Trading Static for Adaptive Security in Universally Composable Zero-Knowledge

Aggelos Kiayias* and Hong-Sheng Zhou*

Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
{aggelos,hszhou}@cse.uconn.edu

Abstract. Adaptive security, while more realistic as an adversarial model, is typically much harder to achieve compared to static security in cryptographic protocol design. Universal composition (UC) provides a very attractive framework for the modular design of cryptographic protocols that captures both static and adaptive security formulations. In the UC framework, one can design protocols in hybrid worlds that allow access to idealized functionalities and then apply the universal composition theorem to obtain more concrete protocol instances. The zero-knowledge (ZK) ideal functionality is one of the most useful sub-protocols in modular cryptographic design. Given an adaptively secure protocol in the ideal ZK-hybrid-world do we always need an adaptively secure realization of the ZK functionality in order to preserve adaptive security under composition? In this work, perhaps surprisingly, we find that this is not so and in fact there are useful protocol instances that we can "trade static security for adaptive security."

We investigate the above setting, by introducing a weakened ZK ideal functionality, called the ideal leaking-zero-knowledge functionality (LZK) that leaks some information about the witness to the adversary in a certain prescribed way. We show that while LZK is interchangeable to ZK against static adversaries, ZK is more stringent when adaptive adversaries are considered. We then proceed to characterize a class of protocols in the hybrid-ZK-world that can be "transported" to the LZK-hybrid-world without forfeiting their security against adaptive adversaries. Our results demonstrate that in such settings a static protocol realization of ZK is sufficient for ensuring adaptive security for the parent hybrid protocol something that enables simplified and substantially more efficient UC realizations of such protocols.

1 Introduction

When analyzing the security of cryptographic protocols there typically exists a divide between adaptive and static security, cf. [6]. In an adaptive security setting

* Research partly supported by NSF CAREER Award CNS-0447808.

the adversary is allowed to corrupt parties dynamically and this makes simulation based proofs difficult: in particular without assuming erasures [2] the simulator would be forced to reconstruct the internal state of a corrupted machine that has been simulated. In fact, depending on the arguments used to prove the indistinguishability of simulated protocol transcripts, state reconstruction can be impossible. In contrast, in the static security setting, state reconstruction is not needed since the adversary is forced to decide a-priori on which parties are to be corrupted; this gives the leeway to the simulator to communicate to the adversary simulated transcripts that even though they substantially deviate from real protocol transcripts they are still indistinguishable from the point of view of a static adversary.

The divide between static and adaptive security in simulation based security proofs naturally impacts the complexity of attaining these levels of security for many cryptographic functionalities (both in terms of protocol efficiency as well as in terms of necessary idealized setup assumptions). In particular, for a given functionality, an adaptively secure protocol realizing it, is typically much more complicated compared to a protocol that only realizes it in the static sense. In the Universal Composition (UC) setting most interesting functionalities can be realized much more easily in the static security sense; (a notable exception is the ideal functionality of a digital signature [4, 5]). This holds true also for the Zero-Knowledge ideal functionality $\mathcal{F}_{\text{ZK}}$ that idealizes the operation of a zero-knowledge protocol [5]. Realizing $\mathcal{F}_{\text{ZK}}$ in the UC-setting is based on the notion of UC-commitment [7]. Obtaining UC-commitments in the adaptive security sense is a rather arduous task [9, 10].

The functionality $\mathcal{F}_{\text{ZK}}$ is arguably one of the most useful sub-component functionalities in the design of complex cryptographic protocols (cf. [12, 11]). The UC setting gives us the flexibility to focus on how to realize $\mathcal{F}_{\text{ZK}}$ with some protocol ρ individually; then, given such realization, the universal composition theorem [3, 5] enables us to focus on protocol design in the $\mathcal{F}_{\text{ZK}}$-hybrid world.

While the design of protocols within the $\mathcal{F}_{\text{ZK}}$-hybrid world is particularly attractive (given the power of the included ideal functionality that is supplied "for free" in the hybrid world) one cannot undervalue the substantial cost that will be incurred when $\mathcal{F}_{\text{ZK}}$ will be substituted with some protocol ρ that realizes the ideal functionality in the adaptive security sense. This brings forth the following fundamental question that is the central theme of the present work: Are there useful $\mathcal{F}_{\text{ZK}}$-like functionalities that are (1) substantially cheaper to realize than $\mathcal{F}_{\text{ZK}}$ against adaptive adversaries and (2) still sufficiently powerful to be useful as $\mathcal{F}_{\text{ZK}}$ substitutes within a certain UC modular design scenario? Or, to pose this question more specifically, is it always necessary to use an adaptively secure realization of the ZK functionality in order to preserve the adaptive security of an $\mathcal{F}_{\text{ZK}}$ hybrid protocol under composition?

Contributions. In this work we answer the question posed above. In particular we define the ideal functionality of "leaking zero-knowledge" $\mathcal{F}_{\text{LZK}}$ that has the following characteristics:

