NIWI and New Notions of Extraction for Algebraic Languages

Chaya Ganesh[1], Hamidreza Khoshakhlagh[2], and Roberto Parisella[3]

[1] Indian Institute of Science
chaya@iisc.ac.in
[2] Aarhus University
hamidreza@cs.au.dk
[3] Simula UiB
robertoparisella@hotmail.it

Abstract. We give an efficient construction of a computational non-interactive witness indistinguishable (NIWI) proof in the plain model, and investigate notions of extraction for NIZKs for algebraic languages. Our starting point is the recent work of Couteau and Hartmann (CRYPTO 2020) who developed a new framework (CH framework) for constructing non-interactive zero-knowledge proofs and arguments under falsifiable assumptions for a large class of languages called algebraic languages. In this paper, we construct an efficient NIWI proof in the plain model for algebraic languages based on the CH framework. In the plain model, our NIWI construction is more efficient for algebraic languages than state-of-the-art Groth-Ostrovsky-Sahai (GOS) NIWI (JACM 2012). Next, we explore knowledge soundness of NIZK systems in the CH framework. We define a notion of strong f-extractability, and show that the CH proof system satisfies this notion.

We then put forth a new definition of knowledge soundness called *semantic extraction*. We explore the relationship of semantic extraction with existing knowledge soundness definitions and show that it is a general definition that recovers black-box and non-black-box definitions as special cases. Finally, we show that NIZKs for algebraic languages in the CH framework cannot satisfy semantic extraction. We extend this impossibility to a class of NIZK arguments over algebraic languages, namely quasi-adaptive NIZK arguments that are constructed from smooth projective hash functions.

1 Introduction

Zero-knowledge proofs, introduced by Goldwasser, Micali and Rackoff [39], are cryptographic primitives that allow a prover to convince a verifier that a statement is true without revealing any other information. Zero-knowledge proof systems have a rich history in cryptography [37,31,12] finding numerous applications in cryptographic constructions such as identification schemes [30], public-key encryption [49], signature schemes [21], anonymous credentials [20], secure multi-party computation [38], and a wide variety of emerging applications.

The notion of zero-knowledge proof was later extended to non-interactive zero-knowledge (NIZK) proofs by Blum, Feldman and Micali [16] where there is a single message sent from the prover to the verifier. NIZKs are particularly useful in low-interaction settings, and feasibility is known for all of NP in the common reference string (CRS) model.

Pairing-based NIZKs. Starting from the work of Groth and Sahai [42], many pairing-based NIZK proof systems have been constructed. These proof systems avoid the need for expensive reductions to NP-complete languages and can directly handle a large class of languages over abelian groups.

Another line of work for constructing pairing-based NIZKs is via a smooth projective hash function (SPHF) [27]. For a language over some abelian group $\mathbb{G}_1$, a secret hashing key is embedded in group $\mathbb{G}_2$, and this NIZK proof can be verified via a pairing operation between $\mathbb{G}_1$ and $\mathbb{G}_2$. The SPHF-based approach leads to very efficient proofs for linear languages. However, they only provide a quasi-adaptive type of soundness, where the CRS can depend on the language.

NIWIs. One can relax the security of a NIZK argument to a Non-Interactive Witness Indistinguishable (NIWI) argument by replacing the zero-knowledge property with a weaker witness indistinguishability (WI) property. Unlike NIZKs for which we know impossibility in the plain model [16], and can therefore only exist in the CRS model, NIWIs are possible in the plain model. Informally, witness indistinguishability means that the verifier at the end of protocol, cannot guess which of the possible witnesses the prover used to compute the proof.

The general idea to construct a NIWI in the plain model, is to start from zero-knowledge proofs that are perfectly sound for some choice of the verifier randomness (or some choice of the CRS). Namely, we let the prover sample the randomness by itself and include additional checks to force the prover to compute at least one proof for such choice of randomness. The first NIWI construction in the plain model was proposed by Barak et al. [8] obtained by derandomizing any two-round zero-knowledge proof (ZAP) [28]. The idea behind the construction is to let the prover send a "high enough" number of proofs, each for a different choice of randomness, such that it is hard to cheat for all of them. There are however drawbacks that make such NIWI schemes unsuitable in practical applications. In the NIWI of [8], the prover has to compute a logarithmic (in the security parameter) number of proofs, which leads to inefficient schemes, both in terms of computation and communication, even starting from efficient, say, linear ZAPs. Also, security is based on a complexity theoretic assumption (namely $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ has a function of circuit complexity $2^{\Omega(n)}$) that implies $\mathbf{BPP} = \mathbf{P}$.

Groth, Ostrovsky and Sahai [41] proposed a different framework for NIWI proofs, which leads to more efficient proofs for concrete languages (instead of circuit satisfiability). The key idea in [41] is to force the prover to produce two CRSs, such that at least one of them guarantees perfect soundness. Moreover, the structure of the CRS is such that multiplication of one element can always transform a computationally sound CRS into a perfectly sound CRS. The NIWI proof system can now take advantage of the structure in the CRS as follows: the prover generates a CRS on its own and provides proofs under both the chosen CRS and its transformation. Perfect soundness holds by the fact that at least one of the two CRSs guarantees this property. Some of the issues in the construction of [8] mentioned above are overcome by the NIWI proof system of [41], thanks also to further optimizations [52]. Namely, it is based on well-established assumptions, and the number of proof elements is constant instead of logarithmic in the security parameter. However, for some applications, having communication complexity that is twice the size of a Groth-Sahai (GS) proof is still not practical, particularly considering that GS NIZK, and consequently the NIWI often comes with a drastic efficiency reduction due to the need for reducing the original language to an intermediate language supported by the GS proof system.

In this work, we construct more efficient computational NIWI proofs in the plain model for a larger class of languages.

CH framework. Recently, Couteau and Hartmann [25] developed a new framework (henceforth referred to as the CH framework) for constructing non-interactive zero-knowledge proofs and arguments for a broad class of languages under a falsifiable assumption. They provide several constructions whose efficiency is satisfactory for many applications and enjoy a number of interesting features such as having proofs that are as short as proofs resulting from the Fiat-Shamir transformation applied to Σ-protocols. Their approach, at a very high level, consists of compiling a Σ-protocol over an abelian group $\mathbb{G}_1$ into a non-interactive zero-knowledge argument over Type III pairings by embedding the challenge e into a group $\mathbb{G}_2$ and adding the embedded challenge to the CRS.

The work of [25] also obtains a simple and efficient ZAP argument in the plain model where the WI property holds statistically as opposed to all previous pairing-based constructions that satisfy computational WI. While this ZAP argument can be compiled directly into a non-interactive ZAP using the compiler of [8], the prover, as mentioned above, needs to send logarithmically many proofs, hence decreasing the efficiency of the original scheme.

CH framework with knowledge soundness. All aforementioned proof systems based on CH framework only guarantee soundness meaning that accepting proofs cannot be computed for false statements. Typically, applications require a stronger notion of soundness called *knowledge soundness* which guarantees that the prover *knows* a witness for a statement if it can make the verifier accept. This notion of knowledge soundness is formalized by the existence of an efficient extractor that can extract a valid witness from the prover whenever the prover provides a valid proof. Given that the NIZK systems in [25] only guarantee soundness, we investigate the possibility of knowledge soundness of the CH protocol, and pairing-based arguments in general.

> *Can we construct NIZK proofs in the CH framework with knowledge soundness?*

A naïve solution to provide extractability in the CRS model is to use well-known techniques to augment the statement with a trapdoor for extraction. In particular, given a CRS that contains a public key pk, the most efficient currently known approach is to ask the prover to encrypt the witness under pk and then prove that the ciphertext is computed correctly. The extractor can then use the secret key of pk to recover a valid witness from the proof. This however makes the proof size much larger. On a high level, this is because existing algebraic encryption schemes are not friendly enough with the CH framework, unless we perform the encryption bit-by-bit as in [47,14], which makes the construction undesirable. More importantly, the underlying NP relation is now changed into an augmented relation that should also manage the correctness of ciphertext computations. Our goal is however to study the (im)possibility of extractability for the standard CH framework without changing the underlying relation.

Another solution is to show extractability under knowledge assumptions, or in idealized models such as generic group model (GGM) [55] or algebraic group model (AGM) [33]. Indeed, it is not hard to show that CH NIZKs are knowledge sound in the AGM [4]. Gentry and Wichs [36] show impossibility of a black-box reduction to a falsifiable assumption to prove soundness for succinct arguments, where the proof size is logarithmic in the size of the witness and the statement. However, the use of idealized models or knowledge assumptions to prove knowledge soundness of *non-succinct* proof systems seems to be less justifiable.

At first look, it might seem like knowledge assumptions for extraction are justified since soundness of some CH NIZK is already based on a *non-falsifiable* version of the **extKerMDH** assumption. As per Naor's classification [48], knowledge assumptions are a class of non-falsifiable assumptions. However, since knowledge assumptions stipulate "feasibility" of efficient extraction, they do not fit within a taxonomy of *intractability* assumptions [51]. On the other hand, an assumption such as **extKerMDH**, while non-falsifiable, is still an intractability assumption that can be phrased as a game between an adversary and a challenger, albeit with an inefficient challenger.

1.1 Our Contributions

We study NIZK and NIWI constructions in the pairing-based setting and make the following contributions.

NIWI in the plain model. Different from the aforementioned idea of constructing NIWI in the plain model based on the CH framework [25] using the compiler of [8], we investigate a more efficient strategy inspired by the approach of [41] which allows the verifier to verify if, given a (small) set of CRSs, at least one of them is perfectly binding, without breaking soundness.

Our construction is based on the existence of an efficient algorithm that, given one CRS of the NIZK proof of [25], allows the verifier to check if it is a perfectly binding one without compromising the soundness property. The key idea in constructing such an algorithm is, at a high level, to add two additional group elements to the CRS, chosen such that assuming the existence of Type III pairings, it allows the verifier to (efficiently) check the distribution of the

[4] We show knowledge soundness of the CH argument in the AGM in Appendix D.1.

CRS (with a technique similar to what was done in [2]) while not compromising the WI property. Now, with the verifier equipped with such an algorithm, we construct a non-interactive ZAP by letting the prover compute this CRS and output it together with the proof.

We need additional ideas to prove security of the resulting construction. First, as noted in [25], the soundness of the resulting NIZK proof is based on the special soundness property of the underlying Σ-protocol. Soundness of our NIWI proof follows from the same reasoning and from the correctness of the algorithm that checks the distribution of the CRS. Indeed, if the verifier accepts, then the prover correctly sampled a perfectly binding CRS and thus soundness holds. To show WI, we rely on a new decisional assumption, which we validate in the AGM. The ability of the verifier to check the distribution of the CRS relies on DDH being easy, and therefore it is not possible to rely on DDH for WI.

CH framework with knowledge soundness. The proof and argument systems presented in [25] and our NIWI construction ensure only soundness. As our second contribution, we investigate knowledge soundness of NIZK systems in the CH framework.

f-extractability. We define a notion of *strong f-extractability* that extends related notions of partial extraction used in literature. Informally, an argument system satisfies f-extractability if there exists an efficient extractor that outputs $\widetilde{\mathtt{w}}$ whenever the verifier accepts the proof for statement $\mathtt{x}$, where $\widetilde{\mathtt{w}} = f(\mathtt{w})$ and $\mathtt{w}$ is a valid witness for $\mathtt{x}$. We extend the notion to strong f-extractability where we ask that the partial witness $\widetilde{\mathtt{w}}$ allows for efficiently deciding membership of the statement. We show that the CH proof system satisfies this notion where the extracted value is the encoding of a witness to $\mathbb{G}_2$.

Semantic extraction. We then investigate the possibility of *knowledge soundness* of the CH NIZKs, and pairing-based arguments in general. We show that the CH argument is knowledge sound in the Algebraic Group Model (AGM), and then ask the following question: can we show knowledge soundness in the standard model without relying on knowledge assumptions or show impossibility of extraction? Towards this end, we put forth a notion of extraction called *semantic extraction*, and prove that this notion of extraction is impossible for the CH argument. The intuition behind the definition of semantic extraction is to consider the random coins of the adversary as an input from a certain distribution. This makes it possible to associate a function to each adversary: the function that it computes on certain inputs including its random coins. We then require that adversaries that implement the same function, have the same extractor. We allow the flexibility to split the random coins in two distinct portions, and then allow the extractor to see only one of the two portions. This gives a general definition that, depending on how much randomness we allow the extractor to see, gradually makes the extractor more powerful. We then investigate the relationship between semantic extraction and classic notions of extraction. We show that semantic extraction is a *general* definition, that captures both white-box($\mathsf{n\text{-}BB}$) and black-box(BB) extraction. In particular, BB extraction trivially implies semantic extraction. Also a slightly weaker version of the other direction is true, when we give no randomness to the semantic extractor. Moreover, semantic extraction is equivalent to n-BB extraction, where we give to the extractor all the random coins of the adversary. Finally, we show impossibility of semantic extraction for CH argument: that no extractor that sees only a portion of the adversary's randomness can succeed. We then generalize this impossibility to a class of NIZK arguments over algebraic languages, namely *quasi-adaptive* NIZK arguments based on SPHFs. As a concrete case, we show that the most efficient Quasi-Adaptive NIZK construction of Kiltz and Wee [45] cannot be semantically extractable. While black-box extraction is impossible since the arguments are shorter than the witnesses, the impossibility of semantic extraction is a stronger result. We present this in Appendix D.4.

1.2 Technical Overview

In this section we provide a technical overview of our results. We start with an overview of our NIWI construction in the plain model. Then we discuss our definition of semantic extraction and sketch our impossibility result for semantic extractability of CH NIZKs.

NIWI in the plain model. The starting point of our construction is the NIZK proof for algebraic languages in [25] which is based on a compiler that converts a Σ-protocol with linear answers over a group $\mathbb{G}_1$ into a NIZK argument by embedding the verifier's challenge into a group $\mathbb{G}_2$ in the CRS.

Σ-protocols for linear languages. A linear language with language parameter $[\mathbf{M}]_1 \in \mathbb{G}_1^{n\times k}$ is defined as $\mathcal{L}_{\mathbf{M}} = \left\{[\mathbf{x}]_1 \in \mathbb{G}_1^n | \exists \mathbf{w} \in \mathbb{Z}_p^k : [\mathbf{x}]_1 = [\mathbf{M}]_1 \cdot \mathbf{w}\right\}$. A Σ-protocol for a linear language $\mathcal{L}_{\mathbf{M}}$ with corresponding relation $\mathcal{R}_{\mathbf{M}}$ is a three-move honest-verifier zero-knowledge (HVZK) proof system between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ with the following syntax. First, $\mathcal{P}$ with an input pair $([\mathbf{x}]_1, \mathbf{w})$ selects $\mathbf{r} \leftarrow \mathbb{Z}_p^k$ and sends a first message $[\mathbf{a}]_1 := [\mathbf{M}]_1 \cdot \mathbf{r} \in \mathbb{G}_1^n$ to $\mathcal{V}$. Next, $\mathcal{V}$ sends a random string $e \in \mathbb{Z}_p$ to $\mathcal{P}$. Finally, $\mathcal{P}$ sends a reply $\mathbf{d} := \mathbf{w}e + \mathbf{r} \in \mathbb{Z}_p^k$ to $\mathcal{V}$, who accepts the proof if $[\mathbf{M}]_1 \cdot \mathbf{d} = [\mathbf{x}]_1 e + [\mathbf{a}]_1$. The *special soundness* property states that for any $[\mathbf{x}]_1$ and any pair of accepting conversations $([\mathbf{a}]_1, e, \mathbf{d}), ([\mathbf{a}]_1, e', \mathbf{d}')$ on $[\mathbf{x}]_1$ where $e \neq e'$, one can efficiently compute a valid witness $\mathbf{w}$ for $[\mathbf{x}]_1$.

CH Compiler. Couteau and Hartmann [25] proposed the following approach to compile a Σ-protocol into a NIZK in the CRS model: the setup algorithm picks a random $e \in \mathbb{Z}_p$ and sets $[e]_2$ as the CRS. The prover computes $[\mathbf{a}]_1$ as in the Σ-protocol, and an embedding of $\mathbf{d}$ in $\mathbb{G}_2$ by computing $[\mathbf{d}]_2 := \mathbf{w} \cdot [e]_2 + \mathbf{r} \cdot [1]_2$. The proof can (publicly) be verified by checking if the pairing equation $[\mathbf{M}]_1[\mathbf{d}]_2 = [\mathbf{x}]_1[e]_2 + [\mathbf{a}]_1[1]_2$ holds. While this leads to an argument system with computational soundness, [25] further shows how to turn the argument into a proof by providing two challenges with two different generators in the CRS and having the prover answer both with the same randomness. The (unconditional) special soundness property of the underlying Σ-protocol now guarantees that a witness exists, resulting in perfect soundness.

The idea behind our NIWI construction is as follows: consider the CRS of the CH NIZK proof $[s_1, s_2, e_1 s_1, e_2 s_2]_2 \in \mathbb{G}_2^4$, where $e_1, e_2, s_1, s_2 \in \mathbb{Z}_p$, and $[e_1, e_2]_2$

play the role of the two challenges (embedded in $\mathbb{G}_2$) in the underlying Σ-protocol. Now, we have the prover pick the CRS and the verifier checks that this CRS computed by a potentially malicious prover is such that $e_1 \neq e_2$, so we can rely on the special soundness of the underlying Σ-protocol. We then prove that the proof is witness-indistinguishable by relying on a new decisional assumption that we show secure in the AGM. This observation leads us to a NIWI proof in the plain model, where we let the prover to choose the "crs" parameters by itself, such that it is verifiable that $e_1 \neq e_2$.

Extractability in the CH framework. We now give an overview of the extractability notions we explore, the new notion of *semantic extraction* we propose, and the impossibility of semantic extraction for CH NIZKs.

The standard definition of knowledge extraction asks for the existence of an efficient algorithm called *extractor* that takes as input a proof π of a statement $\mathtt{x}$ and returns a value $\mathtt{w}'$ such that $\mathtt{w}'$ is a witness for the truth of $\mathtt{x}$, i.e., $(\mathtt{x}, \mathtt{w}') \in \mathcal{R}$. While such *full extractability* captures the fact that the prover must have known the witness, there are instances where the existence of such a powerful extractor is unlikely; however, it is still possible to extract some partial information about the witness. One concrete example is the Groth-Sahai non-interactive proof of knowledge [42] from which one can only extract a one-way function of the witness $f(\mathtt{w})$ where $f : \mathbb{F} \rightarrow \mathbb{G}$ is the encoding of the witness in the underlying group. The barrier to full extractability is the fact that there does not seem to be a trapdoor that can be used to compute, in an efficient way, a witness $\mathtt{w}$ from $f(\mathtt{w})$ (i.e., discrete logarithm problem). To capture this notion of *partial extractability*, Belenkiy et al. [10] formalized the notion of f-extractability by the existence of an efficient algorithm that outputs $\widetilde{\mathtt{w}}$ such that there exists some $\mathtt{w}$ with $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ and $\widetilde{\mathtt{w}} = f(\mathtt{w})$[5]. In their context of constructing anonymous credentials, f-extractability is used by relaxing the notion of unforgeability to f-unforgeable signatures where the adversary produces $(f(m), \sigma)$ pair (as opposed to (m, σ)) without previously obtaining a signature on m. Since then, f-extractability has been used as a standard property in many privacy-preserving authentication mechanisms [4,18,29,40,43,53].

We begin with this observation that the CH NIZK proof is not only f-extractable for $f := [\cdot]_2$, but the extracted value also allows to decide the membership of the statement via pairing checks. To see this, let $([\mathbf{x}]_1, \mathbf{w})$ be a pair of statement-witness in the linear relation $\mathcal{R}_{\mathbf{M}}$ that returns 1 if $[\mathbf{x}]_1 = [\mathbf{M}]_1 \cdot \mathbf{w}$. One can observe that extracting $[\mathbf{w}]_2$ suffices to decide the membership of $[\mathbf{x}]_1$ by checking if $[\mathbf{M}]_1[\mathbf{w}]_2 = [\mathbf{x}]_1[1]_2$. The primary distinction between partial and full extractability is in the ability to decide membership of the statement being proven via the extracted value. We fill the gap between the two notions by defining a stronger form of partial extractability called *strong f-extractability* which guarantees the existence of an efficient procedure D that for any given statement $\mathtt{x}$ and f-extracted value $\widetilde{\mathtt{w}} := f(\mathtt{w})$, $\mathsf{D}(\mathtt{x}, \widetilde{\mathtt{w}})$ can decide the membership of $\mathtt{x}$. Note that $\widetilde{\mathtt{w}}$ still falls short of being a full witness for the relation; assuming that f is

[5] Note that this a generalization of the standard notion as the identity function $f(\cdot)$ implies full extractability.

one-way, $\widetilde{\mathtt{w}}$ cannot be used to produce a valid proof for $\mathtt{x}$. This is what separates strong f-extractability from full extractability.

Impossibility of Semantic Extraction. We show impossibility of semantic extraction for the CH NIZK argument for algebraic languages. Note that this is a stronger result than ruling out BB extraction. Our impossibility holds only for semantic extraction where there exists a portion of the adversary's randomness that the extractor cannot see.

We now articulate the implications of ruling out semantic extraction for pairing-based arguments. In these systems, a proof consists *only* of group elements, while witnesses are elements of the underlying field[6]. Soundness relies on the hardness of discrete logarithm in order to argue that the exponents of elements in the CRS remain hidden from the prover. As a concrete example, let us consider the CH NIZK argument that essentially compiles a Σ-protocol with three-round messages $([\mathbf{a}], e, \mathbf{d})$ into a NIZK argument in the CRS model in such a way that the CRS includes $[e]_2$ and the proof consists of two (vector of) group elements $([\mathbf{a}]_1, [\mathbf{d}]_2)$. Informally, the security relies on the fact that the prover cannot compute e (or $[e]_1$) and the second component $[\mathbf{d}]_2$ should have been computed as $[\mathbf{d}]_2 = \mathbf{d}_0[1]_2 + \mathbf{d}_1[e]_2$. But now, one can observe that from a *semantic* point of view, there is no distinction between the case that $[\mathbf{d}]_2$ is computed honestly as above and the case where the CRS trapdoor e is used for generating $[\mathbf{d}]_2$ as $\mathbf{d}_0[1]_2 + e[\mathbf{d}_1]_2$. In fact, if an extractor Ext that is limited to being *semantic* is able to extract the witness $\mathbf{d}_1$, then one can invoke Ext to break the discrete logarithm in $\mathbb{G}_2$ by sampling e in the reduction. The above reduction does not go through if Ext is a semantic extractor that has access to all the adversary random coins (we show that such Ext is equivalent to a classic white-box extractor). But as soon as some randomness is hidden from the extractor, we can define an adversary that embeds a DL challenge in this hidden part of the execution, for which no extractor can exist. This means that a valid proof in such argument systems does not prove "knowledge" of $\mathbf{w}$, but only knowledge of $[\mathbf{w}]_1, [\mathbf{w}]_2$, and in order to extract $\mathbf{w}$, one must rely on the hypothesis of asymmetric pairings to conclude that the prover actually knew $\mathbf{w}$ as a field element, which is essentially a knowledge-of-exponent type assumption.

