Succinct Spooky Free Compilers Are Not Black Box Sound

Zvika Brakerski* Yael Kalai† Renen Perlman‡

Abstract

It is tempting to think that if we encrypt a sequence of messages $\{x_i\}$ using a semantically secure encryption scheme, such that each x_i is encrypted with its own independently generated public key pk_i, then even if the scheme is malleable (or homomorphic) then malleability is limited to acting on each x_i independently. However, it is known that this is not the case, and in fact even non-local malleability might be possible. This phenomenon is known as *spooky interactions*.

We formally define the notion of *spooky free compilers* that has been implicit in the delegation of computation literature. A spooky free compiler allows to encode a sequence of queries to a multi-prover interactive proof system (MIP) in a way that allows to apply the MIP prover algorithm on the encoded values on one hand, and prevents spooky interactions on the other. In our definition, the compiler is allowed to be tailored to a specific MIP.

We show that (under a plausible complexity assumption) spooky free compilers that are sufficiently succinct to imply delegation schemes for NP with communication n^α (for any constant $\alpha < 1$) cannot be proven secure via black-box reduction to a falsifiable assumption. On the other hand, we show that it is possible to construct *non-succinct spooky free fully homomorphic encryption*, the strongest conceivable flavor of spooky free compiler, in a straightforward way from any fully homomorphic encryption scheme.

Our impossibility result relies on adapting the techniques of Gentry and Wichs (2011) which rule out succinct adaptively sound delegation protocols. We note that spooky free compilers are only known to imply *non-adaptive* delegation, so the aforementioned result cannot be applied directly. Interestingly, we are still unable to show that spooky free compilers imply adaptive delegation, nor can we apply our techniques directly to rule out arbitrary non-adaptive NP-delegation.

1 Introduction

The PCP Theorem [AS98, ALM+98] is one of the most formidable achievements of computer science in the last decades. Probabilistically Checkable Proofs (PCPs) and Multi-Prover Interactive Proofs (MIPs) allow to reduce the communication complexity of verifying an NP statement to logarithmic in the input length (and linear in the security parameter), in a single round of communication. However, they require sending multiple queries to isolated non-colluding provers.[1] It is impossible (under plausible complexity assumptions) to achieve the same communication complexity with a

*Weizmann Institute of Science. Supported by the Israel Science Foundation (Grant No. 468/14), Binational Science Foundation (Grants No. 2016726, 2014276) and ERC Project 756482 REACT.

†Microsoft Research and Massachusetts Institute of Technology.

[1] We purposely refrain from distinguishing between a PCP, where multiple queries are made to a fixed proof string, and a single round MIP, where there are multiple provers. The difference is insignificant for the purpose of our exposition and the two forms are often equivalent.

single computationally unbounded prover. However, if we only require computational soundness this may be possible.

Indeed, it has been shown by Micali [Mic94] and Damgård et al. and Bitansky et al. [DFH12, BCCT12,BCCT13,BCC+14] that in the random oracle model, or relying on knowledge assumptions, it is indeed possible. However, in the standard model and under standard hardness assumptions (in particular falsifiable [Nao03]), this is not known. Gentry and Wichs [GW11] showed that if adaptive security is sought, i.e. if the adversary is allowed to choose the NP instance after seeing the challenge message from the verifier, then soundness cannot be proved under any falsifiable assumption, so long as the security reduction uses the adversary as a black-box, and relying on the existence of sufficiently hard languages in NP. This still leaves open the possibility of non-adaptive protocols which seems to be beyond the reach of the techniques of [GW11].[2]

A notable attempt to construct such a protocol was made by Biehl, Meyer and Wetzel [BMW98], and by Aiello et al. [ABOR00]. They suggested to generate MIP queries and encode them using independent instances of a private information retrieval (PIR) scheme. Intuitively, since each query is encoded separately, it should be impossible to use the content of one encoding to effect another. However, as Dwork et al. [DLN+01] showed, the provable guarantees of PIR (or semantically secure encryption) are insufficient to imply the required soundness. They showed that semantic security does not preclude non-local *spooky interactions* which cannot be simulated by independent provers.

Dodis et al. [DHRW16] recently showed that there exist explicit secure PIR schemes (under widely believed cryptographic assumptions) that actually exhibit spooky interactions, and thus fail the [BMW98, ABOR00] approach. They complemented this negative result with a construction of a *spooky free* fully homomorphic encryption (FHE) scheme, which is an FHE scheme with the additional guarantee that if multiple inputs are encrypted using independently generated public keys, then any operation on the collection of ciphertexts can be simulated by independent processes applied to each encrypted message separately. In particular, a spooky free FHE has strong enough security guarantees to allow proving the [BMW98, ABOR00] approach, since a single computationally bounded prover "has no choice" but to behave like a collection of isolated provers as is required for MIP soundness. However, the spooky free encryption scheme constructed by Dodis et al. relies on knowledge assumptions, the same knowledge assumptions that imply short computationally sound proofs (and in fact uses such proofs as building blocks).

Our Results. In this work, we notice that spooky free FHE is a flavor of a more general notion that we call *spooky free compiler*. This notion has been implicit in previous works since [BMW98, ABOR00]. A spooky free compiler provides a way to encode and decode a set of queries in such a way that any operation on an encoded set, followed by decoding, is equivalent to performing an independent process on each of the queries separately. In addition, for functionality purposes, it should be possible to apply the MIP prover algorithm on encoded queries. This notion generalizes much of the research efforts in providing a proof for [BMW98, ABOR00]-style protocols. In particular, spooky free FHE can be viewed as a *universal* spooky free compiler that is applicable to all MIPs.

We show that spooky free compilers cannot have succinct encodings if they are proven based on a falsifiable hardness assumption using a reduction that uses the adversary as black-box. Our negative result holds for any compiler where the encoding is succinct enough to imply a delegation scheme

[2] Extending the black-box impossibility to non-adaptive delegation is a well motivated goal by itself and has additional implications, e.g. for the study of program obfuscation.

with sub-linear communication complexity. We note that this does not follow from [GW11] since spooky free compilers are only known to imply non-adaptive delegation protocols whereas [GW11] only rules out adaptive protocols.

On the other hand, we show that if succinctness is not imposed, then it is straightforward to achieve spooky free FHE based on the existence of any FHE scheme. Namely, spooky free compilation in its strongest sense becomes trivial. Specifically, we present a scheme where the encoding size corresponds to the size of the query space for the MIP, i.e. the length of the truth table of the MIP provers.[3]

Other Related Works. Kalai, Raz and Rothblum [KRR13, KRR14] showed that the [BMW98, ABOR00] approach is in fact applicable and sound when using *no signaling MIP*. These are proof systems that remain sound even when spooky interactions are allowed. However, such MIPs can only be used to prove statements for P and not for all of NP unless NP=P.

1.1 Overview of Our Techniques

We provide an overview of our techniques. For this outline we only require an intuitive understanding of the notion of spooky free compiler as we tried to convey above. The formal definition appears in Section 3.

Ruling Out Succinct Compilers. Our method for ruling out succinct compilers draws from the [GW11] technique for showing the impossibility of reductions for adaptively secure delegation schemes, i.e. ones where the instance x can be chosen after the encoded MIP queries are received. At a high level, [GW11] produce an adversary that chooses instances x that are not in the NP language in question, but are computationally indistinguishable from ones that are in the language. This allows to simulate accepting short delegation responses for those x's using a brute force process, since the complexity of the exhaustive process is still insufficient to distinguish whether x is in the language or not (this argument makes use of the dense model theorem [DP08, RTTV08, VZ13]). The crucial property that is required is that each x is only used once, since otherwise the combined complexity of applying the brute force process many times will not allow us to rely on the computational indistinguishability. The adaptive setting allows to choose a new x for each query, and thus to apply this argument.

We notice that a spooky free compiler is similar to an adaptive delegation protocol, since it does not preclude the adversary from using a fresh x for each set of queries. We will consider an adversary that samples x not in the language similarly to [GW11], but instead of performing the MIP evaluation on the encoded queries it uses the dense model theorem to produce an accepting response.

We would like to then argue that this adversary breaks the spooky-freeness, since it cannot be simulated by a sequence of local operations on the queries due to the unconditional soundness of the MIP. However, we need to be rather careful here, since an attempt to simulate will only fail w.r.t. a distinguisher who knows x (otherwise the soundness of the MIP is meaningless). It may seem that this can be handled by giving x to the distinguisher together with the MIP answers, e.g. by considering an additional "dummy MIP prover" that always returns x, so that x is now sent

[3] We note that we have neither positive nor negative results for compilers (or delegation protocols) with communication complexity (super-)linear in the instance size but sub-linear in the MIP truth table length.

together with the MIP answers. Alas, this approach seems to fail, since a simulator can simulate the adversary by using x in the language, and answering the queries locally. The dense model theorem implies that the two views are indistinguishable, which in turn implies that this adversary does not break the spooky freeness.

We overcome this obstacle by confining the adversary to choose x from a small bank $\overline{X}$ of randomly chosen x's that are not in the language, and are a priori sampled and hardwired to the adversary's code. We consider a distinguisher that also has this bank $\overline{X}$ hardwired into its code, and will output 1 if and only if the answers are accepting with respect to *some* $x \in \overline{X}$. This requires a definition of spooky free compiler relative to auxiliary input (which intuitively specifies the $x \in \overline{X}$ that the adversary chose). We show that this auxiliary input notion is implied by spooky free FHE, and that the required auxiliary input is very short (specifically independent of the length of x). We denote this adversary and distinguisher pair by $(\overline{\mathcal{A}}, \overline{\Psi})$, and use the soundness of the MIP to argue that the distinguisher $\overline{\Psi}$ can distinguish between the adversary $\overline{\mathcal{A}}$ and any local process, which implies that $(\overline{\mathcal{A}}, \overline{\Psi})$ break the spooky freeness.

The fact that $(\overline{\mathcal{A}}, \overline{\Psi})$ break spooky freeness implies that the black-box reduction breaks the assumption given oracle access to $(\overline{\mathcal{A}}, \overline{\Psi})$.[4] We reach a contradiction by showing efficient (probabilistic polynomial time) algorithms $(\mathcal{A}, \Psi)$ which are indistinguishable from $(\overline{\mathcal{A}}, \overline{\Psi})$ in the eyes of the reduction, which implies that the underlying assumption is in fact solvable in probabilistic polynomial time.

See Section 5 for the full details of this negative result.

Straightforward Non-Succinct Spooky Free FHE. We show that any FHE scheme with message space Σ, implies a spooky free FHE scheme with message space Σ and ciphertext size $\approx |\Sigma|$. We explain the construction for $\Sigma = \{0,1\}$, the extension to the general case is fairly straightforward, and we refer the reader to Section 4 for the full details.