(1) The leaking zero-knowledge functionality $\mathcal{F}_{\text{LZK}}$ is based on $\mathcal{F}_{\text{ZK}}$ with the difference that it leaks to the adversary some information about the witness in a controlled way: in particular $\mathcal{F}_{\text{LZK}}$ encompasses a specialized commitment scheme (that we call R-commitment where R is the ZK-relationship and we formalize herein) and when the prover issues a "prove" command to the functionality $\mathcal{F}_{\text{LZK}}$, the functionality leaks a commitment to w to the adversary. If the prover is corrupted at any moment after the commitment has been released, the commitment is opened to the adversary.
(2) We prove that $\mathcal{F}_{\text{LZK}}$ is interchangeable with $\mathcal{F}_{\text{ZK}}$ against static adversaries. Thus in some sense, one can say, that $\mathcal{F}_{\text{LZK}}$ is a "static version" of the $\mathcal{F}_{\text{ZK}}$ zero-knowledge functionality. This also immediately implies that as long as one is interested in static security, $\mathcal{F}_{\text{LZK}}$ can be used in place of $\mathcal{F}_{\text{ZK}}$. Moreover, it hints that $\mathcal{F}_{\text{LZK}}$ may be "cheaper" to realize against adaptive adversaries when compared to $\mathcal{F}_{\text{ZK}}$. Indeed we present a simple protocol that realizes $\mathcal{F}_{\text{LZK}}$ in the $(\mathcal{F}_{\text{PRS}}, \mathcal{F}_{\text{ZKPM}})$-hybrid world against adaptive adversaries (and thus automatically also $\mathcal{F}_{\text{ZK}}$ against static adversaries); it seems difficult to obtain a protocol of similar complexity that realizes $\mathcal{F}_{\text{ZK}}$ against adaptive adversaries in the $(\mathcal{F}_{\text{PRS}}, \mathcal{F}_{\text{ZKPM}})$-hybrid world.
(3) It is possible to construct an environment that uses adaptive corruptions and separates the two functionalities $\mathcal{F}_{\text{LZK}}$ and $\mathcal{F}_{\text{ZK}}$, unless the involved ZK-relation is a trivial relationship (to be clarified further in section 3.3). Moreover, we show that $\mathcal{F}_{\text{ZK}}$ emulates $\mathcal{F}_{\text{LZK}}$ against any adversary something that is indicative of the fact that $\mathcal{F}_{\text{ZK}}$ is more powerful as a functionality.
(4) In the adaptive adversary setting, we characterize a family of protocols (using a sufficient condition cf. section 5.2) that operate in the $\mathcal{F}_{\text{ZK}}$- hybrid world and have the property that they *retain adaptive security when transported to the $\mathcal{F}_{\text{LZK}}$-hybrid world*. To put it simply, for such protocols using $\mathcal{F}_{\text{ZK}}$ is an "overkill" and it would be sufficient to consider them as protocols in the $\mathcal{F}_{\text{LZK}}$-hybrid world.

Interpreting the above in the context of the $\mathcal{F}_{\text{ZK}}$-hybrid world leads to the somewhat surprising result that there exist protocols where a certain static security realization of $\mathcal{F}_{\text{ZK}}$ (which is an adaptive realization of $\mathcal{F}_{\text{LZK}}$) is *still sufficient to achieve adaptive security* in the UC setting. In such settings we can say that we have traded static for adaptive security!

As expected the family of protocols we characterize in item (4) above excludes many functionalities that apparently require the adaptive security properties of a realization of $\mathcal{F}_{\text{ZK}}$. Still, many useful protocols fall into the class of protocols that we can trade static for adaptive security. In fact, the class, intuitively, contains all protocols that employ $\mathcal{F}_{\text{ZK}}$ for "consistency purposes" (rather than say for witness hiding purposes).

A simple example of a protocol that belongs to the class is the usage of the $\mathcal{F}_{\text{ZK}}$ functionality that is part of the adaptive commit-and-prove protocol ACP of [9]: the ACP protocol involves three different instances of the $\mathcal{F}_{\text{ZK}}$ functionality where one of them (the one employed by the verifier to ensure that his com-

mitment key is valid) can in fact be substituted by $\mathcal{F}_{\text{LZK}}$ without affecting the protocol's adaptive security (cf. $\mathcal{F}_{\text{ZK}}^{\text{T}}$ in Figure 10, page 57 in [9]; we note that the $\mathcal{F}_{\text{ZK}}^{\text{T}}$ functionality can also be simulated by an $\mathcal{F}_{\text{CRS}}$ box — a fact remarked in [9]). A more complex example of usage of $\mathcal{F}_{\text{ZK}}$ within a protocol that can be substituted by $\mathcal{F}_{\text{LZK}}$ is exhibited in [13] for the design of UC blind signatures: in this type of signatures it turns out that the signer requires only $\mathcal{F}_{\text{LZK}}$ (as opposed to $\mathcal{F}_{\text{ZK}}$ that is required for the user side).

Other related work. Relaxations of ideal functionalities were also seen in the context of the "monitored functionalities" of [15]; note that the goal there was to relax w.r.t. correctness rather than security as we do here. A relaxation w.r.t. security for the key-exchange ideal functionality was performed in [8]; in their setting the ideal-functionality leaks a function of the exchanged key (by including the so called "non-information" oracle).

Notations. $a \xleftarrow{\mathtt{r}} \mathtt{RND}$ denotes randomly selecting a in its domain; $\mathsf{negl}()$ denotes negligible function.

2 Preliminaries

The Universal Composability Framework [5]. Defining security in the universal composibility framework involves the following steps: we first specify an ideal functionality $\mathcal{F}$, which describes the desired behavior of the protocol by using a trusted party; this functionality $\mathcal{F}$ communicates also with an ideal world adversary. Then, we prove that a particular protocol π operating in the real world securely realizes this ideal functionality. Here, securely realizing means that for any adversary $\mathcal{A}$ in the real world, there exists a simulator $\mathcal{S}$ in the ideal world, and no environment $\mathcal{Z}$ can distinguish its interaction with the real protocol π and $\mathcal{A}$, or with the functionality $\mathcal{F}$ and $\mathcal{S}$. Once this is established, we can take advantage of the UC composition theorem and "plug in" the protocol π as a sub-routine in any arbitrary environment in place of the functionality $\mathcal{F}$. For a complete definition of UC framework please refer to [5].

