Succinct Malleable NIZKs and an Application to Compact Shuffles

Melissa Chase[1], Markulf Kohlweiss[2], Anna Lysyanskaya[3],
and Sarah Meiklejohn[4]

[1] Microsoft Research Redmond
melissac@microsoft.com
[2] Microsoft Research Cambridge
markulf@microsoft.com
[3] Brown University
anna@cs.brown.edu
[4] UC San Diego
smeiklej@cs.ucsd.edu

Abstract. Depending on the application, malleability in cryptography can be viewed as either a flaw or — especially if sufficiently understood and restricted — a feature. In this vein, Chase, Kohlweiss, Lysyanskaya, and Meiklejohn recently defined malleable zero-knowledge proofs, and showed how to *control* the set of allowable transformations on proofs. As an application, they construct the first *compact* verifiable shuffle, in which one such controlled-malleable proof suffices to prove the correctness of an entire multi-step shuffle.

Despite these initial steps, a number of natural problems remained: (1) their construction of controlled-malleable proofs relies on the inherent malleability of Groth-Sahai proofs and is thus not based on generic primitives; (2) the classes of allowable transformations they can support are somewhat restrictive.

In this paper, we address these issues by providing a generic construction of controlled-malleable proofs using succinct non-interactive arguments of knowledge, or SNARGs for short. Our construction can support very general classes of transformations, as we no longer rely on the transformations that Groth-Sahai proofs can support.

1 Introduction

Recently, malleability is increasingly being viewed more as a feature than as a bug [27,28,18,1,13,16,6]. In this vein, we (called CKLM in the sequel to disambiguate between our current and prior work) [7] introduced controlled-malleable non-interactive zero-knowledge proof systems (cm-NIZKs for short). At a high level, a cm-NIZK allows one, given a proof π for an instance $x \in L$, to compute a proof π' for the related instance $T(x) \in L$ for transformations T under which the language is closed. This malleability property can be additionally *controlled*, meaning there is some specified class of allowable transformations $\mathcal{T}$ such that, given the proof π for $x \in L$, a new proof π' for $T(x) \in L$ may be obtained only

A. Sahai (Ed.): TCC 2013, LNCS 7785, pp. 100–119, 2013.

for $T \in \mathcal{T}$. The notion of a cm-NIZK is non-trivial when the proof system also needs to be concise or *derivation-private*; i.e., in addition to π' being the same size as π, it should be impossible to tell whether π' was obtained using a witness or by mauling a proof for a previous statement.

The notion of a derivation-private cm-NIZK is well motivated: as one application, CKLM showed that it allows for the modular design of schemes that satisfy randomizable and homomorphic chosen-ciphertext security. Another application they presented is a *compactly verifiable shuffle* for an election, wherein a set of encrypted votes, submitted by N different voters, is shuffled (i.e. re-randomized and permuted), in turn, by L voting authorities. To ensure that the authorities are behaving honestly, each authority provides a non-interactive zero-knowledge proof that it has correctly shuffled the votes; if this is done using standard NIZKs, then in order to verify that the overall shuffling process was correct a verifier would need to access L separate proofs, each proving that an authority correctly performed the shuffling process. If each proof is of size $s(N)$, this means that the verifier's work is $\Theta(Ls(N))$ (here we ignore the security parameter). Using derivation-private cm-NIZKs, the verifier's workload can be reduced: each authority can, instead of producing a brand new proof, "maul" the proof of the previous authority; the proof produced by the last authority should then convince the verifier that the ciphertexts output at the end are a valid shuffling of the input ciphertexts. This makes vote shuffling a factor of L more efficient, as the verifier needs to verify a proof of size only $\Theta(s(N) + L)$. (The size of the proof is still dependent on L because each authority needs to, intuitively, add a "stamp of participation" in order for a verifier to ascertain that the shuffling process was performed correctly.)

CKLM then showed how to construct derivation-private cm-NIZK proof systems for a limited, but nevertheless expressive, class of transformations. Specifically, their approach builds heavily on the Groth-Sahai proof system [24]; this means that they can consider only relations on group elements in groups that admit bilinear pairings, and it might therefore seem as though controlled malleability were just a property of the Groth-Sahai proof system and not necessarily something that could be realized using more general building blocks. Interestingly, as a consequence of this limitation, CKLM did not fully deliver on the promise of a compactly verifiable shuffle: in order to prove that a given set of ciphertexts is a shuffle, they needed to represent everything, including the transformations applied to the set of ciphertexts, as a set of elements in the underlying group. The way they chose to do this was using a permutation matrix; since this permutation matrix needs to be extractable from the proof, the size of each proof in their construction was $\Theta(N^2 + L)$. For the usual voting scenario, in which the number of voters far exceeds the number of mix authorities, a vote shuffling scheme wherein each authority produces its own proof but the proofs are only of size $\Theta(N)$ (such as the verifiable shuffle of Groth and Lu [23]), therefore has a shorter proof overall.

Thus, the two important, and somewhat related open problems were: first, can a derivation-private controlled-malleable NIZK be realized in a modular fashion from general building blocks, without requiring the specific number-theoretic assumptions underlying the Groth-Sahai proof system? Second, can it be realized for general classes of languages and transformations, and not just those languages whose membership is expressible using pairing product equations over group elements as needed to invoke the Groth-Sahai proof system? In this paper, we give a positive answer to both.

Our Contributions. We first investigate how to construct a derivation-private cm-NIZK from succinct non-interactive arguments (SNARGs) [22,6]. We limit our attention to t-tiered languages and transformations; briefly, a language is t-tiered if each instance x can be efficiently labeled with an integer $i = \mathsf{tier}(x)$, $1 \leq i \leq t$, and a transformation T for a t-tiered language L is t-tiered if $\mathsf{tier}(T(x)) > \mathsf{tier}(x)$ for all $x \in L$ where $\mathsf{tier}(x) < t$, and $T(x) = \bot$ if $\mathsf{tier}(x) = t$. Some transformations are naturally t-tiered: for example, a vote shuffling transformation carried out by authority i should output a set of ciphertexts and stamps of approval from each authority up to i; furthermore, all transformations can be made t-tiered if one is willing to reveal how many times a transformation has been applied.

Intuitively, our construction works as follows: given a proof π for an instance $x \in L$, to provide a proof for a new instance $x' = T(x) \in L$, a user can form a "proof of a proof;" i.e., prove knowledge of this previous instance x and its proof π, as well as the transformation T from x to x', and call this proof π'. By the succinctness property of SNARGs, this new proof π' can in fact be the same size as the previous proof π, and thus this "proof of a proof" approach can be continued without incurring any blowup in size.

Although the intuition is relatively simple, going from SNARGs to cm-NIZKs is in fact quite challenging. While the outline above describes how to build malleability into SNARGs, it is still the case that SNARGs satisfy only the non-black-box notion of adaptive knowledge extraction, whereas cm-NIZKs require a much stronger (black-box) version of extractability. (This stronger notion is crucially used in the CCA encryption and the shuffle applications in CKLM.) To therefore break all these requirements up into smaller pieces, we begin with SNARGs and then slowly work our way up to cm-NIZKs in three separate constructions, with each construction incorporating an additional requirement.

We begin in Section 3.1 with a construction of a malleable SNARG. This construction closely follows the intuition above (which is itself inspired by the "targeted malleability" construction of Boneh et al. [6]): malleability is achieved by proving knowledge of either a fresh witness or a previous instance and proof, and a transformation from that instance to the current one. As observed by Bitansky et al. [3,4], care must be taken with this kind of recursive composition of SNARGs, as the size of the extractor can quickly blow up as we continue to extract proofs from other proofs; we can therefore construct t-tiered malleable SNARGs (i.e., SNARGs malleable with respect to the class of all t-tiered transformations) for only constant t. Furthermore, a formal treatment of our particular recursive technique reveals that a stronger notion of extraction, in which the extractor gets to see not only the random tape but also the code for the adversary, is necessary for both our construction and the original one of Boneh et al.

With our construction in Section 3.1, we therefore added malleability to the SNARG while preserving succinctness. In Section 3.2, we next tackle the issue of extractability; in particular, we want to boost from the non-black-box notion of extractability supported by SNARGs to the standard black-box notion of a proof of knowledge (NIZKPoK). To do this, we in fact rely only on the soundness of the SNARG, and do not attempt to use the (non-black-box) extractor at all. Instead, we perform a sort of verifiable encryption, in which we encrypt the witness and then prove knowledge (using the malleable SNARG) of the value inside the ciphertext; in this our approach is perhaps most similar to that of Damgård et al. [11]. A black-box extractor is then simple to construct: it just decrypts the ciphertext and thus, provided the proof is sound, recovers the witness. In addition, to preserve the full generality of our t-tiered transformations one would instantiate the encryption scheme using fully homomorphic encryption, although we will also see in Section 4 that interesting classes of transformations can still be supported by more limited schemes (such as ones that are multiplicatively homomorphic).