Our starting point is an FHE scheme with message space $\{0,1\}$. Our spooky free scheme is essentially an *equivocable* variant of the FHE scheme, namely one where there is a special ciphertext that can be explained as either an encryption of 0 or an encryption of 1 given an appropriate secret key. Formally, the spooky free key generation generates two key sets for the FHE scheme: $(\mathsf{fhepk}_0, \mathsf{fhesk}_0)$, $(\mathsf{fhepk}_1, \mathsf{fhesk}_1)$, it also flips a coin $b \xleftarrow{\$} \{0,1\}$. Finally it outputs the spooky free key pair: $\mathsf{sfpk} = (\mathsf{fhepk}_0, \mathsf{fhepk}_1)$ and $\mathsf{sfsk} = (b, \mathsf{fhesk}_b)$. To encrypt, encrypt the same message with both fhepk's to obtain $c' = (c_0, c_1)$. Homomorphic evaluation can be performed on c_0, c_1 independently, and since both components of the ciphertext will always encrypt the same value, then decrypting with fhesk_b will be correct regardless of the value of b. Note that the size of the ciphertext blew up by a factor of $|\Sigma| = 2$.

To show that the scheme is spooky free, we notice that it is possible to generate an equivocable ciphertext $c^* = (\mathsf{Enc}_{\mathsf{fhepk}_0}(\beta), \mathsf{Enc}_{\mathsf{fhepk}_1}(\bar{\beta}))$, for a random $\beta \xleftarrow{\$} \{0,1\}$. Note that for $b = \beta \oplus x$, it holds that $\mathsf{sfsk}_x = (b, \mathsf{fhesk}_b)$ decrypts c^* to the value x, and furthermore, the joint distribution $(\mathsf{sfpk}, \mathsf{sfsk}_x, c^*)$ is computationally indistinguishable from the case where b was chosen randomly and c^* was a proper encryption of x.

To see why this scheme is spooky free, we consider an adversary that receives a number of ciphertexts under independently generated sfpk's and attempts to perform some non-local spooky

[4] In fact, the situation is more delicate since $(\overline{\mathcal{A}}, \overline{\Psi})$ is actually a *distribution* over adversaries and distinguishers, where the distribution is over the choice of the bank $\overline{X}$. We argue that almost all $(\overline{\mathcal{A}}, \overline{\Psi})$ break the spooky freeness, and then prove that the average advantage is also non-negligible (see Lemma 2.6 in Section 2).

interaction. Namely, the adversary takes $\{\mathsf{sfpk}_i, c'_i = \mathsf{Enc}_{\mathsf{sfpk}_i}(x_i)\}_i$, performs some operation to produce $\{\tilde{c}_i\}_i$ s.t. when decrypting $y_i = \mathsf{Dec}_{\mathsf{sfsk}_i}(\tilde{c}_i)$, the entries y_i should be distributed in a way that cannot be simulated locally by operating on each x_i independently. We will show that this is impossible and in fact there is a local way to generate the y_i values, up to computational indistinguishability.

To this end, we first consider a setting where instead of c'_i, we feed the adversary with the equivocable ciphertext c^*_i. Recall that the value x_i that c^*_i encrypts is determined by sfsk and not by c^* itself. Still, as we explained above, the distribution of (public key, secret key, ciphertext) is indistinguishable from the previous one. Therefore, in this experiment the adversary should return a computationally indistinguishable distribution over the y_i's as it did before. However, notice that now the adversary's operation does not depend on the x_i's at all. Namely, it is possible to decide on the value of x_i only at decryption time and not at encryption time, and it is possible to do so for each i independently (by selecting an appropriate value for b in the i'th instantiation of the scheme). It follows that the distribution of y_i in this experiment, which is computationally indistinguishable from the original one, is spooky free in the sense that it can be generated by executing a local process on each x_i to compute y_i.[5]

2 Preliminaries

Definition 2.1. *Two distributions $\mathcal{X}, \mathcal{Y}$ are said to be $(\epsilon(\lambda), s(\lambda))$-*$\mathsf{indistinguishable}$ *if for every distinguisher Ψ of size $\mathrm{poly}(s(\lambda))$ it holds that*

$$|Pr\left[\Psi(\mathcal{X}) = 1\right] - Pr\left[\Psi(\mathcal{Y}) = 1\right]| \leq \epsilon(\lambda) \ .$$

We say that the distributions $\mathcal{X}, \mathcal{Y}$ are α-$\mathsf{sub\text{-}exponentially}$ *indistinguishable if they are $(2^{-n^\alpha}, 2^{n^\alpha})$-indistinguishable.*

Lemma 2.1 (Borel-Cantelli)**.** *For any sequence of events $\{E_\lambda\}_{\lambda\in\mathbb{N}}$, if the sum of the probabilities of E_λ is finite, i.e. $\sum_{\lambda\in\mathbb{N}} \Pr[E_\lambda] < \infty$, then the probability that infinitely many of them occur is 0.*

Definition 2.2 (One-Round Multi-Prover Interactive Proofs (MIP))**.** *Let R be an NP relation, and let L be the induced language. A one-round p-prover interactive proof for L is a triplet of* PPT *algorithms $\Pi = (\mathcal{G}, (\mathcal{P}_1, \ldots, \mathcal{P}_p), \mathcal{V})$ as follows:*

- ***Query Generation*** $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$: *Outputs a set of queries $\vec{q} = (q_1, \ldots, q_p)$ for the provers.*
- ***Provers*** $a_i \leftarrow \mathcal{P}_i(q_i, x, w)$: *Given the query corresponding to the i'th prover, outputs an answer a_i for x using the query q_i, the instance x and its witness w.*
- ***Verifier*** $b \leftarrow \mathcal{V}(\vec{q}, \vec{a}, x)$: *Using the set of queries $\vec{q}$ with matching answers $\vec{a}$ and the instance x outputs a bit b.*

We require that there is a soundness parameter $\sigma > 0$ such that $\sigma(\kappa) < 1 - 1/\mathrm{poly}(\kappa)$, for which the following two properties hold:

[5] A meticulous reader may have noticed that it is required that for all i the local process uses the same sequence of c^*_i. Indeed the definition of spooky freeness allows the provers to pre-share a joint state.

- ***Completeness:*** *For every* $(x, w) \in R$ *such that* $x \in \{0,1\}^{\leq 2^\kappa}$,

$$\Pr\left[\mathcal{V}(\vec{q}, \vec{a}, x) = 1\right] = 1 \ ,$$

where $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$, $\vec{a} = (a_1, \ldots, a_p)$ *and* $a_i \leftarrow \mathcal{P}_i(q_i, x, w)$ *for every* $i \in [p]$.

- ***Soundness:*** *For every* $x \in \{0,1\}^{\leq 2^\kappa} \setminus L$ *and for every (not necessarily efficient) cheating provers* $\mathcal{P}'_1, \ldots, \mathcal{P}'_p$ *the following holds:*

$$Pr\left[\mathcal{V}(\vec{q}, \vec{a'}, x) = 1\right] < \sigma(\kappa) \ ,$$

where $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$, $\vec{a'} = (a'_1, \ldots, a'_p)$ *and* $a'_i \leftarrow \mathcal{P}'_i(q_i, x)$ *for every* $i \in [p]$.

Definition 2.3. *An* **NP** *language* $L \subset \{0,1\}^*$, *is said to have* sub-exponentially hard subset-membership problem $(\mathcal{L}, \overline{\mathcal{L}}, \mathsf{Sam})$ *if the following holds:*

- $\mathcal{L} = \{\mathcal{L}_n\}_{n\in\mathbb{N}}$ *is a* PPT *distribution ensemble, each over* $L \cap \{0,1\}^n$.
- $\overline{\mathcal{L}} = \{\overline{\mathcal{L}}_n\}_{n\in\mathbb{N}}$ *is a* PPT *distribution ensemble, each over over* $\overline{L} \cap \{0,1\}^n = \{0,1\}^n \setminus L$.
- Sam *is a* PPT *algorithm, that on input* 1^n *outputs a tuple* $(x, w) \in R_L$ *where* x *is distributed as in* $\mathcal{L}_n$.
- $\mathcal{L}, \overline{\mathcal{L}}$ *are* $(\epsilon(n), s(n))$*-indistinguishable for* $\epsilon(n) = 1/2^{n^\alpha}$, $s(n) = 2^{n^\alpha}$, *where* $\alpha > 0$ *is some constant referred to the* hardness-parameter.

In such case we will say that $(\mathcal{L}, \overline{\mathcal{L}}, \mathsf{Sam})$ *is* α-sub-exponentially hard.

Theorem 2.2 (Dense Model Theorem [VZ13, Lemma 6.9])**.** *There exists a fixed polynomial* p *such that the following holds: Let* $\mathcal{X}$ *and* $\mathcal{Y}$ *be two* $(\epsilon(\lambda), s(\lambda))$*-indistinguishable distributions. Let* $\mathcal{A}$ *be a distribution over* $\{0,1\}^\ell$ *jointly distributed with* $\mathcal{X}$. *Then there exists a (probabilistic) function* $h : \mathcal{Y} \to \{0,1\}^\ell$ *such that* $(\mathcal{X}, \mathcal{A})$ *and* $(\mathcal{Y}, h(\mathcal{Y}))$ *are* $(\epsilon^*(\lambda), s^*(\lambda))$*-indistinguishable, where* $\epsilon^*(\lambda) = 2 \cdot \epsilon(\lambda)$ *and* $s^*(\lambda) = s(\lambda) \cdot p(\epsilon(\lambda), 1/2^{\ell(\lambda)})$.

Corollary 2.3. *Let* $(\mathcal{X}, \mathcal{A})$ *be a joint distribution s.t.* $\mathcal{A}$ *is supported over* $\{0,1\}^\ell$ *for* $\ell = O(n^\alpha)$, *and let* $\mathcal{Y}$ *be a distribution such that* $\mathcal{X}$ *and* $\mathcal{Y}$ *are* α*-sub-exponentially indistinguishable. Then there exists a probabilistic function* h *s.t.* $(\mathcal{X}, \mathcal{A})$ *and* $(\mathcal{Y}, h(\mathcal{Y}))$ *are* $(2 \cdot 2^{-n^\alpha}, 2^{n^\alpha})$ *indistinguishable.*

Proof. Let $\epsilon(n) = 2^{-n^\alpha}$, $s(n) = 2^{n^\alpha}$ be such that $\mathcal{X}, \mathcal{Y}$ are $(\epsilon(n), s(n))$-indistinguishable. Then it follows from Definition 2.1 that they are also $(\epsilon(n), s'(n))$-indistinguishable for any $s'(n) = \text{poly}(s(n))$, in particular let $s'(n) = s(n)/p(\epsilon, 1/2^\ell) = 2^{O(n^\alpha)} = \text{poly}(s(n))$. Theorem 2.2 implies that there exists a probabilistic function h s.t. $(\mathcal{X}, \mathcal{A})$ and $(\mathcal{Y}, h(\mathcal{Y}))$ are $(2\epsilon(n), s(n))$-indistinguishable.

□

Definition 2.4 (fully-homomorphic encryption)**.** *A fully-homomorphic (public-key) encryption scheme* $\mathsf{FHE} = (\mathsf{FHE.Keygen}, \mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$ *is a 4-tuple of* PPT *algorithms as follows (*λ *is the security parameter):*

- ***Key generation*** $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Keygen}(1^\lambda)$***:*** *Outputs a public encryption key* pk *and a secret decryption key* sk.