Functionality $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}}$. Next we describe functionality $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}}$, which is similar to the KS, KR, PRS functionalities employed respectively in [5, 1, 14]. Here we only consider the case for two parties, P and V (and thus we modify it accordingly).

Functionality $\mathcal{F}_{\text{ZK}}^{R}$. A zero-knowledge proof is a two-party protocol parameterized by a binary relation R; the two parties called the prover and the verifier share a common input, the statement x. The prover has an additional input, the witness w. If $(x, w) \in R$, the verifier accepts; if not, the verifier will reject. Furthermore the verifier learn nothing from the protocol with the prover except of whether the prover knows the witness w s.t. $(x, w) \in R$ or not. The functionality in figure 2 is taken from [5] which captures properly the security properties of a zero-knowledge proof.

Functionality $\mathcal{F}_{\text{PRS}}^{\texttt{gen}}$

$\mathcal{F}_{\text{PRS}}^{\texttt{gen}}$ proceeds as follows, running with two parties P, V, and an adversary $\mathcal{S}$, and parameterized with a function $\texttt{gen}$.

Register: Upon receiving $(\texttt{RegisterPRS}, sid)$ from party P, verify that $sid = (P, V, sid')$ for some sid'. If not, then ignore the input. Else, randomly select dk and produce $ek \leftarrow \texttt{gen}(dk)$, record (P, ek) in *history*, and send $(\texttt{ReturnPRS}, sid, ek)$ to $\mathcal{S}$. If $(\texttt{ReturnPRS}, sid, ek, \mathsf{ok})$ is received from $\mathcal{S}$, then return $(\texttt{ReturnPRS}, sid, ek)$ to P.
Upon receiving $(\texttt{RegisterPRS}, sid, dk, P)$ from a corrupted party P, produce $ek \leftarrow \texttt{gen}(dk)$. Record (P, ek) in *history*.
Retrieve: Upon receiving $(\texttt{RetrievePRS}, sid, P)$ from some party V, send $(\texttt{RetrievePRS}, sid, P, V)$ to $\mathcal{S}$ and obtain a value ek in return. If (P, ek) has been recorded in *history* then return $(\texttt{ReturnPRS}, sid, ek, P)$ to V. Else, return $(\texttt{ReturnPRS}, sid, \perp, P)$ to V.
Corrupt: Upon receiving $(\texttt{CorruptPRS}, sid, P)$ from $\mathcal{S}$, record the fact that party P has been corrupted.

Fig. 1. Private reference string functionality $\mathcal{F}_{\text{PRS}}^{\texttt{gen}}$ for two parties.

3 The Leaking Zero-Knowledge Functionality

3.1 *R*-Commitment

An R-commitment scheme is a special non-interactive commitment scheme that is bound to a given relation R. It is an extractable commitment where the hiding property is only required to hold with respect to the witnesses of the relation R. In particular, if the witness is computationally hidden by the statement, then an R-commitment can be at most computationally hiding. Formally, an R-commitment scheme $\mathcal{E}$ is a tuple $\langle \texttt{gen}_{\mathcal{E}}, \texttt{com}_{\mathcal{E}}, \texttt{ver}_{\mathcal{E}}, \texttt{dec}_{\mathcal{E}} \rangle$. The key generation algorithm $\texttt{gen}_{\mathcal{E}}$ produces a public parameter ek based on a randomly selected $dk \in \mathcal{K}$. The procedures $\texttt{com}_{\mathcal{E}}, \texttt{ver}_{\mathcal{E}}$ correspond to the commitment algorithm and the testing algorithm for the decommit information for a given commitment; they satisfy the correctness property $\texttt{ver}_{\mathcal{E}}(x, ek, \texttt{com}_{\mathcal{E}}(ek, x, w, \gamma), w, \gamma) = 1$ for any $ek \leftarrow \texttt{gen}_{\mathcal{E}}(dk)$ with $dk \in \mathcal{K}$. The procedure $\texttt{dec}_{\mathcal{E}}$ always extracts the witness given the trapdoor key dk; in particular, we require $\forall E\ \exists w, \gamma$ such that $E = \texttt{com}_{\mathcal{E}}(ek, x, w, \gamma)$ and $\texttt{dec}_{\mathcal{E}}(x, ek, E, dk) = w$. Note that we may generalize these requirements to allow for partial correctness and extractability but this would not have any significant impact on our results.

We say that $\mathcal{E}$ is an R-commitment for a given relation R if additionally to the above, it satisfies the ***R-hiding*** property:

Definition 1 (R-hiding). *We say a commitment $\mathcal{E}$ is R-hiding, if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathsf{Adv}_{\text{hiding}}^{R,\mathcal{E}}(\lambda) \stackrel{\mathsf{def}}{=} |2\mathsf{Prob}[\mathbf{Exp}_{\text{hiding}}^{R,\mathcal{E}}(\lambda)] - 1| = \mathsf{negl}(\lambda)$, where the experiment $\mathbf{Exp}_{\text{hiding}}^{R,\mathcal{E}}(\lambda)$ is defined below.*

Additionally, we say that $\mathcal{E}$ is ***R-unequivocal*** for some $\texttt{sample}_R$ if it satisfies:

Functionality $\mathcal{F}_{\text{ZK}}^{R}$

$\mathcal{F}_{\text{ZK}}^{R}$ proceeds as follows, running with a prover P, a verifier V and an adversary $\mathcal{S}$, and parameterized with a binary relation R.