Our results suggest that for most algebraic languages, extracting a witness given the statement $[\mathbf{x}]_1$ is as hard as extracting a witness given $[\mathbf{x}]_1$, a valid proof π together with used randomness r and trapdoor of the CRS e. Thus, if an extractor that is *not* based on knowledge assumption exists, it completely ignores the proof and just recomputes sampling a true statement together with its relative witness. This can also be seen in the following way: consider a language whose hardness relies on the hardness of discrete logarithm. Now, computing the witness from the statement is as hard as discrete logarithm; computing the witness given the statement, a proof, randomness used to compute the proof, and trapdoor is (in the case of CH20) as hard as symmetric discrete logarithm

[6] In structure preserving systems, the witness can be group elements as well, but in this work, we are only interested in proof systems where witnesses are field elements.

(SDL). This implies that either there is a gap between DL and SDL; or computing $\mathbf{w}$ from $[\mathbf{x}]_1$ is as hard as computing $\mathbf{w}$ from $([\mathbf{x}]_1, r, \pi, e)$. In the case of SPHF, both hardness of the language and our result rely on hardness of discrete logarithm, implying that computing $\mathbf{w}$ from $[\mathbf{x}]_1$ is as hard as computing $\mathbf{w}$ from $([\mathbf{x}]_1, \pi, r, \mathtt{td})$. This gives an explanation for why in the pairing-based setting, we have perfect soundness and f-extractability, like we show the CH proof is, while no fully extractable scheme exists under falsifiable assumptions.

2 Preliminaries

Notation. For any positive integer n, $[n]$ denotes the set $\{1, \ldots, n\}$. Let $k \in \mathbb{N}$ be the security parameter. Let $\mathsf{negl}(k)$ be an arbitrary negligible function. We write $a \approx_k b$ if $|a - b| \leq \mathsf{negl}(k)$. Moreover a is a negligible function if $a \approx_k 0$. When a function can be expressed in the form $1 - \mathsf{negl}(k)$, we say that it is overwhelming in k. We use DPT (resp. PPT) to mean a deterministic (resp. probabilistic) polynomial time algorithm. We write $Y \leftarrow \mathsf{F}(X)$ to denote an algorithm with input X and output Y. Further, we write $a \overset{\$}{\longleftarrow} S$ to denote that a is sampled according to distribution S, or uniformly randomly if S is a set. For two interactive machines $\mathcal{P}$ and $\mathcal{V}$, we denote by $\langle \mathcal{P}(\alpha), \mathcal{V}(\beta) \rangle(\gamma)$ the output of $\mathcal{V}$ after running on private input β with $\mathcal{P}$ using private input α, both having common input γ. All adversaries will be stateful. To represent matrices and vectors, we use bold upper-case and bold lower-case letters, respectively.

2.1 Bilinear Groups

We use additive notation for groups. Throughout the paper we let $\mathcal{G}$ be a bilinear group generator that on input security parameter k returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^k)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of prime order p, $[1]_1$ and $[1]_2$ are respectively the generators for $\mathbb{G}_1$ and $\mathbb{G}_2$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear map such that $\forall [u]_1 \in \mathbb{G}_1, \forall [v]_2 \in \mathbb{G}_2, \forall a, b \in \mathbb{Z}_p : \hat{e}(a[U]_1, b[V]_2) = (ab)\hat{e}([U]_1, [V]_2)$.

We denote $\hat{e}([U]_1, [V]_2)$ as $[U]_1[V]_2$. We consider only type III pairings, where there does not exist an efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

2.2 Algebraic Languages

We refer to algebraic languages as the set of languages associated to a relation that can be described by algebraic equations over an abelian group. More precisely, let $\mathtt{gpar} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^k)$. For the rest of the paper, we suppose that these global parameters $\mathtt{gpar}$ are implicitly given as input to each algorithm. Let $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ be a set of language parameters generated by a polynomial-time algorithm $\mathtt{setup.lpar}$ which takes $\mathtt{gpar}$ as input. Here, $\mathbf{M} : \mathbb{G}^\ell \mapsto \mathbb{G}^{n \times k}$ and $\boldsymbol{\theta} : \mathbb{G}^\ell \mapsto \mathbb{G}^n$ are linear maps such that their different coefficients are not necessarily in the same algebraic structures. Namely, in the most common case, given a bilinear group $\mathtt{gpar} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, they can belong to either $\mathbb{Z}_p$, $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$ as long as the equation $\boldsymbol{\theta}(\mathbf{x}) = \mathbf{M}(\mathbf{x}) \cdot \mathbf{w}$ is "well-consistent". However, in this paper we only use algebraic languages where the statement is defined as elements in $\mathbb{G}_1$. Formally, we define the algebraic language $\mathcal{L}_{\mathtt{lpar}} \subset \mathcal{X}_{\mathtt{lpar}}$ as

$$\mathcal{L}_{\mathtt{lpar}} = \left\{ [\mathbf{x}]_1 \in \mathbb{G}_1^{\ell} | \exists \mathbf{w} \in \mathbb{Z}_p^k : [\boldsymbol{\theta}(\mathbf{x})]_1 = [\mathbf{M}(\mathbf{x})]_1 \cdot \mathbf{w} \right\} \quad . \tag{1}$$

An algebraic language where $\mathbf{M}$ is independent of $\mathbf{x}$ and $\boldsymbol{\theta}$ is the identity is called a *linear language*. We sometimes require algebraic languages to satisfy a property we call 1DL-friendly. Roughly, this is to enable the embedding of a symmetric simple discrete logarithm challenge, which is given as a pair of group elements, into an algebraic statement in the reduction. We give the definition(Definition 13) in Appendix A.4. We note that algebraic languages are as expressive as NP, since every Boolean circuit can be represented by sets of linear equations.

2.3 Non-interactive Zero-knowledge Arguments

A NIZK (non-interactive zero-knowledge) argument Π, for a family of languages $\mathcal{L}_{\mathtt{lpar}}$ consists of four PPT algorithms.

- CRSGen on input a security parameter 1^k generates a pair $(\mathtt{crs}, \mathtt{td})$.
- $\mathcal{P}$ on input a $\mathtt{crs}$, a statement $\mathtt{x}$ and a witness $\mathtt{w}$, computes a proof π.
- $\mathcal{V}$ on input a $\mathtt{crs}$, a statement $\mathtt{x}$ and a proof π outputs 1 (accept) or 0 (reject).
- Sim on input $\mathtt{td}$, a true statement $\mathtt{x}$ computes a simulated proof π.

Here we are implicitly supposing that $\mathtt{lpar}$ is always given as input. We assume that each $\mathtt{td}$ corresponds to only one $\mathtt{crs}$ and also that given $\mathtt{td}$ it is possible to efficiently and deterministically compute the corresponding $\mathtt{crs}$. This is w.l.o.g., since it is always possible to define the trapdoor in a way that the previous property is satisfied. The following properties are required for a NIZK argument:

- *Perfect completeness*: for any pair of true statement $\mathtt{x}$ with a relative witness $\mathtt{w}$, for any $\mathtt{crs}$ computed by CRSGen
$$\Pr\left[\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 | \pi \leftarrow \mathcal{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w})\right] = 1.$$
- *Computational soundness*: for any PPT adversary $\mathcal{A}$
$$\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 & (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); \\ \wedge\ \mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}} & (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}) \end{array}\right] \leq \mathsf{negl}(k)$$
- *(Perfect) zero-knowledge*: for any true statement, witness pair $(\mathtt{x}, \mathtt{w})$, for any $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k)$ the following distributions are identical
$$\mathcal{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w}) \equiv \mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{x}).$$

If the zero-knowledge property requires the two distributions to only be computationally insitinguishable, then we get a computational NIZK. If soundness holds even against unbounded adversaries, we say that the protocol is a NIZK proof system, with perfect soundness. We say that Π is black-box knowledge sound if there exists an efficient extractor that computes a witness, given a statement, an accepting proof and the $\mathtt{crs}$ trapdoor.

Definition 1 (BB Knowledge soundness). *Let $\mathit{\Pi} = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}_{\mathtt{lpar}}$, defined by some language parameter* $\mathtt{lpar}$. *We say that $\mathit{\Pi}$ is black-box knowledge sound, if there exists an extractor* $\mathsf{Ext}_{\mathsf{bb}}$ *such that, for any PPT adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{l}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1\\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R}_{\mathtt{lpar}}\end{array}\middle|\begin{array}{c}(\mathtt{crs},\mathtt{td})\leftarrow\mathsf{CRSGen}(1^k);\\ (\mathtt{x},\pi)\leftarrow\mathcal{A}(\mathtt{crs},\mathtt{lpar};r);\mathtt{w}\leftarrow\mathsf{Ext}_{\mathsf{bb}}(\mathtt{td},\mathtt{x},\pi)\end{array}\right]\le\mathsf{negl}(k)$$

where r is the random coins of the adversary.

If the extractor is allowed to depend on the adversary and we also give it as additional input, the random coins used by the adversary, we say that $\mathit{\Pi}$ is white-box knowledge sound.

Definition 2 (n-BB Knowledge soundness). *Let $\mathit{\Pi} = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}_{\mathtt{lpar}}$, defined by some language parameter* $\mathtt{lpar}$. *We say that $\mathit{\Pi}$ is white-box knowledge sound, if for any PPT adversary $\mathcal{A}$, there exists an efficient extractor $\mathsf{Ext}_{\mathsf{wb},\mathcal{A}}$ such that:*

$$\Pr\left[\begin{array}{l}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1\\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R}_{\mathtt{lpar}}\end{array}\middle|\begin{array}{c}(\mathtt{crs},\mathtt{td})\leftarrow\mathsf{CRSGen}(1^k);\\ (\mathtt{x},\pi)\leftarrow\mathcal{A}(\mathtt{crs},\mathtt{lpar};r);\mathtt{w}\leftarrow\mathsf{Ext}_{\mathsf{wb},\mathcal{A}}(\mathtt{td},\mathtt{x},\pi,r)\end{array}\right]\le\mathsf{negl}(k)$$

where r is the random coins of $\mathcal{A}$.

We also consider the concrete security variants of the above definitions. Roughly, $\mathit{\Pi}$ is (t,ϵ)-BB knowledge sound if the extraction property holds with respect to all $t(k)$-time bounded provers (as opposed to all PPT provers), and that the extractor succeeds except with probability ϵ (as opposed to being negligible). We give the formal definitions of the concrete-security versions in Appendix D.2.

Lastly, we state the witness indistinguishability definition for non-interactive protocols. Recall that we are interested in non-interactive witness indistinguishable proof systems in the plain model without a trusted setup.

Definition 3 (Witness Indistinguishability (WI)). *A non-interactive proof system $\mathit{\Pi} = (\mathcal{P}, \mathcal{V})$ for language $\mathcal{L}_{\mathtt{lpar}}$ is WI if for every PPT verifier $(\mathcal{V}_1^*, \mathcal{V}_2^*)$, for all $(\mathtt{x}, \mathtt{w}_1, \mathtt{w}_2)$ such that $(\mathtt{x}, \mathtt{w}_1) \in \mathcal{R}_{\mathtt{lpar}}, (\mathtt{x}, \mathtt{w}_2) \in \mathcal{R}_{\mathtt{lpar}}$, we have*

$$\Pr\left[b\leftarrow\mathcal{V}_2^*(\mathsf{st},\pi)\middle|(\mathtt{x},\mathtt{w}_1,\mathtt{w}_2,\mathsf{st})\leftarrow\mathcal{V}_1^*(\mathtt{lpar});b\xleftarrow{\$}\{0,1\};\pi\leftarrow\mathcal{P}(\mathtt{lpar},\mathtt{x},\mathtt{w}_b)\right]\approx_k\frac{1}{2}$$

2.4 From $\mathit{\Sigma}$-protocols to NIZKs

Recently, Couteau and Hartmann [25] propose a new approach for building pairing-based non-interactive zero-knowledge arguments for algebraic languages. At a high level, their approach is based on compiling a $\mathit{\Sigma}$-protocol (see Appendix A.1) into a non-interactive zero-knowledge argument by embedding the challenge in $\mathbb{G}_2$ and publishing it once in the $\mathtt{crs}$. The NIZK argument is depicted in Fig. 2, where we denote as $\mathcal{S}_{\Sigma}$ the simulator for special honest verifier zero-knowledge property of the $\mathit{\Sigma}$-protocol. A variant of their compiler yields NIZK *proofs*, depicted in Fig. 3, based on standard assumptions. We refer to Appendix A.5 for more details.

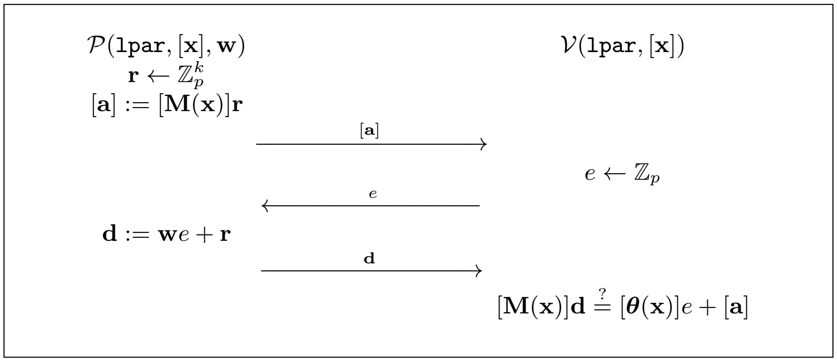
$\mathcal{P}(\mathtt{lpar}, [\mathbf{x}], \mathbf{w})$
$\mathbf{r} \leftarrow \mathbb{Z}_p^k$
$[\mathbf{a}] := [\mathbf{M}(\mathbf{x})]\mathbf{r}$
$\mathcal{V}(\mathtt{lpar}, [\mathbf{x}])$
$\xrightarrow{[\mathbf{a}]}$
$e \leftarrow \mathbb{Z}_p$
$\xleftarrow{e}$
$\mathbf{d} := \mathbf{w}e + \mathbf{r}$
$\xrightarrow{\mathbf{d}}$
$[\mathbf{M}(\mathbf{x})]\mathbf{d} \overset{?}{=} [\boldsymbol{\theta}(\mathbf{x})]e + [\mathbf{a}]$

Fig. 1: Σ-protocol for algebraic language $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$

```
$\mathsf{CRSGen}(1^k)$
    $\mathsf{gpar} \leftarrow \mathsf{setup.gpar}(1^k)$
    $e \leftarrow \mathbb{Z}_p$
    $\mathtt{crs} := (\mathsf{gpar}, [e]_2); \mathtt{td} := e$
    return $(\mathtt{crs}, \mathtt{td})$

$\mathcal{P}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \mathbf{w})$
    $\mathbf{r} \leftarrow \mathbb{Z}_p^k$
    $[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$
    $[\mathbf{d}]_2 := \mathbf{w}[e]_2 + [\mathbf{r}]_2$
    return $\pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$

$\mathsf{Sim}(\mathtt{lpar}, \mathtt{crs}, e, [\mathbf{x}]_1)$
    $([\mathbf{a}]_1, \mathbf{d}) := \mathcal{S}_\Sigma([\mathbf{x}]_1, e)$
    return $\pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$

$\mathcal{V}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$
    $[\mathbf{M}(\mathbf{x})]_1 \cdot [\mathbf{d}]_2 \overset{?}{=} [\boldsymbol{\theta}(\mathbf{x})]_1 \cdot [e]_2 + [\mathbf{a}]_1 \cdot [1]_2$
```

Fig. 2: NIZK argument for algebraic language $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ [25]

2.5 Cryptographic Assumptions

The DL (discrete logarithm) assumption in group $\mathbb{G}_\iota$ of order p states that it is hard to compute the discrete logarithm of a random element in $\mathbb{G}_\iota$.

Assumption 1 (Discrete logarithm assumption) *For any PPT adversary $\mathcal{A}$, it holds that:*

$$\Pr\left[\, w[1]_\iota = [x]_\iota \middle|\, w \leftarrow \mathcal{A}([1, x]_\iota) \right] \leq \mathsf{negl}(k)$$

where x is sampled from the uniform distribution over $\mathbb{Z}_p$.

Assumption 2 (Symmetric discrete logarithm assumption) *For any PPT adversary $\mathcal{A}$, it holds that:*

$$\Pr\left[\, w[1]_\iota = [x]_\iota \ ; \ \iota = 1, 2 \middle|\, w \leftarrow \mathcal{A}([1, x]_1, [1, x]_2) \right] \leq \mathsf{negl}(k)$$

where x is sampled from the uniform distribution over $\mathbb{Z}_p$.

```
$\mathsf{CRSGen}(1^k)$
    $s_1, s_2, e_1, e_2 \leftarrow \mathbb{Z}_p$
    $\mathtt{crs} := ([s_1, s_2, s_1 e_1, s_2 e_2]_2)$
    $\textbf{return } \mathtt{crs}$

$\mathcal{P}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \mathbf{w})$
    $\mathbf{r} \leftarrow \mathbb{Z}_p^k$
    $[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$
    $[\mathbf{d}_i]_2 := \mathbf{w}[s_i e_i]_2 + \mathbf{r}[s_i]_2$
    $\textbf{return } \pi := ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$

$\mathsf{Sim}(\mathtt{lpar}, [\mathbf{x}]_1)$
    $e, s_1, s_2 \leftarrow \mathbb{Z}_p$
    $([\mathbf{a}]_1, \mathbf{d}) := \mathcal{S}_{\Sigma}([\mathbf{x}]_1, e)$
    $\mathtt{crs} = ([s_1, s_2, s_1 e, s_2 e]_2)$
    $\pi := ([\mathbf{a}]_1, [\mathbf{d}s_1, \mathbf{d}s_2]_2)$
    $\textbf{return } (\mathtt{crs}, \pi)$

$\mathcal{V}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2))$
    $\textbf{for } i \in \{1, 2\} \text{ check}$
    $[\mathbf{M}(\mathbf{x})]_1 \cdot [\mathbf{d}_i]_2 \stackrel{?}{=} [\boldsymbol{\theta}(\mathbf{x})]_1 \cdot [s_i e_i]_2 + [\mathbf{a}]_1 \cdot [s_i]_2$
```

Fig. 3: NIZK proof for algebraic language $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ [25]

The co-CDH assumption was first proposed in [17]. Later a modified version of the assumption was proposed in [50] which we adapt as follows.

Assumption 3 (Computational co-Diffie-Hellman (co-CDH) assumption) *For any PPT adversary $\mathcal{A}$, it holds that:*

$$\Pr\left[\, [xy]_2 \leftarrow \mathcal{A}([1, x]_1, [1, x, y]_2) \,\right] \leq \mathsf{negl}(k)$$

where x, y are sampled from the uniform distribution over $\mathbb{Z}_p$.

3 NIWI Proof in the Plain Model

Our NIWI proof system in the plain model is given in Fig. 4. We show that our construction is perfectly sound and computationally WI. To show WI, we rely on a new assumption that we validate in the algebraic group model (AGM) in Appendix B.3. While it might seem like we can show WI by relying on DDH in the second group and then invoking the WI of the underlying sigma protocol, the presence of $[s_2]_1$ in the proof makes this impossible. In fact, we rely on DDH being easy for perfect soundness by enabling the verifier to check that the two challenges are indeed distinct. We show that the new assumption holds in the AGM introduced by Fuchsbauer, Kiltz and Loss [33]. The model is a relaxation of the generic group model [55] that captures adversaries exploiting the representation of the underlying group, and has been shown to be useful in reasoning about security properties of various constructions [46,34,23]. The work of [54] extends this model to handle decisional assumptions by introducing the notion of algebraic distinguishers. We use this model to show the algebraic equivalence

```
$\mathcal{P}(\texttt{lpar}, [\mathbf{x}]_1, \mathbf{w})$
$s_1, s_2, e_1, e_2 \leftarrow \mathbb{Z}_p$ s.t $e_1 \neq e_2$
$\mathbf{r} \leftarrow \mathbb{Z}_p^k$
$[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$
$\mathbf{d}_i := s_i e_i \mathbf{w} + s_i \mathbf{r}$ for $i = 1, 2$
return $\pi := \Big([\mathbf{a}, s_1, s_2]_1,$
$[s_1, s_2, s_1 e_1, s_2 e_2, \mathbf{d}_1, \mathbf{d}_2]_2\Big)$
```

```
$\mathcal{V}(\texttt{lpar}, [\mathbf{x}]_1, \pi)$
parse $\pi$ as $\Big([\mathbf{a}, c_1, c_2]_1, [s_1, s_2, E_1, E_2, \mathbf{d}_1, \mathbf{d}_2]_2\Big)$
accept if all the following checks pass
$[c_i]_1[1]_2 \overset{?}{=} [1]_1[s_i]_2$ for $i \in \{1, 2\}$        (1)
$[c_2]_1[E_1]_2 \overset{?}{\neq} [c_1]_1[E_2]_2$        (2)
for $i \in \{1, 2\}$ :
    $[\mathbf{M}(\mathbf{x})]_1[\mathbf{d}_i]_2 \overset{?}{=} [\boldsymbol{\theta}(\mathbf{x})]_1[E_i]_2 + [\mathbf{a}]_1[s_i]_2$        (3)
```

Fig. 4: NIWI proof for algebraic language $\mathcal{L}_{\texttt{lpar}}$ with $\texttt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$

```
$G_{\mathbf{ADHR},b}(\mathcal{A}, \texttt{lpar})$
$([\mathbf{x}]_1, \mathbf{w}_0, \mathbf{w}_1) \leftarrow \mathcal{A}([1]_1, [2]_2, \texttt{lpar})$;
$s_1, s_2, e_1, e_2 \leftarrow \mathbb{Z}_p; \mathbf{r} \leftarrow \mathbb{Z}_p^k; (e_1 \neq e_2)$;
$\pi = ([\mathbf{M}(\mathbf{x})\mathbf{r}, s_1, s_2]_1, [s_1, s_1 e_1, s_2, s_2 e_2, s_1 e_1 \mathbf{w}_b + s_1 \mathbf{r}, s_2 e_2 \mathbf{w}_b + s_2 \mathbf{r}]_2)$;
$b' \leftarrow \mathcal{A}([\mathbf{M}(\mathbf{x})]_1, \mathbf{w}_0, \mathbf{w}_1, \pi)$;
if $b = b'$ then return 1; else return 0 fi ;
```

Fig. 5: Algebraic decisional hidden range games $G_{\mathbf{ADHR},i}$.

between our assumption and *symmetric power discrete logarithm* (SPDL) assumption. While the assumption we make is a tautological assumption, we hope it will be analysed further and will find other applications, just like the tautological Kiltz-Wee assumption for QA-NIZK [45,3]. We believe it is an interesting open problem to prove the security of our construction under standard decisional assumptions.