- ***Encryption*** $c \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, \mu)$***:*** *Using the public key* pk*, encrypts a single bit message* $\mu \in \{0,1\}$ *into a ciphertext* c*.*
- ***Decryption*** $\mu \leftarrow \mathsf{FHE.Dec}(\mathsf{sk}, c)$***:*** *Using the secret key* sk*, decrypts a ciphertext* c *to recover the message* $\mu \in \{0,1\}$*.*
- ***Homomorphic evaluation*** $\widehat{c} \leftarrow \mathsf{FHE.Eval}(\mathcal{C}, (c_1, \ldots, c_\ell), \mathsf{pk})$***:*** *Using the public key* pk*, applies a boolean circuit* $\mathcal{C} : \{0,1\}^\ell \to \{0,1\}$ *to* $c_1, \ldots, c_\ell$*, and outputs a ciphertext* $\widehat{c}$*.*

A homomorphic encryption scheme is said to be secure if it is semantically secure.
A scheme FHE *is fully homomorphic, if for any circuit* $\mathcal{C}$ *and any set of inputs* $\mu_1, \ldots, \mu_\ell$*, letting* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Keygen}(1^\lambda)$ *and* $c_i \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, \mu_i)$*, it holds that*

$$\Pr\left[\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{FHE.Eval}(\mathcal{C}, (c_1, \ldots, c_\ell), \mathsf{pk})) \neq \mathcal{C}(\mu_1, \ldots, \mu_\ell)\right] = \mathrm{negl}(\lambda) \ ,$$

A fully homomorphic encryption scheme is compact if the output length of $\mathsf{FHE.Eval}$ *is a fixed polynomial in* λ *(and does not depend on the length of* $\mathcal{C}$*).*

2.1 Spooky-Free Encryption

The following is adopted and adapted from [DHRW16] (see discussion below). Let $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ be a public-key encryption scheme. Let $\mathcal{D}$ be some distribution and let $\mathcal{A}$ and $\mathcal{S}$ be some algorithms. Consider the following experiments:

$\mathbf{REAL}_{\mathcal{D},\mathcal{A}}(1^\kappa)$

1. Sample messages and auxiliary information $(\vec{m}, \alpha) = (m_1, \ldots, m_n, \alpha) \leftarrow \mathcal{D}(1^\kappa)$.
2. Generate keys and encryptions for every $i \in [n]$: set $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{PKE.KeyGen}(1^\kappa)$ and $c_i \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_i, m_i)$.
3. Evaluate $\vec{c'} \leftarrow \mathcal{A}(1^\kappa, \vec{\mathsf{pk}}, \vec{c})$.
4. Decrypt each evaluated ciphertext $m'_i := \mathsf{PKE.Dec}(\mathsf{sk}_i, c'_i)$.
5. Output $(\vec{m}, \vec{m'}, \alpha)$.

$\mathbf{SIM}_{\mathcal{D},\mathcal{S}}(1^\kappa)$

1. Sample messages and auxiliary information $(\vec{m}, \alpha) = (m_1, \ldots, m_n, \alpha) \leftarrow \mathcal{D}(1^\kappa)$.
2. Sample random coins r for the simulator $\mathcal{S}$, and evaluate for every $i \in [n]$ $m'_i \leftarrow \mathcal{S}(1^\kappa, 1^n, i, m_i; r)$.
3. Output $(\vec{m}, \vec{m'}, \alpha)$.

Definition 2.5. *Let* $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Enc})$ *be a public-key encryption scheme. We say that* PKE *is* $\mathsf{strongly\ spooky - free}$ *if there exists a* PPT *simulator* $\mathcal{S}$ *such that for every* PPT *adversary* $\mathcal{A}$*, distribution* $\mathcal{D}$ *and distinguisher* Ψ*, the following holds:*

$$\left| \Pr\left[\Psi(\vec{m}, \vec{m'}, \alpha) = 1 \mid (\vec{m}, \vec{m'}, \alpha) \leftarrow \boldsymbol{REAL}_{\mathcal{D},\mathcal{A}}(1^\kappa)\right] - \Pr\left[\Psi(\vec{m}, \vec{m'}, \alpha) = 1 \mid (\vec{m}, \vec{m'}, \alpha) \leftarrow \boldsymbol{SIM}_{\mathcal{D},\mathcal{S}^{\mathcal{A}}}(1^\kappa)\right] \right| = \mathrm{negl}(\kappa)$$

We say that PKE *is* weakly spooky − free *if the simulator can be chosen after the adversary, the distribution and the distinguisher have been set. Similarly, we say that* PKE *is* strongly spooky-free without auxiliary information *(weakly spooky-free without auxiliary information), if it is strongly spooky-free (weakly spooky-free), and the distribution $\mathcal{D}$ must output $\alpha = \perp$.*

For our negative result, we prove the impossibility with respect to the *weak* definition *without auxiliary information*, thus strengthening the impossibility result. On the other hand, for the positive result we construct a *strongly spooky-free (with auxiliary information)* scheme. We note that in the original definition in [DHRW16] the order of quantifiers was somewhere "in between" our two definitions. They allowed the simulator to be chosen after seeing the adversary $\mathcal{A}$, but before seeing the distribution $\mathcal{D}$ and the distinguisher Ψ.

2.2 Falsifiable Assumptions and Black-Box Reductions

In what follows, we recall the notion of falsifiable assumptions as defined by Naor [Nao03]. We follow the formalization of Gentry and Wichs [GW11]. We also show here a variety of properties that we use for our proof. We present a notion of equivalence between falsifiable assumptions and prove that any falsifiable assumption is equivalent, under this notion, to one with a special form. Finally we show a claim about the asymptotic behavior of the average advantage of a *distribution* of successful adversaries against a falsifiable assumption.

Definition 2.6 (falsifiable assumption)**.** *A* falsifiable assumption *consists of a* PPT *interactive challenger $\mathcal{C}(1^\lambda)$ and a constant $\eta \in [0, 1)$. The challenger $\mathcal{C}$ interacts with a machine $\mathcal{A}$ and may output a special symbol* win*. If this occurs, $\mathcal{A}$ is said to win $\mathcal{C}$. For any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ over $\mathcal{C}$ is defined as:*

$$\mathsf{Adv}_{\mathcal{A}}^{(\mathcal{C},\eta)}(1^\lambda) = \Pr[\mathcal{A}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] - \eta,$$

where the probability is taken over the random coins of $\mathcal{A}$ and $\mathcal{C}$. The assumption associated with the tuple $(\mathcal{C}, \eta)$ states that for every (non-uniform) adversary $\mathcal{A}(1^\lambda)$ running in polynomial time,

$$\mathsf{Adv}_{\mathcal{A}}^{(\mathcal{C},\eta)}(1^\lambda) = \mathrm{negl}(\lambda).$$

If the advantage of $\mathcal{A}$ is non-negligible in λ then $\mathcal{A}$ is said to break *the assumption.*

Definition 2.7. *A falsifiable assumption $(\mathcal{C}_1, \eta_1)$ is* black-box stronger *than a falsifiable assumption $(\mathcal{C}_2, \eta_2)$, denoted $(\mathcal{C}_1, \eta_1) \geq (\mathcal{C}_2, \eta_2)$ if there exists a reduction $\mathcal{R}$ such that for every adversary $\mathcal{A}$ with non-negligible advantage against $(\mathcal{C}_2, \eta_2)$, it holds that $\mathcal{R}^{\mathcal{A}}$ has non-negligible advantage against $(\mathcal{C}_1, \eta_1)$.*

We say that $(\mathcal{C}_1, \eta_1)$ and $(\mathcal{C}_2, \eta_2)$ are black-box equivalent*, denoted $(\mathcal{C}_1, \eta_1) \equiv (\mathcal{C}_2, \eta_2)$, if $(\mathcal{C}_1, \eta_1) \geq (\mathcal{C}_2, \eta_2)$ and $(\mathcal{C}_2, \eta_2) \geq (\mathcal{C}_1, \eta_1)$.*

Definition 2.8. *Let $(\mathcal{C}, \eta)$ be a falsifiable assumption, and define the challenger $\mathcal{C}_\eta^\otimes$ that interacts with an adversary $\mathcal{A}$ as follows. First $\mathcal{A}$ sends a polynomially bounded unary number 1^t to the challenger. Then the challenger executes the $\mathcal{C}$ game with $\mathcal{A}$ sequentially and independently t times. Finally $\mathcal{C}_\eta^\otimes$ declares that $\mathcal{A}$ won if and only if $\mathcal{A}$ won in at least $\lceil \eta t \rceil + 1$ of the games.*

Lemma 2.4. *For any falsifiable assumption $(\mathcal{C}, \eta)$ it holds that $(\mathcal{C}, \eta) \equiv (\mathcal{C}_\eta^\otimes, 0)$.*

Proof. Let $\mathcal{A}$ be an adversary with non-negligible advantage δ in $(\mathcal{C}, \eta)$. Then $\mathcal{R}^{\mathcal{A}}(1^\lambda)$ is an adversary against $\mathcal{C}_\eta^\otimes$ as follows. It starts by sending 1^t for $t = \lceil \lambda/\delta \rceil$ in the first message. Then for every iteration it simply executes $\mathcal{A}$. By definition, the expected number of wins is at least $\lfloor \eta t + \lambda \rfloor > \lceil \eta t \rceil + 1 + \lambda/2$. By a Chernoff argument the probability to win against $\mathcal{C}_\eta^\otimes$ is at least $1 - \mathrm{negl}(\lambda)$.[6]

Now let $\mathcal{A}$ be an adversary with non-negligible advantage δ against $(\mathcal{C}_\eta^\otimes, 0)$. Then $\mathcal{R}^{\mathcal{A}}(1^\lambda)$ is an adversary against $(\mathcal{C}, \eta)$ as follows. It simulates $\mathcal{C}_\eta^\otimes$ for $\mathcal{A}$ by first reading 1^t, then sampling $i^* \xleftarrow{\$} [t]$, simulating $\mathcal{C}$ in all iterations except i^*, and in iteration i^* forward messages back and forth to the real challenger. By definition the advantage of $\mathcal{R}^{\mathcal{A}}(1^\lambda)$ is at least $1/t$ which is noticeable. □

Definition 2.9 (black box reduction)**.** *We say that the security of a scheme* Π *can be proven via a* black-box reduction *to a falsifiable assumption, if there is an oracle-access machine* $\mathcal{R}$ *such that for every (possibly inefficient) adversary* $\mathcal{A}$ *that breaks the security of* Π*, the oracle machine* $\mathcal{R}^{\mathcal{A}}$ *runs in time* $\mathrm{poly}(\lambda)$ *and breaks the assumption.*

Corollary 2.5. *If* Π *can be proven via a black-box reduction to a falsifiable assumption* $(\mathcal{C}, \eta)$ *then it can also be proven via a black-box reduction to a falsifiable assumption* $(\mathcal{C}', 0)$*, and furthermore if* $(\mathcal{C}, \eta)$ *is hard for all polynomial adversaries then so is* $(\mathcal{C}', 0)$*.*

Proof. Letting $\mathcal{C}' = \mathcal{C}_\eta^\otimes$, the corollary directly follows from Lemma 2.4 and Definition 2.9. □

Lemma 2.6. *Let* Π *be a scheme whose security can be proven via a* black-box reduction $\mathcal{R}$ *to a falsifiable assumption* $(\mathcal{C}, 0)$ *(note that* $\eta = 0$*). Let* $\widetilde{\mathcal{A}}$ *be a distribution on adversaries such that with probability* 1*,* $\mathcal{A} \xleftarrow{\$} \widetilde{\mathcal{A}}$ *breaks the security of* Π*. Then there exists a non-negligible* δ *such that*

$$\Pr_{\mathcal{A},\mathcal{R},\mathcal{C}}[\mathcal{R}^{\mathcal{A}}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \geq \delta(\lambda) \ .$$

Namely, the expected advantage of $\mathcal{R}$ *against* $(\mathcal{C}, 0)$ *is non-negligible.*

Proof. For every $\mathcal{A}$ denote:

$$\tilde{\delta}_{\mathcal{A}}(\lambda) = \Pr_{\mathcal{R},\mathcal{C}}[\mathcal{R}^{\mathcal{A}}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \ .$$

By the correctness of the reduction $\mathcal{R}$ we are guaranteed that with probability 1 over $\mathcal{A} \xleftarrow{\$} \widetilde{\mathcal{A}}$, it holds that $\tilde{\delta}_{\mathcal{A}}$ is a non-negligible function. Furthermore, notice that by definition

$$\Pr_{\mathcal{A},\mathcal{R},\mathcal{C}}[\mathcal{R}^{\mathcal{A}}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] = \mathop{\mathbb{E}}_{\mathcal{A}}[\tilde{\delta}_{\mathcal{A}}(\lambda)] \ ,$$

and our goal therefore is to prove that $\mathbb{E}_{\mathcal{A}}[\tilde{\delta}_{\mathcal{A}}(\lambda)]$ is non-negligible.

