Concurrent Knowledge Extraction in the Public-Key Model*

Andrew C. Yao[a], Moti Yung[b], and Yunlei Zhao[c]**

[a]ITCS, Tsinghua University, Beijing, China
[b]Google Inc. and Columbia University, New York, USA.
[c]Software School, Fudan University, Shanghai, China

Abstract. Knowledge extraction is a fundamental notion, modeling machine possession of values (witnesses) in a computational complexity sense and enabling one to argue about the internal state of a party in a protocol without probing its internal secret state. However, when transactions are concurrent (e.g., over the Internet) with players possessing public-keys (as is common in cryptography), assuring that entities "know" what they claim to know, where adversaries may be well coordinated across different transactions, turns out to be much more subtle and in need of re-examination. Here, we investigate how to formally treat knowledge possession by parties (with registered public-keys) interacting over the Internet. Stated more technically, we look into the relative power of the notion of "concurrent knowledge-extraction" (CKE) in the concurrent zero-knowledge (CZK) bare public-key (BPK) model where statements being proven can be dynamically and adaptively chosen by the prover.

We show the potential vulnerability of man-in-the-middle (MIM) attacks turn out to be a real security threat to existing natural protocols running concurrently in the public-key model, which motivates us to introduce and formalize the notion of CKE, alone with clarifications of various subtleties. Then, both generic (based on standard polynomial assumptions), and efficient (employing complexity leveraging in a novel way) implementations for $\mathcal{NP}$ are presented for constant-round (in particular, round-optimal) concurrently knowledge-extractable concurrent zero-knowledge (CZK-CKE) arguments in the BPK model. The efficient implementation can be further practically instantiated for specific number-theoretic language.

1 Introduction

Zero-knowledge (ZK) protocols allow a prover to assure a verifier of validity of theorems without giving away any additional knowledge (i.e., computational advantage) beyond validity. This notion was introduced in [14], and its generality was demonstrated in [13]. Traditional notion of ZK considers the security in a stand-alone (or sequential) execution of the protocol. Motivated by the use of such protocols in an asynchronous network like the Internet, where many protocols run simultaneously, studying security properties of ZK protocols in such concurrent settings has attracted much research efforts in recent years. Informally, a ZK protocol is called concurrent zero-knowledge if

* This work was supported in part by the National Basic Research Program of China Grant Nos.2007CB807900, 2007CB807901, the National Natural Science Foundation of China Grant Nos.60553001, 60703091, and the QiMingXing Program of Shanghai.

** Contact author. ylzhao@fudan.edu.cn

concurrent instances are all (expected) polynomial-time simulatable, namely, when a possibly malicious verifier concurrently interacts with a polynomial number of honest prover instances and schedules message exchanges as it wishes.

The concept of "proof of knowledge" (POK), informally discussed in [14], was then formally treated in [1, 11, 2]. POK systems, especially zero-knowledge POK (ZKPOK) systems, play a fundamental role in the design of cryptographic schemes, enabling a formal complexity theoretic treatment of what does it mean for a machine to "know" something. Roughly speaking, a "proof of knowledge" means that a possibly malicious prover can convince the verifier that an $\mathcal{NP}$ statement is true if and only if it, in fact, "knows" (i.e., possesses) a witness to the statement (rather than merely conveying the fact that a corresponding witness exists). With the advancement of cryptographic models where parties first publish public-keys (e.g., for improving round complexity [5]) and then may choose the statements to prove, knowledge extraction becomes more subtle (due to possible dependency on published keys), and needs re-examination. Here, we investigate the relative power of the notion of "concurrent knowledge-extraction" in the concurrent zero-knowledge BPK model with adaptive input selection.

The BPK model, introduced in [4], is a natural cryptographic model. A protocol in this model simply assumes that all verifiers have each deposited a public key in a public file (which are referred to as the *key generation stage*), before user interactions take place (which are referred to as the *proof stage*). No assumption is made on whether the public-keys deposited are unique or valid (i.e., public keys can even be "nonsensical," where no corresponding secret-keys exist or are known). In many cryptographic settings, availability of a public key infrastructure (PKI) is assumed or required, and in these settings the BPK model is, both, natural and attractive (note that the BPK model is, in fact, a weaker version of PKI where in the later added key certification is assumed). It was pointed out by Micali and Reyzin [16] that the BPK model is, in fact, applicable to interactive systems in general.

Verifier security in the BPK model (against malicious provers) turned out to be more involved than anticipated, as was demonstrated by Micali and Reyzin [16] who showed that under standard intractability assumptions there are four distinct meaningful notions of soundness, i.e., from weaker to stronger: one-time, sequential, concurrent and resettable soundness. Here, we focus on concurrent soundness, which, roughly speaking, means that a possibly malicious probabilistic polynomial-time (PPT) prover P^* cannot convince the honest verifier V of a *false* statement even when P^* is allowed multiple interleaving interactions with V in the public-key model. They also showed that any black-box ZK protocol with concurrent soundness in the BPK model (for non-trivial languages outside $\mathcal{BPP}$) must run at least four rounds [16].

