n over $\{0, 1\}^*$[4], we define the joint distribution $\{(X_n, W_n, Z_n)\}_{n \in N}$ over $R_{L_{OR}} \times \{0, 1\}^*$ in the following way: Sample $(x_1, w_1) \leftarrow D_n$,$(x_2, w_2) \leftarrow D_n$, $z \leftarrow Z_n$, $b \leftarrow \{1, 2\}$, and output $((x_1, x_2), w_b)$.

Theorem 1. *Assume that there exists an injective one-way function f. Then,* at least *one of the following statements is true:*

- *(Infinitely-often) Non-uniform public-key encryption and key agreement exist;*
- *For every inverse polynomial ϵ, the Feige-Shamir protocol instantiated with f is distributional concurrent zero knowledge on $\{(X_n, W_n, Z_n)\}_{n \in N}$ defined as above with distinguishability gap bounded by ϵ.*

In an infinitely-often version of a primitive, the correctness and security of a construction are required to hold only for infinitely many security parameter n. The notion of *ϵ-distributional* concurrent zero knowledge (first defined in [CLP15b]) differs from the traditional zero knowledge in that its zero knowledge property holds on average (i.e., holds for distributions over the statements), and that the indistinguishability gap for any efficient distinguisher is bounded by an arbitrary inverse polynomial (instead of a negligibly function).

Very roughly, Theorem 1 says the Feige-Shamir protocol is concurrent secure in the Minicrypt: In the world where there are injective one-way functions but no public-key encryption, the Feige-Shamir protocol satisfies certain version of concurrent zero knowledge.

Remark 1. We note that the black-box lower bounds [IR89, CKPR01] also hold for the infinitely-often version of public-key encryption and the *ϵ-distributional* concurrent zero knowledge[5]. We stress that our public-key encryption (and the key agreement) is based on the injective one-way function f and the specific

[3] For simplicity, we consider only the OR composition of the same NP language L, but our result holds with respect to the OR composition of any two NP languages.

[4] The element z from Z_n will be given as auxiliary input to the verifier of Feige-Shamir protocol.

[5] Our result holds with respect to distributions that are not always over YES instances. By applying the lower-bound proof strategy of [CKPR01], we conclude that the Feige-Shamir protocol cannot be ϵ-distributional concurrent *black-box* zero knowledge for any non-trivial distribution over hard problems, see the full version of this work for more details.

attacker against the Feige-Shamir protocol, and is non-uniform and non-black-box in nature: The key generation, encryption and decryption algorithms in our public-key encryption scheme are all non-unform, and make non-black-box usage of the underlying function f and the attacker.

Dissecting a complex adversary: Revealing the Creation of a Trapdoor. The basic proof strategy of Theorem 1 is to transform a magic verifier against the Feige-Shamir protocol into constructions for (infinitely-often) public-key encryption and key agreement. This proof idea is somewhat similar in spirit to the one appeared in [DNRS03] but still quite unusual in cryptography. In our setting, formalizing such a proof idea is very complicated and requires substantially new techniques.

To deal with the complex concurrent adversary, we introduce a dissection procedure to pinpoint where a supposed successful adversary magically endow a set of images of the injective one-way function f with a trapdoor, which is the key step towards our construction of public-key encryption via the Goldreich-Levin Theorem. On the very high level, if an adversarial verifier V^* that can break concurrent zero knowledge of the Feige-Shamir protocol, then in the real interaction there must exist a step i (verifier steps are ordered according to their appearance in the concurrent setting) such that:

- With high probability, V^* will output a pair of images β_1 and β_2, i.e., the first verifier message of some session j at this step i, and at a later time it will reach its second step of session j, i.e., completes its 3-round proof that it knows one pre-image of β_1 and β_2 under f.
- But for any efficient algorithm T, even taking the code of V^* and the history prefix up to its i-th step, the probability that T inverts any one of these two images β_1 and β_2 is bounded away from 1.

The intuition behind this observation is as follows. If the above two items does not hold simultaneously, then at each verifier step, either V^* does not output a pair of images of a session, or it outputs a pair of images of session j but will never reach its second message of session j, or there is an efficient algorithm that can find one of the corresponding pre-images. In each case we will have a simple simulator that can simulate the view of the V^*, which leads to a contradiction.

Thus, for a given successful adversary V^* the above two items must hold simultaneously. This means V^* magically endow the images β_1 and β_2 output at its step i with a trapdoor (i.e., the witness w to the common input x): With the trapdoor w, one can play the role of honest prover until V^* completes his 3-round proof, then using standard rewinding technique to obtain one of the pre-images; while, without the knowledge of w, no efficient algorithm can invert any one of β_1 and β_2 with overwhelming probability. This is the key observation that enables us to construct public key encryption and key agreement from the injective one-way f.

The major challenge in the actual dissection is to show the existence of *infinitely many* security parameter n for each of which the above conditions hold (as required by infinitely-often public key encryption and key agreement). To cope with this difficulty, we develop a set of techniques that convert concrete security into asymptotic security, which may be of independent interest.

An overview of the proof. We divide the proof into four steps, which will be presented in sections 3 to 6 respectively. Roughly, the proof proceeds as follows.

STEP I: We introduce a dissection procedure and prove that there must be infinitely many n, for each of which there exists a step i of V^*, such that the above two items hold simultaneously. This illustrates the power of V^* that magically endows the images of f output by V^* at its step i with a sort of trapdoor.

STEP II: Note that V^* outputs a pair of images of f at its step i. To avoid that the sender and the receiver (both with a witness to x) may recover different pre-images from V^*, we construct a pair of non-interactive algorithms C and E from the code of V^* such that for each (n, i) obtained in the above step:

- C (with knowledge of a witness w to x) outputs a *single* image β of f with high probability;
- E (with knowledge of a witness w to x) will extract the pre-image of β output by C;
- No efficient algorithm can compute the pre-image of β with probability close to 1.

STEP III: Using standard techniques, we amplify the gap between the success probability of E and the success probability of any efficient inverting algorithm without knowing a witness to x, and obtain two algorithms M and Find, where M takes a sequence of (x, w) as input and outputs a sequence of images β of f, and Find takes the same sequence of (x, w) and outputs all pre-images corresponding to the sequence of images β, both with probability negligibly close to 1; further, there is no efficient algorithm that can invert all the images output by M simultaneously with non-negligible probability.

STEP IV: Note that the Feige-Shamir protocol is concurrent witness indistinguishable, and thus the above holds when M and Find use different witnesses. Starting with a magic adversary V^* that breaks the distributional concurrent zero knowledge of the Feige-Shamir protocol for distribution over OR NP-statements of the form $(x_1 \vee x_2)$, we construct the public-key encryption scheme (and key-exchange scheme) in a natural way: The receiver generates a sequence of (x_1, w_1) as the public/secret key pair; to encrypt a bit, the sender generates a sequence of (x_2, w_2) and runs M on input the sequence of OR statements $(x_1 \vee x_2)$ and their corresponding witnesses w_2 to generate a set of images of f, computes the hard-core of the corresponding pre-images and XOR the plaintext bit with the hardcore; to decrypt, the receiver runs Find on input the ciphertext and the sequence of witnesses w_1 to obtain the corresponding pre-images, and then computes the hardcore and gets the plaintext.

Remark 2. We use the code of V^* in our final construction of public-key encryption. However, what we actually need to construct public-key encryption is the *functionality* of V^*, that is, we can replace the code of V^* with *any* code[6] of the *same* functionality in the intermediate algorithms in each of above steps along the way.

1.3 A Wide Perspective on Reductions

As mentioned, the mostly common used security proof techniques– black-box techniques (see [RTV04, BBF13] for refined treatments) and the known non-black-box techniques [Bar01, DGS09, BP15]– are universal, where a single universal reduction algorithm works for all possible adversaries. Here in this section we abuse the term *reduction* and view *simulation* as a type of reduction. Note that the description of an adversary that the reduction algorithm has access to probably is an obfuscated code. This causes a trouble in cases where the *functionality* of the adversary is crucial for the reduction to go through (as showed in the above example of simulation for the adversary in [CKPR01], and see also [DGL$^+$16]), since we cannot expect the efficient reduction algorithm to figure out the functionality from a given obfuscated code of an arbitrary adversary.

