Witness Indistinguishability for any Single-Round Argument with Applications to Access Control [*]

Zvika Brakerski[†] Yael Tauman Kalai[‡]

Abstract

Consider an access policy for some resource which only allows access to users of the system who own a certain set of attributes. Specifically, we consider the case where such an access structure is defined by some *monotone* function $f : \{0,1\}^N \to \{0,1\}$, belonging to some class of function F (e.g. conjunctions, space bounded computation), where N is the number of possible attributes.

In this work we show that any succinct single-round delegation scheme for the function class F can be converted into a *succinct* single-round *private* access control protocol. That is, a verifier can be convinced that an approved user (i.e. one which holds an approved set of attributes) is accessing the system, without learning any additional information about the user or the set of attributes.

As a main tool of independent interest, we show that assuming a quasi-polynomially secure two-message oblivious transfer scheme with statistical sender privacy (which can be based on quasi-polynomial hardness of the DDH, QR, DCR or LWE assumptions), we can convert *any* single-round protocol into a *witness indistinguishable* one, with similar communication complexity.

1 Introduction

The main goal in the study of *delegation of computation* is to construct a single-round *succinct* argument system for a wide class of functions, in which the communication complexity and verification computational complexity are independent (or at least sublinear) in the computational complexity of deciding the statement, and where the prover (given a witness if needed) can compute a proof efficiently (i.e. with comparable complexity to that of deciding the statement). Delegation schemes for polynomially computable functions under standard assumptions were presented by [GKR08, KRR13, KRR14, KP15, RRR16, BHK17, KPY18]. In this work, we consider delegation for **NP**. Constructing delegation for all of **NP** under standard assumptions is an important open problem, and such schemes are only known in the random oracle model [Mic94], and under knowledge assumptions [DFH12, BCCT13, BCC$^+$14]. However, for restricted classes of **NP** languages, there are delegation schemes from standard assumptions [BHK17, BKK$^+$17].

[*] Prior versions of this ePrint report contained an additional result about a new batch delegation scheme for monotone formulae (see ePrint history for this report). We removed that result from this work because we believe that it better fits into a different work which is currently in preparation.

[†] Weizmann Institute of Science. Supported by the Israel Science Foundation (Grant No. 468/14), Binational Science Foundation (Grants No. 2016726, 2014276) and European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).

[‡] Microsoft Research and Massachusetts Institute of Technology.

When delegating an **NP** statement, the prover needs to hold a witness that allows to decide the statement. In such a case a natural question is whether the privacy of the witness is preserved by the delegation scheme. In this work we show a general transformation that translates any delegation scheme into a witness indistinguishable one, without blowing up the communication by much. We then apply this transformation to known delegation schemes based on standard assumptions, and construct an object that we call "succinct access control scheme". These objects allow a master authority to distribute credentials of attributes to parties, in a way that will allow them to provide a succinct proof that the credentials that they hold satisfy a predicate, without revealing the credentials or their identity.

1.1 Our Witness Indistinguishability Transformation

We show a *generic transformation* that converts any single-round (2-message) delegation scheme into one that is also *witness indistinguishable* (WI), *without blowing up the communication complexity.* This transformation relies on the existence of a quasi-poly secure OT scheme, which can be based on the quasi-polynomial hardness of the DDH, QR, Paillier's decisional composite residuosity assumption (DCR) and recently also the Learning with Errors assumption (LWE). The communication complexity and verifier complexity remain unchanged up to poly(λ) factors. This transformation relies on a recent 2-message strong WI protocol in the delayed input setting, proposed by [JKKR17]. (In this work we achieve computational WI, but we believe it may be possible to achieve statistical witness indistinguishability using the results and techniques of [KKS18].) See details in Section 1.3 and Section 2.

It should be noted that the high level approach of executing a delegation scheme homomorphically in order to achieve privacy for the witness can be traced back to prior works, e.g. [BBK$^+$16]. However, our result statement and analysis are different from what is done in prior works.

1.2 Application: Succinct Single-Round Access Control

By applying our WI transformation to a class of succinct single-round argument systems in the literature, that we call "batch **NP** families", we get a succinct single-round witness indistinguishable argument system that allows a user to prove that they contain a set of attributes that satisfies a given monotone access structure. We call this "a succinct access control scheme". We start by explaining what delegation for batch **NP** family is, and proceed with our construction.

Delegation for Batch NP Families. The work of [BHK17] considered a special setting of delegation for **NP** languages. They considered a conjunction (AND function) of a number of "small' **NP** statements, and showed a delegation protocol whose communication complexity scaled with the witness length of a small statement, rather than a concatenation of the witnesses. We can consider an extension of this paradigm, replacing the conjunction with other classes of functions. Note that this only makes sense for monotone functions, since a prover can always claim not to have a witness for a specific small instance.

Formally, our batch **NP** families will be characterized by a family of monotone functions F. The statements to be proven will be characterized by a collection of instances $x_1, \ldots, x_N$ respective to a language L, and a monotone function (i.e. without negation gates) $f : \{0,1\}^N \to \{0,1\}$ in F. The statement $((x_1, \ldots, x_N), f)$ holds if $f(\mathbf{1}_{x_1 \in L}, \ldots, \mathbf{1}_{x_N \in L}) = 1$, where $\mathbf{1}_{x_i \in L} = 1$ if and only if $x_i \in L$. For example, we can consider statements of the form $(((x_1 \in L) \wedge (x_2 \in L)) \vee (x_3 \in L)) \wedge (x_4 \in L)$, and much more. In order to produce an accepting proof, an honest prover needs a set of witnesses for a subset $S \subseteq [N]$ of the x_i's that makes f accept. Namely a set of witnesses $\{w_i\}_{i \in S}$ so that w_i is a witness for x_i and the set S is sufficient for f to accept; i.e., $f(\mathbf{1}_{1 \in S}, \ldots, \mathbf{1}_{N \in S}) = 1$. Since f is monotone, this indeed implies that $f(\mathbf{1}_{x_1 \in L}, \ldots, \mathbf{1}_{x_N \in L}) = 1$ (since S is a subset of the x_i's that are in L).

A delegation scheme for such a family is said to be succinct if the communication complexity is independent of N (most desirably $(m + \text{polylog}(n, N)) \cdot \text{poly}(\lambda)$) and the verifier computational complexity only depends on N to the extent that it is required to read the input and a description of the function f. In particular, if f has a succinct representation, e.g. it can be generated by a Turing machine, then the verification complexity can be lower. Indeed, our results are interesting for families F that consist of functions f that have a succinct description. We also require a *proof-of-knowledge* property, meaning that one can efficiently extract a valid witness $\{w_i\}_{i \in S}$ from any (possibly cheating) prover that convinces the verifier to accept with non-negligible probability.

As mentioned above, if the class F is the class of conjunctions, [BHK17] provide a delegation scheme with the aforementioned properties. We also notice that the work of Badrinarayanan et al. [BKK+17] implies such a delegation scheme for space-bounded non-deterministic computations.

Access Control Schemes. Consider a setting where there are N public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_N$ (for a very large N), and each user receives for some subset $S \subset [N]$ (corresponding to his credentials) a set of secret keys $\{\mathsf{sk}_i\}_{i \in S}$, where each sk_i corresponds to pk_i. Now suppose a user wishes to prove *anonymously* and *succinctly* that his credentials satisfy some monotone formula $f : \{0,1\}^N \to \{0,1\}$. Namely, he wishes to prove that his set S satisfies $f(\mathbf{1}_{1 \in S}, \ldots, \mathbf{1}_{n \in S}) = 1$. Combining our two main results (monotone **NP** delegation and our WI transformation) we obtain a single-round *succinct* and *anonymous* scheme, where a user can succinctly prove that his set of secret keys satisfies some monotone access structure (formulated as a monotone formula), where the anonymity property is WI and the length of a proof is $|\mathsf{sk}_i| \cdot \text{poly}(\log N, \lambda)$, where $|\mathsf{sk}_i|$ is the length of a *single* secret key, and λ is the security parameter. We call such a scheme a *succinct single-round access control scheme.*

Moreover, we can make our scheme *collusion resilient.* Namely, we can ensure that if two users have credentials corresponding to two sets $S_1, S_2 \subseteq [N]$, then together they cannot get credentials corresponding to $S_1 \cup S_2$, and moreover together they cannot prove more than what each user could have proven individually. This is done by introducing a signature scheme and setting each secret key to be a signature on the attribute concatenated with a random tag that is unique for the user. The random tags will prevent mixing an matching between different users' attributes. We refer to Section 3 for the formal definition and the construction.

We note that our notion of access control systems is similar to the notion of *anonymous credentials* [Cha85]. We identify two main differences between the two notions. One is that anonymous credentials require anonymity even against the issuer of the credentials, whereas in our model the issuer is a trusted party. The second is that anonymous credentials are not required to be succinct, in the sense that the proof could depend on the number of attributes, whereas succinctness is a cornerstone in the definition of access control systems. We believe that our techniques may be useful towards the construction of *succinct* anonymous credential schemes under standard assumptions by replacing the signature scheme from our construction in Section 3 with *blind signatures* [Cha82].

1.3 Technical Overview of Our WI Transformation

We show how to convert any single-round (2-message) argument system (and in particular, our single-round delegation protocol) with super-polynomial security into a witness indistinguishable one, with minimal (asymptotic) blowup to the communication complexity, albeit witness indistinguishability holds only against polynomial time distinguishers. We note that we can get super-polynomial security by properly strengthening the assumption, namely for any function $T(\lambda) \geq \lambda$ (where λ is the security parameter), if the original scheme was secure against any poly(T)-size adversary then we get witness indistinguishability against all $T^{o(1)}$-size adversaries. Furthermore, if the original protocol is extractable then the transformation would allow to apply the extractor as well.

The basic idea is for the verifier to simply send the first message of the protocol, and for the prover to compute its response according to the protocol, but rather than sending it to the verifier "in the clear", it will send a *statistically binding commitment* to the response. The idea is then for the prover to provide a WI proof (in parallel) that he indeed sent a commitment to an accepting response to the verifier's first message.

This idea runs into several obstacles, let us present the most severe ones. First, the original protocol may not be publicly verifiable (and indeed we would like to apply it to our aforementioned privately verifiable protocol), in which case the prover cannot prove that he is committing to a message that corresponds to an accepting response, since he does not know the verifier's verdict function. Second, we require that the prover commits to the accepting response using a statistically binding commitment, but this means that there is only one accepting witness and WI becomes meaningless. We next explain how to address these obstacles.