With our construction in Section 3.2, we therefore achieved the same properties that the Groth-Sahai proof system already provided (namely, a malleable NIWIPoK), but with respect to a more general class of transformations. As such, to now construct cm-NIZKs in Section 3.3, we can follow approximately the same construction as CKLM, who also used malleable NIWIPoKs to construct their cm-NIZK. Once again, however, care must be taken in this step, as we would like to preserve the generality in the class of transformations that we supported in the previous two sections. We therefore modify the CKLM construction to allow for this, and thus achieve cm-NIZKs for all t-tiered transformations.

In summary, we show that if zero-knowledge SNARGs exist for all languages in NP and fully homomorphic encryption exists, then derivation-private cm-NIZK proof systems exist for all t-tiered classes of transformations, where t is a constant. We do this by constructing three distinct types of proofs, each of which may be of independent interest: first, a malleable SNARG, then a malleable NIZKPoK, and finally a cm-NIZK. While each of our constructions builds from the previous one, we stress that our constructions are all fully generic; e.g., any malleable SNARG can be used to construct a malleable NIZKPoK, not just the specific one we construct.

Finally, in Section 4, we show how to use our SNARG-based proofs for t-tiered transformation classes (using just multiplicatively homomorphic encryption rather than the heavyweight requirement of fully homomorphic encryption) to construct a compact verifiable shuffle with proof size $\Theta(N + L)$ under general assumptions. This enhances CKLM in two ways: (1) CKLM had proof size $\Theta(N^2 + L)$; (2) CKLM required Groth-Sahai proofs, rather than general assumptions. In a separate paper [9], we showed that, by making additional assumptions about groups that admit bilinear pairings (similar to those made by Groth and

Lu [23]), we can also obtain a compact verifiable shuffle with proofs of size $\Theta(N + L)$ using the Groth-Sahai proof system.

2 Definitions and Notation

We recall the main security notions we use. We begin with the recent definitions for malleability due to CKLM [7], as well as their definition for compactly verifiable shuffles; we then define succinct non-interactive zero-knowledge arguments (SNARGs), which form the basis for our construction of malleable proofs in Section 3.

2.1 Malleable Proofs

Let $R(\cdot,\cdot)$ be a relation such that the corresponding language $L_R = \{x \mid \exists w \text{ such that } (x,w) \in R\}$ is in NP. As defined by CKLM, the relation is *closed* with respect to a transformation $T = (T_{\text{inst}}, T_{\text{wit}})$ if, for every $(x,w) \in R$, $(T_{\text{inst}}(x), T_{\text{wit}}(w)) \in R$ as well. We define zero knowledge and related notions formally in the full version of the paper [8], but recall briefly here that a non-interactive zero-knowledge (NIZK) proof system [5,14,20] is a set of algorithms $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V})$ for which there exists an efficient simulator (S_1, S_2) such that no adversary can distinguish between proofs formed by the prover and proofs formed by the simulator, and an efficient extractor (E_1, E_2) that can produce a witness w such that $(x,w) \in R$ from any valid proof π for x. For zero knowledge, we discuss here two additional variants: the first, *composable* zero knowledge, says that the adversary should still be unable to distinguish even give the simulation trapdoor, and the second, *statistical* zero knowledge, says that the distribution of proofs formed by the simulator and prover are indistinguishable even to an unbounded adversary; composable zero knowledge is thus implied by statistical zero knowledge, as an unbounded adversary could produce the simulator trapdoor itself.

To incorporate malleability, CKLM extend a NIZK $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V})$ to add an additional algorithm, ZKEval, that given a transformation T, a previous instance x, and a previous proof π such that $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$, computes a valid proof for $T_{\text{inst}}(x)$; i.e., a proof π' such that $\mathcal{V}(\mathsf{crs}, T_{\text{inst}}(x), \pi') = 1$. They then say that the proof system is *malleable* with respect to a set of transformations $\mathcal{T}$ if for every $T \in \mathcal{T}$, this computation can be performed efficiently. In terms of controlling malleability, the main definition of CKLM reconciles simulation soundness [29,12] and simulation-sound extractability [21] with malleability by requiring that, for a set of transformations $\mathcal{T}$, if an adversary can produce a proof π that $x \in L_R$ then the extractor can extract from π either a witness w or a transformation $T \in \mathcal{T}$ and previously proved instance x' such that $x = T_{\text{inst}}(x')$. This is defined more formally as:

Definition 2.1. [7] *Let* $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V}, \mathsf{ZKEval})$ *be a NIZKPoK system for an efficient relation* R*, with a simulator* (S_1, S_2) *and an extractor* (E_1, E_2)*. Let*

$\mathcal{T}$ be a set of unary transformations for the relation R such that membership in $\mathcal{T}$ is efficiently testable. Let SE_1 be an algorithm that, on input 1^k, outputs $(\mathsf{crs}, \tau_s, \tau_e)$ such that (crs, τ_s) is distributed identically to the output of S_1. Let $\mathcal{A}$ be given, let $Q := Q_{inst} \times Q_{proof}$ be a table for storing the instances queried to S_2 and the proofs given in response, and consider the following game:

- *Step 1.* $(\mathsf{crs}, \tau_s, \tau_e) \xleftarrow{\$} SE_1(1^k)$.
- *Step 2.* $(x, \pi) \xleftarrow{\$} \mathcal{A}^{S_2(\mathsf{crs}, \tau_s, \cdot)}(\mathsf{crs}, \tau_e)$.
- *Step 3.* $(w, x', T) \leftarrow E_2(\mathsf{crs}, \tau_e, x, \pi)$.
- *Step 4.* $b \leftarrow ((w \neq \bot \wedge (x, w) \notin R) \vee$

$$((x', T) \neq (\bot, \bot) \wedge (x' \notin Q_{inst} \vee x \neq T_{inst}(x') \vee T \notin \mathcal{T})) \vee$$
$$(w, x', T) = (\bot, \bot, \bot))$$

The NIZKPoK satisfies controlled-malleable simulation-sound extractability *(CM-SSE, for short) with respect to $\mathcal{T}$ if for all PPT algorithms $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that the probability (over the choices of SE_1, $\mathcal{A}$, and S_2) that $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$ and $(x, \pi) \notin Q$ but $b = 1$ is at most $\nu(k)$.*

CKLM also defined the notion of *derivation privacy* for malleable proofs, which says that proofs should not reveal whether they were formed fresh or via transformation.

Definition 2.2. [7] *For a non-interactive proof* $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V}, \mathsf{ZKEval})$*, an efficient relation R malleable with respect to $\mathcal{T}$, an adversary $\mathcal{A}$, and a bit b, let $p_b^{\mathcal{A}}(k)$ be the probability of the event that $b' = 0$ in the following game:*

- *Step 1.* $\mathsf{crs} \xleftarrow{\$} \mathsf{CRSSetup}(1^k)$.
- *Step 2.* $(\mathsf{state}, x_1, w_1, \pi_1, \ldots, x_q, w_q, \pi_q, T) \xleftarrow{\$} \mathcal{A}(\mathsf{crs})$.
- *Step 3. If $\mathcal{V}(\mathsf{crs}, x_i, \pi_i) = 0$ for some i, $(x_i, w_i) \notin R$ for some i, or $T \notin \mathcal{T}$, abort and output $\bot$. Otherwise, form*

$$\pi \xleftarrow{\$} \begin{cases} \mathcal{P}(\mathsf{crs}, T_{inst}(x_1, \ldots, x_q), T_{wit}(w_1, \ldots, w_q)) & \text{if } b = 0 \\ \mathsf{ZKEval}(\mathsf{crs}, T, \{x_i, \pi_i\}_{i=1}^q) & \text{if } b = 1. \end{cases}$$

- *Step 4.* $b' \xleftarrow{\$} \mathcal{A}(\mathsf{state}, \pi)$.

Then the proof system is derivation private *if for all PPT algorithms $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that $|p_0^{\mathcal{A}}(k) - p_1^{\mathcal{A}}(k)| < \nu(k)$.*

CKLM give a zero-knowledge variant of this definition called *strong* derivation privacy, in which proofs output by ZKEval should be indistinguishable from those output by the simulator. The security experiment is almost the same, with the only differences being that $\mathcal{A}$ is given the simulation trapdoor, $\mathcal{A}$ is not required to output any witnesses, and S_2 is used in place of $\mathcal{P}$. Putting these all together, if a proof system is zero knowledge, strongly derivation private, and CM-SSE, then CKLM call it a *cm-NIZK*.