Concurrent soundness only guarantees that concurrent interactions cannot help a malicious prover validate a *false* statement in the public-key model. However, it does *not* prevent a malicious prover from validating a *true* statement *but* without knowing any witness for the statement being proved. This potential vulnerability is not merely a theoretical concern: In fact, most concurrent ZK protocols in the BPK model involve a sub-protocol in which the verifier proves to the prover the knowledge of the secret-key corresponding to its public-key. A malicious prover, in turn, can (as we show) exploit these sub-proofs by the verifier in other sessions, without possessing a witness to these

sessions' statements. This issue, in turn, motivates the need for careful definitions and for achieving concurrent verifier security for concurrent ZK in the BPK model for adaptively chosen proofs, so that one can remedy the above security vulnerability.

Our contributions. We first investigate the subtleties of concurrent verifier security in the public-key model in the case of proof of knowledge for dynamically chosen input languages. Specifically, we show concurrent interleaving and malleating attacks against some existing natural protocols running concurrently in the BPK model, which shows that concurrent soundness and normal arguments of knowledge (and also traditional concurrent non-malleability) do not guarantee concurrent verifier security in the public-key model.

Then, we formulate concurrent verifier security that remedies the vulnerability as demonstrated by the concrete attacks which are of the concurrent man-in-the-middle (CMIM) nature, along with subtlety clarifications and discussion. The security notion defined is named `concurrent knowledge-extraction (CKE)` in the public-key model, which essentially means that for adaptively chosen statements whose validations are successfully conveyed by a possibly malicious prover to an honest verifier by concurrent interactions, the prover must "know" the corresponding witnesses in a sense that the knowledge known by the prover is "independent" of honest verifier's secret-key.

We then present both generic (based on standard polynomial assumptions) and efficient (employing complexity leveraging in a novel way) black-box implementations of constant-round (in particular, round-optimal) CZK-CKE arguments for $\mathcal{NP}$ in the BPK model. The efficient implementation can be, further, practically instantiated for specific important number-theoretic languages.

2 Preliminaries

In this section, we briefly recall some basic tools and definitions.

Commitments. Commitment schemes enable a party, called the *sender*, to bind itself to a single value in the initial *commitment* stage, while keeping it unknown to the *receiver* (this property is called *hiding*). Furthermore, when the commitment is opened in a later *decommitment* stage, it is guaranteed that the "opening" can yield only the single value determined in the commitment phase (this property is called *binding*).

One-round perfectly-binding commitments can be based on any one-way permutation (OWP) [13], whereas tow-round statistically-binding commitments can be based on any one-way function (OWF) [17]. In addition, practical statistically-binding commitments can be implemented under the decisional Diffie-Hellman (DDH) assumption. On the other hand, one-round statistically-hiding commitments can be based on any collision-resistant hash function [15]. Two-round statistically-hiding commitments can be based on any claw-free collection with efficiently recognizable indices [11], and three-round statistically-hiding commitments can be based on any OWF admitting Σ-protocols [22].

Σ-protocols and Σ_{OR}-protocols. Informally, a Σ-protocol is itself a 3-round public-coin *special* honest verifier zero-knowledge (SHVZK) protocol with special soundness in the knowledge-extraction sense. A Σ-protocol is called computational/statistical Σ-protocol, if it is computational/statistical SHVZK. A very large number of Σ-protocols

have been developed in the literature. In particular, (the parallel repetition of) Blum's protocol for DHC [3] is a computational Σ-protocol for $\mathcal{NP}$, and most practical Σ-protocols for number-theoretical languages are of *perfect* SHVZK property. One basic construction with Σ-protocols is the OR of a real and simulated transcript, called Σ_{OR} [6], that is a concrete witness indistinguishability protocol.

Witness Indistinguishability (WI). A protocol is called WI (resp., statistical WI) for an $\mathcal{NP}$-language L, if the views of any PPT malicious verifier V^* in two runs of the protocol, w.r.t. the same common input $x \in L$ and the same auxiliary input $z \in \{0,1\}^*$ to V^* but (possibly) different private witnesses to the prover, are computationally (resp., statistically) indistinguishable. WI is preserved under concurrent composition.

In this work, we employ, in a critical way, constant-round *statistical* WI argument/proof of knowledge (WIA/POK). We briefly note two simple ways to implement statistical WIA/POK. First, for any statistical/perfect Σ-protocol, the OR-proof (i.e., the Σ_{OR}-protocol [6]) is statistical/perfect WIPOK. The second approach is to modify Blum's protocol for DHC [3] (that is computational WIPOK) into constant-round statistical WIAOK, by replacing the statistically-binding commitments used in the first-round of Blum's protocol by constant-round *statistically-hiding* commitments.