To address the first obstacle, we consider the secret state that the verifier keeps and is used to render the verdict of acceptance on the prover's response. In our new protocol, the verifier will send, along with its delegation query, its random tape in an encoded form. This encoded form should allow to apply the functionality of the prover under the encoding and send the encoded result back to the verifier, and at the same time hide the state so that soundness is maintained. To this end, we present an abstraction that we call *private remote evaluation scheme*, which can be thought of as a one-time non compact homomorphic encryption scheme with malicious circuit privacy. We show that this primitive can be constructed using garbled circuits and using an oblivious transfer protocol with security against malicious receivers (the same assumption is required for the WI proof system that we need to use). Given the verifier's random tape encoded in this way, the prover can "homomorphicly" check that indeed applying the verifier's query generation on the encoded random tape results in the query string sent by the verifier, and that the prover's response to this query string will result in the verifier accepting. The prover will perform this operation on the encoded random tape (note that the expected output should always be an encoding of 1) and prove in WI that the resulting encoding was indeed generated using the aforementioned operation. Since our encoding scheme is circuit-private, the verifier will not learn anything from the encoding itself (since it is just an encoding of 1), but the WI proof will guarantee that indeed the prover committed to a message that would have made the verifier of the original protocol accept.

The communication complexity of the generic remote evaluation scheme that we present is proportional to the running time of the verifier in the underlying argument system. This aspect could be improved by using a *succinct* remote evaluation scheme (i.e., a circuit private fully homomorphic encryption scheme), where the communication complexity does not grow with the running time. Such an evaluation scheme requires fully homomorphic encryption and can therefore is currently only known based on the learning with errors assumption (LWE), whereas our generic solution can be based on a variety of assumptions. We chose not to specify the succinct version in this work since we anyway inherit a communication blowup from the WI protocol that we use (see below), which in general can anyway grow with the running time of the verifier. Thus, we chose to avoid introducing a new assumption for this purpose.

Let us now specify the properties of the two message WI protocol that is required for this approach to go through. First of all, we notice that we need a protocol with *adaptive soundness*, i.e. soundness holds even against a prover that chooses the statement to be proven after seeing the verifier's first message. We emphasize that even though we use as a building block a WI protocol with adaptive soundness, our resulting (succinct) WI protocol is not adaptively sound (i.e., soundness holds only against provers that choose the statement to be proven before seeing the verifier's message).

Second, we need to address the aforementioned vacuousness of the standard notion of WI when proving with respect to a committed value. This is resolved by resorting to the notion of *strong WI*, which considers two distributions over instance-witness pairs, and requires that if the instance components of the two distributions are computationally indistinguishable, then the verifier cannot distinguish which instance-witness pair was used to generate the proof. Indeed, the recently proposed protocol of Jain et al. [JKKR17] has the required properties (in the delayed input setting), under the assumption that a quasi-poly secure OT scheme exists (we refer to Section 2 for details, and in particular to Theorem 2.1).

Lastly, we require extractability, namely being able to extract the committed response to the delegation protocol in case the WI protocol accepted. However, since the prover only sends a single message, we cannot get extractability under standard assumptions. We therefore rely on complexity leveraging, and extract the prover answer by brute-force breaking the hiding of the commitment scheme. This means that in order for soundness to hold, we need all components other than the commitment scheme to be secure even in the presence of this brute-force extractor, i.e. to have super-polynomial security. This way, we can scale down the hardness of the commitment scheme and allow it to be broken while leaving the other building blocks secure.

2 Witness Indistinguishability for any Argument System

In this section we present our general transformation for converting any 2-message argument system into a 2-message witness indistinguishable one with only modest increase in communication complexity.

2.1 Preliminaries

Our transformation makes use of several cryptographic building blocks, which we present below.

Garbled Circuits. We rely on a decomposable randomized encoding scheme. For the sake of concreteness we consider garbled circuits.

Definition 2.1 (Garbled Circuits)**.** *A garbling scheme consists of a tuple of three algorithms* $(\mathsf{Garble}, \mathsf{GCEval}, \mathsf{GCSim})$ *where:*

1. $\mathsf{Garble}(1^\lambda, C)$ *is a PPT algorithm that takes as input the security parameter* λ *(ommitted when clear from the context) and a circuit* $C : \{0,1\}^n \to \{0,1\}^m$*, and outputs a garbled circuit* $\widehat{C}$ *along with input labels* $(\mathsf{lab}_{i,b})_{i\in[n],b\in\{0,1\}}$ *where each label* $\mathsf{lab}_{i,b} \in \{0,1\}^\lambda$.

2. $\mathsf{GCEval}(1^\lambda, \widehat{C}, \widehat{\mathsf{lab}})$ *is a deterministic algorithm that takes as input a garbled circuit* $\widehat{C}$ *along with a set of* n *labels* $\widehat{\mathsf{lab}} = (\mathsf{lab}_i)_{i\in[n]}$*, and outputs a string* $y \in \{0,1\}^m$.

3. $\mathsf{GCSim}(1^\lambda, 1^{|C|}, 1^n, y)$ *is a* PPT *algorithm that takes as input the security parameter, the description length of* C*, an input length* n *and a string* $y \in \{0,1\}^m$*, and outputs a simulated garbled circuit* $\widetilde{C}$ *and labels* $\widetilde{\mathsf{lab}}$*.*

We often omit the first input to these algorithms (namely, 1^λ*) when it is clear from the context. We require that the garbling scheme satisfies two properties:*

1. *Correctness: For all circuits* C*, inputs* x*, and all* $(\widehat{C}, (\mathsf{lab}_{i,b})_{i,b}) \leftarrow \mathsf{Garble}(C)$ *and* $\widehat{\mathsf{lab}} = (\mathsf{lab}_{i,x_i})_{i\in[n]}$*, we have that* $\mathsf{GCEval}(\widehat{C}, \widehat{\mathsf{lab}}) = C(x)$*.*

2. *Simulation Security: For all circuits* $C : \{0,1\}^n \to \{0,1\}^m$ *and all inputs* $x \in \{0,1\}^n$*, the following two distributions are computationally indistinguishable:*

$$\left\{(\widehat{C}, \widehat{\mathsf{lab}}) : (\widehat{C}, (\mathsf{lab}_{i,b})_{i,b}) \leftarrow \mathsf{Garble}(C), \widehat{\mathsf{lab}} = (\mathsf{lab}_{i,x_i})_{i\in[n]}\right\}$$
$$\overset{c}{\approx} \left\{(\widetilde{C}, \widetilde{\mathsf{lab}}) : (\widetilde{C}, \widetilde{\mathsf{lab}}) \leftarrow \mathsf{GCSim}(1^\lambda, 1^{|C|}, 1^n, C(x))\right\} .$$

Oblivious Transfer Secure Against Malicious Receivers. We use a notion of oblivious transfer that has computational security against senders (i.e. receiver privacy) but also (statistical) security against malicious receivers (sender privacy). That is, regardless of the receiver's first message, the sender's response never reveals more than one of its inputs, even to an unbounded adversary.

Definition 2.2 (Two-Message Oblivious Transfer with Statistical Sender Security)**.** *A two-message oblivious transfer is a protocol between two parties, a sender* S *with messages* (m_0, m_1) *and receiver* $R = (R_1, R_2)$ *with a choice bit* b*, such that* R *obtains output* m_b *at the end of the protocol. Specifically,* $R_1(b) = R_1(1^\lambda, b)$ *outputs* (σ, e)*, where* e *is the message sent to the receiver and* σ *is a local state that is kept private. The sender responds with an answer* $v = S(1^\lambda, (m_0, m_1), e)$*. Finally* $R_2(1^\lambda, \sigma, v)$ *outputs a message* m*. We omit the security parameter input to these procedures when it is clear from the context.*

We consider OT that satisfies the following properties:

- **Computational Receiver Security.** *The distributions* $R_1(0)$ *and* $R_1(1)$ *are computationally indistinguishable. We sometimes require super-polynomial security, specifically, we say that the OT scheme is* T*-receiver secure if* $T \cdot \mathrm{poly}(\lambda)$*-size distinguishers have advantage less than* $\frac{\mathrm{negl}(\lambda)}{T}$*.*

- **Statistical Sender Security.** *For all* λ *and for all* $e^* \in \{0,1\}^*$ *there exists a bit* b^* *such that the distributions* $S(1^\lambda, (m_0, m_1), e^*)$ *and* $S(1^\lambda, (m_{b^*}, m_{b^*}), e^*)$ *are statistically indistinguishable. It would sometimes be convenient to think about* b^* *as produced by a computationally unbounded procedure* Ext *so that* $b^* = \mathsf{Ext}(1^\lambda, e^*)$ *(we sometimes omit* 1^λ *when it is clear from the context).*

Oblivious transfer protocols satisfying these definitions have been introduced based on assumptions such as DDH, QR, DCR and LWE [NP01, Kal05, HK07, BD18].

Delayed-Input Interactive Protocols and Strong Witness Indistinguishability. A ℓ-message delayed-input interactive protocol (P, V) for deciding an **NP** language L with associated relation R_L proceeds in the following manner:

- At the beginning of the protocol, P and V receive the size of the instance and the security parameter, denoted by n and λ, respectively, and execute the first $\ell - 1$ messages.
- Before sending the last message, P receives as input a pair $(x, w) \in R_L$, where $|x| = n$, and V receives x. Upon receiving the last message from P, V outputs 1 or 0.

An execution of (P, V) with instance x and witness w is denoted as $\langle P, V\rangle(x, w)$. Whenever clear from context, we also use the same notation to denote the output of V.

A ℓ-message delayed-input interactive argument for a language L must satisfy the standard notion of completeness (in the delayed-input setting) as well as *adaptive soundness*, where the soundness requirement holds even against malicious PPT provers who choose the statement adaptively, depending upon the first $\ell - 1$ messages of the protocol.

Definition 2.3 (Delayed-Input Interactive Arguments)**.** *A ℓ-message delayed-input interactive protocol (P, V) for deciding a language L is an interactive argument for L if it satisfies the following properties:*

- **Adaptive Completeness:** *For every $(x, w) \in R_L$ chosen adaptively after $\ell - 1$ rounds of interaction,*
$$\Pr\left[\langle P, V\rangle(x, w) = 1\right] = 1,$$
where the probability is over the random coins of P and V.
- **Adaptive Soundness:** *For every (non-uniform)* PPT *prover P^* that chooses $n = \mathrm{poly}(\lambda)$ and chooses $x \in \{0,1\}^n \setminus L$ adaptively, depending upon the first $\ell - 1$ messages,*
$$\Pr\left[\langle P^*, V\rangle(x) = 1\right] = \mathrm{negl}(\lambda),$$
where the probability is over the random coins of V.

