Non-Trivial Witness Encryption and Null-iO from Standard Assumptions

Zvika Brakerski* Aayush Jain† Ilan Komargodski‡ Alain Passelègue†
Daniel Wichs§

Abstract

A *witness encryption (WE)* scheme can take any NP statement as a public-key and use it to encrypt a message. If the statement is true then it is possible to decrypt the message given a corresponding witness, but if the statement is false then the message is computationally hidden. Ideally, the encryption procedure should run in polynomial time, but it is also meaningful to define a weaker notion, which we call *non-trivially exponentially efficient* WE (XWE), where the encryption run-time is only required to be much smaller than the trivial 2^m bound for NP relations with witness size m. We show how to construct such XWE schemes for all of NP with encryption run-time $2^{m/2}$ under the sub-exponential learning with errors (LWE) assumption. For NP relations that can be verified in NC^1 (e.g., SAT) we can also construct such XWE schemes under the sub-exponential Decisional Bilinear Diffie-Hellman (DBDH) assumption. Although we find the result surprising, it follows via a very simple connection to *attribute-based encryption*.

We also show how to upgrade the above results to get non-trivially exponentially efficient *indistinguishability obfuscation for null circuits (niO)*, which guarantees that the obfuscations of any two circuits that always output 0 are indistinguishable. In particular, under the LWE assumptions we get a XniO scheme where the obfuscation time is $2^{n/2}$ for all circuits with input size n. It is known that in the case of indistinguishability obfuscation (iO) for all circuits, non-trivially efficient XiO schemes imply fully efficient iO schemes (Lin et al., PKC '16) but it remains as a fascinating open problem whether any such connection exists for WE or niO.

Lastly, we explore a potential approach toward constructing fully efficient WE and niO schemes via multi-input ABE.

*Weizmann Institute of Science, Israel. Email: zvika.brakerski@weizmann.ac.il. Supported by the Israel Science Foundation (Grant No. 468/14), Binational Science Foundation (Grants No. 2016726, 2014276), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).

†UCLA, Los Angeles, USA. Emails: aayushjain@cs.ucla.edu,alapasse@gmail.com. Research supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

‡Cornell Tech, New York, USA. Email: komargodski@cornell.edu. Supported in part by a Packard Foundation Fellowship and by an AFOSR grant FA9550-15-1-0262. Most of this work was done at the Weizmann Institute of Science, supported by a grant from the Israel Science Foundation (no. 950/16) and by a Levzion Fellowship.

§Department of Computer Science, Northeastern University, Boston, USA. Email: wichs@ccs.neu.edu. Research supported by NSF grants CNS-1314722, CNS-1413964, CNS-1750795.

1 Introduction

In the last few years, much research in cryptography has focused on exploring powerful new cryptographic primitives such as *witness encryption* (WE) [GGSW13] and *indistinguishability obfuscation* (iO) [BGI+12, GGH+13b]. Although we have candidate constructions of these primitives, they rely on a new class of assumptions over *multilinear maps* (MMAPs) [GGH13a] whose computational hardness properties are poorly understood and we lack a high degree of confidence in their security. The grand challenge is to construct WE and iO under standard and well established hardness assumptions, such as the *learning with errors (LWE)* assumption [Reg05]. In this work we show that this is possible for a non-trivial relaxation of these primitives. But first, let us review what these primitives are.

Witness Encryption. Witness encryption (WE), introduced by Garg et al. [GGSW13], allows us to use an arbitrary NP statement x as a public key to encrypt a message. If x is a true statement then any user who knows the corresponding witness w for x will be able to decrypt the message, but if x is a false statement then the encrypted message is computationally hidden. For example, we could encrypt a bitcoin reward under the NP statement that corresponds to the Riemann hypothesis being true and having a proof of some polynomially bounded size. If anyone comes up with such a proof for the Riemann hypothesis, then they can use that as the witness to decrypt the ciphertext and recover the bitcoin reward.

Indistinguishability Obfuscation (for Null Circuits). The goal of obfuscation [BGI+12] is to convert a program/circuit C into a functionally equivalent program/circuit in a way that hides all aspects of the internal implementation of C, but still allows to evaluate it on arbitrary inputs. Ideally, seeing an obfuscated version of C would reveal nothing more than what one could learn via black-box access to the functionality that C implements. Unfortunately, this strong definition of obfuscation, called *virtual black box (VBB)* is known to be unachievable in general for all programs [BGI+12]. A weaker variant called *indistinguishability obfuscation (iO)* [BGI+12, GGH+13b] only insists that if two equal size circuits C, C' are functionally equivalent, meaning that $C(x) = C'(x)$ for all inputs x, then their obfuscations should be indistinguishable. A huge body of recent works starting with [SW14] shows how to use iO to construct a plethora of advanced cryptographic primitives for which no constructions were previously known. An even weaker variant called null iO (niO, see [WZ17, GKW17]) only insists that the obfuscations of C and C' are indistinguishable if the two circuits are both null circuits meaning that $C(x) = C'(x) = 0$ for all inputs x. Although security is only defined for null circuits, we still require the niO obfuscator to work correctly and preserve the functionality of all circuits, including ones that are not null.

It is obvious that iO implies niO and relatively easy to see that niO implies WE. In particular, to encrypt a message b under an NP statement x we can use an niO scheme to obfuscate the circuit $C[x, b]$ that outputs b given a valid witness w for x as an input and otherwise outputs 0; to argue security we rely on the fact that when x is not in the language then this is a null circuit. The works of [WZ17, GKW17] show that, under the Learning-With-Errors (LWE) assumption, witness encryption (WE) also implies null iO (niO). It remains as a major open problem whether niO implies full iO.