Assumption 4 (Algebraic decisional hidden range) *Let* $\texttt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ *be any pair of language parameter that defines the algebraic language* $\mathcal{L}_{\texttt{lpar}}$. *Let* $G_{\boldsymbol{ADHR},i}$, *for* $i \in \{0, 1\}$ *be the games depicted in Fig. 5. The* $(\mathbf{M}, \boldsymbol{\theta})$-***ADHR*** *assumption states that for any PPT adversary* $\mathcal{A}$,

$$\mathbf{Adv}_{\mathcal{A},\texttt{lpar}}^{G_{\boldsymbol{ADHR},0,1}} = |\Pr[G_{\boldsymbol{ADHR},0}(\mathcal{A}, \texttt{lpar}) = 1] - \Pr[G_{\boldsymbol{ADHR},1}(\mathcal{A}, \texttt{lpar}) = 1]| \leq \mathsf{negl}(k).$$

Theorem 1. *For any algebraic language* $\mathcal{L}_{\texttt{lpar}}$, *with* $\texttt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$, *the protocol in Fig. 4 is a non-interactive witness indistinguishable proof under the* $(\mathbf{M}, \boldsymbol{\theta})$-***ADHR*** *assumption.*

Proof. (Perfect completeness). We show that an honest prover convinces an honest verifier with probability 1. For an honestly generated proof $\pi = ([\mathbf{a}, c_1, c_2]_1, [s_1, s_2, E_1, E_2, \mathbf{d}_1, \mathbf{d}_2]_2)$, by construction, we have that $c_i = s_i^{-1}$, $E_i = s_i e_i$ and $\mathbf{d}_i = s_i(e_i \mathbf{w} + \mathbf{r})$. It is easy to see that all the verifier checks pass.

1. $[c_i]_1[s_i]_2 = [s_i^{-1}]_1[s_i]_2 = [1]_T$.
2. $[c_1]_1[E_1]_2 = [s_1^{-1}]_1[s_1 e_1]_2 = [e_1]_T$, and $[c_2]_1[E_2]_2 = [s_2^{-1}]_1[s_2 e_2]_2 = [e_2]_T$, and since $e_1 \neq e_2$, we have $[c_1]_1[E_1]_2 \neq [c_2]_1[E_2]_2$.
3. $\mathbf{M}(\mathbf{x})\mathbf{d}_i = s_i e_i \mathbf{M}(\mathbf{x})\mathbf{w} + s_i \mathbf{M}(\mathbf{x})\mathbf{r} = E_i \boldsymbol{\theta}(\mathbf{x}) + \mathbf{a} s_i$.

(Perfect soundness). Let $\mathcal{A}$ be any (possibly unbounded) adversary that breaks the soundness property by outputting a proof $\tilde{\pi} = ([\tilde{a}, \tilde{c}_1, \tilde{c}_2]_1, [\tilde{s}_1, \tilde{s}_2, \tilde{E}_1, \tilde{E}_2, \tilde{d}_1, \tilde{d}_2]_2)$ relative to an (adaptively) chosen statement $\mathtt{x} = [\mathbf{x}]_1 \notin \mathcal{L}_{\mathtt{lpar}}$, such that the NIWI verifier accepts $\tilde{\pi}$. We show that such an accepting proof contradicts with the assumption that $\mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}$. In what follows, the index i will always be used as for each $i \in \{1, 2\}$.

From the verifier's check (1), it must be that $\tilde{c}_i = \tilde{s}_i$. Moreover, from check (3) we have that $\mathbf{M}(\mathbf{x})\tilde{d_i} = \boldsymbol{\theta}(\mathbf{x})\tilde{E_i} + \tilde{a}\tilde{s}_i$, which means that $\mathbf{M}(\mathbf{x})\tilde{d_i}/\tilde{c}_i = \boldsymbol{\theta}(\mathbf{x})\tilde{E_i}/\tilde{c}_i + \tilde{a}$. Now, since the NIWI verifier accepts the proof, from check (2), we have that $\tilde{c}_2\tilde{E}_1 \neq \tilde{c}_1\tilde{E}_2$. Therefore, there exists a pair of valid transcripts $([\tilde{a}]_1, \tilde{E_i}/\tilde{c}_i, \tilde{d_i}/\tilde{c}_i)$ for $\mathtt{x}$, with the same first message $[\tilde{a}]_1$ and different challenges. From special soundness of the underlying Σ-protocol, there exists an extractor that outputs a witness for $\mathtt{x}$ given two such transcripts. This contradicts the assumption that $\mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}$.

(Witness indistinguishability). Let $\mathcal{L}_{\mathtt{lpar}}$ be an algebraic language with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$. Let $\mathcal{A}$ be a PPT adversary that wins the WI game with non-negligible probability ϵ. We build an efficient adversary $\mathcal{B}$ against $(\mathbf{M}, \boldsymbol{\theta})$-**ADHR** assumption as follows: $\mathcal{B}$ first calls $\mathcal{A}$ and obtains $\mathsf{st} = ([\mathbf{x}]_1, \mathbf{w}_0, \mathbf{w}_1)$. It then outputs st and receives π from the challenger. Lastly, $\mathcal{B}$ calls $\mathcal{A}$ on π and returns $\mathcal{A}$'s decision bit. Since the challenger of $G_{\mathbf{ADHR},i}$ computes π exactly as the honest prover of the NIWI in Fig. 4, $\mathcal{B}$ breaks the assumption with the same non-negligible probability ϵ.

□

We discuss the efficiency of our construction and applications of NIWI in the plain model in Appendix B.

4 Partial Extractability for the CH Framework

In this section, we first recall the definition of f-extractability and show the NIZK proof system in Fig. 3 is $[\cdot]_2$-extractable. Next, we strengthen this property by introducing a new notion called *strong f-extractability* where the partial witness $\widetilde{\mathtt{w}}$ can be used by an efficient algorithm to decide membership of the statement. In more detail, here we also require the existence of an efficiently computable decision procedure D such that for $\widetilde{\mathtt{w}} = f(\mathtt{w})$ output by the extractor, $\mathsf{D}(\mathtt{x}, \widetilde{\mathtt{w}})$ decides membership of $\mathtt{x}$ (i.e., $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ iff $\mathsf{D}(\mathtt{x}, \widetilde{\mathtt{w}}) = 1$). However, $\widetilde{\mathtt{w}}$ falls short of being a witness for the relation; assuming that f is one-way, $\widetilde{\mathtt{w}}$ cannot be used to produce a valid proof for $\mathtt{x}$.

Definition 4 (f-extractability [10]). *Let $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}$, defined by some language parameter* $\mathtt{lpar}$.

Let f be an efficiently computable function. We say that Π is (black-box) f-extractable if there exists a PPT extractor Ext *such that for any PPT adversary that returns an accepting proof π for a statement* $\mathtt{x}$, Ext *outputs a value $\widetilde{\mathtt{w}}$ for which there exists some $\mathtt{w}$ such that $(\mathtt{x},\mathtt{w}) \in \mathcal{R}$ and $\widetilde{\mathtt{w}} = f(\mathtt{w})$ with overwhelming probability. More formally, for any PPT adversary $\mathcal{A}$, we have*

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi) = 1 \\ \wedge(\mathtt{x}, f^{-1}(\widetilde{\mathtt{w}})) \notin \mathcal{R} \end{array}\middle| \begin{array}{c} (\mathtt{crs},\mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); \\ (\mathtt{x},\pi) \leftarrow \mathcal{A}(\mathtt{crs},\mathtt{lpar}; r); \widetilde{\mathtt{w}} \leftarrow \mathsf{Ext}(\mathtt{td},\mathtt{x},\pi) \end{array}\right] \leq \mathsf{negl}(k)$$

where r is the random coins of the adversary.

We show that the CH proof system satisfies f-extractability where $f(x)$ is the encoding of x to $\mathbb{G}_2$. We state the lemma below and give the proof in Appendix C.

Lemma 1. *The NIZK proof system of [25] depicted in Fig. 3 is $[\cdot]_2$-extractable.*

4.1 Strong f-extractability

We now define strong f-extractability as an strengthening of f-extractability where the extracted value further allows to decide membership of the statement (although it cannot be used to produce a valid proof for it).

Definition 5 (Strong f-extractability). *Let $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}$, defined by some language parameter* $\mathtt{lpar}$. *Let f be an efficiently computable function. We say that Π is strong f-extractable if the following properties hold:*

Extractability. *Π is f-extractable (see Definition 4).*

Decidability. *There exists a DPT algorithm* D*, such that for any statement* $\mathtt{x}$ *and string $\widetilde{\mathtt{w}}$, it holds that $\mathsf{D}(\mathtt{x},\widetilde{\mathtt{w}}) = 1$ iff $(\mathtt{x},\mathtt{w}) \in \mathcal{R}$, where $\widetilde{\mathtt{w}} = f(\mathtt{w})$.*

One-wayness. *For any $(\mathtt{x},\widetilde{\mathtt{w}})$ sampled uniformly at random s.t $\mathsf{D}(\mathtt{x},\widetilde{\mathtt{w}}) = 1$, if there exists a PPT adversary $\mathcal{A}$ and a polynomial $p(\cdot)$, such that*

$$\Pr\left[\mathcal{V}(\mathtt{crs},\mathtt{x},\pi') = 1 \middle| \pi' \leftarrow \mathcal{A}(\mathtt{crs},\mathtt{x},\widetilde{\mathtt{w}})\right] \geq \frac{1}{p(k)},$$

there exists a PPT algorithm $\mathcal{I}$, and polynomial $q(\cdot)$ such that

$$\Pr\left[f(\bar{\mathtt{w}}) = \widetilde{\mathtt{w}} \middle| \bar{\mathtt{w}} \leftarrow I(\widetilde{\mathtt{w}})\right] \geq \frac{1}{q(k)}.$$

Remark 1. Similar to Definition 4, strong f-extractability is defined without any restriction on f and hence it can recover full extractability for the case when f is the identity function. However, we only focus on strong f-extractability for non-trivial f in this work. Having no restriction on f in Definitions 4 and 5 makes strong f-extractability a middle ground between full and f-extractability.

We show in Appendix C.2 that the proof system in Fig. 3 is strong $[\cdot]_2$-extractable under a standard hardness assumption. We remark that it is not clear whether the argument system in Fig. 2 satisfies strong f-extractability. Intuitively, if it did, then such an algorithm could likely be used to compute the witness $\mathbf{w}$ in the case of the underlying Σ-protocol, given only one transcript, which is impossible by SHVZK.

5 Full Extractability for the CH Framework

The CH argument system from Fig. 2 is knowledge sound in the AGM. (We show in Sec. D.1). Now, we turn to showing limitations of proving knowledge soundness.[7] We begin this section by defining a notion of knowledge soundness called *semantic extraction*. We study the relationship between semantic knowledge soundness and standard notions of black-box (BB) and white-box ($\mathsf{n\text{-}BB}$) knowledge soundness. Then, we show impossibility of the existence of semantic extractors for the CH argument system in Fig. 2. The generalization of this impossibility to quasi-adaptive NIZK arguments constructed from SPHFs is in Appendix D.4.

Notation. We introduce some additional notation for this section. We denote by $\mathbf{CRS}$ the set of all possible $\mathtt{crs}$'s. We denote by χ the set of the statements $\mathtt{x}$ and by Ψ the set of all possible proofs π We also split the randomness of PPT-s into two strings s and t. We denote by $\mathbf{\Gamma_t}$ the set of all possible strings t and by $\mathbf{\Gamma_s}$ the set of all possible strings s. Looking ahead, for adversarial provers, this split, at a high level, is to distinguish between the portion of randomness that is provided to the semantic extractor (t), and the portion that is not (s). Note that, while $\mathbf{CRS}, \chi, \Psi$ are defined by the NIZK construction, the randomness spaces are not fixed by the NIZK. We only assume that s, t have polynomial size.

5.1 Semantic Extractor

We now define our new notion of extraction. Informally, this extractor inverts the "semantic" function implemented by an adversarial prover regardless of *how* the computation was done. The key difference from $\mathsf{n\text{-}BB}$ notion is that we will not ask for a different extractor for every PPT $\mathcal{A}$, instead, we ask for an extractor associated with a function f; this extractor is universal for all TMs (even unbounded ones) that implement f. We begin by modeling the function implemented by a knowledge soundness adversary. To capture any possible adversarial strategy, we consider functions f and a distribution D from which random coins are sampled for a machine that implements f.

Definition 6 (Knowledge soundness strategy (KSS)). *Consider NIZK $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$. Let $f : \mathbf{CRS} \times \mathbf{\Gamma_s} \times \mathbf{\Gamma_t} \to \chi \times \Psi$ be a function, and D be the uniform distribution over $\mathbf{\Gamma_s} \times \mathbf{\Gamma_t}$. f is said to be a knowledge soundness strategy for Π if*

$$\Pr\left[\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s||t) \leftarrow D \\ \\ f(\mathtt{crs}; (s,t)) = (\mathtt{x}, \pi) \end{array}\right] = \eta(k)$$

[7] Recently, [5] instantiated AGM under falsifiable assumptions. However, their construction relies on indistinguishability obfuscation. It is inherently inefficient and not a practical group for applications. Here, we focus on feasibility of knowledge soundness of the CH framework as is in the standard model, without compromising on the efficiency.

where $\eta(k)$ is non-negligible. We say that a TM $\mathcal{A}$ implements the knowledge soundness strategy f, if for any $\mathtt{crs} \in \mathbf{CRS}$ *and* $(s,t) \leftarrow D$*, we have* $z \leftarrow \mathcal{A}(\mathtt{crs}; s, t)$*, where* $z = f(\mathtt{crs}, s, t)$*. If there exists a PPT $\mathcal{A}$ that implements a knowledge soundness strategy f, we say that f is efficiently implementable.*

We now define semantic knowledge soundness for a KSS.

Definition 7 (Semantic knowledge soundness). *Consider a NIZK argument* $\mathit{\Pi} = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$*. Let D be the uniform distribution over* $\mathbf{\Gamma_s} \times \mathbf{\Gamma_t}$*. We call $\mathit{\Pi}$ semantic knowledge sound if for every efficiently implementable KSS f, there exists a PPT extractor* $\mathsf{Ext} = \mathsf{Ext}_f$*, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f, we have*

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge (\mathtt{x}, \mathtt{w}) \notin \mathcal{R} \end{array} \middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s||t) \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; (s,t)); \boxed{\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, t)} \end{array}\right] \leq \mathsf{negl}(k)$$

Remark 2. We note that asking for extraction only against provers that implement a KSS is *not* a weakening of the extraction definition, since we only care about extracting from provers that make the verifier accept with non-negligible probability.

Remark 3. Note that this definition is a generalization of the usual knowledge soundness definitions. In particular, if we hide all the randomness from the extractor (that is $\mathbf{\Gamma_t}$ is the set that contains only the empty string), then we recover the usual black-box knowledge soundness. On the other hand, if we give the extractor all the randomness used by the adversary (that is $\mathbf{\Gamma_s}$ is the set that contains only the empty string), then we recover the canonical white-box knowledge soundness. We discuss these connections formally in Appendix D.2. We define semBB and $\mathsf{semn\text{-}BB}$ exactly as in Definition 7 with the boxed part replaced with $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)$, and $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, s||t)$ respectively.

Remark 4 (Canonical knowledge soundness adversary). The usual definition of knowledge soundness naturally handles the existence of an extractor for the honest prover. Our definition handles the case of the honest prover too; we show the honest efficiently implementable KSS for a NIZK $\mathit{\Pi}$ below:

1. Sample uniformly random strings $(s,t) \leftarrow \mathbf{\Gamma_s} \times \mathbf{\Gamma_t}$.
2. Sample a true statement $\mathtt{x}$ together with $\mathtt{w}$, from the uniform distribution over pair of $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, using random seed s. Note that this can be done efficiently. That is, there exists a PPT $\mathcal{A}$ that computes $(\mathtt{x}, \mathtt{w})$ on random coins s. Let us define the function $g : \mathbf{\Gamma_s} \to \chi \times \{0,1\}^*$ as $g(s) = (\mathtt{x}, \mathtt{w})$.
3. Run the honest prover algorithm on input $(\mathtt{crs}, \mathtt{x}, \mathtt{w})$ and random coins t, to compute a proof π. Define the function $g' : \mathbf{CRS} \times \chi \times \{0,1\}^* \times \mathbf{\Gamma_t} \to \mathit{\Psi}$ as $g'(\mathtt{crs}, \mathtt{x}, \mathtt{w}, t) = \mathcal{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w}; t)$.
4. Define $f : \mathbf{CRS} \times \mathbf{\Gamma_s} \times \mathbf{\Gamma_t} \to \chi \times \mathit{\Psi}$ as $f(\mathtt{crs}, (s,t)) = (\mathtt{x}, \pi)$ where $(\mathtt{x}, \mathtt{w}) = g(s)$ and $\pi = g'(\mathtt{crs}, \mathtt{x}, \mathtt{w}, t)$.

We call this f the canonical knowledge soundness strategy, and a PPT algorithm that implements it the *canonical adversary* of knowledge soundness.

We illustrate the meaningfulness of the new notion by showing relationships of semantic extraction with BB and $\mathsf{n\text{-}BB}$ extraction definitions in Appendix D.2. Here we point out that the notion of semantic extraction has been implicitly used in other works. For instance, standard Σ-protocols satisfy the semantic extraction notion. By special soundness, given a certain number of accepting transcripts for the same statement, and the same prover's first message, an extractor exists that outputs a valid witness. The extractor, therefore, does not depend on the prover's computation, instead, on a "semantic" function: one that outputs two different accepting transcripts relative to the same statement, and the same first message. One advantage in thinking of an extractor as a semantic one is the possibility to use it in a reduction, without its relative "native" adversary. This is indeed what is done in the proof of soundness for the NIZK proof of [25] described in Fig. 3, which is based on the existence of an (unbounded) TM that computes a valid input for the special soundness extractor, and then relying on the implicit semantic property of the latter.

The non-black-box nature of the semantic definition is limited to making non-black-box use of the malicious prover's randomness, but otherwise the prover's TM is treated as a black-box. There are instances in literature where a $\mathsf{n\text{-}BB}$ technique in fact corresponds to a semantic technique. Consider the case of simulation – Barak's non-black-box zero-knowledge protocol [7]. Though simulation is defined to make non-black-box use of the verifier's TM, it can be modified to only make non-black-box use of the auxiliary input and running time of the verifier, and not its TM. The property needed to define the simulator is the existence of an efficient (with bounded-length description) adversary. Then in the security proof, the next-message function implemented by the adversary is used, together with the ability to choose its random coins. This means that the security proof works for any adversary (even an unbounded one) that computes the same next-message function. Moreover, the zero-knowledge simulator for each of these adversaries would be exactly the same simulator as the one defined for the efficient adversary. For concreteness, we may think that, given the code of one efficient adversary, we define a simulator that works for each TM that computes the same function, in the sense that we use the code in a black-box way; by just fixing the random coins and taking partial outputs.

5.2 Impossibility of Semantic Knowledge Soundness for CH-NIZK

In this section we focus on semantic knowledge soundness of NIZK argument in Fig. 2 for a large and useful class of algebraic languages. We show in Appendix D.1 that when the adversary is algebraic, knowledge soundness holds in the AGM for this NIZK argument. We ask for knowledge soundness in the standard model, and show that CH NIZK argument cannot be semantic knowledge sound. The impossibility can be interpreted as an adversary explicitly violating AGM rules by hiding some exponent about the statement, and thus making the extractor fail. We refer to Remark 5, for more remarks on the interpretation of this result, while we focus on technical details for the rest of this section.

We now show the impossibility proof of semantic knowledge soundness of CH arguments for linear languages $\mathcal{L}_{\texttt{lpar}}$, where $\texttt{lpar} = [\mathbf{M}]_1$ is a constant matrix. The proof of Theorem 2 for general case of 1DL-friendly languages is deferred to Appendix D.3.

Lemma 2. *Let $\mathcal{L}_{\texttt{lpar}}$ be a linear language defined by constant matrix $\texttt{lpar} := [\mathbf{M}]_1$. The NIZK argument in Fig. 2 cannot be semantic knowledge sound for $\mathcal{L}_{\texttt{lpar}}$ under the SDL assumption.*

Proof. We denote as w_i components of the vector $\mathbf{w}$. The description of the canonical prover adversary on input ($\texttt{crs} = [e]_2$) and random coins (s, t), where $t = (\mathbf{r}, r')$ is given in Fig. 6a. Let Ext_f be the semantic extractor for the function $f([e]_2; (s, t)) = ([\mathbf{x}]_1, \pi)$, with $\pi = ([\mathbf{a}]_1, [\mathbf{d}]_2)$ that is implemented by the canonical prover adversary. By completeness of the NIZK argument, $\mathsf{Ext}_f(e, [\mathbf{x}, \mathbf{a}]_1, [\mathbf{d}]_2, t)$ outputs a valid witness $\mathbf{w}$ for $[\mathbf{x}]_1$ with overwhelming probability. Let us consider the (not polynomial-time) TM $\mathcal{P}^*$ as in Fig. 6b that implements f. $\mathcal{P}^*$ implements the same f of the canonical adversary and therefore its output can be used to feed the same extractor Ext_f.