Definition 2.4 ((Strong) Witness Indistinguishability)**.** *Let $n = n(\lambda) \leq \mathrm{poly}(\lambda)$. An interactive argument (P, V) for a language L is* strong witness indistinguishable *(which we denote* sWI*) if for every pair of distributions over pairs $\{(\mathcal{X}_{1,n(\lambda)}, \mathcal{W}_{1,n(\lambda)})\}_{\lambda\in\mathbb{N}}$ and $\{(\mathcal{X}_{2,n(\lambda)}, \mathcal{W}_{2,n(\lambda)})\}_{\lambda\in\mathbb{N}}$ supported over R_L, for which the distributions $\{\mathcal{X}_{1,n(\lambda)}\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{X}_{2,n(\lambda)}\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable, for every* PPT *verifier V^*, and for every (non-uniform)* PPT *distinguisher $\mathcal{D}$,*

$$\Bigg| \Pr_{(x,w)\leftarrow(\mathcal{X}_{1,n(\lambda)},\mathcal{W}_{1,n(\lambda)})}\left[\mathcal{D}(x, \mathrm{View}_{V^*}[\langle P, V^*\rangle(x, w)] = 1\right]$$
$$- \Pr_{(x,w)\leftarrow(\mathcal{X}_{2,n(\lambda)},\mathcal{W}_{2,n(\lambda)})}\left[\mathcal{D}(x, \mathrm{View}_{V^*}[\langle P, V^*\rangle(x, w)] = 1\right]\Bigg| \leq \mathrm{negl}(\lambda)\ .$$

Standard (as opposed to strong) witness indistinguishability (which we denote simply by WI*) only requires that the above holds for singleton distributions, which is equivalent (due to the indistinguishability condition) to defining a deterministic sequence of input and witness pairs*

$$\{(x_{n(\lambda)}, w_{1,n(\lambda)}, w_{2,n(\lambda)})\}_{\lambda\in\mathbb{N}}.$$

In delayed input *strong witness indistinguishability, the above is only required to hold with respect to* PPT *verifiers V^* who obtain the instance together with the last prover message in the protocol (i.e., who generate their messages obliviously of x). Note that this notion is vacuous for standard (non-strong)* WI.

Theorem 2.1 ([JKKR17]). *For any $T = \lambda^{\omega(1)}$, assume the existence of a non-interactive statistically binding commitment scheme, that is hiding against poly-size adversaries, but where the hiding property can be broken by* $\mathrm{poly}(T)$ *adversaries, and assume the existence of a* $\mathrm{poly}(T)$*-secure OT scheme as in Definition 2.2. Then there exists a 2-message delayed-input strong WI protocol for every language in* **NP** *such that soundness holds against* $\mathrm{poly}(T)$*-size adversaries, but (strong) WI property holds only against poly-size cheating verifiers.*

Remark 2.1. *The strong WI property can be strengthened to hold against* $\mathrm{poly}(T^*)$*-size cheating verifiers, for any $T^* = T^{o(1)}$. However, this requires assuming that the underlying commitment scheme that can be broken in time* $\mathrm{poly}(T)$*, is secure against* $\mathrm{poly}(T^*)$ *size adversaries.*

2.2 Private Remote Evaluation

Our transformation makes use of a primitive that we call a *private remote evaluation scheme*. Loosely speaking, this can be thought of as a *one-time* non-succinct fully homomoprhic encryption scheme with strong malicious circuit privacy [GHV10, OPP14].

Rather than formally defining this primitive, we construct it following the outline of Yao's 2-party 2-round secure function evaluation protocol [Yao82] (using a garbling scheme satisfying Definition 2.1 and using an oblivious transfer protocol satisfying Definition 2.2), and state its properties.

Let $(R = (R_1, R_2), S)$ be an OT scheme that satisfies Definition 2.2 and let $(\mathsf{Garble}, \mathsf{GCEval}, \mathsf{GCSim})$ be a garbling scheme. Our private remote evaluation scheme consists of a tuple of four algorithms $(\mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Sim})$, defined as follows.

- The encoding algorithm Enc takes an input a security parameter 1^λ and a string $x \in \{0,1\}^n$, and outputs an encoded output ψ and a secret state σ. Specifically, for every bit of x, Enc runs $R_1(1^\lambda, x_i)$ to compute the first OT receiver message $\psi^{(i)}$ and the state $\sigma^{(i)}$. It outputs $\psi = \{\psi^{(i)}\}_{i\in[n]}$, $\sigma = \{\sigma^{(i)}\}_{i\in[n]}$. We sometimes denote by Enc_1 the algorithm that computes Enc and only outputs the ψ component, and we often omit the security parameter from the notation.

- The evaluation algorithm Eval takes as input a circuit $C : \{0,1\}^n \to \{0,1\}^m$ and an encoded input $\psi = \{\psi^{(i)}\}_{i\in[n]}$. It runs $\mathsf{Garble} C$ to generate a garbled circuit $\widehat{C}$ for C with labels $\mathsf{lab}_{i,b}$, and computes the sender response for each OT execution $\psi'^{(i)} = S((\mathsf{lab}_{i,0}, \mathsf{lab}_{i,1}), \psi^{(i)})$. It finally outputs $\psi' = (\{\psi'^{(i)}\}_{i\in[n]}, \widehat{C})$.

- The decoding procedure Dec takes as input $\psi' = (\{\psi'^{(i)}\}_{i\in[n]}, \widehat{C})$ and $\sigma = \{\sigma^{(i)}\}_{i\in[n]}$, and applies the OT receiver protocol to obtain $\mathsf{lab}_i = R_2(\sigma^{(i)}, \psi'^{(i)})$. It finally runs $\mathsf{GCEval}(\widehat{C}, \{\mathsf{lab}_i\}_{i\in[n]})$ and outputs the resulting $y \in \{0,1\}^m$.

- For all $1^n, 1^m, 1^c$ representing input, output and circuit size (these inputs are often omitted when they are clear from the context), there exists a simulator

$$\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2) \ ,$$

  such that the following holds. Let Ext be the OT extractor from Definition 2.2. The simulator Sim_1 takes as input a (possibly adversarially chosen) sequence $\psi = \{\psi^{(i)}\}_{i\in[n]}$, and runs Ext on each $\psi^{(i)}$ to obtain a bit x_i. Let $x \in \{0,1\}^n$ denote the collection of the extracted bits.

The simulator Sim_2, takes as input (ψ, x) together with a string $y \in \{0,1\}^m$, it runs in probabilistic polynomial time, and does the following:

1. It runs the PPT garbled circuit simulator GCSim, on input y (and input $1^\lambda, 1^{|C|}, 1^n$), to generate simulated circuit $\widetilde{C}$ and labels $\widetilde{\mathsf{lab}}$.
2. It generates simulated sender messages $\{\widetilde{\psi}^{(i)}\} \leftarrow S((\widetilde{\mathsf{lab}}_i, \widetilde{\mathsf{lab}}_i), x_i)$.
3. It outputs $\widetilde{\psi} = (\{\widetilde{\psi}^{(i)}\}, \widetilde{C})$.

Claim 2.2. *For any $\psi = \{\psi^{(i)}\}_{i\in[n]}$ and any circuit $C : \{0,1\}^n \to \{0,1\}^m$, it holds that*

$$\mathsf{Eval}(C, \psi) \overset{c}{\approx} \mathsf{Sim}_2(\psi, x, C(x)) \ ,$$

where $x \leftarrow \mathsf{Sim}_1(\psi)$.

Proof. By definition,

$$\mathsf{Eval}(C, \psi) = \left(\left\{ S((\mathsf{lab}_{i,0}, \mathsf{lab}_{i,1}), \psi^{(i)}) \right\}_{i\in[n]}, \widehat{C} \right) \ .$$

Since ψ is fixed, then the value $x \leftarrow \mathsf{Sim}_1(\psi)$ is also fixed. It follows from Definition 2.2 that

$$\mathsf{Eval}(C, \psi) \overset{s}{\approx} \left(\left\{ S((\mathsf{lab}_{i,x_i}, \mathsf{lab}_{i,x_i}), \psi^{(i)}) \right\}_{i\in[n]}, \widehat{C} \right) \ .$$

Now we use the garbled circuit security to argue that

$$\mathsf{Eval}(C, \psi) \overset{c}{\approx} \left(\left\{ S((\widetilde{\mathsf{lab}}_i, \widetilde{\mathsf{lab}}_i), \psi^{(i)}) \right\}_{i\in[n]}, \widetilde{C} \right) = \mathsf{Sim}_2(\psi, x, C(x)) \ ,$$

where $\widetilde{\mathsf{lab}}, \widetilde{C}$ are produced by the garbled circuit simulator given $y = C(x)$. □

The following claims are immediate from the OT correctness and receiver security.

Claim 2.3 (Correctness)**.** *For every $n = n(\lambda)$ (not necessarily polynomially bounded), every $x \in \{0,1\}^n$, every $C : \{0,1\}^n \to \{0,1\}$, letting $(\psi, \sigma) \leftarrow \mathsf{Enc}(1^\lambda, x)$, $\psi' \leftarrow \mathsf{Eval}(C, \psi)$, $y = \mathsf{Dec}(\sigma, \psi')$, it holds that $y = C(x)$ with probability 1.*

Claim 2.4 (Receiver Privacy.)**.** *For every $n = n(\lambda) \leq \mathrm{poly}(\lambda)$ and every sequences of inputs $x, x' \in \{0,1\}^n$ it holds that $\mathsf{Enc}_1(1^\lambda, x) \overset{c}{\approx} \mathsf{Enc}_1(1^\lambda, x')$.*

2.3 Making Single-Round Protocols Witness Indistinguishable

We show how to convert any single-round (2-message) protocol (P, V) with super-polynomial security and perfect completeness into a single-round (2-message) *witness indistinguishable* (WI) protocol, such that if the communication complexity of the original protocol (P, V) is $\mathrm{cc}(n, \lambda)$ then the communication complexity of the resulting WI protocol $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ is $\mathrm{cc}(n, \lambda) + \mathrm{poly}(v(n, \lambda))$, where $v(n, \lambda)$ is the total runtime of the original verifier V, both in generating the query string to be sent to the prover and in verifying the response received by the prover. We use the term *verdict function* to refer to the second step on V, namely the function that takes as input the communication transcript and an internal secret state of the verifier, and outputs whether the verifier accepts or rejects. Our transformation requires that the original protocol (P, V) is sound against super-polynomial time adversaries (as we intend to use complexity leveraging). Our theorem statement follows.