Non-Trivially Exponentially-Efficient Schemes. In the standard definition of witness encryption, the encryption procedure is required to run in polynomial time. Indeed, otherwise there would be a trivial perfectly secure witness encryption scheme where the encryption procedure simply checks whether the statement x is true (by trying every possible witness) and if so it outputs

the message in the clear and otherwise it outputs a dummy value as the ciphertext. For NP relations where the witness is of size m, the run-time of the trivial encryption procedure is $\widetilde{O}(2^m)$. Similarly, there are trivial perfectly secure iO and niO schemes where, for circuits with input size n, the obfuscation procedure runs in $\widetilde{O}(2^n)$ time and outputs the entire truth table of the circuit. Such schemes are trivially exponentially efficient.

We define the notion of *non-trivially exponentially efficient WE (XWE)* as a relaxation of WE where we require that for NP relations with witness length m, the encryption run-time is $\widetilde{O}(2^{\gamma m})$ for some constant $\gamma < 1$. Similarly, we define *non-trivially exponentially efficient niO (XniO)* analogously by requiring that for circuits with input size n the obfuscator run-time is $\widetilde{O}(2^{\gamma n})$ for some constant $\gamma < 1$. We call γ the *compression factor*. The above notions are analogous to the notion of non-trivially exponentially efficient iO (XiO) defined by Lin et al. [LPST16], which requires that the size of the obfuscated program is $\widetilde{O}(2^{\gamma n})$.[1] In [LPST16] it was shown that XiO implies fully efficient iO under the sub-exponential LWE assumptions. Unfortunately, we do not have any such connections showing that XWE implies WE or that XniO implies niO and it remains as an open problem to explore whether any such connections hold. Nevertheless, we believe that XWE and XniO are interesting relaxations of WE and niO and are worthy of study.

Our Results. We show how to construct XWE and XniO with compression factor $\gamma = \frac{1}{2}$ under the sub-exponential LWE assumption. For NP relations that can be verified in NC^1 (e.g., SAT) we also get XWE with compression factor $\gamma = \frac{1}{2}$ under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. Our constructions turn out to be extremely simple applications of *attribute based encryption (ABE)* [SW05, BSW12, GVW13, BGG+14].

Improving on our result and pushing the compression factor further below $\frac{1}{2}$ remains an open problem. Note that XWE and XniO with a sufficiently small compression factor $O(\log m/m)$ is equivalent to the standard notions of WE and niO respectively. Currently even achieving a compression factor of $\frac{1}{3}$ would be significant progress. Our only result in this direction is a scheme under the sub-exponential LWE assumption which achieves ciphertext length as short as $\widetilde{O}(2^{m/3})$, but at the cost of increasing the encryption complexity to $\widetilde{O}(2^{2m/3})$. We also suggest an approach for getting smaller compression factors and ultimately fully efficient WE and niO schemes via *multi-input* ABE. Unfortunately, we currently do not have any instantiation of this primitive under standard assumptions.

Our Techniques: from ABE to XWE. An (unbounded collusion) ABE scheme allows us to create ciphertexts $c = \mathsf{Enc}(\alpha, b)$ encrypting a message b with respect to an attribute α. Furthermore, we can release secret keys sk_f that are tied to some functions f. If $f(\alpha) = 1$ then the secret key sk_f can correctly decrypt c and recover b. However, given only secret keys $\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_p}$ for functions such that $f_1(\alpha) = \cdots = f_p(\alpha) = 0$, the ciphertext c cannot be decrypted and the message b remains hidden. We can use ABE to construct an XWE scheme for any NP language having witness size m where the running time of the encryption procedure is $\widetilde{O}(2^{m/2})$. To create a WE encryption of a message b under a statement x, we create $2^{m/2}$ secret keys $\mathsf{sk}_{f_{w_1}}$ for all choices of $w_1 \in \{0,1\}^{m/2}$ and we create $2^{m/2}$ ciphertexts $c_{w_2} = \mathsf{Enc}(w_2, b)$ for all choices of $w_2 \in \{0,1\}^{m/2}$, where we define the function $f_{w_1}(w_2) = 1$ if $w = w_1 w_2$ is a valid witness for the statement x. Given a witness $w = w_1 w_2$

[1] One difference is that XiO only restricts the size of the obfuscated programs but not the run-time of the obfuscation procedure, while XWE and XniO also restricts the run-time of the encryption and obfuscation procedures (which then implicitly restricts the size of the ciphertexts and obfuscated programs). This is important since, without restricting the run-time, trivial WE and niO constructions can achieve short ciphertext and obfuscated program sizes.

we can recover b by decrypting the ciphertext c_{w_2} with the secret key $\mathsf{sk}_{f_{w_1}}$.[2] However, if x is a false statement, we can rely on sub-exponential ABE security to argue that the bit b is computationally hidden. This gives us an XWE scheme with compression $\gamma = \frac{1}{2}$ by instantiating the ABE with known constructions based on LWE and DBDH. An analogous idea was used by Bitansky et al. [BNPW16] to go from *symmetric-key functional encryption* to XiO, but we currently do not have any constructions of the former primitive under any standard assumptions.

It turns out that the transformation from WE to niO from [WZ17, GKW17] also transforms XWE to XniO while preserving the compression factor and therefore, under the sub-exponential LWE assumption, the above technique also gives us XniO schemes with compression $\gamma = \frac{1}{2}$. Alternately, if we apply the above technique but start with a *predicate encryption (PE)* [GVW15] instead of ABE then the above transformation gives an XWE scheme where the ciphertext also hides the statement x (as long as it is a false statement) which is equivalent to XniO.

We show that the above technique can also be extended to get more general tradeoffs between encryption time, ciphertext size and decryption time in XWE. For example, under the sub-exponential LWE assumption, we can decrease the ciphertext size to $\widetilde{O}(2^{m/3})$ at the cost of increasing the encryption time to $\widetilde{O}(2^{2m/3})$.

In Appendix A, we also show that the above technique can be extended to getting a better compression factor by relying on multi-input ABE. In particular, if we had a k-input ABE scheme we would get an XWE scheme with compression factor $1/(k+1)$ for languages with instances of size n and witnesses of size $k \cdot \log n$.