However, in almost all cases, in a security proof the reduction can be *arbitrary*. This means the reduction is allowed to depend not only on the code of the adversary, but also on any "nice" properties of the adversary (if exist), such as functionality, good random tapes, etc. Furthermore, to show the mere existence of such an arbitrary reduction, we do not need to care about whether such properties can be efficiently extracted from the code of the adversary, but just assume that the reduction takes these properties as input. We refer to an arbitrary reduction as *individual* reduction, which is also called non-constructive reduction or non-uniform reduction in some previous work [BU08, CLMP13]. We stress that it is not always possible to turn an individual reduction into a universal reduction with a non-uniform advice because, in many cases, even if

[6] As long as it is of polynomial size.

we can prove all possible adversaries share a certain property, this property may not have a short description. (This will be clear in the following example.)

Recall that, to complete a security proof, we have to show for *every* adversary there is an individual reduction. This would be impossible unless we can prove that all possible adversaries have certain properties *in common*. Indeed, we observe that a few exceptional individual reductions in complexity (e.g., [Adl78]) and hardness amplification (e.g., [GNW95, CHS05, HS11]) literature are based on a property– the existence of "good" random tapes– shared by all possible adversaries. Let's take the reduction for BPP $\subseteq$ P/poly [Adl78] as an example. The first step of the proof of [Adl78] is to show a common property that every machine deciding a language $L \in$ BPP must have at least one good random tape on which this machine will make correct decisions on all instances of a given size. Using the mere existence of a good random tape, we can then simply hardwire this good random tape into the circuit family that decide the language L deterministically. This circuit family can be thought of as a reduction, which varies depending on the specific BPP machine since different machines may have different good random taps.

Besides the structure (success/failure) of the random tapes, there seems to be a more important structure of the adversaries, i.e., the structure of the adversary's computation, that would empower the individual reduction greatly. In cryptography, we actually already exploited structures of this type, such as the knowledge of exponent assumption and extractable one-way functions [Dam91, BCPR14], but most of them are viewed as just non-standard assumption. Our work seems to raise some hope that we may be able to prove highly non-trivial structures of the adversary's computation in some settings under standard assumptions in the future.

1.4 Related Work

There have been numerous efficient constructions ([RSA78, Rab79, GM82, CS99, Reg09, HKS03], to name a few) for public-key encryption with various security notions based on specific assumptions with various algebraic structures, and some less efficient constructions [NY90, BHSV98, Sah99, Lin03a] based on more abstract assumptions– enhanced trapdoor permutations or trapdoor functions with polynomial pre-image size. Since public-key encryption implies key agreement (secure against eavesdropping adversaries), the same assumptions are sufficient for the latter. On the negative side, the recent work of [DS16] strengthens the black-box separation of public-key encryption and general one-way functions in [IR89] by allowing the reduction to take the code of the underlying primitive as input.

In the line of research on concurrent zero knowledge, Goyal [Goy13] extended Barak's idea to achieve fully concurrent zero knowledge in polynomial rounds. In the globe hash model, Canetti et al. [CLP13a] showed that public-coin concurrent zero knowledge can be obtained with logarithmic round-complexity. Recently, Chung et al. [CLP15a] (based on [CLP13b]) presented the first constant-round concurrent zero knowledge protocol based on indistinguishability obfuscation with super-polynomial security. Assuming the existence public-coin input-differing obfuscation, Pandey et al. [PPS15] presented a 4-round concurrent zero knowledge protocol. Over the last two decades, concurrent zero knowledge protocols have been used as a key building block in the construction of generally composable cryptographic protocols [CLOS02, PR03, Lin03b, PR05, Pas04, Lin08, GGJ13, GGJS12, GGS15, GLP+15].

2 Preliminaries

In this section we mainly present the definition of ϵ-distributional concurrent zero knowledge and some related new notions and definitions that we will use, and refer readers to [Gol01, KL07] for some other standard notions and definitions.

If D is a distribution (or random variable), we denote by $x \leftarrow D$ the process of sampling x according to D, and by $\{x_i\}_{i=1}^{k} \leftarrow D^{\bigotimes k}$ the process of sampling k times x from D independently. Similarly, for a function $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$, $f^{\bigotimes k}$ denotes the function that maps $(x_1, x_2, ..., x_k)$ to $(f(x_1), f(x_2), ..., f(x_k))$.

We abbreviate probabilistic polynomial-time with PPT. Throughout this paper, all PPT algorithms/Turing machines are allowed to be non-uniform, and we use non-uniform PPT algorithms/Turing machines interchangeably with circuit families of polynomial size. In our default setting, the circuit families are also probabilistic.

Given a two-party protocol $\Pi = (P_1, P_2)$, for $i \in \{1, 2\}$, we denote by $\mathsf{Trans}_{P_i}(P_1(x), P_2(y))$ the transcript of an execution of Π (including the input to P_i) when P_1's input is x and P_2's input is y. For a joint distribution (X, Y) over the two parties' inputs, $\mathsf{Trans}_{P_i}(P_1(X), P_2(Y))$ naturally defines the distribution over all possible view of P_i.

Throughout the paper, we let n be the security parameter and denote by $negl(n)$ a negligible function. We write $\{X_n\}_{n\in\mathbb{N}} \overset{c}{\approx} \{Y_n\}_{n\in\mathbb{N}}$ to indicate that the two distribution ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally distinguishable.

A zero knowledge argument system is an interactive argument for which the view of the (even malicious) verifier in an interaction can be efficiently reconstructed. In this paper, we consider *distributional* zero knowledge, defined by Goldreich [Gol93], for which the indistinguishability between the real interaction and the simulation is only required to hold for any distribution over the inputs to each party, rather than to hold for every individual inputs. We follow the definition of [CLP15b], which departs from the one of [Gol93] in that it only requires that for each distribution over the inputs there exists an efficient simulator[7], and consider the case (following [DNRS03, CLP15b]) where the indistinguishability gap between the simulation and the real interaction is less than any inverse polynomial ϵ (instead of a negligible function). As we will show, the size of encryption algorithm of our encryption scheme is polynomial in the value $\frac{1}{\epsilon}$, which needs to be upper-bounded by a fixed (but arbitrary) polynomial.

Steps of the Concurrent Verifier and Steps of a Session. We also allow the adversary V^* to launch a *concurrent* attack [DNS98, PRS02] in which it interacts with a polynomial number of independent provers over an asynchronous network, and fully controls over the scheduling of all messages in these interactions.

We refer to the action of sending a message by V^* as a step (of V^*). In a real concurrent interaction, we order the steps of V^* according to their appearance. Note that in the concurrent setting, sessions of the Feige-Shamir protocol are executed in interleaving way, and thus, "the second verifier step of a session" refers to the second verifier step that appears in this specific session, not to the second step of V^* in the real concurrent interaction.

Definition 1 **(ϵ-Distributional Concurrent zero knowledge).** *We say that an interactive argument (P, V) for language L is ϵ-distributional concurrent zero knowledge if for every concurrent adversary V^*, and every distribution ensemble $\{(X_n, W_n, Z_n)\}_{n\in\mathbb{N}}$ over $R_L^n \times \{0,1\}^*$, there exists a non-uniform PPT Sim such that for all non-uniform PPT D and sufficient large n it holds that*

$$\begin{aligned}\Pr[\mathsf{D}(\mathsf{Trans}_{V^*}(P(X_n, W_n), V^*(Z_n)), Z_n) = 1]&\\ - \Pr[\mathsf{D}(\mathsf{Sim}(V^*, X_n, Z_n), Z_n) = 1] &< \epsilon(n),\end{aligned}$$

where both distributions are over (X_n, W_n, Z_n) and the random tapes of P and V^*.

The Feige-Shamir ZK Argument for NP. We here describe the Feige-Shamir constant-round[8] zero knowledge argument for NP based on an injective one-way function $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$.

[7] Instead, the definition of [Gol93] requires an efficient simulator for all distributions over the inputs.

[8] By merging the first and the second prover messages, one can obtain a 4-round Feige-Shamir protocol.

PROTOCOL FEIGE-SHAMIR

Common input: $x \in L$.
The prover P's input: w such that $(x, w) \in R_L$.
The verifier V's (auxiliary) input:z

First phase:
Execute the n-parallel-repetition of the 3-round Blum's protocol in which V plays the role of the prover:

$V \longrightarrow P$**:** Choose $\alpha_1, \alpha_2 \leftarrow \{0,1\}^n$ independently and at random, compute $\beta_1 = f(\alpha_1)$, $\beta_2 = f(\alpha_2)$, and compute the first prover message a of the 3-round n-parallel-repetition of the Blum's protocol in which V proves to P that he knows one of α_1, α_2.
Send β_1, β_2 and a.
$P \longrightarrow V$**:** Send a random challenge $e \leftarrow \{0,1\}^n$.
$V \longrightarrow P$**:** Send t.