2.2 Succinct Non-Interactive Arguments of Knowledge

Our cm-NIZK construction in Section 3 builds on succinct non-interactive arguments of knowledge, or SNARGs (also called SNARKs) for short. Proofs of this kind were first shown to exist by Micali in 2000 [26], who used the Fiat-Shamir heuristic [15] to eliminate the interaction in previous succinct arguments. More recently, Groth provided a construction using pairings [22] which was improved by Lipmaa [25], Bitansky et al. [3] constructed designated-verifier SNARGs using the new notion of extractable collision-resistant hash functions, and Gennaro et al. [17] constructed constant-sized SNARGs with a relatively short common reference string.

Our definition is based primarily on that of Boneh et al. [6], although for the succinctness property we incorporate the definition of Gentry and Wichs [19] as well. In addition, to perform our recursive composition in Section 3.1, we require a stronger notion of extraction than the original definition provided; essentially, we consider adversaries that take in advice strings as input. Although we present two formulations below, *strong* and *generative* adaptive knowledge extraction, we note that these notions are in fact equivalent; a more in-depth discussion can be found in the full version.

Definition 2.3. *Let* $0 < \gamma < 1$ *be a constant. A* (strong) γ-succinct non-interactive argument of knowledge *for a relation* R *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V})$ *with the following properties:*

1. *Perfect completeness. For all* $k \in \mathbb{N}$, $(x, w) \in R$, $\mathsf{crs} \xleftarrow{\$} \mathsf{CRSSetup}(1^k)$, *and* $\pi \xleftarrow{\$} \mathcal{P}(\mathsf{crs}, x, w)$, *the probability that* $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$ *is* 1.
2. *Strong/generative adaptive knowledge extraction. For a PPT algorithm* $\mathcal{A}$, *let* $E_{\mathcal{A}}$ *be an associated PPT algorithm, and let* z *be a string whose size is polynomial in the security parameter. Then consider the following game:*
   - *Step 1.* $\mathsf{crs} \xleftarrow{\$} \mathsf{CRSSetup}(1^k)$; $r \xleftarrow{\$} \{0,1\}^*$.
   - *Step 2.* $(x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}, z; r)$.
   - *Step 3.* $w \leftarrow E_{\mathcal{A}}(\mathsf{crs}, z; r)$.

   We say the argument system satisfies strong adaptive knowledge extraction if for all PPT $\mathcal{A}$ *there exists an* $E_{\mathcal{A}}$ *and a negligible function* $\nu(\cdot)$ *such that for all* z *the probability (over the choices of* $\mathsf{CRSSetup}$ *and* r*) that* $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$ *but* $(x, w) \notin R$ *is at most* $\nu(k)$*. This corresponds to previous definitions of adaptive knowledge extraction if we consider only* $z = \perp$.

   In addition, it satisfies generative adaptive knowledge extraction if there exists a PPT algorithm $\mathcal{E}$ *such that for all PPT* $\mathcal{A}$ *there exists a negligible function* $\nu(\cdot)$ *such that, on input the code of* $\mathcal{A}$, $\mathcal{E}$ *produces an extractor* $E_{\mathcal{A}}$, *running in time polynomial in that of* $\mathcal{A}$, *such that for all* z *the probability (over the choices of* $\mathsf{CRSSetup}$ *and* r*) that* $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$ *but* $(x, w) \notin R$ *is at most* $\nu(k)$.
3. ϕ*-succinct arguments. For all* $k \in \mathbb{N}$, $(x, w) \in R$, *and* $\mathsf{crs} \xleftarrow{\$} \mathsf{CRSSetup}(1^k)$, *it holds that* $\mathcal{P}(\mathsf{crs}, x, w)$ *produces a distribution over strings of length at most* $\phi(k, |x|, |w|)$, *where* $\phi(k, |x|, |w|)$ *is bounded by* $poly(k) polylog(|x|) + \gamma|w|$ *for some constant* $0 < \gamma < 1$.

While the succinctness property of SNARGs is quite attractive for applications, it comes with a price: all known SNARG constructions are based on so-called "knowledge of exponent" assumptions [10,2]; furthermore, a recent result due to Gentry and Wichs [19] that separates SNARGS from all falsifiable assumptions suggests that this dependence is perhaps inherent. In addition, to satisfy our stronger version of adaptive knowledge extraction (either strong or generative; again, they are equivalent), the knowledge of exponent assumption used to prove the security of existing SNARG constructions [22,17] would have to be potentially strengthened to consider an extractor that has access to the code of $\mathcal{A}$; for more details, we defer to the full version.

The final observation we make about SNARGs is that the definition of adaptive knowledge extraction requires the extractor to have non-black-box access to the malicious prover; as we will see in Section 3.2, this can make SNARGs difficult to integrate into protocol design. Fortunately, we can easily see that this notion relates to the standard notion of soundness for proofs [14] (as used implicitly in Groth's SNARG construction [22]):

Theorem 2.1. *If a proof system* $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V})$ *satisfies adaptive knowledge extraction then it also satisfies adaptive computational soundness.*

Proof. To show this, we take an adversary $\mathcal{A}$ that can break the soundness of the proof system with non-negligible probability ϵ and use it to construct an adversary $\mathcal{B}$ that breaks adaptive knowledge extraction with the same probability ϵ. The code for $\mathcal{B}$ is simple: on input $(\mathsf{crs}; r)$, it gives crs to $\mathcal{A}$ (and implicitly runs it on a random tape $r' \subseteq r$), and when $\mathcal{A}$ outputs a pair (x, π) $\mathcal{B}$ outputs the same. By the definition of soundness, $\mathcal{A}$ will win if $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$ but $x \notin L_R$; this implies that, for any w output by $E_{\mathcal{B}}$, it must be the case that $(x, w) \notin R$, as otherwise $x \in L_R$. $\mathcal{B}$ will therefore succeed whenever $\mathcal{A}$ does and thus succeeds with probability ϵ.

3 A Construction of cm-NIZKs from SNARGs

In this section, we construct cm-NIZK proofs from zero-knowledge SNARGs that are malleable with respect to a wide range of transformations, namely all *t-tiered* transformation classes. Intuitively, a relation is t-tiered if each instance x lives in some tier i. We would like transformations to move up through the tiers, and we would also like ensure that at most t transformations are applied. Formally, we say that a relation $R^{(t)}$ is t-tiered if there exists an efficiently computable function $\mathsf{tier} : L_R^{(t)} \rightarrow [0, t]$ and $(\perp, \perp) \in R^{(t)}$, and that a transformation class $\mathcal{T}^{(t)}$ is t-tiered for $R^{(t)}$ if for all $T = (T_{\text{inst}}, T_{\text{wit}}) \in \mathcal{T}$ the following two conditions hold: (1) if $(x, w) \in R^{(t)}$ and $\mathsf{tier}(x) < t$, then $(T_{\text{inst}}(x), T_{\text{wit}}(w)) \in R^{(t)}$ and $\mathsf{tier}(T_{\text{inst}}(x)) > \mathsf{tier}(x)$; and (2) if $\mathsf{tier}(x) = t$ then $T_{\text{inst}}(x) = \perp$.

We summarize the contributions in this section in Figure 1. As discussed in the introduction, the construction in each subsection is used as a component in the next subsection's construction, with the end goal of constructing a cm-NIZK. In Section 3.1 we construct a SNARG, malleable with respect to a t-tiered transformation class, that we then use in Section 3.2 in combination with

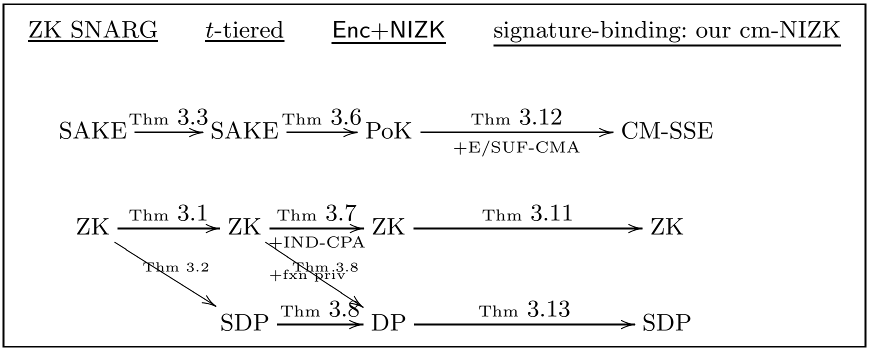

Fig. 1. The various relations among our constructions in this section. The arrows indicate which properties of the previous construction are used to obtain which properties of the next one, and are labeled on the top with the theorem number that proves the relation; the labels on the bottom indicate properties of additional primitives that are used as well. For example, we prove in Theorem 3.12 that our signature-binding construction of a cm-NIZK satisfies CM-SSE if our Enc+NIZK construction is a proof of knowledge, and the additional signature and one-time signature schemes we use are, respectively, unforgeable and strongly unforgeable; this is captured by the top rightmost arrow in the diagram. Strong adaptive knowledge extraction is written as SAKE, zero knowledge as ZK, proof of knowledge as PoK, and (strong) derivation privacy as (S)DP.

encryption to obtain a full NIZKPoK; this step seems necessary because SNARGs satisfy only the weak notion of adaptive knowledge extraction, which seems insufficient for constructing cm-NIZKs. Finally, using this NIZKPoK and a one-time and regular signature scheme, we construct in Section 3.3 a cm-NIZK that is malleable with respect to a broader class of transformations than could be supported by the construction of CKLM [7].