Strong WI (SWI) [11]. A protocol $\langle P, V\rangle$ for a language L (with $\mathcal{NP}$-relation R_L) is called SWI, if the views of any PPT malicious verifier V^* in two runs of the protocol, $\langle P(w_0), V^*(z_0)\rangle(x_0)$ and $\langle P(w_1), V^*(z_1)\rangle(x_1)$, are indistinguishable, whenever the distributions (x_0, z_0) and (x_1, z_1) are indistinguishable (where $(x_b, w_b) \in R_L$ for $b \in \{0,1\}$). Any ZK protocol is itself SWI [11]. Different from regular WI, SWI is not preserved under concurrent composition [12]. But, an SWI protocol can be easily transferred into a regular WI protocol: On common input x and private witness w, the prover commits w to c_w, and then proves that the value committed to c_w is a valid witness for $x \in L$. Such a protocol is called *commit-then-SWI*, which is regular WI for L.

The BPK model *with adaptive language selection*. We say a class of languages $\mathcal{L}$ is *admissible* to a protocol $\langle P, V\rangle$ if the protocol can work (i.e., be instantiated) for any language $L \in \mathcal{L}$. Typically, $\mathcal{L}$ could be the set of all $\mathcal{NP}$-languages (via $\mathcal{NP}$-reduction in case $\langle P, V\rangle$ can work for an $\mathcal{NP}$-complete language) or the set of any languages admitting Σ-protocols (in this case $\langle P, V\rangle$ could be instantiated for any language in $\mathcal{L}$ efficiently without going through general $\mathcal{NP}$-reductions). For protocols in the BPK model, let R_{KEY} be an $\mathcal{NP}$-relation validating the public-key and secret-key pair (PK, SK) generated by any honest verifier, i.e., $R_{KEY}(PK, SK) = 1$ indicates that SK is a valid secret-key corresponding to PK.

In this work, for concurrent verifier security of a protocol in the BPK model, we consider an s-concurrent malicious prover P^* that, on a system parameter n, interacts with honest verifier instances in at most $s(n)$ sessions, where $s(\cdot)$ is a polynomial. Furthermore, different from the traditional BPK model formulation [4, 16], we assume P^* can set the admissible languages (to be proved to honest verifiers) *that may potentially depend on honest verifiers' public-keys*. Though it may be more difficult to achieve concurrent verifier security against adversaries with adaptive language selection in the BPK model, this is a far more realistic model for cryptographic protocols running concurrently in the public-key model where mixing the public-key structure as part of the language is a natural adversarial strategy. For any $(PK, SK) \in R_{KEY}$, we denote

by $view_{P^*}^{V(SK)}(1^n, z, PK)$ the random variable describing the view of P^* specific to PK, which includes its random tape, the auxiliary string z, the public-key PK, and all messages it receives from the instances of the honest verifier V of secret-key SK.

3 Concurrent Knowledge-Extraction: Motivation, Formulation and Discussion

We show a concurrent interleaving and malleating attack on the concurrent ZK protocol of [7, 23] that is both *concurrently sound* and *normal argument of knowledge* (AOK) in the BPK model, in which by concurrent interactions a malicious prover P^* can (with probability 1) convince an honest verifier of a true (*public-key related*) statement but without knowing any witness to the statement being proved. Due to space limitation, the reader is referred to the full paper [20] for the attack details. This shows that concurrent soundness and normal AOK do not guarantee that an adversary does "know" what it concurrently claims to know against an honest verifier in the public-key model. This concrete attack (on *naturally existing* concurrently sound CZKAOK in the BPK model) serves a good motivation for understanding "possession of knowledge on the Internet with registered keys", i.e., the subtleties of concurrent knowledge-extraction in the public-key model. We note that this attack is of a *man-in-the-middle* nature, and is related to malleability of protocols.

Now, we proceed to formulate concurrent verifier security in light of the attack against the protocol of [7, 23]. The security notion assuring that a malicious prover P^* does "know" what it claims to know, when it is concurrently interacting with the honest verifier V, can informally be formulated as: for any x, if P^* can convince V (with public-key PK) of "$x \in L$" (for an $\mathcal{NP}$-language L) by concurrent interactions, then there exists a PPT knowledge-extractor that outputs a witness for $x \in L$. This is a natural extension of the normal arguments of knowledge into the concurrent public-key setting. However, this formulation approach is problematic in the concurrent public-key setting. The reason is: the statements being proved may be related to PK, and thus the extracted witness may be related to the corresponding secret-key SK (even just the secret-key as shown by the concrete attack on the protocol of [7, 23]); But, in knowledge-extraction the PPT extractor may have already possessed SK. To solve this subtlety, we require the extracted witness, together with adversary's view, to be *independent* of SK. But, the problem here is how to formalize such independence, in particular, w.r.t. a CMIM? We solve this in the spirit of non-malleability formulation [9]. That is, we consider the message space (distribution) of SK, and such independence is roughly formulated as follows: let SK be the secret-key and SK' is an element randomly and independently distributed over the space of SK, then we require that, for any polynomial-time computable relation R, the probability $\Pr[R(\bar{w}, SK, view) = 1]$ is negligibly close to $\Pr[R(\bar{w}, SK', view) = 1]$, where $\bar{w}$ is the set of witnesses extracted by the knowledge extractor for successful concurrent sessions and $view$ is the view of P^*. This captures the intuition that P^* does, in fact, "know" the witnesses to the statements whose validations are successfully conveyed by concurrent interactions.

