Verifiable Relation Sharing and Multi-Verifier Zero-Knowledge in Two Rounds: Trading NIZKs with Honest Majority (Extended Abstract) *

Benny Applebaum[1][0000−0003−4792−369X], Eliran Kachlon[1][0000−0001−5913−1636], and Arpita Patra[2][0000−0002−8036−4407]

[1] Tel-Aviv University, Tel-Aviv, Israel {benny.applebaum,elirn.chalon}@gmail.com
[2] Indian Institute of Science, Bangalore, India arpita@iisc.ac.in

Abstract. We introduce the problem of *Verifiable Relation Sharing* (VRS) where a client (prover) wishes to share a vector of secret data items among k servers (the verifiers) while proving in zero-knowledge that the shared data satisfies some properties. This combined task of sharing and proving generalizes notions like verifiable secret sharing and zero-knowledge proofs over secret-shared data. We study VRS from a theoretical perspective and focus on its round complexity.

As our main contribution, we show that every efficiently-computable relation can be realized by a VRS with an optimal round complexity of two rounds where the first round is input-independent (offline round). The protocol achieves full UC-security against an active adversary that is allowed to corrupt any t-subset of the parties that may include the client together with some of the verifiers. For a small (logarithmic) number of parties, we achieve an optimal resiliency threshold of $t < 0.5(k + 1)$, and for a large (polynomial) number of parties, we achieve an almost-optimal resiliency threshold of $t < 0.5(k+1)(1-\epsilon)$ for an arbitrarily small constant $\epsilon > 0$. Both protocols can be based on sub-exponentially hard injective one-way functions. If the parties have an access to a collision resistance hash function, we can derive *statistical everlasting security*, i.e., the protocols are secure against adversaries that are computationally bounded during the protocol execution and become computationally unbounded after the protocol execution.

Previous 2-round solutions achieve smaller resiliency thresholds and weaker security notions regardless of the underlying assumptions. As a special case, our protocols give rise to 2-round offline/online constructions of multi-verifier zero-knowledge proofs (MVZK). Such constructions were previously obtained under the same type of assumptions that are needed for NIZK, i.e., public-key assumptions or random-oracle type assumptions (Abe et al., Asiacrypt 2002; Groth and Ostrovsky, Crypto 2007; Boneh et al., Crypto 2019; Yang, and Wang, Eprint 2022). Our work shows, for the first time, that in the presence of an honest majority

* A full version of this paper appears in [6]

these assumptions can be replaced with more conservative "Minicrypt"-type assumptions like injective one-way functions and collision-resistance hash functions. Indeed, our MVZK protocols provide a round-efficient substitute for NIZK in settings where honest-majority is present. Additional applications are also presented.

1 Introduction

In recent years, a large amount of research was dedicated to the study of zero-knowledge proofs in *distributed settings*, such as zero-knowledge proofs with multiple verifiers [37, 51, 9] and zero-knowledge proofs over secret-shared data [16, 25, 17, 24]. Those variants of zero-knowledge proofs have applications both in theory and practice, in round-optimal multiparty computation [2], private data aggregation [24], and anonymous communication [25].

A typical scenario of interest consists of a client $\mathcal{P}$ (the prover) that holds a vector of secret data items $\mathbf{s}$, together with several servers $\mathcal{V}_1, \ldots, \mathcal{V}_k$ (the verifiers). The client wishes to share $\mathbf{s}$ among the servers, and also prove in zero-knowledge that the shared data satisfies some properties. Previous works usually let $\mathcal{P}$ send each $\mathcal{V}_i$ its share, and then perform a zero-knowledge proof on the shared data. A natural question is whether considering the sharing and the proving as a single task could result in a protocol with better round-complexity and better security guarantees. To capture this joint task of sharing-and-proving, we present the notion of *verifiable relation sharing* (VRS).