Prove: Upon receiving $(\texttt{ProveZK}, sid, x, w)$ from party P, verify that $sid = (P, V, sid')$ for some sid'. If not, then ignore the input. Else, forward $(\texttt{ProveZK}, sid, x)$ to $\mathcal{S}$.
Upon receiving $(\texttt{ProveZK}, sid, \texttt{ok})$ from the adversary $\mathcal{S}$, if $(x, w) \in R$ then record $\langle x, w \rangle$ into *history* and output $(\texttt{VerifiedZK}, sid, x)$ to party V, else do nothing. From now on, ignore future $(\texttt{ProveZK}, sid, ...)$ input.
Corrupt: Upon receiving $(\texttt{CorruptProverZK}, sid)$ from $\mathcal{S}$, return $\mathcal{S}$ $(\texttt{CorruptedProverZK}, sid, history)$. Record the fact that party P has been corrupted. After the corruption has occurred, upon receiving $(\texttt{PatchZK}, sid, x', w')$, if $(x', w') \in R$ and no output $(\texttt{VerifiedZK}, sid, ...)$ was returned to party V yet, then output $(\texttt{VerifiedZK}, sid, x')$ to party V.

Fig. 2. Zero-knowledge functionality $\mathcal{F}_{\text{ZK}}^{R}$.

Definition 2 (R-unequivocal). *We say a commitment $\mathcal{E}$ is R-unequivocal for some PPT $\texttt{sample}_R$ that returns (x, w) in R, if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathsf{Adv}_{\text{unequivocal}}^{R,\mathcal{E}}(\lambda) \stackrel{\text{def}}{=} \mathsf{Prob}[\mathbf{Exp}_{\text{unequivocal}}^{R,\mathcal{E}}(\lambda) = 1] = \mathsf{negl}(\lambda)$, where the experiment $\mathbf{Exp}_{\text{unequivocal}}^{R,\mathcal{E}}(\lambda)$ is defined below.*

$\mathbf{Exp}_{\text{unequivocal}}^{R,\mathcal{E}}(\lambda)$
$(x, w) \leftarrow \texttt{sample}_R(1^\lambda)$;
$(ek, \widehat{E}, state) \leftarrow \mathcal{A}_1(x)$;
$\gamma \leftarrow \mathcal{A}_2(state, w)$;
if $\texttt{ver}_{\mathcal{E}}(x, ek, \widehat{E}, w, \gamma) = 1$
then output 1
else output 0.

$\mathbf{Exp}_{\text{hiding}}^{R,\mathcal{E}}(\lambda)$
$(x, w) \leftarrow \mathcal{A}_1(1^\lambda)$; if $\texttt{verify}_R(x, w) \neq 1$ then abort;
$dk \xleftarrow{r} \mathcal{K}$; $ek \leftarrow \texttt{gen}_{\mathcal{E}}(dk)$; $b \xleftarrow{r} \{0, 1\}$;
if $b = 0$ then $E \leftarrow \mathsf{com}_{\mathcal{E}}(x, ek, \widehat{w}, \widehat{\gamma})$; $\widehat{w}, \widehat{\gamma} \xleftarrow{r} \texttt{RND}$;
else $E \leftarrow \mathsf{com}_{\mathcal{E}}(x, ek, w, \gamma)$; $\gamma \xleftarrow{r} \texttt{RND}$;
$b^* \leftarrow \mathcal{A}_2(x, w, E)$;
if $b^* = b$ then return 1 else return 0.

3.2 Functionality $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$

In this subsection we introduce our new ZK functionality, called the leaking zero-knowledge functionality, $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$, in figure 3; it is parameterized by a relation R as well as an R-commitment $\mathcal{E}$. The design of $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ is based on $\mathcal{F}_{\text{ZK}}^{R}$. Recall that in the "prove" stage of $\mathcal{F}_{\text{ZK}}^{R}$, upon receiving the statement-witness pair $\langle x, w \rangle$, $\mathcal{F}_{\text{ZK}}^{R}$ is supposed to communicate the statement x to the adversary (but not the witness). In our case, during the "prove" stage of $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$, we allow $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ to leak more information about the witness that includes the parameter ek and a commitment E of the witness w, that is based on the parameter ek.

Note that we still anticipate $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ to capture some level of the zero-knowledge property, and a computationally bounded adversary still would not obtain any useful information about the witness w from reading the extra information ek and E that is leaked together with the statement (this is based on the R-hiding

property of the commitment as described above). Still, the "quality" of zero-knowledge offered by $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ is substantially impaired compared to $\mathcal{F}_{\text{ZK}}^{R}$. Note that whenever the prover is corrupted the commitment that was issued for proof's witness will be opened (i.e., the adversary will not only receive the witness but also the decommitment information of the released commitment).

Functionality $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$

$\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ proceeds as follows, running with a prover P, a verifier V and an adversary $\mathcal{S}$, and parameterized with a binary relation R and an R-commitment $\mathcal{E}$; it incorporates $\mathcal{F}_{\text{PRS}}^{\text{gen}_{\mathcal{E}}}$ and furthermore it has the additional functions as below.

Prove: Upon receiving $(\texttt{ProveLZK}, sid, x, w)$ from party P, verify that $sid = (P, V, sid')$ for some sid'. If not, then ignore the input. Else, randomly select γ and compute $E = \texttt{com}_{\mathcal{E}}(x, ek, w, \gamma)$, and forward $(\texttt{ProveLZK}, sid, x, ek, E)$ to $\mathcal{S}$. Upon receiving $(\texttt{ProveLZK}, sid, \texttt{ok})$ from the adversary $\mathcal{S}$, if $(x, w) \in R$ then record $\langle x, E, w, \gamma \rangle$ in *history* and output $(\texttt{VerifiedLZK}, sid, x)$ to party V, else do nothing. From now on, ignore future $(\texttt{ProveLZK}, sid, ...)$ input.