Definition 1 (concurrent knowledge-extraction (CKE) in the public-key model). *We say that a protocol $\langle P, V \rangle$ is* concurrently knowledge-extractable *in the BPK model w.r.t. some admissible language set $\mathcal{L}$ and some key-validating relation R_{KEY},*

if for any positive polynomial $s(\cdot)$, any s-concurrent malicious prover P^*, there exist a pair of (expected) polynomial-time algorithms S (the simulator) and E (the extractor) such that for any sufficiently large n, any auxiliary input $z \in \{0,1\}^*$, and any polynomial-time computable relation R (with components drawn from $\{0,1\}^* \cup \{\perp\}$), the following hold in accordance with the experiment $\mathbf{Expt}_{\mathbf{CKE}}(1^n, z)$ described below:

$\mathbf{Expt}_{\mathbf{CKE}}(1^n, z)$

The simulator $S = (S_{KEY}, S_{PROOF})$:
$(PK, SK, SK') \longleftarrow S_{KEY}(1^n)$, where the distribution of (PK, SK) is identical with that of the output of the key-generation stage of the honest verifier V, $R_{KEY}(PK, SK) = R_{KEY}(PK, SK') = 1$ and the distributions of SK and SK' are identical and *independent*. In other words, SK and SK' are two random and independent secret-keys corresponding to PK.

$(str, sta) \longleftarrow S_{PROOF}^{P^*(1^n,\, PK,\, z)}(1^n, PK, SK, z)$. That is, on inputs $(1^n, PK, SK, z)$ and with oracle access to $P^*(1^n, PK, z)$ (by providing random tape to P^* and running P^* as subroutine), the simulator S outputs a simulated transcript str, and some state information sta to be transformed to the knowledge-extractor E.

We denote by $S_1(1^n, z)$ the random variable str (in accordance with above processes of S_{KEY} and S_{PROOF}). For any $(PK, SK) \in R_{KEY}$ and any $z \in \{0,1\}^*$, we denote by $S_1(1^n, PK, SK, z)$ the random variable describing the first output of $S_{PROOF}^{P^*(1^n,\, PK,\, z)}(1^n, PK, SK, z)$ (i.e., str specific to (PK, SK)).

The knowledge-extractor E:
$\overline{w} \longleftarrow E(1^n, sta, str)$. On (sta, str), E outputs a list of witnesses to statements whose validations are successfully conveyed in str.

- **Simulatability.** *The following ensembles are indistinguishable: $\{view_{P^*}^{V(SK)}(1^n, z, PK)\}_{(PK,SK)\in R_{KEY}, z\in\{0,1\}^*}$ and $\{S_1(1^n, PK, SK, z)\}_{(PK,SK)\in R_{KEY}, z\in\{0,1\}^*}$.*
- **Secret-key independent knowledge-extraction.** *E, on inputs $(1^n, str, sta)$, outputs witnesses to all statements successfully proved in accepting sessions in str. Specifically, E outputs a list of strings $\overline{w} = (w_1, w_2, \cdots, w_{s(n)})$, satisfying the following:*
  - *w_i is set to be $\perp$, if the i-th session in str is not accepting (due to abortion or verifier verification failure), where $1 \le i \le s(n)$.*
  - Correct knowledge-extraction (for individual statements): *In any other cases (i.e., for successful sessions), with overwhelming probability $(x_i, w_i) \in R_L$, where x_i is the common input selected by P^* for the i-th session in str and R_L is the admissible $\mathcal{NP}$-relation for $L \in \mathcal{L}$ set by P^* in str.*
  - (Joint) knowledge extraction independence (KEI): $\Pr[R(SK, \overline{w}, str) = 1]$ *is negligibly close to* $\Pr[R(SK', \overline{w}, str) = 1]$.

  The probabilities are taken over the randomness of S in the key-generation stage (i.e., the randomness for generating (PK, SK, SK')) and in all proof stages, the randomness of E, and the randomness of P^*. If the KEI property holds for any (not necessarily polynomial-time computable) relation R, we say the protocol $\langle P, V\rangle$ satisfies statistical *CKE.*

We first note that the above CKE formulation follows the simulation-extraction approach of [19]. Here, the key augmentation, besides some other adaptations in the public-key model, is the property of knowledge-extraction independence (KEI) explicitly required (the KEI notion originally appeared in the incomplete work of [23], August 2006 update). Though the CKE and KEI notions are formulated in the framework of public-key model, they are actually applicable to protocols in the plain model, in general, in order to capture knowledge extractability against concurrent adversaries interacting with honest players of secret values.