Theorem 2.5. *For any super-polynomial function $T : \mathbb{N} \to \mathbb{N}$, there is a generic transformation that transforms any (privately or publicly verifiable) single-round argument (P, V) for an* **NP** *language L with perfect completeness and with soundness against $\mathrm{poly}(T)$-size cheating provers, into a* privately verifiable *witness indistinguishable single-round argument $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ for L with the following properties:*

- **Succinctness.** *If the communication complexity of (P, V) is $\mathrm{cc}(n, \lambda)$, and V has total time complexity $v(n, \lambda)$, then the communication complexity of $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ is*

$$\mathrm{cc}_{\mathrm{WI}}(n, \lambda) \triangleq \mathrm{cc}(n, \lambda) + \mathrm{poly}(\lambda, v(n, \lambda)).$$[1]

- **Completeness.** *For every $x \in L$ and any witness w for x, it holds that $(P_{\mathrm{WI}}(x.w), V_{\mathrm{WI}}(x))$ accepts with probability 1.*
- **Soundness.** *$(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ is sound against (non-uniform) cheating provers of size $\mathrm{poly}(T)$.*[2]
- **Witness Indistinguishability.** *$(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ is witness indistinguishable against (non-uniform)* PPT *cheating verifiers (but not against $\mathrm{poly}(T)$-size cheating verifiers, see also Remark 2.3 below).*

This transformation requires the following building blocks:

- *A statistically binding non-interactive commitment scheme* Com *that can be broken in time* $\mathrm{poly}(T)$ *for all sufficiently large value of λ.*
- *The private remote evaluation scheme* (Enc, Dec, Eval, Sim)*, as described in Section 2.2, where the underlying OT scheme has receiver privacy against* $\mathrm{poly}(T)$*-size adversaries (i.e, Claim 2.4 is satisfied against* $\mathrm{poly}(T)$*-size adversaries).*
- *A delayed-input single-round (2-message) strong WI (*sWI*) argument system $(P_{\mathrm{sWI}}, V_{\mathrm{sWI}})$ for* **NP***, that is sound against* $\mathrm{poly}(T)$ *size cheating provers.*

In fact, we will show that our transformation enjoys an even stronger soundness guarantee as described next. There exist black-box non-rewinding, instance preserving (where applicable) $\mathrm{poly}(T)$*-time reductions M_1, M_2, M_3, such that for every (possibly inefficient) cheating prover P^*_{WI} it holds that $M_1^{P^*_{\mathrm{WI}}}$ is a cheating prover against the sWI proof system, $M_2^{P^*_{\mathrm{WI}}}$ is a distinguisher for the remote evaluation scheme, and $M_3^{P^*_{\mathrm{WI}}}$ is a cheating prover against the original argument system, and it holds that the sum of advantages of these adversaries in their related game is at least the advantage of P^*_{WI} in the compiled protocol (up to negligible terms).*

We note that the resulting WI protocol is only privately verifiable, even if the underlying protocol was publicly verifiable.

[1] This guarantee is of interest only if $v(n, \lambda)$ is significantly smaller than the witness size, which is the case for example the argument systems constructed in [BHK17, BKK$^+$17].

[2] We emphasize that the soundness property (both for the underlying argument (P, V) and the resulting one $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$) is non-adaptive soundness, where soundness is required to hold only against cheating provers that choose the statement to be proven before seeing the verifier's message.

Remark 2.2. *We note that if we rely on a* succinct *remote evaluation scheme (e.g., a circuit-private fully homomorphic encryption scheme), then the communication complexity would be* $\mathrm{poly}(\lambda) \cdot \mathrm{cc}(n, \lambda) + \mathrm{cc}(\mathrm{sWI})$*, where* $\mathrm{cc}(\mathrm{sWI})$ *is the communication complexity of the underlying strong WI protocol, which in general can be as large as* $v(n, \lambda)$*, but can be smaller if the underlying strong WI protocol is succinct.*

Remark 2.3. *One can strengthen the above theorem so that WI holds against any* $\mathrm{poly}(T^*)$*-size adversaries, for any* $T^* = T^{o(1)}$*, by relying on a quantified version of Theorem 2.1 (see Remark 2.1), with WI against* $\mathrm{poly}(T^*)$*-size adversaries. This requires assuming that the underlying commitment scheme* Com*, which can be broken in time* $\mathrm{poly}(T)$*, is secure against* $\mathrm{poly}(T^*)$*-adversaries.*

Proof. Consider a language L, time complexity bound T, a protocol (P, V), a private remote evaluation scheme $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}, \mathsf{Sim})$ and a delayed input strong WI argument system $(P_{\mathrm{sWI}}, V_{\mathrm{sWI}})$, all as described in the theorem statement. We denote by (Q, A) the first and second message respectively exchanged in the protocol (P, V).

Let Com be a statistically binding non-interactive commitment scheme that can be broken in time $\mathrm{poly}(T)$, as described in the theorem statement. Such commitment schemes can be constructed from injective one-way functions. We note that for our purposes it is possible to use Naor's two-message commitment scheme from any one-way function [Nao89] since we can allow a message from the receiver to the sender prior to the commitment message, but for the sake of simplicity we will assume that Com is non-interactive. We further assume w.l.o.g that the length of the commitment string is equal to the length of the committed message plus an additive $\mathrm{poly}(\lambda)$ term. This can be achieved generically using "key encapsulation" (committing to a PRG seed and using the PRG output to mask the message).

We show how to convert (P, V) into a 2-message witness indistinguishable argument, denoted by $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$, which preserves the succinctness property of (P, V), as stated in the theorem statement. Since (P, V) is not necessarily publicly verifiable, in order to verify a transcript (Q, A) the verifier may need a private state, which we denote by st. We will assume w.l.o.g that st is simply the random tape of the verifier V. This will allow to check, given some possible query string Q whether Q is the string generated when V starts with random tape st. If this condition holds, we say that st is consistent with Q, we denote this by $\mathsf{st} \models Q$. The resulting protocol $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ makes use of an underlying (not necessarily succinct) delayed-input strong WI 2-message argument $(P_{\mathrm{sWI}}, V_{\mathrm{sWI}})$ for the **NP** language L', defined as follows:

$$\begin{aligned} L' = &\{(1^\lambda, x, Q, c, \mathsf{st}) : \exists (A, r) \text{ s.t.} \\ &\big(\mathsf{st} \not\models Q\big) \ \vee\ \big(c = \mathrm{Com}(A, r) \wedge V(1^\lambda, x, Q, A, \mathsf{st}) = 1\big)\} \ . \end{aligned} \tag{1}$$

Note that every instance where Q is *inconsistent* with st is trivially *in the language.* Intuitively, this is to force witness indistinguishability also against verifiers who produce inconsistent transcripts. This condition will never be relevant for honest verifiers.

In the protocol $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$, the prover will send a commitment to his answer A (as opposed to sending it in the clear, which may reveal information), followed by a proof that the committed value is an accepting answer. However, to generate such a proof he needs to know the verdict function, and thus, needs the verifier's secret state. However, he cannot receive this secret state "in the clear", since that may breech soundness. Instead, the verifier will send the prover an encoding of his secret state st using the private remote evaluation scheme.

We are now ready to define the protocol $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$:

1. On input 1^λ and $x \in \{0,1\}^n$ the verifier does the following:
   (a) Compute $(Q, \mathsf{st}) \leftarrow V(1^\lambda, x)$, where Q is the message to be sent to the prover P, and st is the corresponding secret state of V.
   (b) Compute $(\psi, \sigma) \leftarrow \mathsf{Enc}(\mathsf{st})$.
   (c) Compute $(\mathrm{sWI}_1, \mathsf{st}_{\mathrm{sWI}}) \leftarrow V_{\mathrm{sWI}}(1^\lambda)$.
   Note that the first message sWI_1 is independent of the instance since $(P_{\mathrm{sWI}}, V_{\mathrm{sWI}})$ is a delayed-input 2-message argument (see Definition 2.3).

   Send $(Q, \mathrm{sWI}_1, \psi)$ to the prover, and store $(\sigma, \mathsf{st}, \mathsf{st}_{\mathrm{sWI}})$ as the secret state for verification.

2. The prover, on input $(1^\lambda, x, w)$, and given the message $(Q, \mathrm{sWI}_1, \psi)$, does the following:
   (a) Compute $A \leftarrow P(1^\lambda, x, w, Q)$
   (b) Choose a random string $r \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$ and compute $c = \mathrm{Com}(A, r)$.
   (c) Define (implicitly since st is not known) $x' = (1^\lambda, x, Q, c, \mathsf{st})$, and $w' = (A, r)$ as its corresponding witness with respect to $R_{L'}$, i.e. $(x', w') \in \mathcal{R}_{L'}$.
   (d) Given ψ, compute $\psi' = \mathsf{Eval}(f, \psi)$ where $f = f_{1^\lambda, x, Q, c, w', \mathrm{sWI}_1}$, is the function that on input st outputs $\mathrm{sWI}_2 \leftarrow P_{\mathrm{sWI}}(1^\lambda, x', w', \mathrm{sWI}_1)$.

   Send (c, ψ') to the verifier.

3. Upon receiving a message (c, ψ') from the prover, and given a secret state $(\sigma, \mathsf{st}, \mathsf{st}_{\mathrm{sWI}})$ the verifier does the following:
   (a) Decrypt the ciphertext ψ', by computing $\mathrm{sWI}_2 \leftarrow \mathsf{Dec}(\sigma, \psi')$.
   (b) Accept if and only if $V_{\mathrm{sWI}}(1^\lambda, x', \mathrm{sWI}_1, \mathrm{sWI}_2, \mathsf{st}_{\mathrm{sWI}}) = 1$, where $x' = (1^\lambda, x, Q, c, \mathsf{st})$.

Succinctness. We first argue that $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ satisfies the succinctness property as in the theorem statement. To do this, we argue that

$$\mathrm{cc}(P_{\mathrm{WI}}, V_{\mathrm{WI}}) = \mathrm{cc}(P, V) + \mathrm{poly}(\lambda) + \mathrm{poly}(\lambda, v(n, \lambda)) ,$$

which would immediately imply the required succinctness.

The first additive $\mathrm{poly}(\lambda)$ term is due to the overhead of sending a commitment to the answer A rather than sending A itself (as explained above, we can assume additive overhead w.l.o.g). The second $\mathrm{poly}(\lambda, v(n, \lambda))$ term is an upper bound on the length of ψ'. The value ψ' is the output of applying Eval on a function f of size $v(n, \lambda) + \mathrm{poly}(\lambda) \leq \mathrm{poly}(\lambda, v(n, \lambda))$ (an upper bound on the prover complexity of P_{sWI} when proving $(x', w') \in \mathcal{R}_{L'}$). Verifying that $(x', w') \in \mathcal{R}_{L'}$ can be done in time proportional to the total complexity of V since checking whether $\mathsf{st} \models Q$ is proportional to running the first phase of V, and checking the value of the verdict function is proportional to the second phase. Add to that checking the commitment which is polynomial in $(\lambda, |A|)$. Since Eval introduces a fixed polynomial overhead, its output length is at most $\mathrm{poly}(\lambda, v(n, \lambda))$.

It remains to prove that $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ satisfies the standard completeness and soundness guarantees, and in addition that it is witness indistinguishable.

Completeness. The completeness of $(P_{\rm WI}, V_{\rm WI})$ follows immediately from the completeness of (P, V), the delayed-input completeness of $(P_{\rm sWI}, V_{\rm sWI})$, and the correctness of the underlying private remote evaluation scheme.