Second phase:
P and V execute the n-parallel-repetition of the 3-round Blum's protocol in which P proves to V that either $x \in L$ or he knows one of α_1, α_2.

3 The Dissection of a Concurrent Verifier

In this section we develop a technique to dissect concurrent verifiers that reveals where a supposed concrete attacker against the Feige-Shamir protocol magically endows some images of an injective one-way function with a trapdoor. This is the key step towards constructing public-key encryption (and key agreement) from an injective one-way function.

As mentioned in the introduction, we show that a magic adversary V^* will endow a set of images of f with a trapdoor in the following sense: there are infinitely many n, for each of which there exists a step index i_n, such that the images (β_1, β_2) output by V^* at its step i_n can *only* be inverted by PPT algorithms with the trapdoor knowledge of a witness to the common input x with overwhelming probability.

3.1 The Main Lemma

We need the following notations to give a formal statement of our main lemma:

- Trans^{i_n} and $h \leftarrow \mathsf{Trans}^{i_n}$: The former denotes the distribution of the history prefix in the view of V^* up to its i_n-th step in the real concurrent interaction $\mathsf{Trans}_{V^*}(P(X_n, W_n), V^*(Z_n))$; the latter denotes the event of drawing a history prefix h from Trans^{i_n}, i.e., the event of generating h in the real concurrent interaction between honest prover(s) and V^*, where h consists of the statement x, the auxiliary input z to V^* and the interaction history prefix upto the step i_n of the verifier.

- $V^*|_h \rightsquigarrow (j, 2)$ denotes the event that, conditioned on the given history prefix h, V^* reaches the second verifier step of session j in the real concurrent interaction, i.e., V^* completes its proof of knowledge of one pre-image in session j.
- PartR_h consists of the randomness used by V^* and the *partial* randomness used by honest provers in those *incomplete* sessions in h (i.e., sessions in which the last prover message does not appear in h) in a real concurrent interaction.

  Observe that in a session of the Feige-Shamir protocol, the honest prover uses the knowledge of corresponding witness w *only* in its last step, and the transcript of a session before the prover last step is independent of w. Thus, the transcript of an *incomplete* session together with the prover's randomness used do not help reveal the witness w, but this is not the case for a *complete* session.

In the real concurrent interaction, given a history prefix h up to the i_n-th step of V^*, we denote by $h = h'||(\beta_1^j, \beta_2^j, a^j)$ the event that V^* outputs the first verifier message $(\beta_1^j, \beta_2^j, a^j)$ of some session j at its i_n-th step, where "$||$" denotes concatenation of messages.

Let ϵ be an arbitrary inverse polynomial, and $\mathrm{poly}(\cdot)$ be an arbitrary polynomial. Define

$$p(\cdot) := \frac{\epsilon(\cdot)}{2\mathrm{poly}^2(\cdot)}.$$

Lemma 1. *(Main Lemma) Let ϵ, p,* poly *be as above, and f be the one-way function used in the Feige-Shamir protocol. Assume that there is a non-uniform PPT verifier V^*, running in at most* $\mathrm{poly}(n)$ *steps, that breaks ϵ-distributional concurrent zero knowledge of the Feige-Shamir protocol on a joint distribution ensemble $\{(X_n, W_n, Z_n)\}_{n \in N}$ over a NP relation R_L[9] and auxiliary inputs. Then, there exists an infinite set $I = \{(n, i_n)\}$ for which the following two conditions simultaneously hold:*

1. *For a random history prefix generated in the real concurrent interaction,*

[9] Though in our final construction of public-key encryption we need to assume a magic adversarial verifier against the Feige-Shamir protocol for a distribution $\{(X_n, W_n)\}_{n \in N}$ over some *OR* NP-relation, Lemma 1 and the results in Section 4 and 5 hold with respect to distribution $\{(X_n, W_n)\}_{n \in N}$ over *any* NP relation.

$$\Pr\left[h \leftarrow \mathsf{Trans}^{i_n} : \begin{array}{c} h = h' || (\beta_1^j, \beta_2^j, a^j) \wedge \\ \Pr[V^* \mid_h \rightsquigarrow (j,2)] \geq p(n) \end{array}\right] \geq p(n).$$

2. *For every circuit family T of polynomial size, there is N_0 such that for every $n > N_0$ (s.t. $(n, \cdot) \in I$) it holds that,*

$$\Pr\left[T(h, \mathrm{PartR}_h) \in \{f^{-1}(\beta_1^j), f^{-1}(\beta_2^j)\} \middle| \begin{array}{c} h' || (\beta_1^j, \beta_2^j, a^j) = h \leftarrow \mathsf{Trans}^{i_n} \\ \wedge\ \Pr[V^* \mid_h \rightsquigarrow (j,2)] \geq p(n) \end{array}\right]$$
$$\leq 1 - p(n).$$

Remark 3. Note that if, conditioned on outputting the first verifier message $(\beta_1^j, \beta_2^j, a^j)$ of session j at its i_n-th step, V^* reaches the second verifier step of session j (i.e., completes the proof of knowledge of one pre-image) in the real concurrent interaction with probability greater than an inverse polynomial, we can construct an efficient algorithm, taking the corresponding witness w as input and playing the role of the honest prover, that extracts one of pre-images of (β_1^j, β_2^j) from V^* by rewinding it with probability negligibly close to 1. The first condition of our lemma asserts that it is relatively easy to obtain images of f for which there is an efficient algorithm with knowledge of w can invert one of them with overwhelming probability, while the second condition of the above lemma guarantees that for any efficient algorithm without knowledge of w the success probability of inversion is bounded away from 1. This illustrates the magic power that the supposed adversary V^* endows the images output at its step i_n with a sort of trapdoor.

As we shall see later, in the final construction of public key encryption, the partial randomness PartR_h together with some images of f will be part of cipher-text, and to ensure the semantic security it is naturally required that for any efficient algorithm with PartR_h as input the success probability of inverting the images of f is small. This is guaranteed by the second condition of the above lemma.

Remark 4. (On the role of the value ϵ) The main reason we deal only with ϵ-distributional concurrent zero knowledge, rather than the standard one, is that, as we will see later, our approach will yield encryption algorithm that runs in time $poly(\frac{1}{\epsilon})$, and thus the value $\frac{1}{\epsilon}$ has to be upper-bounded by a fixed (but arbitrarily) polynomial.

3.2 The Dissection Procedure Leading to a Proof of Lemma 1

Formally, if for an arbitrary inverse polynomial ϵ, V^* breaks ϵ-distributional concurrent zero knowledge of Feige-Shamir protocol over distribution $\{(X_n, W_n,$

$Z_n)\}_{n\in\mathbb{N}}$, then $\forall\, \mathsf{Sim}\ \exists\, \mathsf{D}$ and infinitely many n, such that

$$\begin{aligned}\Pr[\mathsf{D}(\mathsf{Trans}_{V^*}(P(X_n, W_n), V^*(Z_n)), Z_n) = 1]\\ - \Pr[\mathsf{D}(\mathsf{Sim}(V^*, X_n, Z_n), Z_n) = 1] > \epsilon(n). \qquad (1)\end{aligned}$$

As mentioned, the intuition behind Lemma 1 is quite straightforward: For a successful V^*, there must exist a step i at which V^* outputs a pair of images and will complete the proof of knowledge of one pre-image at a later time in the real concurrent interaction with high probability, but without knowledge of the corresponding witness no efficient algorithm can invert one of the images, since otherwise, if for every step of V^* there is an efficient algorithm that can extract the target pre-images with overwhelming probability, we are able to show that there *exists* a simulator, incorporating all these efficient inverting algorithms as its subroutines, that will simulate the view of V^* successfully.

To formalize this intuition in the asymptotic setting, we view the behaviour of V^* as an infinite table, in which the entry in the i-th row and n-th column represents the i-th step of V^* (followed immediately by the response from the honest prover) in its concurrent interaction on input the security parameter n (c.f. Fig 1).

