SIAM J. COMPUT.
Vol. 0, No. 0, pp. 000–000

ZERO-KNOWLEDGE FROM SECURE MULTIPARTY COMPUTATION[*]

YUVAL ISHAI[†], EYAL KUSHILEVITZ[‡], RAFAIL OSTROVSKY[§], AND AMIT SAHAI[§]

Abstract. A *zero-knowledge proof* allows a prover to convince a verifier of an assertion without revealing any further information beyond the fact that the assertion is true. *Secure multiparty computation* allows n mutually suspicious players to jointly compute a function of their local inputs without revealing to any t corrupted players additional information beyond the output of the function. We present a new general connection between these two fundamental notions. Specifically, we present a general construction of a zero-knowledge proof for an NP relation $R(x, w)$, which makes only a *black-box* use of any secure protocol for a related *multiparty* functionality f. The latter protocol is required only to be secure against a small number of "honest but curious" players. We also present a variant of the basic construction that can leverage security against a large number of *malicious* players to obtain better efficiency. As an application, one can translate previous results on the efficiency of secure multiparty computation to the domain of zero-knowledge, improving over previous constructions of efficient zero-knowledge proofs. In particular, if verifying R on a witness of length m can be done by a circuit C of size s, and assuming that one-way functions exist, we get the following types of zero-knowledge proof protocols: (1) *Approaching the witness length.* If C has constant depth over $\wedge, \vee, \oplus, \neg$ gates of unbounded fan-in, we get a zero-knowledge proof protocol with communication complexity $m \cdot \text{poly}(k) \cdot \text{polylog}(s)$, where k is a security parameter. (2) *"Constant-rate" zero-knowledge.* For an *arbitrary* circuit C of size s and a bounded fan-in, we get a zero-knowledge protocol with communication complexity $O(s) + \text{poly}(k, \log s)$. Thus, for large circuits, the ratio between the communication complexity and the circuit size approaches a constant. This improves over the $O(ks)$ complexity of the best previous protocols.

Key words. cryptography, zero-knowledge, secure computation, black-box reductions

AMS subject classifications. AUTHOR: PLEASE PROVIDE

DOI. 10.1137/080725398

1. Introduction.

In this work we establish a new general connection between two of the most fundamental tasks in cryptography: zero-knowledge proofs and secure multiparty computation. Before explaining and motivating this connection, we give some relevant background to put it in context.

A *zero-knowledge proof* protocol [24] allows a prover to convince a verifier of the validity of a statement without revealing any further information about the proof beyond the fact that the statement is true. A more general problem is that of *secure*

[*]Received by the editors February 14, 2008; accepted for publication (in revised form) December 2, 2008; published electronically DATE. A preliminary version of this paper appeared in the Proceedings of STOC 2007 [32]. This work was done in part while the authors were visiting IPAM.
http://www.siam.org/journals/sicomp/x-x/72539.html

[†]Computer Science Department, Technion, Haifa 32000, Israel, and Computer Science Department, UCLA, Los Angeles, CA 90095 (yuvali@cs.technion.ac.il). This author's research was supported by BSF grant 2004361, ISF grant 1310/06, and NSF grants 0205594, 0430254, 0456717, 0627781, 0716835, and 0716389.

[‡]Computer Science Department, Technion, Haifa 32000, Israel (eyalk@cs.technion.ac.il). This author's research was supported by grant 1310/06 from the Israel Science Foundation and by grant 2002354 from the U.S.-Israel Binational Science Foundation.

[§]Computer Science Department and Math Department, UCLA, Los Angeles, CA 90095 (rafail@cs.ucla.edu, sahai@cs.ucla.edu). The third author's research was supported in part by an IBM Faculty Award, a Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389, and 0830803, and a U.C. MICRO grant. The fourth author's research was supported in part by grants from the NSF ITR and Cybertrust programs (including grants 0627781, 0456717, 0830803, and 0205594), BSF grant 2004361, a subgrant from SRI as part of the Army Cyber-TA program, an equipment grant from Intel, an Alfred P. Sloan Foundation Fellowship, and an Okawa Foundation Research Grant.

multiparty computation (MPC) [56, 21, 3, 8]. An MPC protocol allows n players to compute a function of their inputs (also referred to as an "n-party functionality") while maintaining the privacy of their inputs and the correctness of the output. A bit more precisely, the protocol should prevent an adversary which may corrupt at most t players from achieving more than it could have achieved by attacking an idealized protocol in which the function is computed by an external trusted party. In particular, corrupted players should not learn further information about the remaining inputs beyond the output of the function. Zero-knowledge protocols can be viewed as a special case of secure two-party computation, where the function verifies the validity of a witness held by the prover.

Honest versus dishonest majority

MPC protocols can be divided into two categories: ones that rely on the existence of an honest majority (namely, assume that $t < n/2$) [3, 8, 50] and ones that can offer security even when there is no honest majority [56, 21, 17]. In the case of an honest majority, it is possible to obtain "information-theoretic" security that holds unconditionally, whereas in the case of no honest majority, one needs to settle for computational security that holds under cryptographic assumptions. On the other hand, the honest majority assumption is often too strong and is meaningless in the important two-party case.

Constructions of these two types of protocols differ vastly in the types of techniques they employ. While protocols from the second category (most notably, zero-knowledge protocols) have occasionally proved useful for the design of protocols from the first category, we are not aware of applications that go in the other direction. This work is motivated in part by the hope of leveraging the efficiency and simplicity of MPC with honest majority (as well as the large body of relevant work) for the design of efficient protocols in the case of no honest majority, and in particular for the design of efficient zero-knowledge protocols.

Semihonest versus malicious players

Another major distinction is between MPC protocols that offer security against *semihonest* (i.e., "honest but curious") players, who follow the protocol's instructions but may try to gain additional information from what they see, and protocols that offer security against *malicious* players, who may arbitrarily deviate from the protocol's instructions in order to compromise the privacy of the inputs or the correctness of the output. We refer to these two types of protocols as being secure in the semihonest model and the malicious model, respectively. Security in the semihonest model is typically much easier to realize than security in the malicious model. Remarkably, the celebrated result of Goldreich, Micali, and Wigderson [20, 21] shows how to compile an *arbitrary* MPC protocol Π which securely computes a function f in the semihonest model into a protocol Π' which securely computes f in the malicious model. The high level technique, known as the "GMW paradigm," is to require players to validate every message they send by supplying a zero-knowledge proof that the message is consistent with protocol's Π specification. To implement the zero-knowledge proofs, the compiler needs to look into the code[1] of Π rather than use it as a black-box, and the resulting protocol Π' needs to perform multiple cryptographic operations for every gate in the circuit implementing Π. Thus, applying the GMW paradigm may

[1]When we talk of "black-box" use of the protocol Π, we mean simply invoking the next-message-functions of each player in the protocol without looking inside the details of the circuits or TMs that describe these functions. This is in keeping with the standard usage of the term "black-box" when talking of reductions between cryptographic primitives (cf. [52]).

involve a considerable efficiency overhead, which gets worse with the computational complexity of Π. A second source of motivation for our work is the goal of finding alternative "black-box" approaches for boosting security in the semihonest model into security in the malicious model.

1.1. Our contribution. We present a general construction of a zero-knowledge proof system Π_R for an NP relation $R(x, w)$ which makes a black-box [2] use of an MPC protocol Π_f for a related multiparty functionality f along with a bit commitment protocol. The functionality f can be efficiently defined by making only a black-box (oracle) access to R.

The MPC protocol Π_f may involve an arbitrary number of players n, and needs to be secure only against two *semihonest* players. In particular, the MPC protocol needs to be secure only in the presence of an honest majority. (Our main construction allows Π_f to employ an ideal oblivious transfer (OT) [49, 14] channel between each pair of players. For the case that Π_f does not require OT channels, we present a variant of the construction in which Π_f needs to be secure only against *one* semihonest player.)

The basic variant of our construction proceeds as follows. Define $f(x, w_1, w_2, \ldots, w_n) = R(x, w_1 \oplus w_2 \oplus \cdots \oplus w_n)$. The function f is viewed as an n-party functionality, where x (an NP statement) is known to all n players, and w_i (the ith share of the witness) is a private input of Player i. Note that f is efficiently defined using an oracle to R. The zero-knowledge protocol Π_R begins with the prover secret-sharing (on her private work-tape) the witness w into n additive shares, picking random $w_1, \ldots, w_n$ such that $w = w_1 \oplus \cdots \oplus w_n$. The prover then "runs in her head" the given n-party protocol Π_f for the functionality f, using the statement x and the shares $w_1, \ldots, w_n$ as inputs for the n players. After this execution is completed, the prover begins her interaction with the verifier. The prover commits to the view of each of the n players, and the verifier picks a random pair of distinct players i, j and challenges the prover to open the committed views of these players. Finally, the verifier accepts if the opened views are consistent with each other (with respect to Π_f) and the outputs in these views are 1. (If Π_f employs OT channels, this consistency check applies also to the inputs and output of each OT channel.)

The zero-knowledge property of the protocol follows from the security of Π_f against two semihonest players. Assuming that Π_f is perfectly correct, violating the soundness requires a cheating prover to generate at least one pair of inconsistent views, which is detected with probability at least $1/\binom{n}{2}$. Using $O(kn^2)$ repetitions, the soundness error can be decreased to 2^{-k}.

The black-box nature of our transformation suggests the possibility of significantly better efficiency. In particular, the *communication* complexity of the resulting zero-knowledge protocols can be smaller than the *computational* complexity of the MPC protocols from which they are derived. This should be contrasted with traditional zero-knowledge proofs (obtained via a Karp reduction to some fixed NP-complete problem) whose communication complexity is typically much bigger than the computational complexity of the relation R.

Towards obtaining more efficient variants of the construction which avoid the cost of soundness amplification via repetition, we employ MPC protocols that tolerate a larger number of corruptions and allow the verifier to open many views at once. In this case, security of Π_f against semihonest players does not suffice, and we need to rely on MPC protocols that have some form of robustness against t malicious players

[2]See previous footnote for an explanation of the use of the term "black-box" here.

(where typically t will be a constant multiple of the soundness parameter k, and n a constant multiple of t).

The intuition behind this modification is that it will allow the verifier to obtain t views, rather than just two views, from a single execution of the protocol. Security against t *malicious* players guarantees that in order to violate the correctness of the MPC protocol Π_f (or the soundness of the zero-knowledge protocol) the inconsistencies between the views must be "wellspread" in such a way that opening t random views reveals an inconsistency with overwhelming probability. Note that security against t semihonest players does not suffice here. Indeed, in this case a single malicious player can cause all other players to have an incorrect output; if this particular player is not picked by the verifier, no inconsistency is revealed and still the output may be incorrect. On the other hand, we do not need full security against malicious players: it suffices for the protocol to be *correct* in the presence of malicious players, whereas its privacy is required to hold, as before, only against semihonest players.

Another extension of the basic construction that is needed for our applications is to the "statistical" case where Π_f may produce an incorrect output with negligible probability. This case makes even the simple basic construction fail, as it allows the adversary to cheat by biasing the randomness used for generating the views (but otherwise behaving according to the protocol). We show how to overcome this obstacle using standard cryptographic techniques, though for the efficient version of our construction this involves some additional complications.

Applications. To demonstrate the usefulness of our general approach, we translate previous results on the efficiency of MPC to the domain of zero-knowledge, improving over some previous constructions of efficient zero-knowledge protocols. In particular, if verifying $R(x, \cdot)$ on a witness w of length m can be done by a circuit C of size s, we get the following types of zero-knowledge proof protocols.

Simple zero-knowledge. By plugging in standard MPC protocols for the semihonest model, such as the 2-private 3-player GMW protocol [21, 17] (which relies on OT channels) or the 1-private 3-player BGW protocol [3], we get simple zero-knowledge protocols with complexity $O(ks)$. (Here and in the following, k denotes a security parameter, and the soundness error is at most 2^{-k}.) These protocols provide more efficient alternatives to the classical zero-knowledge protocols based on 3-colorability or Hamiltonicity, since they apply directly to the circuit representation and do not require a Karp reduction to these NP-complete problems. In contrast to the classical protocols, zero-knowledge protocols derived from the BGW protocol can be efficiently extended to prove relations defined by *arithmetic* circuits, without requiring us to emulate arithmetic computations via Boolean operations.

Approaching the witness length. If C has constant depth over $\wedge, \vee, \oplus, \neg$ gates of unbounded fan-in, we get a zero-knowledge proof protocol with $m \cdot \text{poly}(k) \cdot \text{polylog}(s)$ bits of communication. This protocol can be implemented using a one-way function.[3] Alternatively, our technique yields a zero-knowledge proof system in a noninteractive model with preprocessing. The complexity of our protocols is comparable to that of a previous protocol of Kalai and Raz [34]. The latter, however, yields only zero-knowledge *argument* systems and relies on stronger assumptions, but

[3]A similar result was obtained independently by Kalai and Raz [35]. In a subsequent work, Goldwasser, Kalai, and Rothblum [23] extended the class of circuits for which $m \cdot \text{poly}(k, \log s)$ communication complexity can be achieved to polylog-depth circuits (capturing the complexity class NC). We note that our approach is very different from the approaches taken in [35, 23], which rely on variants of interactive proofs rather than MPC.

requires less communication for implementing the setup. Our zero-knowledge protocols, described above, rely on an MPC protocol from [1], whose basic version combines the BGW protocol [3] with the polynomial representation techniques of [51, 55]. We note that the protocols we obtain are essentially the best one could hope for in terms of both the underlying assumption (one-way functions) and the communication complexity. Indeed, the existence of nontrivial zero-knowledge proofs implies the existence of "nonuniform" one-way functions [47], and a lower communication complexity for zero-knowledge *proofs* (as opposed to arguments [37]) would imply surprisingly efficient probabilistic algorithms for SAT [19].