3.1 From SNARGs to Malleable but Weakly Extractable Proofs

We begin by constructing a derivation-private NIZK for a relation $R^{(t)}$, malleable with respect to a t-tiered transformation class $\mathcal{T}^{(t)}$, that achieves some degree of knowledge extraction. Our approach in this endeavor is inspired by that of Boneh et al. [6], who use SNARGs to construct a "targeted malleable" encryption scheme. To form a proof for an instance x_0 at the bottom level, one can use the SNARG directly to obtain a proof π_0. Now, suppose we would like to further form a proof for an instance $x_1 = T_{\text{inst}}(x_0)$; one option is to use the witness $T_{\text{wit}}(w_0)$ and form a fresh proof just as we did for x_0. Another option, however, is to "maul" the proof π_0: this can be accomplished by forming a new proof π_1 that proves knowledge of the old proof π_0 and instance x_0, as well as a transformation T such that $x_1 = T_{\text{inst}}(x_0)$.

The reason why SNARGs are attractive for this application is that, because the extraction procedure is non-black-box and therefore the proofs can be succinct, the proof π_1 can in fact be the same size as the proof π_0. Continuing in this fashion, we can see that at the i-th level, a proof for x_i can be proved using either knowledge of a witness w_i for the relation $R^{(t)}$, or knowledge of a proof π_{i-1} for x_{i-1} and a transformation T such that $x_i = T_{\text{inst}}(x_{i-1})$.

It turns out that, if the SNARG proof system used is zero knowledge (or even just witness indistinguishable), then the resulting proof system is derivation private. As mentioned above, however, the notion of extractability we can satisfy is still only the weak notion of adaptive knowledge extraction that SNARGs provide. In the next section, we show how to bootstrap this construction to obtain a proof system that satisfies the standard notion of extractability for proofs of knowledge (and still satisfies all the malleability and derivation privacy requirements).

To begin our construction, we first formalize the intuition developed above by defining the languages we use: at the bottom level at $i = 0$ we have $L_0 := \left\{x \mid \exists\ w \text{ s.t. } (x, w) \in R^{(t)}\right\}$, and for i such that $1 \le i \le t$, we have

$$L_i := \left\{ (x, \mathsf{crs}_{i-1}, \ldots, \mathsf{crs}_0) \;\middle|\; \begin{array}{c} \exists\ (w, x', \pi', T) \text{ s.t } (x, w) \in R^{(t)} \text{ or} \\ \mathcal{V}_{i-1}(\mathsf{crs}_{i-1}, (x', \mathsf{crs}_{i-2}, \ldots, \mathsf{crs}_0), \pi') = 1, \\ T_{\text{inst}}(x') = x, \text{ and } T \in \mathcal{T}^{(t)} \end{array} \right\}$$

Using these languages and $t + 1$ SNARG systems $(\mathsf{CRSSetup}_i, \mathcal{P}_i, \mathcal{V}_i)$, we now define our malleable t-tiered construction for $R^{(t)}$.

- $\mathsf{CRSSetup}(1^k)$: Generate $\mathsf{crs}_i \xleftarrow{\$} \mathsf{CRSSetup}_i(1^k)$ for all i, $0 \le i \le t$. Output $\mathsf{crs} := (\mathsf{crs}_0, \ldots, \mathsf{crs}_t)$.
- $\mathcal{P}(\mathsf{crs}, x, w)$: Compute $i := \mathsf{tier}(x)$; output $\pi \xleftarrow{\$} \mathcal{P}_i(\mathsf{crs}_i, (x, \mathsf{crs}_{i-1}, \ldots, \mathsf{crs}_0), (w, \perp, \perp, \perp))$.
- $\mathcal{V}(\mathsf{crs}, x, \pi)$: Compute $i := \mathsf{tier}(x)$ and output $\mathcal{V}_i(\mathsf{crs}_i, (x, \mathsf{crs}_{i-1}, \ldots, \mathsf{crs}_0), \pi)$.
- $\mathsf{ZKEval}(\mathsf{crs}, T, x, \pi)$: Compute $i := \mathsf{tier}(x)$, define $x' := T_{\text{inst}}(x)$, and output $\pi \xleftarrow{\$} \mathcal{P}_{i+1}(\mathsf{crs}_{i+1}, (x', \mathsf{crs}_i, \ldots, \mathsf{crs}_0), (\perp, x, \pi, T))$.

Recall that there are three properties we would like this proof system to satisfy: (1) zero knowledge, (2) derivation privacy, and (3) strong adaptive knowledge extraction; we deal with each of these in turn. For the first, zero knowledge, if we assume that our underlying proof systems are zero knowledge then we get a proof of the following theorem for free:

Theorem 3.1. *If the SNARG systems* $(\mathsf{CRSSetup}_i, \mathcal{P}_i, \mathcal{V}_i)$ *are zero knowledge for all* i*,* $0 \le i \le t$*, then the* t*-tiered construction is zero knowledge.*

We next turn to derivation privacy. At first glance, it would seem impossible that our construction could meet derivation privacy: after all, $\mathsf{tier}(x)$ openly reveals exactly how many times a transformation has been applied! Looking at the definition of the prover $\mathcal{P}$, however, we see that for x such that $\mathsf{tier}(x) = i$ it does in fact output a proof that "looks like" i transformations have been applied, even though it is using a fresh witness; as this is what the definition of derivation privacy requires (i.e., that the proof, rather than the instance, not reveal the transformation), we therefore use the witness indistinguishability of the SNARGs (which trivially follows from zero knowledge) to show that derivation privacy does hold. In addition, to show that strong derivation privacy holds, we require our SNARGs to be composable zero knowledge (as the adversary in the strong derivation privacy game gets to see the simulation trapdoor, and thus the zero knowledge adversary needs to as well); this requirement is met, for example, by the SNARG constructions of Groth [22] and Gennaro et al. [17], both of which actually satisfy the significantly stronger property of statistical zero knowledge. Due to space constraints, a proof of the following theorem can be found in the full version [8].

Theorem 3.2. *If the SNARG systems* $(\mathsf{CRSSetup}_i, \mathcal{P}_i, \mathcal{V}_i)$ *satisfy witness indistinguishability for all* i*, then the* t*-tiered construction satisfies derivation privacy for transformations in* $\mathcal{T}^{(t)}$*. Furthermore, if* $(\mathsf{CRSSetup}_i, \mathcal{P}_i, \mathcal{V}_i)$ *satisfy composable zero-knowledge for all* i*, then the* t*-tiered construction satisfies both derivation privacy and strong derivation privacy for transformations in* $\mathcal{T}^{(t)}$*.*

Next, we turn to adaptive knowledge extraction; here, we can show that if the number of times the "proof of a proof" method has been applied is constant, then the t-tiered construction is strongly adaptive knowledge extractable. As do Boneh et al. [6], we require t be constant so the runtime of the extractor does not blow up: if $\mathcal{A}$ runs in time τ, and we require the runtime of the extractor to be only polynomial in the runtime of $\mathcal{A}$, then the extraction of the t-th nested proof (i.e., if $\mathcal{A}$ has formed a proof of a proof t times) might take time $a^t\tau + tb$ for some constants a and b, which for arbitrary t could be exponential. To ensure that the time taken to extract from these nested proofs instead remains polynomial, we therefore require that t be constant. Furthermore, as we will see in the proof we rely on strong adaptive knowledge extraction to perform our recursive extraction (again, as do Boneh et al.). A proof of the following theorem can be found in the full version.