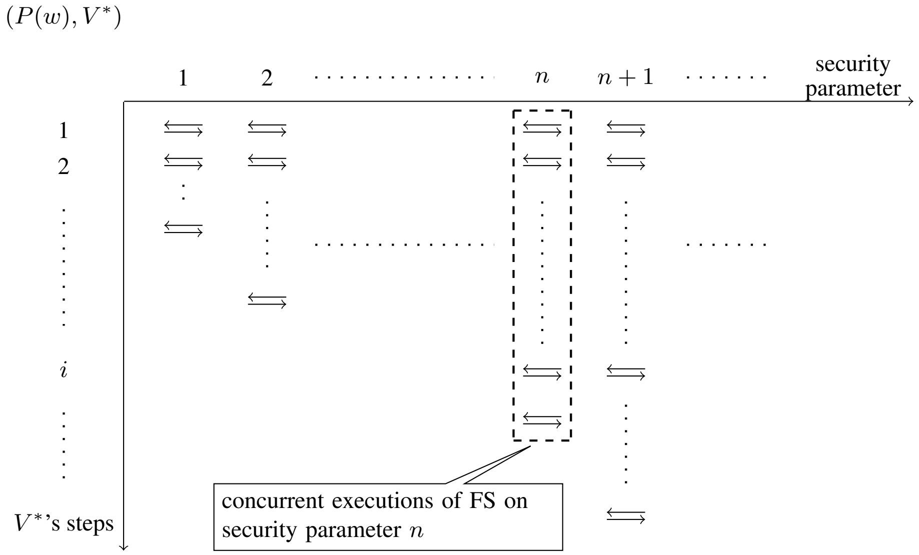

Fig. 1: V^*'s behaviour.

With this table, we dissect V^* and examine its every step *across all security parameters* $n \in \mathbb{N}$, i.e., examine the set of entries $\{(n, i_n = i)\}_{n \in \mathbb{N}}$. A few terminologies follow.

Imaginary steps. Note that for the i-th row of the table (i.e., V^*'s step i), if a security parameter n satisfies $\mathrm{poly}(n) < i$, V^* on the input security parameter n will never reach step i. To simplify the presentation, we think of the step i in every n-th column with $\mathrm{poly}(n) < i$ as an *imaginary step* of V^* with

$$\Pr\left[h \leftarrow \mathsf{Trans}^{i} : \begin{array}{c} h = h'||(\beta_1^j, \beta_2^j, a^j) \wedge \\ \Pr[V^* \mid_{h \rightsquigarrow} (j, 2)] \geq p(n) \end{array}\right] = 0.$$

Significant/insignificant entries. Given a (possibly infinite) set K of security parameters, and a set $K' = \{(n, i_n)\}_{n \in K}$, we say the entry $(n, i_n) \in K'$ is *significant* if for which the first condition of Lemma 1 holds, i.e.,

$$\Pr\left[h \leftarrow \mathsf{Trans}^{i_n} : \begin{array}{c} h = h'||(\beta_1^j, \beta_2^j, a^j) \wedge \\ \Pr[V^* \mid_{h \rightsquigarrow} (j, 2)] \geq p(n) \end{array}\right] > p(n).$$

Otherwise, we call it *insignificant*.

Solving a set of entries. Given a set (possibly infinite) K of security parameters, and a set $K' = \{(n, i_n)\}_{n \in K}$, we say a circuit family T of size $\mathbb{P}$ *solves* the set K', if for every *significant* entry $(n, i_n) \in K'$, T breaks the second condition of Lemma 1 on (n, i_n), i.e., for all $n \in K$,

$$\Pr\left[T(h, \mathrm{PartR}_h) \in \{f^{-1}(\beta_1^j), f^{-1}(\beta_2^j)\} \middle| \begin{array}{c} h'||(\beta_1^j, \beta_2^j, a^j) = h \leftarrow \mathsf{Trans}^{i_n} \\ \wedge \Pr[V^* \mid_{h \rightsquigarrow} (j, 2)] \geq p(n) \end{array}\right]$$
$$> 1 - p(n). \qquad (2)$$

Otherwise, we say T fails to solve the set K', i.e., there are *some* entries in K' on which the above inequality does not hold for T. When we say T of size $\mathbb{P}$ fails to solve *any* entry in the set K', we mean that every entry in K' is significant and T cannot solve even a single entry in K'.

Note that we don't make any requirement on T for those *insignificant* entries K' (i.e., those entries for which the first condition of Lemma 1 does not hold). To take an extreme example, if for *all* $(n, i_n) \in K'$ the first condition of Lemma 1 fails to hold, i.e.,

$$\Pr\left[h \leftarrow \mathsf{Trans}^{i_n} : \begin{array}{c} h = h'||(\beta_1^j, \beta_2^j, a^j) \wedge \\ \Pr[V^* \mid_{h \rightsquigarrow} (j, 2)] \geq p(n) \end{array}\right] < p(n),$$

then, by definition, any circuit family can solve the set K'. For simplicity, we let the circuit family that solves such a set K' to be a special dummy circuit family denoted by ϕ, which is of size 0.

With these definitions, we observe the following fact.

Fact 1. Fix a verifier step i. If for any polynomial $\mathbb{P}$, there does not exist a circuit family of size $\mathbb{P}$ that solves the set $\{(n, i_n = i)\}_{n\in\mathbb{N}}$, then there is an infinite set I on which both conditions of Lemma 1 hold.

Proof. Observe first that if for any polynomial $\mathbb{P}$, there is no $\mathbb{P}$-size circuit family that solves the set $\{(n, i)\}_{n\in\mathbb{N}}$, then for every $\mathbb{P}$-size circuit family T, there exists an *infinite* set K of security parameters such that T cannot solve any entry in the set $\{(n, i)\}_{n\in K}$. To see this, suppose for the sake of contradiction that, there is a $\mathbb{P}$-size circuit family T for which there is a *finite* set K such that T solves the set $\{(n, i_n = i)\}_{n\in\mathbb{N}\setminus K}$. Let c_k be the largest security parameter in K, and the circuit family T' be the inverting algorithm that, upon receiving a pair of images, inverts one of them by exhausting all possible pre-images. We now have a new circuit family of size $\mathbb{P}(n) + 2^{c_k}$, denoted by T_i, which applies T on the security parameters $n \in \mathbb{N} \setminus K$ and T' on $n \in K$, can solve the set $\{(n, i)\}_{n\in\mathbb{N}}$, which contradicts the hypothesis of this fact since $\mathbb{P}(n) + 2^{c_k}$ is still a polynomial in n.

We now fix a polynomial (monomial) n^c, and construct a *best possible* n^c-size circuit family $T := \{T^n\}$: Each circuit T_n is of size n^c and achieves the highest success probability of inverting. It follows from the observation above that there is an infinite set K_c of security parameters such that T cannot solve *any* entry in $\{(n, i)\}_{n\in K_c}$.

Since for each security parameter n, the circuit T^n is best possible, we conclude that, for any n^c-size circuit family $T' := \{T'^n\}$, T' cannot solve any entry in $\{(n, i)\}_{n\in K_c}$ (note that the success probability of the inverting circuit T'^n is less than the one of T^n).

Note that $K_c \subseteq K_{c-1}$ for all $c \in \mathbb{N}$. The desired infinite set I can be constructed as follows. Let $n_0 = 0$ and $n_c := \min\{K_c \backslash \{n_{c-1}, n_{c-1}, \cdots, n_0\}\}$[10] for each $c \in \mathbb{N}$. We define I to be

$$I := \{(n_c, i)\}_{c\in\mathbb{N}}.$$

It is easy to verify that the first condition of Lemma 1 holds on I.[11] Consider an arbitrary polynomial size circuit family T, say, of size $\mathbb{P}^\dagger$, and suppose that

[10] Note that in case K_c is identical to K_{c-1}, then $n_{c-1} \in K_c$.

[11] Note that for every $c \in \mathbb{N}$, for any entry (n, i) in $\{(n, i)\}_{n\in K_c}$, the first condition of Lemma 1 holds for (n, i), since otherwise the entry (n, i) is insignificant, and by definition can be solved by any circuit family.

$\mathbb{P}^{\dagger}(n) \leq n^{c'}$[12]. Then T cannot solve *any* entry $(n_c, i) \in I$ for any $c > c'$. Note that $c > c'$ implies $n_c > n_{c'}$, we have that T cannot solve any entry $(n_c, i) \in I$ for any $n_c > n_{c'}$. This establishes the second condition of Lemma 1.

□

The following dissection procedure (c.f. Fig 2) will yield an infinite set I as desired.

The dissection procedure. Initially set $I_0 := \{(n_0 = 0, i_{n_0} = 0)\}$, $S_0 := \{(T_0 = \phi, \mathbb{P}_0 = 0)\}$.