We now exploit Ext_f to define an adversary $\mathcal{A}$ against SDL assumption. On input an SDL challenge $([w_1]_1, [w_1]_2)$, $\mathcal{A}$ is defined as in Fig. 6c. Since $\mathcal{A}$ computes inputs of Ext_f exactly as $\mathcal{P}^*$ does, they are correctly distributed, and hence $\mathcal{A}$ breaks SDL with the same probability that Ext_f succeeds.

Theorem 2. *Let $\mathcal{L}_{\texttt{lpar}}$ be a 1DL-friendly algebraic language (Definition 13) defined by language parameters $\texttt{lpar} := (\mathbf{M}, \boldsymbol{\theta})$. The NIZK argument in Fig. 2 cannot be semantic knowledge sound for $\mathcal{L}_{\texttt{lpar}}$ under the SDL assumption.*

Remark 5. Since our reduction exploits the knowledge of the trapdoor to compute a proof, (as a typical ZK simulator would do), it might seem like we are arguing about extracting from the simulator. However this is not the case, at least in general. We note that the procedure defined by the SDL adversary is very different from the zero-knowledge simulator. First, the adversary knows something that the simulator does not, which is $[\mathbf{x}]_2$. Moreover, the adversary is able to compute $[\mathbf{a}]_1$ before computing $[\mathbf{d}]_2$ as the honest prover; while the simulator, in order to compute a proof must compute $\mathbf{d}$ before. This can be also seen as the fact that the honest prover and simulator do not implement the same function. In fact, given the language parameter $\mathbf{M} \in \mathbb{Z}_p^{n \times m}$ the prover computes a proof π as a function of $\mathbf{x}, \mathbf{w}, r$ where $r \in \mathbb{Z}_p^{n \times 1}$, while the simulator computes a proof which is a function of random coins $r_{\mathsf{Sim}} \in \mathbb{Z}_p^{m \times 1}$. In order to invoke the semantic extractor associated to the honest prover, we must have a function that defines a relation between the two randomness. This, for instance, can be done (inefficiently) only in some particular cases, like when $\mathbf{M}$ is a square invertible matrix. Finally, the existence of such cases is evidence towards the impossibility of extraction. In fact, given the latter case, since we have perfect zero-knowledge for a relation that defines only true statement, given a proof from the NIZK

```
1. Sample uniformly random $w_1$ using random coins $s$.
2. Sample other components of $\mathbf{w}$, $w_i$ for $i \neq 1$, using random coins $r'$.
3. Compute $[\mathbf{x}]_1 = [\mathbf{M}]_1\mathbf{w}$.
4. Compute $[\mathbf{a}]_1 = [\mathbf{M}]_1\mathbf{r}$.
5. Compute $[\mathbf{d}]_2 = \mathbf{w}[e]_2 + \mathbf{r}[1]_2$.
6. Output $([\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$.
```

(a) Canonical prover adversary

```
1. Use random coins $s$ to sample $w_1$ and compute $[w_1]_1, [w_1]_2$.
2. Compute (inefficiently) $e$ from $[e]_2$.
3. Sample other components of $\mathbf{w}$ using random coins $r'$.
4. Compute $[\mathbf{x}]_1 = [\mathbf{Mw}]_1$.
5. Compute $[\mathbf{a}]_1 = [\mathbf{Mr}]_1$.
6. Compute $[\mathbf{d}]_2 = e[\mathbf{w}]_2 + \mathbf{r}[1]_2$.
```

(b) Unbounded adversary

```
1. Sample $e, \mathbf{r}, r'$.
2. Sample other components of $\mathbf{w}$, $w_i$ for $i \neq 1$, using random coins $r'$.
3. Compute $[\mathbf{x}]_1 = [\mathbf{Mw}]_1$.
4. Compute $[\mathbf{a}]_1 = [\mathbf{Mr}]_1$.
5. Compute $[\mathbf{d}]_2 = e[\mathbf{w}]_2 + \mathbf{r}[1]_2$.
6. Compute $\mathbf{w} \leftarrow \mathsf{Ext}_f(([\mathbf{a}]_1, [\mathbf{d}]_2), e, [\mathbf{x}]_1, (\mathbf{r}, r'))$.
7. Output $w_1$.
```

(c) SDL adversary

Fig. 6: Procedures for Lemma 2

argument, it is impossible to distinguish the case when the prover was honest, from the case when a powerful adversary just computes the discrete logarithm of the CRS and runs the simulator. Furthermore, it is impossible to distinguish the case that adversary had $[\mathbf{w}]_2$ and the trapdoor e, instead of $\mathbf{w}$ without relying on knowledge-type assumptions.

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Disjunctions for hash proof systems: New constructions and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 69–100. Springer, Heidelberg (Apr 2015)
2. Abdolmaleki, B., Baghery, K., Lipmaa, H., Zajac, M.: A subversion-resistant SNARK. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part III. LNCS, vol. 10626, pp. 3–33. Springer, Heidelberg (Dec 2017)
3. Abdolmaleki, B., Lipmaa, H., Siim, J., Zajac, M.: On QA-NIZK in the BPK model. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part I. LNCS, vol. 12110, pp. 590–620. Springer, Heidelberg (May 2020)
4. Acar, T., Nguyen, L.: Revocation for delegatable anonymous credentials. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 423–440. Springer, Heidelberg (Mar 2011)

5. Agrikola, T., Hofheinz, D., Kastner, J.: On instantiating the algebraic group model from falsifiable assumptions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 96–126. Springer, Heidelberg (May 2020)
6. Ananth, P., Asharov, G., Dahari, H., Goyal, V.: Towards accountability in crs generation. IACR Eurocrypt 2021, `https://eprint.iacr.org/2021/1090.pdf`
7. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd FOCS. pp. 106–115. IEEE Computer Society Press (Oct 2001)
8. Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 299–315. Springer, Heidelberg, Santa Barbara, USA (Aug 17–21, 2003)
9. Bauer, B., Fuchsbauer, G., Loss, J.: A classification of computational assumptions in the algebraic group model. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 121–151. Springer, Heidelberg (Aug 2020)
10. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and noninteractive anonymous credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (Mar 2008)
11. Ben Hamouda-Guichoux, F.: Diverse Modules and Zero-Knowledge. Ph.D. thesis, PSL Research University (2016)
12. Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 37–56. Springer, Heidelberg (Aug 1990)
13. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHFs and efficient one-round PAKE protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 449–475. Springer, Heidelberg (Aug 2013)
14. Benhamouda, F., Pointcheval, D.: Trapdoor Smooth Projective Hash Functions. Tech. Rep. 2013/341, IACR (Jun 3 2013), available at `http://eprint.iacr.org/2013/341`, last retrieved version from 27 Aug 2013
15. Bitansky, N.: Verifiable random functions from non-interactive witness-indistinguishable proofs. Cryptology ePrint Archive, Report 2017/018 (2017), `http://eprint.iacr.org/2017/018`
16. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and Its Applications. In: STOC 1988. pp. 103–112. ACM Press, Chicago, Illinois, USA (May 2–4, 1988)
17. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (May 2003)
18. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J., Petit, C.: Short accountable ring signatures based on DDH. Cryptology ePrint Archive, Report 2015/643 (2015), `http://eprint.iacr.org/2015/643`
19. Boyen, X.: The uber-assumption family (invited talk). In: Galbraith, S.D., Paterson, K.G. (eds.) PAIRING 2008. LNCS, vol. 5209, pp. 39–56. Springer, Heidelberg (Sep 2008)
20. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (May 2001)
21. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (Aug 1997)

22. Campanelli, M., Fiore, D., Querol, A.: LegoSNARK: Modular design and composition of succinct zero-knowledge proofs. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2075–2092. ACM Press (Nov 2019)
23. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, N., Ward, N.P.: Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 738–768. Springer, Heidelberg (May 2020)
24. Chung, K.M., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 66–92. Springer, Heidelberg (Mar 2015)
25. Couteau, G., Hartmann, D.: Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 768–798. Springer, Heidelberg (Aug 2020)
26. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (Aug 1994)
27. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (Apr / May 2002)
28. Dwork, C., Naor, M.: Zaps and their applications. In: 41st FOCS. pp. 283–293. IEEE Computer Society Press (Nov 2000)
29. Faonio, A., Fiore, D., Herranz, J., Ràfols, C.: Structure-preserving and rerandomizable RCCA-secure public key encryption and its applications. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 159–190. Springer, Heidelberg (Dec 2019)
30. Feige, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. In: Aho, A. (ed.) 19th ACM STOC. pp. 210–217. ACM Press (May 1987)
31. Fortnow, L.: The complexity of perfect zero-knowledge (extended abstract). In: Aho, A. (ed.) 19th ACM STOC. pp. 204–209. ACM Press (May 1987)
32. Freund, Y., Schapire, R.E.: Adaptive game playing using multiplicative weights. Games and Economic Behavior 29(1-2), 79–103 (1999)
33. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Heidelberg (Aug 2018)
34. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953 (2019), https://eprint.iacr.org/2019/953
35. Garg, S., Ostrovsky, R., Visconti, I., Wadia, A.: Resettable statistical zero knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 494–511. Springer, Heidelberg (Mar 2012)
36. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (Jun 2011)
37. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In: 27th FOCS. pp. 174–187. IEEE Computer Society Press (Oct 1986)
38. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press (May 1987)

39. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-Systems. In: Sedgewick, R. (ed.) STOC 1985. pp. 291–304. ACM Press, Providence, Rhode Island, USA (May 6–8, 1985)
40. Green, M., Hohenberger, S.: Practical adaptive oblivious transfer from simple assumptions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 347–363. Springer, Heidelberg (Mar 2011)
41. Groth, J., Ostrovsky, R., Sahai, A.: New Techniques for Noninteractive Zero-Knowledge. Journal of the ACM 59(3) (2012)
42. 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 (Apr 2008)
43. Izabachène, M., Libert, B., Vergnaud, D.: Block-wise P-signatures and non-interactive anonymous credentials with efficient attributes. In: Chen, L. (ed.) 13th IMA International Conference on Cryptography and Coding. LNCS, vol. 7089, pp. 431–450. Springer, Heidelberg (Dec 2011)
44. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (Dec 2013)
45. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (Apr 2015)
46. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2111–2128. ACM Press (Nov 2019)
47. Meiklejohn, S.: An extension of the groth-sahai proof system (2009)
48. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (Aug 2003)
49. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. pp. 427–437. ACM Press (May 1990)
50. Ng, T., Tan, S., Chin, J.: A variant of BLS signature scheme with tight security reduction. In: Mobile Networks and Management - 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings. pp. 150–163 (2017), https://doi.org/10.1007/978-3-319-90775-8_13
51. Pass, R.: Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 334–354. Springer, Heidelberg (Mar 2013)
52. Ràfols, C.: Stretching groth-sahai: NIZK proofs of partial satisfiability. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 247–276. Springer, Heidelberg (Mar 2015)
53. Rial, A., Kohlweiss, M., Preneel, B.: Universally composable adaptive priced oblivious transfer. In: Shacham, H., Waters, B. (eds.) PAIRING 2009. LNCS, vol. 5671, pp. 231–247. Springer, Heidelberg (Aug 2009)
54. Rotem, L., Segev, G.: Algebraic distinguishers: From discrete logarithms to decisional uber assumptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part III. LNCS, vol. 12552, pp. 366–389. Springer, Heidelberg (Nov 2020)
55. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (May 1997)

Supporting Material

A Additional Preliminaries

A.1 Σ-Protocols

A Σ-protocol is a public-coin three-round interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. A Σ-protocol should satisfy completeness, special soundness, and special honest verifier zero-knowledge (SHVZK), defined as follows:

Definition 8 (Completeness). *A Σ-protocol is complete for relation $\mathcal{R}$, if for any PPT adversary $\mathcal{A}$, and any honest $\mathcal{P}$ and $\mathcal{V}$,*

$$\Pr\left[\langle \mathcal{P}(\mathtt{w}), \mathcal{V}\rangle(\mathtt{x}) = 1 \vee\ (\mathtt{x}, \mathtt{w}) \notin \mathcal{R} \middle| (\mathtt{x}, \mathtt{w}) \leftarrow \mathcal{A}(1^k)\right] = 1$$

Definition 9 (Special Soundness). *A Σ-protocol for a relation $\mathcal{R}$ is special sound, if there exists a PPT algorithm* Ext *that given a statement* $\mathtt{x}$ *and two accepting transcripts $(a, e, d), (a, e', d')$ with the same first message and $e \neq e'$ outputs a witness* $\mathtt{w}$*, such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ with overwhelming probability.*

Definition 10 (Special Honest-Verifier Zero-Knowledge (SHVZK)). *A Σ-protocol for a relation $\mathcal{R}$ is SHVZK, if there exists a PPT simulator* Sim *such that for $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ and $e \in \{0,1\}^k$, the distributions of* $\mathsf{Sim}(\mathtt{x}, e)$ *is identical to the distribution of the 3-move honest transcript obtained when $\mathcal{V}$ sends e as challenge and $\mathcal{P}$ runs on common input* $\mathtt{x}$ *and private input* $\mathtt{w}$ *such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$.*

For the sake of completeness, we also recall the definition of witness indistinguishability for Σ-protocols. As shown in [26], every Σ-protocol that enjoys Completeness, Special Soundness and perfect Honest Verifier Zero Knowledge (HVZK) is perfect WI.

Definition 11 (Witness Indistinguishability (WI)). *A Σ-protocol for a relation $\mathcal{R}$ is perfect WI [8] if for every malicious verifier $\mathcal{V}^*$, for all $\mathsf{st} = (\mathtt{x}, \mathtt{w}_1, \mathtt{w}_2)$ such that $(\mathtt{x}, \mathtt{w}_1) \in \mathcal{R}, (\mathtt{x}, \mathtt{w}_2) \in \mathcal{R}$, we have*

$$\Pr\left[\langle \mathcal{P}(\mathtt{w}_1, 1^k), \mathcal{V}^*(\mathsf{st})\rangle(\mathtt{x}) = 1\right] = \Pr\left[\langle \mathcal{P}(\mathtt{w}_2, 1^k), \mathcal{V}^*(\mathsf{st})\rangle(\mathtt{x}) = 1\right]$$

[8] WI is used to mean both “witness indistinguishability” and “witness indistinguishable”.

A.2 Witness Sampleable (WS) Languages

For a witness sampleable language $\mathcal{L}$, the language parameters come together with a trapdoor which allows to check whether $\mathtt{x} \in \mathcal{L}$. In this case, we suppose that $\mathsf{setup}.\mathtt{lpar}$ also outputs a (language) trapdoor $\mathtt{ltrap}$ associated with $\mathtt{lpar}$ and allows to decide whether a given $\mathtt{x} \in \mathcal{X}$ is in $\mathcal{L}$ or not. It is easy to see that for linear languages, this trapdoor is the exponents of all matrix entries. We refer to [25] for formal definition and more details of WS languages.

A.3 Smooth Projective Hash Function (SPHF)

A SPHF is defined as follows (cf. [13]).

Definition 12. *A SPHF for* $\{\mathcal{L}_{\mathtt{lpar}}\}$ *is a tuple of PPT algorithms* $(\mathsf{setup}, \mathsf{hashkg}, \mathsf{projkg}, \mathsf{hash}, \mathsf{projhash})$*, which are defined as follows:*

$\mathsf{setup}(1^k)$**:** *Takes a security parameter k and generates the global parameters* pp *together with the language parameters* $\mathtt{lpar}$ *(we assume that all algorithms have access to* pp*).*

$\mathsf{hashkg}(\mathtt{lpar})$**:** *Takes a language parameter* $\mathtt{lpar}$ *and outputs a hashing key* hk*.*

$\mathsf{projkg}(\mathtt{lpar};\mathsf{hk},\mathtt{x})$**:** *Takes a hashing key* hk*,* $\mathtt{lpar}$*, and a statement* $\mathtt{x}$ *and outputs a projection key* hp*, possibly depending on* $\mathtt{x}$*.*

$\mathsf{hash}(\mathtt{lpar};\mathsf{hk},\mathtt{x})$**:** *Takes a hashing key* hk*,* $\mathtt{lpar}$*, and a statement* $\mathtt{x}$ *and outputs a hash value* H*.*

$\mathsf{projhash}(\mathtt{lpar};\mathsf{hp},\mathtt{x},\mathtt{w})$**:** *Takes a projection key* hp*,* $\mathtt{lpar}$*, a statement* $\mathtt{x}$*, and a witness* $\mathtt{w}$ *for* $\mathtt{x} \in \mathcal{L}$ *and outputs a hash value* pH*.*

A SPHF needs to satisfy the following properties:

Correctness. It is required that $\mathsf{hash}(\mathtt{lpar};\mathsf{hk},\mathtt{x}) = \mathsf{projhash}(\mathtt{lpar};\mathsf{hp},\mathtt{x},\mathtt{w})$ for all $\mathtt{x} \in \mathcal{L}$ and their corresponding witnesses $\mathtt{w}$.

Smoothness. It is required that for any $\mathtt{lpar}$ and any $\mathtt{x} \notin \mathcal{L}$, the following distributions are statistically indistinguishable:

$$\{(\mathsf{hp},\mathsf{H}) : \mathsf{hk} \leftarrow \mathsf{hashkg}(\mathtt{lpar}), \mathsf{hp} \leftarrow \mathsf{projkg}(\mathtt{lpar};\mathsf{hk},\mathtt{x}), \mathsf{H} \leftarrow \mathsf{hash}(\mathtt{lpar};\mathsf{hk},\mathtt{x})\}$$
$$\{(\mathsf{hp},\mathsf{H}) : \mathsf{hk} \leftarrow \mathsf{hashkg}(\mathtt{lpar}), \mathsf{hp} \leftarrow \mathsf{projkg}(\mathtt{lpar};\mathsf{hk},\mathtt{x}), \mathsf{H} \leftarrow \Omega\} \quad .$$

where Ω is the set of hash values.

A.4 Construction of SPHF from Diverse Vector Space

A diverse vector space (DVS) [13,1,11] is a representation of a language $\mathcal{L} \subseteq \mathcal{X}$ as a subspace $\hat{\mathcal{L}}$ of some vector space. Let $\mathcal{R} = \{(\mathtt{x},\mathtt{w})\}$ be a relation with $\mathcal{L} = \{\mathtt{x} : \exists \mathtt{w}, (\mathtt{x},\mathtt{w}) \in \mathcal{R}\}$. Let pp be system parameters, including say the description of a bilinear group. A (pairing-based) DVS $\mathcal{V}$ is defined as $\mathcal{V} = (\mathsf{pp}, \mathcal{X}, \mathcal{L}, \mathcal{R}, n, k, \mathbf{M}, \boldsymbol{\theta}, \boldsymbol{\lambda})$, where $\mathbf{M}(\mathtt{x})$ is an $n \times k$ matrix, $\boldsymbol{\theta}(\mathtt{x})$ is an n-dimensional vector, and $\boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$ a k-dimensional vector. The matrix $\mathbf{M}(\mathtt{x})$ can depend on $\mathbf{x}$ (in this case, it is called GL-DVS) or not (KV-DVS). Moreover, different coefficients of $\boldsymbol{\theta}(\mathtt{x})$, $\mathbf{M}(\mathtt{x})$, and $\boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$ can belong to different algebraic structures as long as the equation $\boldsymbol{\theta}(\mathtt{x}) = \mathbf{M}(\mathtt{x}) \cdot \boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$ is well-consistent. In the most common case, this means that given a bilinear group $\mathsf{pp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g_1, g_2)$, they belong to either $\mathbb{Z}_p$, $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$ as long as the consistency of the above equation is preserved.

A DVS $\mathcal{V}$ satisfies the following properties [11]:

- *coordinate-independence of groups:* the group in which each coordinate of $\boldsymbol{\theta}(\mathtt{x})$ lies is independent of $\mathtt{x}$.
- *perfect completeness:* for any $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, $\boldsymbol{\theta}(\mathtt{x}) = \mathbf{M}(\mathtt{x}) \cdot \boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$.
- *statistical ε-soundness:* $\forall \mathtt{x} \in \mathcal{X} \setminus \mathcal{L}$, $\Pr[\boldsymbol{\theta}(\mathtt{x}) \in \mathrm{colspace}(\mathbf{M}(\mathtt{x}))] \leq \varepsilon$.

In this work, we only deal with DVSs where $\boldsymbol{\lambda}$ is the identity function. I.e., $\boldsymbol{\lambda}(\mathtt{x}, \mathtt{w}) = \mathtt{w}$. Given a GL/KV-DVS for $\mathcal{L}$, one can construct an efficient GL/KV-SPHF for $\mathtt{x}' \in \mathcal{L}$, where $\mathtt{w} = \mathtt{w}'$ and $\mathtt{x} = [\boldsymbol{\theta}(\mathtt{x}')]_\iota = [\mathbf{M}(\mathtt{x}')]_\iota \mathtt{w}'$ [13], see Fig. 7. Here, the only possible nonlinear operation is the dependency of $\boldsymbol{\theta}$ and $\mathbf{M}$ on the actual input $\mathtt{x}'$. It is known that if $\mathcal{V}$ is a 0-sound GL-DVS/KV-DVS, then the PHF in Fig. 7 is a perfectly smooth GL/KV-SPHF, see Theorem 3.1.11 in [11].

- $\mathsf{hashkg}(\mathtt{lpar})$: sample $\vec{\alpha} \leftarrow \mathbb{Z}_p^n$, and output $\mathsf{hk} \leftarrow \vec{\alpha}$;
- $\mathsf{projkg}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x} = [\boldsymbol{\theta}(\mathtt{x}')]_\iota)$: $[\boldsymbol{\gamma}]_\iota^\top \leftarrow \vec{\alpha}^\top [\mathbf{M}(\mathtt{x}')]_\iota \in \mathbb{G}_\iota^{1 \times k}$; return $\mathsf{hp} \leftarrow [\boldsymbol{\gamma}]_\iota$;
- $\mathsf{hash}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x})$: return $\mathsf{H} \leftarrow \vec{\alpha}^\top [\boldsymbol{\theta}(\mathtt{x}')]_\iota$;
- $\mathsf{projhash}(\mathtt{lpar}; \mathsf{hp}, \mathtt{x}, \mathtt{w} = \mathtt{w}')$: return $\mathsf{pH} \leftarrow [\boldsymbol{\gamma}]_\iota^\top \mathtt{w}'$;

Fig. 7: DVS-based SPHF construction for $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$.