Let us consider a random $\mathcal{A}^* \xleftarrow{\$} \widetilde{\mathcal{A}}$ and define $\tilde{\delta}^*(\lambda) = \tilde{\delta}_{\mathcal{A}^*}(\lambda)$. We define a sequence of events $\{E_\lambda\}_{\lambda \in \mathbb{N}}$, where E_λ is the event that

$$\Pr_{\mathcal{A}}[\tilde{\delta}_{\mathcal{A}^*}(\lambda) \leq \tilde{\delta}_{\mathcal{A}}(\lambda)] \leq 1/\lambda^2.$$

Trivially, $\Pr[E_\lambda] \leq 1/\lambda^2$. Therefore, by the Borel-Cantelli Lemma, with probability 1 on the choice of $\mathcal{A}^*$ it holds that only finitely many of the events E_λ can occur.

[6] We assumed that δ is known to the reduction, which could be viewed as non-black-box access. However, note that δ can be estimated by running the oracle many times, simulating $\mathcal{C}$.

Let us consider some value of λ for which E_λ does not hold (as explained above, this includes all but finitely many λ values). That is, where

$$\Pr_{\mathcal{A}}[\tilde{\delta}_{\mathcal{A}^*}(\lambda) \leq \tilde{\delta}_{\mathcal{A}}(\lambda)] > 1/\lambda^2.$$

By definition, for these values, we can apply the Markov inequality

$$\mathop{\mathbb{E}}_{\mathcal{A}}[\tilde{\delta}_{\mathcal{A}}(\lambda)] \geq \Pr_{\mathcal{A}}[\tilde{\delta}_{\mathcal{A}^*}(\lambda) \leq \tilde{\delta}_{\mathcal{A}}(\lambda)] \cdot \tilde{\delta}_{\mathcal{A}^*}(\lambda) > \tilde{\delta}_{\mathcal{A}^*}(\lambda)/\lambda^2 \ .$$

Since with probability 1 it holds that both $\tilde{\delta}_{\mathcal{A}^*}(\lambda)$ is noticeable and that only finitely many of the E_λ can occur, then obviously there exists $\mathcal{A}^*$ for which both hold, which implies that indeed $\Pr_{\mathcal{A},\mathcal{R},\mathcal{C}}[\mathcal{R}^{\mathcal{A}}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]$ is non-negligible. □

3 Spooky-Free Compiler

Definition 3.1 (Spooky-Free Compiler)**.** *Let $\Pi = (\mathcal{G}, \vec{\mathcal{P}}, \mathcal{V})$ be a p-provers, one-round MIP with soundness σ for an* **NP** *language L with an induced relation R. A* Spooky-Free Compiler *for Π is a triplet of* PPT *algorithms* $\mathsf{SFC} = (\mathsf{SFC.Enc}, \mathsf{SFC.Dec}, \mathsf{SFC.Eval})$ *as follows:*

- ***Encoding*** $(e, \mathsf{dk}) \leftarrow \mathsf{SFC.Enc}(1^\kappa, \vec{q})$: *Outputs an encoding of the queries, and a decoding-key.*
- ***Evaluation*** $e' \leftarrow \mathsf{SFC.Eval}(e, x, w)$: *Evaluates the MIP answers on the encoded queries, instance x, and witness w.*
- ***Decoding*** $\vec{a} \leftarrow \mathsf{SFC.Dec}(e', \mathsf{dk})$: *Decodes the evaluated queries using the decoding-key.*

We require the following properties:

- ***Completeness*** *For every $(x, w) \in R$ such that $x \in \{0,1\}^{\leq 2^\kappa}$, the following holds: Sample queries $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$ and encode $(e, \mathsf{dk}) \leftarrow \mathsf{SFC.Enc}(\vec{q})$. Evaluate $e' \leftarrow \mathsf{SFC.Eval}(e, x, w)$, and decode $\vec{a} \leftarrow \mathsf{SFC.Dec}(e', \mathsf{dk})$. Then,*

$$\Pr[\mathcal{V}(\vec{q}, \vec{a}, x) = 1] = 1 \ .$$

- ***Spooky-Freeness*** *Define the following experiments:*

$\mathbf{REAL}_{\mathcal{A}}(1^\kappa)$	$\mathbf{SIM}_{\mathcal{S}}(1^\kappa)$
1. Sample queries $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$.	1. Sample queries $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$.
2. Encode $(e, \mathsf{dk}) \leftarrow \mathsf{SFC.Enc}(1^\kappa, \vec{q})$.	2. Sample random coins r, and using these coins simulate the auxiliary information $z \leftarrow \mathcal{S}(1^\kappa, 1^p, 0, 0; r)$.[7]
3. Evaluate $(e', z) \leftarrow \mathcal{A}(1^\kappa, e)$, where z is some auxiliary information that is passed to the distinguisher.	3. Using the same coins, compute $a_i \leftarrow \mathcal{S}(1^\kappa, 1^p, i, q_i; r)$ for all $i \in [p]$.
4. Decode $\vec{a} \leftarrow \mathsf{SFC.Dec}(e', \mathsf{dk})$.	4. Output $(z, \vec{q}, \vec{a})$.
5. Output $(z, \vec{q}, \vec{a})$.	

[7] For notational convenience this is like applying $\mathcal{S}$ with $i = 0$.

We say that SFC *is* strongly spooky-free *if there exists a* PPT *simulator* $\mathcal{S}$ *such that for every* PPT *adversary* $\mathcal{A}$ *the experiments* $\boldsymbol{REAL}_{\mathcal{A}}(1^\kappa)$ *and* $\boldsymbol{SIM}_{\mathcal{S}^{\mathcal{A}}}(1^\kappa)$ *are computationally-indistinguishable. Similarly, we say that* SFC *is* weakly spooky-free *if the simulator can be chosen after the adversary and the distinguisher have been set.*

Discussion. Note that our definition allows for additional "auxiliary information" z which can be communicated between the prover and distinguisher. This is intuitively understandable since we expect an SFC adversary to attempt to forge w.r.t some instance of the MIP language, but it would have been even better to prove a negative result relative to a definition where such auxiliary information is not allowed. However, we note that the type of auxiliary information used in our negative result (Theorem 5.1) is very mild. Specifically, z is generated independently of e (i.e. in a *non-adaptive* manner) and is just a random string (which can be made of length roughly $|e|$, but for convenience of presentation we make it slightly longer). That is, in our separation first a string z is chosen randomly, and the e' is calculated based on the values e, z. We further observe that, depending on the MIP in question, it may be possible to embed z within e' and remove it from the definition altogether. This could occur in MIPs that have some slackness in the verification procedure, e.g. if the MIP verifier ignores some of the bits of the answer a.

Finally, we stress that even given a spooky free compiler that allows z of arbitrary length and full dependence on e, we are not aware of a construction of a consequent adaptively secure delegation protocol. Furthermore, since we show that spooky free FHE, which is currently not known to imply adaptively secure delegation, implies SFC according to our definition.

On Black-Box Reductions of Spooky Free Compilers. Let us explicitly instantiate the definition of black box reductions (Definition 2.9 above) in the context of (weak) spooky free compilers. This is the definition that will be used to prove our main technical result in Theorem 5.1.

Consider a candidate spooky free compiler as in Definition 3.1 above. Then a pair of (not necessarily efficient) algorithms $(\mathcal{A}, \Psi)$ breaks weak spooky freeness if for any simulator $\mathcal{S}$ (possibly dependent on $\mathcal{A}, \Psi$) allowed to run in time $\text{poly}(\mathsf{time}(\mathcal{A}), \mathsf{time}(\Psi))$, it holds that Ψ can distinguish between the distributions $\mathbf{REAL}_{\mathcal{A}}$ and $\mathbf{SIM}_{\mathcal{S}}$ with non-negligible probability (we refer to this as "breaking spooky freeness").

A black-box reduction from a falsifiable assumption $(\mathcal{C}, \eta)$ to a weakly spooky free compiler is an oracle machine $\mathcal{R}$ that, given oracle access to a pair of machines $(\mathcal{A}, \Psi)$ that break weak spooky freeness as defined above, $\mathcal{R}^{(\mathcal{A},\Psi)}$ has non-negligible advantage against $(\mathcal{C}, \eta)$. We note that we will prove an even stronger result that places no computational restrictions at all on the running time of $\mathcal{S}$.

3.1 Spooky-Free FHE Implies Spooky Free Compiler

In the following lemma, we show that weakly and strongly spooky-free compiler are respectively weaker notions than a weakly and strongly spooky-free FHE. In the light of the lemma, spooky-free FHE can be seen as a "universal" SFC as it is not tied to a specific MIP.

Lemma 3.1. *There exists an efficient transformation that takes as input a weakly (respectively strongly) spooky-free FHE scheme* FHE*, and an MIP* Π*, and outputs a weakly (resp. strongly) spooky-free compiler for* Π*. The transformation and the proof make black-box use in the homomorphic encryption scheme.*

Proof. Let $\mathsf{FHE} = (\mathsf{FHE.Keygen}, \mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$ be a spooky-free FHE scheme, and let $\Pi = (\mathcal{G}, \vec{\mathcal{P}}, \mathcal{V})$ be a p-prover MIP. Consider the following spooky free compiler $\mathsf{SFC} = (\mathsf{SFC.Enc}, \mathsf{SFC.Dec}, \mathsf{SFC.Eval})$ for Π:

- $\mathsf{SFC.Enc}(1^\kappa, \vec{q})$: Generates p pairs of keys $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{FHE.Keygen}(1^\kappa)$ for FHE for all $i \in [p]$, and computes $c_i \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}_i, q_i)$. Finally, it sets $e = (\overrightarrow{\mathsf{pk}}, \vec{c})$, $\mathsf{dk} = \overrightarrow{\mathsf{sk}}$ and outputs (e, dk).
- $\mathsf{SFC.Eval}(e, x, w)$: Parses $e = (\{\mathsf{pk}_i\}_i, \{c_i\}_i)$. Evaluates $c'_i \leftarrow \mathsf{FHE.Eval}(\mathcal{P}_i(\cdot, x, w), c_i, \mathsf{pk}_i)$ for every $i \in [p]$, and outputs $e' = \vec{c}'$.
- $\mathsf{SFC.Dec}(e', \mathsf{dk})$: Parses $e' = \vec{c}'$, $\mathsf{dk} = \overrightarrow{\mathsf{sk}}$. Decrypts the evaluated FHE ciphertexts $a_i = \mathsf{FHE.Dec}(\mathsf{sk}_i, c'_i)$ for every $i \in [p]$, and outputs $\vec{a}$.