For $i = 1, 2, ...,$ given $I_{i-1} = \{(n_0, i_{n_0}), ..., (n_{k-1}, i_{n_{k-1}})\}$[13],$S_{i-1} = \{(T_0, \mathbb{P}_0), ..., (T_{i-1}, \mathbb{P}_{i-1})\}$ and $\mathbb{P} = \max\{\mathbb{P}_0, \mathbb{P}_1, ..., \mathbb{P}_{i-1}\}$, we check the i-th step of V^* for all $n \in \mathbb{N}$ and do the following:

1. If for any polynomial $\mathbb{P}'$ there is no $\mathbb{P}'$-size circuit family that solves the set $\{(n, i_n = i)\}_{n \in \mathbb{N}}$, let I be as defined in the above Fact 1, and stop this process;
2. If there are a polynomial $\mathbb{P}_i$ such that $\mathbb{P}_i \leq \mathbb{P}$, and a $\mathbb{P}_i$-size circuit family T_i that solves the set $\{(n, i_n = i)\}_{n \in \mathbb{N}}$, set $S_i \leftarrow S_{i-1} \cup (T_i, \mathbb{P}_i)$, and $I_i \leftarrow I_{i-1}$ (Note that we do not update the set I_{i-1});
3. If there are a polynomial $\mathbb{P}_i$ such that $\mathbb{P}_i > \mathbb{P}$, and a $\mathbb{P}_i$-size circuit family T_i that solves the set $\{(n, i_n = i)\}_{n \in \mathbb{N}}$, but no circuit family of size less than $\mathbb{P}$ that can solve the set $\{(n, i_n = i)\}_{n \in \mathbb{N}}$, then
   (a) set $S_i \leftarrow S_{i-1} \cup \{(T_i, \mathbb{P}_i)\}$, and,
   (b) if $i > \text{poly}(n_{k-1})$[14], find a $n_k > n_{k-1}$ on which the first condition of Lemma 1 holds, but no circuit family of size less than $\mathbb{P}$ can solve the set $I_{i-1} \cup \{(n_k, i_{n_k} = i)\}$[15]. Set $I_i \leftarrow I_{i-1} \cup \{(n_k, i_{n_k} = i)\}$.

Denote by I the set resulted from the above dissection procedure, which is either of the form $\{(n_c, i)\}_{c \in \mathbb{N}}$ (when we encounter the first case during the dissection procedure), or of the form $\{(n_k, i_{n_k})\}$ (otherwise).

Lemma 1 follows from the following two claims. Due to space limitations, we provide detailed proofs of these claims in the full version of this work [Den16].

Claim 1. If we encounter the first case during the above dissection, or there is no polynomial $\mathbb{P}$ s.t. $\mathbb{P} = \sup\{\mathbb{P}_i : i \in \mathbb{N}\}$, i.e., there is no polynomial upper-bound on the infinite set $\{\mathbb{P}_i : i \in \mathbb{N}\}$, then the set I is infinite and on which both conditions of Lemma 1 hold.

[12] A little bit oversimplified. In case that, for some N, $\mathbb{P}^{\dagger}(n) \leq n^{c'}$ only when $n > N$, we should set N_0 to be $\max\{N, n_{c'}\}$ and conclude that T cannot solve any entry $(n_c, i) \in I$ for any $n_c > N_0$.

[13] Here $k \leq i - 1$. Note that we may not update the set I at each step i.

[14] This means that the current i-step is an imaginary step of V^* for those $n \leq n_{k-1}$.

[15] As will be showed in proof of claim 1 in the next section, we can always find such a n_k.

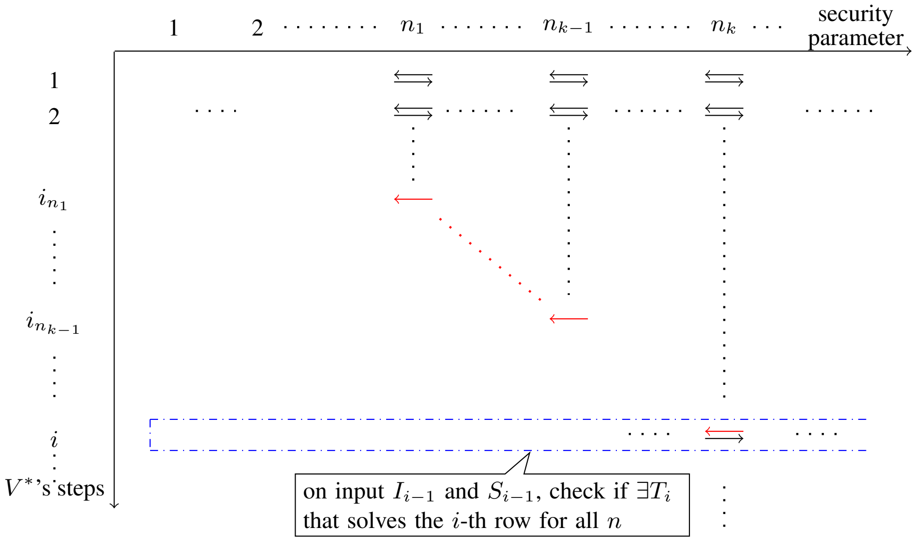
1
2
n_1
n_{k-1}
n_k
security parameter
1
2
i_{n_1}
$i_{n_{k-1}}$
i
V^*'s steps
on input I_{i-1} and S_{i-1}, check if $\exists T_i$ that solves the i-th row for all n

Fig. 2: The dissection procedure. For a magic adversary V^* there must exist either a single row (a step of V^*) from which we find the desired infinite set I, or infinite many rows from each of which we add a new entry to the set I.

Claim 2. If we will never encounter the first case during the above dissection, and there is a polynomial $\mathbb{P}$ s.t. $\mathbb{P} = \sup\{\mathbb{P}_i : i \in \mathbb{N}\}$, then there is a non-uniform PPT simulator that breaks the inequality (1).

Remark 5. (On the mere existence of T_i and the dependence between T_i's) Note that at each step of the dissection procedure we only ask if there *exists* a good extractor T_i, and that these algorithms may depend on a specific verifier. It may be the case that these T_i exist but we cannot construct them from the code V^* efficiently, as we showed for the concrete adversary from [CKPR01].

However, the mere existence of *good* extractors T_i, satisfying that all of them have size upper-bounded by a fixed polynomial as in Claim 2, helps us show the *existence* of a simulator for V^* under the natural security definition of "$\forall V^* \exists S$".

We stress that the dependence between the possible algorithms T_i's is irrelevant here. Note that at each step i, we set a clear bar $\mathbb{P}$ and check if there exists a circuit family T_i of size less than $\mathbb{P}$ that can solve all those significant entries in the i-th row. If there exists a circuit family T_i that solves this row but the minimal size $\mathbb{P}_i$ required is strictly greater than $\mathbb{P}$, we record this new $\mathbb{P}_i$ and when we enter the next step $(i + 1)$, we have a higher bar on the circuit size for checking the existence of T_{i+1}.

Nevertheless, if one can construct a verifier V^* for which there is a deep dependence between these T_i's such that, say, the size of T_{i-1} is twice that of T_i for many i, then we will soon find a desired set I as required by Lemma 1.

4 Tuning in to the Same Channel

As showed in the previous section, the real concurrent interaction between the honest prover and a successful adversary V^* will magically generate a history prefix of the form $h'||(\beta_1, \beta_2, a)$ for which only algorithms with knowledge of the corresponding witness can extract one of the pre-images of (β_1, β_2) with overwhelming probability. However, different algorithms using different witnesses/randomness may recover different pre-images from this history. Thus, to exploit the power of V^* in our setting, we first need to make sure that all parties are in the same channel, i.e., recover the same pre-image from a given history.

In this section we construct non-interactive algorithms C and E from the magic adversary V^* such that, taking as input the witness to x, C generates a β and E can obtain the pre-image of the *same* β. Detailed analyses of these two algorithms can be found in [Den16].