Verifiable relation sharing. The VRS functionality of a public relation R receives from the prover an input $\mathbf{x} = (x_0, x_1, \ldots, x_k)$, where we think of x_0 as a private information of the prover, and of x_i as the share of $\mathcal{V}_i$. The functionality verifies that $R(\mathbf{x}) = 1$, and if the verification fails, then it returns a failure-symbol $\perp$ to all the verifiers. If the verification succeeds, the functionality returns x_i to $\mathcal{V}_i$. Observe that the VRS functionality captures the typical scenario discussed above, as well as several cryptographic primitives, including verifiable secret sharing [23], verifiable function secret sharing [17], secure multicast [33], and zero-knowledge proofs with multiple verifiers.

We formalize the VRS functionality under the definitions of secure multiparty computation (MPC) in the universal-composability (UC) framework of [21]. We strive for full-security, including guaranteed output delivery, at the presence of an honest majority in the plain model. We note that honest-majority is necessary due to impossibility of UC-secure Zero-knowledge proofs in the plain model [22]. The active (aka Byzantine or malicious) adversary is allowed to corrupt any minority subset of the $k + 1$ parties $\{\mathcal{P}, \mathcal{V}_1, \ldots, \mathcal{V}_k\}$ that may include the prover together with some of the verifiers. The use of MPC-based "full-security" definitions provides strong guarantees that are not supported by related notions of distributed zero-knowledge. Specifically, when the prover $\mathcal{P}$ is honest, we get *correctness*, i.e., every honest $\mathcal{V}_i$ outputs x_i even in the presence of corrupt active verifiers, as well as simulation-based *privacy*, which implies that the adversary only learns the outputs of the corrupt verifiers. For a corrupt $\mathcal{P}$, we

get *soundness* and *knowledge extraction* even when $\mathcal{P}$ colludes with some of the verifiers. In contrast, previous works on weaker notions, such as zero-knowledge proofs over secret-shared data, achieve correctness only for semi-honest verifiers [16, 17, 25, 24], and in some cases (e.g., [24, 25]) provide soundness only when all the verifiers are honest. Further discussion of related works and a comparison of known results appear in Section 1.2 and Table 1.

We study the VRS problem from a theoretical perspective while focusing on the best-achievable *round complexity*. It is known that VRS cannot be realized in 1 round even for relatively simple relations (e.g., VSS [7]). Looking for the second best, we ask:

> **Q1:** Can VRS be realized by a 2-round protocol? Moreover, can we make the first round input-independent ("offline round")? If so, under what assumptions?

The question of obtaining a 2-round protocol in the plain model is open even for weaker notions like distributed zero-knowledge over secret-shared data.

Multi-verifier zero-knowledge. It is useful to consider the somewhat degenerate version of VRS in which all the verifiers get the same information except for some private witness that is kept by $\mathcal{P}$. This variant essentially corresponds to *multi-verifier zero-knowledge* proofs (MVZK) [20]. When modeled as an ideal functionality, MVZK is parameterized by a public relation R, it receives from $\mathcal{P}$ a statement x and a witness w, and verifies that $R(x, w) = 1$. If the verification fails, then the functionality returns a failure-symbol $\perp$ to all the verifiers $\mathcal{V}_1, \ldots, \mathcal{V}_k$, and if the verification succeeds, the functionality returns x to all the verifiers. Again, we strive for a 2-round offline/online solution in the plain model.

Observe that the single verifier case (where the adversary can either corrupt the verifier or the prover) corresponds to the standard notion of zero-knowledge proofs. Classical impossibility results [36] show that a plain-model protocol that consists of a single message from the prover to the verifier, also known as *non-interactive zero-knowledge* (NIZK), exist only for languages in BPP, even when one considers only stand-alone security. Assuming a minimal trusted setup in the form of a common reference string (CRS), one can achieve NIZK for every language in NP from public-key assumptions [15, 30, 38, 50, 13, 48], or, alternatively, in the random oracle model [31, 11]. In a related notion, called *Zaps* [28], the CRS is replaced with a preprocessing round in which only the verifier communicates by broadcasting its random coins, at the expense of downgrading zero-knowledge to witness-indistinguishability. Assuming the existence of one-way functions, it is known that Zaps are equivalent to NIZK [28].