Below, we discuss and clarify various subtleties surrounding the CKE formulation. More details are referred to the full paper [20].

Simulated public-keys vs. real public-keys. In our CKE formulation, the simulation-extraction is w.r.t. *simulated* public-keys. A natural and intuitive strengthening of the CKE formulation might be: the simulator/extractor uses the *same* public-keys of the honest verifiers. In this case, as the simulator/extractor does not possess honest verifier's secret-key, the KEI property can be waived. But, the observation here is: constant-round CKE (*whether ZK or not*) with real public-keys are impossible. Specifically, constant-round CKE with real public-keys implies constant-round CZK (potentially, concurrent non-malleable ZKPOK) *in the plain model* by viewing verifier's public-keys as a part of common inputs, which is however impossible at least in the black-box sense [5].

On CKE with independent language. With the above KEI formulation, we are actually formulating the independence of the witnesses, used ("*known*") by CMIM adversary, on the secret-key (witness) used by verifier (who may in turn play the role of prover in some sub-protocols). A naive solution for KEI, which appears to make sense, may be to require the language and statements being proved are independent of verifier's public-keys. But, this approach has the following problems: Firstly, if the protocol is for $\mathcal{NP}$-Complete, the statements being proved, selected adaptively by the adversary, can always be related to verifier's public-key (e.g., via $\mathcal{NP}$-reductions); Secondly, as the statements being proved are selected adaptively by the CMIM adversary on the fly, in general it is hard to distinguish whether the maliciously chosen statements are independent of verifiers' public-keys or not; Thirdly, the applicability of this approach is significantly limited (and even useless in practice, where keys are used in essential ways in malicious settings like the Internet).

CKE vs. concurrent soundness. As a consequence of the attack on the CZK protocol of [7, 23] that is both concurrently sound and can be implemented based on any OWF, we show that, assuming any OWF, CKE is a strictly stronger notion for concurrent verifier security than concurrent soundness in the public-key model. We note that, prior to our work, whether A/POK is strictly stronger than soundness (in the concurrent public-key setting) is unknown.

Taking adversary's view, i.e., str, into account for capturing KEI. We note this is necessary for the completeness of KEI formulation. Specifically, consider the following (seemingly impossible) case that: for any extracted w_i in $\bar{w}$, $w_i = PRF_s(SK)$, where the seed s could be either a part of the adversary's random tape or a value computed from its view. In other words, the witnesses extracted are always dependent on the secret-key used by the simulator/extracotor, and thus the adversary may not necessarily be aware of the extracted knowledge. But, without taking account of adversary's view,

$\Pr[R(SK, \bar{w}) = 1]$ is still negligibly close to $Pr[R(SK', \bar{w}) = 1]$ in this case for any polynomial-time computable relation R.

We note that, explicitly taking account of adversary's view seems to be necessary for correct and complete CNM formulations, whenever (not necessarily extractable) knowledge independence is a necessary property to be considered. We note that this issue is applicable to some related works, and can also be traced back to the origin of NM formulation [9].

On extending the Bellare-Goldreich (BG) quantitative approach for stand-alone POK into the concurrent setting. We note that, besides the subtle KEI issue, there are some difficulties (or inconveniences) to extend the BG quantitative approach for stand-alone POK [1, 11, 2] (i.e., the quantitative definition of expected knowledge-extraction time that is in inverse proportion to the probability the adversary convinces of the statement) into the concurrent setting. Below, we consider two possible approaches to extend the BG quantitative approach (for stand-alone POK) into the concurrent setting.

The first approach is: for each of all the concurrent sessions, we consider the probability that the adversary (i.e., the malicious prover P^*) successfully finishes the session. Denote by p_i the probability that the adversary successfully finish the i-th session. Note that this probability is particularly taken over the random coins of P^* and all random coins of the honest verifier instances in all concurrent sessions. But, within the simulation-extraction formulation framework, it is difficult to give a precise quantitative definition of the knowledge-extraction time inversely proportional to p_i. The reason is: when we apply the underlying stand-alone knowledge-extractor (guaranteed by the Bellare-Goldreich POK definition) on the successful i-th session in the simulated transcript, the knowledge-extraction is actually with respect to the probability, denote p'_i, that P^* successfully finish the i-th session when the coins of the honest verifier instances in all other sessions (other than the i-th session) are fixed (i.e., determined by the simulated transcript str). Clearly, p'_i can be totally different from p_i (e.g., p_i may be non-negligible, but p'_i can be negligible), and thus the knowledge-extraction time w.r.t p'_i can be totally different from that w.r.t p_i.