Paper Organization. The rest of the paper is organized as follows: In Section 2, we recall basic cryptographic notions involved in this work. Our transform from ABE to non-trivially exponentially efficient witness encryption is then described in Section 3. The latter section also contains instantiations under standard assumptions and our extension to non-trivially exponentially efficient null-iO. Finally, Appendix A details our generalized transform from multi-input ABE. Definitions of null-iO and multi-input ABE are provided in the relevant sections.

2 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For a randomized function f and an input $x \in \mathcal{X}$, we denote by $y \leftarrow f(x)$ the process of sampling a value y from the distribution $f(x)$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. A function $\mathsf{neg} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{neg}(\lambda) < \lambda^{-c}$ for all $\lambda > N_c$. Throughout this paper we denote by λ the security parameter.

Two sequences of random variables $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are (t, ϵ)-*computationally indistinguishable* for $t = t(\lambda)$ and $\epsilon = \epsilon(\lambda)$, denoted by $X \approx_{t,\epsilon} Y$, if for any probabilistic distinguisher D that runs in time $t = t(\lambda)$, it holds that $\left|\Pr[D(1^\lambda, X_\lambda) = 1] - \Pr[D(1^\lambda, Y_\lambda) = 1]\right| \leq \epsilon(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$. We say that X, Y are *sub-exponentially indistinguishable* if they are (t, ϵ)-computationally indistinguishable with $t(\lambda) = 2^{\lambda^\delta}$ and $\epsilon(\lambda) = 2^{-\lambda^\delta}$ for some $\delta > 0$.

[2] Notice that in the RAM model, decryption is very efficient as it requires accessing only one key and one ciphertext.

2.1 Attribute-Based Encryption

We provide a definition of (key-policy, unbounded collusion) attribute-based encryption (ABE). We focus on the private-key variant which suffices for our purposes. An ABE scheme is a standard (private-key) encryption scheme for bits augmented with an additional key-generation procedure for an ensemble of Boolean function families $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ each mapping $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ to $\{0,1\}$, where $\mathcal{X}$ is some sequence of finite sets. Such a scheme is described by four procedures $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

1. $\mathsf{Setup}(1^\lambda)$ gets as input a security parameter and outputs a master secret key msk.
2. $\mathsf{KG}(\mathsf{msk}, f)$ gets as input a master secret key msk and a function $f \in \mathcal{F}_\lambda$ and outputs a key sk_f.
3. $\mathsf{Enc}(\mathsf{msk}, \alpha, b)$ gets as input a master secret key msk, an attribute $\alpha \in \mathcal{X}_\lambda$ and a message $b \in \{0,1\}$, and outputs a ciphertext $\mathsf{ct}_{\alpha,b}$. We assume, without loss of generality, that $\mathsf{ct}_{\alpha,b}$ contains α in the clear.
4. $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_{\alpha,b})$ gets as input a key for the function f and ciphertext of (α, b) and outputs a message b'.

The correctness and security of such a scheme are provided in the next definition.

Definition 2.1. A tuple of four procedures $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is said to be a (t, ϵ)-selectively-secure unbounded collusion ABE scheme if

1. **Correctness:** For every $\lambda \in \mathbb{N}$, $b \in \{0,1\}$, $\alpha \in \mathcal{X}$, $f \in \mathcal{F}$, it holds that if $f(\alpha) = 1$, then

$$\Pr[\mathsf{Dec}(\mathsf{KG}(\mathsf{msk}, f), \mathsf{Enc}(\mathsf{msk}, \alpha, b)) = b] = 1$$

   where the probability if over the choice of $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$ and over the internal randomness of KG and Enc.

2. **Security:** For every polynomial $p = p(\lambda)$, every (selectively chosen) $f_1, \ldots, f_p \in \mathcal{F}$, and every $\alpha_1, \ldots, \alpha_p \in \mathcal{X}$, it holds that if $f_i(\alpha_j) = 0$ for all $i, j \in [p]$, then

$$\{\mathsf{KG}(\mathsf{msk}, f_i), \mathsf{Enc}(\mathsf{msk}, \alpha_j, 0)\}_{i,j\in[p]} \approx_{t,\epsilon} \{\mathsf{KG}(\mathsf{msk}, f_i), \mathsf{Enc}(\mathsf{msk}, \alpha_j, 1)\}_{i,j\in[p]},$$

   where the randomness is over the choice of $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$ and the internal randomness of KG and Enc.

Known instantiations. There are several known constructions of ABE schemes based on different assumptions and offering various notions of efficiency. Three of the most well-known schemes are those of Goyal et al. [GPSW06], of Gorbunov et al. [GVW13], and of Boneh et al. [BGG+14]. The work of Goyal et al. gives a construction of an ABE scheme for all NC^1 circuits based on the existence of a bilinear map where the decisional bilinear Diffie-Hellman problem is hard.

Theorem 2.2 ([GPSW06])**.** *Assuming a group with a bilinear map in which the decisional bilinear Diffie-Hellman problem is sub-exponentially hard, there exists a sub-exponentially-secure ABE scheme for all* NC^1 *circuits.*

The works of Gorbunov et al. and of Boneh et al. achieved an ABE scheme for all a-priori depth-bounded polynomial-size circuits based on the sub-exponential hardness of the learning with errors assumption (LWE). Both of these ABE schemes satisfy that the key generation algorithm runs in time $|f| \cdot \mathsf{poly}(\lambda, d)$ on input a function f of depth d. We call this property *time-efficient key generation*. The scheme by Boneh et al. has an additional unique property that we will use: The size of an ABE functional key is independent of the size of the function and only depends on its depth. Specifically, given a function $f \in \mathcal{F}$, the size of a functional key for it is $\mathsf{poly}(d, \lambda)$ for some fixed polynomial function poly. We henceforth call this property *short functional keys*. Note that in order to decrypt, the description of f needs to be provided in addition to the key sk_f.