"Constant-rate" zero-knowledge. Assuming that one-way functions exist, we get a zero-knowledge proof protocol for an *arbitrary* circuit C of size s and bounded fan-in with communication complexity $O(s) + \mathrm{poly}(k, \log s)$. Thus, for large circuits, the ratio between the communication complexity and the circuit size approaches a constant. This improves over the $O(ks)$ complexity of the best previous protocols (e.g., [10]). Our zero-knowledge protocol relies on an MPC protocol from [13], optimized to work over constant-size fields using secret sharing based on algebraic-geometric codes [9].

Perspective. Our construction has direct relevance to the two motivating goals discussed above. First, it establishes a new general connection between zero-knowledge and MPC, which in particular allows us to exploit the large body of work on MPC with honest majority for the design of zero-knowledge proofs. The latter, in turn, can be used for the design of general secure two-party protocols and MPC with no honest majority. This new connection also establishes a link between some remaining open questions concerning the efficiency of zero-knowledge proofs and similar open questions in the area of MPC. For instance, if *all* circuits can be securely evaluated with communication complexity that depends (say) quadratically on the input size, then all NP-relations have zero-knowledge proofs whose communication complexity is roughly quadratic in the witness size.

Second, our construction shows a useful class of applications for which security in the semihonest model can be boosted in a *black-box* way into security in the malicious model. It is instructive to note that it is generally impossible to *directly* construct a protocol Π' which securely computes a function R in the malicious model by making a black-box use of a protocol Π which securely computes R in the semihonest model. For instance, if R is a zero-knowledge functionality, in the semihonest model we can consider a trivial protocol Π in which the prover simply sends to the verifier the output of R on its input (x, w). Clearly, Π is generally useless for the construction of Π'. By modifying R into a related multiparty functionality f, we avoid this impossibility.[4]

1.2. Related work. *Black-box reductions*

A rich body of work, initiated by the seminal paper of Impagliazzo and Rudich [30], attempts to draw the lines between possibility and impossibility of black-box reductions in cryptography. In particular, black-box constructions of MPC protocols from various cryptographic primitives have been previously suggested [36, 41, 38, 12, 31, 26]. Our work differs from all this work in that it treats as a black-box not only the underlying cryptographic primitives but also the *functionality* that should be realized. In contrast, previous constructions address either the case of some fixed functionality

[4]Here it is crucial to insist that f make a black-box use of R; otherwise, the output of f can encode the description of a circuit computing R, allowing us to use the standard (non–black-box) GMW paradigm.

or alternatively make a non–black-box use of the functionality.

Efficiency of zero-knowledge

There has been a vast body of work on improving the efficiency of interactive and noninteractive zero-knowledge proofs; e.g., see [5, 10, 39, 40, 11, 6, 25, 34] and references therein. It is important to note in this context that by settling for computational soundness, one can get asymptotically very efficient zero-knowledge *argument* systems with a polylogarithmic communication complexity. These argument systems can be either interactive and based on collision-resistant hash functions [37] or noninteractive in the random oracle model [43]. Our focus on proofs rather than arguments is motivated both from a theoretical and from a practical point of view. From a theoretical point of view, we get a stronger notion of soundness under weaker assumptions. From a more practical point of view, the probabilistically checkable proof (PCP)-based constructions of arguments, as in [37, 43], are still quite far from being efficient in practice. Finally, our results are independently motivated by the general goal of boosting security against semihonest players into security against malicious players in a *black-box* way. This motivation applies both to proofs and to arguments.

Organization. In section 2 we give some preliminaries. The basic construction of zero-knowledge protocols from MPC in the semihonest model is described in section 3, including (in section 3.2) the application to zero-knowledge protocols whose communication complexity is close to the witness length. In section 4, we present our constructions of zero-knowledge protocols from MPC in the malicious model. These are motivated by the application to constant-rate zero-knowledge, presented in section 4.4. We conclude with some final remarks in section 5.

2. Preliminaries.

In this section we define the notions of zero-knowledge proofs and MPC protocols.

Indistinguishability. A function $\epsilon(\cdot)$ is *negligible* if it is asymptotically smaller than the inverse of any polynomial. That is, for every constant $c > 0$ and all sufficiently large n we have $\epsilon(n) < 1/n^c$. We say that $\epsilon(\cdot)$ is *overwhelming* if $1 - \epsilon$ is negligible. Let $X \subseteq \{0,1\}^*$ be an infinite set of strings. Two distribution ensembles $\{A_x\}_{x \in X}$ and $\{B_x\}_{x \in X}$ are said to be *computationally indistinguishable* if, for every efficient nonuniform distinguisher D, there is a negligible function $\epsilon(\cdot)$ such that for every $x \in X$ we have $|\Pr[D(A_x) = 1] - \Pr[D(B_x) = 1]| \leq \epsilon(|x|)$.

2.1. Zero-knowledge.

We use the standard notion of zero-knowledge proofs from the literature [24, 20, 16], adapted to the case where the honest prover should be efficient.

An NP-relation $R(x, w)$ is an efficiently decidable binary relation which is polynomially bounded, in the sense that there is a polynomial $p(\cdot)$ such that if $R(x, w)$ is satisfied, then $|w| \leq p(|x|)$. We naturally view R as a Boolean function which outputs 0 or 1. Note that any NP-relation R defines an NP-language $L = \{x : \exists w \; R(x, w) = 1\}$.

A zero-knowledge proof protocol for an NP-relation $R(x, w)$ is defined by two PPT interactive algorithms, a prover P and a verifier V. Initially the prover is given an NP statement x and a corresponding witness w, and the verifier is given only the statement x. The prover and the verifier interact using the next-message function defined by their respective algorithms. This function determines the next message to be sent based on the inputs, messages received so far, and the coin tosses of the corresponding party.

DEFINITION 2.1 (zero-knowledge proof). *The protocol* (P, V) *is a* zero-knowledge proof protocol *for the relation* R *if it satisfies the following requirements:*

- Completeness. *If $R(x, w) = 1$ and both players are honest (namely, compute messages they send according to the protocol), the verifier always accepts.*
- Soundness. *For every malicious and* computationally unbounded *prover P^*, there is a negligible function $\epsilon(\cdot)$ such that if x is a false statement (namely, $R(x, w) = 0$ for all w), the interaction of P^* with V on input x makes V reject except with at most $\epsilon(|x|)$ probability.*
- Zero-knowledge. *For any malicious PPT verifier V^* there is a PPT simulator M^* such that the view of V^*, when interacting with P on inputs (x, w) for which $R(x, w) = 1$, is computationally indistinguishable from the output distribution of $M^*(x)$. That is, there exists a negligible $\delta(\cdot)$ such that for every efficient nonuniform distinguisher D and every x, w such that $(x, w) \in R$ we have*

$$|\Pr[D(\mathsf{View}_{V^*}(x, w)) = 1] - \Pr[D(M^*(x)) = 1]| \leq \delta(|x|),$$

where View_{V^*} denotes the view of V^*, consisting of its input x, its coin-tosses, and its incoming messages.

We will sometimes relax the completeness requirement to allow a negligible error probability. Finally, we will also consider zero-knowledge protocols that have a non-negligible (typically constant) soundness error ϵ. In such cases the soundness error will be specified.

For the sake of simplicity, we do not consider the stronger *proof of knowledge* property (cf. [16]) of zero-knowledge protocols. However, our general constructions can be shown to satisfy this property as well.

Using an explicit security parameter. Note that the above standard definition of zero-knowledge proofs implicitly treats the length of the statement x as a security parameter, requiring the advantage of a dishonest party (prover or verifier) to be negligible in $|x|$. While this is convenient for establishing feasibility results, when studying the concrete *efficiency* of cryptographic protocols it is useful to introduce a separate security parameter k which is given as an additional input to all algorithms. In such a case, we modify the soundness and zero-knowledge requirements in Definition 2.1 by replacing $\epsilon(|x|)$ and $\delta(|x|)$ by $\epsilon(k)$ and $\delta(k)$, respectively. Furthermore, in the zero-knowledge requirement we restrict the running time of V^* and D to be polynomial in k. A zero-knowledge protocol satisfying the latter definition can be turned into a zero-knowledge protocol satisfying the original definition by letting $k = |x|^c$, where $c > 0$ can be an arbitrarily small constant.[5] Thus, k can be thought of as being asymptotically much smaller than the input length.

2.2. Idealized primitives. When describing our zero-knowledge protocols, it will be convenient to first describe and analyze them in a *hybrid* model in which idealized versions of primitives such as bit commitment or coin-flipping are available. We then explicitly describe how to instantiate invocations of the ideal primitives (only making a black-box use of a one-way function) in order to get a protocol in the plain model. Such a modular design approach is common in the area of MPC (cf. [7, 17]). However, since we are dealing with *unconditionally sound* proofs rather than computationally sound arguments, we cannot apply off-the-shelf composition theorems to deduce the security of the final protocol, and need to argue each case separately.

[5]In fact, for all protocols presented in this paper one can let k be as small as polylog$(|x|)$ if the underlying one-way function is assumed to be exponentially strong.

We note that, despite the lack of general tools for a modular design of zero-knowledge proofs, the analysis of the zero-knowledge protocols we obtain by instantiating the ideal primitives combines in a rather straightforward way the analysis of our protocols in the hybrid model with the analysis of previous zero-knowledge protocols from the literature. Thus, the main contribution of this work may best be viewed within the hybrid model.

2.3. Secure MPC. We use standard definitions of MPC from the literature [7, 16]. Our basic model assumes synchronous communication over secure point-to-point channels. (Some protocols will also rely on other ideal primitives such as OT channels, broadcast primitive, or ideal coin-flipping. These will be explained when they are used.)

Let n be the number of players, which will be denoted by $P_1, \ldots, P_n$. All players share a public input x, and each player P_i holds a local private input w_i. We consider protocols which securely realize an n-party functionality f, where f maps the inputs $(x, w_1, \ldots, w_n)$ to an n-tuple of outputs. (When only a single output is specified, this output is assumed to be given to all players.)

A protocol Π is specified via its next message function. That is, $\Pi(i, x, w_i, r_i, (m_1, \ldots, m_j))$ returns the set of n messages (and, possibly, a broadcast message) sent by P_i in round $j + 1$, given the public input x, its local input w_i, its random input r_i, and the messages $m_1, \ldots, m_j$ that it *received* in the first j rounds. In the case of statistical or computational security, Π receives a security parameter k as an additional input. The output of Π may also indicate that the protocol should terminate, in which case Π returns the local output of P_i. The *view* of P_i, denoted by V_i, includes w_i, r_i, and the messages *received* by P_i during the execution of Π. Note that the messages sent by an uncorrupted player P_i as well as its local output can be inferred from V_i and x by invoking Π. It will be useful to define the following natural notion of consistency between views.

DEFINITION 2.2 (consistent views). *We say that a pair of views V_i, V_j is* consistent *(with respect to the protocol Π and some public input x) if the outgoing messages implicit in V_i, x are identical to the incoming messages reported in V_j and vice versa.*

The following simple lemma asserts that an n-tuple of views corresponds to some honest execution of Π if and only if every pair of views is consistent.

LEMMA 2.3 (local versus global consistency). *Let Π be an n-party protocol as above and x be a public input. Let $V_1, \ldots, V_n$ be an n-tuple of (possibly incorrect) views. Then all pairs of views V_i, V_j are consistent with respect to Π and x* if and only if *there exists an honest execution of Π with public input x (and some choice of private inputs w_i and random inputs r_i) in which V_i is the view of P_i for every $1 \leq i \leq n$.*

Proof. The "if" direction follows directly from the definitions. For the "only if" direction, let $V_1, \ldots, V_n$ be pairwise consistent views, and let w_i, r_i be the input and random input reported in view V_i. The pairwise consistency of the views implies, by induction on the number of rounds d, that the actual view of every player P_i after d rounds when Π is invoked on $(x, (w_1, r_1), \ldots, (w_n, r_n))$ is the same as the view of P_i in the first d rounds reported in V_i. It follows that the n views $V_1, \ldots, V_n$ are consistent with the full execution of Π, as required. $\square$

We consider security of protocols in both the semihonest and the malicious models. In the semihonest model, one may break the security requirements into the following correctness and privacy requirements.

DEFINITION 2.4 (correctness). *We say that Π realizes a deterministic n-party* *functionality* $f(x, w_1, \dots, w_n)$ *with* perfect (resp., statistical) correctness *if for all inputs* $x, w_1, \dots, w_n$ *the probability that the output of some player is different from the output of* f *is* 0 *(resp., negligible in* k*), where the probability is over the independent choices of the random inputs* $r_1, \dots, r_n$.

Definition 2.5 (t-privacy). *Let* $1 \leq t < n$. *We say that* Π *realizes* f *with* perfect t-privacy *if there is a PPT simulator* Sim *such that for any inputs* $x, w_1, \dots, w_n$ *and every set of corrupted players* $T \subseteq [n]$, *where* $|T| \leq t$, *the joint view* $\mathsf{View}_T(x, w_1, \dots, w_n)$ *of players in* T *is distributed identically to* $\textsc{Sim}(T, x, (w_i)_{i \in T}, f_T(x, w_1, \dots, w_n))$. *The relaxations to statistical or computational privacy are defined in the natural way. That is, in the statistical (resp., computational) case we require that for every distinguisher* D *(resp.,* D *with circuit size* $\text{poly}(k)$*) there is a negligible function* $\delta(\cdot)$ *such that*

$$\begin{aligned}&|\Pr[D(\mathsf{View}_T(k, x, w_1, \dots, w_n)) = 1]\\ &\quad - \Pr[D(\textsc{Sim}(k, T, x, (w_i)_{i \in T}, f_T(x, w_1, \dots, w_n))) = 1]| \ \leq \ \delta(k).\end{aligned}$$

In the malicious model, in which corrupted players may behave arbitrarily, security cannot be generally broken into correctness and privacy as above. However, for our purposes we only need the protocols to satisfy a weaker notion of security in the malicious model that is implied by the standard general definition. Specifically, it suffices that Π be t-private as defined above, and moreover it should satisfy the following notion of correctness in the malicious model.