The second approach is to separate the simulation and knowledge-extraction. Specifically, besides indistinguishable simulation, we *separately* require (regardless of the simulated transcript) that for any x *selected adaptively by the adversary during its concurrent attack*, if the adversary P^* can, with probability p_x, convince the honest verifier of the statement "$x \in L$" in one of the $s(n)$ sessions by concurrent interactions, the knowledge-extraction time should be in inverse proportion to p_x. We note that this approach does not work. On the one hand, suppose P^* convinces $x \in L$ in one of the $s(n)$ sessions (say the i-th session) with some non-negligible probability, but with negligible probability in all other sessions. In this case, it is okey if the knowledge extraction is w.r.t. the i-th session, but will fail w.r.t other sessions. On the other hand, one may argue that to remedy the above subtlety, we can add a (polynomial-time) bound on the knowledge-extraction in each session, but this solution fails if the adversary convinces of the statement "$x \in L$" with negligible probability in all sessions. In general, it may be hard to distinguish the two cases, i.e., the case that P^* succeeds with negligible probability in all sessions and the case that P^* may succeed with non-negligible probability in some (but not all) sessions.

We note that the work [8] takes the approach of extending the BG (stand-alone) POK formulation into the concurrent setting in the BPK model, *without clarifying the above subtleties*. For example, the running time of the knowledge-extractor E formulated in [8] is w.r.t. the probability p_i, but it is unclear how to handle the issue of p_i' versus p_i as clarified above. In addition, the work [8] does not capture adaptive language selection by the concurrent malicious prover, and does not capture the KEI issue (it is unclear how about if the knowledge extracted by E is dependent on verifier's secret-key that is actually generated by E itself). As a consequence, the formulation approach of [8] may be less convenient to use (particularly for analyzing complex cryptographic protocols running concurrently with public-keys). To our knowledge, still no formal proofs in accordance with the formulation approach of [8] are presented in existing works. In comparison, we suggest our CKE formulation is of conceptual clarity and simplicity, is easier to work with and can be efficiently achievable, and is well compatibility of the normal simulation/extraction formulation approach for concurrent security of protocols. We also remind that our CKE formulation implicitly assumes that verifier's public-key corresponds to multiple secret-keys (in the sense that protocols with unique secret-key for the verifier may trivially *not* satisfy the CKE security), which however can typically be achieved with the common key-pair trick [18]. In general, cryptography literature should welcome diversified approaches for modeling and achieving security goals of cryptographic systems, particularly witnessed by the evolution history of public-key encryption.

4 Overview of Achieving CZK-CKE in the BPK Model

In this section, we present the high-level overview of achieving constant-round CZK-CKE arguments in the BPK model, with details referred to the full paper [20].

The starting point is the basic and central Feige-Shamir ZK (FSZK) structure [10]. The FSZK structure is conceptually simple and is composed of two WIPOK sub-protocols. In more details, let f be a OWF, in the first WIPOK sub-protocol with the verifier V serving as the knowledge-prover, V computes $(y_0 = f(s_0), y_1 = f(s_1))$ for randomly chosen s_0 and s_1; then V proves to the prover P the knowledge of the preimage of either y_0 or y_1. In the second WIPOK sub-protocol with P serving as the knowledge-prover for an $\mathcal{NP}$-language L, on common input x, P proves to V the knowledge of either a valid $\mathcal{NP}$-witness w for $x \in L$ or the preimage of either y_0 or y_1. FSZK is also argument of knowledge, and can be instantiated practically (without going through general $\mathcal{NP}$-reductions) by the Σ_{OR} technique [6, 23].

Let (y_0, y_1) serve as the public-key of V and s_b (for a random bit b) as the secret-key, the public-key version of FSZK is CZK in the BPK model [23]. But, we show that the public-key version of FSZK is not of concurrently soundness [21], needless to say concurrent knowledge-extractability (indeed, FSZK was not designed for the public-key model). We hope to add the CKE property to FSZK in the BPK model (and thus get concurrent security both for the prover and for the verifier simultaneously), while maintaining its conceptual simplicity and its suitability to be instantiated practically.

The subtle point is: we are actually dealing with a CMIM attacker who manages to malleate, in a malicious and unpredictable way, the public-keys and knowledge-proof interactions of the verifier in one session into the statements and knowledge-proof inter-

actions in another concurrent session. To add CKE security to FSZK in the BPK model, some non-malleable tools seem to be required. Here, we show how to do so without employing such tools.