Lemma 2. *Let p, f, $\{(X_n, W_n, Z_n)\}_{n\in N}$, the infinite set I, and V^* be as in Lemma 1. Then there exist two non-unifrom PPT algorithms C and E such that for every $(n, i_n) \in I$ the following conditions hold:*

1. *C generates β, α and an auxiliary string aux satisfying $\beta = f(\alpha)$ with probability*

$$\Pr[(x, w, z) \leftarrow (X_n, W_n, Z_n) : C(x, w, z) = (\beta, \alpha, aux)] \geq p^2 - negl(n).$$

2. *It is easy for E with knowledge of w to invert the image output by C with probability*

$$\Pr\left[(x, w, z) \leftarrow (X_n, W_n, Z_n) : E(\beta, aux, w) = f^{-1}(\beta) \middle| \begin{matrix} C(x, w, z) \\ = (\beta, \alpha, aux) \end{matrix}\right]$$
$$\geq 1 - negl(n).$$

3. *For any polynomial-size circuit family T without knowing w, there is N_0 such that for every $n > N_0$ (s.t. $(n, \cdot) \in I$) it holds that:*

$$\Pr\left[(x, w, z) \leftarrow (X_n, W_n, Z_n) : T(\beta, aux) = f^{-1}(\beta) \middle| \begin{matrix} C(x, w, z) \\ = (\beta, \alpha, aux) \end{matrix}\right]$$
$$\leq 1 - p.$$

```
The Algorithm $C$
input : $(x, w, z) \leftarrow (X_n, W_n, Z_n)$
  1. Run $P$ and $V^*$ on input $(x, w, z)$ until obtain the history prefix $h$ up to the step $i$ of
     $V^*$. If the $V^*$'s step $i$ message $v_i$ is the first verifier message of the form $(\beta_1, \beta_2, a)$
     in a session, say, session $j$, then continue; otherwise, return $\perp$.
  2. Resume the interaction between $P$ and $V^*$ until $V^*$ terminates. If the second accepting
     verifier message $t$ in session $j$ appears in this interaction, continue; otherwise, return $\perp$.
  3. Repeat the following two steps $\frac{n}{p}$ times (there are at most $\frac{n^2}{p^2}$ iterations of step 2
     within this step):
     (a) Run the above step 2 using fresh randomness (based on the same history prefix
         $h$) until either the second accepting verifier message in session $j$ appears *twice*
         or the $\frac{n}{p}$-th iteration is reached. If two accepting transcripts of the first phase in
         session $j$ of the Feige-Shamir protocol are obtained within these $\frac{n}{p}$ iterations
         (for the purpose of simplifying the analysis of the algorithm $E$, here we don't
         use the transcript obtained in step 2), compute $\alpha$ such that $\beta_b = f(\alpha)$ from
         them; otherwise, return $\perp$.
     (b) Store $(\beta_b, \alpha)$ in a list.
  4. Set $\beta$ to be $\beta_b$ for which the corresponding pair $(\beta_b, \alpha)$ appears most often in the
     above list, and $aux$ to be $(h, \text{PartR}_h, x, z)$, where $\text{PartR}_h$ includes only the randomness
     used by $V^*$ and the randomness used by honest provers in those *incomplete*
     sessions in producing $h$.
    output: $(\beta, \alpha, aux)$.
```

Fix $(n, i) \in I$ (from here on we drop the n on i_n for simplicity). Incorporating V^* and the honest prover P, (n, i) and the inverse polynomial p, the algorithm C, on input (x, w, z), plays the role of the honest prover and extracts (by rewinding) one-pre-image of the pair images of f output by V^* at its i-th step, and then outputs the pre-image extracted and the corresponding image (together with some auxiliary information). To make sure that different algorithms can extract the same pre-image, we have C repeat the extraction precedure many times and output the image corresponding to the *most-often* extracted pre-image. See below for the detailed description of C.

The algorithm E, taking (β, aux, w) as input, simply repeats $\frac{n}{p}$ times the step 3(a) of the algorithm C to extract the pre-image of β.

```
The Algorithm $E$
input : $(\beta, aux, w)$
  1. Parse $aux$ into $(h, \text{PartR}_h, x, z)$, and parse the last message $v_i$ in $h$ into $(\beta_1, \beta_2, a)$.
  2. Suppose that $\beta = \beta_b$. Repeat the step 3(a) of $C$ until the pre-image $\alpha$ of $\beta_b$ is
     extracted or the $\frac{n}{p}$-th iteration is reached, and if all iterations fail, return $\perp$.
    output: $\alpha$.
```

5 Hardness Amplification and a Tailored Hard-Core Lemma

For our applications, we need to increase the success probability of the algorithm C significantly while decreasing T's success probability (as in the third condition of Lemma 2) to a negligible level. In addition, if the statement x has multiple witnesses, we also want the algorithm E to work when given an arbitrary one (not necessarily the same as the one given as input to C) as input.

Our basic strategy for achieving these goals is to use classic hardness amplification method with some careful modifications. Let p be as in Lemma 1, and define

$$q_1 := \frac{n}{(p)^2}, q_2 := \frac{n}{p} \text{ and } q := q_1 q_2.$$

Given as input a $q_1 \times q_2$ matrix of simples from (X_n, W_n, Z_n), M runs C on each column and outputs a vector of q_2 number of images of f (together with the corresponding pre-images and some auxiliary strings). The formal descriptions of algorithms M and Find are given below.

```
The Algorithm M
  input : ${(x_k, w_k, z_k)}_{k=1}^{q}$
    1. Arrange ${(x_k, w_k, z_k)}_{k=1}^{q}$ into $q_1 \times q_2$ tuples, denoted by ${(x_i^j, w_i^j, z_i^j)}_{i,j=1}^{q_2,q_1}$ .
    2. For $i = 1, 2, ..., q_2$, run $C$ on each $(x_i^j, w_i^j, z_i^j)$, $j \in [1, q_1]$, until $C$ outputs $(\beta, \alpha, aux)$. If for some $i$ all these $q_1$ runs of $C$ fail, return $\perp$; otherwise, set $(\beta_i, \alpha_i, aux_i)$ to be $(\beta, \alpha, aux)$.
  output: ${(\beta_i, \alpha_i, aux_i)}_{i=1}^{q_2}$.
```

```
The Algorithm Find
  input : ${(x_k, w_k, z_k)}_{k=1}^{q}$, ${(\beta_i, aux_i)}_{i=1}^{q_2}$
    1. Arrange ${(x_k, w_k, z_k)}_{k=1}^{q}$ in the same way as M and obtain ${(x_i^j, w_i^j, z_i^j)}_{i,j=1}^{q_2,q_1}$ .
    2. For $i = 1, 2, ..., q_2$, obtain the statement $x_i$ from $aux_i$, find the $j$-th entry $(x_i^j, w_i^j, z_i^j)$ from ${(x_i^j, w_i^j, z_i^j)}_{j=1}^{q_1}$ such that $x_i^j = x_i$ and fetch the corresponding $w_i^j$, set $w_i = w_i^j$ and run $E$ on input $(\beta_i, aux_i, w_i)$. If $E$ fails, output $\perp$, otherwise, set $\alpha_i$ to be the output of $E$.
  output: ${\alpha_i}_{i=1}^{q_2}$.
```

It easily follows from Lemma 2 that the algorithms M and Find enjoys the following security properties.

Lemma 3. *The following properties hold for algorithms* M *and* Find*:*

1. *The probability that* M *outputs* $\{(\beta_i, \alpha_i, aux_i)\}_{i=1}^{q_2}$ *such that* $\beta_i = f(\alpha_i)$ *holds for each* i *is negligibly close to* 1.
2. *Conditioned on* M *outputting* $\{(\beta_i, \alpha_i, aux_i)\}_{i=1}^{q_2}$*, the probability that* Find *inverts all these* β_i*'s successfully is negligibly close to* 1.
3. *Conditioned on* M *outputting* $\{(\beta_i, \alpha_i, aux_i)\}_{i=1}^{q_2}$ *, for any polynomial-size circuit family* T*, given as input only* $(\{(x_k, z_k)\}_{k=1}^{q}, \{(\beta_i, aux_i)\}_{i=1}^{q_2})$ *(without any witnesses to the* x_k*'s), the probability that* T *inverts all these* β_i*'s successfully is negligible.*
4. *For any two inputs to* Find *with different witnesses,* $(\{(x_k, w_k, z_k)\}_{k=1}^{q}, \{(\beta_i, aux_i)\}_{i=1}^{q_2})$ *and* $(\{(x_k, w'_k, z_k)\}_{k=1}^{q}, \{(\beta_i, aux_i)\}_{i=1}^{q_2})$ *such that* $\{w_k\}_{k=1}^{q} \neq \{w'_k\}_{k=1}^{q}$*,* Find *succeeds on each input with almost (negligibly close to each other) the same probability.*

The algorithm M generates q_2 number of images $(\beta_1, \beta_2, ..., \beta_{q_2})$ of one-way function $f : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ in a way such that they are hard for any polynomial-size circuit family (without knowing the corresponding witnesses) to invert simultaneously. This enables us to apply Goldreich-Levin hardcore predicate for the function of $f^{\bigotimes q_2}$ *with respect to the distribution on* $(\beta_1, \beta_2, ..., \beta_{q_2})$ generated by M. Formally, we need the following form of the Goldreich-Levin theorem.