Definition 2.6 (t-robustness). *We say that* Π *realizes* f *with perfect (resp., statistical)* t-robustness *if it is perfectly (resp., statistically) correct in the presence of a semihonest adversary as in Definition* 2.4, *and furthermore for any computationally unbounded malicious adversary corrupting a set* T *of at most* t *players, and for any inputs* $(x, w_1, \dots, w_n)$, *the following robustness property holds. If there is no* $(w'_1, \dots, w'_n)$ *such that* $f(x, w'_1, \dots, w'_n) = 1$, *then the probability that some uncorrupted player outputs* 1 *in an execution of* Π *in which the inputs of the honest players are consistent with* $(x, w_1, \dots, w_n)$ *is* 0 *(resp., is negligible in* k*).*

Finally, we will also consider robustness against an *adaptive* adversary, which is allowed to dynamically pick the set T of corrupted players based on the views of the players it has seen so far (as long as the total number of corrupted players does not exceed t). Definition 2.6 can be naturally extended to the case of adaptive adversaries. (An instance of such a definition will be formally defined later.) By default, we consider the nonadaptive notion of robustness defined above.

3. Zero-knowledge from MPC in the semihonest model.

In this section we present our basic constructions of zero-knowledge protocols, which rely on MPC protocols in the semihonest model.

Let L be a language in NP, and let $R(x, w)$ be a corresponding NP-relation. Let f be the following $(n+1)$-argument function ($n \geq 3$), corresponding to R:

$$f(x, w_1, \dots, w_n) = R(x, w_1 \oplus \cdots \oplus w_n),$$

where $\oplus$ here denotes bitwise exclusive-or of strings (all of the same length). We view f as an n-party functionality, where the first argument x is a public input known to all n players, w_i is a private input of player P_i, and the output is received by all players. We will sometimes ignore the public input x, viewing f as an n-argument function specified by x.

Let Π_f be an n-party protocol which realizes f with *perfect* correctness and either perfect, statistical, or computational 2-privacy (in the semihonest model). We now describe a zero-knowledge protocol Π_R for the NP-relation R. The prover and verifier are both given an input x (an instance of L). The prover is also given a witness w, and they both have a black-box access to the MPC protocol Π_f. The zero-knowledge protocol also makes use of a statistically binding *commitment* protocol Com [44], which can in turn be based on any one-way function [29, 16].

We start by describing and analyzing Π_R using an idealized implementation of Com, where a string can be committed to by secretly sending it to a trusted third party, and later opened (or "decommitted") by having the committer instruct the trusted party to reveal the committed string. (Note that the committer can refuse to open its commitment, but this fact becomes known to the designated receiver of the committed string.) Alternatively, one can think of a direct physical implementation in which the committed string is sent in a locked box, and decommitment is done by sending a key to unlock the box. We refer to a protocol which employs such an ideal commitment primitive as a protocol in the *commitment-hybrid* model.

The implementation of Π_R in the commitment-hybrid model can be viewed as our main contribution. We will later describe the (standard) modifications to the analysis that are required when instantiating the ideal commitment primitive with an actual cryptographic commitment protocol. The protocol is presented in Figure 3.1.

Zero-knowledge protocol $\boldsymbol{\Pi_R}$ in the commitment-hybrid model.

1. The prover picks at random $w_1, \ldots, w_n \in \{0,1\}^m$, whose exclusive-or equals the witness w. She emulates "in her head" the execution of Π_f on input $(x, w_1, \ldots, w_n)$ (this involves choosing randomness for the n players and running the protocol). Based on this execution, the prover prepares the views $V_1, \ldots, V_n$ of the n players; she separately commits to each of these n views.
2. Verifier picks at random distinct player indices $i, j \in [n]$ and sends them to the prover.
3. Prover "opens" the commitments corresponding to the two views V_i, V_j.
4. Verifier accepts if and only if:
   (a) the prover indeed successfully opened the two requested views,
   (b) the outputs of both P_i and P_j (which are determined by their views) are 1, and
   (c) the two opened views are consistent with each other (with respect to x and Π_f; see Definition 2.2).

FIG. 3.1. *Zero-knowledge protocol* Π_R.

THEOREM 3.1. *Let $n \geq 3$ and R, f be as above. Suppose that Π_f realizes the n-party functionality f with* perfect *correctness and either perfect, statistical, or computational* 2*-privacy (in the semihonest model). Then Π_R, described in Figure* 3.1, *is a zero-knowledge proof protocol for the NP-relation R in the commitment-hybrid model, with soundness error $\epsilon \leq 1 - 1/\binom{n}{2}$.*

Proof. We separately argue completeness, soundness, and zero-knowledge.

Completeness. If $(x, w) \in R$ and the prover is honest, then, since $w_1 \oplus \cdots \oplus w_n = w$ and Π_f is perfectly correct, the views $V_1, \ldots, V_n$ always have output 1. Since these views are produced by an honest execution of the protocol, they are always consistent with each other.

Soundness. Suppose that $R(x, w) = 0$ for all w. Hence, by the perfect correctness of Π_f, for all choices of $w_1, \ldots, w_n$ and all choices of random inputs $r_1, \ldots, r_n$, the outputs of all n players in an honest execution of the protocol must be 0. Therefore, considering the n views committed to by the prover in step 1, either in all views the output is 0 or by Lemma 2.3 there exist two views which are inconsistent with respect to x and Π_f. In the former case the verifier always rejects, while in the latter case he rejects with probability at least $1/\binom{n}{2}$, which is his probability of selecting an inconsistent pair i, j.

Zero-knowledge. Let V^* be a malicious verifier. We describe a simulator M^* for the view of V^* which invokes both V^* and the MPC simulator Sim as a black-box. We start with the case where Π_f is *perfectly* 2-private. The simulator M^* on input x proceeds as follows:

1. Run V^* on input x. Let i, j be the pair of indices sent by V^* in step 2 of Π_R.
2. M^* simulates the two views V_i, V_j, received from the (honest) prover in step 3 of Π_R, by picking random w_i, w_j and running $\textsc{Sim}(T = \{i, j\}, x, (w_i, w_j), 1)$.

(Note that the prover's commitments in step 1 are trivial to simulate in the commitment-hybrid model.) We argue that the above simulation is perfect. Fix any (x, w) such that $R(x, w) = 1$. The randomness of V^* chosen by the simulator is distributed identically to the randomness of V^* in the actual execution of Π_R. It thus suffices to prove that the simulation is perfect, conditioned on every choice of such randomness r_{V^*}. Let i, j be the verifier's selections determined by r_{V^*}. In the real execution of Π_R on (x, w), the inputs $w_1, \ldots, w_n$ to Π_f (picked by the honest prover) are such that w_i and w_j are uniform and independent. Thus, conditioned on r_{V^*}, the choice of w_i, w_j made by M^* is distributed identically to the choice of w_i, w_j by the prover in the execution of Π_R on inputs (x, w). It remains to show that, conditioned on each possible choice of r_{V^*}, w_i, w_j, the distribution of the views (V_i, V_j) received from the prover in step 3 of Π_R is identical to the distribution of $\textsc{Sim}((i, j), x, (w_i, w_j), 1)$. Indeed, the fact that Sim is a perfect 2-private simulator for Π_f guarantees that the latter holds even when further conditioning on every possible choice of $w_1, \ldots, w_n$ which is consistent with w_i, w_j, w.

Finally, if Π_f is only statistically or computationally 2-private, the quality of the simulation changes accordingly. (In these cases the simulator receives a security parameter k as an additional input, which it passes to V^* and Sim.) The only change to the previous proof is in the final step: when further conditioning on every choice of $w_1, \ldots, w_n$ which is consistent with w_i, w_j, w, the distribution of (V_i, V_j) in Π_R is now only statistically or computationally close to the output of $\textsc{Sim}(k, (i, j), x, (w_i, w_j), 1)$. By a standard averaging argument, this implies that the final output of M^* is statistically or computationally close to the view of V^*. $\square$

The ideal commitment primitive in the above protocol can be instantiated with any statistically binding commitment protocol; see [16] for a formal definition. Such a commitment protocol can be based (in a black-box way) on an arbitrary one-way function [29, 44].

Let Com be a (perfectly binding, computationally hiding) commitment protocol, and let Π_R^{Com} denote the protocol obtained from Π_R by implementing each invocation of an ideal commitment (or decommitment) in the natural way using a corresponding invocation of Com with security parameter k. Then, we have the following result.

Theorem 3.2. *Let $n \geq 3$ and R, f be as in Theorem 3.1. Suppose that Π_f realizes the n-party functionality f with* perfect *correctness and either perfect, statistical, or computational 2-privacy (in the semihonest model). Let Com be any statistically binding commitment protocol. Then Π_R^{Com}, described above, is a zero-knowledge proof protocol for the NP-relation R with soundness error $\epsilon(k) \leq 1 - 1/\binom{n}{2} + \delta(k)$, where $\delta(\cdot)$ is some negligible function. Furthermore, the protocol obtained from kn^2* sequential *repetitions of* Π_R^{Com} *is a zero-knowledge proof with soundness error* $2^{-\Omega(k)}$.

Note that when Com is implemented using a black-box reduction to a one-way function, the implementation of Π_R^{Com} (as well as its security reduction) makes only a black-box use of Π_f and the one-way function.

Proof. The analysis of Π_R^{Com} closely mimics the analysis of the classical GMW zero-knowledge proof for 3-colorability [20]; we provide details here for self-containment.

The completeness of Π_R^{Com} follows from the completeness of Π_R and the fact that a decommitment made by an honest prover will always be accepted by an honest verifier.

Soundness. The statistical binding property of Com guarantees that each invocation of Com defines a committed string such that, except with negligible probability over the verifier's coins, only this string can later be decommitted. Thus, the soundness error of Π_R^{Com} is bigger than that of Π_R by at most a negligible amount (namely, the probability of the prover breaking the binding of any of the commitments).

Zero-knowledge. Similarly to the GMW protocol, a feature of Π_R^{Com} that is crucial for the proof of its zero-knowledge property is that the verifier has only a polynomial number of choices he can make in the protocol. As such, the simulator M^* works as follows:

1. The simulator makes the following $\ell = n^2k$ "attempts" until it succeeds. If the simulator fails in all these attempts, it aborts.
2. The simulator first chooses a pair of distinct parties $\{i, j\}$ and inputs w_i, w_j at random, and invokes the MPC simulator $\textsc{Sim}(k, (i, j), x, (w_i, w_j), 1)$ to obtain the simulated views V_i and V_j of these parties. For all $h \notin \{i, j\}$, the simulator prepares random views V_h.
3. For each view V_i, the simulator runs the commitment protocol with the verifier algorithm.
4. If the verifier responds with the challenge $\{i, j\}$, then we say that the attempt succeeds; otherwise the attempt fails, and we start over. If the attempt did succeed, then the simulator opens V_i and V_j to the verifier and outputs the view of the verifier resulting from the successful interaction.

Clearly, each simulation attempt succeeds with independent probability at least $1/n^2$. Therefore the simulator succeeds at least once with probability $1 - 2^{-\Omega(k)}$.

The computational indistinguishability of the simulation follows from the following standard hybrid argument. (We refer the reader to [20, 16] for a more detailed exposition of this hybrid argument in the context of the GMW zero-knowledge protocol for 3-colorability.)

$\mathsf{Hybrids}$ $A_1, \dots, A_\ell$. Consider a sequence of hybrid experiments in which the simulator is given the witness. In experiment A_γ, in the first γ "attempts," instead of applying the MPC simulator, it acts as the honest prover would and prepares all views $\{V_i\}$ according to an honest execution of the MPC protocol. However, it then chooses a pair of parties $\{i, j\}$ at random, replaces all V_h (where $h \notin \{i, j\}$) with random views, and continues as the simulator does in the last step. (The remaining $\ell - \gamma$ attempts follow the above simulator.)

Any attacker that can distinguish Hybrid A_1 from the simulator's output would immediately yield a distinguisher for the MPC simulator. Similarly, each A_γ is indistinguishable from $A_{\gamma+1}$ for the same reason.

Hybrid B. Consider now a Hybrid B which is identical to Hybrid A_ℓ, except that now all views $\{V_i\}$ remain as generated from the honest execution of the MPC protocol (as the honest prover would). Recall that in Hybrid A_ℓ, some of these views—the ones that are committed to but never opened—were random views. By indistinguishability of commitments, Hybrid A_ℓ is indistinguishable from Hybrid B.

The only difference between Hybrid B and the real-world interaction of the prover and verifier in Π_R^{Com} is that Hybrid B tries (many times independently) to guess the verifier's choices of i and j ahead of time. By independence, Hybrid B succeeds with probability $1 - 2^{-\Omega(k)}$. Therefore, Hybrid B is within statistical distance $2^{-\Omega(k)}$ from the real-life interaction between the prover and the verifier, and this concludes the proof of the zero-knowledge property.

Since the above simulator makes a black-box use of the malicious verifier, the second part of the theorem is implied by a general sequential composition theorem (cf. [17]). ◻

Additional remarks.