The crucial idea behind achieving our goal is to strengthen the first sub-protocol to be *statistical* WIPOK, and require the prover to first, before starting the second WIPOK sub-protocol, commit to the supposed witness to c_w by running a *statistically-binding* commitment scheme. This guarantees that if the witness committed to c_w is dependent on the secret-key used by V, there are, in fact, certain differences between the interaction distribution when V uses $SK = s_0$ and the one when V uses $SK = s_1$. We can, in turn, use such distribution differences to violate the statistical WI of the first sub-protocol, which then implies *statistical* CKE. This solution, however, loses CZK in general, since the second WI sub-protocol is run w.r.t. commitments to different values in real interactions and in the simulation. To deal with this problem we employ a stronger second sub-protocol, i.e., strong WI argument/proof-of-knowledge (strong WIPOK) [11]. Note that composing the commitment and the SWI yields a regular WI, and thus the CZK property is salvaged.

Employing SWI complicates the protocol structure, and incurs protocol inefficiency. It is, therefore, desirable to still use a regular WIPOK in the second sub-protocol, for conceptual simplicity and efficiency. To bypass the subtleties of SWI for the CZK proof, we employ a double-commitments technique. Specifically, we require the prover to produce a *double* of statistically-binding commitments, c_w and c_{sk}, before starting the second WIPOK sub-protocol of FSZK, where c_w is supposed to commit to a valid $\mathcal{NP}$-witness for $x \in L$ and c_{sk} is supposed to commit to the preimage of either y_0 or y_1. Double commitments can bypass, by hybrid arguments, the subtleties of SWI for the CZK proof. But, the provable CKE property with double commitments turns out to be much subtler. Specifically, due to the double commitments used, the value extracted can be either the value committed to c_w or that to c_{sk}. If it is ensured that the valued extracted is always the one committed to c_w (i.e., satisfying the correct knowledge-extraction property of Definition 1), we can get statistical CKE in the same way as the SWI-based solution. By the one-wayness of f, the value extracted in polynomial time cannot be the preimage of y_{1-b} (recall the secret-key is s_b). But, how about the possibility that the value extracted is just the secret-key s_b committed to c_{sk}? Consider the following adversarial strategy:

With non-negligible probability p, P^* commits s_0 (resp., s_1) to c_{sk} in a session (possibly by malleating verifier's public-key into c_{sk}); Then, *possibly by malleating the first WIPOK sub-protocol concurrent interactions*, P^* successfully finishes the second WIPOK sub-protocol of the session with s_0 (resp., s_1) as the witness, in case the verifier V uses s_0 (resp., s_1) as the secret-key; However, with the same probability p, P^* commits both a valid witness w to c_w and s_0 (resp. s_1) to c_{sk}, and then successfully finishes the second WIPOK sub-protocol with w as the witness in case V uses s_1 (resp., s_0) as secret-key. Note that, for this adversarial strategy, with non-negligible probability p the value extracted will just be the secret-key (that is also used by the extractor itself). But, we do not know how to reach contradiction under standard polynomial assumptions in this case. In particular, this adversarial strategy does not violate the statistical WI of the first WIPOK sub-protocol: with probability $2p$, the value committed to c_{sk} is s_σ for *both* $\sigma \in \{0, 1\}$, no matter which secret-key is used by the verifier.

To overcome this technical difficulty, we employ complexity leveraging in a novel way. Specifically, on the system parameter n, we assume the OWF f is hard against sub-exponential 2^{n^c}-time adversaries for some constant c, $0 < c < 1$. But, the commitment c_{sk} is generated on a relatively smaller security parameter n_{sk} such that n_{sk} and n are polynomially related (i.e., any quantity that is a polynomial of n is also another polynomial of n_{sk}) but $poly(n) \cdot 2^{n_{sk}} \ll 2^{n^c}$. This complexity leveraging ensures that, with at most negligible probability, the value extracted can be the secret-key s_b committed to c_{sk}, from which the correctness of knowledge-extraction (and then the *statistical* CKE security) is established. The reasoning is as follows: For any i, $1 \leq i \leq s(n)$, suppose with non-negligible probability p an s-concurrent malicious P^* can successfully finish the i-th session with c_{sk} committing to s_σ, $\sigma \in \{0, 1\}$, when the honest verifier (and also the extractor) uses s_σ as its secret-key; Then, by the *statistical* WI property of the first WIPOK sub-protocol, with the same probability p, P^* successfully finishes the i-th session with c_{sk} committing to s_σ, when the honest verifier uses $s_{1-\sigma}$ as the secret-key. In the later case, we can open c_{sk} to get s_σ by brute force in $poly(n) \cdot 2^{n_{sk}}$-time, which however violates the sub-exponential hardness of y_σ because $poly(n) \cdot 2^{n_{sk}} \ll 2^{n^c}$.

We stress that complexity leveraging via the sub-exponential hardness assumption on verifier's public-key is only for provable security analysis to frustrate concurrent man-in-the-middle. Both CZK simulation and CKE knowledge-extraction are still in polynomial-time. We suggest that the use of complexity leveraging for frustrating CMIM could be a useful paradigm, different from the uses of complexity leveraging in existing works for protocols in the BPK model (e.g., [4]). The complexity-leveraging based efficient and conceptually simple CZK-CKE solution can be further practically instantiated for some common number-theoretic languages.