Theorem 2.3 ([BGG+14])**.** *Assuming the sub-exponential hardness of LWE, there exists a sub-exponentially-secure ABE scheme with time-efficient key generation and short functional keys.*

2.2 Witness Encryption for NP

Definition 2.4 (Witness encryption [GGSW13])**.** A *witness encryption* scheme for an NP relation $R \subseteq \left\{\{0,1\}^n \times \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$ with induced language L has the following syntax:

- $\mathsf{Enc}(1^\lambda, x, b)$: Takes as input a security parameter 1^λ, a string $x \in \{0,1\}^n$ and a bit $b \in \{0,1\}$, and outputs a ciphertext $\mathsf{ct}_{x,b}$.
- $\mathsf{Dec}(\mathsf{ct}, w)$: Takes as input a ciphertext $\mathsf{ct}_{x,b}$ and a string $w \in \{0,1\}^m$, and outputs a bit b' or the symbol $\perp$.

These algorithms satisfy the following two conditions:

1. **Correctness:** For any security parameter λ, any $b \in \{0,1\}$ and any $x \in L$ with witness w, it holds that
$$\Pr[\mathsf{Dec}(\mathsf{Enc}(1^\lambda, x, b), w) = b] = 1,$$
where the probability is over the internal randomness of the encryption procedure.
2. **Security:** A witness encryption scheme is (t, ϵ)-secure if for every ensemble $x = \{x_\lambda\}$ of false statements $x_\lambda \notin L$ it holds that
$$\mathsf{Enc}(1^\lambda, x_\lambda, 0) \approx_{t,\epsilon} \mathsf{Enc}(1^\lambda, x_\lambda, 1)$$
where the randomness is over the internal randomness of the encryption procedure.

3 Non-Trivial Witness Encryption and ABE

In this section we show that any attribute encryption scheme directly implies a non-trivially exponentially-efficient witness encryption scheme (XWE). This gives us a construction of the latter under the DBDH or LWE assumptions. Lastly, we recall the notion of null-iO, define non-trivially exponentially-efficient null-iO (XniO) and construct it based on previously built XWE.

3.1 Non-Trivially Exponentially-Efficient Witness Encryption

Our notion of exponentially-efficient witness encryption (XWE) allows the encryptor to have running time almost as large as the brute-force algorithm that solves the instance. This is analogous to the notion of XiO introduced by Lin et al. [LPST16] which requires the size of an obfuscation to be slightly smaller than the truth-table of the function. See comparison below.

Definition 3.1. A witness encryption scheme for a relation $R \subseteq \left\{\{0,1\}^n \times \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$ with induced language L is said to be γ-exponentially-efficient if for any $\lambda, n \in \mathbb{N}$ with $m = m(n)$ and every instance $x \in \{0,1\}^n$ and $b \in \{0,1\}$, the run-time of $\mathsf{Enc}(1^\lambda, x, b)$ is at most $2^{\gamma m} \cdot \mathsf{poly}(\lambda, n)$.

Comparison with XiO and SXiO. The notion of XiO, introduced by Lin et al. [LPST16], requires an obfuscator to output a circuit of size $2^{\gamma n} \cdot \mathsf{poly}(\lambda, |C|)$ given a circuit C that accepts n bits as input. This notion has been proven to be very useful in constructions of iO when combined with LWE. SXiO is a strengthening of XiO in which we require not only the obfuscated circuit to be of non-trivial size, but also the running time of the obfuscator.

Our notion of XWE only concerns the time it takes to encrypt a bit (which gives an upper bound on the size of the obfuscation). The reason is that an encryptor can always brute-force all possible witnesses and try each one to decide whether the instance is in the language or not. If so, it can output the message in the clear, and if not it can output some fixed output (recall that in WE correctness holds only for instances that are in the language while security is required only for instances that are not in the language).

3.2 From ABE to Non-Trivial Witness Encryption

We observe a connection between ABE schemes and exponentially-efficient WE schemes. This is similar to the observation of [BNPW16] in the context of functional encryption and exponentially-efficient iO. However, in our case we will be able to instantiate our ABE scheme based on somewhat standard assumptions.

Theorem 3.2. *Let $R \subseteq \left\{\{0,1\}^n \times \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$ be an NP relation with induced language L. Assume the existence of a sub-exponentially-secure ABE scheme for all circuits. Then, there exists a polynomial poly and a witness encryption scheme for R with the following properties. For any $\lambda, n \in \mathbb{N}$ with $m = m(n)$ and every instance $x \in \{0,1\}^n$ and $b \in \{0,1\}$:*

1. *The run-time of the encryption procedure $\mathsf{Enc}(1^\lambda, x, b)$ is at most $2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$.*
2. *The ciphertext size is at most $2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$.*
3. *The decryption time is at most $2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$. In particular, it is $\mathsf{poly}(\lambda, n, m)$ in the* RAM *model.*[3]

Proof. Assume that we have an ABE scheme $\mathsf{ABE} = (\mathsf{ABE.Setup}, \mathsf{ABE.KG}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ for all circuits. The ABE scheme is sub-exponentially-hard so when instantiated with security parameter λ, no adversary that runs in time 2^{λ^τ} can break it for a constant $\tau > 0$. We construct a witness encryption scheme $\mathsf{WE} = (\mathsf{WE.Enc}, \mathsf{WE.Dec})$.

[3]The property that in the RAM model our decryption is very efficient is common to all of our results. We only state it here and avoid repeating it in the other results.

Denote by $V^{(L)}$ the verification procedure of the NP language L. This procedure gets as input x and a possible witness w split into two parts w_1 and w_2, and it outputs a bit that specifies whether w is a valid witness attesting to the fact that $x \in L$. Given an instance $x \in \{0,1\}^n$ and a message $b \in \{0,1\}$, the witness encryption $\mathsf{WE.Enc}(1^\lambda, x, b)$ is computed as follows:

1. Sample a master secret key for the ABE scheme $\mathsf{msk} \leftarrow \mathsf{ABE.Setup}(1^{\tilde{\lambda}})$, where $\tilde{\lambda} = \max\{\lambda, m^{2/\tau}\}$.