Corrupt: Upon receiving $(\texttt{CorruptProverLZK}, sid)$ from $\mathcal{S}$, return $\mathcal{S}$ $(\texttt{CorruptedProverLZK}, sid, history, ek)$. Record this fact that party P has been corrupted. After the corruption has occurred, upon receiving $(\texttt{PatchLZK}, sid, x', w')$, if $(x', w') \in R$ and no output $(\texttt{VerifiedLZK}, sid, ...)$ was returned to party V yet, then output $(\texttt{VerifiedLZK}, sid, x')$ to party V.

Fig. 3. Leaking zero-knowledge functionality $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$.

3.3 Relation between $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ and $\mathcal{F}_{\text{ZK}}^{R}$

In this subsection we explore the essential relation between $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ and $\mathcal{F}_{\text{ZK}}^{R}$. First, we show that the functionality $\mathcal{F}_{\text{ZK}}^{R}$ can UC-emulate $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$; on the other hand, the other direction can only hold against static adversaries. Please refer to figure 4 below.

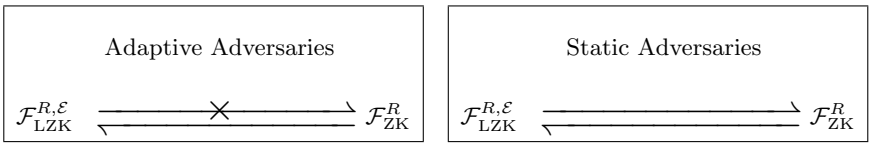

Fig. 4. Relation between $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ and $\mathcal{F}_{\text{ZK}}^{R}$. $\mathcal{F}_1 \rightharpoonup \mathcal{F}_2$ stands for "$\mathcal{F}_1$ UC-emulates $\mathcal{F}_2$."

To establish the emulation result we show that a dummy protocol in the $\mathcal{F}_{\text{ZK}}^{R}$-hybrid world realizes $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$. It is easy to see that a simulator interacting

with $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ can perfectly simulate transcripts to an environment that operates with dummy parties in the $\mathcal{F}_{\text{ZK}}^{R}$-hybrid world by simply suppressing the extra information provided by $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$.

Theorem 1. *Let $\mathcal{F}_{\text{ZK}}^{R}$ be the ideal ZK functionality, and $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ be the leaking version of $\mathcal{F}_{\text{ZK}}^{R}$. Let ρ_{d} be a dummy ZK protocol. Then for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{S}$ such that for any adaptive environment machine $\mathcal{Z}$ we have:* $\text{EXEC}_{\rho_{\text{d}},\mathcal{A},\mathcal{Z}}^{\mathcal{F}_{\text{ZK}}^{R}} = \text{EXEC}_{\rho_{\text{d}},\mathcal{S},\mathcal{Z}}^{\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}}$.

We then investigate the other direction of theorem 1; we prove that a dummy protocol in $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$-hybrid world can *statically* realize functionality $\mathcal{F}_{\text{ZK}}^{R}$ as described in theorem 2. The simulation is not perfect as it relies on the hiding properties of the R-commitment $\mathcal{E}$.

Theorem 2. *Let $\mathcal{F}_{\text{ZK}}^{R}$ be the ideal ZK functionality, $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ the leaking version of $\mathcal{F}_{\text{ZK}}^{R}$, and ρ_{d} a dummy ZK protocol. If $\mathcal{E}$ is an R-hiding commitment, then for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{S}$ such that for any static environment machine $\mathcal{Z}$ we have:* $|\text{EXEC}_{\rho_{\text{d}},\mathcal{A},\mathcal{Z}}^{\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}} - \text{EXEC}_{\rho_{\text{d}},\mathcal{S},\mathcal{Z}}^{\mathcal{F}_{\text{ZK}}^{R}}| \leq \mathsf{Adv}_{\text{hiding}}^{R,\mathcal{E}}(\lambda)$.

Regarding adaptive adversaries, we *cannot* extend the result of the previous theorem. We establish this in theorem 3. The basic reason is that in the simulation of theorem 2 the simulator for $\mathcal{F}_{\text{ZK}}^{R}$ has to simulate the extra information $\langle ek, E\rangle$. The simulator can easily simulate ek by just using the key-generator $\mathtt{gen}_{\mathcal{E}}$. However the simulator gets in trouble when it needs to simulate E for an adaptive environment $\mathcal{Z}$. Note that the simulator does not know the witness w, which is "blocked" inside the functionality $\mathcal{F}_{\text{ZK}}^{R}$. The simulator may produce E based on a fake witness or simulate E in some other way; but when the adaptive $\mathcal{Z}$ corrupts the prover after the simulated commitment has been released, the simulator must explain E to $\mathcal{Z}$ for the real witness (that is now released from the ideal functionality). This would require that the underlying R-commitment scheme to be "equivocal" (which it is not).

Given the inflexibility of the R-commitment the simulator may still succeed if the relation R is somewhat trivial, and an adversary can obtain the correct witness by observing the statement x. In such case, the simulator now has chance to develop a successful simulation even if the environment is adaptive. The R-unequivocal property was designed appropriately so that it captures all these scenarios; based on this, we obtain the following theorem that demonstrates that the functionality $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ is weaker as a security notion compared to $\mathcal{F}_{\text{ZK}}^{R}$.