Let us move back to the setting of multiple verifiers. Striving for a 2-round simulation-based zero-knowledge, we make the necessary assumption of an honest majority among the set of all parties (including the prover).[3] To the best

[3] Without an honest majority, a 2-round plain-model MVZK protocol (where in each round both the verifiers and prover can talk simultaneously) implies a 2-step ZK protocol (where the verifier sends a message and gets a response from the prover) which is ruled-out by [36] for non-trivial languages outside BPP.

of our knowledge, the only known solution in this setting follows from the work of Groth and Ostrovsky on Multi-string NIZK Proofs [37]. Specifically, their work implicitly give rise to a 2-round offline/online honest-majority MVZK that achieves simulation-based security based on Zaps and public-key encryption [37, Theorem 3]. These assumptions are as strong (or even stronger) than the ones needed for NIZK protocols in the seemingly "harder" 2-party settings. We therefore ask:

> **Q2:** Are NIZK/Zaps assumptions inherently needed for an MVZK protocol with 1-offline and 1-online round in the honest-majority setting? Is it possible to replace these assumptions with weaker assumptions?

1.1 Our Contribution

1.1.1 Round-Optimal VRS and MVZK in Minicrypt We answer Questions 1 and 2 in the affirmative. Our main result is a protocol with 1-offline round and 1-online round for VRS in the UC-framework, assuming the existence of perfectly-binding non-interactive commitment scheme (NICOM) with sub-exponential privacy. Such a NICOM scheme can be based on injective one-way functions with sub-exponential hardness or even on standard one-way function with sub-exponential hardness assuming worst-case complexity-theoretic derandomization assumptions [45, 8].[4] Throughout, we assume that the parties communicate over pairwise secure and authenticated point-to-point channels, as well as over a common broadcast channel, which allows each party to send a message to all parties and ensures that the received message is identical.

Theorem 1. *Assuming the existence of injective one-way functions with sub-exponential hardness, for every $\epsilon > 0$ the VRS functionality of every efficiently computable relation R can be realized in 1-offline round and 1-online round, with full security against an active rushing adversary, in any of the following settings.*

- (Optimal resiliency for small number of verifiers) *The number of verifiers k is at most logarithmic in the security parameter, and the adversary corrupts less than $(k+1)/2$ parties.*
- (Almost-optimal resiliency for polynomially-many verifiers) *The number of verifiers k grows polynomially with the security parameter and the adversary corrupts less than $(k+1) \cdot (\frac{1}{2} - \epsilon)$ parties.*

Since MVZK is a special case of VRS, we obtain the following corollary.

Corollary 1. *Assuming the existence of injective one-way functions with sub-exponential hardness, the MVZK functionality of every efficiently computable relation R can be realized in 1-offline round and 1-online round, with full security against an active rushing adversary, in the same settings of Theorem 1.*

[4] For technical reasons, the NICOM should satisfy some level of security against selective opening that, by "complexity leveraging", follows from the assumption that the underlying one-way function (or injective one-way function) cannot be inverted in polynomial-time with more than sub-exponential probability. This seems to be a relatively mild assumption; See Remark 2.

For optimal resiliency, we obtain a protocol with complexity polynomial in the security parameter, but exponential in the number of verifiers k. On the other hand, for every $\epsilon > 0$ we obtain a protocol with resiliency $(k+1) \cdot (\frac{1}{2} - \epsilon)$, whose complexity is polynomial both in the security parameter and in k. (In fact, we can push ϵ to be as small as $\epsilon = \Omega(\frac{1}{\sqrt{\log k}})$; see the full version [6] for full details.)

The difference between optimal resiliency and "almost-optimal resiliency" is mostly relevant when the number of verifiers is small, e.g., constant. In this setting, the first protocol provides an efficient solution. Specifically, we highlight the case of 3-party computation, with a single prover and two verifiers, and we note that by adding just a single verifier to the standard zero-knowledge settings, we can obtain a protocol with 1-offline round and 1-online round for the case of a single corruption from Minicrypt-type assumptions. (In contrast, general-purpose 3-party MPC for honest majority requires 3 rounds [47].)