Soundness. Consider a cheating prover $P^*_{\rm WI}$ that for any security parameter 1^λ generates $x \in \{0,1\}^n \setminus L$, where $n \leq \text{poly}(\lambda)$, such that for some non-negligible function $\alpha = \alpha(\lambda)$

$$\Pr[\mathsf{Output}_{V_{\rm WI}}(P^*_{\rm WI}, V_{\rm WI})(1^\lambda, x) = 1] \geq \alpha \ . \tag{2}$$

Recall that $P^*_{\rm WI}$, upon receiving a message (Q, sWI_1, ψ), where $\psi \leftarrow \mathsf{Enc}_1(\mathsf{st})$, from the verifier, generates a response (c, ψ'). Since Com is a statistically binding commitment scheme that can be broken in $\text{poly}(T)$ time, there exists a $\text{poly}(T)$-time algorithm that given c outputs (A', r') such that $c = \text{Com}(A', r')$.

Define

$$\alpha_1 = \alpha_1(\lambda) \triangleq \Pr[\big(\mathsf{Output}_{V_{\rm WI}}(P^*_{\rm WI}, V_{\rm WI})(1^\lambda, x) = 1\big) \wedge \big(V(1^\lambda, x, Q, A', \mathsf{st}) = 0\big)]. \tag{3}$$

We consider a cheating prover $P^*_{\rm sWI} = M_1^{P^*_{\rm WI}}$ (where M_1 is a non-rewinding reduction) that succeeds in breaking the delayed input soundness of $(P_{\rm sWI}, V_{\rm sWI})$ with probability α_1. The reduction M_1, takes as input a message sWI_1 from the verifier $V_{\rm sWI}$, and does the following:

1. Generate $x \leftarrow P^*_{\rm WI}(1^\lambda)$ using the $P^*_{\rm WI}$ oracle.
2. Compute $(Q, \mathsf{st}) \leftarrow V(1^\lambda, x)$.
3. Compute $(\psi, \sigma) \leftarrow \mathsf{Enc}(\mathsf{st})$.
4. Send (Q, sWI_1, ψ) to the $P^*_{\rm WI}$ oracle to obtain $(c, \psi') = P^*_{\rm WI}(Q, \text{sWI}_1, \psi)$. Recall that for an honest $P_{\rm WI}$, it holds that ψ' decrypts to sWI_2.
5. Let $x' = (1^\lambda, x, Q, c, \mathsf{st})$.
6. Compute $\text{sWI}_2 \leftarrow \mathsf{Dec}(\sigma, \psi')$.
7. Send (x', sWI_2) to the verifier.

Note that it suffices to argue that

$$\Pr[V_{\rm sWI}(1^\lambda, \text{sWI}_1, (x', \text{sWI}_2), \mathsf{st}_{\rm sWI}) = 1 \wedge (x' \notin L')] \geq \alpha_1.$$

This follows immediately from Eq. (3), together with the fact that $x' \notin L'$ if and only if $V(1^\lambda, x, Q, A', \mathsf{st}) = 0$, where A' is the value that c commits to, and the fact that

$$V_{\rm WI}(1^\lambda, x, (Q, \text{sWI}_1, \psi), (c, \psi'), \mathsf{st}) = 1$$

only if

$$V_{\rm sWI}(1^\lambda, \text{sWI}_1, (x', \text{sWI}_2), \mathsf{st}_{\rm sWI}) = 1 \ .$$

Note that by Eq. (2) and (3),

$$\Pr[\big(\mathsf{Output}_{V_{\rm WI}}(P^*_{\rm WI}, V_{\rm WI})(1^\lambda, x) = 1\big) \wedge \big(V(1^\lambda, x, Q, A', \mathsf{st}) = 1\big)] \geq \alpha - \alpha_1. \tag{4}$$

We now present poly(T)-time straight line reductions M_2, M_3 converting P^*_{WI} into an adversary $\mathcal{A}$ that breaks the indistinguishability property of the encoding scheme (i.e., breaks Claim 2.4 with respect to a poly(T)-size adversary), and into cheating prover P^* for the underlying 2-message argument (P, V), respectively, so that the sum of the advantages of the resulting adversaries, denoted α_2, α_3 respectively is $\alpha_2 + \alpha_3 \geq \alpha - \alpha_1$. Furthermore, M_3 is also input preserving.

The distinguisher $\mathcal{A} = M_2^{P^*_{\mathrm{WI}}}$ runs as follows.

1. Generate $x \leftarrow P^*_{\mathrm{WI}}(1^\lambda)$.
2. Run the verifier $V(1^\lambda, x)$ to generate (Q, st).
3. Send $(\mathsf{st}, 0^{|\mathsf{st}|})$ as the two messages for the distinguishing advantage, and receive a challenge encoding ψ from the encoding scheme challenger.
4. Generate $(\mathrm{sWI}_1, \mathsf{st}_{\mathrm{sWI}}) \leftarrow V_{\mathrm{sWI}}(1^\lambda)$.
5. Send $(Q, \mathrm{sWI}_1, \psi)$ to the P^*_{WI} oracle to obtain $(c, \psi') = P^*_{\mathrm{WI}}(Q, \mathrm{sWI}_1, \psi)$.
6. Run in time poly(T) to find (A', r') such that $c = \mathrm{Com}(A', r')$.
7. Return $V(1^\lambda, x, Q, A', \mathsf{st})$.

The cheating prover $P^* = M_3^{P^*_{\mathrm{WI}}}$ is as follows.

1. Upon receiving a security parameter 1^λ, generate $x \leftarrow P^*_{\mathrm{WI}}(1^\lambda)$.
2. Upon receiving a message Q from the verifier $V(1^\lambda, x)$, do the following:
   (a) compute $(\mathrm{sWI}_1, \mathsf{st}_{\mathrm{sWI}}) \leftarrow V_{\mathrm{sWI}}(1^\lambda)$.
   (b) Generate $\psi \leftarrow \mathsf{Enc}_1(0^{|\mathsf{st}|})$ (while st itself is unknown, its length is specified by the protocol, we recall that Enc_1 is the algorithm that executes Enc but only outputs the ψ component, see Section 2.2).
   (c) Send $(Q, \mathrm{sWI}_1, \psi)$ to the P^*_{WI} oracle to obtain $(c, \psi') = P^*_{\mathrm{WI}}(Q, \mathrm{sWI}_1, \psi)$.
3. Run in time poly(T) to find (A', r') such that $c = \mathrm{Com}(A', r')$.
4. Send A' to the verifier.

Note that σ is not used at all by our P^* (and of course also not by V which is the distinguisher for the original protocol). Consider an experiment with a prover $\tilde{P}^*$ which is identical to P^* except it uses $\psi = \mathsf{Enc}_1(\mathsf{st})$, where st is the actual secret state corresponding to Q. Then by Eq. (4),

$$\Pr[(\tilde{P}^*, V)(x) = 1] \geq \alpha - \alpha_1 \ .$$

However, by definition of $\tilde{P}^*$, it is identical to P^* except for the use of ψ that encodes st instead of $0^{|\mathsf{st}|}$. If the two behave differently this translates to advantage for the distinguisher $\mathcal{A}$. In other words, the success probability of $\mathcal{A}$ is exactly

$$\alpha_2 = \Pr[(P^*, V)(x) = 1] - \Pr[(\tilde{P}^*, V)(x) = 1] \ .$$

We conclude that $\alpha_1 + \alpha_2 + \alpha_3 \geq \alpha$ as required.

Witness Indistinguishability. It remains to argue that $(P_{\mathrm{WI}}, V_{\mathrm{WI}})$ satisfies the WI criterion. Fix a function $n = n(\lambda) \leq \mathrm{poly}(\lambda)$, and fix any ensemble $\{(x_n, w_{1,n}, w_{2,n})\}_{\lambda \in \mathbb{N}}$, such that $(x_n, w_{1,n}) \in \mathcal{R}_L$ and $(x_n, w_{2,n}) \in \mathcal{R}_L$. Suppose for the sake of contradiction that there exists a (non-uniform) poly-size cheating verifier V^*_{WI}, such that

$$\mathsf{View}_{V^*_{\mathrm{WI}}}(P_{\mathrm{WI}}(1^\lambda, x_n, w_{1,n}), V^*_{\mathrm{WI}}(1^\lambda, x_n)) \not\approx \mathsf{View}_{V^*_{\mathrm{WI}}}(P(1^\lambda, x_n, w_{2,n}), V^*_{\mathrm{WI}}(1^\lambda, x_n)).$$

Assume w.l.o.g that V^*_{WI} is deterministic and denote $V^*_{\mathrm{WI}} = (V^*_{\mathrm{WI},1}, V^*_{\mathrm{WI},2})$ s.t. $(Q, \mathrm{sWI}_1, \psi) = V^*_{\mathrm{WI},1}(1^\lambda, x_n)$ generates the first message of V^*_{WI}, and $V^*_{\mathrm{WI},2}(c, \psi')$ is the distinguisher that takes the message from P_{WI} and outputs a bit. Note that λ determines n and thus also x and $(Q, \mathrm{sWI}_1, \psi)$. Let $\mathsf{st} = \mathsf{Sim}_1(\psi)$, where $\mathsf{Sim}_1(\cdot)$ is the possibly inefficient first part of the simulator for the remote evaluation scheme (see Section 2.2). Note that st is uniquely well defined per λ.

We design a cheating non-uniform adversary V^*_{sWI} for the strongly witness indistinguishable scheme. Note that since the adversary is allowed to be non-uniform, we can hard-code the values $(x_n, w_{1,n}, w_{2,n}, Q, \mathrm{sWI}_1, \psi, \mathsf{st})$ into V^*_{sWI}.

We start by defining the two distributions

$$\{\mathcal{X}'_{1,n(\lambda)}, \mathcal{W}'_{1,n(\lambda)}\}_{\lambda \in \mathbb{N}} \text{ and } \{\mathcal{X}'_{2,n(\lambda)}, \mathcal{W}'_{2,n(\lambda)}\}_{\lambda \in \mathbb{N}} \ ,$$

as required by the definition of sWI. The samplers for these distributions can also depend on $(x_n, w_{1,n}, w_{2,n}, Q, \mathrm{sWI}_1, \psi, \mathsf{st})$. Formally, for $b \in \{1, 2\}$, the distribution $(\mathcal{X}'_{b,n(\lambda)}, \mathcal{W}'_{b,n(\lambda)})$ generates pairs $(x'_{b,n}, w'_{b,n}) \in \mathcal{R}_{L'}$ as follows:

1. Emulate the prover $P_{\mathrm{WI}}(1^\lambda, x_n, w_{b,n}, Q, \mathrm{sWI}_1, \psi)$, as follows.
   (a) Compute $A \leftarrow P(1^\lambda, x_n, w_{b,n}, Q)$.
   (b) Compute $c = \mathrm{Com}(A, r)$ with uniformly chosen $r \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$.
2. Set $x'_{b,n} = (1^\lambda, x_n, Q, c, \mathsf{st})$ and $w'_{b,n} = (A, r)$.