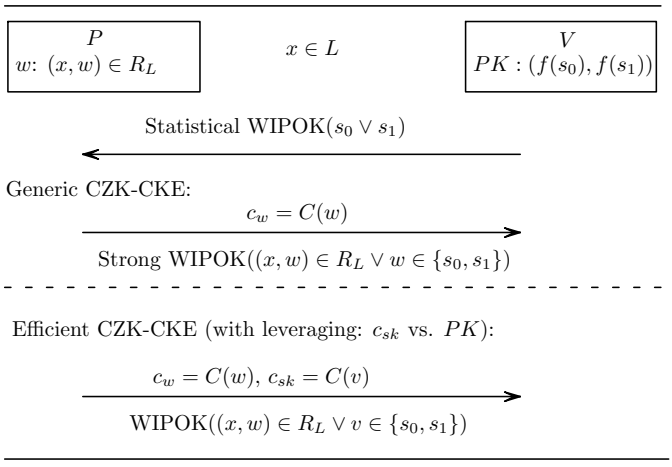

Fig. 1. Depiction of CZK-CKE from FSZK

The CZK-CKE protocols from FSZK are roughly depicted in Figure 1. We also show that all other FSZK possible component variants within the given protocol structure of Figure 1, are essentially not provably (black-box) CZK-CKE secure in the BPK model, which is, perhaps, somewhat puzzling.

Acknowledgment: Yunlei Zhao thanks Giovanni Di Crescenzo, Ivan Visconti and Frances F. Yao for helpful discussions.

References

1. M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. In *CRYPTO 1992, LNCS 740*, pages 390-420.
2. M. Bellare and O. Goldreich. On Probabilistic versus Deterministic Provers in the Definition of Proofs Of Knowledge. Cryptology ePrint Archive, Report 2006/359.
3. M. Blum. How to Prove a Theorem so No One Else can Claim It. In Proceedings of the International Congress of Mathematicians, pages 1444-1451, 1986.
4. R. Canetti, O. Goldreich, S. Goldwasser and S. Micali. Resettable Zero-Knowledge. In *ACM Symposium on Theory of Computing*, pages 235-244, 2000.
5. R. Canetti, J. Kilian, E. Petrank and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires (Almost) Logarithmically Many Rounds. In *SIAM Journal on Computing*, 32(1): 1-47, 2002.
6. R. Cramer, I. Damgard and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO 1994*, pages 174-187.
7. G. Di Crescenzo and I. Visconti. Concurrent Zero-Knowledge in the Public-Key Model. In *ICALP 2005*, pages 816-827.
8. G. Di Crescenzo and I. Visconti. On Defining Proofs of Knowledge in the Bare Public-Key Model. In *ICTCS*, 2007.
9. D. Dolev, C. Dwork and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2): 391-437, 2000.
10. U. Feige and Shamir. Zero-Knowledge Proofs of Knowledge in Two Rounds. In *CRYPTO 1989*, pages 526-544.
11. O. Goldreich. *Foundation of Cryptography-Basic Tools*, 2001.
12. O. Goldreich. *Foundations of Cryptography-Basic Applications*, 2002.
13. O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing But Their Validity or All languages in $\mathcal{NP}$ Have Zero-Knowledge Proof Systems. *JACM*, 38(1): 691-729, 1991.
14. S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems In *ACM Symposium on Theory of Computing*, pages 291-304, 1985.
15. S. Halevi and S. Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. In Crypto 1996.
16. S. Micali and L. Reyzin. Soundness in the Public-Key Model. In *CRYPTO 2001*, pages 542–565.
17. M. Naor. Bit Commitment Using Pseudorandomness. *Journal of Cryptology*, 4(2): 151-158, 1991.
18. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks. In *STOC'90*, pages 427-437.
19. R. Pass and A. Rosen. New and Improved Constructions of Non-Malleable Cryptographic Protocols. *SIAM Journal on Computing*, 38(2): 702-752 (2008).
20. A. C. Yao, M. Yung and Y. Zhao. Concurrent Knowledge Extraction in the Public-Key Model. *ECCC*, Report 2007/002. Available also from *Cryptology ePrint Archive*, Report 2010/.
21. M. Yung and Y. Zhao. Interactive Zero-Knowledge with Restricted Random Oracles. In *TCC 2006*, pages 21-40.
22. Y. Zhao, J. B. Nielsen, R. Deng and D. Feng. Generic yet Practical ZK Arguments from any Public-Coin HVZK. *ECCC*, 2005/162.
23. Y. Zhao. Concurrent/Resettable Zero-Knowledge With Concurrent Soundness in the Bare Public-Key Model and Its Applications. *Cryptology ePrint Archive*, Report 2003/265.