We sometimes require algebraic languages to satisfy a property we call 1DL-friendly. The reason we need this property is to enable the embedding of a symmetric simple discrete logarithm challenge, which is given as a pair of group elements, into an algebraic statement in the reduction. We give the definition below.

Definition 13. *An algebraic languages is* 1*DL-friendly if given a uniformly random element, $c \leftarrow \mathbb{Z}_p$, there exist a tuple of functions λ_x, λ_w such that the following procedure can be used to generate a pair of true statement $[\mathbf{x}]_1$ with a relative witness $\mathbf{w}$.*

- *Define $w_1 = c$ and sample uniformly random $w_2, \ldots, w_d$ (independently from w_1). Compute $\mathbf{w} = \lambda_w(w_1, \ldots, w_n)$. We restrict here to functions λ_w that are affine in w_1.*

- *Compute* $[\mathbf{x}]_1 = \lambda_x(w_1, \dots, w_n)$*, such that* $\mathbf{M}(\mathbf{x})\mathbf{w} = \boldsymbol{\theta}(\mathbf{x})$*. Again, we restrict to functions* λ_x *that are affine in* w_1*.*

Practically, given group elements $[x]_1, [x]_2$, we implicitly define $x = w_1$ and sample $w_2, \dots, w_n \leftarrow \mathbb{Z}_p$. Then, we compute $[\mathbf{x}]_1 = \lambda_x([x]_1, w_2, \dots, w_n)$ and a $\mathbb{G}_2$-encoding of the relative witness $[\mathbf{w}]_2 = \lambda_w([x]_2, w_2, \dots, w_n)$, which are efficiently computable because λ_x, λ_w are required to be affine as a function of w_1. We remark that this condition is only assumed for simplicity, in order to state our formal theorem under simple assumptions. However, our framework could in principle, work for any hard algebraic language, at the cost of using more structured assumptions.

A.5 From Σ-protocols to NIZKs (Extended)

The work of [25] proposed a framework for compiling a Σ-protocol for algebraic languages into a non-interactive zero-knowledge argument by embedding the challenge in $\mathbb{G}_2$ and publishing it once in the `crs`. The soundness of the compiled NIZK is based on a new family of assumptions *extended-kernel Matrix Diffie-Hellman* (**extKerMDH**) that are not necessarily falsifiable [9].

Couteau and Hartmann [25] also showed how to achieve perfect soundness by making use of the unconditional special soundness of the Σ-protocol. More precisely, they proved that the compiled protocol in Fig. 3 is a NIZK proof with computational zero-knowledge if the DDH assumption holds in $\mathbb{G}_2$, and the underlying Σ-protocol is complete, special sound and SHVZK.

We remark that there is no efficient extractor to compute the witness in the latter proof system. In fact the existence of a witness is guaranteed by the special soundness of the underlying Σ-protocol, however, to be able to extract it, we need an unbounded extractor to compute the exponent of group elements. To be more precise, an efficient extractor can compute, in the best case, only exponentiations of the witness in either $\mathbb{G}_1$ or $\mathbb{G}_2$ as shown in Section 4. It is worth mentioning that the soundness proof is based on the existence of this unbounded extractor, to compute a pair of proofs of the underlying Σ-protocol. More precisely, given a valid proof for a false statement and under an honestly generated CRS, we can (inefficiently) compute the field elements $(s_1, e_1, s_2, e_2, d_1, d_2)$ and output two valid proofs for the underlying Σ-protocol with the same first message and different challenges (with overwhelming probability). This contradicts the special soundness property, which states that two such proofs cannot exist for a false statement.

A.6 Algebraic Group Model

Algebraic algorithms. We recall that AGM essentially states that for every efficient algorithm $\mathcal{A}$ that outputs the vector $[\mathbf{y}]_\iota$ of group elements in $\mathbb{G}_\iota$ when given inputs the vector $[\mathbf{x}]_\iota$ of group elements in $\mathbb{G}_\iota$, there exists an efficient

[9] Although the assumption is falsifiable for all witness-sampleable languages (A.2).

extractor $\mathsf{Ext}_{\mathcal{A}}$ that returns a matrix $\mathbf{A}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. In particular, since we are working in the setting of asymmetric bilinear pairings, we require that any outputs in one group must depend only on the inputs it receives in that group.

Algebraic Distinguishers. Here, we briefly recall the notion of algebraic distinguishers and refer the reader to [54] for more details.

A distinguisher is an algorithm that aims to distinguish between 2 games. Particularly, we consider adversaries $\mathcal{A}$ that engage in games with challengers, parametrized by a bit $b \in \{0, 1\}$. We refer to G_b as the game where the bit b is chosen. A distinguisher $\mathcal{A}$ aims to detect if it is playing the game G_0, or G_1. At the end of its interaction with the challenger, it outputs a decisional bit b'. $\mathcal{A}$ wins the game if $b = b'$. Let us denote by $\mathbf{View}_{\mathcal{A}}^{G_b}$ the random variable that describes the view of $\mathcal{A}$ in the game G_b (that is the input it received so far and the internal random tape). Moreover, let $[x_0, \ldots, x_{n_1}]_1, [y_0, \ldots, y_{n_2}]_2$ be $\mathcal{A}$'s input, with $x_0 = y_0 = 1$, and let $\vec{w}$ be a vector indexed by two indices $i \in \{0, \ldots, n_1\}, j \in \{0, \ldots, n_2\}$ such that the component w_{ij} is naturally associated to the pairing of inputs $[x_i]_1$ and $[y_j]_1$, i.e., $[x_i]_1[y_j]_2 = [x_i y_j]_T$. We indicate with $\left[\mathbf{View}_{\mathcal{A}}^{G_b}\right]_{supp(\vec{w})}$ the random variable that is defined by the view $\mathcal{A}$ in the game G_b omitting all group elements whose corresponding entry in $\vec{w}$ is 0. A distinguisher $\mathcal{A}$ participating in an algebraic game G_b, is said to be algebraic if there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that computes a vector $\vec{w}$ that explains the decision in an algebraic way, at least with a certain probability.

Definition 14 (Algebraic distinguisher). *A distinguisher $\mathcal{A}$ participating in an algebraic game G_b is said to be algebraic if there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that computes a vector of field element $\vec{w}$ such that the following condition holds.*

1. $\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_{ij} [x_i y_j]_T = [0]_T$.
2. *Let t be an upper bound over the running time of $\mathcal{A}$ and ϵ be the probability that $\mathcal{A}$ successfully distinguishes between G_0 and G_1. Then*

$$\Pr\left[\left[\mathbf{View}_{\mathcal{A}}^{G_0}\right]_{supp(\vec{w})} \neq \left[\mathbf{View}_{\mathcal{A}}^{G_1}\right]_{supp(\vec{w})}\right] \geq \epsilon/t^2,$$

where the inequality is intended as distributions and the probability is over the choice of $\vec{w}$ induced by a random execution of $G_b(\mathcal{A})$ and $\mathsf{Ext}_{\mathcal{A}}$.

B NIWI Proof in the Plain Model

B.1 Efficiency of Our NIWI Proof

We give an informal comparison between the Groth-Sahai NIWI [41] in the plain model and our NIWI in Fig. 4.

Recall that Groth-Sahai techniques [41] to construct NIWI in the plain model consists of sending two distinct Groth-Sahai proofs, along with two different $\texttt{crs}$-s chosen by the prover. This results in communication complexity that is two times the size of proof plus the $\texttt{crs}$. With our technique, a NIWI proof has communication complexity of one CH proof, on top of 6 group elements that are sent as the "crs" that the prover chooses. As noted in [25], proofs in the CH framework has the same size as optimized Groth-Sahai proofs, for many languages of interest, such as disjunctions of linear languages. Hence our NIWI proof, in these cases, has better communication complexity compared to Groth-Sahai NIWI [41].

When proving statements where the statement is augmented with intermediate commitments, our resulting statement size is much shorter, since we only need to commit in $\mathbb{G}_1$, while usually one needs to commit in both groups with GS. This results in better communication complexity in scenarios where we embed a circuit satisfiability problem as an algebraic language and commitments are part of the statement that are sent along with the proof.

Finally, we point out that our NIWI construction is the first that achieves constant overhead for communication and computational complexity (with respect to the language size), compared to the corresponding NIZK proof in the CRS model.

B.2 Applications

There are several works [15,35,6] which show how one can make use of NIWI in the plain model to construct more complex cryptographic primitives. Bitansky et al. [15] showed how to construct verifiable random functions and verifiable function commitment schemes using NIWI in the plain model. In [35], Garg et al. introduced the notion of *Efficiently Extractable Non-Interactive Instance-Dependent Commitment Scheme* and constructed a two-round resettable statistical witness-indistinguishable argument for languages that have such type of commitments. The key idea in their construction is to make use of a NIWI proof system in the plain model to ensure that verifier's challenge in the first round of the argument is well-formed. The fact that the verifier's challenge is a commitment to a random message indicates that the NIWI language is "natively" algebraic, and hence our NIWI can be used to improve the efficiency of the resulting argument in [35], wherein the NIWI is instantiated with [41].

The recent work of Ananth et al. [6] which provides a notion of accountability towards the CRS generation authority employs a NIWI proof system. To this end, the authority is required to include some valid transcript in the CRS and since he is the one who generates the CRS, the idea of using a NIZK proof does not work. The authority instead proves a statement about the transcript using a NIWI proof. In more detail, the authority provides four commitments $(\mathsf{cm} = (\mathsf{cm}_0, \mathsf{cm}_1), \overline{\mathsf{cm}} = (\overline{\mathsf{cm}}_0, \overline{\mathsf{cm}}_1))$ and uses a NIWI in the plain model to prove that one of cm or $\overline{\mathsf{cm}}$ are commitments to both bits 0 and 1. Interestingly, the NIWI language corresponding to the statements defined by the commitments is again natively algebraic, for which our NIWI is suitable.

B.3 New Computational Assumption and AGM Proof of Security

Assumption 5 (Symmetric power discrete logarithm (SPDL)) *Let q_1, q_2 be two integers. The (q_1, q_2)-**SPDL** assumption holds if for any PPT adversary $\mathcal{A}$,*

$$\Pr\left[\, y^* = y \middle|\, y^* \leftarrow \mathcal{A}([1, y, y^2, \ldots, y^{q_1}]_1, [1, y, y^2, \ldots, y^{q_2}]_2) \,\right] \leq \mathsf{negl}(k)$$

where y is sampled from the uniform distribution over $\mathbb{Z}_p$.

Following the framework of [54], we prove the security of our new assumption in the AGM. We start by restating the new assumption.

Assumption 6 (Algebraic decisional hidden range) *Let $G_{\boldsymbol{ADHR},i}$, for $i \in \{0, 1\}$ be the games depicted in Fig. 5. Let $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ be any pair of language parameter that defines the algebraic language $\mathcal{L}_{\mathtt{lpar}}$. The $(\mathbf{M}, \boldsymbol{\theta})$-**ADHR** assumption states that for any PPT adversary $\mathcal{A}$,*

$$\mathbf{Adv}_{\mathcal{A},\mathtt{lpar}}^{G_{\boldsymbol{ADHR},0,1}} = |\Pr[G_{\boldsymbol{ADHR},0}(\mathcal{A}, \mathtt{lpar}) = 1] - \Pr[G_{\boldsymbol{ADHR},1}(\mathcal{A}, \mathtt{lpar}) = 1]| \leq \mathsf{negl}(k).$$

Theorem 3. *If the $(1, 2)$-**SPDL** holds, then for any PPT algebraic distinguisher $\mathcal{A}$, it holds that*

$$\mathbf{Adv}_{\mathcal{A},\mathtt{lpar}}^{G_{\boldsymbol{ADHR},0,1}} \leq \mathsf{negl}(k)$$

for any $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ that defines the algebraic language $\mathcal{L}_{\mathtt{lpar}}$.

Proof. Let us first consider the case that $\mathbf{M}$ is of dimension $d \times 1$ for any $d > 0$, so $\mathbf{r}$ is a single element. Let $\mathcal{A}$ be an algebraic PPT distinguisher for the games depicted in Fig. 5. In Fig. 8 we show how to exploit $\mathcal{A}$ in order to define an adversary $\mathcal{B}$ to $(1, 2)$-SPDL problem. The reduction proceeds as follows: $\mathcal{B}$ first picks some language parameter $\mathtt{lpar}$ and then runs the first stage of $\mathcal{A}$ in order to obtain $(\mathbf{x}, \mathbf{w}_0, \mathbf{w}_1)$. Note that since $\mathcal{B}$ knows $\mathtt{lpar}$ as field elements, $\mathcal{A}$ is algebraic and $\mathcal{A}$ receives as input only the generators, we can assume that $\mathcal{B}$ knows $\mathbf{x}$ as field elements. Next, $\mathcal{B}$ samples some uniformly random elements $(u_1, u_2, u_r, t_1, t_2, t_r)$ to embed the challenge as elements $u_i y + t_i$. This is a standard procedure frequently used to embed a univariate challenge in a multivariate polynomial [33,9]. Note that elements u_1, u_2, u_r and y are perfectly hidden to $\mathcal{A}$ as they are "one-time padded" with t_i-s. This property will be used later in the proof. Then, $\mathcal{B}$ samples uniformly random trapdoors e_1, e_2, computes π and then runs the second phase of $\mathcal{A}$ in order to obtain the distinguisher bit b'. Note that $\mathcal{B}$ needs $[Y^2]_2$ in order to compute elements of the form $[s_i\mathbf{r}]_2$.

```
$\mathcal{B}^{\mathcal{A}}_{2\text{-}SPDL}([y]_1, [y, y^2]_2)$
Fix any $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$;
$(\mathbf{x}, \mathbf{w}_0, \mathbf{w}_1) \leftarrow \mathcal{A}([1]_1, [1]_2, \mathtt{lpar})$;
$b \leftarrow \{0, 1\}$;
$u_r, u_1, u_2, t_r, t_1, t_2 \leftarrow \mathbb{Z}_p$;
if $u_r[y]_1 + t_r[1]_1 = [0]_1,$ then return $-t_r/u_r;$ fi ; (1)
if $\exists i \in \{1, 2\}$ such that $u_i[y]_1 + t_i[1]_1 = [0]_1,$ then return $-t_i/u_i;$ fi ; (2)
$e_1, e_2 \leftarrow \mathbb{Z}_p; (e_1 \neq e_2)$
for $\iota \in \{1, 2\}$   $[\mathbf{r}]_\iota = u_r[y]_\iota + t_r[1]_\iota$;
  for $i \in \{1, 2\}$     $[s_i]_\iota = u_r[y]_\iota + t_r[1]_\iota$;
    $[s_i\mathbf{r}]_2 = u_i u_r [y^2]_2 + (u_i t_r + u_r t_i)[y]_2 + t_i t_r [1]_2$;
    $[d_i]_2 = e_i \mathbf{w}_b [s_i]_2 + [s_i\mathbf{r}]_2$;
    if $[d_i]_2 = [0]_2$ then return $-(t_r + e_i \mathbf{w}_b)/u_r;$ fi ; (3)
  endfor
endfor $(*)$
$\pi = ([\mathbf{M}(\mathbf{x})\mathbf{r}, s_1, s_2]_1, [s_1, s_1 e_1, s_2, s_2 e_2, d_1, d_2]_2)$;
$b' \leftarrow \mathcal{A}([\mathbf{M}(\mathbf{x})]_1, \mathbf{w}_0, \mathbf{w}_1, \pi)$;
  $(\vec{\phi}, \vec{\psi}, \vec{\sigma}) \leftarrow \mathsf{Ext}_{\mathcal{A}}$;
  Find all the roots of the univariate polynomials $V(Y) = V(u_1 Y + t_1, u_2 Y + t_2, e_1, e_2, u_r Y + t_r, \mathbf{x})$;
  Check if one of the roots $y^*$ is equal to $y$; If yes return $y^*$ else return $\perp$;
```

Fig. 8: SPDL reduction for the new assumption. Polynomials Φ, Σ, Ψ are as defined in Eq. (2) and polynomial V is as defined in Eq. (3).

Let us define the polynomials

$$\begin{aligned}
\Phi(\vec{S}, \vec{E}) &= \sum_{i=1}^{2} [\phi_{0i} S_i + \phi_{1i} S_i E_i] + \sum_{i,j=1}^{2} [\phi_{2ij} S_i S_j + \phi_{3ij} S_i S_j E_j]; \\
\Sigma(R, X) &= \sigma_0 + \sigma_1 \mathbf{M}(X) R. \\
\Psi(\vec{S}, \vec{E}, R, X) &= \sum_{i=1}^{2} [\psi_{0i}(S_i E_i \mathbf{w} + S_i R) + \psi_{1i} \mathbf{M}(X) R S_i + \psi_{2i} \mathbf{M}(X) R S_i E_i] \\
&\quad + \sum_{i=1}^{2} [\psi_{3i} \mathbf{M}(X) R (S_i E_i \mathbf{w} + S_i R)] + \sum_{i,j=1}^{2} [\psi_{4ij} S_j (S_i E_i \mathbf{w} + S_i R)];
\end{aligned} \tag{2}$$

Since $\mathcal{A}$ is supposed to be an algebraic distinguisher, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that computes coefficients $(\vec{\psi}, \vec{\phi}, \vec{\sigma})$ such that the following verification polynomial $V(\vec{S},\vec{E},R,X) = \Phi(\vec{S},\vec{E}) + \Psi(\vec{S},\vec{E},R,X) + \Sigma(R,X)$ is 0 when evaluated in the point defined by $\mathcal{A}$'s inputs. That is,

$$V(\vec{s},\vec{e},\mathbf{r},\mathbf{x}) = \Phi(\vec{s},\vec{e}) + \Psi(\vec{s},\vec{e},\mathbf{r},\mathbf{x}) + \Sigma(\mathbf{r},\mathbf{x}) = 0. \qquad (3)$$

It is easy to see that V is the polynomial taking all the possible pairings among $\mathcal{A}$'s inputs. As shown in Fig. 8, $\mathcal{B}$ invokes $\mathsf{Ext}_{\mathcal{A}}$ to compute coefficients $(\vec{\phi},\vec{\psi},\vec{\sigma})$ of V.

Recall that by definition of algebraic distinguisher, with high probability, V must have a number of non-zero coefficients, such that the view of $\mathcal{A}$, when restricted to the input corresponding to the non-zero monomials, is distributed differently in the two games. Particularly, in our case this implies that V must explicitly depend on the used witness $\mathbf{w}_b$. Note that the monomials of V in which $\mathbf{w}_b$ is multiplied by $\mathbf{M}(\mathbf{x})$ are the same in both games, since $\mathbf{M}(\mathbf{x})\mathbf{w}_0 = \mathbf{M}(\mathbf{x})\mathbf{w}_1 = \boldsymbol{\theta}(\mathbf{x})$. Thus $[\mathbf{View}_{\mathcal{A}}^{\mathbf{w}_b}]_{supp(\vec{\phi},\vec{\psi},\vec{\sigma})}$ for $b \in \{0,1\}$ are distributed differently if and only if V has a non-zero coefficient that corresponds to a monomial in which $\mathbf{w}_b$ but not $\mathbf{M}(\mathbf{x})$ appears. Formally, let ϵ be the advantage of $\mathcal{A}$ in distinguishing the two distributions, and t be the running time of $\mathcal{A}$. Let Hit be the event that V explicitly depends on the used witness $\mathbf{w}_b$. Then $\Pr[\mathsf{Hit}] \geq \epsilon/t^2$ by the definition of algebraic distinguishers.

We first observe that $\mathcal{B}$ stops before the point labeled as $(*)$ with negligible probability. This can be concluded by the fact that $(u_1, u_2, u_r, t_1, t_2, t_r)$ and e_1, e_2 are sampled from uniform distributions which implies that the elements $e_i\mathbf{w}_b$ are distributed uniformly at random too.

We now show that $\Pr[\mathcal{B} \text{ wins}] \geq \mathsf{negl}(k) + \Pr[\mathsf{Hit}]$. Note that the variable R appears in Ψ only multiplied by at least one of the variables S_1, S_2. Suppose that $\Psi(\vec{S},\vec{e},R,\mathbf{x})$ is a polynomial of degree at least 1 in R. So, there exists a non-zero element $\tilde{\psi}$ of $\vec{\psi}$ that corresponds to a monomial in which the variable R appears. Let $S_1^z S_2^q RJ(e_1,e_2,\mathbf{x})$, for $z,q \in \{0,1,2\}$ and some J, be this monomial. Since Σ is independent from S_i and Φ is independent from R, then $V(S_1,S_2,\vec{e},R,\mathbf{x}) = \tilde{\psi}S_1^z S_2^q RJ(e_1,e_2,\mathbf{x}) + P(S_1,S_2,\vec{e},R,\mathbf{x})$ where P is a trivariate polynomial that does not contain a monomial of the type $S_1^z S_2^q R$. Thus, $V(S_1,S_2,\vec{e},R,\mathbf{x})$ is also a non-zero polynomial of degree at least 2. Suppose now that $\Psi(\vec{S},\vec{e},R,\mathbf{x})$ is of degree 0 in R. This can happen if and only if $\Psi = -\sum_{i=1}^{2}\mathbf{M}(\mathbf{x})\mathbf{w}\psi_{1i}S_iE_i$, ($\psi_{0i} = -\psi_{1i}$, and other coefficients of Ψ are equal to 0). Thus Ψ, and also V are independent from $\mathbf{w}$ and the view of $\mathcal{A}$ is the same in the two games. Note in fact that $\mathbf{M}(S_iE_i\mathbf{w}) = \boldsymbol{\theta}(\mathbf{x})S_iE_i$ for each valid witness. By definition of Hit, this cannot happen. Thus, we have shown that, conditioned on the event Hit, V is a non-zero polynomial of (total) degree at least 2.

We now recall a lemma from [9] that we use in our proof.

Lemma 3 ([9]). *Let $V(X_1,...,X_m)$ be a non-zero multivariate polynomial in $\mathbb{Z}_p$ of total degree d. For each vectors $\vec{u},\vec{t}$ of length m, define $V(Y)$ as $V(Y) = P(u_1Y + t_1, \ldots, u_mY + t_m)$. Then the coefficient of maximal degree of Q is a polynomial in $u_1, \ldots, u_m$ of degree d.*

By applying this lemma, we have that the coefficient of the term with maximal degree in $V(Y) = V(u_1Y + t_1, u_2Y + t_2, e_1, e_2, u_rY + t_r, \mathbf{x})$ is polynomial in

u_1, u_2, u_r of degree at least 2. Let $v(u_1, u_2, u_r)$ be this term. Since u_1, u_2, u_r are perfectly hidden to $\mathcal{A}$, the probability that $v(u_1, u_2, u_r) = 0$ is negligible based on the Schwartz-Zippel lemma.