1. If Π_f is not perfectly correct, a cheating prover may violate the soundness of Π_R: on $x \notin L$, she picks $w_1, \ldots, w_n$ and $r_1, \ldots, r_n$, on which the protocol incorrectly outputs 1, thereby making the verifier incorrectly accept (no matter what indices i, j he chooses). We deal with the issue of imperfect correctness in section 3.1 below.
2. Since the above protocol requires the prover to open at most two views, it suffices to share the witness w among 3 of the n players rather than among all of them.
3. We will later be interested in minimizing the amount of communication in the zero-knowledge protocols. It is instructive to (briefly) analyze the communication complexity of this basic construction. The communication in Π_R consists mainly of commitments to the views in Π_f. Implementing a statistically binding commitment to a string m of length ℓ costs $\ell + \text{poly}(k)$ bits of communication, where k is the security parameter. This can be done by first committing to a random seed $s \in \{0,1\}^k$ to a pseudorandom generator G, and then sending $m \oplus G(s)$. Note that the total length of the views in Π_R is equal to $O(|w|)$ plus the communication and randomness complexity of Π_f.
4. The description and analysis of Π_R can easily be extended to accommodate protocols Π_f that use more powerful communication channels. For instance, we can assume that each pair of players has oracle access to an arbitrary 2-party functionality, where both players send inputs to and receive outputs from this functionality. The notion of consistent views can naturally be extended for this case; i.e., the verifier can check that the reported outputs from the oracle are consistent with its inputs. (Note that such oracles do not trivialize the design of Π_f because of the 2-privacy requirement.) In particular, we may assume that each pair of players is connected via an *OT channel* [49, 14], in which the sender has two inputs s_0, s_1 and the receiver a selection bit b, and the receiver gets s_b. It is also easy to extend Π_R to the case where Π_f employs broadcast channels, by simply having the prover send to the verifier all broadcast messages.[6]
5. This protocol, as well as most of the following ones, is actually an Arthur–Merlin protocol (namely, it requires the verifier to send public random coins only to the prover).

[6]A broadcast channel allows a sender to deliver the same message to all players (preventing a malicious sender from sending different messages to different players). This extension is mostly relevant to MPC in the malicious model which is used in section 4. Interestingly, broadcast in our setting is very cheap, while usually in the context of MPC it is considered an expensive means of communication.

Basing zero-knowledge on 1*-private MPC.* We now describe a modified version of the basic zero-knowledge protocol Π_R that can employ any 1-private (rather than 2-private) MPC protocol Π_f. In addition to committing to the n views V_i, the prover will commit to the messages communicated over each of the $\binom{n}{2}$ communication channels. The verifier now challenges the prover to open the view V_i of a random player P_i together with all $n-1$ channels incident to P_i. The zero-knowledge property follows from the fact that the information revealed to the verifier is implied by the view of the single player P_i. The soundness error is at most $1-1/n$: similarly to Lemma 2.3 one can show that any invalid execution of the protocol must include at least one inconsistency between a local view and an incident channel. This soundness error is better than the $1-1/\binom{n}{2}$ error of the basic construction, making the current variant somewhat preferable in terms of efficiency. (We will later show how to obtain more substantial efficiency gains by relying on MPC with stronger robustness properties.) A disadvantage of the current variant over the original protocol Π_R is that it cannot support MPC protocols Π_f which employ ideal OT channels.

Using MPC protocols from the literature. Most standard MPC protocols for the semihonest model can be used in the above transformation. Perhaps the simplest instance is the 2-private 3-player GMW protocol [21, 17], implemented using ideal OT channels between each pair of players. One can also employ the 2-private 5-player BGW protocol [3] (or even the 1-private 3-player BGW protocol using the alternative variant of the basic construction), as well as a somewhat simpler protocol from [42]. Finally, there is a large body of work on efficient special-purpose MPC protocols for specific classes of functions, e.g., linear algebra functions. These protocols can now be used to obtain efficient zero-knowledge protocols for the corresponding classes of relations.

3.1. Coping with statistical correctness. As noted above, our basic construction relies on the MPC protocol's being perfectly correct. We now describe a variant of the basic construction that allows Π_f to be statistically correct.[7] The following modification of Π_R starts with the prover committing to the inputs w_i of the players. Then, the prover and the verifier invoke a coin-flipping procedure whose final outcome, $r_1, \ldots, r_n$, is known only to the prover. Finally, when revealing two views, the verifier is able to verify that the "correct" random inputs were used.

As before, we will mainly focus on describing and analyzing the construction in the commitment-hybrid model, and later describe the modifications required for basing the protocol on a one-way function.

The modified zero-knowledge protocol Π'_R proceeds, in the commitment-hybrid model, as described in Figure 3.2.

THEOREM 3.3. *Let $n \geq 3$ and R, f be as in Theorem* 3.1*. Suppose that Π_f realizes the n-party functionality f with* statistical *correctness and either perfect, statistical, or computational* 2*-privacy (in the semihonest model). Then the protocol Π'_R described in Figure* 3.2 *is a zero-knowledge proof protocol for the NP-relation R in the commitment-hybrid model, with statistical completeness and soundness error $\epsilon \leq 1-1/\binom{n}{2}+\delta(k)$ for some negligible function $\delta(\cdot)$.*

Proof. Completeness, as before, is easy to verify: the probability of the verifier rejecting in a correct interaction with an honest prover is precisely the error probability of Π_f. (Perfect completeness can be easily achieved by letting the prover send w if her

[7]One could also consider a further relaxation to *computational* correctness. However, we cannot make use of this relaxation in the current context of *unconditionally* sound proofs.

Zero-knowledge protocol Π'_R in the commitment-hybrid model.

1. The prover picks at random $w_1, \ldots, w_n \in \{0,1\}^m$ such that $w_1 \oplus \cdots \oplus w_n = w$. She separately commits to each input w_i as well as to n random strings $r_1^P, \ldots, r_n^P$, where each r_i^P is of the same length as the random input of P_i in Π_f.
2. Verifier sends to the prover n random strings $r_1^V, \ldots, r_n^V$, where $|r_i^V| = |r_i^P|$.
3. Prover emulates the execution of Π_f on input $(x, w_1, \ldots, w_n)$, using randomness $r_i = r_i^P \oplus r_i^V$ for each player P_i. Based on this execution, the prover prepares the views $V_1, \ldots, V_n$ of the n players in the protocol and commits to these views.
4. Verifier picks at random distinct $i, j \in [n]$ and sends them to the prover.
5. Prover opens the two views V_i, V_j as well as the corresponding committed inputs w_i, w_j and the shares of the random inputs r_i^P and r_j^P.
6. Verifier accepts if and only if all decommitments are successful, the outputs of P_i and P_j implied by V_i, V_j are 1, the two views are consistent with each other and with the opened inputs w_i, w_j (with respect to x and Π_f), and their random inputs satisfy $r_i = r_i^P \oplus r_i^V$ and $r_j = r_j^P \oplus r_j^V$.

FIG. 3.2. *Zero-knowledge protocol* Π'_R.

invocation of Π_f produces an incorrect output; since this only happens with negligible probability, the zero-knowledge property will not be affected.)

Soundness. Suppose that x is such that $R(x, w) = 0$ for all w. We show that the verifier rejects with at least $1/\binom{n}{2} - \delta(k)$ probability for some negligible $\delta(\cdot)$. Since we are considering soundness, we can assume, without loss of generality, that the prover always finishes the protocol and appropriately opens all of the committed values it is requested to open. (Failing to open a commitment automatically makes the verifier reject.) That is, any prover that occasionally fails to open its commitments or fails to finish the protocol can be trivially converted into a prover that always opens its commitments and finishes the protocol, and has at least as high a probability of getting the verifier to accept. Since the inputs w_i for Π_f as well as the shares of the random inputs r_i^P are committed to in step 1, it is guaranteed that the effective random inputs $r_i = r_i^P \oplus r_i^V$ determined in step 2 are independent of the effective inputs w_i. (Indeed, the verifier selects r_i^V uniformly and independently at random once the prover is already committed to r_i^P.) We can now distinguish between the following cases:

- The inputs w_i and the random inputs r_i lead some player in an honest execution of Π_f to an (incorrect) output of 1. By the statistical correctness of Π_f and the independence of r_i from w_i, this event occurs with negligible probability.
- Otherwise (namely, w_i, r_i lead all players to output 0); we distinguish between two subcases:
  - The views V_i committed to in step 3 are obtained by honestly running Π_f with the inputs w_i and the random inputs r_i. In this case, all views V_i imply an output of 0, and the verifier always rejects.
  - Otherwise, either the input in some view V_i is different from the string w_i determined in step 1, or the random input in some V_i is different from the r_i determined in step 2, or there is a pair of views V_i, V_j that

are not consistent with respect to x and Π_f. (If neither of these types of inconsistencies occur, then we must be in the first subcase.) Either way, an inconsistency will be detected with at least $1/\binom{n}{2}$ probability.

Overall, the verifier's rejection probability is at least $1/\binom{n}{2} - \delta(k)$, where δ is the error probability of Π_f. Note that the statistical correctness of Π_f not only relies on the random inputs r_i being uniformly distributed, but also assumes that they are chosen independently of the inputs x, w_i. Hence, it is essential that the prover first commit to $w_1, \ldots, w_n$ (as done in step 1).

Zero-knowledge. The zero-knowledge property is proved similarly to the basic case. Intuitively, the hiding property of the prover's commitments guarantees that the verifier cannot influence the randomness used when invoking Π_f, nor can it learn anything about the random inputs of players other than P_i, P_j. These views, as before, can be simulated by using the simulator SIM guaranteed by the 2-privacy of Π_f.

The simulator M^* of a dishonest verifier V^* proceeds on input (k, x) as follows:

1. Run V^* on input (k, x). Let $r_1^{V^*}, \ldots, r_n^{V^*}$ be the strings sent by V^* in step 2, and i, j be the pair of indices sent in step 4. (As before, the prover's commitments are trivial to simulate in the commitment-hybrid model.)
2. Pick random w_i, w_j and run $\text{SIM}(k, \{i, j\}, x, (w_i, w_j), 1)$ to simulate the pair of views (V_i, V_j). Let r_i, r_j be the pair of random inputs contained in the simulated views.
3. Use the above w_i, w_j, V_i, V_j and $r_i^P = r_i \oplus r_i^{V^*}$, $r_j^P = r_j \oplus r_j^{V^*}$ to simulate the prover's decommitments in step 5.

The correctness of M^* is argued similarly to the previous case of a perfectly correct Π_f. The only difference is that now the random inputs r_i used by the prover for running Π_f are generated jointly with V^*. But since the prover's contributions r_i^P are independent of the verifier's contributions $r_i^{V^*}$, the views (V_i, V_j) in Π'_R are indistinguishable from the output of $\text{SIM}(k, \{i, j\}, x, (w_i, w_j), 1)$ even when conditioning on the randomness of V^*. ◻

We turn to the question of implementing Π'_R using a statistically binding commitment protocol Com. Unlike the basic construction, here the protocol obtained by naturally substituting Com for every invocation of an ideal commitment is not known to be zero-knowledge. The problem is that the randomness picked by a dishonest verifier in step 2 can depend on the messages received from the prover in step 1. This turns out to be problematic in the context of efficient simulation.

We solve this problem in a standard way, by replacing step 2 by a joint generation of r_i^V using a simulatable coin-flipping protocol involving both the prover and the verifier. For this purpose, one can use a sequential repetition of Blum's coin-flipping protocol [4]. Specifically, to generate a random string r_i^V of length m, the prover and the verifier interact in m phases. In each phase the prover uses Com to commit to a random bit μ_P, the verifier sends a random bit μ_V, and the prover opens μ_P. (If the prover fails to open μ_P, the verifier rejects.) The random bit resulting from this phase is taken to be $\mu_V \oplus \mu_P$.

We describe this modified version, which (with a slight abuse of notation) we denote by Π'^{Com}_R, in Figure 3.3.

THEOREM 3.4. *Let $n \geq 3$ and R, f be as in Theorem 3.1. Suppose that Π_f realizes the n-party functionality f with* statistical *correctness and either perfect, statistical, or computational 2-privacy (in the semihonest model). Let* Com *be any statistically binding commitment protocol. Then Π'^{Com}_R, described in Figure 3.3, is a zero-knowledge proof protocol for the NP-relation R with soundness error $\epsilon(k) \leq 1 - 1/\binom{n}{2} + \delta(k)$,*

Zero-knowledge protocol Π'^{Com}_R in the plain model.

1. The prover picks at random $w_1, \ldots, w_n \in \{0,1\}^m$ such that $w_1 \oplus \cdots \oplus w_n = w$. She separately uses Com to commit to each input w_i as well as to n random strings $r_1^P, \ldots, r_n^P$, where r_i^P is of the same length as the random input of P_i in Π_f. Let $z = |r_1^P| + \cdots + |r_n^P|$.
2. For $\gamma = 1$ to z, the following protocol Π_{coin} ensues, which determines a string μ of length z:
   (a) Prover commits using Com to a randomly chosen bit ψ^P.
   (b) Verifier sends a randomly chosen bit ψ^V to the prover.
   (c) Prover opens her commitment to ψ^P. If the prover fails to open her commitment, the verifier aborts. At this point, we set the γth bit of the string μ, denoted μ_γ, to be $\psi^V \oplus \psi^P$.
3. Through the procedure above, the n strings $r_1^V, \ldots, r_n^V$, where $|r_i^V| = |r_i^P|$, are defined so that the γth bit of the string $(r_1^V \circ \cdots \circ r_n^V)$ is equal to μ_γ.
4. Prover emulates the execution of Π_f on input $(x, w_1, \ldots, w_n)$, using randomness $r_i = r_i^P \oplus r_i^V$ for each player P_i. Based on this execution, the prover prepares the views $V_1, \ldots, V_n$ of the n players in the protocol and uses Com to separately commit to each view.
5. Verifier picks at random distinct $i, j \in [n]$ and sends them to the prover.
6. Prover opens the two views V_i, V_j as well as the corresponding committed inputs w_i, w_j and the shares of the random inputs r_i^P and r_j^P.
7. Verifier accepts if and only if all decommitments are successful, the outputs of P_i and P_j implied by V_i, V_j are 1, the two views are consistent with each other and with the opened inputs w_i, w_j (with respect to x and Π_f), and their random inputs satisfy $r_i = r_i^P \oplus r_i^V$ and $r_j = r_j^P \oplus r_j^V$.