Completeness follows from the correctness and full homomorphism of FHE. To prove that SFC is spooky-free, we first consider a transformation that takes an adversary $(\mathcal{A}, \Psi)$ against the spooky-freeness of SFC (Definition 3.1) and produces an adversary $(\mathcal{D}', \mathcal{A}', \Psi')$ against the spooky-freeness of FHE (Definition 2.5).

- Set the sampler $\mathcal{D}'(1^\kappa)$ to be the sampler that first generates a sequence of MIP queries $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$, and outputs the $(p+1)$-tuple $(0, \vec{q})$, i.e. adds a leading zero. For notational convenience we denote the first coordinate by 0, so that q_i is the i-th entry in the new tuple as well.
- The algorithm $\mathcal{A}'$, given the ciphertexts $(c_0, \ldots, c_p)$ computes $\mathcal{A}(1^\kappa, (c_1, \ldots, c_p))$ to obtain $(z, (c'_1, \ldots, c'_p))$. Letting Const_z be the constant function z, i.e. $\mathsf{Const}_z(x) = z$ for all x. Let $c'_0 = \mathsf{FHE.Eval}(\mathsf{Const}_z, c_0)$. Output $\vec{c}' = (c'_0, \ldots, c'_p)$.
- The distinguisher Ψ' parses its input as $((0, \vec{q}), (z, \vec{a}))$ and outputs $\Psi(z, \vec{q}, \vec{a})$.

Since FHE is spooky-free, there exists a simulator $\mathcal{S}'$ against $(\mathcal{D}', \mathcal{A}', \Psi')$. Furthermore, if FHE is strongly spooky-free then $\mathcal{S}'$ does not depend on $\mathcal{A}', \Psi'$ and thus also not on $\mathcal{A}, \Psi$. We use $\mathcal{S}'$ to define a simulator $\mathcal{S}$ for SFC against $(\mathcal{A}, \Psi)$. The definition of $\mathcal{S}$ is almost verbatim identical to $\mathcal{S}'$, where we note that since we set the zeroth coordinate of $\mathcal{D}'$ to always be zero, then the syntax of $\mathcal{S}, \mathcal{S}'$ is identical up to rearrangement of the order of outputs. Namely, $\mathcal{S}'$ outputs $((0, \vec{q}), (z, \vec{a}))$, and $\mathcal{S}$ outputs $(z, \vec{q}, \vec{a})$. The indistinguishability of $\mathbf{REAL}_{\mathcal{A},\Psi}$ and $\mathbf{SIM}_{\mathcal{S}}$ follows directly from that of $\mathbf{REAL}_{\mathcal{D}',\mathcal{A}',\Psi'}$ and $\mathbf{SIM}_{\mathcal{S}'}$, since the distributions output in both cases are the same up to reordering. Which completes the proof. □

4 Non-Succinct Spooky Freeness is Trivial

In this section, we construct a non-succinct spooky free FHE, where the length of each ciphertext and the length of each public-key is exponential in the length of the messages. Specifically, we show how to convert any FHE scheme into a spooky-free FHE scheme such that the length of each ciphertext and each public key is $2^k \cdot \mathrm{poly}(\lambda)$, where k is the length of the messages. Naturally this is only applicable in settings where communication and computational complexity $2^k \cdot \mathrm{poly}(\lambda)$ is considered efficient.

We note that a spooky-free FHE implies a spooky-free compiler as shown in Lemma 3.1.

Theorem 4.1. *There exists an efficient generic transformation from any fully-homomorphic encryption scheme* $\mathrm{FHE} = (\mathsf{FHE.Keygen}, \mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$ *into a scheme* $\mathrm{FHE}' = (\mathsf{FHE.Keygen}', \mathsf{FHE.Enc}', \mathsf{FHE.Dec}', \mathsf{FHE.Eval}')$ *that is fully-homomorphic and* strongly spooky-free. *The length of each ciphertext generated by* $\mathsf{FHE.Enc}'$ *and the length of each public-key generated by* $\mathsf{FHE.Keygen}'$ *is* $2^k \cdot \mathrm{poly}(\lambda)$*, where* k *is the length of each message.*

Proof Overview. We transform the scheme to have equivocal properties. Specifically, the transformed scheme's ciphertexts can be replaced with ones that can be decrypted to any value using different pre-computed secret-keys. The joint distribution of each secret-key and the special ciphertext are indistinguishable from a properly generated secret-key and ciphertext. This allows us to define a simulator that precomputes those secret-keys and queries the adversary using an equivocable ciphertext. Then, it decrypts with the secret-key corresponding to the given message to extract the adversary's answers. By indistinguishability, this is the same answer that would be produced by querying the adversary, if it was queried with an encryption of that message.

We achieve this property by simply generating independently 2^k public-keys, whereas the secret-key corresponds only to one of the public-keys. Each ciphertext is 2^k encryptions, under each public-key. The equivocable ciphertext is produced by encrypting each of the 2^k possible messages under some public-key, in a randomly chosen order. Indistinguishability follows from the semantic-security of the original scheme.

Remark 4.1. *We assume, without the loss of generality, that the length of an encryption of* k *bits is bounded by* $k \cdot \mathrm{poly}(\lambda)$*, since we can always encrypt bit-by-bit while preserving security and homomorphism.*

Proof. Let $k = k(\lambda)$ be an upper-bound on the length of the messages $|m_i| \leq k$, where $(m_1, \ldots, m_n, \alpha) \leftarrow \mathcal{D}$. We define the scheme FHE' as follows:

- $\mathsf{FHE.Keygen}'(1^\lambda)$: Generate 2^k pairs of keys $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{FHE.Keygen}(1^\lambda)$. Then, choose uniformly at random an index $j \xleftarrow{\$} [2^k]$, and output $(\vec{\mathsf{pk}}, (\mathsf{sk}_j, j))$.
- $\mathsf{FHE.Enc}'(\vec{\mathsf{pk}}, \mu)$: Encrypt the message under each public-key $c_i \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}_i, \mu)$, then output $\vec{c}$.
- $\mathsf{FHE.Dec}'((\mathsf{sk}, j), \vec{\hat{c}})$: Decrypt according to the indexed secret-key and output $\mu' := \mathsf{FHE.Dec}(\mathsf{sk}, \hat{c}_j)$.
- $\mathsf{FHE.Eval}'(\vec{\mathsf{pk}}, \vec{c}, \mathcal{C})$: For every $i \in [2^k]$ compute $\hat{c}_i \leftarrow \mathsf{FHE.Eval}(\mathsf{pk}_i, c_i, \mathcal{C})$ and output $\vec{\hat{c}}$.

Clearly FHE' is a fully-homomorphic encryption scheme. It is thus left to prove that it is strongly spooky-free.

The simulator $\mathcal{S}^{\mathcal{A}}(1^\lambda, 1^n, i, m_i; r)$ First, the simulator uses its randomness r to sample $n \cdot 2^k$ pairs of keys $(\mathsf{pk}_{\ell,j}, \mathsf{sk}_{\ell,j}) \leftarrow \mathsf{FHE.Keygen}(1^\lambda), \ell \in [n], j \in [2^k]$. Then, for every 2^k-tuple $\vec{\mathsf{pk}}_\ell = (\mathsf{pk}_{\ell,1}, \ldots, \mathsf{pk}_{\ell,2^k})$, it chooses a permutation $\pi_\ell : [2^k] \to [2^k]$, encrypts the message $\pi_\ell^{-1}(j)$ under the public-key $\mathsf{pk}_{\ell,j}$, for every $j \in \{0,1\}^k$

$$c_{\ell,j} = \mathsf{FHE.Enc}(\mathsf{pk}_{\ell,j}, \pi_\ell^{-1}(j)) \ .$$

Next, it sets $\vec{c}_\ell = (c_{\ell,1}, \ldots, c_{\ell,2^k})$ and queries the adversary to get

$$(\vec{c'}_1, \ldots, \vec{c'}_n) \leftarrow \mathcal{A}((\vec{\mathsf{pk}}_1, \ldots, \vec{\mathsf{pk}}_n), (\vec{c}_1, \ldots, \vec{c}_n)) \ .$$

Finally it outputs m'_i:=$\mathsf{FHE.Dec}(\mathsf{sk}_{i,\pi_\ell(m_i)}, c'_{i,\pi_\ell(m_i)})$.

Claim 4.1.1. *For every* PPT *adversary $\mathcal{A}$ and distribution $\mathcal{D}$, the experiments $\boldsymbol{REAL}_{\mathcal{D},\mathcal{A}}$ and $\boldsymbol{SIM}_{\mathcal{D},\mathcal{S}^{\mathcal{A}}}$ are computationally indistinguishable.*

Proof. We prove using a sequence of hybrids.

- $\mathcal{H}_0$: This is simply the distribution $\mathbf{REAL}_{\mathcal{D},\mathcal{A}}$.
- $\mathcal{H}_{1,i}$ ($i \in [n]$): In these hybrids we modify the key generation step in $\mathbf{REAL}_{\mathcal{D},\mathcal{A}}$: Instead of choosing $j_i \xleftarrow{\$} [2^k]$ uniformly at random, we choose uniformly at random a permutation $\pi_i : [2^k] \to [2^k]$ and set $j_i = \pi_i(m_i)$. These hybrids are identically distributed, since the π_i's are random permutations, so each j_i is distributed uniformly over $[2^k]$.
- $\mathcal{H}_{2,i,j}$ ($i \in [n], j \in [2^k]$): In these hybrids we modify the encryption step in $\mathbf{REAL}_{\mathcal{D},\mathcal{A}}$: Instead of letting $c_{i,j} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}_{i,j}, m_i)$, set $c_{i,\pi_i(j)} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}_{i,\pi_i(j)}, j)$, where π_i is the permutation from the previous hybrids. These hybrids are computationally indistinguishable by the semantic security of FHE.

Finally, note that $\mathcal{H}_{2,n,2^k}$ is actually $\mathbf{SIM}_{\mathcal{D},\mathcal{S}^{\mathcal{A}}}$. This is since for every $i \in [n]$, the simulator queries the adversary the same query every time, and that query is distributed as the one in $\mathcal{H}_{2,n,2^k}$. Moreover, the adversary's answer is decrypted in the same manner both in $\mathbf{SIM}_{\mathcal{D},\mathcal{S}^{\mathcal{A}}}$ and $\mathcal{H}_{2,n,2^k}$. Thus $\mathbf{REAL}_{\mathcal{D},\mathcal{A}} \overset{c}{\approx} \mathbf{SIM}_{\mathcal{D},\mathcal{S}^{\mathcal{A}}}$, as desired. ■

Which completes the proof. □

5 Succinct Spooky Freeness Cannot be Proven using a Black-Box Reduction

We state and prove our main theorem.