Summing up, we have $V(y) = 0$ and, conditioned on Hit, $V(Y) \neq 0$ as a polynomial, except with negligible probability. This shows that $\Pr[\mathcal{B} \text{ wins}] \geq \mathsf{negl}(k) + \Pr[\mathsf{Hit}] - \mathsf{negl}(k) \geq \mathsf{negl}(k) + \epsilon/t^2$.

Thus, $\epsilon \leq t^2 \Pr[\mathcal{B} \text{ wins}] + \mathsf{negl}(k)$. The fact that $\Pr[\mathcal{B} \text{ wins}]$ is negligible by assumption concludes the proof for the case of $(d \times 1)$-dimension $\mathbf{M}$.

What is left is to generalize the proof to the case where $\mathbf{M}$ is a $n \times k$ matrix. In this case, $\mathcal{B}$ will sample k different and independently chosen uniformly random u_{ir}, t_{ir} and define each value of $\mathbf{r}$ as $u_{ir}Y + t_{ir}$. Then, instead of having just one verification polynomial V, we have k verification polynomials $\{V_i\}_{i \in [k]}$, one for each line of $\mathbf{M}$. By the definition of AGM distinguisher, at least one of these polynomials, say V_i, must explicitly depend on $\mathbf{w}_b$. Applying the same procedure to V_i as described above completes the proof. □

C Partial Extractability of CH Framework

C.1 f-extractability of CH Proof systems

We show that the CH NIZK proof system satisfies f extractability where $f(x)$ is the encoding of x to $\mathbb{G}_2$.

Lemma 4 (Lemma 1 restated). *The NIZK proof system of [25] depicted in Fig. 3 is $[\cdot]_2$-extractable.*

Proof. We show the existence of an efficient extractor Ext that given a trapdoor $\mathtt{td}$ and a valid proof π for any statement $[\mathbf{x}]_1$, outputs partial witness $\widetilde{\mathbf{w}}$. Let $\mathtt{td} = (e_1, e_2, s_1, s_2)$. By relying on the soundness of the NIZK proof and the fact that a valid proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$ must satisfy the verification equations, Ext computes a partial witness $\widetilde{\mathbf{w}}$ as follows:

- $[\mathbf{d}'_i]_2 := [\mathbf{d}_i]_2 s_i^{-1} = \mathbf{w}[e_i]_2 + \mathbf{r}[1]_2$
- $\vec{u} = [\mathbf{d}'_1]_2 - [\mathbf{d}'_2]_2$
- **return** $\widetilde{\mathbf{w}} = \vec{u}(e_1 - e_2)^{-1}$

It is easy to see that $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$ with probability 1.

C.2 Strong Partial Extractability of CH proof systems

To ease exposition we first show the proof for linear languages and then prove the general case for any 1DL-friendly language.

Lemma 5. *Let $\mathcal{L}_{\mathtt{lpar}}$ be any linear language defined by $\mathtt{lpar} = [\mathbf{M}]_1$. Assuming that co-CDH problem is hard, the NIZK proof system in Fig. 3 for $\mathcal{L}_{\mathtt{lpar}}$ is strong $[\cdot]_2$-extractable.*

Proof. From Lemma 1, we have that the proof system is $[\cdot]_2$-extractable. This means that for any adversarially generated $([\mathbf{x}]_1, \pi)$ which passes the verification, the extractor can extract $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$. To prove that it satisfies decidability property, we define the algorithm D as follows: $\mathsf{D}([\mathbf{x}]_1, \widetilde{\mathbf{w}} = [\mathbf{w}]_2)$ returns 1 if $[\mathbf{M}]_1[\mathbf{w}]_2 = [\mathbf{x}]_1[1]_2$. It is clear that D is efficient. To show how D decides the membership of $[\mathbf{x}]_1$, note that the pairing equality holds iff $\mathbf{w} = f^{-1}([\mathbf{w}]_2)$ (for $f(x) := [x]_2$) is a valid witness for $[\mathbf{x}]_1$, i.e., $([\mathbf{x}]_1, \mathbf{w}) \in \mathcal{R}_{\mathtt{lpar}}$. We now argue that, compute $\mathbf{w} = f^{-1}([\mathbf{w}]_2)$, is as hard as computing a valid proof π' for $[\mathbf{x}]_1$ given $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$. Clearly, computing $\mathbf{w} = f^{-1}([\mathbf{w}]_2)$ is hard given the hardness of discrete logarithm in $\mathbb{G}_2$. We now show the hardness of computing a valid proof, given a partial witness $\widetilde{\mathbf{w}}$, by a reduction to the co-CDH problem. Recall that co-CDH problem asks to compute $[XY]_2$, given $([1, X, Y]_2) \in \mathbb{G}_2$ and $([1, X]_1) \in \mathbb{G}_1$ as input.

Consider the linear language $\mathcal{L}_{\mathtt{lpar}}$, defined by $\mathtt{lpar} = [\mathbf{M}]_1$, where $\mathbf{M} = (m_{ij}) \in \mathbb{Z}_p^{n\times k}$. W.l.o.g we can assume that the first entry of $\mathbf{M}$ (i.e., m_{11}) is non-zero [10]. Let $\mathcal{A}$ be an efficient algorithm that on input $(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \widetilde{\mathbf{w}})$ computes a valid proof π with non-negligible probability ϵ. We construct an efficient algorithm $\mathcal{B}$ against co-CDH problem so that on input challenge $([1, X]_1, [1, X, Y]_2)$ proceeds as follows:

- Generate the CRS parameters by sampling $s_1, s_2, e_1 \leftarrow \mathbb{Z}_p$ and set $e_2 = [Y]_2$. Let $\mathtt{crs} = ([s_1, s_2, s_1e_1, s_2e_2]_2)$. It is clear that the distribution of $\mathtt{crs}$ is the same as an honestly generated CRS.
- Define $[w_1]_1 = [X]_1$ and sample uniformly random elements $w_2, \ldots, w_k \leftarrow \mathbb{Z}_p$. Let $[\mathbf{w}]_1 = [w_1, \ldots, w_k]_1$ and compute $[\mathbf{x}]_1 = \mathbf{M}[\mathbf{w}]_1$. Compute also $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$, where $[w_1]_2 = [X]_2$ is from the challenge.
- Run $\mathcal{A}$ on input $(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \widetilde{\mathbf{w}})$ to obtain a proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$.
- Check if π makes the verifier accepts; and abort otherwise.
- Let u be the first entry of the vector $(s_1\mathbf{M}[\mathbf{d}_2]_2 + s_1s_2e_1[\mathbf{x}]_2 - s_2\mathbf{M}[\mathbf{d}_1]_2)/(s_1s_2)$. Return $([u]_2 - (\sum_{i=2}^{d} m_{1i}w_i)[e_2]_2)/m_{11}$.

To see that the output is $[XY]_2$, we note that, if the verifier accepts, then $\mathbf{M}\mathbf{d}_i = \mathbf{x}s_ie_i + s_i\mathbf{a}$ for $i \in 1, 2$. Thus, we have $\mathbf{M}\mathbf{w}Y = \mathbf{x}Y = \mathbf{x}e_2 = (s_1\mathbf{M}\mathbf{d}_2 + s_1s_2e_1\mathbf{x} - s_2\mathbf{M}\mathbf{d}_1)/(s_1s_2)$. To complete the proof we note that, since $X = \mathbf{w}_1$, the first entry of $\mathbf{M}\mathbf{w}Y$ is equal to $m_{11}XY + (\sum_{i=2}^{d} m_{1i}w_i)Y$. This shows that $\mathcal{B}$ returns $[XY]_2$ with at least the same probability ϵ that $\mathcal{A}$ computes a valid proof given only $\widetilde{\mathbf{w}}$ as the witness. □

Strong partial extractability for 1DL-friendly languages. Fix any $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ such that the defined algebraic language is 1DL-friendly. Thus, there exists two affine functions λ_x, λ_w such that, $\mathbf{M}(\lambda_x(X))\lambda_w(X) = \boldsymbol{\theta}(\lambda_x(X))$. Here, with a little abuse of notation, we implicitly assume that $X = w_1, w_2, ..., w_n$ are fixed values and $\lambda_x(X)$ indicates $\lambda_x(X, w_2, ..., w_n)$. Same for λ_w. Since the

[10] This is without loss of generality since columns of $\mathbf{M}$ can be assumed to be linearly independent.

composition of a linear and an affine map is still affine, we have that each entry of $(\mathbf{M}(\lambda_x(X)))_{ij}$ is defined by an affine function $m_{1ij}X + m_{0ij}$. Moreover, each entry of $(\lambda_w(X))_j$ is defined by an affine function $w_{1j}X + w_{0j}$. Note that each $w_2, ..., w_n$ corresponds to a set of coefficients $m_{1ij}, m_{0ij}, w_{1j}, w_{0j}$; and viceversa, i.e., to each set of coefficients, at least one choice of $w_2, ..., w_n$ is corresponded. Thus, we can assume that the reduction below knows $w_2, ..., w_n$. Given, any $e \in \mathbb{Z}_p$, for any $i \in \{1, ..., n\}$ let us define the polynomial

$$g_{iT}(X, Y) = X^2Y\sum_j(m_{1ij}w_{1j}) + XY\sum_j(m_{1ij}w_{0j} + m_{0ij}w_{1j}) - X^2e\sum_j(m_{1ij}w_{1j}) \tag{4}$$

Note that this polynomial is not in the subspace generated by the base $\{1, X, X^2, Y, XY\}$, as long as $\sum_j(m_{1ij}w_{1j}) \neq 0$. Following the framework of Uber-assumptions (see [19]) we define the following assumption.

Assumption 7 *Let g_{iT} be any polynomial as defined in Eq. (4), such that $\sum_j(m_{1ij}w_{1j}) \neq 0$. For any PPT adversary $\mathcal{A}$ it holds that:*

$$\Pr\left[\, t = g_{iT}(x, y) \middle| x, y \leftarrow \mathbb{Z}_p; [t]_T \leftarrow \mathcal{A}([1, x]_1, [1, x, y]_2) \,\right] \leq \mathsf{negl}(k)$$

Lemma 6. *If Assumption 7 holds, then the NIZK proof system in [25], depicted in Fig. 3 for any* 1*DL-friendly language, is strong* $[\cdot]_2$*-extractable.*

Proof. The first part of the proof is the same as in Lemma 5. We just need to show that, given $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$, computing a proof is hard. We prove the hardness under Assumption 7.

Consider any g_{it} defined by the choice of any 1DL-friendly language and any $e \in \mathbb{Z}_p$. Suppose that $\mathcal{A}$ can efficiently compute a valid proof with high probability, having on input $(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \widetilde{\mathbf{w}} = [\mathbf{w}]_2)$. We show how to define an efficient adversary $\mathcal{B}$ to compute $[g_{it}(x, y)]_T$, given the challenge $([1, x]_1, [1, x, y]_2)$. $\mathcal{B}$ is defined as follows.

- Let e be as in Eq. (4). If $e = y$ then compute $[g_{iT}(x, y)]_T$. Note that in this case $[g_{iT}(x, y)]_T$ can be easily computed having $([1, x, xy]_1, [1, x, y]_2)$.
- Else, generate the CRS parameters by sampling $s_1, s_2 \leftarrow \mathbb{Z}_p$ and set $[e_1]_2 = [y]_2, e_2 = e$. Let $\mathtt{crs} = ([s_1, s_2, s_1e_1, s_2e_2]_2)$. It is clear that the distribution of $\mathtt{crs}$ is the same as one that is generated honestly.
- Define $[\mathbf{x}]_1 = [\lambda_x(x)]_1, [\mathbf{w}]_1 = [\lambda_w(x)]_1, [\mathbf{w}]_2 = [\lambda_w(x)]_2$. Note that $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$.
- Run $\mathcal{A}$ on input $([1, \mathbf{x}]_1, [1, s_1, s_2, s_1e_1, s_2e_2, \mathbf{w}]_2)$ to obtain a proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$.
- Check if π makes the verifier accepts, otherwise $\mathcal{B}$ abort.
- Compute $[\delta^{\mathbf{d}}]_2 = (s_2/s_1)[\mathbf{d}_1]_2 - [\mathbf{d}_2]_2$.
- Let $[out]_T$ be

$$\begin{aligned}[x]_1(\sum_j(m_{1ij}[\delta^{\mathbf{d}}_j]_2 + s_2em_{1ij}w_{0ij}[1]_2 + s_2em_{0ij}w_{1ij}[1]_2))\\ +[-1]_1(\sum_j s_2m_{0ij}w_{0j}[y]_2) + [1]_1(\sum_j(m_{0ij}[\delta^{\mathbf{d}}_j]_2 + s_2em_{0ij}w_{0ij}[1]_2)).\end{aligned}$$

Output $(1/s_2)[out]_T$.

To see that $\mathcal{B}$'s output is equal to $g_{i,T}(x,y)$, we note that, if the verifier accepts, then, for $i \in \{1,2\}$,

$$\mathbf{M}([\lambda_x(X)]_1)[\mathbf{d}_i]_2 = \boldsymbol{\theta}([\lambda_x(X)]_1)[s_i e_i]_2 + [\mathbf{a}]_1[s_i]_2,$$

which implies

$$\mathbf{M}([\lambda_x(X)]_1)[\delta^{\mathbf{d}}]_2 = \boldsymbol{\theta}([\lambda_x(X)]_1)[s_2(Y-e)]_2,$$

where the last equality follows by observing that $e_1 = Y$, and $e_2 = e$, and then by multiplying the first equation with s_2/s_1 and subtracting the second. The i-th row of the previous equation defines the polynomial

$$\sum_j ((m_{1ij}X + m_{0ij})\delta^{\mathbf{d}}_j) = s_2(Y-e)\sum_j ((m_{1ij}X + m_{0ij})(w_{1j}X + w_{0j})).$$

Thus, we have

$$\begin{aligned} s_2 g_{i,T}(X,Y) = X &\left[\sum_j (m_{1ij}\delta^{\mathbf{d}}_j + s_2 e(m_{1ij}w_{0ij} + m_{0ij}w_{1ij}))\right] \\ &- Y(s_2\sum_j m_{0ij}w_{0j}) + \sum_j (m_{0ij}\delta^{\mathbf{d}}_j + s_2 e m_{0ij}w_{0ij}). \end{aligned}$$

This completes the proof. □

D Full Extractability for the CH Framework

D.1 Knowledge Soundness of CH Argument Systems in the AGM

We show knowledge soundness of the argument system in Fig. 2 in the AGM framework. We recall that AGM essentially states that for every efficient algorithm $\mathcal{A}$ that outputs the vector $[\mathbf{y}]_\iota$ of group elements in $\mathbb{G}_\iota$ when given inputs the vector $[\mathbf{x}]_\iota$ of group elements in $\mathbb{G}_\iota$, there exists an efficient extractor $\mathsf{Ext}_{\mathcal{A}}$ that returns a matrix $\mathbf{A}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. In particular, since we are working in the setting of asymmetric bilinear pairings, we require that any outputs in one group must depend only on the inputs it receives in that group.

Lemma 7. *The NIZK argument in Fig. 2 is knowledge sound in the algebraic group model for asymmetric pairings, under DL-assumption in* $\mathbb{G}_2$.

Proof. Let $\mathcal{A}$ be a knowledge soundness adversary that on input $[1]_1, [1,e]_2$ outputs $[\mathbf{x},\mathbf{a}]_1, [\mathbf{d}]_2$. Since the verification equations hold, we have that,

$$\mathbf{M}(\mathbf{x}) \cdot \mathbf{d} = \boldsymbol{\theta}(\mathbf{x}) \cdot e + \mathbf{a} \cdot$$

Now, since $\mathcal{A}$ is an algebraic algorithm, there exists an extractor that outputs vectors $\mathbf{d}_0, \mathbf{d}_1, \mathbf{a}_0, \mathbf{a}_1$ such that $\mathbf{d} = \mathbf{d}_0 + \mathbf{d}_1 e$, $\mathbf{a} = \mathbf{a}_0$, $\mathbf{x} = \mathbf{a}_1$. The knowledge soundness extractor simply outputs $\mathbf{w} = \mathbf{d}_1$.

We show that this extractor outputs a witness whenever the verifier accepts, except with negligible probability. Each equation defined by the verifier's test can be written as a a univariate polynomial $Q_i(X) := d_{0i} + d_{1i}X = x_i X + a_i$ where $d_{0i} = (\mathbf{M}(\mathbf{a}_1) \cdot \mathbf{d}_0)_i$, $d_{1i} = (\mathbf{M}(\mathbf{a}_1) \cdot \mathbf{d}_1)_i$, $x_i = \boldsymbol{\theta}(\mathbf{a}_1)_i$ and $a_i = \mathbf{a}_{0i}$. Suppose for the sake of contradiction that $\mathcal{A}$ computed a valid proof $[\mathbf{a}]_1, [\mathbf{d}]_2$ for an adaptively chosen statement $[\mathbf{x}]_1$, but $\mathbf{w}$ output by the extractor as described above is not a valid witness, that is, $\mathbf{M}(\mathbf{x}) \cdot \mathbf{w} \neq \boldsymbol{\theta}(\mathbf{x})$. Then we can use $\mathcal{A}$ to break the DL assumption in $\mathbb{G}_2$.

The DL adversary receives a challenge $[e]_2$ and invokes $\mathcal{A}$ on input $\mathtt{crs} = [e]_2$. Then, the DL adversary obtains $\mathbf{d}_0, \mathbf{d}_1, \mathbf{a}_0, \mathbf{a}_1$ as defined above by the extractor for the algebraic adversary $\mathcal{A}$. If each polynomial $Q_i(X)$ is identically 0, that is $Q_i(X) \equiv 0$ as a polynomial, then $\mathbf{M}(\mathbf{a}_1) \cdot \mathbf{d}_1 = \boldsymbol{\theta}(\mathbf{a}_1)$ which implies that $\mathbf{w} = \mathbf{d}_1$ and the extractor doesn't fail. Otherwise, there exists i such that $Q_i(e) = 0$, for a non-zero polynomial $Q_i(X)$. Then $e = (a_i - d_{0i})/(d_{1i} - x_i)$ is the only root of $Q_i(e)$. Note that $Q_i(X) \not\equiv 0$ implies that $d_{1i} \neq x_i$ and thus the DL adversary succeeds in breaking the DL-assumption in $\mathbb{G}_2$.

D.2 Semantic, BB and n-BB Extraction

Semantic extraction demands that for every adversary that implements a strategy (an efficiently computable function that outputs an accepting proof) there exists an extractor. Unlike n-BB extraction where there could be a different extractor for every machine, in semantic extraction, one extractor for a function is a good extractor for all machines that implement that function. Semantic extraction is non-blackbox only in the randomness of the adversary but treats the adversary's machine as a black-box; our formal definition allows the extractor access to a part of the adversary's randomness. By allowing the extractor to see all or none of the prover's randomness, the semantic definition recovers standard n-BB and BB extraction definitions. We show that a NIZK satisfies semantic extraction where the extractor is given all the randomness of the adversary (called semn-BB) if and only if it satisfies the standard n-BB extraction definition. While it seems intuitive that the extractor's (in)ability to see the adversary's random coins makes the semantic extractor (BB)n-BB, this is not straightforward, especially the equivalence with BB definition. A BB extractor is also a semantic extractor. For the other direction, consider the case when the semantic extractor is not allowed to see the adversary's randomness; here we would like to argue that such a semantic extractor (called semBB) is a BB extractor. However, semantic extraction only guarantees a (potentially different) extractor for every function implemented by a prover. We therefore have to switch the order of quantifiers in order to construct one *universal* extractor that works for all provers. For a relaxed concrete security notion of extraction, we can indeed show this concluding that a special case of semantic extraction semBB implies BB extraction.

At a high-level, we rely on the minimax theorem from game theory to construct a universal extractor from function-dependent extractors. We define a utility function to capture how well the extractor performs. The minimax theorem guarantees the existence of a distribution over extractors. Computing this distribution is not guaranteed to be efficient. This can be done efficiently by using a multiplicative weights algorithm [32] to implement an approximate minimax strategy by knowing the randomness used by the adversary. However, this use of the adversary's randomness makes the universal extractor non-blackbox. We then show how to make the universal extractor BB without the randomness of the adversary. Our use of minimax is reminiscent of its use in proving the equivalence of distinguisher-dependent and universal simulators in [24], and in switching the order of quantifiers in the proof of the leakage lemma in [36].

From Semantic to BB and n-BB. In the definition of semantic extraction, the function implemented by the adversary uses randomness (s, t), and the extractor receives t, but not s; thus the extractor is allowed to see a part of the adversary's randomness. Let us consider the two extremes of the extractor's access: (i) the extractor is not given even r, that is, does not see the randomness of the adversary. (ii) the extractor is given both (s, t), that is, the extractor sees the entire randomness of the adversary. Intuitively, the former is black-box in the adversary, and the latter is white-box. However, in order to establish the equivalences, we also have to be careful with the order of quantifiers in the definition of extraction, which is different in the black-box and the semantic notion. In this section, we show that versions of semantic extraction where we control the randomness access of the extractor as in (i) and (ii) are equivalent to standard black-box (one side of the equivalence additionally needs a relaxed concrete (t, ϵ) variant of the definitions) and white-box definitions respectively.

We first give the concrete security definitions of black-box extraction, and *semantic black-box extraction* which is the semantic definition where the extractor is not given the randomness of the prover.

Definition 15. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is semantic black-box knowledge sound (semBB) if for each efficiently implementable knowledge soundness strategy f there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f*

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} \end{array} \middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s, t) \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; s, t); \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] \leq \mathsf{negl}(k).$$

Definition 16. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is (t, ϵ) black-box knowledge sound, if there exists an extractor $\mathsf{Ext}_{\mathsf{bb}}$ such that, for any t-time adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} \end{array} \middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}; r); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathsf{bb}}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] \leq \epsilon(k)$$

where r is the random coins of the adversary.