Theorem 3.3. *If the SNARG systems* $(\mathsf{CRSSetup}_i, \mathcal{P}_i, \mathcal{V}_i)$ *satisfy strong adaptive knowledge extraction (as defined in Definition 2.3) for all* i*, then the* t*-tiered construction satisfies strong adaptive knowledge extraction for constant* t*.*

Finally, we discuss the size of the proofs. Looking at the language L_i for some level, we see that an instance for the next language L_{i+1} consists of the same elements as an instance of L_i, with the addition of the CRS crs_i. If we consider, for example, the SNARG construction of Groth [22], then the size of crs_i is $O(|x^{(i)}|^2)$ for $x^{(i)} \in L_i$. Let f be the function that computes the size of the instance at level $i+1$ given the size of the instance x at level i. Then, because an element of size $|x|^2$ is added to obtain the instance for the next level up, we have that $f(f(|x|)) = |x|^4$, and, after t transformations, that $f^t(|x_0|) > |x_0|^{2^t}$. If t is constant, the fact that we require SNARGs to be of size polylog$(|x|)$ accounts for every such polynomial factor. Considering next the witness, we observe that the size of the witness $w^{(i)}$ for $i > 0$ is $|w_i| + |x_{i-1}| + |\pi_{i-1}| + |T_i|$. In order for our proofs to be succinct, we require that $|\pi_i| \leq |\pi_{i-1}|$. If we assume that $|w_i| \leq |w_{i-1}|$, $|x_i| \leq |x_{i-1}|$, and $|T_i| \leq |T_{i-1}|$ and that $w^{(i)} = |w_i| + |x_{i-1}| +$

$|\pi_{i-1}| + |T_i| \leq 4|\pi_{i-1}|$, then a $\mathrm{poly}(k)\mathrm{polylog}(|x|) + \gamma|w|$ succinct SNARG with $\gamma = 1/4$ is sufficient for our construction.

3.2 From Weak Malleable Proofs to Malleable Proofs of Knowledge

With our malleable NIZK in place, we might now try to use it to directly construct a cm-NIZK or, because we can satisfy only adaptive knowledge extraction, a weakened notion of cm-NIZK that accomodates this weaker extractability property. Looking back at the definition of controlled malleability (CM-SSE) in Definition 2.1, however, we can see that $\mathcal{A}$ is given access to a simulation oracle S_2. This oracle access seems to be fundamental to the definition: to achieve any kind of simulation soundness, in which we want $\mathcal{A}$ to be unable to produce its own proofs of false statements even after seeing many such proofs, we must give it an oracle that can produce false proofs. If we attempt to then use any non-black-box notion of extractability in conjunction with such an oracle, it is not clear how such an extractor would even be defined, as it cannot simply run the code for $\mathcal{A}$ (in particular, because the oracle's ability to produce false proofs must be presumably unavailable to $\mathcal{A}$ and therefore $E_{\mathcal{A}}$).

To avoid this obstacle altogether, we instead augment the construction from the previous section to achieve full extractability. To do this, our proofs consist of a ciphertext encrypting the witness, and a malleable zero-knowledge SNARG proving knowledge of the value inside of this ciphertext. Now, rather than require the use of the non-black-box extractor to prove any kind of extractability, we can instead give an extractor the secret key, and it can extract by decrypting the ciphertext. As we will see in our proof of Theorem 3.6, this means that all is required of the SNARG is soundness (which, we recall by Theorem 2.1, is implied by adaptive knowledge extraction).

In more detail, to construct a malleable NIZKPoK for a relation $R^{(pok)}$ and transformation class $\mathcal{T}^{(pok)}$, we use an encryption scheme and a proof system for the relation $R^{(t)}$ such that

$$((pk, x, c), (w, r)) \in R^{(t)} \iff c = \mathsf{Enc}(pk, w; r) \wedge (x, w) \in R^{(pok)}.$$

As for malleability, suppose we want to be able to transform the proofs for $R^{(pok)}$ with respect to some transformation class $\mathcal{T}^{(pok)}$. In order to implement ZKEval for a transformation $T = (T_{\mathrm{inst}}, T_{\mathrm{wit}}) \in \mathcal{T}^{(pok)}$, we will need to be able to transform the proof for $R^{(t)}$ and the ciphertext c. For the latter, this means we need to be able to apply a transformation T_c on the ciphertext that produces an encryption of $T_{\mathrm{wit}}(w)$; i.e., the homomorphic property of the encryption scheme must be robust enough to allow us to apply T_{wit} to the encrypted message. For the proof, we also require a transformation T_r on the randomness r of the ciphertext, as we require a transformation that maps (pk, x, c) to $(pk, T_{\mathrm{inst}}(x), T_c(c))$ and (w, r) to $(T_{\mathrm{wit}}(w), T_r(r))$.

A bit more formally, for every $T = (T_{\mathrm{inst}}, T_{\mathrm{wit}}) \in \mathcal{T}^{(pok)}$ and r' from the randomness space $\mathcal{R}$, let T_c be the transformation that maps $c = \mathsf{Enc}(w; r)$ to $\mathsf{Eval}(c, T_{\mathrm{wit}}; r') = \mathsf{Enc}(T_{\mathrm{wit}}(w); r \circ r')$ (where $\circ$ denotes the operation that composes the randomness, and Eval denotes the homomorphic operation on ciphertexts), let T_r be the resulting transformation on the randomness, and let $\tau(T, r')$ be the transformation that maps instances (x, c) to new instances $(T_{\text{inst}}(x), T_c(c))$, and witnesses (w, r) to new witnesses $(T_{\text{wit}}(w), T_r(r))$ (i.e., the exact transformation we need for the proof). Finally, let $\mathcal{T}^{(t)}$ be the set of transformations that includes $\tau(T, r')$ for all $T \in \mathcal{T}^{(pok)}$, $r' \in \mathcal{R}$, and let $\mathcal{T}^{(E)}$ be the set of all T_{wit}.

To give our Enc+NIZK construction for $R^{(pok)}$, let $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a function-private homomorphic encryption schemewith randomness space $\mathcal{R}$ and let $(\mathsf{CRSSetup}', \mathcal{P}', \mathcal{V}', \mathsf{ZKEval}')$ be a malleable zero-knowledge SNARG for the relation $R^{(t)}$ with transformation set $\mathcal{T}^{(t)}$. Our construction of a NIZKPoK is as follows:

- $\mathsf{CRSSetup}(1^k)$: Generate $\mathsf{crs}' \xleftarrow{\$} \mathsf{CRSSetup}'(1^k)$ and $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$ and output $\mathsf{crs} := (\mathsf{crs}', pk)$.
- $\mathcal{P}(\mathsf{crs}, x, w)$: Parse $\mathsf{crs} = (\mathsf{crs}', pk)$ and pick randomness $r \xleftarrow{\$} \mathcal{R}$. Then compute $c \leftarrow \mathsf{Enc}(pk, w; r)$ and $\pi' \xleftarrow{\$} \mathcal{P}'(\mathsf{crs}', (pk, x, c), (w, r))$ and output $\pi := (\pi', c)$.
- $\mathcal{V}(\mathsf{crs}, x, \pi)$: Parse $\mathsf{crs} = (\mathsf{crs}', pk)$ and $\pi = (\pi', c)$, and output $\mathcal{V}'(\mathsf{crs}', (pk, x, c), \pi')$.
- $\mathsf{ZKEval}(\mathsf{crs}, T, x, \pi)$: Parse $\mathsf{crs} = (\mathsf{crs}', pk)$, $\pi = (\pi', c)$, and $T = (T_{\text{inst}}, T_{\text{wit}})$. Then choose random $r' \xleftarrow{\$} \mathcal{R}$, compute $T' := \tau(T, r')$, and compute $\pi_T \xleftarrow{\$} \mathsf{ZKEval}'(\mathsf{crs}', T', (pk, x, c), \pi')$ and $c_T := \mathsf{Eval}(pk, T_{\text{wit}}, c; r')$. Output (π_T, c_T).

We make the following requirements on the underlying SNARG to obtain the completeness and malleability properties; both of them follow directly from the Enc+NIZK construction:

Theorem 3.4. *Let $\mathbb{W}^{(E+N)}$ be the witness space for $R^{(pok)}$. If the SNARG is complete for $R^{(t)}$ and the encryption scheme has message space $\mathcal{M}$ such that $\mathbb{W}^{(E+N)} \subseteq \mathcal{M}$, then the Enc+NIZK construction is complete.*

Theorem 3.5. *The Enc+NIZK construction is malleable with respect to $\mathcal{T}^{(pok)}$ whenever the SNARG is malleable with respect to the corresponding set $\mathcal{T}^{(t)} = \tau(\mathcal{T}^{(pok)}, \mathcal{R})$ and the encryption scheme is malleable with respect to $\mathcal{T}^{(E)}$ (as defined above).*

If $\mathcal{T}^{(pok)}$ is a t-tiered class of transformations on $R^{(pok)}$, then $\tau(\mathcal{T}^{(pok)})$ will also be t-tiered on $R^{(t)}$. Thus, if we instantiate $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ using a fully homomorphic encryption scheme and we use the SNARGs constructed in the previous section, we can obtain a malleable proof system for any t-tiered $\mathcal{T}^{(pok)}$ with constant t. (On the other hand, we will see in Section 4 that there are interesting relations and transformation classes we can obtain without fully homomorphic encryption as well.) As for size efficiency, we know by the succinctness property of SNARGs that the size of π' will not grow through transformation. For the ciphertext c, if we assume that T_{wit} does not increase the size of the witness, then the size of c will stay the same as well and thus the proof will remain compact even as it is transformed.