Still, the existence of a strict-honest-majority 2-round VRS protocol whose complexity scales polynomially with the number of parties, remains an interesting open problem. We show that such a protocol can be constructed if one is willing to make stronger assumptions (e.g., random oracle or correlation-intractable functions) or if the adversary is non-rushing. In fact, we note that a weak limitation of the rushing capabilities of the adversary suffices, and present a new notion of *semi-rushing* adversary to model such a behavior.[5]

1.1.2 VRS and MVZK with Everlasting Security in Minicrypt

It is known that if we do not put restriction on the round complexity, then, in the setting of honest-majority, one can obtain *unconditional* results and no assumptions are needed at all! Specifically, as shown by Rabin and Ben-Or [49], every efficiently computable function can be securely computed with statistical security against computationally-unbounded adversaries. While we do not know whether it is possible to achieve statistical security in 2 rounds, we show that VRS and MVZK can be implemented by a protocol that achieves *statistical everlasting security* assuming an access to a collision-resistant hash function h. The notion of statistical everlasting security [44] can be viewed as a hybrid version of statistical and computational security. During the run-time, the adversary is assumed to be computationally-bounded (e.g., cannot find collisions in the hash function) but after the protocol terminates, the adversary hands its view to a computationally-unbounded analyst who can apply arbitrary computations in

[5] The difference between rushing and non-rushing adversary boils down to the scheduling of the messages within a single round of a protocol. A *non-rushing* adversary must send the messages of the corrupt parties in a given round before receiving the messages of the honest parties in that round, whereas a *rushing* adversary may delay sending the messages of the corrupt parties until receiving the messages from the honest parties. Thus, the messages of the corrupt parties may depend on the messages of the honest parties in the same round. Our notion of *semi-rushing* adversary allows the adversary to see all the messages of the honest parties, except for one. For more about this model and its relevance, see the full version [6].

order to extract information on the inputs of the honest parties (e.g., finding collisions or even reading the whole truth table of h).[6] This feature is one of the main advantages of information-theoretic protocols: after-the-fact secrecy holds regardless of technological advances and the time invested by the adversary.

Theorem 2. *Given an access to a collision-resistant hash function, the VRS and MVZK functionalities of efficiently computable relations can be realized in 1-offline round and 1-online round, with full security and* everlasting security *against an active rushing adversary, in the same settings (honest-majority with few verifiers or almost-honest majority with many verifiers) of Theorem 1.*

Remark 1 (On the use of hash function). Our protocol assumes that all parties are given an access to a collision resistance hash function h. Theoretically speaking, such a function should be chosen from a family of functions $\mathcal{H}$ in order to defeat non-uniform adversaries. One may assume that h is chosen once and for all by some simple set-up mechanism. In particular, by using the standard concatenation-based combiner for hash functions [41], this set-up mechanism may be realized distributively by a single round of public random coins where security holds against an active rushing adversary that may corrupt all the participants except for a single one. The choice of the hash function can be abstracted by a CRS functionality, or even, using the multi-string model of [37] with a single honestly-generated string. However, it should be emphasized that this CRS is being used in a very *weak* way: It is "non-programmable" (the simulator receives h as an input) and it can be sampled once and for all by using the above trivial public-coin mechanism. Even if one counts this extra set-up step as an additional round, to the best of our knowledge, everlasting security was not known to be achievable regardless of the underlying assumptions.

The difference between everlasting and computational security is *fundamental* and is analogous to the difference between statistical commitments and computational commitments or statistical ZK vs. computational ZK (see, e.g., the discussions in [19, 46]). Indeed, Theorem 2 provides (UC-secure) MVZK with a *statistical zero-knowledge* property. As a side bonus, Theorem 2 does not require sub-exponential hardness assumptions.