The computational hiding property of the commitment scheme implies that indeed

$$\{\mathcal{X}'_{1,n(\lambda)}\}_{\lambda \in \mathbb{N}} \overset{c}{\approx} \{\mathcal{X}'_{2,n(\lambda)}\}_{\lambda \in \mathbb{N}} \ .$$

We still need to prove that $(x'_{b,n}, w'_{b,n}) \in R_{L'}$ for $b \in \{1, 2\}$. If $\mathsf{st} \models Q$ then this follows from the perfect completeness of (P, V). If $\mathsf{st} \not\models Q$ this follows by definition (see Eq. (1)).

For this pair of distributions, the cheating verifier V^*_{sWI} runs as follows.

1. Send the fixed value sWI_1 as the first message.
2. Receive $x' = (1^\lambda, x_n, Q, c, \mathsf{st})$ and message $\mathrm{sWI}_2 = P_{\mathrm{sWI}}(x', w', \mathrm{sWI}_1)$.
3. Generate simulated $\psi' = \mathsf{Sim}_2(\psi, \mathsf{st}, \mathrm{sWI}_2)$, where Sim_2 is the simulator for the remote evaluation scheme (see Section 2.2), and output $V^*_{\mathrm{WI},2}(c, \psi', \mathrm{sWI}_2)$.

To prove that V^*_{sWI} indeed distinguishes between the distributions $\{\mathcal{X}'_{b,n(\lambda)}, \mathcal{W}'_{b,n(\lambda)}\}_{\lambda \in \mathbb{N}}$, we consider a hybrid where ψ' is generated as $\mathsf{Eval}(f_{1^\lambda, x, Q, c, w'_{b,n}, \mathrm{sWI}_1}, \psi)$. This hybrid is computationally indistinguishable from the original experiment by Claim 2.2. However, in this hybrid the distribution given to $V^*_{\mathrm{WI},2}$ is identical to the one produced by P_{WI}, and since we assume that V^*_{WI} is a successful adversary against WI, it follows that our V^*_{sWI} successfully distinguishes between the distributions $\{\mathcal{X}'_{b,n(\lambda)}, \mathcal{W}'_{b,n(\lambda)}\}_{\lambda \in \mathbb{N}}$ in contradiction to the strong witness indistinguishability property. □

3 Succinct Single-Round Access Control Scheme

In this section we formalize the notion of succinct single-round access control presented in Section 1.2. The motivation is to allow authorities to provide users with certificates of owning certain attributes (coming from a very large attribute universe). An authority is specified by a pair of master secret and public keys. After being issued a certificate, the user can succinctly prove in a witness indistinguishable manner that its attributes (issued by a specific authority) satisfy a predicate from a given class of predicates. Note that similarly to the setting of secret sharing, only monotone predicates make sense in this setting, since users can always behave as if they do not have a certain attribute, even if they do. We can now formally define the notion of succinct access control schemes.

Definition 3.1. *A succinct access control scheme with respect to a class of monotone functions F consists of a tuple of* PPT *algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Query}, \mathsf{Proof}, \mathsf{Verdict})$*, with the following syntax:*

- Setup *takes as input the security parameter* 1^λ *and outputs a pair* $(\mathsf{mpk}, \mathsf{msk})$ *of master public and secret keys.*

- KeyGen *takes as input a tuple* $(1^\lambda, \mathsf{msk}, N, S, \mathsf{id})$*, where* λ *is the security parameter,* msk *is a master secret key (supposedly generated by* $\mathsf{Setup}(1^\lambda)$*),* $N \in \mathbb{N}$ *is a parameter such that* $N < 2^\lambda$*,* $S \subseteq [N]$*, and* $\mathsf{id} \in \{0,1\}^\lambda$*. It outputs a secret key* sk*.*

- Query *takes as input the security parameter* 1^λ *and outputs a pair* $(\mathsf{query}, \mathsf{state})$*.*

- Proof *takes as input a tuple* $(1^\lambda, f, \mathsf{query}, \mathsf{sk})$*, where* $f : \{0,1\}^N \to \{0,1\}$ *is a predicate from the class* F*,* query *is supposedly generated by running* $\mathsf{Query}(1^\lambda)$*, and* sk *is supposedly generated by running* KeyGen*. It outputs a* succinct *proof, denoted by* pf*, of length* $\leq \mathrm{poly}(\lambda)$*.*

- $\mathsf{Verdict}$ *takes as input a tuple* $(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf})$ *where* $f : \{0,1\}^N \to \{0,1\}$ *is a predicate from the class* F*,* $(\mathsf{query}, \mathsf{state})$ *is supposedly generated by* $\mathsf{Query}(1^\lambda)$*,* mpk *is supposedly generated by* $\mathsf{Setup}(1^\lambda)$*, and outputs* 1 *if and only if* pf *is accepting with respect to* $(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk})$*.*

  Moreover, the running time of $\mathsf{Verdict}$ *should be sublinear in the complexity of* f*, and only depend polynomially (or preferably quasi-linearly) on the input length and the description length of* f*.*

In addition, an access control scheme must satisfy the following conditions:

- **Completeness.** *For any* $\lambda \in \mathbb{N}$ *any* $N < 2^\lambda$*, any poly-size* $f : \{0,1\}^N \to \{0,1\}$*, any identity* $\mathsf{id} \in \{0,1\}^\lambda$*, and any set* $S \subseteq [N]$ *such that* $f(\mathbf{1}_{1\in S}, \ldots, \mathbf{1}_{N\in S}) = 1$*,*

$$\Pr[\mathsf{Verdict}(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf}) = 1] = 1 \ ,$$

  where the probability is over the random coin tosses of $\mathsf{Verdict}$*, over* $(\mathsf{query}, \mathsf{state}) \leftarrow \mathsf{Query}(1^\lambda)$*, over* $\mathsf{pf} \leftarrow \mathsf{Proof}(1^\lambda, f, \mathsf{query}, \mathsf{sk})$*, where* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda, \mathsf{msk}, N, S, \mathsf{id})$ *and* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*.*

- **Soundness.** *For any $\lambda \in \mathbb{N}$, any polynomially-bounded $N = N(\lambda)$, any poly-size $f : \{0,1\}^N \to \{0,1\}$, we consider an oracle $\mathcal{O} = \mathcal{O}_{\mathsf{msk}}$, that on input (id, S), outputs $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda, \mathsf{msk}, N, S, \mathsf{id})$ if and only if $\mathsf{id} \in \{0,1\}^\lambda$, $S \subseteq [N]$, and $f(\mathbf{1}_{1\in S}, \ldots, \mathbf{1}_{N\in S}) = 0$ (recall that f is monotone); and otherwise output $\perp$.*

  The soundness requirement is that for any PPT *adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ it holds that*

  $$\Pr[\mathsf{Verdict}(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf}^*) = 1] = \mathrm{negl}(\lambda),$$

  where $\mathsf{pf}^* \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{query}, \mathcal{A}_1^{\mathcal{O}_{\mathsf{msk}}}(1^\lambda, \mathsf{mpk}))$, and in addition $(\mathsf{query}, \mathsf{state}) \leftarrow \mathsf{Query}(1^\lambda)$ and $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.

  Note that $\mathcal{A}_2$ does not take any oracle access, i.e. the adversary is first allowed to interact with the oracle, and only then sees the protocol query. We can hope for a stronger variant where oracle access is allowed after seeing the queries as well, but we cannot currently achieve this stronger notion.

- **Witness Indistinguishability (WI).** *For any $\lambda \in \mathbb{N}$, any polynomially-bounded $N = N(\lambda)$, any poly-size $f : \{0,1\}^N \to \{0,1\}$ in F, any $\mathsf{id}_0, \mathsf{id}_1 \in \{0,1\}^\lambda$, and any sets $S_0, S_1 \subseteq [N]$ such that $f(\mathbf{1}_{1\in S_b}, \ldots, \mathbf{1}_{N\in S_b}) = 1$ for both $b = 0$ and $b = 1$, the following holds: For any* PPT *adversary $\mathcal{A}$ that generates $(\mathsf{query}^*, \mathsf{state}^*) = \mathcal{A}(1^\lambda, \mathsf{msk}, \mathsf{mpk})$,*

  $$(\mathsf{query}^*, \mathsf{state}^*, \mathsf{msk}, \mathsf{mpk}, \mathsf{pf}_0) \approx (\mathsf{query}^*, \mathsf{state}^*, \mathsf{msk}, \mathsf{mpk}, \mathsf{pf}_1),$$

  where

  $$\mathsf{pf}_b(\lambda) \leftarrow \mathsf{Proof}(1^\lambda, f, \mathsf{query}^*, \mathsf{sk}_b),$$

  where $\mathsf{sk}_b \leftarrow \mathsf{KeyGen}(1^\lambda, \mathsf{msk}, N, S_b, \mathsf{id}_b)$.

Remark 3.1. *We note that Definition 3.1 above guarantees that the identity of the prover remains hidden, even if the prover issues many proofs. This is the case since we require the WI property to hold even given* msk.

We next define delegation for batch **NP** families, which will be a building block for our construction. The construction and proof will then follow.

3.1 Delegation for Batch-NP Families

In this section we define the notion of a (succinct, single-round) delegation scheme for a subclass of languages in **NP**. While achieving the above for all of **NP** is still out of reach under falsifiable assumptions, many meaningful subclasses of **NP** admit such proof systems with short proof length [BHK17, BKK+17]. We start with a definition, and then proceed to derive corollaries based on known schemes in the literature.

Definition 3.2 (**NP**-Batching of a Function Family)**.** *Let $F \subseteq \{\{0,1\}^* \to \{0,1\}\}$ be a class of functions. Let $\mathcal{R}$ be an* **NP** *relation corresponding to an* **NP***-language L, with witness length $m = m(n)$ for length n instances. For $x \in \{0,1\}^n$ we let $\mathcal{R}_x : \{0,1\}^m \to \{0,1\}$ denote the function where $\mathcal{R}_x(w) = 1$ if and only if $(x, w) \in \mathcal{R}$. Let N be a polynomial.*

*We define the $\mathcal{R}^N$-*batching *of F, denoted $F^{(\mathcal{R},N)}$ as follows. For all n (recalling that m, N are a function of n), let $F_n = F \cap \left(\{\{0,1\}^N \to \{0,1\}\}\right)$. For all $f \in F_n$ define $g_f : (\{0,1\}^n)^N \times (\{0,1\}^m)^N \to \{0,1\}$ as*

$$g_f((x_1, \ldots, x_N), (w_1, \ldots, w_N)) = f(\mathcal{R}_{x_1}(w_1), \ldots, R_{x_N}(w_N)) \ . \tag{5}$$

We frequently denote $\mathbf{x} = (x_1, \ldots, x_N)$, $\mathbf{w} = (w_1, \ldots, w_N)$.

Finally, $F^{(\mathcal{R},N)}$ is the class of all such functions g_f, formally:

$$F_n^{(\mathcal{R},N)} = \{g_f : f \in F_n\} \tag{6}$$

$$F^{(\mathcal{R},N)} = \cup_{n \in \mathbb{N}} F_n^{(\mathcal{R},N)} \ . \tag{7}$$

We omit the superscript where $\mathcal{R}, N$ are clear from the context.