2. For every $w_1 \in \{0,1\}^{m/2}$, use the ABE scheme to generate a key for the function $V^{(L)}_{x,w_1}(w_2) = V^{(L)}(x, w_1 w_2)$:

$$\mathsf{sk}_{f,w_1} \leftarrow \mathsf{ABE.KG}(\mathsf{msk}, V^{(L)}_{x,w_1}).$$

3. For every $w_2 \in \{0,1\}^{m/2}$, use the ABE scheme to encrypt b under attribute w_2:

$$\mathsf{ct}_{w_2,b} \leftarrow \mathsf{ABE.Enc}(\mathsf{msk}, w_2, b).$$

4. Output $\{\mathsf{sk}_{f,w_1}\}_{w_2 \in \{0,1\}^{m/2}}$ and $\{\mathsf{ct}_{w_2,b}\}_{w_2 \in \{0,1\}^{m/2}}$.

To decrypt $\mathsf{WE.Dec}(\mathsf{ct}, w)$, where

$$\mathsf{ct} = (\{\mathsf{sk}_{f,w_1}\}_{w_2 \in \{0,1\}^{m/2}}, \{\mathsf{ct}_{w_2,b}\}_{w_2 \in \{0,1\}^{m/2\}})$$

and $w = w_1 w_2 \in \{0,1\}^m$, we execute the decryption procedure of the ABE scheme as follows:

$$\mathsf{ABE.Dec}(\mathsf{sk}_{f,w_1}, \mathsf{ct}_{w_2,b}).$$

Correctness immediately follows from the correctness of the underlying ABE scheme. Security also easily follows from the security of the latter. Namely, if $x \notin L$, then for any $w_1 w_2 \in \{0,1\}^m$, we have $V^{(L)}(x, w_1 w_2) = 0$. Let ct denote an encryption of 0 for a statement $x \notin L$, that is:

$$\mathsf{ct} = \mathsf{WE.Enc}(1^\lambda, x, 0) = (\{\mathsf{sk}_{f,w_1}\}_{w_2 \in \{0,1\}^{m/2}}, \{\mathsf{ct}_{w_2,0}\}_{w_2 \in \{0,1\}^{m/2}}) \ .$$

For security, first observe that we instantiated our ABE scheme with security parameter $\tilde{\lambda} = \max\{\lambda, m^{2/\tau}\}$. This means that our scheme is secure against adversaries that run in time $\max\{2^{\lambda^\tau}, 2^{m^2}\}$. In particular, it is secure for all adversaries running in time $\mathsf{poly}(2^m)$ which is the size of our ciphertext (see below). Moreover, since for any $w_1, w_2 \in \{0,1\}^{m/2}$, we have $V^{(L)}(x, w_1 w_2) = 0$, it is clear that, assuming the security of ABE, $\mathsf{ct}_{w_2,0} \approx_c \mathsf{ct}_{w_2,1}$, and security follows.

Let us analyze the complexity of the scheme and in particular the running time of the encryption procedure. When encrypting a message b under instance x our scheme generates and outputs $2^{m/2}$ functional keys (for a function whose complexity is at most the complexity of $V^{(L)}$) and $2^{m/2}$ ciphertexts of the underlying ABE scheme. This takes time at most

$$2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$$

for some fixed polynomial poly which depends on the complexity of encryption of the underlying ABE scheme and the complexity of $V^{(L)}$. The same bound holds for the ciphertext size. Decryption upon witness $w = w_1 w_2$ requires reading the functional key and ciphertext and a single invocation of the decryption procedure of the underlying ABE scheme on the key for the function $f(w_1, \cdot) = V^{(L)}_{x,w_1}(\cdot)$ and the ciphertext that corresponds to w_2. ■

3.3 Instantiations

We instantiate Theorem 3.2 using known attribute-based encryption schemes mentioned in Section 2.1. The first construction of Goyal et al. [GPSW06] which works only for NC^1 circuits and is based on the decisional bilinear Diffie-Hellman assumption leads to non-trivially exponentially-efficient witness encryption for any NP relation with verification in NC^1. One can also instantiate a similar corollary based on the LWE-based constructions of Gorbunov et al. [GVW13] and of Boneh et al. [BGG+14] and get a construction that works for *all languages* with a polynomial-size circuit verifier, so for any NP relation.

Corollary 3.3. *Let $R \subseteq \left\{\{0,1\}^n \times \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$ be an NP relation with induced language L. Assume the sub-exponential security of the learning with errors assumption. Then, there exists a polynomial poly and a sub-exponentially-secure witness encryption scheme $\mathsf{WE} = (\mathsf{WE.Enc}, \mathsf{WE.Dec})$ for R with the following properties:*

1. *The time it takes to encrypt a bit is at most $2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$.*
2. *The ciphertext size is at most $2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$.*
3. *The decryption time is at most $2^{m/2} \cdot \mathsf{poly}(\lambda, n, m)$.*

Moreover, assuming also that the verification for L is in NC^1, the same is true assuming the sub-exponential security of the decisional bilinear Diffie-Hellman assumption.

A variant based on ABE with short functional keys. Below we provide a variant of Theorem 3.2 in which we take advantage of an ABE scheme that has a particular notion of succinctness we referred to as *short functional keys*[4]. This property is satisfied by the LWE-based scheme by Boneh et al.

Theorem 3.4. *Let $R \subseteq \left\{\{0,1\}^n \times \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$ be an NP relation with induced language L. Assume an attribute-based encryption scheme for all circuits with time-efficient key generation and short functional keys. Let $m_1(n), m_2(n), m_3(n) \geq 0$ be polynomials such that $m_1 + m_2 + m_3 = m$. Then, there exists a sub-exponentially-secure witness encryption scheme with the following properties:*

1. *The time it takes to encrypt a bit is at most $2^{\max\{m_1+m_3, m_2\}} \cdot \mathsf{poly}(\lambda, n, m)$.*
2. *The ciphertext size is at most $2^{\max\{m_1, m_2\}} \cdot \mathsf{poly}(\lambda, n, m)$.*
3. *The decryption time is at most $2^{\max\{m_1, m_2, m_3\}} \cdot \mathsf{poly}(\lambda, n, m)$.*

Proof. Assume that we have a ABE scheme $\mathsf{ABE} = (\mathsf{ABE.Setup}, \mathsf{ABE.KG}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ with time-efficient key generation and short functional keys. The ABE scheme is secure for adversaries running in time 2^{λ^τ} for a constant $\tau > 0$. We construct a witness encryption scheme $\mathsf{WE} = (\mathsf{WE.Enc}, \mathsf{WE.Dec})$.