FIG. 3.3. *Zero-knowledge protocol* Π'^{Com}_R.

where $\delta(\cdot)$ is some negligible function. Furthermore, the protocol obtained from kn^2 sequential *repetitions of Π'^{Com}_R is a zero-knowledge proof with soundness error $2^{-\Omega(k)}$.*

Proof. Completeness follows exactly as in protocol Π'_R. Soundness also follows almost exactly as in protocol Π'_R, the only difference being that the uniformity and independence of the r_i values is guaranteed by the honest verifier's choice of μ_i^V in the coin-flipping subprotocol.

Zero-knowledge. For arguing zero-knowledge, we will make use of the following standard lemma regarding the coin-flipping protocol Π_{coin} carried out in step 2 of our protocol (see, e.g., [17, Chapter 7]).

LEMMA 3.5 (coin-flipping lemma). *In protocol Π_{coin}, there exists a polynomial-time oracle machine S_{coin} such that for any polynomial-time adversarial verifier V^*, we have that*

1. *the output of $S_{coin}^{V^*}(k, b)$, where b is a bit chosen uniformly at random, is computationally indistinguishable from real interactions between V^* and the honest prover, and*
2. *the outputs of $S_{coin}^{V^*}(k, b)$, for any bit b, in any nonaborting simulated interaction have the property that $\psi^P \oplus \psi^V = b$.*

We can now describe the simulation. The simulator M^* of a dishonest verifier V^* proceeds on input (k, x) as follows:

1. The simulator makes the following $\ell = n^2k$ "attempts" until it succeeds. If the simulator fails in all these attempts, it aborts.
2. The simulator first chooses a pair of distinct parties $\{i, j\}$ and inputs w_i, w_j at random, and invokes the MPC simulator $\textsc{Sim}(k, (i, j), x, (w_i, w_j), 1)$ to obtain the simulated views V_i and V_j of these parties. Let r_i, r_j be the pair of random inputs contained in the simulated views. For all $h \notin \{i, j\}$, the simulator prepares random views V_h.
3. The simulator acts as the honest prover would in step 1.
4. For step 2, the simulator invokes S_{coin} guaranteed by Lemma 3.5 repeatedly and uses it to guarantee that $r_i^P = r_i \oplus r_i^{V^*}$, $r_j^P = r_j \oplus r_j^{V^*}$, by making a suitable requirement on each of the $\psi^P \oplus \psi^V$.
5. For each view V_i, the simulator runs the commitment protocol with the verifier algorithm.
6. If the verifier responds with the challenge $\{i, j\}$, then we say that the attempt succeeds; otherwise the attempt fails, and we start over. If the attempt did succeed, then the simulator opens V_i and V_j to the verifier and outputs the view of the verifier resulting from the successful interaction.

The correctness of this simulator follows the same argument as the correctness of the simulator of the basic protocol (using the same sequence of hybrids as in the proof of Theorem 3.2), additionally using the simulator guarantee provided by Lemma 3.5. ☐

3.2. Application: Approaching the witness length. In this section, we present zero-knowledge protocols whose communication complexity is $m \cdot \text{poly}(k) \cdot \text{polylog}(s)$, where m is the witness length, k is a cryptographic security parameter, and s is the size of the witness verification circuit $C_x(w)$ verifying $R(x, \cdot)$. This holds whenever C_x has constant depth.

The zero-knowledge protocols rely on an MPC protocol for constant-depth circuits from [1]. The basic version of this protocol combines the polynomial representation techniques of Razborov [51] and Smolensky [55] with the MPC protocol of BGW [3]. (The protocol from [1] includes some additional optimizations that are not important for our crude complexity analysis.)

FACT 3.6 (see [1]). *Let C be a constant depth circuit (using $\vee, \wedge, \oplus, \neg$ gates with unbounded fan-in) of size s and input v of length m. Then, for $n = \text{polylog}(s)$ and any partition $(v_1, \ldots, v_n)$ of v between n players, there is an n-party MPC protocol Π which computes $C(v_1, \ldots, v_n)$ with statistical correctness and 2-privacy, where the communication complexity and randomness complexity of Π are at most $m \cdot k \cdot \text{polylog}(s)$.*

Combining Fact 3.6 with Theorem 3.4 we get the following.

COROLLARY 3.7. *Suppose that one-way functions exist. Then, for any NP-relation $R(x, w)$ that can be verified by a circuit $C_x(w)$ of size s and constant depth (using $\vee, \wedge, \oplus, \neg$ gates with unbounded fan-in), there exists a zero-knowledge proof protocol with communication complexity $|w| \cdot \text{poly}(k) \cdot \text{polylog}(s)$, where k is a cryptographic security parameter.*

In the appendix we describe a noninteractive version of the above protocol which achieves a similar communication complexity.

4. Zero-knowledge from MPC in the malicious model. In this section, we aim at getting zero-knowledge protocols with negligible soundness error 2^{-k} while avoiding the naive repetition that carries a multiplicative overhead of $\Omega(k)$ in the complexity. We do this by strengthening the security requirement from the underlying

MPC protocol Π_f to *t-security* in the *malicious* model. (This should be contrasted with 2-privacy or 1-privacy in the semihonest model used in the previous section.) In fact, we only need the MPC protocol to be t-private in the semihonest model and (perfectly or statistically) *t-robust* in the malicious model, as defined in Definition 2.6.

We start by assuming that Π_f is perfectly t-robust. We will pick our parameters such that t is a constant multiple of the soundness parameter k and n is a constant multiple of t. The zero-knowledge protocol in this case is identical to the protocol Π_R described in the basic construction (Figure 3.1), except that the verifier asks the prover to open t randomly selected views and checks for their consistency. The protocol $\Pi_{R,t}$ is formally described in Figure 4.1.

Zero-knowledge protocol $\Pi_{R,t}$ in the commitment-hybrid model.

1. Prover picks at random $w_1, \ldots, w_n \in \{0,1\}^m$ whose exclusive-or equals the witness w. She emulates the execution of Π_f on input $(x, w_1, \ldots, w_n)$. Based on this execution, the prover prepares the views $V_1, \ldots, V_n$ of the n players; she separately commits to each of these n views.
2. Verifier picks at random t distinct player indices $i_1, \ldots, i_t \in [n]$ and sends them to the prover.
3. Prover opens the commitments corresponding to the t views $V_{i_1}, \ldots, V_{i_t}$.
4. Verifier accepts if and only if
   (a) the prover successfully opened the t requested views,
   (b) the outputs of all players in these views are 1, and
   (c) for each $1 \leq j, h \leq t$, the views V_{i_j} and V_{i_h} are consistent with each other (with respect to x and Π_f).

FIG. 4.1. *Zero-knowledge protocol* $\Pi_{R,t}$.

THEOREM 4.1. *Suppose that Π_f realizes the n-party functionality f with* perfect *t-robustness (in the malicious model) and perfect, statistical, or computational t-privacy (in the semihonest model), where $t = \Omega(k)$ and $n = ct$ for some constant $c > 1$. Then $\Pi_{R,t}$, described in Figure 4.1, is a zero-knowledge proof protocol for the NP-relation R in the commitment-hybrid model, with soundness error $2^{-\Omega(k)}$.*

Proof. The arguments for completeness and zero-knowledge are as in the proof of Theorem 3.1, and only the soundness requires a new proof. The intuition that underlies the soundness proof is as follows. Consider the views $V_1, \ldots, V_n$ committed to by the prover. If all inconsistencies between these views can be resolved by eliminating at most t views (in other words, if there are $n-t$ views which are consistent with each other), then the protocol execution committed to via $V_1, \ldots, V_n$ could be realized by an adversary corrupting at most t players. (We do not require the t excluded views to be consistent with the other $n-t$ views; the excluded views can be thought of as being reported by the adversary.) By the perfect t-robustness of Π_f, in such an execution on an input $x \notin L$ all uncorrupted players must output 0. This implies that there are at least $n-t$ committed views V_i such that the output of P_i is 0. Since t is a constant fraction of n, the verifier will open at least one of these views (and therefore reject) except with negligible probability in n. On the other hand, if resolving all inconsistencies requires eliminating more than t views, then the t opened views will reveal to the verifier, with overwhelming probability, at least one inconsistency.

To formally argue the soundness, consider the following *inconsistency graph* G, defined based on the n committed views $V_1, \ldots, V_n$. The graph G has n vertices (corresponding to the n views), and there is an edge (i, j) in G if the views V_i, V_j are inconsistent (with respect to Π_f and x). That is, the incoming messages from P_j in the view V_i are different from the outgoing messages to P_i implicit in V_j, or vice versa. We analyze the soundness according to two cases, showing that the verifier rejects any $x \notin L$ with overwhelming probability in both cases.

Case 1. There exists in G a vertex cover set B of size at most t. We argue that in this case, the output in all views V_i, for $i \notin B$, must be 0. (The intuition is that a "small" vertex cover can "explain" all the inconsistencies among the views.) To see this, consider an execution of the protocol Π_f where the adversary corrupts the set of players B and behaves in a way that the views of any player P_j, for $j \notin B$, is V_j. Such an execution is obtained by choosing all the messages from $P_i \in B$ to $P_j \notin B$ as in the view V_j; since B is a vertex cover, every pair of views V_i, V_j with $i, j \in \bar{B}$ is not connected in the graph G and hence consistent. Finally, by the perfect t-robustness of Π_f, such a corruption should not influence the output of the honest players, which must be 0. Hence, if in all views V_j with $j \notin B$ the output is 0, it suffices that the verifier, among his t choices, will pick at least one player not in B. By the choice of parameters, the probability that this does *not* happen is at most $(t/n)^t = 2^{-\Omega(t)} = 2^{-\Omega(k)}$.

Case 2. min-VC$(G) > t$. (Here and in the following we denote by min-VC(G) the size of a minimum vertex cover of G.) We would like to argue that in such a graph, a random choice of a constant fraction of the vertices will hit an edge with overwhelming probability. For this we use the well-known connection between the size of a minimum vertex cover and the size of a maximum matching. (The advantage of considering a matching is that we get independence between the edges.) In more detail, the graph G must have a matching of size $> t/2$ (otherwise, if the maximum matching contains $\leq t/2$ edges, the vertices of this matching form a vertex cover of size $\leq t$). If the verifier picks at least one edge of G, he will reject. The probability that the t vertices (views) that the verifier picks miss all the edges of G is smaller than the probability that he misses all edges of the matching which is again at most $2^{-\Omega(t)} = 2^{-\Omega(k)}$. (Indeed, suppose that the first $t/2$ vertices $i_{j_1}, i_{j_{t/2}}$ do not hit an edge of the matching; then their $t/2$ matching neighbors will have $\Omega(t/n) = \Omega(1)$ probability of being hit by each subsequent vertex i_j.) ∎

The proof of Theorem 4.1 illustrates why we cannot rely on t-privacy alone and need the MPC protocol to be t-robust against a malicious adversary. Indeed, in a t-private protocol it may be the case that a single malicious player P_i causes all honest players to have an incorrect output by simply sending incorrect messages. This corresponds to an inconsistency graph G in which P_i alone forms a vertex cover. In such a case the verifier will detect an inconsistency only if P_i is randomly picked, which happens with constant probability.

We will delay addressing how to realize the commitments in the above protocol until we reach the final protocol of this section (see below).

4.1. Changing the MPC model. A final change that we will need in order to eliminate the soundness amplification overhead of the basic construction is to avoid the secret sharing of w among the n players. Instead, we will use an MPC protocol Π_f for the following modified functionality f. The functionality takes the entire input w from a special player I, called an "input client" (following the terminology of [13]), and outputs $R(x, w)$ to all n players P_i. (The NP statement x, as before, is known to all players, but no player other than I has a private input.) The protocol Π_f can be assumed to be secure in the malicious model against an adversary who may corrupt I

and at most t players P_i. As before, we only need the protocol to satisfy the t-privacy and (perfect) t-robustness requirements from Definitions 2.5 and 2.6, except that t-robustness allows the adversary to corrupt I in addition to at most t players P_i. We note that the MPC protocol from [13] which we will use later is already presented in this "input clients" framework.

The zero-knowledge protocol, denoted $\Pi_{R,I,t}$, proceeds in this case in the same way as the protocol $\Pi_{R,t}$ from Figure 4.1, except for the following changes: (1) instead of secret-sharing w between n players, w is directly used by the prover as an input to I when invoking Π_f; (2) the verifier asks to see the views of t randomly selected players P_i (*excluding* the input client I) and checks that they all output 1 and are consistent with each other. The view of I cannot be opened (and hence need not be committed), as this would reveal w.

THEOREM 4.2. *Let f be the following functionality for an input client I and n players $P_1, \dots, P_n$. Given a public input x and a private input w received from I, the functionality delivers $R(x, w)$ to all players P_i. Suppose that Π_f realizes f with* perfect *t-robustness (in the malicious model) and perfect, statistical, or computational t-privacy (in the semihonest model), where $t = \Omega(k)$ and $n = ct$ for some constant $c > 1$. Then $\Pi_{R,I,t}$, described above, is a zero-knowledge proof protocol for the NP-relation R in the commitment-hybrid model, with soundness error $2^{-\Omega(k)}$.*

Proof. The completeness and zero-knowledge properties are argued in essentially the same way as for $\Pi_{R,t}$. (For the zero-knowledge property, the MPC simulator does not need to feed the simulated players P_i with inputs w_i, since these players now have no private inputs.)