Theorem 3. *Let $\mathcal{F}_{\text{ZK}}^{R}$ be the ideal ZK functionality, $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ the leaking version of $\mathcal{F}_{\text{ZK}}^{R}$, and ρ_{d} a dummy ZK protocol. If $\mathcal{E}$ is R-unequivocal for some $\mathtt{sample}_R$, then there exists an adversary $\mathcal{A}$ and an adaptive environment machine $\mathcal{Z}$ such that for any adversary $\mathcal{S}$, we have:* $|\text{EXEC}_{\rho_{\text{d}},\mathcal{A},\mathcal{Z}}^{\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}} - \text{EXEC}_{\rho_{\text{d}},\mathcal{S},\mathcal{Z}}^{\mathcal{F}_{\text{ZK}}^{R}}| \geq 1 - \mathsf{Adv}_{\text{unequivocal}}^{R,\mathcal{E}}(\lambda)$.

4 Implementation of $\mathcal{F}_{\mathbf{LZK}}^{R,\mathcal{E}}$ in the $(\mathcal{F}_{\mathbf{PRS}}^{\mathsf{gen}_\mathcal{E}}, \mathcal{F}_{\mathbf{ZKPM}}^{R'})$-Hybrid World

In this section we show that $\mathcal{F}_{\mathrm{LZK}}^{R,\mathcal{E}}$ can be realized with the protocol π_{LZK} in the $(\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}, \mathcal{F}_{\mathrm{ZKPM}}^{R'})$-hybrid world presented in figure 5. Note that based on Nielsen's result (refer to theorem 5.1 in page 180 in Nielsen's PhD thesis [14]), $\mathcal{F}_{\mathrm{ZKPM}}$ can be very efficiently implemented in the $\mathcal{F}_{\mathrm{PRS}}$-hybrid world as $\mathcal{F}_{\mathrm{ZKPM}}$ does not require witness extraction. So π_{LZK} can be implemented in the $\mathcal{F}_{\mathrm{PRS}}$-hybrid world without requiring UC commitments. Next we prove that the protocol π_{LZK} from figure 5 realizes $\mathcal{F}_{\mathrm{LZK}}^{R,\mathcal{E}}$.

Protocol π_{LZK} in the $(\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}, \mathcal{F}_{\mathrm{ZKPM}}^{R'})$-Hybrid World

On input $(\texttt{ProveLZK}, sid, x, w)$ from $\mathcal{Z}$, party P sends $(\texttt{RegisterPRS}, sid)$ to $\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}$. Whenever Party P receives $(\texttt{ReturnPRS}, sid, ek)$ from $\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}$, it randomly selects γ and computes $E = \mathsf{com}_\mathcal{E}(x, ek, w, \gamma)$, and sends $\mathcal{F}_{\mathrm{ZKPM}}^{R'}$ the message $(\texttt{ProveZKPM}, sid, (x, ek, E), (w, \gamma))$.
Whenever Party V receives $(\texttt{VerifiedZKPM}, sid, (x, ek, E))$ from $\mathcal{F}_{\mathrm{ZKPM}}^{R'}$, it sends $(\texttt{GetPRS}, sid)$ to $\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}$ and get ek from $\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}$; if the ek is same as the one from $\mathcal{F}_{\mathrm{ZKPM}}^{R'}$ then returns $(\texttt{VerifiedLZK}, sid, x)$ to $\mathcal{Z}$.

Fig. 5. A protocol realizing $\mathcal{F}_{\mathrm{LZK}}^{R,\mathcal{E}}$ in the $(\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}, \mathcal{F}_{\mathrm{ZKPM}}^{R'})$-hybrid world. Here $\mathcal{E}$ is an R-commitment, and relation R' is based on relation R and key generator gen, i.e. $R' = \{(x, ek, E), (w, \gamma) \mid (x, w) \in R \ \wedge \ E = \mathsf{com}_\mathcal{E}(x, ek, w, \gamma)\}$.

Theorem 4. *Consider protocol π_{LZK} in the $(\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}, \mathcal{F}_{\mathrm{ZKPM}}^{R'})$-hybrid world in figure 5, where $\mathcal{E}$ is an R-commitment. Let π_{d} be a dummy ZK protocol. Then for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{S}$ such that for any adaptive environment machine $\mathcal{Z}$ we have:* $\mathrm{EXEC}^{\mathcal{F}_{\mathrm{PRS}}^{\mathsf{gen}_\mathcal{E}}, \mathcal{F}_{\mathrm{ZKPM}}^{R'}}_{\pi_{\mathrm{LZK}}, \mathcal{A}, \mathcal{Z}} = \mathrm{EXEC}^{\mathcal{F}_{\mathrm{LZK}}^{R,\mathcal{E}}}_{\pi_{\mathrm{d}}, \mathcal{S}, \mathcal{Z}}$.

Based on theorem 2 and theorem 4, we obtain immediately that the protocol of figure 5 statically realizes $\mathcal{F}_{\mathrm{ZK}}^{R}$ (with an $\mathsf{Adv}_{\mathrm{hiding}}^{R,\mathcal{E}}(\lambda)$ distance).

In general, we can design a protocol π_{ZK} to realize $\mathcal{F}_{\mathrm{ZK}}$ in the $(\mathcal{F}_{\mathrm{COM}}, \mathcal{F}_{\mathrm{ZKPM}})$-hybrid world by committing the witness and then using ZKPM to bind the commitment and the ZK statement as in the figure 5 where E is computed based on $\mathcal{F}_{\mathrm{COM}}$. Note that in π_{LZK} we compute E based on the R-commitment, but in π_{ZK} we need the commitment to be both extractable and equivocal: in the case that the prover is corrupted, $\mathcal{F}_{\mathrm{ZK}}$ only supplies the witness and the simulator needs to figure out the random coins involved; on the contrary $\mathcal{F}_{\mathrm{LZK}}$ supplies all witness and coins for E. Combining equivocality and extractability seems that it requires more work (more rounds or more communication), cf. [9, 10].