Given an instance $x \in \{0,1\}^n$ and a message $b \in \{0,1\}$, the witness encryption $\mathsf{WE.Enc}(1^\lambda, x, b)$ is done as follows:

[4]Recall that a scheme with short functional keys has the property that the size of a functional key for a function of size s and depth d is $\mathsf{poly}(d, \lambda)$ for some fixed polynomial function poly.

1. Sample a master secret key for the ABE scheme $\mathsf{msk} \leftarrow \mathsf{ABE.KG}(1^{\tilde{\lambda}})$, where $\tilde{\lambda} = \max\{\lambda, m^{2/\tau}\}$.

2. For every $w_1 \in \{0,1\}^{m_1}$, use the ABE scheme to generate a key for the function $V^{(L)}_{x,w_1}(w_2) = \bigvee_{w_3 \in \{0,1\}^{m_3}} V^{(L)}(x, w_1 w_2 w_3)$:

$$\mathsf{sk}_{f,w_1} \leftarrow \mathsf{ABE.KG}(\mathsf{msk}, V^{(L)}_{x,w_1}).$$

3. For every $w_2 \in \{0,1\}^{m_2}$, use the ABE scheme to encrypt b under attribute w_2:

$$\mathsf{ct}_{w_2,b} \leftarrow \mathsf{ABE.Enc}(\mathsf{msk}, w_2, b).$$

4. Output $\{\mathsf{sk}_{f,w_1}\}_{w_1 \in \{0,1\}^{m_1}}$ and $\{\mathsf{ct}_{w_2,b}\}_{w_2 \in \{0,1\}^{m_2}}$.

Correctness is immediate and security follows as in the proof of Theorem 3.2, since for $x \notin L$, there are no w_1 and w_2 for which $V^{(L)}_{x,w_1}(w_2)$ evaluates to 1. Thus, we can directly reduce security of our construction to the security of the underlying ABE scheme.

Given $x \in \{0,1\}^m$ and $b \in \{0,1\}$, the time it takes to compute $\mathsf{Enc}(1^\lambda, x, b)$ is at most

$$2^{m_1} \cdot (|V^{(L)}_{x,w_1}| \cdot \mathsf{poly}(\lambda, d)) + 2^{m_2} \cdot \mathsf{poly}(\lambda, n, m),$$

where d is the depth of the circuit $V^{(L)}_{x,w_1}$ (recall that the LWE-based ABE scheme has time-efficient key generation; see Theorem 2.3). Notice that d is bounded by the depth of $V^{(L)}$ which is at most some polynomial in n and m. Furthermore, notice that $|V^{(L)}_{x,w_1}|$, the size of $V^{(L)}_{x,w_1}$, is at most 2^{m_3} times some polynomial in n and m. Overall, we get that the time it takes to generate a ciphertext is at most

$$2^{\max\{m_1+m_3, m_2\}} \cdot \mathsf{poly}(\lambda, n, m).$$

The size of a ciphertext is shorter because the size of a key does not depend on the size of the function but only on its depth (which is $\mathsf{poly}(n, m)$). This means that the ciphertext size is

$$(2^{m_1} + 2^{m_2}) \cdot \mathsf{poly}(\lambda, n, m) = 2^{\max\{m_1, m_2\}} \cdot \mathsf{poly}(\lambda, n, m).$$

For decryption, one needs to read the whole ciphertext and perform a single decryption operation of the underlying ABE scheme. However, notice that the size of the function is $2^{m_3} \cdot \mathsf{poly}(\lambda, n, m)$ which means that time to decrypt is at most:

$$2^{\max\{m_1, m_2, m_3\}} \cdot \mathsf{poly}(\lambda, n, m).$$

Note that for decryption, the description of the function must be known. This can be done by providing a (single) generic description of

$$V_{x,\cdot}(w_2) = \bigvee_{w_3 \in \{0,1\}^{m_3}} V^{(L)}(x, \cdot || w_2 || w_3)$$

as a public parameter. ■

We then obtain the following corollary using the construction by Boneh et al. [BGG+14] in Theorem 3.4 with $m_1 = m_2 = m_3 = m/3$.

Corollary 3.5. *Let $R \subseteq \left\{\{0,1\}^n \times \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$ be an* NP *relation with induced language L. Assuming the sub-exponential hardness of the learning with errors problem, there exists a sub-exponentially-secure witness encryption scheme* $\mathsf{WE} = (\mathsf{WE.Enc}, \mathsf{WE.Dec})$ *for R with the following properties:*

1. *The time it takes to encrypt a bit is at most $2^{2m/3} \cdot \mathsf{poly}(\lambda, n, m)$.*
2. *The ciphertext size is at most $2^{m/3} \cdot \mathsf{poly}(\lambda, n, m)$.*
3. *The decryption time is at most $2^{m/3} \cdot \mathsf{poly}(\lambda, n, m)$.*

3.4 A Similar Transformation for Null-iO

A similar result, i.e., a non-trivially exponentially-efficient construction based on the LWE assumption, can be obtained for a weakening of iO called null-iO (niO, see [WZ17, GKW17]). An niO is an obfuscation scheme which takes as input an arbitrary circuit and outputs a functionally equivalent one but security only guarantees that we cannot distinguish the obfuscations of any two circuits C, C' of the same size such that $C(x) = C'(x) = 0$ for all inputs x.