The soundness proof needs to be slightly modified to account for the different MPC network and zero-knowledge protocol. Specifically, the difference from the proof of Theorem 4.1 is that the view of one player, I, is never opened. However, this is already accounted for by the fact that the modified definition of t-robustness allows I along with up to t additional players to be corrupted. For completeness, we sketch the modified argument for soundness in the current, slightly modified MPC model.

Suppose that x is a false statement, and consider the inconsistency graph G (on the n vertices $P_1, \dots, P_n$) defined by the views committed to by a malicious prover. We distinguish between the following two cases:

1. min-VC$(G) \leq t$. Similarly to Case 1 of the previous analysis, the committed views can be explained by an execution of the protocol in which I and at most t players P_i are corrupted. By perfect t-robustness, in such an execution all remaining $n-t$ players should output 0, which will be detected by the verifier except with $2^{-\Omega(k)}$ probability.
2. min-VC$(G) > t$. Similarly to Case 2 of the previous analysis, the graph must have a minimal matching of size bigger than $t/2$. Thus, an inconsistency between a pair of players P_i, P_j will be detected except with $2^{-\Omega(k)}$ probability. ◻

4.2. Coping with statistical robustness. In this section we describe a variant of the previous protocol from section 4.1 that applies to the case where Π_f is only statistically robust. This is motivated by the application to constant-rate zero-knowledge, which is described in section 4.4.

If Π_f is only statistically t-robust, a malicious adversary corrupting I and at most t players P_i can have a negligible probability of cheating, namely making some players output 1 when x is a false statement. Interestingly, in this case the coin-flipping–based modification of the basic protocol presented in section 3.1 does not suffice to achieve the desired level of soundness. This is because the prover, while being committed to random inputs on which she has no control, is allowed to generate the views $V_1, \ldots, V_n$ *after* she is given the random inputs of *all* players. (This should be contrasted with real executions of Π_f, in which the adversary knows the randomness only of corrupted players.) In such a case, even a single malicious player may suffice for making all outputs incorrect with high probability. If this particular player is not picked by the verifier, no inconsistency is revealed, but still the outputs may be incorrect.

To get around this difficulty, it is convenient to separate the random inputs r_i that are used for the privacy of the protocol from those that are used to ensure its robustness. Below we describe a construction that works for protocols that are broken into two phases, Phase 1 and Phase 2, where following Phase 1, the players obtain a public random string r of length ℓ that is used in Phase 2. This can be generalized to protocols that have more than two phases; however, the crucial point is that each string r generated in the middle of the protocol must be unpredictable during previous phases. The robustness of the two-phase protocol should hold also with respect to an *adaptive* adversary that may choose (some of) the t corrupted players after seeing the random string r. More precisely, we make the following adaptive t-robustness requirement.

Definition 4.3 (adaptively t-robust two-phase protocol). *Let Π be a two-phase MPC protocol as above (involving an input client I and n players P_i), and let f be as in Theorem 4.2. We say that Π realizes f with* adaptive statistical t-robustness *if it is statistically correct as in Definition 2.4, and furthermore any computationally unbounded adversary can only win the following game with negligible probability in k. First, the adversary picks a false statement x (such that $R(x, w) = 0$ for* all *w), a set T_1 of at most t corrupted players, and random inputs r_i for all uncorrupted players. Now the adversary runs Phase 1 of Π, arbitrarily controlling I and players in T_1. After Phase 1 terminates, a coin-flipping oracle is invoked, generating a random challenge r. Based on r, the adversary can corrupt at most $t - |T_1|$* additional *players and continues to interact with the honest players during Phase 2 of the protocol. The adversary wins if some player that was never corrupted outputs 1.*

This adaptive robustness property, that will be satisfied by the concrete MPC protocol we will employ in section 4.4, is crucial for the soundness analysis to go through.

The zero-knowledge protocol $\Pi'_{R,I,t}$ obtained from such a two-phase MPC protocol Π_f is described in Figure 4.2. For convenience, we assume here that the prover and the verifier have access not only to an ideal commitment but also to an ideal coin-flipping oracle. However, the latter oracle can be easily implemented in our setting based on the former.

Theorem 4.4. *Let f be as in Theorem 4.2. Suppose that Π_f is a two-phase protocol which realizes f with* adaptive statistical *t-robustness (in the malicious model; see Definition 4.3) and perfect, statistical, or computational t-privacy (in the semi-honest model), where $t = \Omega(k)$ and $n = ct$ for some constant $c > 1$. Then $\Pi'_{R,I,t}$, described in Figure 4.2, is a zero-knowledge proof protocol for the NP-relation R in the commitment and coin-flipping-hybrid model with soundness error $2^{-\Omega(k)} + \delta(k)$, where $\delta(k)$ is the robustness error of Π_f.*

Proof. The completeness and zero-knowledge properties are argued as before. We argue soundness by assigning to any cheating prover's strategy in the zero-knowledge protocol a corresponding adversarial strategy in the MPC protocol that generates the same views. Such a simulation is possible only when the zero-knowledge prover does not create too many inconsistent views; however, when she does, the zero-knowledge

Zero-knowledge protocol $\Pi'_{R,I,t}$ in the commitment and coin-flipping hybrid model.

1. Prover picks a random input r_I for I and a random input r_i for every player P_i. She computes the views of the players up to the end of Phase 1, denoted by $(V_1^1, \dots, V_n^1)$, and commits to all of them. Any messages broadcasted in this phase of the MPC protocol are directly sent from P to V.
2. The prover and the verifier invoke a coin-flipping oracle to generate a random challenge string r of length ℓ.
3. Prover continues to run the protocol in her head, using the string r generated in the previous step, and produces the views $(V_1^2, \dots, V_n^2)$ of Phase 2. The prover commits to these n views. Again, any messages broadcasted in Phase 2 of the MPC protocol are directly sent from P to V.
4. Verifier picks at random distinct $i_1, \dots, i_t \in [n]$ and sends them to the prover.
5. Prover opens the corresponding $2t$ commitments $V_{i_1}^1, \dots, V_{i_t}^1, V_{i_1}^2, \dots, V_{i_t}^2$.
6. Verifier accepts if and only if the prover successfully opened the $2t$ requested views, the opened views are all consistent (given the public values x, r and the broadcast messages sent by the prover), and the output in all these views is 1.

FIG. 4.2. *Zero-knowledge protocol* $\Pi'_{R,I,t}$.

verifier will detect an inconsistency and reject with overwhelming probability.

We now formalize this argument. Fix a false statement x. Let P^* be a (deterministic, computationally unbounded) prover strategy in the zero-knowledge protocol on input x. Denote by $V_{P^*}^1$ the views $V_1^1, \dots, V_n^1$ which P^* commits to in step 1, and by $V_{P^*}^2(r)$ the views $V_1^2, \dots, V_n^2$ which P^* commits to in step 3 in response to the random challenge r. Finally let $G_{P^*}(r)$ be the inconsistency graph defined by the complete set of views $V_{P^*}(r) = (V_{P^*}^1, V_{P^*}^2(r))$.

We define a corresponding (deterministic, computationally unbounded) MPC adversary A^* that will attempt to generate the same views as P^* by corrupting I and at most t players P_i in the MPC protocol. (Recall that we allow the MPC adversary to determine the random input r_i of *all* players, even ones it does not corrupt. However, it cannot predict the random challenge r generated between the two phases.)

The MPC adversary A^* starts by considering the inconsistency graph $G_{P^*}^1$ defined by the partial views $V_{P^*}^1$. Note that this is a subgraph of the final inconsistency graph $G_{P^*}(r)$, which is yet unknown to A^*. If the graph does not have a vertex cover T_1 of size $t/2$, A^* gives up. Otherwise, A^* corrupts all players in T_1. It sets the input w of I and the private random inputs $r_I, r_1, \dots, r_n$ as specified in $V_{P^*}^1$. Now A^* interacts with Phase 1 of the MPC protocol, sending messages on behalf of corrupted players that make the views of all uncorrupted players consistent with $V_{P^*}^1$. (This is possible by the assumption that T_1 forms a vertex cover.) Next, A^* obtains the random challenge r. The r it receives defines the final inconsistency graph $G_{P^*}(r)$. Again, if the graph does not have a vertex cover T_2 of size $t/2$, A^* gives up. Otherwise, it corrupts all uncorrupted players in T_2 and interacts with Phase 2 of the MPC protocol. (Note that overall at most t players were corrupted, so this is indeed an admissible adaptive corruption strategy for A^*.) As before, A^* sends messages on behalf of corrupted players that make the final views of all uncorrupted players consistent with $V_{P^*}(r)$. The final MPC views $V_{A^*}(r)$ are defined as the actual views of uncorrupted players, together with the views appearing in $V_{P^*}(r)$ for corrupted players.

LEMMA 4.5. *For every r, if $G_{P^*}(r)$ has a vertex cover of size at most $t/2$, then $V_{A^*}(r) = V_{P^*}(r)$.*

Proof. Since $G^1_{P^*}$ is a subgraph of $G_{P^*}(r)$, the assumption implies that both T_1 and T_2 as required exist, and thus A^* doesn't give up. The lemma follows by noting that any message sent to an uncorrupted player throughout the two phases of the protocol is consistent with $V_{P^*}(r)$. □

The following lemma is proved similarly to the soundness analysis in the case of perfect robustness.

LEMMA 4.6. *For every r such that $G_{P^*}(r)$ does* not *have a vertex cover of size $t/2$, the zero-knowledge verifier rejects except with probability $2^{-\Omega(k)}$.*

We are now ready to prove the desired $2^{-\Omega(k)} + \delta(k)$ bound on the soundness error of the zero-knowledge protocol. We consider three events that cover all possible choices of r:

1. r makes A^* break the t-robustness of Π_f. By the assumption on Π_f, this event occurs with at most $\delta(k)$ probability.
2. The previous event does not occur, and moreover $G_{P^*}(r)$ has a vertex cover of size at most $t/2$. By Lemma 4.5, there are at least $n-t$ views in $V_{P^*}(r)$ whose output is 0. Thus, conditioned on this event, the verifier will open at least one of these views, and consequently reject, except with $2^{-\Omega(k)}$ probability.
3. $G_{P^*}(r)$ does not have a vertex cover of size at most $t/2$. By Lemma 4.6, conditioned on this event, the verifier accepts with probability at most $2^{-\Omega(k)}$.

Overall, the verifier's acceptance probability is at most $2^{-\Omega(k)} + \delta(k)$, as required.

This concludes the proof of Theorem 4.4. □

We note that the gap between the size $t/2$ of the vertex cover and the MPC corruption threshold t is related to the "on-line" nature of adaptive corruptions. In the end of Phase 1, the MPC adversary cannot predict the set of players that will serve as a minimal vertex cover in the final graph.

4.3. Implementing the commitment and coin-flipping. The proofs of zero-knowledge given for our previous protocols in the plain model all relied on the verifier making one of only a polynomial number of choices for his challenge about which views he will see. Here, the choices that the verifier makes are from a much larger set and therefore cannot simply be guessed by the zero-knowledge simulator. Instead, we employ the standard technique (cf. [18, 2, 53, 48]) of having a "preamble" in which the verifier commits in advance to his choices using a *statistically hiding* commitment scheme $\mathsf{Com}_{\mathsf{SH}}$ so as to preserve soundness. In this preamble the verifier also needs to "prove knowledge" of his choices.

Using the recent result of [27], the statistically hiding commitment scheme $\mathsf{Com}_{\mathsf{SH}}$ can be based on any one-way function. As in previous protocols, we will also make use of a standard statistically binding commitment scheme $\mathsf{Com}_{\mathsf{SB}}$.

The resulting protocol is described in Figure 4.3.

THEOREM 4.7. *Let f be as in Theorem 4.2. Let $\mathsf{Com}_{\mathsf{SH}}$ be a statistically hiding commitment scheme, and $\mathsf{Com}_{\mathsf{SB}}$ be a statistically binding commitment scheme. Suppose that Π_f is a two-phase protocol which realizes f with* adaptive statistical *t-robustness (in the malicious model; see Definition 4.3) and perfect, statistical, or computational t-privacy (in the semihonest model), where $t = \Omega(k)$ and $n = ct$ for*

Zero-knowledge protocol $\Pi'^{\mathsf{Com}}_{R,I,t}$ in the plain model.

1. Verifier picks at random distinct $i_1, \ldots, i_t \in [n]$. He encodes these choices into a string ζ of length $q = O(t \log n)$. The verifier then chooses at random strings $\tau_{1,0}, \ldots, \tau_{k,0}$, each of length q, and sets $\tau_{i,1} = \tau_{i,0} \oplus \zeta$ for each i. The verifier invokes $\mathsf{Com}_{\mathsf{SH}}$ to commit to each of the $2k$ strings $\{\tau_{i,0}, \tau_{i,1}\}$ separately.
2. For $i = 1$ to k, the following protocol ensues:
   (a) The prover sends a random bit b_i.
   (b) The verifier opens his commitment to τ_{i,b_i}.
3. Prover picks a random input r_I for I and a random input r_i for every player P_i. She computes the views of the players up to the end of Phase 1, denoted by $(V_1^1, \ldots, V_n^1)$, and commits to all of them using $\mathsf{Com}_{\mathsf{SB}}$. Any messages broadcast in this phase of the MPC protocol are directly sent from P to V.
4. The prover and the verifier invoke the following coin-flipping protocol Π_{coin} to generate a random challenge string r of length ℓ: For $i = 1$ to ℓ,
   (a) Prover commits using $\mathsf{Com}_{\mathsf{SB}}$ to a randomly chosen bit ψ^P.
   (b) Verifier sends a randomly chosen bit ψ^V to the prover.
   (c) Prover opens her commitment to ψ^P. At this point, we define the ith bit of r, denoted r_i, to be $\psi^V \oplus \psi^P$.
5. Prover continues to run the protocol in her head, using the string r generated above, and produces a vector $(V_1^2, \ldots, V_n^2)$ of the views of Phase 2. The prover commits using $\mathsf{Com}_{\mathsf{SB}}$ to these n views. Again, any messages broadcasted in Phase 2 of the MPC protocol are directly sent from P to V.
6. Verifier opens all commitments made using $\mathsf{Com}_{\mathsf{SH}}$ made in step 1. The prover checks consistency (aborting if there is an inconsistency) and extracts the set $i_1, \ldots, i_t \in [n]$.
7. Prover opens the corresponding $2t$ commitments $V_{i_1}^1, \ldots, V_{i_t}^1, V_{i_1}^2, \ldots, V_{i_t}^2$.
8. Verifier accepts if and only if the prover successfully opened the $2t$ requested commitments, the opened views are all consistent (given the public values x, r and the broadcast messages sent by the prover), and the output in all these views is 1.