We can now define the notion of a succinct delegation scheme for a batch family.

Definition 3.3 (Delegation for Batch Families)**.** *Let $\mathcal{R}, N, F$ be as in Definition 3.2 and let $F^{(\mathcal{R},N)}$ be the N-batching of $\mathcal{R}$ with respect to F.*

A succinct and doubly efficient delegation scheme (or simply "a delegation scheme" for the purpose of this work) for $F^{(\mathcal{R},N)}$ with communication overhead cov and verification overhead vov (both fixed polynomial functions), is a single-round proof system (P, V) running on inputs $(1^\lambda, \mathbf{x}, M)$, where M is a Turing machine s.t. $M(1^n)$ runs in time $\mathsf{T}(n)$ and outputs a circuit that computes a function $g_n \in F_n^{(\mathcal{R},N)}$.

with the following guarantees.

- **Efficiency.** *The protocol (P, V), on input $(1^\lambda, \mathbf{x}, M)$, has the following efficiency guarantees, for $\mathbf{x} = (x_1, \ldots, x_N)$ where each $|x_i| = n$, and assuming λ is such that $\mathsf{T}(n) \in [\lambda, 2^\lambda]$:*

  1. *The communication complexity is $\mathsf{cov}(m, \lambda)$ (ideally $m \cdot \mathrm{poly}(\lambda)$), where m is the length of a witness corresponding to an instance of length n in $\mathcal{R}_L$.*
  2. *The runtime of V is $\mathsf{vov}(nN, |M|, m, \lambda)$, where $|M|$ denotes the size of the non-uniform advice of M.*
  3. *The runtime of P, given a witness $\mathbf{w}$ such that $g_n(\mathbf{x}, \mathbf{w}) = 1$, is $\mathrm{poly}(\mathsf{T}(n))$.*

- **Perfect Completeness.** *For every security parameter λ and any inputs $\mathbf{x} \in (\{0,1\}^n)^N$ and M such that $\mathsf{T}(n) \in [\lambda, 2^\lambda]$, and every satisfying assignment $\mathbf{w} \in (\{0,1\}^m)^N$ such that $g_n(\mathbf{x}, \mathbf{w}) = 1$:*

  $$\Pr\left[(P(\mathbf{w}), V)(1^\lambda, \mathbf{x}) = 1\right] = 1,$$

  where the probability is over the random coin tosses of V.

- **Soundness.** *For every machine M, for any function $n(\lambda)$ s.t. $\mathsf{T}(n(\lambda)) = \mathrm{poly}(\lambda)$ and for every constant $c \in \mathbb{N}$, there exists a* PPT *oracle machine $\mathcal{E}_c$ such that if there exists a non-uniform* PPT *cheating prover P^*, that on input 1^λ generates $\mathbf{x} = \mathbf{x} \in (\{0,1\}^n)^N$, such that for infinitely many $\lambda \in \mathbb{N}$,*

  $$\Pr\left[(P^*, V)(1^\lambda, \mathbf{x}, M) = 1\right] \geq \frac{1}{\lambda^c},$$

  then for these values of λ,

  $$\Pr[\mathcal{E}_c^{P^*}(1^\lambda) = (\mathbf{x}, \mathbf{w}) \ s.t. \ g_n(\mathbf{x}, \mathbf{w}) = 1] = 1 - \mathrm{negl}(\lambda).$$

3.2 Known Batch Delegation Schemes

Starting from the work of [BHK17], delegation schemes are known for a number of function classes F. (Recall a batch delegation scheme for the set of all functions from a standard assumption may be too much to hope for, unless major progress in delegation is made and succinct arguments for all of **NP** are constructed.)

Theorem 3.1 ([BHK17]). *There exists a succinct delegation scheme when the class F is the class of all conjunctions, with quasi-linear verification overhead, under the assumption that computational private-information retrieval (PIR) exists. Furthermore, the delegation scheme is sound even against time T adversaries if the PIR scheme is secure against $\mathrm{poly}(T)$ time adversaries.*

A different result can be achieved based on the construction of [BKK$^+$17].

Corollary 3.2 ([BKK$^+$17]). *For any constant c, there exists a succinct delegation scheme for $F = \mathsf{DSPACE}(n^c)$, with fixed polynomial verification overhead, under the assumption that private-information retrieval (PIR) with sub-exponential security exists. Furthermore, the delegation scheme can be made sound even against sub-exponential time adversaries.*

3.3 Our Scheme

We now formally state the result that is hinted in Section 1.2.

Theorem 3.3. *Let F be a class of monotone functions. There exists a succinct single-round access control scheme for F if the following exist for some super-polynomial function $T : \mathbb{N} \to \mathbb{N}$.*

- *A batch delegation scheme for the class F as per Definition 3.3 that is sound against $\mathrm{poly}(T)$-size cheating provers.*
- *A $\mathrm{poly}(T)$-secure 2-message oblivious transfer with statistical sender privacy (as in Definition 2.2 where Claim 2.4 is satisfied w.r.t. $\mathrm{poly}(T)$-size adversaries).*
- *A $\mathrm{poly}(T)$-secure signature scheme.*
- *A statistically-binding commitment scheme that can be broken in time $\mathrm{poly}(T)$.*

The required building blocks can be instantiated from various assumptions. For the delegation scheme, we can rely on the schemes in Theorem 3.1 and Corollary 3.2. Given the new maliciously statistical sender private oblivious transfer and private information retrieval schemes from the DDH, QR, LWE and Decisional Composite Residuosity (DCR, a.k.a Paillier) assumptions [DGI$^+$19], the following corollary follows.

Corollary 3.4. *There exists a succinct single-round access control schemes as follows:*

- *For conjunctions, assuming the quasi-poly hardness of either $\mathrm{DDH}, \mathrm{QR}, \mathrm{LWE}, \mathrm{DCR}$, and assuming the existence of a sub-exponentially secure one-way function.*
- *For monotone space-bounded computation, assuming the sub-exponential hardness of either $\mathrm{DDH}, \mathrm{QR}, \mathrm{LWE}, \mathrm{DCR}$, and assuming the existence of a sub-exponentially secure one-way function.*

The access control scheme uses the following components:

- A batch delegation scheme for the class F as per Definition 3.3 that is sound against $\mathrm{poly}(T)$-size cheating provers The schemes in Theorem 3.1 and Corollary 3.2 are examples for such schemes in the literature. We denote this delegation scheme by (P, V).
- WI compiler w.r.t the super polynomial function $T = T(n)$ (as in Theorem 2.5). This can be constructed assuming the existence of a $\mathrm{poly}(T)$-secure 2-message oblivious transfer with statistical sender privacy (as in Definition 2.2 where Claim 2.4 is satisfied w.r.t. $\mathrm{poly}(T)$-size adversaries), and assuming the existence of a statistically binding commitment scheme Com that can be broken in time $\mathrm{poly}(T)$.
- A $\mathrm{poly}(T)$-secure signature scheme SIG (i.e., one that is existentially unforgeable against chosen message attacks by a $\mathrm{poly}(T)$-size adversary), which can be based on any $\mathrm{poly}(T)$-hard to invert one-way function, and does not require additional assumptions.

In what follows, we first present an access control scheme without the WI guarantee. We denote the algorithms in this (non WI) scheme by

$$\mathsf{AccessControl}' = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Query}', \mathsf{Proof}', \mathsf{Verdict}')$$

We then use our WI compiler from Section 2 to compile $(\mathsf{Query}', \mathsf{Proof}', \mathsf{Verdict}')$ into a witness indistinguishable protocol $(\mathsf{Query}, \mathsf{Proof}, \mathsf{Verdict})$, thus obtaining our final access control scheme

$$\mathsf{AccessControl} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Query}, \mathsf{Proof}, \mathsf{Verdict}).$$

- $\mathsf{Setup}(1^\lambda)$ generates a pair of keys $(\mathsf{mpk}, \mathsf{msk})$ by running the key generation algorithm of the signature scheme SIG (with security parameter λ).
- $\mathsf{KeyGen}(1^\lambda, \mathsf{msk}, N, S, \mathsf{id})$ samples a random tag $\mathsf{tag} \in \{0,1\}^\lambda$, it computes a signature $\sigma_{\mathsf{tag},i} = \mathsf{Sign}_{\mathsf{msk}}(\mathsf{tag}\|i)$ for every attribute $i \in S$ and in addition $\sigma_{\mathsf{tag},0} = \mathsf{Sign}_{\mathsf{msk}}(\mathsf{tag}\|0)$. It outputs $(\mathsf{tag}, \{\sigma_{\mathsf{tag},i}\}_{i \in S \cup \{0\}})$.
- $\mathsf{Query}'(1^\lambda)$ generates a pair $(Q, \mathsf{st}) \leftarrow V(1^\lambda)$, where Q is the query string and st is the internal state. Note that we use the property that query string generation in the monotone delegation scheme is independent of the instance to be proven.
- $\mathsf{Proof}'(1^\lambda, f, Q, (\mathsf{tag}, \{\sigma_{\mathsf{tag},i}\}_{i \in S \cup \{0\}}))$ runs the prover P (from the monotone **NP** delegation scheme), respective to $(R, f, \{x_i\}_{i \in [N]})$, where $x_i = \mathsf{mpk}\|\mathsf{tag}\|i$, R is the **NP** relation defined by

  $$(\mathsf{mpk}\|\mathsf{tag}\|i, \sigma) \in R \Leftrightarrow \mathsf{Verify}_{\mathsf{mpk}}(\mathsf{tag}\|i, \sigma) = 1,$$

  and $f \in F$ is the function corresponding to the access structure. Let A denote the answer generated by P. Then Proof' outputs

  $$\mathsf{pf} = (A, \mathsf{tag}, \sigma_{\mathsf{tag},0}).$$

- $\mathsf{Verdict}'(1^\lambda, f, Q, \mathsf{pf}, \mathsf{st}, \mathsf{mpk})$ parses $\mathsf{pf} = (A, \mathsf{tag}, \sigma_{\mathsf{tag},0})$, checks that $\mathsf{Verify}_{\mathsf{mpk}}(\mathsf{tag}\|0, \sigma_{\mathsf{tag},0}) = 1$, and checks that A verifies correctly respective to the **NP** statement $\{x_i\}_{i \in [N]}$, where $x_i = \mathsf{mpk}\|\mathsf{tag}\|i$.