Theorem 5.1. *Let L be a language with a sub-exponentially hard subset-membership problem $(\mathcal{L}, \overline{\mathcal{L}}, \mathsf{Sam})$ with hardness parameter α. Let $\Pi = (\mathcal{G}, \vec{\mathcal{P}}, \mathcal{V})$ be a succinct one-round MIP for L with a spooky free compiler SFC, with $|e'| = \mathrm{poly}(\kappa) \cdot |x|^{\alpha'}$ for some $\alpha' < \alpha$. Then there is no black-box reduction showing the weakly spooky-freeness of SFC based on a falsifiable assumption $(\mathcal{C}, \eta)$, unless $(\mathcal{C}, \eta)$ is polynomially solvable.*

Proof Overview. We start by defining an inefficient adversary $(\overline{\mathcal{A}}, \overline{\Psi})$ against SFC, or more precisely a distribution over adversaries specified by a family of sets $\overline{X}$. These sets contain, for each value of the security parameter 1^κ, a large number of inputs from $\overline{\mathcal{L}}$ of length $n = \mathrm{poly}(\kappa)$ for a sufficiently large polynomial to make $|e'|$ bounded by n^α. The adversary $\overline{\mathcal{A}}$ picks a random x from the respective set and generates a response e' as follows. As a thought experiment, if it was the case that $x \in \mathcal{L}$, then SFC allows us to generate e' that will be accepted by the MIP verifier, by applying $\mathsf{SFC.Eval}$ using a witness w for x. Therefore, the Dense Model Theorem states that

it is possible to generate a computationally indistinguishable e' also for $x \in \overline{\mathcal{L}}$. The distinguisher $\overline{\Psi}$ takes $(z, \vec{q}, \vec{a})$ sets x to be the zth element in $\overline{X}$ and applies the MIP verifier. The soundness of MIP guarantees that this distribution cannot be simulated by independent provers. Note that z is essential in order for $\overline{\mathcal{A}}$ and $\overline{\Psi}$ to agree on an input $x \in \overline{\mathcal{L}}$. The use of a common $\overline{X}$ allows $\overline{\mathcal{A}}$ and $\overline{\Psi}$ to share a set of inputs for which they know the simulator cannot work.

Since $(\overline{\mathcal{A}}, \overline{\Psi})$ is successful against SFC, it means that the reduction breaks the assumption given oracle access to $(\overline{\mathcal{A}}, \overline{\Psi})$.[8] Our goal now is to show an efficient procedure $(\mathcal{A}, \Psi)$ which is indistinguishable from $(\overline{\mathcal{A}}, \overline{\Psi})$ in the eyes of the reduction. More accurately, $(\mathcal{A}, \Psi)$ can be a stateful machine and not necessarily an oracle, what we need is that it creates an interface with the reduction that is indistinguishable from the one it gets from interacting with $(\overline{\mathcal{A}}, \overline{\Psi})$. Once we are able to do that, it will show that the underlying assumption is in fact polynomially solvable.

To do this, we notice that the reduction can only ever see polynomially many x's, so there is no need to sample a huge set $\overline{X}$, and an appropriately defined polynomial subset would be sufficient. Furthermore, instead of sampling from $\overline{\mathcal{L}}$, we can sample from $\mathcal{L}$ together with a witness, and compute e' as a legitimate $\mathsf{SFC.Eval}$ response. The Dense Model Theorem ensures that this strategy will be indistinguishable to the reduction, and therefore it should still be successful in breaking the assumption. We have to be careful since the reduction might query its oracle on tiny security parameter values for which n is not large enough to apply the Dense Model Theorem. For those small values we create a hard-coded table of adversary responses (since these are tiny values, the table is still not too large).

Finally, we see that our simulated adversary runs in polynomial time since it only needs to sample from $\mathcal{L}$, which is efficient using Sam, and use the witness to compute e' via $\mathsf{SFC.Eval}$. We conclude that we have a polynomial time algorithm that succeeds in breaking the assumption, as required in the theorem.

Proof. We proceed as in the sketch above. By the properties of SFC as stated in the theorem, there exist constants $\beta_1, \beta_2, \beta_3 > 0$ such that $\beta_1 = \alpha - \alpha'$, $|e'| \leq O(\kappa^{\beta_2} \cdot |x|^{\alpha - \beta_1})$, $|e| \leq \kappa^{\beta_3}$. We define

$$n(\kappa) \triangleq \kappa^{\max\{2\beta_2/\beta_1, \beta_3/\alpha'\}} ,$$

and note that $|e|, |e'| = o(n^\alpha)$, when $|x| = n(\kappa)$. .

Proofs Can Be Spoofed. We start by showing how to inefficiently spoof SFC answers for non-accepting inputs. Consider an encoded query e for SFC w.r.t. security parameter 1^κ, and define the distribution $(\mathcal{L}, \mathcal{E})$ as follows:

1. Sample $(x, w) \leftarrow \mathsf{Sam}(1^{n(\kappa)})$.
2. Evaluate $e' \leftarrow \mathsf{SFC.Eval}(e, x, w)$.
3. Output (x, e').

The following claim shows that it is possible to sample from a distribution that is computationally indistinguishable from $(\mathcal{L}, \mathcal{E})$, but where the first component comes from $\overline{\mathcal{L}}$.

[8] In fact, the situation is more delicate since as explained above $(\overline{\mathcal{A}}, \overline{\Psi})$ is a *distribution over adversaries*, and while almost all adversaries in the support succeed against SFC, it still requires quite a bit of work to prove that the average advantage is also non-negligible (see Lemma 2.6 in Section 2).

Claim 5.1.1. *For every e, there exists a randomized function $h = h_e$ such that the distributions $(\mathcal{L}, \mathcal{E})$ and $(\overline{\mathcal{L}}, h(\overline{\mathcal{L}}))$ are $(2 \cdot 2^{-n^\alpha}, 2^{n^\alpha})$ indistinguishable.*

Proof. Follows from Corollary 2.3 since $\mathcal{L}_n$ and $\overline{\mathcal{L}}_n$ are α-sub-exponentially indistinguishable. ■

Constructing a Spooky Adversary. We define an adversary $\overline{\mathcal{A}}$, along with a distinguisher $\overline{\Psi}$ for the spooky-free experiment in SFC. We note that both $\overline{\mathcal{A}}$ and $\overline{\Psi}$ are *inefficient* algorithms, and more precisely, they are *distributions* over algorithms.

For every value of κ, define $\nu(\kappa) = 2^{0.1 \cdot n(\kappa)^\alpha}$. Define a vector $\overline{X}_{n(\kappa)} \xleftarrow{\$} \overline{\mathcal{L}}_{n(\kappa)}^{\otimes \nu(\kappa)}$, i.e. a sequence of independent samples from $\overline{\mathcal{L}}_n$.

The functionality of $\overline{\mathcal{A}}$ and $\overline{\Psi}$ is as follows:

- $\overline{\mathcal{A}}(1^\kappa, e)$: Samples $z \xleftarrow{\$} [\nu(\kappa)]$, sets $\overline{x} = \overline{X}_{n(\kappa)}[z]$ (i.e. the z'th element in the vector), and outputs $(h_e(\overline{x}), z)$.
- $\overline{\Psi}(1^\kappa, z, \vec{q}, \vec{a})$: Outputs 1 if and only if $\mathcal{V}(\vec{q}, \vec{a}, \overline{X}_{n(\kappa)}[z]) = 1$.

The following claim asserts that the adversary $\overline{\mathcal{A}}$ and the distinguisher $\overline{\Psi}$ win the spooky-freeness game for the compiler SFC with probability 1 over the choice of the respective $\overline{X} = \{\overline{X}_{n(\kappa)}\}_\kappa$.

Claim 5.1.2. *With probability 1 over the choice of $\overline{X} = \{\overline{X}_{n(\kappa)}\}_\kappa$ it holds that $(\overline{\mathcal{A}}, \overline{\Psi})$ has non-negligible advantage in the spooky freeness game against SFC with any (possibly computationally unbounded) simulator.*

Proof. Let σ denote the soundness of the underlying MIP system. According to the definition of an MIP (see Definition 2.2), the soundness gap, $\sigma_{\text{gap}} \triangleq 1 - \sigma$, is non-negligible.

We start by showing that for all $\overline{X}$, any value of κ, and any (possibly unbounded) spooky-free simulator $\mathcal{S}$ for the compiler SFC, it holds that

$$\Pr[\overline{\Psi}(\mathbf{SIM}_{\mathcal{S}}(1^\kappa)) = 1 \,|\, \overline{X}] \leq \sigma(\kappa).$$

This follows since by the definition of the simulator, each value of its random string r defines an auxiliary z and induces a sequence of algorithms $\vec{\mathcal{S}}$ where

$$(\mathcal{S}_1(q_1), \ldots, \mathcal{S}_p(q_p)) = \vec{\mathcal{S}}(\vec{q}) \ .$$

Since $\overline{X}_{n(\kappa)}[z] \notin \mathcal{L}$, then by the soundness of Π, the probability that the verifier $\mathcal{V}$ accepts answers generated by $\vec{\mathcal{S}}$ is at most $\sigma(\kappa)$, and thus $\overline{\Psi}$ outputs 1 with probability at most $\sigma(\kappa)$.

Next, we turn to show that $\Pr[\overline{\Psi}(\mathbf{REAL}_{\overline{\mathcal{A}}}(1^\kappa)) = 1 \,|\, \overline{X}]$ is bounded away from $\sigma(\kappa)$ with probability 1 on $\overline{X}$. To this end, we define a sequence of events $\{E_\kappa\}_{\kappa \in \mathbb{N}}$, where E_κ is the event that

$$\Pr[\overline{\Psi}(\mathbf{REAL}_{\overline{\mathcal{A}}}(1^\kappa)) = 1 \,|\, \overline{X}] \leq 1 - \sigma_{\text{gap}}(\kappa)/2 \ ,$$

where the probability is over everything except the choice of $\overline{X}$. We show that with probability 1 over the choice of $\overline{X}$, only finitely many of the events E_κ occur.

To see this, fix queries $\vec{q} \leftarrow \mathcal{G}(1^\kappa)$ and encoding $(e, \mathsf{dk}) \leftarrow \mathsf{SFC.Enc}(\vec{q})$ for the experiment $\mathbf{REAL}_{\overline{\mathcal{A}}}$. Note that since the compiler's decoding algorithm $\mathsf{SFC.Dec}$ can be described by a $\text{poly}(\kappa)$

sized circuit, then we can describe the MIP's verifier $\mathcal{V}$ as a $\text{poly}(\kappa)$ sized circuit that takes inputs from $(\overline{\mathcal{L}}, h(\overline{\mathcal{L}}))$.