1.1.3 Round-Optimal Linear Function Computation in Minicrypt

Using the machinery we develop for VRS and MVZK, we obtain a 3-round protocol for linear function computation. By the lower-bound of [34] our protocol has optimal round complexity. Like in previous results, we assume the existence of injective one-way functions with sub-exponential hardness in order to obtain a protocol with computational security in the plain model, or an access to a collision resistance hash-function in order to obtain a protocol with everlasting security. In contrast, previous works achieve only computational security by assuming public-key encryption and Zaps [2]. We emphasize that in Theorem 3 we

[6] Technically, in the UC-framework we allow the environment to output its view and require statistical indistinguishability between the real and ideal experiments.

obtain *optimal resiliency* even when the number of parties is polynomial in the security parameter.

Theorem 3. *Assuming the existence of injective one-way functions with sub-exponential hardness, every efficiently computable linear function can be realized in 3 rounds, with full security against an active rushing adversary, that corrupts a minority of the parties. If we replace the one-way function with an access to a collision resistance hash-function, we also obtain everlasting security.*

1.1.4 Applications We present some applications of our protocols. For full details, see the full version [6].

MVZK as a NIZK-substitute for honest majority. We notice that our MVZK protocol captures an important aspect of NIZK, its *minimal round complexity*, while using only Minicrypt-type assumptions. Indeed, our MVZK protocol implies that the CRS for NIZK is not required, and can be replaced with only a *single* offline-round of communication. Similar to NIZK, the proof itself requires only *one online round*. However, unlike NIZK, in our protocol all the parties have to communicate in the online round.

Round-efficient manipulation of non-homomorphic commitments. In a common scenario in multiparty computation, a party $\mathcal{P}$ holds openings to public commitments $C_1, \ldots, C_\ell$. $\mathcal{P}$ wishes to apply some function f on the committed values $z_1, \ldots, z_\ell$ and let the rest of the parties learn $y := f(z_1, \ldots, z_\ell)$, while proving in zero-knowledge that she used the committed values in the computation of f. Alternatively, $\mathcal{P}$ may want to generate another commitment C, that hides y, while proving in zero-knowledge that C was honestly generated. Both the tasks can be solved in 1-offline round and 1-online round by using our MVZK. Since the offline round can be executed in parallel to the generation of $C_1, \ldots, C_\ell$, both tasks require only one additional round!

Round-efficient GMW-type compilers in Minicrypt. Using VRS one can obtain round-efficient GMW-type compilers in Minicrypt, for the case of honest majority. Given a protocol π which is secure against a semi-malicious adversary,[7] we obtain a protocol π' with unanimous abort against an active adversary at the expense of adding a single offline round. If π is secure against a passive (aka semi-honest) adversary, the overhead grows to 4 rounds. Notably, unlike the GMW compiler, our transformation avoids the use of public-key encryption.

Round-optimal honest-majority MPC in Minicrypt. A followup work by the same authors [5] shows that general secure multiparty computation with *full-security* (including guaranteed output delivery) in the presence of an honest

[7] A *semi-malicious* adversary is allowed to choose its input and randomness but otherwise follows the protocol. Many passively secure protocols (e.g., [12]) actually offer semi-malicious security.

majority can be achieved in an optimal number of 3 rounds based on Minicrypt-type assumptions (e.g., NICOMs). A main building block of the protocol is our 2-round offline/online VRS protocol.

Bibliographic Note. Previous unpublished version of [5] contained a weak form of some of the current results based on the Fiat-Shamir heuristic. These results were removed from the new version of [5], and are fully subsumed by the current paper.

1.2 Related Works and Comparison

The VRS functionality was implicitly studied by Gennaro *et al.* [34], in the context of single input functionalities. Gennaro *et al.* provided a two-round perfect protocol with resiliency $(k+1)/6$. The resiliency was improved to $(k+1)/3$ by Applebaum et al. [3], at the cost of degrading the perfect security to computational security, assuming the existence of NICOMs.