5 Using $\mathcal{F}_{\text{LZK}}$ in place of $\mathcal{F}_{\text{ZK}}$

5.1 A Protocol Transformation

We describe a useful transformation which allows a protocol π in the $\mathcal{F}_{\text{ZK}}^{R}$-hybrid world to be modified into a slightly different protocol $\widetilde{\pi}$ based on an R-commitment $\mathcal{E}$. The protocol $\widetilde{\pi}$ operates in the $(\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}, \mathcal{F}_{\text{ZK}}^{R'})$-hybrid world for a relation R' defined as follows: $R' = \{(x, ek, E), (w, \gamma) \,|\, (x, w) \in R \wedge E = \mathsf{com}_{\mathcal{E}}(x, ek,\ w, \gamma)\}$, where ek is obtained from $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}$, γ is randomly selected. In section 5.2 we will use such transformation to explore the application of the functionality $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$.

Transformation from π into $\widetilde{\pi}$

Each time in protocol π, party P_{π} sends $(\mathtt{ProveZK}, sid, x, w)$ to $\mathcal{F}_{\text{ZK}}^{R}$, in protocol $\widetilde{\pi}$ party $P_{\widetilde{\pi}}$ sends $(\mathtt{RegisterPRS}, sid)$ to $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}$; when it receives $(\mathtt{ReturnPRS}, sid, ek)$ from $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}$, party $P_{\widetilde{\pi}}$ randomly selects γ, it computes $E = \mathsf{com}_{\mathcal{E}}(x, ek, w, \gamma)$ and sends $(\mathtt{ProveZK}, sid, (x, ek, E), (w, \gamma))$ to $\mathcal{F}_{\text{ZK}}^{R'}$.

Each time in protocol $\widetilde{\pi}$, party $V_{\widetilde{\pi}}$ receives $(\mathtt{VerifiedZK}, sid, (x, ek, E))$ from $\mathcal{F}_{\text{ZK}}^{R'}$, it sends $(\mathtt{RetrievePRS}, sid, P_{\pi})$ to $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}$, and obtains ek from $\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}$; if ek is same as the one from $\mathcal{F}_{\text{ZK}}^{R'}$, then party $V_{\widetilde{\pi}}$ sends $(\mathtt{Verified}, sid, x)$ to $\mathcal{Z}$.

Fig. 6. A transformation from π in the $\mathcal{F}_{\text{ZK}}^{R}$-hybrid world into $\widetilde{\pi}$ in the $(\mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}}, \mathcal{F}_{\text{ZK}}^{R'})$-hybrid world.

Note that functionally the protocols π and $\widetilde{\pi}$ are identical; nevertheless, the protocol $\widetilde{\pi}$ is possibly exposing some more information to the adversary as compared to π with respect to the witnesses that are employed within the $\mathcal{F}_{\text{ZK}}$ version. If an adversary can see little difference between $\widetilde{\pi}$ and π then we can use $\mathcal{F}_{\text{LZK}}$ in place of $\mathcal{F}_{\text{ZK}}$. We elaborate on this in the next subsection.

5.2 A Sufficient Condition

The goal of this section is to characterize the protocols for which we can substitute an $\mathcal{F}_{\text{ZK}}^{R}$ implementation with a (potentially cheaper) $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$ implementation in the setting of adaptive adversaries. The protocol transformation of the previous subsection serves as a "bridge" between the protocol in the $\mathcal{F}_{\text{ZK}}$-hybrid world and the protocol in the $\mathcal{F}_{\text{LZK}}$-hybrid world. We show in theorem 5 that if π realizes some $\mathcal{F}$ and the transformed protocol $\widetilde{\pi}$ maintains this functionality, this implies that the original protocol π can be transported into the $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$-hybrid world without any impact.

Theorem 5. *(Sufficient Condition) Let π be a protocol in the $\mathcal{F}_{\text{ZK}}^{R}$-hybrid-world, $\widetilde{\pi}$ the transformation of π as described in section 5.1. If $\widetilde{\pi}$ in the $(\mathcal{F}_{\text{ZK}}^{R'}, \mathcal{F}_{\text{PRS}}^{\mathtt{gen}_{\mathcal{E}}})$-hybrid world realizes functionality $\mathcal{F}$, then π in the $\mathcal{F}_{\text{LZK}}^{R,\mathcal{E}}$-hybrid world also realizes $\mathcal{F}$.*

The theorem is illustrated in figure 7. The sufficient condition for transporting a protocol π from the $\mathcal{F}_{\mathrm{ZK}}^{R}$ hybrid world into the $\mathcal{F}_{\mathrm{LZK}}$-hybrid world is marked with "?".

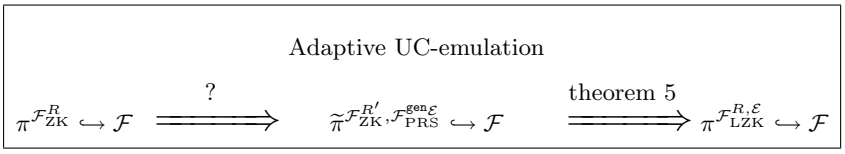

Fig. 7. Trading $\mathcal{F}_{\mathrm{LZK}}$ for $\mathcal{F}_{\mathrm{ZK}}$. Note that $\pi^{\mathcal{F}_1} \hookrightarrow \mathcal{F}_2$ stands for "π realizes $\mathcal{F}_2$ in the $\mathcal{F}_1$-hybrid world."