We would now like to show that if the SNARG satisfies adaptive knowledge extraction then the $\mathsf{Enc+NIZK}$ construction satisfies extractability; i.e., is an argument of knowledge. We also must show that the construction retains the original zero knowledge and derivation privacy properties as well. Due to space constraints, proofs of the following three theorems can be found in the full version.

Theorem 3.6. *If the SNARG satisfies adaptive knowledge extraction with respect to $R^{(t)}$ then the $\mathsf{Enc+NIZK}$ construction is a proof of knowledge with respect to $R^{(pok)}$.*

Theorem 3.7. *If the SNARG is zero knowledge and the encryption scheme is IND-CPA secure, then the $\mathsf{Enc+NIZK}$ construction is zero knowledge.*

Theorem 3.8. *If the SNARG is zero knowledge and strongly derivation private with respect to the class of transformations $\mathcal{T}^{(t)}$ and the encryption scheme is function private with respect to $\mathcal{T}^{(E)}$ then the $\mathsf{Enc+NIZK}$ construction is derivation private with respect to $\mathcal{T}^{(pok)}$.*

3.3 From Malleable NIWIPoKs to cm-NIZKs

With our malleable NIZKPoK in place, we are finally ready to construct cm-NIZKs (although, as we will see, we require only witness indistinguishability rather than full zero knowledge). We first recall the construction of CKLM, who used a relation R' such that $((x, vk), (w, x', T, \sigma)) \in R'$ if $(x, w) \in R$ or $\mathsf{Verify}(vk, \sigma, x') = 1$, $x = T_{\text{inst}}(x')$, and $T \in \mathcal{T}$, where σ was a signature for a secure signature scheme. We use the CKLM construction as a rough guideline for our own; the crucial alteration we make, however, is that CKLM were willing to retain the natural re-randomizability of Groth-Sahai proofs, whereas we want to consider classes of transformations that do not contain the identity (for example, the t-tiered transformation classes).

Suppose we want to construct a cm-NIZK for relation $R^{(cm)}$ and transformation class $\mathcal{T}^{(cm)}$. We use a NIWIPoK for an augmented relation $R^{(pok)}$ such that $((x, vk, vk_{\mathsf{ot}}), (w, x', vk'_{\mathsf{ot}}, T, \sigma)) \in R^{(pok)}$ if (1) $(x, w) \in R^{(cm)}$ or (2) $\mathsf{Verify}(vk, \sigma, (x', vk'_{\mathsf{ot}})) = 1$ and either (2a) $x = T_{\text{inst}}(x')$ for $T = (T_{\text{inst}}, T_{\text{wit}}) \in \mathcal{T}^{(cm)}$, or (2b) $x' = x$ and $vk'_{\mathsf{ot}} = vk_{\mathsf{ot}}$, where vk_{ot} is a verification key for a one-time signature scheme.

Intuitively, to simulate proofs, we can use this last type of witness; i.e., on a query x, the simulator can use sk as a trapdoor to sign (x, vk_{ot}) and produce a signature σ, and then form a proof using $(\perp, x, vk_{\mathsf{ot}}, \perp, \sigma)$ as a witness. To ensure that an adversary cannot simply reuse this proof and claim it as its own (i.e., apply the identity transformation), proofs are accompanied by a one-time signature, on both the instance and the proof, to indicate that the proof was formed fresh for this instance. Because the one-time signature thus binds together the instance and the proof, we call this construction “signature binding.”

Now, if we want to allow transformations $(\widehat{T}_{\text{inst}}, \widehat{T}_{\text{wit}}) \in \mathcal{T}^{(cm)}$ for our cm-NIZK, we will have to be able to transform the underlying NIWIPoK accordingly. To do this for any $\widehat{T} = (\widehat{T}_{\text{inst}}, \widehat{T}_{\text{wit}}) \in \mathcal{T}^{(cm)}$, and any $\widehat{vk}_{\mathsf{ot}} \in VK_{\mathsf{ot}}$ (where VK_{ot} is the set of all possible verification keys), let $\rho(\widehat{T}, \widehat{vk}_{\mathsf{ot}})$ be a transformation that maps $(x, vk, vk_{\mathsf{ot}})$ to $(\widehat{T}_{\text{inst}}(x), vk, \widehat{vk}_{\mathsf{ot}})$ and $(w, x', vk'_{\mathsf{ot}}, T, \sigma)$ to $(\widehat{T}_{\text{wit}}(w), x', vk'_{\mathsf{ot}}, \widehat{T} \circ T, \sigma)$. We require the underlying NIWIPoK to be malleable with respect to this class $\mathcal{T}^{(pok)}$.

More formally, let $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be an unforgeable signature scheme, $(\mathsf{KeyGen}_{\mathsf{ot}}, \mathsf{Sign}_{\mathsf{ot}}, \mathsf{Verify}_{\mathsf{ot}})$ be a strongly unforgeable one-time signature scheme, and let $(\mathsf{CRSSetup}_{\mathsf{WI}}, \mathcal{P}_{\mathsf{WI}}, \mathcal{V}_{\mathsf{WI}})$ be a malleable derivation-private NIWIPoK for $R^{(pok)}$. We give our construction of a cm-NIZK using these primitives as follows:

- $\mathsf{CRSSetup}(1^k)$: Generate $\mathsf{crs}_{\mathsf{WI}} \xleftarrow{\$} \mathsf{CRSSetup}_{\mathsf{WI}}(1^k)$; $(vk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$. Output $\mathsf{crs} := (\mathsf{crs}_{\mathsf{WI}}, vk)$.
- $\mathcal{P}(\mathsf{crs}, x, w)$: Parse $\mathsf{crs} = (\mathsf{crs}_{\mathsf{WI}}, vk)$ and compute $\pi' \xleftarrow{\$} \mathcal{P}_{\mathsf{WI}}(\mathsf{crs}_{\mathsf{WI}}, (x, vk, vk_{\mathsf{ot}}), (w, \perp, \perp, \perp, \perp))$. Generate $(vk_{\mathsf{ot}}, sk_{\mathsf{ot}}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathsf{ot}}(1^k)$, compute $\sigma_{\mathsf{ot}} \xleftarrow{\$} \mathsf{Sign}_{\mathsf{ot}}(sk_{\mathsf{ot}}, (x, \pi'))$, and output $\pi := (\pi', \sigma_{\mathsf{ot}}, vk_{\mathsf{ot}})$.
- $\mathcal{V}(\mathsf{crs}, x, \pi)$: Parse $\pi = (\pi', \sigma_{\mathsf{ot}}, vk_{\mathsf{ot}})$ and check that $\mathsf{Verify}_{\mathsf{ot}}(vk_{\mathsf{ot}}, \sigma_{\mathsf{ot}}, (x, \pi')) = 1$; if this fails then output 0. Otherwise, parse $\mathsf{crs} = (\mathsf{crs}_{\mathsf{WI}}, vk)$ and output $\mathcal{V}_{\mathsf{WI}}(\mathsf{crs}_{\mathsf{WI}}, (x, vk, vk_{\mathsf{ot}}), \pi')$.
- $\mathsf{ZKEval}(crs, T, x, \pi)$: Parse $\mathsf{crs} = (\mathsf{crs}_{\mathsf{WI}}, vk)$ and $\pi = (\pi', \sigma_{\mathsf{ot}}, vk_{\mathsf{ot}})$. Generate $(\widehat{vk}_{\mathsf{ot}}, \widehat{sk}_{\mathsf{ot}}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathsf{ot}}(1^k)$ and compute $\pi'' \xleftarrow{\$} \mathsf{ZKEval}_{\mathsf{WI}}(\mathsf{crs}_{\mathsf{WI}}, \rho(T, \widehat{vk}_{\mathsf{ot}}), (x, vk, vk_{\mathsf{ot}}), \pi')$ and $\sigma'_{\mathsf{ot}} \xleftarrow{\$} \mathsf{Sign}_{\mathsf{ot}}(\widehat{sk}_{\mathsf{ot}}, (x, \pi''))$. Output $(\pi'', \sigma'_{\mathsf{ot}}, \widehat{vk}_{\mathsf{ot}})$.