Boneh et al. [16] initiated the formal study of zero-knowledge proofs over secret-shared data. They considered information-theoretic security in the following models of corruptions: (1) the adversary corrupts the prover *or* up to $k-1$ verifiers, and (2) the adversary corrupts the prover *and* less than $k/2$ verifiers. In both corruption models, they only provide *security with abort*. Their protocols exploit PCP machinery to achieve low communication complexity (sub-linear in the description of the relation), but have a super-constant number of rounds. Based on a random oracle, the number of rounds can be collapsed to 2, assuming that the data is already secret-shared among the verifiers.

MVZKs were first introduced in [20]. The most relevant MVZK for us can be derived from [37] which provides a construction of NIZK in the *multi-string model* assuming the existence of Zaps. In the multi-string model, the CRS is replaced with several authorities, each providing the protocol with a public random string, and the protocol is secure as long as a majority of those authorities are honest (that is, if a majority of the strings are uniformly distributed). An MVZK protocol with an honest majority of parties can be obtained in the plain model by letting each party broadcast a random string in the offline round, so that a majority of the strings are uniformly distributed. Simulation-based security can be obtained via the additional help of public-key encryption [37, Theorem 3].

Other non-interactive variants of MVZK were presented in [1]. Translated to our model, their work yield 2-round MVZK for $t < k/3$ and a 3-round protocol for $t < n/2$. Both results hold under public-key (discrete-log) hardness assumptions. Recently, [51] and [9] constructed MVZK with practical real-world efficiency in honest and super-honest majority settings. However, their low round (2 or 3) variants rely on random oracle and achieve either selective or identifiable abort.

Comparison. We compare our results with the relevant existing results in Table 1. Except for this work and [37], none of the works achieves an offline/online construction.

Ref.	Primitive	Rounds	Threshold	Assumptions	Security†
[34]	VRS	2	$t < (k+1)/6$	–	it and full security
[3]	VRS	2	$t < (k+1)/3$	NICOM	cs and full security
[16]	ZK over shared data	2⋆	$t < (k+1)/2$‡	Random Oracle	it and abort
[37]	MVZK	2	$t < (k+1)/2$	PKE	cs and full security
[1]	MVZK	3	$t < (k+1)/2$	Discrete-log	cs and full security
[51]	MVZK	2	$t < (k+1)/2$	Random Oracle	it and abort
[9]	MVZK	2	$t < (k+1)/3$	Random Oracle	it and identifiable abort
This paper	VRS	2	$t < (k+1)(\frac{1}{2} - \epsilon)$§	NICOM⋆⋆	cs/es and full security

† it: information-theoretic, es: everlasting security, cs: computational security,
‡ They assume the adversary corrupts (1) the prover *or* up to $k-1$ verifiers, and (2) the prover *and* less than $k/2$ verifiers
⋆ The round complexity does not include the rounds needed for data sharing.
⋆⋆ Perfectly-binding and sub-exponentially hiding NICOM for cs security and Computationally-binding and statistically-hiding NICOM for es security.
§ We achieve $t < (k+1)/2$ when k is logarithmic in the security parameter.

Table 1: Comparison of our work with the state-of-the-art relevant results

2 Preliminaries

Single-Input Functionalities. We adopt an MPC-based notation and replace VRS with the following notion of *single-input functionalities* (SIF). We assume that there are n parties, $\mathsf{P} = \{P_1, \ldots, P_n\}$, where one party (e.g., P_n) takes the role of a *Dealer* D. The SIF functionality $\mathcal{F}$ is parameterized with a function $f : \{0,1\}^* \to (\{0,1\}^*)^n$, it takes an input string $\mathbf{z}$ from the dealer, computes the outputs $(\mathbf{y}_1, \ldots, \mathbf{y}_n) = f(\mathbf{z})$ and delivers $\mathbf{y}_i$ to the ith party P_i. It is not hard to see that VRS is a special case of SIF, and that VRS implies SIF in a round-preserving way. (Indeed, to realize $\mathcal{F}$ define the relation R that accepts a vector $(x_0, x_1, \ldots, x_{n-1})$ if $x_i = f_i(x_0)$ for $i \in [n-1]$, and let D invoke a VRS for R with the input $(z, f_1(z), \ldots, f_{n-1}(z))$.) We will mostly focus on the special case of *public-SIF* that delivers the same output to all the parties. In the full version [6] we show that a 2-round offline/online general-SIF reduces to 2-round offline/online public-SIF via the aid of NICOMs.