Definition 3.6 (Null-iO)**.** A null-iO (niO) obfuscation scheme is an efficient compiler $\mathcal{O}$ for circuits that satisfies the following properties:

1. **Correctness:** For any security parameter λ and all circuits $C\colon \{0,1\}^n \to \{0,1\}$:
$$\Pr[\forall x \in \{0,1\}^n : C(x) = \tilde{C}(x) | \tilde{C} \leftarrow \mathcal{O}(1^\lambda, C)] = 1,$$
where the probability is taken over the randomness of $\mathcal{O}$.
2. **Security:** Let $C = \{C_\lambda\}$, $C' = \{C'_\lambda\}$ be two ensembles of circuits with equal input length $n(\lambda)$ and circuit size, which satisfy $C_\lambda(x) = C'_\lambda(x) = 0$ for all $x \in \{0,1\}^{n(\lambda)}$. Then, we have that:
$$\mathcal{O}(1^\lambda, C_\lambda) \approx_{t,\epsilon} \mathcal{O}(1^\lambda, C'_\lambda).$$

It is natural to define the exponentially-efficient version of niO such that the running time of the obfuscator (and thus the size of the obfuscated circuit as well) is smaller than 2^n.

Definition 3.7 (XniO)**.** A null-iO is said to be γ-exponentially-efficient (XniO) if for any security parameter $\lambda \in \mathbb{N}$ and every circuit C, the running time obfuscation $\mathcal{O}(1^\lambda, C)$ is at most $2^{\gamma n} \cdot \mathsf{poly}(|C|)$.

In a recent work, Wichs and Zirdelis [WZ17] showed that assuming LWE one can generically translate any witness encryption scheme into a niO. Thus, using our Corollary 3.3 (instantiated with LWE) together with their transformation, we get a 1/2-XniO (for all polynomial-size circuits) assuming sub-exponentially-secure LWE. Using our Corollary 3.5 together with their transformation, we get an XniO whose running time is $2^{2n/3}$ and such that the size of the obfuscated circuit is $2^{n/3}$, assuming sub-exponentially-secure LWE.

Remark 3.8. A different way to get the same result is to directly construct an XniO based on any predicate encryption scheme [GVW15], similarly to our construction of an XWE based on any ABE scheme.

Acknowledgments

We thank Nir Bitansky for many initial discussions on the topics of this work. We thank Antigoni Polychroniadou and Hoeteck Wee for their helpful comments on a previous version of our work. We also thank the anonymous reviewers for their remarks.

References

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Advances in Cryptology - EUROCRYPT*, pages 533–556, 2014.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012. Preliminary version appeared in CRYPTO 2001.

[BNPW16] Nir Bitansky, Ryo Nishimaki, Alain Passelègue, and Daniel Wichs. From Cryptomania to Obfustopia through secret-key functional encryption. In *Theory of Cryptography - 14th International Conference, TCC 2016-B*, pages 391–418, 2016.

[BSW12] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11):56–64, 2012.

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *Advances in Cryptology - EUROCRYPT*, pages 578–602, 2014.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17. Springer, Heidelberg, May 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Symposium on Theory of Computing Conference, STOC*, pages 467–476, 2013.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 612–621. IEEE Computer Society, 2017.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pages 89–98. ACM, 2006.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554. ACM, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523. Springer, 2015.

[KNY17] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. *J. Cryptology*, 30(2):444–469, 2017.

[KS17] Ilan Komargodski and Gil Segev. From Minicrypt to Obfustopia via private-key functional encryption. In *Advances in Cryptology - EUROCRYPT*, pages 122–151, 2017.

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In *Public-Key Cryptography - PKC*, pages 447–462, 2016.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, Heidelberg, May 2005.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484. ACM Press, May / June 2014.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 600–611. IEEE Computer Society, 2017.

A Multi-Input ABE and Non-Trivial Witness Encryption

In this section, we introduce the notion of multi-input attribute based encryption and show that, in the most general setting, it implies witness encryption for NP.

Recall that in a standard ABE scheme, one can encrypt a message b relative to some attribute α to get $\mathsf{ct}_{\alpha,b}$ and independently generate keys for Boolean functions f to get sk_f. Together, $\mathsf{ct}_{\alpha,b}$ and sk_f can be used to recover b if $f(\alpha) = 1$, and otherwise, b should remain computationally hidden. We extend this notion to the multi-input setting. Here f takes as input a sequence of attributes $\alpha_1, \ldots, \alpha_k$ for $k \geq 1$ and the encryption functionality takes an additional parameter $i \in [k]$ (it ignores b for $i \neq 1$). Given ciphertexts $\mathsf{ct}_{\alpha_1,b}, \mathsf{ct}_{\alpha_2,\cdot}, \ldots, \mathsf{ct}_{\alpha_k}$ and a key sk_f for such a function, one is able to recover b if $f(\alpha_1, \ldots, \alpha_k) = 1$ while it should remain hidden if $f(\alpha_1, \ldots, \alpha_k) = 0$. Details follow.

A k-input ABE scheme is parametrized over an attribute space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and function space $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$, where each function maps $\mathcal{X} = \{(\mathcal{X}_\lambda)^k\}_{\lambda\in\mathbb{N}}$ to $\{0,1\}$. Such a scheme is described by four procedures $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

1. $\mathsf{Setup}(1^\lambda)$ gets as input a security parameter and outputs a master secret key msk.
2. $\mathsf{KG}(\mathsf{msk}, f)$ gets as input a master secret key msk and a function $f \in \mathcal{F}_\lambda$ and outputs a key sk_f.
3. $\mathsf{Enc}(\mathsf{msk}, \alpha, b, i)$ gets as input a master secret key msk, an attribute $\alpha \in \mathcal{X}_\lambda$ and a message $b \in \{0,1\}$ and an index $i \in [k]$, and outputs a ciphertext $\mathsf{ct}_{\alpha,b,i}$.
4. $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_{\alpha_1,b_1,1}, \ldots, \mathsf{ct}_{\alpha_k,b_k,k})$ gets as input a key for the function f and a sequence of ciphertext of $(\alpha_1, b_1), \ldots, (\alpha_k, b_k)$ and outputs a string b'.