FIG. 4.3. *Zero-knowledge protocol* $\Pi'^{\mathsf{Com}}_{R,I,t}$.

some constant $c > 1$. Then $\Pi'^{\mathsf{Com}}_{R,I,t}$, described in Figure 4.3, is a zero-knowledge proof protocol for the NP-relation R with negligible soundness error.

Proof. We note that the proof of completeness and soundness is essentially the same as in the proof of Theorem 4.4, since steps 1 and 2 keep the verifier's selections statistically hidden.

We need only provide a proof of the zero-knowledge property. The simulator M^* of a dishonest (and, without loss of generality, deterministic) verifier V^* on input (k, x) is described below. The correctness of the simulation is straightforward.

1. The simulator acts as the honest prover in step 1 and receives the verifier's commitments according to the $\mathsf{Com}_{\mathsf{SH}}$ protocol.
2. In step 2, for each i, the simulator proceeds as follows:
   (a) The simulator saves the state σ_i of the verifier, then picks b_i at random

just as the honest prover would, sends it, and obtains the response from the verifier.

(b) If the decommitment response is invalid, the simulator halts (just as the honest prover would have done) and outputs the partial view of the verifier so far. If the response is valid, then the simulator saves the current state σ_i' and rewinds the verifier to state σ_i. It then sends $1 - b_i$ instead.

(c) If the verifier responds with a valid decommitment, then the simulator has obtained both $\tau_{i,0}$ and $\tau_{i,1}$ and can thus reconstruct the verifier's choices of $i_1, \ldots, i_t \in [n]$. Regardless of whether it received a valid response, the simulator rewinds the state of the verifier back to σ_i' and continues.

3. If the simulator was never able to obtain both $\tau_{i,0}$ and $\tau_{i,1}$ for any i, then the simulator aborts and outputs $\perp$. Note that if the simulator aborts, then the probability of the verifier completing step 2 of the protocol conditioned on its step 1 messages is 2^{-k}, and thus the probability of the simulator aborting is negligible.[8] Otherwise, the simulator continues.
4. The simulator now invokes the MPC simulator Sim, providing Sim with the list of indices $\{i_j\}$ extracted from the verifier above. In turn, Sim provides the simulator with the simulated views $\{V_{i_j}\}$ of these parties, together with any broadcasted messages during the MPC protocol and the random challenge r. For all $h \notin \{i_j\}$, the simulator prepares random views V_h. Each of these views V_i is split into two parts V_i^1 and V_i^2 corresponding to the view in Phase 1 and Phase 2, respectively, of the MPC protocol.
5. For step 3, the simulator commits to all the Phase 1 views $(V_1^1, \ldots, V_n^1)$ and sends all the broadcasted messages of Phase 1.
6. For step 4, the simulator invokes S_{coin} as guaranteed by the coin-flipping lemma repeatedly, and uses it to guarantee that each bit of r is set to be the random challenge specified by the MPC simulator above.
7. For step 5, the simulator commits to all the Phase 2 views $(V_1^2, \ldots, V_n^2)$ and sends all the broadcasted messages of Phase 2.
8. For step 6, the simulator acts as the honest prover would and verifies that all commitments that are opened by the verifier are opened in a manner consistent with the extraction above. If not, then the simulator aborts. Note that if this kind of simulator abort happens with noticeable probability, it can directly be used to break[9] the computational binding property of the commitment scheme $\mathsf{Com}_{\mathsf{SH}}$.
9. For step 7, the simulator opens the $2t$ commitments $\{V_{i_j}^1, V_{i_j}^2\}$ corresponding to the set $\{i_j\}$ of indices requested by the verifier, and outputs the view of the verifier resulting from the successful interaction. $\square$

4.4. Application: "Constant-rate" zero-knowledge proofs. In this section, we describe the construction of zero-knowledge protocols whose communication complexity is linear in the circuit size s, by applying the general construction from

[8]Here we make use of the fact that, without loss of generality, we may assume that $\mathsf{Com}_{\mathsf{SH}}$ has a noninteractive decommitment phase, and therefore the verifier's actions are completely deterministic based only on the sequence of query bits b_i received from the prover. For more details, see [17, Chapter 4], for similar arguments regarding "strong proofs of knowledge."

[9]For more details, see [17, Chapter 4], for similar arguments that arise in the (slightly more complicated) context of "constant-round zero-knowledge proofs."

section 4.2 on top of a variant of the MPC protocol from [13].

The protocol from [13] conforms to the requirements of a two-phase protocol, as defined in section 4.2, with $t = \Omega(n)$ and with a random challenge r of size $\ell = \mathrm{poly}(k, \log s)$. Thus, the general construction from section 4.2 can be applied. In fact, Phase 2 can take the degenerate form of only involving broadcast messages. Thus, in step 3 of the general construction the prover can send these messages to the verifier instead of committing.

The communication complexity of the MPC protocol from [13] in the case where the inputs originate from a constant number of clients is $\mathrm{poly}(k) \cdot \log n \cdot s + \mathrm{poly}(k, n, \log s)$.[10] We now describe how to reduce this complexity to $O(s) + \mathrm{poly}(k, n, \log s)$ for the type of functionalities f required by our application.

The source of the multiplicative $\mathrm{poly}(k)$ overhead in [13] is the need to deal with circuits of an arbitrary depth. However, in the context of proof verification, one may assume without loss of generality that the witness is of size s (and includes the intermediate values in the computation of $C(w)$ in addition to w itself) and that R has s output bits, each verifying the consistency of the output of one gate of C with its two inputs. (An output of 1^s of the augmented R is interpreted as an output of 1 of the original R; every other output of the augmented R is interpreted as 0.)

Note that each of the s output bits of R depends on only three of the s input bits. In this setting the total communication complexity of the protocol from [13] is only $O(\log n \cdot s) + \mathrm{poly}(k, n, \log s)$. Here the $O(\log n)$ multiplicative overhead results from the use of Shamir's secret-sharing [54] (more precisely, the variant of this secret-sharing due to Franklin and Yung [15]), which requires the field size to be larger than the number of players. As a final optimization, one can implement the secret-sharing of [15] over fields of constant size by basing the secret-sharing on algebraic-geometric codes [9]. This results in an additional fractional decrease in the security threshold, which does not affect the asymptotic complexity of the zero-knowledge protocol.

The following lemma summarizes the properties of the MPC protocol we use to obtain constant-rate zero-knowledge protocols.

LEMMA 4.8 (see [13, 9]). *Let f be a functionality which takes its input w from an input client I and delivers its output $f(w)$ to n players $P_1, \dots, P_n$. Suppose that f can be computed by a circuit of size s with bounded fan-in and constant depth. Then, f can be realized by a perfectly t-private and adaptively statistically t-robust two-phase MPC protocol Π_f with $t = \Omega(n)$ and the following efficiency features:*

- *Phase 1 involves a total of $O(s) + \mathrm{poly}(k, n, \log s)$ bits sent from I to the n players P_i.*
- *The random challenge r between the two phases is of length $\mathrm{poly}(k, n, \log s)$.*
- *Phase 2 involves broadcast messages whose total length is $O(s) + \mathrm{poly}(k, n, \log s)$.*

Combining Lemma 4.8 with the general transformation of Theorem 4.7,

COROLLARY 4.9. *Suppose that one-way functions exist. Then, for any NP-relation $R(x, w)$ that can be verified by a circuit $C_x(w)$ of size s (using gates of bounded fan-in), there exists a zero-knowledge proof protocol with communication complexity $O(s) + \mathrm{poly}(k, \log s)$, where k is a cryptographic security parameter.*

Proof. Combining Lemma 4.8 and Theorem 4.4 directly yields a protocol as required in the commitment-hybrid model. To derive the desired result in the plain model from Theorem 4.7, we need to base *efficient* instantiations of the commit-

[10]Here we count each broadcasted bit as a single bit of communication. This allows the protocol to deliver the output to all n players with no asymptotic efficiency overhead.

ment protocols invoked by $\Pi'^{\mathsf{Com}}_{R,I,t}$ on an arbitrary one-way function. The statistically hiding commitment protocol $\mathsf{Com}_{\mathsf{SH}}$ is applied only to strings of total length $\text{poly}(k, \log s)$ and therefore does not form an efficiency bottleneck. It thus suffices to present a statistically binding commitment protocol Com in which commitment and decommitment on a message of length ℓ can be done with communication complexity $O(\ell) + \text{poly}(k)$. Such a commitment protocol Com can be obtained in a standard way from an arbitrary statistically-binding commitment protocol Com' and a pseudorandom generator G (hence also from any one-way function [29, 44]): to commit to a message m, the sender in Com applies Com' to a random seed $r \in \{0,1\}^k$ and then sends $m \oplus G(r)$. $\square$

5. Concluding remarks.

This work establishes a new general connection between zero-knowledge proofs and MPC. In particular, we have demonstrated that MPC protocols for the case of an honest majority can serve as a useful building block in the design of efficient and conceptually simple zero-knowledge proofs, including the first construction of "constant-rate" zero-knowledge proofs for all of NP. The idea of making a general use of MPC with honest majority in the context of secure two-party computation and MPC with no honest majority is likely to find additional applications beyond those considered in this work. Evidence in this direction was recently given in [28, 33].

We conclude by noting that while MPC is a conceptually attractive and very well studied notion, it is not necessarily the most general abstraction of the combinatorial object required by our zero-knowledge protocols. One may view the MPC component in these protocols as implementing a zero-knowledge variant of probabilistically checkable proofs (PCPs) over large alphabets. Other kinds of proof systems—interactive PCPs [35] and interactive proofs with quasi-linear-time verifiers [23]—were recently exploited to construct efficient zero-knowledge protocols, and particularly to improve over the results of section 3.2. It would be interesting to obtain a better understanding of the relations between these proof systems, zero-knowledge proofs, and MPC.

Appendix. Approaching the witness length in the noninteractive setting.

In this section we describe a realization of the protocol from section 3.2 in a model of NIZK with preprocessing, previously considered in [34] and [40].

In the description below, it is assumed that the preprocessing phase is run by a trusted dealer who sends randomized messages to the prover and the verifier (before the input x is available) and then disappears.

A first idea for obtaining a noninteractive protocol is to replace the (interactive) commit-and-challenge approach by the following implementation of noninteractive 2-out-of-n OT. In the preprocessing step, the dealer sends to the prover n random encryption keys and to the verifier a random subset of two of these keys (whose identity is unknown to the prover). In the online phase, the prover generates n views as in the interactive protocol and sends the encryption of each view V_i using the corresponding key. The verifier learns a random pair of views and, as in the basic protocol, verifies their consistency. (This can be repeated in parallel to amplify soundness.)

The statistical correctness of the underlying MPC protocol Π poses an additional difficulty. Picking the randomness to be used in the protocol Π during the preprocessing step is problematic because the prover may choose for some $x \notin L$ its input w *after* getting to see this randomness. Here, however, we can use a simple modification of [1] that gives a family of protocols Π_ρ, $\rho \in \{0,1\}^{mk\cdot\text{polylog}(s)}$, such that each Π_ρ has roughly the same complexity as Π, and a random protocol from this family is perfectly correct and private except with 2^{-k} probability over the choice of ρ. The protocol can now proceed as in the perfect case, except that a random ρ is given to both the prover and the verifier during the preprocessing step, and ρ is used to define a protocol Π_ρ to be used in the proof. (If the choice of x can depend on ρ, the length of ρ should be increased roughly by $|x|$.)

FACT A.1 (modified from [1]). *Let C be a constant-depth circuit, as in Fact 3.6. Then there exists a family of $(2,n)$-secure MPC protocols $\{\Pi_\rho\}_{\rho\in\{0,1\}^{mk\cdot\mathrm{polylog}(s)}}$, with parameters n,m as in Fact 3.6 and communication complexity and randomness complexity of $m\cdot\mathrm{polylog}(s)$. Moreover, with overwhelming probability over the choice of ρ, the protocol Π_ρ computes $C(\cdot)$ with perfect correctness and privacy.*

(Using the terminology and notation of [1], we need to get an RPC $p_\rho(x,r)$ with public randomness ρ of size $mk\cdot\mathrm{polylog}(s)$ that, with overwhelming probability over the choice of ρ, is perfectly correct. This can be achieved by repeating the basic RPC construction from [1] $O(mk)$ times, reducing the error probability on any fixed input $w\in\{0,1\}^m$ to 2^{-km}, and then applying efficient degree-3 randomizing polynomials for a threshold function on the output. By a union bound argument, an overwhelming fraction of the ρ will be good for *all* inputs w.)