Recall that by Claim 5.1.1, the distributions $(\mathcal{L}, \mathcal{E})$ and $(\overline{\mathcal{L}}, h(\overline{\mathcal{L}}))$ are $(2 \cdot 2^{-n^\alpha}, 2^{n^\alpha})$-indistinguishable. Moreover, by the completeness of the MIP, $\mathcal{V}$ outputs 1 with probability 1 on inputs from $(\mathcal{L}, \mathcal{E})$. We conclude that $\mathcal{V}$ accepts inputs from $(\overline{\mathcal{L}}, h(\overline{\mathcal{L}}))$ with overwhelming probability, and therefore $\overline{\Psi}$ also accepts with overwhelming probability inputs from $\mathbf{REAL}_{\overline{\mathcal{A}}}$. In other words, we have that

$$\mathop{\mathbb{E}}_{\overline{X}} \left[\Pr[\overline{\Psi}(\mathbf{REAL}_{\overline{\mathcal{A}}}(1^\kappa)) = 1 \,|\, \overline{X}] \right] \geq 1 - \text{negl}(\kappa) \ .$$

By a Markov argument this implies that

$$\Pr_{\overline{X}} \left[\Pr[\overline{\Psi}(\mathbf{REAL}_{\overline{\mathcal{A}}}(1^\kappa)) = 1 \,|\, \overline{X}] \leq 1 - \sigma_{\text{gap}}(\kappa)/2 \right] \leq \text{negl}(\kappa) \ .$$

Finally, we apply the Borel-Cantelli Lemma to conclude that with probability 1 over the choice of $\overline{X}$, only finitely many of the events E_κ occur, as desired.

Thus, with probability 1 (over the choice of $\overline{X}$), it holds that $(\overline{\mathcal{A}}, \overline{\Psi})$ has advantage at least $\sigma_{\text{gap}}/2$ in the spooky free game. This completes the proof of the claim. ■

Fooling the Reduction. We now notice that by Corollary 2.5, it is sufficient to prove the theorem for $\eta = 0$. Assume that there exists a black-box reduction $\mathcal{R}$ as in the theorem statement, and we will prove that $(\mathcal{C}, 0)$ is solvable in polynomial time. We notice that since $(\overline{\mathcal{A}}, \overline{\Psi})$ break spooky freeness with probability 1, it follows from Lemma 2.6 that

$$\delta(\lambda) = \Pr[\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]$$

is noticeable, where the probability is taken over the randomness of sampling $(\overline{\mathcal{A}}, \overline{\Psi})$, the randomness of the reduction and the randomness of $\mathcal{C}$.

We turn to define another adversary $\mathcal{A}$ and distinguisher Ψ, by modifying $\overline{\mathcal{A}}$ and $\overline{\Psi}$ in a sequence of changes. Our goal is to finally design $(\mathcal{A}, \Psi)(1^\lambda)$ (possibly stateful) computable in $\text{poly}(\lambda)$ time, while ensuring that $\mathcal{R}^{(\mathcal{A}, \Psi)}$ still has advantage $\Omega(\delta)$.

Throughout the proof, we let $t = t(\lambda)$ be a fixed polynomial upper bound on the run time of $\mathcal{R}$. In particular this implies that $\mathcal{R}$ makes at most t queries to its oracle. We define two values $\kappa_{\min}, \kappa_{\max}$ that will be instrumental for our proof. Intuitively, $\kappa_{\max}$ is the maximal value of κ that can be generated by $\mathcal{R}(1^\lambda)$, and $\kappa_{\min}$ is a value for which $2^{n(\kappa)^\alpha}$ becomes polynomial in λ. Formally we define as follows.

- We define $\kappa_{\max} = \kappa_{\max}(\lambda) = \text{poly}(\lambda)$ to be a bound on the size of the security parameters that the reduction $\mathcal{R}$ uses when interacting with its oracle. Note that $\kappa_{\max}$ is bounded by the runtime of the reduction, which in turn is bounded by some fixed polynomial (in λ).
- We define $\kappa_{\min} = \kappa(\lambda)$ to be the maximal κ such that $2^{n(\kappa)^\alpha} \leq \lambda^c$ for a constant c for which $\lambda^{0.1c} \geq 20t^2(\lambda)/\delta(\lambda)$ (note that since t is polynomial and δ is noticeable, then such constant indeed exists). This choice is in order to satisfy constraints that will be explained below. Note that for all $\kappa \leq \kappa_{\min}$ it holds that $\nu(\kappa) = |\overline{X}_{n(\kappa)}| = 2^{O(n^\alpha)} = \text{poly}(\lambda)$ and for all $\kappa > \kappa_{\min}$ it holds that $\nu(\kappa) = 2^{0.1 \cdot n^\alpha} \geq \lambda^{0.1c}$.

The following claim asserts that the behavior of $(\overline{\mathcal{A}}, \overline{\Psi})$ for all $\kappa \leq \kappa_{\min}$ can be computed in (nonuniform) $\text{poly}(\lambda)$ time.

Claim 5.1.3. *Recall that $(\overline{\mathcal{A}}, \overline{\Psi})$ is a distribution over oracles. For every oracle in the support of this distribution there exists a* $\mathrm{poly}(\lambda) = 2^{O(n(\kappa_{\min})^{\alpha})}$ *size circuit* Min *that implements the functionality of the respective $(\overline{\mathcal{A}}, \overline{\Psi})$ for all $\kappa \leq \kappa_{\min}$.*

Proof. The collection of sets $\{\overline{X}_{n(\kappa)}\}_{\kappa \leq \kappa_{\min}}$ only contains $\sum_{\kappa \leq \kappa_{\min}} 2^{O(n(\kappa)^{\alpha})} \leq \mathrm{poly}(\lambda)$ many instances, each of at most $\mathrm{poly}(\lambda)$ size. The circuit Min will contain a table of all of these instances. This will already allow to implement $\overline{\Psi}$ which is polynomial time computable given the mapping between z and x. As for $\overline{\mathcal{A}}$, we note that its input is of length at most $|e| = o(n^{\alpha})$ so one can write the entire truth table using a $\mathrm{poly}(\lambda)$ size circuit. ■

The rest of the proof follows by a sequence of hybrids.

Hybrid $\mathcal{H}_0$. In this hybrid we execute $\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}$ as defined.

$$\delta(\lambda) = \Pr_{\mathcal{H}_0}[\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}(1^{\lambda}) \text{ wins } \mathcal{C}(1^{\lambda})] \ .$$

Hybrid $\mathcal{H}_1$. We remove all the sets relative to $\kappa > \kappa_{\max}$ from the ensemble $\{\overline{X}_{n(\kappa)}\}$. That is, now $\{\overline{X}_{n(\kappa)}\}$ only contains finite (specifically $\mathrm{poly}(\lambda)$) many sets. Since by definition $\mathcal{R}$ cannot query on such large values of κ this step does not affect the advantage of $\mathcal{R}$. We note that $\kappa_{\max}(\lambda) \leq t(\lambda)$ (where t is the running time of $\mathcal{R}$ as described above).

$$\left| \Pr_{\mathcal{H}_1}[\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}(1^{\lambda}) \text{ wins } \mathcal{C}(1^{\lambda})] - \Pr_{\mathcal{H}_0}[\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}(1^{\lambda}) \text{ wins } \mathcal{C}(1^{\lambda})] \right| = 0 \ .$$

From here on, we will focus on $\kappa \in (\kappa_{\min}, \kappa_{\max})$, since in the other regimes we can indeed execute $\overline{\mathcal{A}}, \overline{\Psi}$ efficiently. We call this *the relevant domain.*

Hybrid $\mathcal{H}_2$. We now change $(\overline{\mathcal{A}}, \overline{\Psi})$ and make them *stateful.* Specifically, instead of sampling $\overline{X}$ ahead of time for κ in the relevant domain, the values are sampled on the fly. The new $(\overline{\mathcal{A}}, \overline{\Psi})$ maintains a list of tuples (z, x) as follows.

1. The list $\overline{X}_{n(\kappa)}$ is initialized empty for all κ in the relevant domain.
2. For any $\overline{\mathcal{A}}(1^{\kappa}, e)$ query, sample a random $z \leftarrow [\nu(\kappa)]$.
   (a) If there exists an entry (z, x) in $\overline{X}_{n(\kappa)}$ then computes $e' = h_e(x)$ and returns (e', z).
   (b) Otherwise, generate (x, e') from the distribution $(\overline{\mathcal{L}}, h_e(\overline{\mathcal{L}}))$, add (z, x) to $\overline{X}_{n(\kappa)}$ and return (e', z).
3. For any $\overline{\Psi}(z, \vec{q}, \vec{a})$ query (corresponding to κ that is implicit in the length of z):
   (a) If there exists an entry (z, x) in $\overline{X}_{n(\kappa)}$ then run the MIP verifier on x.
   (b) Otherwise, sample a new $x \leftarrow \overline{\mathcal{L}}$, add (z, x) to $\overline{X}_{n(\kappa)}$ and run the MIP verifier on x.

This modification does not change the functionality of $(\overline{\mathcal{A}}, \overline{\Psi})$ and in particular

$$\left| \Pr_{\mathcal{H}_2}[\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}(1^{\lambda}) \text{ wins } \mathcal{C}(1^{\lambda})] - \Pr_{\mathcal{H}_1}[\mathcal{R}^{(\overline{\mathcal{A}}, \overline{\Psi})}(1^{\lambda}) \text{ wins } \mathcal{C}(1^{\lambda})] \right| = 0 \ .$$

Hybrid $\mathcal{H}_3$. This hybrid formalizes the claim that $\overline{\mathcal{A}}$ is unlikely to sample values of z that are already in the list. We change the behavior of the stateful $(\overline{\mathcal{A}}, \overline{\Psi})$ from the previous hybrid, so that if step 2a is reached, the adversary responds with $\perp$.

Claim 5.1.4. *Recall that c is chosen so that in particular $t(\lambda)^2/\lambda^{0.1c} \leq \delta/10$. Then it holds that*

$$\left|\Pr_{\mathcal{H}_3}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] - \Pr_{\mathcal{H}_2}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]\right| \leq \delta/10 \ .$$

Proof. We bound the probability that step 2a is reached in the execution of $\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda)$. At any point in time, each $\overline{X}_{n(\kappa)}$ list contains at most t entries (since each query of $\mathcal{R}$ introduces at most one new element), so the probability of the new z hitting this subset is at most $t/\nu(\kappa) \leq t/\lambda^{0.1c}$ (recall that $\nu(\kappa) = 2^{0.1 \cdot n^\alpha} \geq \lambda^{0.1c}$). Applying the union bound over at most t queries, the claim follows. ■

Hybrid $\mathcal{H}_4$. We note that in the previous hybrid $\mathcal{H}_3$, the only inefficient component of $(\overline{\mathcal{A}}, \overline{\Psi})$ are steps 2b, 3b. We notice that in fact step 3b is just a simpler version of 2b, where e is not specified (e.g. it can be the empty string) and the resulting e' is not output. We can therefore treat both of these in the same way and from this point and on we assume that sampling is always with respect to some e value. We collectively refer to these steps as *instance sampling steps*. Our goal is to replace this inefficient step with an efficient one using the Dense Model Theorem. We will do this by a sequence of hybrids, but we need to make sure that when applying the theorem to a certain value of n, the simulation of the hybrid can be done in complexity $2^{O(n^\alpha)}$ so that the conditions of the theorem are met.