Definition 17. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is (t, ϵ) semBB (semantic black-box knowledge sound) if for each t-time implementable knowledge soundness strategy f, there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f*

$$\Pr\left[\begin{matrix}\mathcal{V}(\texttt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}\end{matrix}\middle| \begin{matrix}(\texttt{crs}, \texttt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s, t) \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\texttt{crs}; s, t); \mathtt{w} \leftarrow \mathsf{Ext}(\texttt{td}, \mathtt{x}, \pi)\end{matrix}\right] \le \epsilon(k).$$

Theorem 4. *Let Π be a NIZK argument that is BB knowledge sound as in Definition 1. Π is also semBB knowledge sound as in Definition 15. Conversely, if a NIZK argument Π is (t, ϵ) semBB knowledge sound as in Definition 17, for each polynomial t and inverse polynomial ϵ, then Π is (t', ϵ') BB knowledge sound for every polynomial t' and inverse polynomial ϵ' as in Definition 16.*

Proof. The first implication is straightforward. Let Ext be a BB extractor that satisfies Definition 1. Then this extractor is, by definition, a semantic black-box extractor for each efficiently implementable knowledge soundness strategy as in Definition 1.

We now prove the second implication. Suppose Π is (t, ϵ) semBB as in Definition 17, for each polynomial t and inverse polynomial ϵ. Let t' be any polynomial, and ϵ' any inverse polynomial. We show that Π is (t', ϵ') BB by constructing an extractor $\mathsf{Ext}_{\mathsf{BB}}$ and showing that it satisfies Definition 16.

High-level description of the extractor. The universal extractor $\mathsf{Ext}_{\mathsf{BB}}$ on input $\texttt{td}, \mathtt{x}, \pi$ uses the multiplicative weights algorithm [32] to find a good set of extractors $(\mathsf{Ext}_1, \dots, \mathsf{Ext}_L)$, then runs each of the extractors in the set and outputs a witness if at least one of the extractors succeeds.

We define the "advantage" of an extractor Ext with respect to a knowledge soundness strategy $\mathsf{kss} = f$ as follows. Since kss is efficiently implementable, we fix a PPT adversary $\mathcal{A}_{\mathsf{kss}}$ that implements kss.

$$\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) := \Pr\left[\begin{matrix}\mathcal{V}(\texttt{crs}, \mathtt{x}, \pi) = 1 \\ (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\end{matrix}\middle| \begin{matrix}r \leftarrow D, \\ (\mathtt{x}, \pi) = f(\texttt{crs}; r), \mathtt{w} \leftarrow \mathsf{Ext}(\texttt{td}, \mathtt{x}, \pi)\end{matrix}\right].$$

Note that we define this advantage for a fixed pair of $(\texttt{crs}, \texttt{td})$. We would now like to define this advantage for a distribution over the set kss of knowledge soundness strategies $\mathsf{kss}_1, \dots, \mathsf{kss}_k$; consider the set $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$ of efficient uniform machines with description of size $\le \log k$ that are implementations of the set of kss. We also redefine each $\mathcal{A}_{\mathsf{kss}_j}$ such that it halts and outputs $\perp$ after t' steps. Each fixed t'-time machine $\mathcal{A}$ for $t' = \mathrm{poly}(k)$ will eventually appear in the set.

Given a distribution $\mathcal{D}$ over the set $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$, we define the advantage of the extractor Ext with respect to the distribution as

$$\mu(\mathsf{Ext}, \mathcal{D}) := \mathbb{E}_{\mathcal{A}_{\mathsf{kss}} \sim \mathcal{D}}\left[\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})\right] = \sum_{\mathcal{A}_{\mathsf{kss}} \in Supp(\mathcal{D})} \mathcal{D}(\mathcal{A}_{\mathsf{kss}}) \cdot \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}).$$

where $(\mathcal{D}(\mathcal{A}_{\mathsf{kss}_1}), \dots, \mathcal{D}(\mathcal{A}_{\mathsf{kss}_k}))$ is the vector of probability weights representing $\mathcal{D}$. Our goal is to construct an extractor Ext such that for every t' implementable strategy that is implemented by $\mathcal{A}_{\mathsf{kss}}$, we have that

$$\Pr\left[\begin{array}{c}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1\\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R}\end{array}\middle|\begin{array}{c}(\mathtt{crs},\mathtt{td})\leftarrow\mathsf{CRSGen}(1^k);\\ (\mathtt{x},\pi)\leftarrow\mathcal{A}_{\mathsf{kss}}(\mathtt{crs};r);\mathtt{w}\leftarrow\mathsf{Ext}(\mathtt{td},\mathtt{x},\pi)\end{array}\right]\leq\epsilon'(k).$$

We note that, this is equivalent to constructing Ext such that for every t' implementable strategy (implemented by $\mathcal{A}_{\mathsf{kss}}$),

$$\mathbb{E}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext},\mathcal{A}_{\mathsf{kss}})\right]\geq 1-\mathcal{O}(\epsilon'(k)).$$

We now give an overview of the multiplicative weights algorithm. The extractor emulates a certain number of rounds of a zero-sum game between an extractor player and a knowledge soundness adversary. The payoff function for the extractor is the advantage $\mu(\cdot,\cdot)$. In each round, the knowledge soundness adversary chooses a distribution $\mathcal{D}$, and the extractor chooses Ext_i such that its expected payoff is high. We begin with the uniform distribution $\mathcal{D}^{(1)}$ over $\{\mathcal{A}_{\mathsf{kss}_1},\mathcal{A}_{\mathsf{kss}_2},\ldots,\mathcal{A}_{\mathsf{kss}_k}\}$. In each round, this is updated to $\mathcal{D}^{(i+1)}$ using the multiplicative weights algorithm using the advantage function $\mu(\cdot,\cdot)$. For this, the knowledge soundness adversary in the two player game needs to compute the payoff function $\mu(\mathsf{Ext},\mathcal{A}_{\mathsf{kss}})$ of an extractor that is good with respect to $\mathcal{A}_{\mathsf{kss}}$. We use a universal adversary that takes a description of a knowledge soundness adversary $\mathcal{A}_{\mathsf{kss}}$ as auxiliary input and runs $\mathcal{A}_{\mathsf{kss}}$ in order to compute the payoff function. Then, we choose an extractor that is good with respect to this universal adversary. The extractor, therefore needs to efficiently find the kss-dependent extractor for the mixed strategy $\mathcal{D}^{(i)}$ over kss implementations. This is done by using the universal adversary $\mathcal{A}_U$ that takes the vector of probability weights representing $\mathcal{D}$ as auxiliary input, samples a kss adversary from the distribution, and runs the sampled adversary. Let $\mathsf{Ext}_{\mathcal{A}_U}$ be the extractor for the kss implemented by $\mathcal{A}_U$ that is guaranteed to exist by semantic extraction. In the ith round, we choose Ext_i to be the machine that runs $\mathsf{Ext}_{\mathcal{A}_U}$ given the weights of $\mathcal{D}^{(i)}$ as auxiliary input. The description of this extractor is given in Fig. 9. Later, we show how to make $\mathsf{Ext}_{\mathsf{BB}}$ efficient when Ext_i is not given any auxiliary input.

It can be verified that $\mathsf{Ext}_{\mathsf{BB}}$ runs in time $\mathcal{O}(L[\gamma(t'+T_U)+T_U])=\mathcal{O}(\frac{\log k}{\epsilon'(k^2)}[\frac{\log(kL/\epsilon'(k))}{\epsilon'(k)^2}k(t'+T_U)+T_U])$, that is polynomial in t' and $1/\epsilon'$. To prove the theorem we must show that, for each t' implementable knowledge soundness strategy kss, $\mathbb{E}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_{\mathsf{BB}},\mathcal{A}_{\mathsf{kss}})\right]\geq 1-\mathcal{O}(\epsilon'(k))$. In order to show this, we rely on two auxiliary lemmas: the first shows that if in each round Ext_i does well against $\mathcal{D}^{(i)}$ with respect to $\tilde{\mu}(\cdot,\cdot)$, then Ext does well against each $\mathcal{A}_{\mathsf{kss}}$. This follows from the analysis of the multiplicative weights algorithm. The second lemma shows that the above statement holds for $\mu(\cdot,\cdot)$.

Lemma 8. *For every knowledge soundness strategy implementation $\mathcal{A}_{\mathsf{kss}_j}\in\{\mathcal{A}_{\mathsf{kss}_1},\mathcal{A}_{\mathsf{kss}_2},\ldots,\mathcal{A}_{\mathsf{kss}_k}\}$, the extractor defined in Fig. 9 generates $\mathcal{D}^{(1)},\ldots,\mathcal{D}^{(L)}$ and $\mathsf{Ext}_1,\ldots,\mathsf{Ext}_L$ such that*

$$\frac{1}{L}\sum_{i=1}^{L}\tilde{\mu}(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_j})\geq\frac{1}{L}\sum_{i=1}^{L}\tilde{\mu}(\mathsf{Ext}_i,\mathcal{D}^{(i)})-\mathcal{O}(\epsilon'(k)).$$

$\mathsf{Ext}_{\mathsf{BB}}(\mathtt{td}, \mathtt{x}, \pi)$.

- Let $\mathcal{D}$ be a distribution over a set of kss implementations $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$. Let $\mathfrak{w}_{\mathcal{D}}$ be the vector of weights representing $\mathcal{D}$. That is $(\mathfrak{w}_{\mathcal{D}})_j = \Pr\left[\mathcal{A}_{\mathsf{kss}_j} \xleftarrow{\$} \mathcal{D}\right]$. Let $\mathcal{A}_U$ be the PPT that on input $(\mathtt{crs}, r)$ interprets r as $\mathfrak{w}_{\mathcal{D}}||\chi||r'$, samples a knowledge soundness strategy adversary $\mathcal{A}_{\mathsf{kss}_j}$ from $\mathcal{D}$, using random coins χ, and runs $\mathcal{A}_{\mathsf{kss}_j}$ on $(\mathtt{crs}, r')$. Let f be the function implemented by $\mathcal{A}_U$, and T_U be a polynomial that bounds the running time of $\mathcal{A}_U$. Let $\mathsf{Ext}_{\mathcal{A}_U}$ be the (T_U, ϵ') black-box semantic extractor for $\mathcal{A}_U$ as in Definition 17.
- Let $L = \Theta(\frac{\log k}{\epsilon'(k^2)})$ and $\beta = \frac{1}{1+\sqrt{(2\log k)/L}}$.
- Let $\mathcal{D}^{(1)}$ be the uniform distribution over $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$. For $i = 1, \dots, L$ do
    1. On input $(\mathtt{td}, \mathtt{x}, \pi)$, consider the adversary $\mathcal{A}_{U_i}(\mathtt{crs}, (\chi||r))$ which is defined as $\mathcal{A}_U(\mathtt{crs}, r')$, where $r' = \mathfrak{w}_{\mathcal{D}^{(i)}}||\chi||r$. Let f_i be the function implemented by $\mathcal{A}_{U_i}$. Note that $f_i(\mathtt{crs}, (\chi||r)) = f(\mathtt{crs}, (\mathfrak{w}_{\mathcal{D}^{(i)}}||\chi||r))$. Note also that f_i is an efficiently implementable knowledge soundness strategy, since the running time of $\mathcal{A}_{U_i}$ is bound by T_U. Let Ext_i be the (T_U, ϵ') black-box semantic extractor for $\mathcal{A}_{U_i}$.
    2. Let $\mathcal{D}^{(i+1)}$ be defined as $\beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})} \cdot \mathcal{D}^{(i)}$ up to renormalizing. That is
    $$\mathcal{D}^{(i+1)}(\mathcal{A}_{\mathsf{kss}_j}) = \frac{\beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})} \mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_j})}{\sum_{l=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_l})} \mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_l})}$$
    where $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$ is defined by the procedure in Fig. 10. $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$ can be thought of as an approximation of $\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$.
- Run each extractor Ext_i in the set $\{\mathsf{Ext}_1, \dots, \mathsf{Ext}_L\}$, and verify if one of them succeeded in computing a valid witness.
- Output a valid witness if available, else output $\perp$.

Fig. 9: The black-box (t', ϵ') extractor.

Let $\gamma = \Theta(\frac{\log(kL/\epsilon'(k))}{\epsilon'(k)^2})$. Let $\mathsf{freq} = 0$. For $i = 1, \dots, \gamma$ do

1. Sample $r \leftarrow D$. Compute $(\mathtt{x}, \pi) = \mathcal{A}_{\mathsf{kss}}(\mathtt{crs}, r)$.
2. Compute $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)$.
3. If $\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1$ and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, then $\mathsf{freq} = \mathsf{freq} + 1$.

$\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) = \mathsf{freq}/\gamma$.

Fig. 10: $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$

Proof. Recall that the relative entropy of two random variables X and Y is defined as

$$\mathbf{KL}(X||Y) = \sum_{x \in supp(X)} \Pr[X = x] \ln \frac{\Pr[X = x]}{\Pr[Y = y]}.$$

Now, consider a strategy $\mathcal{A}_{\mathsf{kss}_j} \in \{\mathcal{A}_{\mathsf{kss}_1}, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$. Fix a pair $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}$. Lastly, fix the random tape of the extractor. in this way, all the random variables that appears in Fig. 9, became fixed.

We begin showing that for each $i \in \{1, \ldots, L\}$ we have

$$\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i+1)}) - \mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i)}) \leq (\ln \frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta) \sum_{b=1}^{k} \Pr\left[\mathcal{D}^{(i)} = \mathsf{kss}_b\right] \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}). \tag{5}$$

Upon fixed i, we have

$$\begin{aligned}
\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i+1)}) - \mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i)}) &= \ln \frac{1}{\Pr\left[\mathcal{D}^{(i+1)} = \mathsf{kss}_j\right]} - \ln \frac{1}{\Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_j}\right]} \\
&= \ln \frac{\Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_j}\right]}{\Pr\left[\mathcal{D}^{(i+1)} = \mathcal{A}_{\mathsf{kss}_j}\right]} \\
&= \ln \frac{\sum_{b=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b})} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]}{\beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})}} \\
&= \ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln \sum_{b=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b})} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right].
\end{aligned}$$

Since, $x \in [0, 1]$ and $\beta > 0$ imply that $\beta^x \leq 1 - (1 - \beta)x$, recalling that $\sum_{b=1}^{k} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] = 1$, we have

$$\begin{aligned}
&\ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln \sum_{b=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b})} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] \\
&\leq \ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln \left(1 - (1-\beta) \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]\right)
\end{aligned}$$

Lastly, from $x < 1$ implies that $\ln(1 - x) \leq -x$, we have

$$\begin{aligned}
&\ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln \left(1 - (1-\beta) \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]\right) \\
&\qquad \leq \ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta) \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right],
\end{aligned}$$

which complete the proof of Eq. (5).

Now, summing Eq. (5) over $i \in \{1, ..., L\}$, we have

$$\begin{aligned}&\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(L+1)}) - \mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(1)})\\ \leq \ln(\frac{1}{\beta}) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta) &\sum_{i=1}^{L}\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right].\end{aligned}$$

From the last inequality and using $\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(L+1)}) \geq 0$, $\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(1)}) \leq \ln k$ and $\ln(\frac{1}{\beta}) \leq \frac{1-\beta^2}{2\beta}$ (which holds because $\beta \in (0, 1]$), we have

$$-\ln k \leq \frac{1-\beta^2}{2\beta} \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta) \sum_{i=1}^{L}\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right].$$

Rearranging the last inequality we have

$$\begin{aligned}\sum_{i=1}^{L}\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] &\leq \frac{1-\beta^2}{2\beta(1-\beta)} \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}}j) + \frac{1}{1-\beta} \ln k\\ &= \frac{1+\beta}{2\beta} \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1}{1-\beta} \ln k\\ &= \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \left(\frac{1+\beta}{2\beta} - 1\right) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1}{1-\beta} \ln k.\end{aligned}$$

We recall here that $\beta = \frac{1}{1-\sqrt{(2 \log k)/L}}$. So $\frac{1}{1-\beta} \ln k = \frac{\sqrt{2L \ln k}}{2} + \ln k$ and $\frac{1-\beta}{2\beta} L + \frac{\sqrt{2L \ln k}}{2} = \sqrt{2L \ln k}$, which imply

$$\begin{aligned}\sum_{i=1}^{L}\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] &\leq \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \left(\frac{1+\beta}{2\beta} - 1\right) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1}{1-\beta} \ln k\\ &= \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \left(\frac{1+\beta}{2\beta} - 1\right) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{\sqrt{2L \ln k}}{2} + \ln k\\ &\leq \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1-\beta}{2\beta} L + \frac{\sqrt{2L \ln k}}{2} + \ln k\\ &= \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \sqrt{2L \ln k} + \ln k.\end{aligned}$$

Finally, rearranging the inequality and dividing by L we have the result. For the reader convenience, we also recall here that, by definition,

$$\tilde{\mu}(\mathsf{Ext}_i, \mathcal{D}^{(i)}) = \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right].$$

□

Lemma 9. *For each* $\mathcal{A}_{\mathsf{kss}_j} \in \{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$*, with probability* $1 - \mathcal{O}(\epsilon(k))$ *over the random coins of the extractor, the extractor defined in Fig. 9 generates* $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(L)}$ *and* $\mathsf{Ext}_1, \ldots, \mathsf{Ext}_L$ *such that*

$$\frac{1}{L}\sum_{i=1}^{L} \mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) \geq \frac{1}{L}\sum_{i=1}^{L} \mu(\mathsf{Ext}_i, \mathcal{D}^{(i)}) - \mathcal{O}(\epsilon'(k)).$$

Proof. As done in the previous lemma, fix a pair $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}$, and fix the random tape of the extractor. in this way, all the random variables that appears in Fig. 9, become fixed. We begin to show that, $|\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \leq \mathcal{O}(\epsilon'(k))$ with probability $1 - \mathcal{O}(\frac{\epsilon'(k)}{kL})$, for each extractor Ext and each $\mathcal{A}_{\mathsf{kss}} \in \{\mathcal{A}_{\mathsf{kss}}1, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$. Let X denotes the random variable that counts the number of success of the extractor Ext, when one compute $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$, as prescribed in Fig. 10. That is $X = freq$, where $freq$ is the variable defined in Fig. 10. Formally, $X = \gamma\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$. Note that the expected value of X is $\mathbb{E}(X) = \gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$. Now,

$$\begin{aligned}
\Pr\left[|\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\right] &= \Pr\left[|X - \gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\gamma\right] \\
&\leq \Pr\left[|X - \gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})\right] \\
&= \Pr\left[|X - \mathbb{E}(X)| \geq \epsilon'(k)\mathbb{E}(X)\right].
\end{aligned}$$

We recall here the multiplicative form of Chernoff bound for a random variable X. For each $\delta > 0$, it holds that

$$\Pr\left[|X - \mathbb{E}(X)| \geq \delta\mathbb{E}(X)\right] \leq 2e^{-(\delta^2\mathbb{E}(X))/3}.$$

We also recall that $\gamma = \Theta(\frac{\log(kL/\epsilon'(k))}{\epsilon'(k)^2})$. Applying the Chernoff bound to the last term of the inequality above, we have

$$\begin{aligned}
\Pr\left[|\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\right] &\leq 2e^{-(\epsilon(n)^2\mathbb{E}(X))/3} \\
&= 2e^{-(\epsilon(k)^2\gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}))/3} \\
&= 2\left(\frac{kL}{\epsilon'(k)}\right)^{(-C\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}))} \qquad = \mathcal{O}(\frac{\epsilon'(k)}{kL}),
\end{aligned}$$

where C is a positive constant.

By the union bound, we have

$$|\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \leq \mathcal{O}(\epsilon'(k)), \tag{6}$$

for each $i \in \{1, \ldots, L\}$, with probability at least $1 - kL\mathcal{O}(\frac{\epsilon'(k)}{kL}) = 1 - \mathcal{O}(\epsilon'(k))$.