The correctness and security of such a scheme are provided in the next definition.

Definition A.1. A tuple of four procedures $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is a k-input (t,ϵ)-secure ABE scheme if

1. **Correctness:** For every $\lambda \in \mathbb{N}$, $b_1, \ldots, b_k \in \{0,1\}$, $\alpha_1, \ldots, \alpha_k \in \mathcal{X}$, $f \in \mathcal{F}$, it holds that if $f(\alpha_1, \ldots, \alpha_k) = 1$, then

$$\Pr[\mathsf{Dec}(\mathsf{KG}(\mathsf{msk}, f), \mathsf{Enc}(\mathsf{msk}, \alpha_1, b_1, 1), \ldots, \mathsf{Enc}(\mathsf{msk}, \alpha_k, b_k, k)) = b_1] = 1$$

   where the probability if over the choice of $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$ and over the internal randomness of KG and Enc. Note that only messages encrypted at index 1 can be recovered, thus every message encrypted at a different index could be set to $\perp$ in our definition at the cost of a slightly more complex syntax.

2. **Security:** For every polynomial $p = p(\lambda)$, every $\vec{\alpha}_1, \ldots, \vec{\alpha}_p$, where $\vec{\alpha}_i = (\alpha_1^{(i)}, \ldots, \alpha_k^{(i)}) \in \mathcal{X}^k$ for $i \in [p]$, and every $f_1, \ldots, f_p \in \mathcal{F}$, it holds that if $f_i(\alpha_1^{i_1}, \ldots, \alpha_k^{i_k}) = 0$ for every $i, i_1, \ldots, i_k \in [p]$, then

$$\begin{aligned}
&\{\mathsf{KG}(\mathsf{msk}, f_i)\}_{i\in[p]}, \left\{\mathsf{Enc}(\mathsf{msk}, \alpha_j^{(i)}, 0, j)\right\}_{i\in[p], j\in[k]} \approx_{t,\epsilon} \\
&\{\mathsf{KG}(\mathsf{msk}, f_i)\}_{i\in[p]}, \left\{\mathsf{Enc}(\mathsf{msk}, \alpha_j^{(i)}, 1, j)\right\}_{i\in[p], j\in[k]},
\end{aligned}$$

   where the randomness is over the choice of $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$ and the internal randomness of KG and Enc.

In the next lemma we show that a general-purpose poly-input ABE scheme implies a witness encryption scheme. This is similar to an analogous statement in the functional encryption literature which says that a general purpose multi-input functional encryption scheme implies indistinguishability obfuscation for all circuits [GGG+14].

Lemma A.2. *Let $L \in \mathsf{NP}$ be a language where instances are of size $n = n(\lambda)$ and witnesses are of size $m = m(\lambda)$. An m-input ABE scheme for all polynomial-size circuits implies a witness encryption scheme for L.*

Proof. Let $\mathsf{MIABE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be the m-input ABE scheme. Denote by $V^{(L)}$ the verification procedure of the NP language L. This procedure gets as input x and a possible witness w split into m bits $w_1, \ldots, w_m$, and it outputs a bit that specifies whether w is a valid witness attesting to the fact that $x \in L$. Given an instance $x \in \{0,1\}^n$ and a message $b \in \{0,1\}$, the witness encryption $\mathsf{Enc}(1^\lambda, x, b)$ is computed as follows:

1. Sample a master secret key for the multi-input ABE scheme $\mathsf{msk} \leftarrow \mathsf{KG}(1^\lambda)$.
2. Use the ABE scheme to generate a key for the function $V_x^{(L)}(w_1, \ldots, w_m) = V^{(L)}(x, w_1 \ldots w_m)$:
$$\mathsf{sk}_f \leftarrow \mathsf{KG}(\mathsf{msk}, V_x^{(L)}).$$
3. For $\ell \in \{0,1\}$ and $i \in [m]$, use the ABE scheme to encrypt b under attribute ℓ for the index i:
$$\mathsf{ct}_{\ell,b,i} \leftarrow \mathsf{Enc}(\mathsf{msk}, \ell, b, i).$$
4. Output sk_f, $\{\mathsf{ct}_{\ell,b,i}\}_{\ell \in \{0,1\}, i \in [m]}$.

To decrypt a ciphertext $\mathsf{ct} = (\mathsf{sk}_f, \{\mathsf{ct}_{\ell,b,i}\}_{\ell \in \{0,1\}, i \in [m]})$ with respect to a witness $w = w_1 \ldots w_m \in \{0,1\}^m$, we execute the decryption procedure of the ABE scheme as follows:

$$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_{w_1,b,1}, \ldots, \mathsf{ct}_{w_m,b,m}).$$

The correctness and security of the witness encryption scheme follow immediately from the correctness and security of the underlying multi-input ABE scheme. Correctness holds since given a valid witness w for which $V^{(L)}(x, w) = 1$, the ABE decryption procedure will output b. Security holds since for any $x \notin L$, there is no witness for which $V^{(L)}$ accepts x and thus $V_x^{(L)}$ is always 0, which means that no combination of ciphertexts will lead to a successful decryption. The latter, by the security of the underlying ABE scheme implies that b is computationally hidden. ■

Using fewer-input ABE. Variants of the above theorem can be obtained in case we only have an ABE scheme that supports less inputs. Specifically, similarly to the refinement of [BNPW16] of the result of [GGG+14] mentioned above (see [KS17, Lemma 4.2] for the precise statement), one can show that a k-input ABE scheme for $k = k(\lambda)$ implies a witness encryption scheme for languages with instances of size $n = n(\lambda)$ and witnesses of size $k \cdot \log n$. This means that a k-input ABE scheme for any $k = \omega(1)$, is interesting as it could lead to non-trivial constructions of secret sharing schemes for all NP based on somewhat weaker assumptions than currently known [KNY17].