Lemma 4 (Goldreich-Levin). *Let* $f : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ *be a function computable in polynomial time,* G *be a PPT algorithm. If for every polynomial-size circuit family* T*,*

$$\Pr[(f(x), aux) \leftarrow G(1^n) : T(1^n, f(x), aux) \in f^{-1}(f(x))] \leq negl(n),$$

then, the inner product of x *and a random* r *modulo* 2*, denoted by* $\langle x, r\rangle$*, is a hardcore predicate for* f*, i.e., for every polynomial-size circuit family* T'

$$\Pr[(f(x), aux) \leftarrow G(1^n), r \leftarrow \{0,1\}^n : T'(1^n, f(x), r, aux) = \langle x, r\rangle] \\ \leq \frac{1}{2} + negl(n).$$

The Goldreich-Levin theorem typically states for the distribution $f(U)$, i.e., for x being drawn from uniform distribution, but its proof ignores the distribution on the images of f and the auxiliary input (as long as both T and T' are given the same auxiliary string as input) completely, so the same proof applies to Lemma 4 (c.f. [Gol01]).

In our setting, this means that the inner product (modulo 2) $\langle(\alpha_1, \alpha_2, ..., \alpha_{q_2}), r \leftarrow \{0,1\}^{n \times q_2}\rangle$ is a hard core predicate for $f^{\bigotimes q_2} : \{0,1\}^{n \times q_2} \rightarrow \{0,1\}^{\ell(n) \times q_2}$ against arbitrary circuit families of polynomial size that takes as auxiliary input $(\{(x_k, z_k)\}_{k=1}^{q}, \{(\beta_i, aux_i)\}_{i=1}^{q_2})$.

6 Constructions for Public-Key Encryption and Key Agreement

In this section, we construct semantic secure (under chosen-plaintext-attack) public-key encryption and key agreement from a supposed adversary V^* against the Feige-Shamir protocol and an injective one-way function. This completes the proof of Theorem 1.

Let ϵ, q, q_2, M, Find and the infinite set I be as defined in previous sections. The final construction of public-key encryption scheme proceeds as follows. The key generation algorithm generates q number of YES instances together with their corresponding witnesses, $\{(x_{1,k}, w_{1,k})\}_{k=1}^{q}$, where $\{w_{1,k}\}_{k=1}^{q}$ is kept secret and $\{x_{1,k}\}_{k=1}^{q}$ is made public. To encrypt a bit m, the encryption algorithm generates $\{(x_{2,k}, w_{2,k})\}_{k=1}^{q}$, prepares a sequence of OR statements $\{(x_{1,k} \vee x_{2,k})\}_{k=1}^{q}$ (thus each $\{w_{b,k}\}_{k=1}^{q}$, $b \in [1, 2]$, are valid witnesses), and then applies M on $\{w_{2,k}\}_{k=1}^{q}$ to generate an image of $f^{\bigotimes q_2}$ and encrypts m using Goldreich-Levin; to decrypt the cipher-text, the decryption algorithm applies Find on $\{w_{1,k}\}_{k=1}^{q}$ as witnesses to obtain the corresponding pre-image and then computes the plain-text.

Formally, we need to assume the following for our constructions of public-key encryption and key agreement:

- An arbitrary *injective* one-way function $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ (used in the Feige-Shamir protocol). The injectiveness will be used for one party to recover the same hardcore bit that generated by the other party.
- An arbitrary efficiently samplable distribution ensemble $D = \{D_n\}_{n \in N}$ over R_L for an arbitrary NP language L.
- An arbitrary efficiently samplable distribution ensemble $\{Z_n\}_{n \in N}$ over $\{0,1\}^*$.
- A joint distribution ensemble $\{(X_n, W_n, Z_n)\}_{n \in N}$ on which the adversary V^* breaks the ϵ-distributional concurrent zero knowledge of Feige-Shamir protocol, where each distribution (X_n, W_n, Z_n) defined in the following way: Sample $(x_1, w_1) \leftarrow D_n$, $(x_2, w_2) \leftarrow D_n$, $z \leftarrow Z_n$, $b \leftarrow \{1, 2\}$, and output $((x_1, x_2), w_b)$.

We now construct public-key encryption for a single bit message on each security parameter n s.t. $(n, \cdot) \in I$.

Key generation $\mathsf{Gen}(1^n)$: $\{(x_{1,k}, w_{1,k})\}_{k=1}^{q} \leftarrow D_n^{\bigotimes q}$, and set $pk = \{x_{1,k}\}_{k=1}^{q}$, $sk = \{w_{1,k}\}_{k=1}^{q}$.

Encryption $\mathsf{Enc}(pk = \{x_{1,k}\}_{k=1}^{q}, m)$ ($m \in \{0, 1\}$):

1. $\{(x_{2,k}, w_{2,k})\}_{k=1}^{q} \leftarrow D_n^{\bigotimes q}$, $\{z_k\}_{k=1}^{q} \leftarrow Z_n^{\bigotimes q}$.

2. for $k \in [1, q]$, set x_k to be a random order of the pair $(x_{1,k}, x_{2,k})$.
3. $\{(\beta_i, \alpha_i, aux_i)\}_{i=1}^{q_2} \leftarrow \mathsf{M}(\{(x_k, w_{2,k}, z_k)\}_{k=1}^{q})$.
4. $r \leftarrow \{0,1\}^{n \times q_2}$, $h \leftarrow \langle(\alpha_1, \alpha_2, ..., \alpha_{q_2}), r\rangle \in \{0, 1\}$.
5. Output $c = (\{(x_k, z_k)\}_{k=1}^{q}, \{(\beta_i, aux_i)\}_{i=1}^{q_2}, r, h \bigoplus m)$.

Decryption $\mathsf{Dec}(sk = \{w_{1,k}\}_{k=1}^{q}, c)$:

1. Parse c into $\{(x_k, z_k)\}_{k=1}^{q} || \{(\beta_i, aux_i)\}_{i=1}^{q_2} || r || c'$.
2. $\{\alpha_i\}_{i=1}^{q_2} \leftarrow \mathsf{Find}(\{(x_k, w_{1,k}, z_k)\}_{k=1}^{q}, \{(\beta_i, aux_i)\}_{i=1}^{q_2})$.
3. $h \leftarrow \langle(\alpha_1, \alpha_2, ..., \alpha_{q_2}), r\rangle$.
4. Output $m = h \bigoplus c'$.

Notice that the input to M in the encryption algorithm can be viewed as being drawn from (X_n, W_n, Z_n) defined above. The correctness of this scheme follows from properties 1, 2, 4 of algorithms M and Find presented in the previous section. It should be noted that our scheme is not perfectly correct since it is possible for M/Find to fail during the encryption/decryption process. However, this happens only with negligible probability.

It is also easy to verify the semantic security under chosen-plaintext-attack, which is essentially due to the property 3 of M, together with the security of the hardcore bit for $f^{\bigotimes q_2}$.

Following the well-known paradigm, one can transform a semantic secure (under chosen-plaintext-attack) public-key encryption scheme into a key agreement protocol (A, B) with security against eavesdropping adversary in a simple way: the party A generates a public/secrete key pair and send the public-key to B, and then B sends back a ciphertext of the session secret key under A's public key to A. This establishes a common session secret key between A and B.

Extensions to Multiparty Key Agreement. Our key agreement protocol can be easily extended to the multiparty setting. Roughly, if V^* is able to break ϵ-distributional concurrent zero knowledge of the Feige-Shamir protocol on a distribution over instances of the form $(x_1 \vee x_2 \vee ... \vee x_n)$, then the n parties can establish a session secret key as follows. Each party A_i generates a sequence of pairs $\{(x_{i,k}, w_{i,k})\}_{k=1}^{q})$. In their first round the parties $A_1, A_2, ..., A_{n-1}$ send their sequences of $\{(x_{i,k})\}_{i,k=1}^{n-1,q})$ to the n-th party, then the n-th party uses these sequences as a public key of the above public-key encryption scheme to encrypt the session secret key and send the ciphertext to all $n-1$ parties. Upon receiving the ciphertext, each A_i, $i = [1, n-1]$, decrypts it and obtains the session secret key using their own $\{(w_{i,k})\}_{k=1}^{q}$.

7 Concluding Remarks

We prove a win-win result regarding the complexity of public-key encryption and the round-complexity of concurrent zero knowledge. We believe that when one can prove one of these two statements listed in Theorem 1, one might obtain a much stronger result (e.g., result with respect to the (nicer) standard definition) than the ones stated therein. The ideas and techniques used here may be applied to investigate some other black-box lower bounds in cryptography.

Our result can be viewed as a step toward breaking the known black-box or universal reduction barriers, and a proof (or disproof) of either one of the two statements in Theorem 1 will be exciting. A construction of public-key encryption (key agreement) from general one-way functions will, borrowing from the Impagliazzo's terminology [Imp95], rule out the world Minicrypt and build for the first time the world Cryptomania from (trapdoor/algebraic) structure-free hardness assumption, which definitely is a major achievement in cryptography.