Finally, conditioned on the previous event, we have

$$\begin{aligned}
\frac{1}{L}\sum_{i=1}^{L}\mu(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_j}) &\geq \frac{1}{L}\sum_{i=1}^{L}\tilde{\mu}(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_j}) - \mathcal{O}(\epsilon'(k)) \\
&\geq \frac{1}{L}\sum_{i=1}^{L}\sum_{k=1}^{n}\tilde{\mu}(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_k})\Pr\left[\mathcal{D}^{(i)}=\mathcal{A}_{\mathsf{kss}_k}\right] - \mathcal{O}(\epsilon'(k)) \\
&\geq \frac{1}{L}\sum_{i=1}^{L}\sum_{b=1}^{k}(\mu(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_b}) - \mathcal{O}(\epsilon'(k)))\Pr\left[\mathcal{D}^{(i)}=\mathcal{A}_{\mathsf{kss}_b}\right] - \mathcal{O}(\epsilon'(k)) \\
&= \frac{1}{L}\sum_{i=1}^{L}\mu(\mathsf{Ext}_i,\mathcal{D}^{(i)}) - \mathcal{O}(\epsilon'(k)).
\end{aligned}$$

Here the second inequality holds by Lemma 8 and the other inequalities follows by Eq. (6). □

Now, we show that, for each $i \in \{1,\ldots,L\}$, Ext_i is a good extractor against $\mathcal{D}^{(i)}$, that is $\mathbb{E}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_i,\mathcal{D}^{(i)})\right] \geq \mathcal{O}(\epsilon'(k))$ for each i. Consider,

$$\begin{aligned}
\mu(\mathsf{Ext}_i,\mathcal{D}^{(i)}) &= \sum_{j=1}^{k}\mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_j})\cdot\mu(\mathsf{Ext},\mathcal{A}_{\mathsf{kss}_j}) \\
&= \sum_{j=1}^{k}\mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_j})\cdot\Pr\left[\begin{array}{c}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1\\ \wedge\\ (\mathtt{x},\mathtt{w})\in\mathcal{R}\end{array}\middle|\begin{array}{c}r\leftarrow D_j,\\ (\mathtt{x},\pi)=f_j(\mathtt{crs};r),\\ \mathtt{w}\leftarrow\mathsf{Ext}(\mathtt{td},\mathtt{x},\pi)\end{array}\right]. \\
&= \mu(\mathsf{Ext}_i,\mathcal{A}_{U_i}) = \mu(\mathsf{Ext}_{\mathcal{A}_{U_i}},\mathcal{A}_{U_i})
\end{aligned}$$

The second equality is given by the definition of $\mathcal{A}_{U_i}$, and the third inequality follows from the definition of Ext_i. Let f be the kss implemented by $\mathcal{A}_{U_i}$. Now, since $\mathsf{Ext}_{\mathcal{A}_{U_i}}$ is a good extractor for f, by the (t',ϵ') semantic black-box extraction, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_i,\mathcal{D}^{(i)})\right] &= \mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_{\mathcal{A}_{U_i}},\mathcal{A}_{U_i})\right] \\
&= \Pr\left[\begin{array}{c}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1\\ \wedge(\mathtt{x},\mathtt{w})\in\mathcal{R}\end{array}\middle|\begin{array}{c}(\mathtt{crs},\mathtt{td})\leftarrow\mathsf{CRSGen}(1^k);s\leftarrow D\\ (\mathtt{x},\pi)\leftarrow\mathcal{A}_{U_i}(\mathtt{crs};s);\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_{U_i}}(\mathtt{td},\mathtt{x},\pi)\end{array}\right] \\
&\geq 1-\epsilon'(k) \qquad (7)
\end{aligned}$$

From Lemma 9, with probability $1-\mathcal{O}(\epsilon'(k))$, the generated $\{\mathsf{Ext}_1,\ldots,\mathsf{Ext}_L\}$ are such that

$$\frac{1}{L}\sum_{i=1}^{L}\mu(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_j}) \geq \frac{1}{L}\sum_{i=1}^{L}\mu(\mathsf{Ext}_i,\mathcal{D}^{(i)}) - \mathcal{O}(\epsilon'(k))$$

$$\frac{1}{L}\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}\left[\sum_{i=1}^{L}\mu(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_j})\right] \geq \frac{1}{L}\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}\left[\sum_{i=1}^{L}\mu(\mathsf{Ext}_i,\mathcal{D}^{(i)})\right] - \mathcal{O}(\epsilon'(k)) \qquad (8)$$

From Eq. (7) and Eq. (8),

$$\frac{1}{L}\sum_{i=1}^{L}\underset{(\mathtt{crs},\mathtt{td})}{\mathbb{E}}[\mu(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}})] \geq 1-\mathcal{O}(\epsilon'(k)).$$

Finally, since $\mathsf{Ext}_{\mathsf{BB}}$ generates the set of $\{\mathsf{Ext}_i\}$, and fails only if all of them fail, we have

$$\underset{(\mathtt{crs},\mathtt{td})}{\mathbb{E}}[\mu(\mathsf{Ext}_{\mathsf{BB}},\mathcal{A}_{\mathsf{kss}})] \geq \frac{1}{L}\sum_{i=1}^{L}\underset{(\mathtt{crs},\mathtt{td})}{\mathbb{E}}[\mu(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}})] \geq 1-\mathcal{O}(\epsilon'(k)).$$

□

Unfortunately, the extractor depicted in Fig. 9 is not guaranteed to be polynomial time since it is not efficient to find the distribution-dependent extractor Ext_i. Using an auxiliary input to encode the distribution is an idea used in prior works like [24], however, since we are interested in a BB extractor, we cannot allow the extractor to read auxiliary inputs. Instead, we interpret the universal KSS – that takes an auxiliary input, a string encoding each distribution $\mathcal{D}^{(i)}$, samples a distribution, then samples a KSS as per that distribution – also as a knowledge soundness strategy. We then show that invoking the extractor corresponding to this universal KSS works well against distribution dependent $\mathcal{A}_{U_i}$. Note that such a $\mathcal{A}_U$ is indeed polynomial time: Each distribution $\mathcal{D}$ computed in the "for" loop of Fig. 9 can be represented as a weight vector $\mathfrak{w}_{\mathcal{D}}$ of polynomial length.

Let $\mathsf{Ext}_{\mathcal{A}_U}$ be a (t,ϵ^2) extractor against the knowledge soundness strategy $\mathcal{A}_U$, where $\mathcal{A}_U$ is as defined above. We show in the following lemma that $\mathsf{Ext}_{\mathcal{A}_U}$ is a good approximation of an extractor for any distribution dependent $\mathcal{A}_{U_i}$, with probability greater than $1-\mathcal{O}(\epsilon)$. Thus, we can define the efficient universal black-box extractor as follows: run L independent executions of $\mathsf{Ext}_{\mathcal{A}_U}$ and output a valid witness if at least one of the executions succeeds. The used $\mathsf{Ext}_{\mathcal{A}_U}$ has to be a (t,ϵ) with ϵ much better than ϵ'^2.

Lemma 10. *Let N be the number of times that any semantic extractor is called in the procedure defined in Fig. 9. Let L be defined as in Fig. 9. Let $\mathsf{Ext}_{\mathcal{A}_U}$ be the (t,ϵ) semantic black-box extractor against $\mathcal{A}_U$, where $\epsilon=\mathcal{O}(\epsilon'^2/N)$. Then the procedure defined by running L independent executions of $\mathsf{Ext}_{\mathcal{A}_U}$ is a black-box (Lt,ϵ') extractor, for every inverse polynomial ϵ,ϵ'.*

Proof. Let Y be the conditional expectation of the failure of $\mathsf{Ext}_{\mathcal{A}_U}$ against $\mathcal{A}_U$, given a fixed distribution $\mathfrak{w}_{\mathcal{D}}$ over $\mathcal{A}_{\mathsf{kss}}$-es. Formally we have

$$\begin{aligned} Y(\mathfrak{W}) &= \mathbb{E}\left[1-\mu(\mathsf{Ext}_{\mathcal{A}_U},\mathcal{A}_U)\,|\,[\mathfrak{W}=\mathfrak{w}]\right] \\ &= \Pr\left[\begin{matrix}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1\\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R}\end{matrix}\middle|\begin{matrix}(\chi,r)\leftarrow D\\ (\mathtt{x},\pi)\leftarrow\mathcal{A}_U(\mathtt{crs};(\mathfrak{w}||\chi||r));\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td},\mathtt{x},\pi)\end{matrix}\right].\end{aligned}$$

Note that $\mathbb{E}[Y] = \epsilon(k)$, by definition. We now recall Markov inequality. For each non-negative random variable X that admits expected value $\mathbb{E}[X]$, for each value α it holds that

$$\Pr[X \geq \alpha] \leq \frac{\mathbb{E}[X]}{\alpha}.$$

Applying the inequality to Y we get

$$\Pr[Y \geq \epsilon'(k)] \leq \frac{\epsilon(k)}{\epsilon'(k)} = \mathcal{O}(\epsilon'(k)/N(k)).$$

Note that, for each of the k $\mathcal{A}_{\mathsf{kss}}$, there exists $\mathfrak{w}$ such that

$$\begin{aligned}
&\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 & r \leftarrow D \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} & (\mathtt{x}, \pi) \leftarrow \mathcal{A}_{\mathsf{kss}}(\mathtt{crs}; r); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] \\
=&\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 & (\chi, r) \leftarrow D \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} & (\mathtt{x}, \pi) \leftarrow \mathcal{A}_U(\mathtt{crs}; (\mathfrak{w}||\chi||r)); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] = Y(\mathfrak{w}).
\end{aligned}$$

Here $\mathfrak{w}$ is the distribution that puts a weight of 1 on $\mathcal{A}_{\mathsf{kss}}$. Moreover, for each distributions $\mathcal{D}^{(i)}$, represented by vector of weights $\mathfrak{w}_i$ and the corresponding $\mathcal{A}_{U_i}$, we have

$$\begin{aligned}
&\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 & (\chi||r) \leftarrow D \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} & (\mathtt{x}, \pi) \leftarrow \mathcal{A}_{U_i}(\mathtt{crs}; (\chi||r)); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] \\
=&\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 & (\chi, r) \leftarrow D \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} & (\mathtt{x}, \pi) \leftarrow \mathcal{A}_U(\mathtt{crs}; (\mathfrak{w}_i||\chi||r)); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] = Y(\mathfrak{w}_i).
\end{aligned}$$

Suppose now, we define a universal extractor as the one defined in Fig. 9, except, we run $\mathcal{A}_U$ every time an extractor is called in the procedure. This new universal extractor is a good approximation of the one in Fig. 9, as long as the distribution of $Y(\mathfrak{W})$ is sufficiently dense around its average. Indeed, using Markov inequality, we show how to choose ϵ as a function of ϵ' so that each time we use $\mathsf{Ext}_{\mathcal{A}_U}$ instead of any other extractor in the proof of Theorem 4, with overwhelming probability, we have an average loss of $\mathcal{O}(\epsilon'(k)/N(k))$. Now applying the union bound we have the result. The resulting BB extractor runs in time Lt.

□

Semantic and white-box extraction. We now state the restricted semantic knowledge soundness definition for which the equivalence to white-box knowledge soundness holds. We consider knowledge soundness strategies f such that $f : \mathbf{CRS} \times \mathbf{\Gamma_t} \rightarrow \chi \times \Psi$ and $\mathbf{\Gamma_s}$ is the set that contains only the empty string.

Definition 18. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is semantic white-box knowledge sound (semn-BB) if for each efficiently implementable knowledge soundness strategy f, there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f*

$$\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 & (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); r \leftarrow D \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} & (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; r); \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, r) \end{array}\right] \leq \mathsf{negl}(k).$$

Theorem 5. *A NIZK argument is white-box knowledge sound (semn-BB) as in Definition 2 if and only if it is also semantic knowledge sound as in Definition 18.*

Proof. Any semantic extractor that satisfies Definition 18, is also, by definition, a white-box extractor for each PPT that implements a certain function. So "only if" side is trivial. To show the other direction, suppose that there exists two efficient PPT machines $\mathcal{A}, \mathcal{A}'$ that implement the same function f. Let $\mathsf{Ext}, \mathsf{Ext}'$ be the corresponding white-box extractors as in Definition 2. We show that Ext is a semantic extractor for $\mathcal{A}'$.

Consider the set of tuples $(\mathtt{crs}, \mathtt{x}, \pi)$ such that there exists r for which $(\mathtt{x}, \pi) = f(\mathtt{crs}, r)$. We can divide this set into two disjoint subset. The first subset is defined by the tuples such that $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, r), \mathtt{w}' \leftarrow \mathsf{Ext}'(\mathtt{td}, \mathtt{x}, \pi, r)$ and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}, (\mathtt{x}, \mathtt{w}') \in \mathcal{R}$ with overwhelming probability. Given the tuple belongs to this set, then Ext will also be a good extractor for $\mathcal{A}'$, although in the general case it can return a different (but still valid) witness with respect to Ext'.

We now consider the set of tuples $(\mathtt{crs}, \mathtt{x}, \pi)$ such that at least one extractor fails with non-negligible probability. Consider the subset of tuples such that $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, r), \mathtt{w}' \leftarrow \mathsf{Ext}'(\mathtt{td}, \mathtt{x}, \pi, r)$ and $(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}, (\mathtt{x}, \mathtt{w}') \in \mathcal{R}$. This set is of negligible size, since it is a subset of the set of tuples for which Ext fails, which is negligible dy definition.

Thus, Ext is a good semantic extractor for each PPT that implements the function f. It only remains to show that Ext is a good extractor even against unbounded TMs that implement f. This is true since the set in which Ext fails is of negligible size. So, let $\mathcal{A}^*$ be an unbounded TM that implements f. Clearly, Ext is a good extractor for $\mathcal{A}^*$, since $\mathcal{A}(\mathtt{crs}, r) = \mathcal{A}^*(\mathtt{crs}, r)$ for each $(\mathtt{crs}, r)$. It is also a good semantic extractor for $\mathcal{A}^*$, since the set of $(\mathtt{crs}, \mathtt{x}, \pi)$ tuples such that Ext fails on $\mathcal{A}^*$ but not for $\mathcal{A}$ is the empty set.

D.3 Impossibility of Semantic Knowledge Soundness for CH-NIZK

Theorem 6 (Theorem 2 restated). *Let $\mathcal{L}_{\mathtt{lpar}}$ be any 1DL-friendly algebraic language with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$. The NIZK argument in Fig. 2 cannot be semantic knowledge sound for $\mathcal{L}_{\mathtt{lpar}}$ under the SDL assumption.*

Proof. The proof is similar to the proof of Lemma 2. Fix a language with the properties mentioned in the statement; that is, fix suitable $\mathbf{M}, \boldsymbol{\theta}$. Suppose that the relative NIZK argument is semantic knowledge sound. Define the canonical prover adversary, on input $\mathtt{crs} = [e]_2$ and randomness $(s, \mathbf{r}, r')$, in the following way:

1. Sample uniformly random w_1 using seed s.
2. Using random coins r' sample all the other integer, $w_2, \ldots, w_d$ and define $\mathbf{w} = \lambda_w(w_1, \ldots, w_n)$.
3. Compute $\mathbf{x} = \lambda_x(w_1, \ldots, w_n)$.
4. Using random coins $\mathbf{r}$ compute $\mathbf{a}$, $[\mathbf{d}]_2$ as prescribed by the honest prover.
5. Output $([\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$.

Let Ext_f be the semantic extractor defined for the canonical adversary. We can exploit it to define an adversary $\mathcal{A}$ for the SDL assumption. On input an SDL challenge $([w_1]_1, [w_1]_2)$, $\mathcal{A}$ do the following.

1. Sample $e, \mathbf{r}, r'$.
2. Using random coins r' sample all the other integer, $w_2, ...w_d$ and define $[\mathbf{w}]_2 = \lambda_w([w_1]_2, w_2 \dots, w_n)$. Recall that this is efficiently computable since λ_w is linear in w_1.
3. Compute $[\mathbf{x}]_1 = \lambda_x([w_1]_1, w_2 \dots, w_n)$. Recall that this is efficiently computable since λ_x is linear in w_1.
4. Compute $[\mathbf{a}]_1$ as $\mathbf{r}[\mathbf{M}(\mathbf{x})]_1$.
5. Compute $[\mathbf{d}]_2$ as $e[\lambda(\mathbf{w})]_2 + [\mathbf{r}]_2$.
6. Compute $\mathbf{w} \leftarrow \mathsf{Ext}_f(([\mathbf{a}]_1, [\mathbf{d}]_2), e, [\mathbf{x}]_1, (\mathbf{r}, r'))$.
7. Output w_1.

Since $\mathcal{A}$ computes the same function as an unbounded prover that is able to recover e from $[e]_2$, inputs provided to the extractor are correctly distributed. Thus, $\mathcal{A}$ computes discrete logarithm w_1 with the same probability that Ext_f is successful, breaking the SDL assumption.

D.4 Impossibility of Semantic Extractability for SPHF-based QA-NIZKs

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS can depend on some parameters $\mathtt{lpar}$ of the language for which proofs have to be generated [44]. The language dependent preprocessing improves efficiency and leads to succinct proofs which have size as short as a single group element [44]. QA-NIZK arguments are not arguments of knowledge in general. While in [22] it has been shown that QA-NIZK arguments can satisfy this property in the generic/algebraic group model, showing this property in the case of black-box extraction where the extractor can extract a witness from the prover using only its input/output interface seems counterintuitive as the proof size is shorter than the witness. In this section, we prove this intuition to be correct by giving a stronger impossibility result which shows SPHF-based QA-NIZKs with semantic knowledge soundness cannot exist. More precisely, we consider the most efficient QA-NIZKs Π_{kw} by Kiltz and Wee [45] and show that it cannot be semantic knowledge sound.

D.5 Overview of Kiltz-Wee QA-NIZK

The core idea of the NIZK proof system in [45] for linear space membership languages is as follows: starting from a DVS-based SPHF for the language (see Fig. 7), which can be seen as a symmetric-key analogue of NIZK with a designated verifier and then translating it to the bilinear group setting. To be more precise, let $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = [\mathbf{M}]_1 \in \mathbb{G}_1^{\ell \times k}$ be the linear language defined as

$$\mathcal{L}_{\mathtt{lpar}} = \left\{ [\mathbf{x}]_1 \in \mathbb{G}_1^{\ell} | \exists \mathbf{w} \in \mathbb{Z}_p^k : [\mathbf{x}]_1 = [\mathbf{M}]_1 \cdot \mathbf{w} \right\} \tag{9}$$

```
$\mathsf{CRSGen}(\mathtt{lpar} = [\mathbf{M}]_1 \in \mathbb{G}_1^{\ell \times k})$
$\mathbf{A} \leftarrow \mathcal{D}_t; \boldsymbol{\alpha} \leftarrow \mathbb{Z}_p^{\ell \times (t+1)}$
$[\boldsymbol{\gamma}]_1 := [\mathbf{M}^\top \boldsymbol{\alpha}]_1; \mathbf{C} := \boldsymbol{\alpha}\mathbf{A}$
$\mathtt{crs} := ([\boldsymbol{\gamma}]_1, [\mathbf{C}, \mathbf{A}]_2)$
return $(\mathtt{crs}, \mathtt{td} = \boldsymbol{\alpha})$

$\mathsf{Sim}(\mathtt{crs}, \mathtt{td} = \boldsymbol{\alpha}, [\mathbf{x}]_1)$
return $\boldsymbol{\pi} := [\mathbf{x}^\top \boldsymbol{\alpha}]_1$
```

```
$\mathcal{P}(\mathtt{crs}, [\mathbf{x}]_1 = [\mathbf{M}]_1\mathbf{w}, \mathbf{w})$
return $\boldsymbol{\pi} := ([\mathbf{w}^\top \boldsymbol{\gamma}]_1 \in \mathbb{G}_1^{t+1}$

$\mathcal{V}(\mathtt{crs}, [\mathbf{x}]_1, \boldsymbol{\pi})$
$\hat{e}([\mathbf{x}^\top]_1, [\mathbf{C}]_2) \overset{?}{=} \hat{e}(\boldsymbol{\pi}, [\mathbf{A}]_2)$
```

Fig. 11: QA-NIZK proof system Π_{kw} under $\mathcal{D}_t$-KerMDH assumption

A designated-verifier ZK from a DVS-based SPHF (see A.4) for $\mathcal{L}_{\mathtt{lpar}}$ can be constructed as follows: the verifier first selects a key $\boldsymbol{\alpha} \in \mathbb{Z}_p^{\ell \times (t+1)}$, where t depends on the hardness assumption behind the soundness property. Next, the verifier sends $[\mathbf{M}^\top \boldsymbol{\alpha}]_1$ to the prover, who later computes and sends $\boldsymbol{\pi} = \mathbf{w}^\top [\mathbf{M}^\top \boldsymbol{\alpha}]_1$ to the verifier. Finally, the verifier checks if $[\mathbf{x}^\top]_1 \boldsymbol{\alpha} = \boldsymbol{\pi}$. Starting from this construction, Kiltz and Wee make it a publicly-verifiable QA-NIZK proof system in the CRS model by using pairing techniques as follows: the CRS includes $[\mathbf{M}^\top \boldsymbol{\alpha}]_1$ and $[\mathbf{A}, \boldsymbol{\alpha}\mathbf{A}]_2$ for a vector $\mathbf{A} \in \mathbb{Z}_p^{(t+1) \times t}$ chosen from a distribution $\mathcal{D}_t$. The proof remains the same as before, but the verification is the pairing check

$$\hat{e}([\mathbf{x}^\top]_1, [\boldsymbol{\alpha}\mathbf{A}]_2) \overset{?}{=} \hat{e}(\boldsymbol{\pi}, [\mathbf{A}]_2)$$

The soundness relies on the hardness of finding non-trivial cokernel elements of $\mathbf{A}$ and the smoothness of the underlying projective hash function (PHF). Also, for the right choice of the distribution for $\mathbf{A}$, the most efficient choice that the assumption is believed to hold is $t = 1$ which results in succinct proofs consisting of only two group elements. The protocol Π_{kw} is depicted in Fig. 11.

D.6 Impossibility of semantic extractor for Kiltz-Wee QA-NIZK.

We now prove that Π_{kw} cannot be semantic knowledge sound under the discrete logarithm assumption.

Theorem 7. *Let $\mathcal{L}_{\mathtt{lpar}}$ be a linear language over some cyclic group $\mathbb{G}_1$ with $\mathtt{lpar} = [\mathbf{M}]_1$. The QA-NIZK Π_{kw} depicted in Fig. 11 cannot be semantic knowledge sound under the DL assumption in $\mathbb{G}_1$.*

Proof. The proof is very similar to the proof of Lemma 2. Suppose Π_{kw} is semantic knowledge sound. Let $\mathcal{P}$ be the canonical prover adversary that on input a CRS $\mathtt{crs} = ([\boldsymbol{\gamma}]_1, [\mathbf{C}, \mathbf{A}]_2)$ and random coins s, r' proceeds as follows:

1. Sample the first component of the witness w_1 using random coins s.

2. Sample other components of $\mathbf{w}$ using random coins r'.
3. Compute $\mathbf{x} = [\mathbf{M}]_1\mathbf{w}$.
4. Compute $\boldsymbol{\pi} := ([\mathbf{w}^\top\boldsymbol{\gamma}]_1 \in \mathbb{G}_1^{t+1}$.
5. Return $([\mathbf{x}]_1, \boldsymbol{\pi})$.

Let f be the function of honest prover strategy that $\mathcal{P}$ uses to compute a valid proof. Namely, $f(\mathtt{crs}, (s, r')) = ([\mathbf{x}]_1, \boldsymbol{\pi})$. The semantic extractor $\mathsf{Ext}_{\mathcal{P}}$ given as input $([\mathbf{x}]_1, \boldsymbol{\pi}, \mathtt{td} = \boldsymbol{\alpha}; r)$ can output $\mathbf{w}$ with overwhelming probability. Note that $r = \perp$ as the prover is deterministic. A DL adversary $\mathcal{A}$ can now use this extractor to break the DL assumption. Specifically, $\mathcal{A}$, on a DL challenge $[\varrho]_1$ proceeds as follows:

1. Samples a group element $[\mathbf{x}]_1$ using randomness r' such that the first element of $[\mathbf{x}]_1$ is defined as $[x_1]_1 = [\varrho]_1$.
2. Selects a trapdoor $\mathtt{td} = \boldsymbol{\alpha}$ and computes an accepting proof $\boldsymbol{\pi} = [\mathbf{x}^\top]_1\boldsymbol{\alpha}$.
3. Invokes the extractor on $[\mathbf{x}]_1, \boldsymbol{\pi}$ who outputs $\mathbf{w}$. Return the first element w_1 of $\mathbf{w}$.

Now observe that $\mathcal{A}$ computes inputs of Ext_f exactly as an inefficient prover $\mathcal{P}^*$ for which the extraction is guaranteed. Hence, $\mathcal{A}$ computes the discrete logarithm $\varrho = w_1$ of $[\varrho]_1$ with the same probability that Ext_f succeeds.

□