Security model. We consider an active static, rushing adversary that may corrupt up to t parties. We consider two main settings: the optimal resiliency setting where $n = 2t+1$ and the almost-optimal resiliency setting where $n = (2+\epsilon)t$ for some arbitrarily small constant $\epsilon > 0$. The parties are connected by pairwise secure channels and additionally a broadcast channel is available. We prove security of our protocols in the UC-framework [21]. We identify the set of parties P with $\{1, \ldots, n\}$, and denote the set of honest parties by $\mathsf{H} \subseteq \mathsf{P}$, and the set of corrupt parties by $\mathsf{C} \subseteq \mathsf{P}$. In our protocols, we follow the convention that the honest parties can "disqualify" the dealer whenever it is clear from broadcast messages that the dealer misbehaves. This does not violate "guaranteed output

delivery" since in case of disqualification, the honest parties can always apply f on some predetermined default value and output the result. We denote by κ the security parameter and implicitly assume that all other parameters (e.g., the number of parties, and the complexity of the functionalities and protocols) depend in κ.

NICOM. A NICOM consists of two PPT algorithms $(\mathsf{commit}, \mathsf{open})$ where commit takes a security parameter κ, message x and random coins r, and outputs a commitment C and a corresponding opening information o. The open algorithm takes κ, and a commitment/opening pair (C, o) and outputs the message x or a failure message $\perp$. The algorithms should satisfy the standard properties of correctness, binding (i.e., it must be hard for an adversary to come up with two different openings of any C) and hiding (a commitment must not leak information about the underlying message) properties. NICOM comes in 2 main flavors: (1) with computational hiding and perfect binding, and (2) with statistical hiding and computational binding. Type (1) commitments can be based on injective one-way functions [14, 52, 35], and type (2) commitments can be based on collision resistance hash functions [27, 39]. In the latter case, a description of a collision resistance hash function h (that is sampled from a family $\mathcal{H}$) is given to the algorithms $(\mathsf{commit}, \mathsf{open})$ as an auxiliary public parameter. Our protocols make use of NICOM in a modular way such that a type (1) instantiation (with sub-exponential computational hiding) yield computational protocols and type (2) instantiation yield protocols with everlasting security.

Remark 2 (Sub-exponential hiding). Assuming injective OWF over m-bit inputs that cannot be inverted by a PPT adversary with probability better than $2^{-m^{\delta}}$, it is possible to construct [14, 52, 35] a plain-model (with no public parameters) perfectly-binding NICOM whose computational hiding property holds for $\epsilon \leq 2^{-\kappa}$. We refer to such a commitment as *perfectly binding sub-exponentially hiding* NICOM. Moreover, under worst-case derandomization assumptions [8], such NICOMs can be based on general (not necessarily injective) sub-exponentially hard OWFs. Similar sub-exponential hardness assumptions are quite common in the literature and typical candidate one-way functions seem to achieve sub-exponential hardness. In fact, our variant of sub-exponential hardness is relatively mild compared to other notions, since we do not allow the adversary to run in sub-exponential time, but only allow it to succeed with sub-exponentially small probability.

3 Technical Overview

In this section we provide a high level overview of our SIF protocol. Full details of the protocol appear in the full version [6]. Intuitively, a SIF protocol consists of the following sequential parts: (1) The dealer presents a statement; (2) The other parties challenge it via a random challenge; (3) The dealer sends a respond; and (4) The other parties decide whether to accept or reject. Compressing these steps

into 2 rounds is highly challenging. For comparison, even the task of verifiable secret sharing (without revealing it) takes at least 2 rounds [33, 7]. To bypass this problem, we are forced to run sub-protocols in parallel and with some overlap. Specifically, we make an extensive use of (1) *tentative-output* protocols that prepare a tentative version of the output in an early round and only later, at the end, approve/reject/correct the tentative output; and (2) *offline-phase* protocols that begin with an *offline*, input-independent, round and only later receive the inputs. This allows us to save some rounds by allowing partial overlap between sub-protocols.