Although in using $\widehat{T} \circ T$ we require that $\mathcal{T}^{(cm)}$ be closed under composition, we note that this is not a strong restriction. Indeed, if $\mathcal{T}^{(cm)}$ is not closed under composition, then we can define the closure of $\mathcal{T}^{(cm)}$ to be the class of transformations $\mathcal{T}^{(cm)\prime}$ such that $T \in \mathcal{T}^{(cm)\prime}$ if and only if $T = T_1 \circ \ldots \circ T_j$ for $j < t$ and $T_1, \ldots, T_j \in \mathcal{T}^{(cm)}$. In this case, if we construct the NIWIPoK using our $\mathsf{Enc}{+}\mathsf{NIZK}$ construction, our proofs have to increase in size by a factor of t. (The encryption scheme used will have to have message space large enough to represent $T_1 \circ \ldots \circ T_t$ as $(T_1, \ldots, T_t)$.) On the other hand, this size increase is unavoidable for general transformations if we want to obtain a definition (like CM-SSE) in which a non-interactive black-box extractor must be able to extract the entire transformation performed.

By construction, we directly obtain the following theorems:

Theorem 3.9. *If the proof system* $(\mathsf{CRSSetup}_{\mathsf{WI}}, \mathcal{P}_{\mathsf{WI}}, \mathcal{V}_{\mathsf{WI}}, \mathsf{ZKEval}_{\mathsf{WI}})$ *is complete for relation* $R^{(pok)}$*, and the one-time signature is correct, then the signature-binding construction is complete for relation* $R^{(cm)}$*.*

Theorem 3.10. *If the proof system* $(\mathsf{CRSSetup}_{\mathsf{WI}}, \mathcal{P}_{\mathsf{WI}}, \mathcal{V}_{\mathsf{WI}}, \mathsf{ZKEval}_{\mathsf{WI}})$ *is malleable with respect to the transformation class* $\mathcal{T}^{(pok)} = \rho(\mathcal{T}^{(cm)}, VK_{\mathsf{ot}})$ *(as defined above), then the signature-binding construction is malleable for transformation class* $\mathcal{T}^{(cm)}$*.*

Now, if we want to instantiate the NIWIPoK using our $\mathsf{Enc}{+}\mathsf{NIZK}$ construction from the previous section, we must first ensure that $R^{(pok)}$ and $\mathcal{T}^{(pok)}$ satisfy the constraints discussed therein. In particular, we required that $\mathcal{T}^{(pok)}$ be a t-tiered transformation class for $R^{(pok)}$, and that there is an encryption scheme whose message space contains the witness space for $R^{(pok)}$ that is homomorphic with respect to the class of transformations $\{T_{\mathrm{wit}}\}$ for all $(T_{\mathrm{inst}}, T_{\mathrm{wit}}) \in \mathcal{T}^{(pok)}$.

Expanding on this last requirement, as our witnesses for $R^{(pok)}$ are of the form $(w, x', vk'_{\mathsf{ot}}, T, \sigma)$, we need to use an encryption scheme in which the message space subsumes the space of all of these values; i.e., the witness, instance, and transformation spaces, as well as the space of possible one-time verification keys and signatures. We also need the encryption scheme to be homomorphic with respect to the set of transformations that map $(w, x', vk'_{\mathsf{ot}}, T, \sigma)$ to $(\widehat{T}_{\mathrm{wit}}(w), x', vk'_{\mathsf{ot}}, \widehat{T} \circ T, \sigma)$ for any $(\widehat{T}_{\mathrm{inst}}, \widehat{T}_{\mathrm{wit}}) \in \mathcal{T}^{(cm)}$. Finally, we require that $\mathcal{T}^{(cm)}$ is t-tiered for $R^{(cm)}$, as this will guarantee that $\mathcal{T}^{(pok)}$ is t-tiered for $R^{(pok)}$. If we assume SNARGs for general languages and fully homomorphic encryption, then we can obtain a cm-NIZK for any t-tiered transformation class as long as t is constant; in Section 4, we will also see that we can construct cm-NIZKs for interesting relations using only multiplicatively homomorphic encryption. Moreover, if we continue our assumption from the previous section that $\widehat{T}_{\mathrm{wit}}$ does not increase the size of w, then the size of proofs will not grow by transformation here either.

Finally, in order to show that this is a cm-NIZK, we need to show that it satisfies zero knowledge, CM-SSE, and strong derivation privacy. Due to space constraints, proofs of the following three theorems can be found in the full version.

Theorem 3.11. *If the proof system* $(\mathsf{CRSSetup}_{\mathsf{WI}}, \mathcal{P}_{\mathsf{WI}}, \mathcal{V}_{\mathsf{WI}}, \mathsf{ZKEval}_{\mathsf{WI}})$ *is witness indistinguishable then the signature-binding construction is zero knowledge.*

Theorem 3.12. *If the signature scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is unforgeable (i.e., EUF-CMA secure), the one-time signature* $(\mathsf{KeyGen}_{\mathsf{ot}}, \mathsf{Sign}_{\mathsf{ot}}, \mathsf{Verify}_{\mathsf{ot}})$ *is strongly unforgeable (SUF-CMA secure), and the proof system* $(\mathsf{CRSSetup}_{\mathsf{WI}}, \mathcal{P}_{\mathsf{WI}}, \mathcal{V}_{\mathsf{WI}}, \mathsf{ZKEval}_{\mathsf{WI}})$ *is an argument of knowledge, the signature-binding construction satisfies the CM-SSE property.*

Theorem 3.13. *If the proof system* $(\mathsf{CRSSetup}_{\mathsf{WI}}, \mathcal{P}_{\mathsf{WI}}, \mathcal{V}_{\mathsf{WI}}, \mathsf{ZKEval}_{\mathsf{WI}})$ *is derivation private for* $\mathcal{T}^{(pok)}$ *then the signature-binding construction is strongly derivation private for* $\mathcal{T}^{(cm)}$*.*

4 A Compactly Verifiable Shuffle Using SNARGs

Now that we have just constructed our SNARG-based cm-NIZK, we consider how to use it to construct a compactly verifiable shuffle.

We start by defining formally the relation and transformations we want to use for shuffles. Abstractly, instances for the correctness of a shuffle are of the form $x = (pk, \{c_i\}_i, \{c'_i\}_i)$, where pk is a public key for a re-randomizable encryption scheme, $\{c_i\}_i$ are the original ciphertexts, and $\{c'_i\}_i$ are the shuffled ciphertexts. In addition, to allow each mix authority to prove that it participated in the shuffle, instances also contain a set $\{pk_j\}_j$ that consists of the public keys of the authorities that have participated thus far. Similarly, witnesses are of the form $w = (\varphi, \{R_i\}_i, \{sk_j\}_j)$, where φ is a permutation, $\{R_i\}_i$ are the re-randomization factors, and $\{sk_j\}_j$ are the secret keys corresponding to $\{pk_j\}_j$. The relation R is such that

$$\begin{aligned}((pk, \{c_i\}_i, &\{c'_i\}_i, \{pk_j\}_j), (\varphi, \{R_i\}_i, \{sk_j\}_j)) \in R \\ &\Leftrightarrow\ \{c'_i\}_i = \{\mathsf{ReRand}(pk, \varphi(c_i); R_i)\}_i \wedge (pk_j, sk_j) \in R_{pk}\ \forall j.\end{aligned}$$

Briefly, valid transformations in $\mathcal{T}$ should be shuffles. Ignoring the authority keys for now (details can be found in the original CKLM paper and the full version of this paper), we define transformations on instances as

$$T_{\mathrm{inst}}(x) = T_{(\varphi', \{R'_i\}_i)}(pk, \{c_i\}_i, \{c'_i\}_i) := (pk, \{c_i\}_i, \{\mathsf{ReRand}(pk, \varphi'(c_i); R'_i)\}_i)$$

and on witnesses as

$$T_{\mathrm{wit}}(w) = T_{(\varphi', \{R'_i\}_i)}(\varphi, \{R_i\}_i) := (\varphi' \circ \varphi, \{\varphi'(R_i) * R'_i\}_i),$$

where $*$ is the operation used to compose the randomness (i.e., $\mathsf{ReRand}(pk, \mathsf{ReRand}(pk, c; R), R') = \mathsf{ReRand}(pk, c; R * R')$).

4.1 Our Construction

The shuffle construction of CKLM [7] used four building blocks: a hard relation R_{pk}, a re-randomizable encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{ReRand})$, a proof of knowledge $(\mathsf{CRSSetup}, \mathcal{P}, \mathcal{V})$, and a cm-NIZK $(\mathsf{CRSSetup}', \mathcal{P}', \mathcal{V}', \mathsf{ZKEval}')$. As we just constructed a cm-NIZK, we can simply plug it into this generic construction, which CKLM already proved secure. What it remains to show is that the requirements placed on transformations in Sections 3.2 and Section 3.3 are met by the shuffle transformations.

Recall the general requirement for transformations from Section 3.3: because we must encrypt values of the form $(w, x', vk'_{\mathsf{ot}}, T, \sigma)$, we need an encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ that is homomorphic with respect to the set of transformations that map $(w, x', vk'_{\mathsf{ot}}, T, \sigma)$ to $(\widehat{T}_{\mathrm{wit}}(w), x', vk'_{\mathsf{ot}}, \widehat{T} \circ T, \sigma)$ for any $(\widehat{T}_{\mathrm{inst}}, \widehat{T}_{\mathrm{wit}}) \in \mathcal{T}^{(cm)}$.