On the other hand, a concurrent security proof of the Feige-Shamir protocol will also be an exciting breakthrough, both technically and conceptually. On the technical level, such a proof will reveal a fascinating fact that all possible efficient adversaries against the Feige-Shamir protocol have *in common* a highly non-trivial structure of computation– e.g., the existence of those good extractors $\{T_i\}_{i\in\mathbb{N}}$ from the second claim in Section 3.2, which might shed light on the longstanding open problem of constructing extractable one-way functions from standard assumptions; on the conceptual level, it will bring a new individual reduction/simulation for cryptography and refute the impression that a new reduction technique always gives more complicated and inefficient constructions.

8 Acknowledgement

We are grateful to Nir Bitansky, Omer Paneth, Alon Rosen and anonymous reviewers for constructive comments and suggestions. We thank Yu Chen and Jiang Zhang for helpful discussions, and Yanyan Liu, Shunli Ma, Hailong Wang, Bo Wu, Zhenbin Yan and Jingyue Yu for careful proofreading.

References

[Adl78] Leonard M. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science - FOCS'78*, pages 75–83. IEEE Computer Society, 1978.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42th Annual IEEE Symposium on Foundations of Computer Science - FOCS'01*, pages 106–115. IEEE Computer Society, 2001.

[BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In *Advances in Cryptology - ASIACRYPT'13*, LNCS 8269, pages 296–315. Springer, 2013.

[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.

[BHSV98] Mihir Bellare, Shai Halevi, Amit Sahai, and Salil Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In *Advances in Cryptology - CRYPTO'98*, LNCS 1462, pages 283–298. Springer, 1998.

[BP15] Nir Bitansky and Omer Paneth. On non-black-box simulation and the impossibility of approximate obfuscation. In *SIAM Journal on Computing*, Valume 44(5), pages 1325–1383, 20015.

[BU08] Michael Backes and Dominique Unruh. Limits of constructive security proofs. In *Advances in Cryptology - ASIACRYPT'08*, LNCS 5350, pages 290–307. Springer, 2008.

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *Theory of Cryptography - TCC'05*, LNCS 3378, pages 17–33. Springer, 2005.

[CKPR01] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires omega(log n) rounds. In *Proceedings of the 33rd Annual ACM Symposium Theory of Computing- STOC'01*, pages 570–579. ACM press, 2001.

[CLMP13] Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In *ITCS 2013*, pages 389–400, 2013.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party computation. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing - STOC'02*, pages 494–503. ACM Press, 2002.

[CLP13a] Ran Canetti, Huijia Lin, and Omer Paneth. Public-coin concurrent zero-knowledge in the global hash model. In *Theory of Cryptography - TCC'13*, LNCS 7785, pages 80–99. Springer, 2013.

[CLP13b] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero knowledge from p-certificates. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science - FOCS'13*, pages 50–59. IEEE Computer Society, 2013.

[CLP15a] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO'15*, LNCS 9216, pages 287–307. Springer, 2015.

[CLP15b] 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.

[CS99] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *ACM conference on Computer and Communications Security - CCS'99*, pages 46–52. ACM Press, 1999.

[Dam91] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *Advances in Cryptology - CRYPTO'91*, LNCS 576, pages 445–456. Springer, 1991.

[Den16] Yi Deng. Magic adversaries versus individual reduction: Science wins either way. Cryptology ePrint Archive, Report 2016/1107, 2016.

[DGL$^+$16] Yi Deng, Juan A. Garay, San Ling, Huaxiong Wang, and Moti Yung. On the implausibility of constant-round public-coin zero-knowledge proofs. In *Security in Communication Networks - SCN'16*, LNCS 9841, pages 237–253. Springer, 2016.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science - FOCS'09*, pages 251–260. IEEE Computer Society, 2009.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

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

[DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *Proceedings of the 30rd Annual ACM Symposium Theory of Computing- STOC'98*, pages 409–418. ACM press, 1998.

[DS16] Dana Dachman-Soled. Towards non-black-box separations of public key encryption and one way function. In *Theory of Cryptography - TCC'16B*, LNCS 9986, pages 169–191. Springer, 2016.

[FS89] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In *Advances in Cryptology - CRYPTO'89*, LNCS 435, pages 526–544. Springer, 1989.

[GGJ13] Vipul Goyal, Divya Gupta, and Abhishek Jain. What information is leaked under concurrent composition? In *Advances in Cryptology - CRYPTO'13*, LNCS 8043, pages 220–238. Springer, 2013.

[GGJS12] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai:. Concurrently secure computation in constant rounds. In *Advances in Cryptology - Eurocrypt'12*, LNCS 7237, pages 99–116. Springer, 2012.

[GGS15] Vipul Goyal, Divya Gupta, and Amit Sahai. Concurrent secure computation via non-black box simulation. In *Advances in Cryptology - CRYPTO'15*, LNCS 9216, pages 23–42. Springer, 2015.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[GLP$^+$15] Vipul Goyal, Huijia Lin, Omkant Pandey, Rafael Pass, and Amit Sahai. Round-efficient concurrently composable secure computation via a robust extraction lemma. In *Theory of Cryptography - TCC'15*, LNCS 9014, pages 260–289. Springer, 2015.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the 14rd Annual ACM Symposium Theory of Computing- STOC'82*, pages 365–377. ACM press, 1982.

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

[GNW95] Oded Goldreich, Noam Nisan, and Avi Wigderson. On yaos xor-lemma. In *Electronic colloquium on computational complexity, TR95-050*, 1995.

[Gol93] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol01] Oded Goldreich. *Foundations of Cryptography*, volume Basic Tools. Cambridge University Press, 2001.

[Goy13] Vipul Goyal. Non-black-box simulation in the fully concurrent setting. In *Proceedings of the 45th Annual ACM Symposium on the Theory of Computing - STOC'13*, pages 221–230. ACM Press, 2013.

[HILL99] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM. Journal on Computing*, 28(4):1364–1396, 1999.

[HKS03] Dennis Hofheinz, Eike Kiltz, and Victor Shoup. Practical chosen ciphertext secure encryption from factoring. *Journal of Cryptology*, 26(1):102–118, 2003.

[HR07] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. In *Proceedings of the 39rd Annual ACM Symposium Theory of Computing- STOC'07*, pages 1–10. ACM press, 2007.

[HS11] Thomas Holenstein and Grant Schoenebeck. General hardness amplification of predicates and puzzles. In *Theory of Cryptography - TCC'11*, LNCS 6597, pages 19–36. Springer, 2011.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science - FOCS'89*, pages 230–235. IEEE Computer Society, 1989.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Annual IEEE Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society, 1995.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21th Annual ACM Symposium on the Theory of Computing - STOC'89*, pages 44–61. ACM Press, 1989.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[KP01] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in polyloalgorithm rounds. In *Proceedings of the 33rd Annual ACM Symposium Theory of Computing- STOC'01*, pages 560–569. ACM press, 2001.

[Lin03a] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *Proceedings of the 35rd Annual ACM Symposium Theory of Computing- STOC'03*, pages 683–692. ACM press, 2003.

[Lin03b] Yehuda Lindell. General composition and universal composability in secure multiparty computation. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science - FOCS'03*, pages 394–403. IEEE Computer Society, 2003.

[Lin08] Yehuda Lindell. Lower bounds and impossibility results for concurrent self composition. *Journal of Cryptology*, 21(2):200–249, 2008.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Annual ACM Symposium on the Theory of Computing - STOC'90*, pages 427–437. ACM Press, 1990.

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing - STOC'04*, pages 232–241. ACM Press, 2004.

[PPS15] Omkant Pandey, Manoj Prabhakaran, and Amit Sahai. Obfuscation-based non-black-box simulation and four message concurrent zero knowledge for np. In *Theory of Cryptography - TCC'15*, LNCS 9015, pages 638–667. Springer, 2015.

[PR03] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science - FOCS'03*, pages 404–413. IEEE Computer Society, 2003.

[PR05] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science - FOCS'05*, pages 563–572. IEEE Computer Society, 2005.

[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.

[Rab79] Michael Rabin. Digitalized signatures and public-key encryptions as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6), 2009.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22rd Annual ACM Symposium Theory of Computing-STOC'90*, pages 387–394. ACM press, 1990.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography - TCC'04*, LNCS 2951, pages 1–20. Springer, 2004.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science - FOCS'99*, pages 543–553. IEEE Computer Society, 1999.