Given such a family Π_ρ, the zero-knowledge protocol Π_R in the noninteractive model proceeds as follows:

1. **(Preprocessing phase)** The trusted dealer chooses a random protocol index ρ and sends it to both the prover and the verifier. It also picks n random symmetric encryption keys $K_1,\dots,K_n$ that it sends to the prover, and it picks at random two distinct indices $i,j\in[n]$ and sends to the verifier (i,j,K_i,K_j).
2. **(Proof)** The prover picks at random $w_1,\dots,w_n\in\{0,1\}^m$ such that $w_1\oplus\cdots\oplus w_n=w$, as well as randomness $r_1,\dots,r_n$ for Π_ρ. She emulates the execution of Π_ρ on inputs $(x,w_1,\dots,w_n)$ and random inputs $r_1,\dots,r_n$. Based on this execution, the prover prepares the views $V_1,\dots,V_n$ of the n players in Π_ρ. She separately encrypts each view V_i using the key K_i given to it in the preprocessing phase, and sends $e_1=\textsc{Encrypt}(V_1,K_1),\dots,e_n=\textsc{Encrypt}(V_n,K_n)$ to the verifier.
3. **(Verification)** Verifier decrypts V_i,V_j using e_i,e_j and the keys K_i,K_j he got in the preprocessing phase, and accepts if and only if the output of both P_i and P_j (as follows from their views) is 1 and the two views are consistent (according to Π_ρ).

Due to its similarity to our previous constructions, we only sketch the properties of the protocol. The completeness of the protocol is easily verifiable. For the soundness, since we assume that the preprocessing phase is carried out by a trusted party, then i,j are completely hidden from the prover. If the randomly chosen protocol Π_ρ is not perfectly correct, then protocol Π_R provides no guarantees; this, however, happens with negligible probability. Otherwise (i.e., Π_ρ is perfectly correct), if the views are not all consistent, then this is detected with probability at least $1/n^2$, and if the views are consistent, then, by the perfect correctness, the output is always correct (i.e., 0). Regarding zero-knowledge, by the semantic security of $\textsc{Encrypt}$ the verifier cannot distinguish between the message actually sent by the prover and a message which contains an encryption of the actual views V_i,V_j for which it has the encryption keys along with encryptions of random values for the remaining $n-2$ views. Thus, this message can be simulated by using the MPC simulator to generate the views V_i,V_j.

The soundness of the above protocol relies on the assumption that the input x is picked independently of ρ. (Otherwise it is not guaranteed that a random Π_ρ will be perfectly correct with high probability.) An adaptive choice of x can be handled by increasing the length of ρ (roughly by $|x|$ or, more generally, the description size of x).

Acknowledgments. We thank Amos Fiat and Yishay Mansour for suggesting the variant of the basic construction that uses 1-private MPC. We also thank Salil Vadhan and the anonymous referees for helpful comments and suggestions.

REFERENCES

[1] O. Barkol and Y. Ishai, *Secure computation of constant-depth circuits with applications to database search problems*, in Proceedings of the 25th Annual International Cryptology Conference (CRYPTO 2005), Santa Barbara, CA, Springer-Verlag, Berlin, 2005, pp. 395–411.

[2] M. Bellare, S. Micali, and R. Ostrovsky, *The (true) complexity of statistical zero knowledge*, in Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing (STOC), Baltimore, MD, ACM, New York, 1990, pp. 494–502.

[3] M. Ben-Or, S. Goldwasser, and A. Wigderson, *Completeness theorems for non-cryptographic fault-tolerant distributed computation*, in Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC), Baltimore, MD, ACM, New York, 1988, pp. 1–10.

[4] M. Blum, *Coin flipping by telephone - a protocol for solving impossible problems*, in Proceedings of the 24th IEEE Computer Society International Conference COMPCON 1982, San Francisco, CA, IEEE Computer Society, Piscataway, NJ, 1982, pp. 133–137.

[5] J. Boyar, G. Brassard, and R. Peralta, *Subquadratic zero-knowledge*, J. ACM, 42 (1995), pp. 1169–1193.

[6] J. Boyar, I. Damgård, and R. Peralta, *Short non-interactive cryptographic proofs*, J. Cryptology, 13 (2000), pp. 449–472.

[7] R. Canetti, *Security and composition of multiparty cryptographic protocols*, J. Cryptology, 13 (2000).

[8] D. Chaum, C. Crépeau, and I. Damgård, *Multiparty unconditionally secure protocols (extended abstract)*, in Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC), Baltimore, MD, 1988, ACM, New York, 1988, pp. 11–19.

[9] H. Chen and R. Cramer, *Algebraic geometric secret sharing schemes and secure multi-party computations over small fields*, in Proceedings of the 26th Annual International Cryptology Conference (CRYPTO 2006), Santa Barbara, CA, Springer-Verlag, Berlin, 2006, pp. 521–536.

[10] R. Cramer and I. Damgård, *Linear zero-knowledge—A note on efficient zero-knowledge proofs and arguments*, in Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC), El Paso, TX, ACM, New York, 1997, pp. 436–445.

[11] R. Cramer and I. Damgård, *Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free?*, in Advances in Cryptology (CRYPTO '98), Lecture Notes in Comput. SCi. 1462, Springer, New York, 1998, pp. 424–441.

[12] I. Damgård and Y. Ishai, *Constant-round multiparty computation using a black-box pseudorandom generator*, in Proceedings of the 25th Annual International Cryptology Conference (CRYPTO 2005), Santa Barbara, CA, Springer-Verlag, Berlin, 2005, pp. 378–394.

[13] I. Damgård and Y. Ishai, *Scalable secure multiparty computation*, in Proceedings of the 26th Annual International Cryptology Conference (CRYPTO 2006), Santa Barbara, CA, Springer-Verlag, Berlin, 2006, pp. 501–520.

[14] S. Even, O. Goldreich, and A. Lempel, *A randomized protocol for signing contracts*, in Comm. ACM, 28 (1985), pp. 637–647.

[15] M. K. Franklin and M. Yung, *Communication complexity of secure computation*, in Proceedings of the 24th Annual ACM Symposium on the Theory of Computing (STOC), Victoria, BC, Canada, ACM, New York, 1992, pp. 699–710.

[16] O. Goldreich, *Foundations of Cryptography: Basic Tools*, Cambridge University Press, Cambridge, UK, 2001.

[17] O. Goldreich, *Foundations of Cryptography: Basic Applications*, Cambridge University Press, Cambridge, UK, 2004.

[18] O. Goldreich and A. Kahan, *How to construct constant-round zero-knowledge proof systems for NP*, J. Cryptology, 9 (1996), pp. 167–190.

[19] O. Goldreich and J. Håstad, *On the complexity of interactive proofs with bounded communication*, Inform. Process. Lett., 67 (1998), pp. 205–214.

[20] O. Goldreich, S. Micali, and A. Wigderson, *How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design*, in Advances in Cryptology (CRYPTO '86), Lecture Notes in Comput. Sci. 263, Springer, New York, 1987, pp. 171–185.

[21] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game (extended abstract)*, in Proceedings of the 19th Annual ACM Symposium on the Theory of Computing (STOC), New York, NY, ACM, New York, 1987, pp. 218–229.

[22] O. Goldreich and Y. Oren, *Definitions and properties of zero-knowledge proof systems*, J. Cryptology, 7 (1994), pp. 1–32.

[23] S. Goldwasser, Y. T. Kalai, and G. Rothblum, *Delegating computation: Interactive proofs for muggles*, in Proceedings of the 40th Annual ACM Symposium on the Theory of Computing (STOC), Victoria, BC, Canada, ACM, New York, 2008, pp. 113–122.

[24] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems*, SIAM J. Comput., 18 (1989), pp. 186–208.

[25] J. Groth, R. Ostrovsky, and A. Sahai, *Perfect non-interactive zero knowledge for NP*, in Proceedings of the 25th International Cryptology Conference (EUROCRYPT 2006), Saint Petersburg, Russia, 2006, Lecture Notes in Comput. Sci. 4004, Springer, New York, 2006, pp. 339–358.

[26] I. Haitner, *semihonest to malicious oblivious transfer—The black-box way*, in Proceedings of the 5th IACR Theory of Cryptology Conference (TCC 2008), New York, NY, 2008, pp. 412–426.

[27] I. Haitner and O. Reingold, *Statistically-hiding commitment from any one-way function*, in Proceedings of the 39th Annual ACM Symposium on the Theory of Computing (STOC), San Diego, CA, ACM, New York, 2007, pp. 1–10.

[28] D. Harnik, Y. Ishai, E. Kushilevitz, and J. B. Nielsen, *OT-combiners from secure computation*, in Proceedings of the 5th IACR Theory of Cryptology Conference (TCC 2008), New York, NY, 2008, pp. 393–411.

[29] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, *A pseudorandom generator from any one-way function*, SIAM J. Comput., 28 (1999), pp. 1364–1396.

[30] R. Impagliazzo and S. Rudich, *Limits on the provable consequences of one-way permutations*, in Advances in Cryptology (CRYPTO '88), Lecture Notes in Comput. Sci. 403, Springer, New York, 1990, pp. 8–26.

[31] Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank, *Black-box constructions for secure computation*, in Proceedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC), Seattle, WA, ACM, New York, 2006, pp. 99–108.

[32] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, *Zero-knowledge from secure multiparty computation*, in Proceedings of the 39th Annual ACM Symposium on the Theory of Computing (STOC), San Diego, CA, ACM, New York, 2007, pp. 21–30.

[33] Y. Ishai, M. Prabhakaran, and A. Sahai, *Founding cryptography on oblivious transfer – efficiently*, in Proceedings of the 28th Annual International Cryptology Conference (CRYPTO 2008), Santa Barbara, CA, Springer-Verlag, Berlin, 2008, pp. 572–591.

[34] Y. T. Kalai and R. Raz, *Succinct non-interactive zero-knowledge proofs with preprocessing for LOGSNP*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Berkeley, CA, IEEE Computer Society Press, Piscataway, NJ, 2006, pp. 355–366.

[35] Y. T. Kalai and R. Raz, *Interactive PCP*, in Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP), Reykjank, Iceland, Springer, New York, 2008, pp. 536–547.

[36] J. Kilian, *Founding cryptography on oblivious transfer*, in Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC), ACM, New York, 1988, pp. 20–31.

[37] J. Kilian, *A note on efficient zero-knowledge proofs and arguments (extended abstract)*, in Proceedings of the 24th Annual ACM Symposium on the Theory of Computing (STOC), Victoria, BC, Canada, ACM, New York, 1992, pp. 723–732.

[38] J. Kilian, E. Kushilevitz, S. Micali, and R. Ostrovsky, *Reducibility and completeness in private computations*, SIAM J. Comput., 29 (2000), pp. 1189–1208.

[39] J. Kilian and E. Petrank, *An efficient noninteractive zero-knowledge proof system for NP with general assumptions*, J. Cryptology, 11 (1998), pp. 1–27.

[40] J. Kilian, S. Micali, and R. Ostrovsky, *Minimum resource zero-knowledge proofs*, in Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Research Triangle Park, NC, IEEE Computer Society Press, Piscataway, NJ, pp. 474–479.

[41] E. Kushilevitz, S. Micali, and R. Ostrovsky, *Reducibility and completeness in multi-party private computations*, in Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Santa Fe, NM, IEEE Computer Society Press, Piscataway, NJ, pp. 478–489.

[42] U. Maurer, *Secure multi-party computation made simple*, in Proceedings of the Third Conference on Security in Communication Networks (SCN 2002), Almafi, Italy, Lecture Notes in Comput. Sci. 2576, springer, New York, pp. 14–28.

[43] S. Micali, *Computationally sound proofs*, SIAM J. Comput., 30 (2000), pp. 1253–1298.

[44] M. Naor, *Bit commitment using pseudorandomness*, J. Cryptology, 4 (1991), pp. 151–158.

[45] J. Naor and M. Naor, *Small-bias probability spaces: Efficient constructions and applications*, SIAM J. Comput., 22 (1993), pp. 838–856.

[46] M. Naor and K. Nissim, *Communication preserving protocols for secure function evaluation*, in Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC), Herakilion, Crete, Greece, ACM, New York, 2001, pp. 590–599.

[47] R. Ostrovsky and A. Wigderson, *One-way functions are essential for non-trivial zero-knowledge*, in Proceedings of the Second Israel Symposium on Theory of Computing and Systems (ISTCS), Natanya, Israel, IEEE Computer Society Press, Piscataway, NJ, 1993, pp. 3–17.

[48] M. Prabhakaran, A. Rosen, and A. Sahai, *Concurrent zero-knowledge with logarithmic round complexity*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), Vancouver, BC, Canada, IEEE Computer Society Press, Piscataway, NJ, pp. 366–375.

[49] M. Rabin, *How to Exchange Secrets by Oblivious Transfer*, Technical Memo TR-81, Aiken Computation Laboratory, Harvard University, Cambridge, MA, 1981.

[50] T. Rabin and M. Ben-Or, *Verifiable secret sharing and multiparty protocols with honest majority*, in Proceedings of the 21st Annual ACM Symposium on the Theory of Computing (STOC), Seattle, WA, ACM, New York, 1989, pp. 73–85.

[51] A. Razborov, *Lower bounds for the size of circuits of bounded depth with basis (AND, XOR)*, Math. Notes Acad. Sci. USSR, 41 (1987), pp. 333–338.

[52] O. Reingold, L. Trevisan, and S. P. Vadhan, *Notions of reducibility between cryptographic primitives*, in Proceedings of the 1st Theory of Cryptology (TCC), Cambridge, MA, 2004, pp. 1–20.

[53] A. Rosen, *A note on constant round zero knowledge proofs for NP*, in Proceedings of the 1st Theory of Cryptology (TCC), Cambridge, MA, 2004, pp. 191–202.

[54] A. Shamir, *How to share a secret*, Comm. ACM, 22 (1979), pp. 612–613.

[55] R. Smolensky, *Algebraic methods in the theory of lower bound for Boolean circuit complexity*, in Proceedings of the 19th Annual ACM Symposium on the Theory of Computing (STOC), New York, NY, ACM, New York, 1987, pp. 77–82.

[56] A. C. Yao, *How to generate and exchange secrets*, in Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Toronto, Canada, IEEE Computer Society Press, Piscataway, NJ, 1986, pp. 162–167.