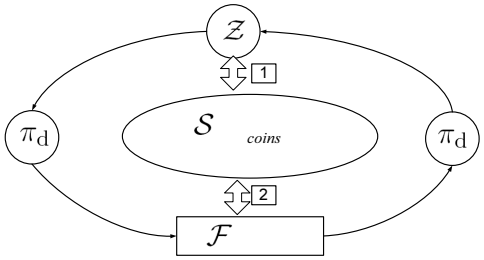

Fig. 8. In constructing $\widetilde{\mathcal{S}}$ the witness used for $\mathcal{F}_{\mathrm{ZK}}^{R}$ would be necessary for a simulation against adaptive adversaries; $\widetilde{\mathcal{S}}$ may recover such witness if it appears in the communication lines $\boxed{1}$, $\boxed{2}$ or can be inferred from the coins of $\mathcal{S}$.

In the remaining of the section we investigate the setting where the sufficient condition can be satisfied. Assume a protocol π in the $\mathcal{F}_{\mathrm{ZK}}^{R}$-hybrid-world that realizes $\mathcal{F}$. This means there exists a simulator $\mathcal{S}$ that can simulate π-protocol-transcripts for any adaptive $\mathcal{Z}$. In particular $\mathcal{S}$ simulates $\mathcal{F}_{\mathrm{ZK}}^{R}$ to produce the statement x and also the direct transcripts between the π parties.

In order to show that the transformed protocol $\widetilde{\pi}$ in the $(\mathcal{F}_{\mathrm{ZK}}^{R'}, \mathcal{F}_{\mathrm{PRS}}^{\mathtt{gen}_{\mathcal{E}}})$-hybrid world also realizes $\mathcal{F}$, we need to construct a simulator $\widetilde{\mathcal{S}}$ for the adaptive environment. We may build $\widetilde{\mathcal{S}}$ based on $\mathcal{S}$ which is given above by the assumption that $\pi^{\mathcal{F}_{\mathrm{ZK}}^{R}}$ realizes $\mathcal{F}$; the statement x and the direct transcripts between the π parties can be simulated verbatim from $\mathcal{S}$. Still $\widetilde{\mathcal{S}}$ needs to simulate the extra $\langle ek, E\rangle$ information since $\mathcal{S}$ does not supply this. Recall that our environment $\mathcal{Z}$ may involve adaptive corruptions. So $\widetilde{\mathcal{S}}$ may not be able to produce the extra $\langle ek, E\rangle$ based on a "fake" witness $\widetilde{w}$ (because when the prover is corrupted and a real witness w is supplied, $\widetilde{\mathcal{S}}$ cannot explain E to the real w, cf. theorem 3). Excluding the case of a non R-unequivocal commitment (which is rather trivial), it follows that the only way for the proof to work would be if there are circumstances for which $\widetilde{\mathcal{S}}$ is capable of inferring the witness from either the coins used by $\mathcal{S}$ and or the "communication lines" of $\mathcal{S}$ with $\mathcal{Z}$ or $\mathcal{F}$ as shown in $\boxed{1}$ or/and

2 in figure 8. For example consider $\mathcal{F}$ to be a functionality extending $\mathcal{F}_{\text{SIG}}$ [4, 5] where the signer wishes to prove knowledge and correct application of his secret key to other parties (e.g., his signing key is involved in some more complex computation for meeting a certain goal of $\mathcal{F}$). This is the case for example for the signer side in the UC blind signatures of [13]; in this protocol, the signer would require only $\mathcal{F}_{\text{LZK}}$ (as opposed to $\mathcal{F}_{\text{ZK}}$) as the key is known to the simulator.

Acknowledgements. We thank Jesper Nielsen for his kind clarifications on [1, 14]. We also thank the anonymous referees for their constructive comments.

References

1. B. Barak, R. Canetti, J. B. Nielsen, and R. Pass. Universally composable protocols with relaxed set-up assumptions. In *FOCS 2004*, pages 186–195, 2004.
2. D. Beaver and S. Haber. Cryptographic protocols provably secure against dynamic adversaries. In *EUROCRYPT 1992*, pages 307–323, 1992.
3. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS 2001*, pages 136–145.
4. R. Canetti. Universally composable signature, certification, and authentication. In *CSFW 2004*, pages 219–235, 2004. `http://eprint.iacr.org/2003/239/`.
5. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Cryptology ePrint Archive: Report 2000/067*, December 2005. Latest version at `http://eprint.iacr.org/2000/067/`.
6. R. Canetti, I. Damgård, S. Dziembowski, Y. Ishai, and T. Malkin. On adaptive vs. non-adaptive security of multiparty protocols. In *EUROCRYPT 2001*, pages 262–279, 2001.
7. R. Canetti and M. Fischlin. Universally composable commitments. In *CRYPTO 2001*, pages 19–40.
8. R. Canetti and H. Krawczyk. Universally composable notions of key exchange and secure channels. In *EUROCRYPT 2002*, pages 337–351, 2002.
9. R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *STOC 2002*, pages 494–503. `http://www.cs.biu.ac.il/~lindell/PAPERS/uc-comp.ps`.
10. I. Damgård and J. B. Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO 2002*, pages 581–596. `http://www.brics.dk/RS/01/41/BRICS-RS-01-41.pdf`.
11. O. Goldreich. *Foundations of Cryptography- Basic Tools*. Cambridge University Press, 2001.
12. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC 1987*, pages 218–229, 1987.
13. A. Kiayias and H.-S. Zhou. Equivocal blind signatures and adaptive UC-security. In *Cryptology ePrint Archive: Report 2007/132*, 2007.
14. J. B. Nielsen. On protocol security in the cryptographic model. *Dissertation Series DS-03-8, BRICS*, 2003. `http://www.brics.dk/DS/03/8/BRICS-DS-03-8.pdf`.
15. M. Prabhakaran and A. Sahai. Relaxing environmental security: Monitored functionalities and client-server computation. In *TCC 2005*, pages 104–127, 2005.