We consider the following (randomized) functions $\Gamma, \overline{\Gamma}$, each takes as input $1^\kappa, e$ and output x, e' as follows. The function $\Gamma(1^\kappa, e; \rho)$ (where ρ indicates the random tape) samples $(x, w) \leftarrow \mathsf{Sam}(1^{n(\kappa)})$, computes $e' = \mathsf{SFC.Eval}(e, x, w)$ and outputs (x, e'). The function $\overline{\Gamma}(1^\kappa, e; \rho)$ samples $x \leftarrow \overline{\mathcal{L}}$ and computes $e' = h_e(x)$. We note that Γ is efficiently computable, whereas for any fixed values of κ, ρ it is possible to non-uniformly compute $f(e) = \overline{\Gamma}(1^\kappa, e; \rho)$ using a $2^{|e|} = 2^{o(n^\alpha)}$ size circuit, by writing down the truth table. We denote this circuit by $\overline{\Gamma}_{\kappa,\rho}$.

We first consider a semantic change in the instance sampling steps. For every value of κ in the relevant domain, $(\overline{\mathcal{A}}, \overline{\Psi})$ will sample t random tapes $\rho_{\kappa,i}$ for $i \in [t]$ and in the ith execution of an instance sampling step on value κ, it will execute $\overline{\Gamma}(1^\kappa, e; \rho_{\kappa,i})$. So far the behavior of $(\overline{\mathcal{A}}, \overline{\Psi})$ did not change at all.

In this hybrid, we replace the calls to $\overline{\Gamma}$ with calls to Γ, i.e. in our new hybrid, the ith execution of an instance sampling steps on value κ, will execute $\Gamma(1^\kappa, e; \rho_{\kappa,i})$.

Claim 5.1.5. *Recall that c is chosen so that in particular $2t(\lambda)^2/\lambda^c \leq \delta/10$. Then*

$$\left|\Pr_{\mathcal{H}_4}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] - \Pr_{\mathcal{H}_3}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]\right| \leq \delta/10 \ .$$

Proof. Consider an intermediate hybrid defined by a threshold security parameter $\hat{\kappa} \in (\kappa_{\min}, \kappa_{\max})$ and a value $j \in \{0, \ldots, t\}$. The intermediate hybrid $\widehat{\mathcal{H}}_{\hat{\kappa},j}$ is defined as follows. For all $\kappa < \hat{\kappa}$, the instance sampling steps are performed using $\overline{\Gamma}$. For all $\kappa > \hat{\kappa}$, the instance sampling steps are performed using Γ. For $\kappa = \hat{\kappa}$, for the first j calls, $\overline{\Gamma}$ is used and beyond that Γ is used. Note that

hybrid $(\hat{\kappa}+1, 0)$ is identical to $(\hat{\kappa}, t)$, so it is sufficient to show indistinguishability between hybrids with the same value of $\hat{\kappa}$.

Consider adjacent hybrids $(\hat{\kappa}, j-1)$, $(\hat{\kappa}, j)$. The difference between them is only in the jth execution of an instance sampling step with $\kappa = \hat{\kappa}$. We denote $\hat{n} = n(\hat{\kappa})$ and show that

$$\left| \Pr_{\widehat{\mathcal{H}}_{\hat{\kappa},j}} [\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] - \Pr_{\widehat{\mathcal{H}}_{\hat{\kappa},j-1}} [\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \right| \leq 2 \cdot 2^{-\hat{n}^\alpha} \leq 2/\lambda^c \ . \tag{1}$$

Since the total number of hybrids is at most $\kappa_{\max}(\lambda) \cdot t(\lambda)$, and recalling that $\kappa_{\max}(\lambda) \leq t(\lambda)$, the claim will follow.

To show that Eq. (1) holds, it is sufficient to show that it holds even when fixing all of the randomness of $\mathcal{R}, \mathcal{C}, \overline{\mathcal{A}}, \overline{\Psi}$, except for $\rho_{\hat{\kappa},j}$. Consider the following algorithm that simulates the interaction of $\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}$ with $\mathcal{C}(1^\lambda)$ as follows.

1. For $\kappa \leq \kappa_{\min}$, the simulation uses the $\text{poly}(\lambda)$ size circuit guaranteed by Claim 5.1.3 (note that since we fixed all randomness except $\rho_{\hat{\kappa},j}$, the functionality for such values of κ is fixed).
2. For all queries except the jth query of $\kappa = \hat{\kappa}$ do the following.
   (a) If Γ is to be executed, then execute it as prescribed (in $\text{poly}(\kappa) \leq \text{poly}(\lambda) = 2^{O(\hat{n}^\alpha)}$ size).
   (b) If $\overline{\Gamma}$ is to be executed, use the $2^{O(n^\alpha)}$ size circuit $\overline{\Gamma}_{\kappa,\rho_\kappa,i}$. Note that this is only required for $\kappa \leq \hat{\kappa}$, and thus each of the circuits $\overline{\Gamma}_{\kappa,\rho_{\kappa,i}}$ is of size at most $2^{O(\hat{n}^\alpha)}$. There are at most $t^2 = \text{poly}(\lambda) = 2^{O(\hat{n}^\alpha)}$ such circuits, and thus their total size is still bounded by $2^{O(\hat{n}^\alpha)}$.
3. For the jth query of $\kappa = \hat{\kappa}$, generate a sample either from $\Gamma(1^{\hat{\kappa}}, e; \rho_{\hat{\kappa},j})$ or from $\overline{\Gamma}(1^{\hat{\kappa}}, e; \rho_{\hat{\kappa},j})$. Note that since all other randomness is fixed, the value of the respective e is fixed as well.
4. The output of the simulation is whether $\mathcal{R}$ won the game $\mathcal{C}$.

The probability that the above algorithm outputs 1 is equal to $\Pr_{\widehat{\mathcal{H}}_{\hat{\kappa},j}}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]$ (conditioned on all the randomness values that we fixed) if we use $\overline{\Gamma}$ in step 3, and is equal to $\Pr_{\widehat{\mathcal{H}}_{\hat{\kappa},j-1}}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]$ (conditioned on all the randomness values that we fixed) if we use Γ in step 3. We thus get an algorithm computable by a circuit of size $2^{O(\hat{n}^\alpha)}$ that can distinguish $\Gamma(1^{\hat{\kappa}}, e; \rho_{\hat{\kappa},j})$ from $\overline{\Gamma}(1^{\hat{\kappa}}, e; \rho_{\hat{\kappa},j})$ with advantage

$$\left| \Pr_{\widehat{\mathcal{H}}_{\hat{\kappa},j}} [\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] - \Pr_{\widehat{\mathcal{H}}_{\hat{\kappa},j-1}} [\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \right| \ ,$$

where the probabilities are conditioned on all of the randomness that we fixed. We can now apply Corollary 2.3 to conclude that this advantage is at most $2 \cdot 2^{-\hat{n}^\alpha}$, and Eq. (1) follows. ■

Finally, let us consider the adversary that we get in the final hybrid $(\overline{\mathcal{A}}, \overline{\Psi})_{\mathcal{H}_4}$. All operations for $\kappa \geq \kappa_{\min}$ are performed in $\text{poly}(\lambda)$ time. However, for values $\kappa < \kappa_{\min}$ the running time might still be super-polynomial. We will resolve this using Claim 5.1.3 again. Formally, we will show the existence of $(\mathcal{A}, \Psi)$ with advantage at least as high as $(\overline{\mathcal{A}}, \overline{\Psi})_{\mathcal{H}_4}$ as stated in the following claim.

Claim 5.1.6. *There exist* $(\mathcal{A}, \Psi)$ *computable by* $\mathrm{poly}(\lambda)$ *size circuit for which*

$$\Pr[\mathcal{R}^{(\mathcal{A},\Psi)}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \geq \Pr_{\mathcal{H}_4}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \ .$$

Proof. We start by noticing that there exists a fixing of the randomness of $(\overline{\mathcal{A}}, \overline{\Psi})_{\mathcal{H}_4}$ for which the advantage is at least as high as the expected value $\Pr_{\mathcal{H}_4}[\mathcal{R}^{(\overline{\mathcal{A}},\overline{\Psi})}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)]$.

For this fixed adversary, we apply Claim 5.1.3 to conclude that its behavior on $\kappa < \kappa_{\min}$ can be implemented by a $\mathrm{poly}(\lambda)$ size circuit. Efficiency for $\kappa \geq \kappa_{\min}$ is preserved since the behavior of $(\overline{\mathcal{A}}, \overline{\Psi})_{\mathcal{H}_4}$ on each value of κ is independent of its behavior on the other values of κ. We define $(\mathcal{A}, \Psi)$ to be the resulting algorithm. ■

Conclusion. Combining the hybrids above, we get that $\mathcal{R}^{(\mathcal{A},\Psi)}$ is a $\mathrm{poly}(\lambda)$-time algorithm with advantage

$$\mathsf{Adv}^{(\mathcal{C},0)}_{\mathcal{R}^{(\mathcal{A},\Psi)}}(1^\lambda) \geq \delta - 2 \cdot \delta/10 = \Omega(\delta) \ .$$

That is, $\mathcal{R}^{(\mathcal{A},\Psi)}$ is a polynomial time algorithm that breaks the assumption $(\mathcal{C}, 0)$ as required. □

Acknowledgments. We wish to thank the Asiacrypt reviewers for the extremely thorough review process, and for their useful and enlightening comments that helped improve this manuscript significantly.

References

[ABOR00] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In *ICALP*, volume 1853 of *Lecture Notes in Computer Science*, pages 463–474. Springer, 2000.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[BCC+14] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *IACR Cryptology ePrint Archive*, 2014:580, 2014.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 326–349, 2012.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In *STOC*, pages 111–120. ACM, 2013.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational complexity*, 1(1):3–40, 1991.

[BMW98] Ingrid Biehl, Bernd Meyer, and Susanne Wetzel. Ensuring the integrity of agent-based computations by short proofs. In Kurt Rothermel and Fritz Hohl, editors, *Mobile Agents, Second International Workshop, MA'98, Stuttgart, Germany, September 1998, Proceedings*, volume 1477 of *Lecture Notes in Computer Science*, pages 183–194. Springer, 1998.

[DBL08] *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 2008.

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 54–74, 2012.

[DHRW16] Yevgeniy Dodis, Shai Halevi, Ron D Rothblum, and Daniel Wichs. Spooky encryption and its applications. In *Annual Cryptology Conference*, pages 93–122. Springer, 2016.

[DLN+01] Cynthia Dwork, Michael Langberg, Moni Naor, Kobbi Nissim, and Omer Reingold. Succinct proofs for np and spooky interactions. Unpublished manuscript, 2001.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA* [DBL08], pages 293–302.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 99–108. ACM, 2011.

[KRR13] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for bounded space. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 565–574. ACM, 2013.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *STOC*, pages 485–494. ACM, 2014.

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 436–453. IEEE Computer Society, 1994.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *Proceedings of the 23rd Annual International Cryptology Conference*, pages 96–109, 2003.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA* [DBL08], pages 76–85.

[VZ13] Salil Vadhan and Colin Jia Zheng. A uniform min-max theorem with applications in cryptography. In *Advances in Cryptology–CRYPTO 2013*, pages 93–110. Springer, 2013.