Adding Witness Indistinguishability. We wish to augment the scheme $\mathsf{AccessControl}'$, and specifically the algorithms $(\mathsf{Query}', \mathsf{Proof}', \mathsf{Verdict}')$ with witness indistinguishability properties using the transformation from Section 2. The first idea would be to simply replace the NP proof system (Q, P, V) with its WI version as obtained from Theorem 2.5, however this is insufficient since Proof' outputs $\mathsf{tag}, \sigma_{\mathsf{tag},0}$ in addition to the response A (note that outputting tag is necessary in order to avoid collusion, the role of the signature will be clarified in the proof). However, we can still consider the proof system defined by $(\mathsf{Query}', \mathsf{Proof}', \mathsf{Verdict}')$ itself, and apply Theorem 2.5 to this proof system in order to obtain witness indistinguishability. We now no longer need to think about batch verification, and simply consider the **NP** relation that takes $(\mathsf{mpk}, \mathsf{tag}, \{\sigma_{\mathsf{tag},i}\}_{i\in S})$ as instance, verifies the signatures w.r.t tag and verifies that $\mathcal{F}$ accepts relative to the resulting structure.

The description of the new $(\mathsf{Query}, \mathsf{Proof}, \mathsf{Verdict})$ thus follows from $(\mathsf{Query}', \mathsf{Proof}', \mathsf{Verdict}')$ above together with Theorem 2.5, which concludes the description of our access control scheme. We now show that the required properties still hold.

3.4 Proof of Theorem 3.3 For Our Construction

We show completeness, soundness and WI of the scheme described above. While completeness will follow straightforwardly based on the correctness of the components, and WI follows via the WI properties guaranteed in Theorem 2.5, soundness is a more serious challenge. The reason is that in the soundness experiment, the adversary has access to the oracle $\mathcal{O}$, which in turn uses msk. We want to show that an adversary that violates soundness can forge signatures, but this will require care with respect to the use of the oracle $\mathcal{O}$. Details follow.

Completeness. The completeness follows immediately from the completeness of (P, V), the completeness of the WI transformation (see Theorem 2.5), and the correctness of the signature scheme.

Soundness. We prove soundness by again referring to the variant without privacy $\mathsf{AccessControl}'$, which has a valid syntax for an access control scheme. We show that an adversary that breaks soundness in $\mathsf{AccessControl}$ can be used to construct an adversary that breaks soundness in $\mathsf{AccessControl}'$, and then proceed with ruling out this option given the properties of the proof system and signature scheme. We start by assuming that there exists $\mathcal{A}$ such that (infinitely often)

$$\Pr[\mathsf{Verdict}(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf}^*) = 1] \geq \delta(\lambda) \ ,$$

where $\delta(\lambda) \geq 1/\mathrm{poly}(\lambda)$, $\mathsf{pf}^* \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{query}, \mathcal{A}_1^{\mathcal{O}_{\mathsf{msk}}}(1^\lambda, \mathsf{mpk}))$, and where $(\mathsf{query}, \mathsf{state}) \leftarrow \mathsf{Query}(1^\lambda)$ and $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.

To reduce the soundness claim from $\mathsf{AccessControl}$ to $\mathsf{AccessControl}'$, we use the fact that the reduction from Theorem 2.5 is black-box, non-rewinding and instance preserving. This means that given an adversary that breaks the soundness of the delegation scheme $(\mathsf{Query}, \mathsf{Proof}, \mathsf{Verdict})$ with access to some oracle, then either there is an adversary with comparable advantage for $(\mathsf{Query}', \mathsf{Proof}', \mathsf{Verdict}')$ with the same instance and with access to the same oracle, or alternatively there is an adversary with the same oracle for the sWI scheme or for the remote evaluation scheme. However, for the latter cases oracle access can easily be simulated given msk, but sWI and the remove evaluation scheme are secure even given msk.

Therefore, an adversary against the soundness of $\mathsf{AccessControl}$ implies an adversary against the soundness of $\mathsf{AccessControl}'$. That is, we assume that thee exists $\mathcal{A}'$ such that (infinitely often)

$$\Pr[\mathsf{Verdict}'(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf}^*) = 1] \geq \delta'(\lambda),$$

where $\delta'(\lambda) \geq 1/\mathrm{poly}(\lambda)$, $\mathsf{pf}^* \leftarrow \mathcal{A}'_2(1^\lambda, \mathsf{query}, \mathcal{A}'^{\mathcal{O}_{\mathsf{msk}}}_1(1^\lambda, \mathsf{mpk}))$, and where $(\mathsf{query}, \mathsf{state}) \leftarrow \mathsf{Query}'(1^\lambda)$ and $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$. The idea now is to use Definition 3.3 to extract a signature out of the $\mathsf{AccessControl}'$ adversary and violate the unforgability of SIG, of course taking into account that $\mathcal{A}'$ has access to $\mathcal{O}$ which is capable of producing signatures. Let $\mathsf{pf}^* = (A^*, \mathsf{tag}^*, \sigma^*)$ denote the forged proof generated by $\mathcal{A}'$, let q denote a polynomial upper bound on the number of $\mathcal{O}$ queries made by $\mathcal{A}'$, and let $\mathrm{TAGS} = \{\mathsf{tag}_1, \ldots, \mathsf{tag}_q\}$ be the tags generated and returned by $\mathcal{O}$ on these calls (we recall for the future that tag_i are generated uniformly and independently).

We now consider two cases:

1. In the first case, the adversary $\mathcal{A}'$ forges on a new tag:

$$\Pr[\mathsf{Verdict}'(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf}^*) = 1 \wedge \mathsf{tag}^* \notin \mathrm{TAGS}] \geq \delta'(\lambda)/2$$

   infinitely often. In this case, we do not need to use the soundness of the argument system (P, V) since pf^* includes σ^* which is a valid signature on $(\mathsf{tag}^* \| 0)$. Therefore $\mathcal{O}$ can be simulated using a chosen message oracle for the signature scheme, we are guaranteed that this oracle will never be called on $(\mathsf{tag}^* \| 0)$ and therefore we can succeed in the forgery attack.

2. In the second case $\mathcal{A}'$ forges on a tag that was generated by $\mathcal{O}$:

$$\Pr[\mathsf{Verdict}'(1^\lambda, f, \mathsf{query}, \mathsf{state}, \mathsf{mpk}, \mathsf{pf}^*) = 1 \wedge \mathsf{tag}^* \in \mathrm{TAGS}] \geq \delta'(\lambda)/2$$

   infinitely often. Note that either this case or the previous one must occur. In this case σ^* is not useful since we queried the chosen message oracle on $(\mathsf{tag}^* \| 0)$. We therefore would like to use the extraction property of (P, V) to extract a signature on $(\mathsf{tag}^* \| i)$ for $i \notin S^*$ (the set that corresponds to the query that produced tag^*).

   We therefore need to convert $\mathcal{A}'$ into a non-adaptive adversary for (P, V). First, we notice that (P, V) is applied to an **NP** language whose instances are of the form $(\mathsf{mpk}, \mathsf{tag})$ and the values $\{x_i\}$ constitute a possible witness. Since Definition 3.3 only allows to extract from a non-adaptive adversary, we will need to create an adversary that decides on $(\mathsf{mpk}, \mathsf{tag})$ before seeing query. The separation between $\mathcal{A}'_1$ and $\mathcal{A}'_2$ will thus be useful.

   Consider the following distribution over non-adaptive adversaries against (P, V). More accurately, our distribution will sample pairs of instance $(\mathsf{mpk}, \mathsf{tag}')$ and algorithm $\mathcal{B}$ s.t. $\mathcal{B}(Q)$ is accepted by V with non-negligible probability, where Q is a sampled according to the V-prescribed distribution. The distribution is generated as follows. Sample $\mathsf{mpk}, \mathsf{msk}$ and a random set TAGS. Set tag' to be a random element from TAGS. Execute $\mathcal{A}'^{\mathcal{O}}_1(1^\lambda, \mathsf{mpk})$ to obtain a value ζ, and set $\mathcal{B}(Q) = \mathcal{A}'_2(1^\lambda, Q, \zeta)$. The expected probability of success of $\mathcal{B}(Q)$ is at least $\delta'(\lambda)/(2q) > 1/\mathrm{poly}(\lambda)$, since with probability $1/q$ the guess tag' hits the correct tag^* (which is an element of TAGS with probability $\delta'(\lambda)/2$). This means with non-negligible probability, we sample $(\mathsf{mpk}, \mathsf{tag}')$ for which $\mathcal{B}$ succeeds with non-negligible probability.

   We can thus apply the extractor that is implied by Definition 3.3, to conclude that with non-negligible probability, it is possible to extract a set of signatures on messages $(\mathsf{tag}^* \| i)$ for all $i \in S$ where S satisfies $\mathcal{F}$. However, in the above experiment, the oracle $\mathcal{O}$ can be replaced with access to a chosen message signature oracle. We are guaranteed that this oracle is not queries on any S that satisfies $\mathcal{F}$. Therefore the extractor will allow to produce a signature on a message for which the chosen message oracle was not queried, thus violating the unforgability of the signature scheme with non-negligible probability. This completes the soundness argument.

Witness Indistinguishability. The WI condition follows immediately from the fact that the commitment scheme is (computationally) hiding, and from the strong WI property of $(P_{\mathrm{sWI}}, V_{\mathrm{sWI}})$.

References

[BBK+16] Nir Bitansky, Zvika Brakerski, Yael Tauman Kalai, Omer Paneth, and Vinod Vaikuntanathan. 3-message zero knowledge against human ignorance. *IACR Cryptology ePrint Archive*, 2016:213, 2016.

[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.

[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.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistical sender-private OT from LWE. *IACR Cryptology ePrint Archive*, 2018:530, 2018.

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 474–482. ACM, 2017.

[BKK+17] Saikrishna Badrinarayanan, Yael Tauman Kalai, Dakshita Khurana, Amit Sahai, and Daniel Wichs. Non-interactive delegation for low-space non-deterministic computation. Cryptology ePrint Archive, Report 2017/1250, 2017. To appear in STOC 2018.

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.*, pages 199–203. Plenum Press, New York, 1982.

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[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.

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications, 2019. To appear in CRYPTO 2019.

[GHV10] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. i-hop homomorphic encryption and rerandomizable yao circuits. In Tal Rabin, editor, *Advances in Cryptology -*

CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings, volume 6223 of *Lecture Notes in Computer Science*, pages 155–172. Springer, 2010.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 113–122. ACM, 2008. Full version in [GKR15].

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27, 2015.

[HK07] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *IACR Cryptology ePrint Archive*, 2007:118, 2007.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 158–189. Springer, 2017.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 78–95. Springer, 2005.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. *IACR Cryptology ePrint Archive*, 2018:168, 2018.

[KP15] Yael Tauman Kalai and Omer Paneth. Delegating RAM computations. *IACR Cryptology ePrint Archive*, 2015:957, 2015.

[KPY18] Yael Kalai, Omer Paneth, and Lisa Yang. On publicly verifiable delegation from standard assumptions. *IACR Cryptology ePrint Archive*, 2018:776, 2018. To appear in STOC 2019.

[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. Full version in SIAM J. Comput., 30(4):1253 1298, 2000.

[Nao89] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 128–136. Springer, 1989.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 448–457, 2001.

[OPP14] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 536–553. Springer, 2014.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62, 2016.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982.