In order to meet this requirement for shuffles, we must therefore consider how to encrypt and appropriately transform all of these values. For all of the values except w and T, however, they are unchanged by the transformation; our only requirement here is therefore that they can be encrypted, meaning the spaces they live in are subsumed by the message space. As for the values that do get transformed, w and T, as they are defined for the shuffle we must consider how to transform the permutation φ, the re-randomization values $\{R_i\}_i$, and the secret keys $\{sk_j\}_j$. We deal with each of these in turn.

To encrypt a permutation $\varphi \in S_n$, we represent it as its component-wise action on indices. Formally, we first consider the collection $(c_1, \ldots, c_n)$ in which $c_i \xleftarrow{\$} \mathsf{Enc}(pk, i)$ for all i; i.e., the collection of ciphertexts encrypting their own index within the set. Now, to represent φ, we compute $c_i^{(\varphi)} \xleftarrow{\$} \mathsf{Enc}(pk, \varphi(i))$ for all i, $1 \leq i \leq n$; the set $\{c_i^{(\varphi)}\}_{i=1}^n$ is then equal to $\varphi(\{c_i\})_{i=1}^n$. When we need to compose this φ with a new permutation φ' (e.g., to compute $T_{\mathrm{wit}}(w)$), we can compute $\{c_i^{(\varphi' \circ \varphi)}\}_{i=1}^n = \varphi'(\{c_i^{(\varphi)}\}_{i=1}^n) = \varphi'(\varphi(\{c_i\}_{i=1}^n))$, which does represent the composed permutation $\varphi' \circ \varphi$ as desired.

Moving on to the re-randomization values $\{R_i\}_i$, we start in the same vein as with the permutations: for all i, we compute $c_i^{(r)} \xleftarrow{\$} \mathsf{Enc}(pk, R_i)$. We now place our only requirement on the encryption scheme ($\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}$), which is that it must be homomorphic with respect to the $*$ operation (i.e., the operation used to compose randomness); namely that there exist a corresponding operation $\circledast$ on ciphertexts such that if c_1 is an encryption of m_1 and c_2 is an encryption of m_2 then $c_1 \circledast c_2$ is an encryption of $m_1 * m_2$. With such an operation in place, when we want to permute using φ and add in new randomness $\{R'_i\}_i$, we can compute $c_i^{(r*r')} := \varphi(c_i^{(r)}) \circledast \mathsf{Enc}(pk, R'_i)$. By the homomorphic properties of $\circledast$, $c_i^{(r*r')}$ will then be an encryption of $\varphi(R_i) * R'_i$.

Finally, for the keys, we note that as long as all values of sk_j lie in the message space then we are fine, as these values are simply appended to a list and thus do not need to be transformed.

As for the size of the resulting shuffle, we know that the CRS for the construction in Section 3.1 consists of t common references strings for the underlying SNARG. If we use the SNARG due to Gennaro et al. [17], in which the size of the CRS is linear in the circuit size, then the total size of the CRS is $O(\ell n)$. At the next level, in the $\mathsf{Enc}+\mathsf{NIZK}$ construction, we add a public key pk, and at the next level, in the signature-binding construction, we add a verification key vk. If the size of each of these values is constant with respect to n (or even of size $O(n)$), then we obtain an overall shuffle parameter size of $O(\ell n)$. For the proofs, we know from our discussion in Section 3 that their size will depend on the representation of the witnesses w, instances x, and transformations T. As we've defined things here, the representations of φ and $\{R_i\}_i$ require n ciphertexts each, which means the representations of w and T are $O(n + \ell)$, as they each also contain ℓ secret keys. Similarly, the size of the instance x is $O(n + \ell)$, as it contains two sets of n ciphertexts and a set of ℓ public keys. The overall size of the proof is therefore $O(n + \ell)$.

Although the proof size is therefore smaller, having parameters of size $O(\ell n)$ means that the total number of bits read by the verifier is still $O(\ell n)$ and thus there is no benefit over previous shuffles. To get a parameter size of only $O(k\ell)$ (for the security parameter k), we assume we have a SNARG with a CRS of length $O(n)$ and proofs of length $O(n)$, and a collision-resistant hash function $H(\cdot)$ that produces k-bit strings. Then a straightforward transformation gives a SNARG where the verifier needs a CRS of length k and proofs are of length $O(n)$ as follows: first, $\mathsf{CRSSetup}$ generates a CRS crs for the underlying scheme, and outputs both crs and $H(\mathsf{crs})$. Then, the prover produces not only a proof π but also a CRS crs' such that $H(\mathsf{crs}') = H(\mathsf{crs})$; the proof must then verify under crs'. In order to verify such a proof, the verifier need only take as CRS input the value $H(\mathsf{crs})$. Knowledge extraction of this SNARG follows from collision resistance and knowledge extraction of the underlying SNARG: if the adversary produces a crs' different from crs but such that $H(\mathsf{crs}') = H(\mathsf{crs})$ then it breaks the collision resistance of the hash function, and if it produces a proof under crs then the underlying extractor will work. If we then use this modified SNARG in our construction in Section 3.1, we get a malleable SNARG where the verifier takes as input a CRS of length $O(k\ell)$ and proofs of length $O(n)$, meaning the elections monitor in our shuffle takes in parameters of size $O(k\ell)$ and proofs of size $O(n + \ell)$.

Acknowledgments. Anna Lysyanskaya was supported by NSF grants 1012060, 0964379, 0831293, and Sarah Meiklejohn was supported in part by a MURI grant administered by the Air Force Office of Scientific Research and in part by a graduate fellowship from the Charles Lee Powell Foundation.

References

1. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable Proofs and Delegatable Anonymous Credentials. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 108–125. Springer, Heidelberg (2009)
2. Bellare, M., Palacio, A.: The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004)
3. Bitanksy, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Proceedings of ITCS 2012 (2012)
4. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: Recursive composition and bootstrapping for SNARKs and proof-carrying data. Cryptology ePrint Archive, Report 2012/095 (2012), http://eprint.iacr.org/2012/095
5. Blum, M., de Santis, A., Micali, S., Persiano, G.: Non-interactive zero-knowledge. SIAM Journal of Computing 20(6), 1084–1118 (1991)
6. Boneh, D., Segev, G., Waters, B.: Targeted malleability: homomorphic encryption for restricted computations. In: Proceedings of ITCS 2012 (2012)
7. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable Proof Systems and Applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 281–300. Springer, Heidelberg (2012)
8. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Succinct malleable NIZKs and an application to compact shuffles. Cryptology ePrint Archive, Report 2012/506 (2012), http://eprint.iacr.org/2012/506
9. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Verifiable elections that scale for free. In: Proceedings of PKC 2013 (to appear, 2013)
10. Damgård, I.B.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)

11. Damgård, I.B., Faust, S., Hazay, C.: Secure Two-Party Computation with Low Communication. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 54–74. Springer, Heidelberg (2012)
12. De Santis, A., Di Crescenzo, G., Ostrovsky, R., Persiano, G., Sahai, A.: Robust Non-interactive Zero Knowledge. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 566–598. Springer, Heidelberg (2001)
13. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: Proceedings of FOCS 2010, pp. 511–520 (2010)
14. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs under general assumptions. SIAM Journal of Computing 29(1), 1–28 (1999)
15. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
16. Fuchsbauer, G.: Commuting Signatures and Verifiable Encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 224–245. Springer, Heidelberg (2011)
17. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. Cryptology ePrint Archive, Report 2012/215 (2012), `http://eprint.iacr.org/2012/215`
18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of STOC 2009, pp. 169–178 (2009)
19. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Proceedings of STOC 2011, pp. 99–108 (2011)
20. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. In: Proceedings of STOC 1985, pp. 186–208 (1985)
21. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
22. Groth, J.: Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340. Springer, Heidelberg (2010)
23. Groth, J., Lu, S.: A Non-interactive Shuffle with Pairing Based Verifiability. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 51–67. Springer, Heidelberg (2007)
24. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
25. Lipmaa, H.: Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 169–189. Springer, Heidelberg (2012)
26. Micali, S.: Computationally sound proofs. SIAM Journal of Computing 30(4), 1253–1298 (2000)
27. Prabhakaran, M., Rosulek, M.: Rerandomizable RCCA Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 517–534. Springer, Heidelberg (2007)
28. Prabhakaran, M., Rosulek, M.: Homomorphic Encryption with CCA Security. In: Aceto, L., Damgård, I.B., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 667–678. Springer, Heidelberg (2008)
29. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: Proceedings of FOCS 1999, pp. 543–553 (1999)