Our protocol makes an extensive use of *verifiable secret sharing* (VSS) [23]. For now, let us think about a VSS protocol as an actively-secure realization of the ideal functionality that takes as an input a secret $s \in \mathbb{F}$ and randomness r from a dealer, and delivers to each party P_i a share s_i that is generated from s and r by using some threshold secret sharing scheme with threshold t. Here and throughout the paper, $\mathbb{F}$ is a finite field whose size is assumed to be exponential in the security parameter κ, by default, $\mathbb{F} = \mathrm{GF}(2^\kappa)$. The underlying secret sharing scheme should be *binding* in the sense that a corrupted party cannot "lie" about its share. (This property implies that correct reconstruction is achievable even at the presence of an active adversary as long as we have $n - t$ honest parties.) To simplify the exposition, let us assume for now that the underlying seceret sharing is *linearly homomorphic* and that the VSS protocol takes a *single round*. We emphasize that both features are unrealistic and even impossible to achieve when $t > n/3$, let alone when t is close to $n/2$.[8] Jumping ahead, a considerable part of this work will be devoted to the removal of these assumption while preserving the round complexity; see Section 3.3.

3.1 SIF for Few Parties

Let us restrict our attention to the case where the number of parties n is small, i.e., $n = O(\log \kappa)$. Recall that our goal is to construct a 2-round protocol for a general SIF functionality whose first round is an offline round that does not depend on the input of the dealer. We will use standard techniques to reduce this problem to the problem of constructing a 2-round protocol for a specific SIF functionality known as *triple secret sharing* (TSS) where the dealer wishes to share a triple (a, b, c) such that $c = ab$. For TSS, let us strive for a "standard" 2 round protocol whose first round is allowed to depend on the input.

2-round TSS against non-rushing adversary. Our starting point is the following 2 round protocol that assumes that a corrupted dealer is non-rushing. In the

[8] Even without homomorphism, computational VSS requires 2 rounds [7] when $n < 3t$. Moreover, even for such a large resiliency threshold, linear homomorphism is non-trivial to achieve. Specifically, for 2-round VSS, it is unknown how to achieve linear homomorphism without relying on strong primitives such as *homomorphic* NICOMs. The latter are typically contructed based on "structured" (public-key type) assumptions and are not known to follow from standard NICOMs.

first round, the dealer D, that holds a triple (a, b, c) with $c = ab$, picks three polynomials $A(x), B(x)$ and $C(x)$ of degree n, n and $2n$, respectively, such that $A(0) = a$, $B(0) = b$, $C(0) = c$ and $C(x) = A(x) \cdot B(x)$. Let A^i, B^i and C^i be the ith coefficient of $A(x), B(x)$ and $C(x)$, and note that $A^0 = a, B^0 = b$ and $C^0 = c$. The dealer shares all the coefficients $\{A^i, B^i\}_{i\in\{0,\ldots,n\}}$, and $\{C^i\}_{i\in\{0,\ldots,2n\}}$ via VSS. The parties now hold the shares of $a = A^0, b = B^0$ and $c = C^0$.

In order to ensure that $c = ab$, it suffices to verify that the polynomial $C(x)$ is equal to the polynomial $A(x) \cdot B(x)$. To this end, we want to compute $A(\alpha), B(\alpha)$ and $C(\alpha)$ for a random non-zero field element α, and verify that $C(\alpha) = A(\alpha)B(\alpha)$. Indeed, if $C(x) = A(x) \cdot B(x)$ then equality always holds, while if $C(x) \neq A(x) \cdot B(x)$ then the probability that the verification succeeds is at most $2n/(|\mathbb{F}| - 1) = \mathsf{negl}(\kappa)$. Therefore, in the first round, concurrently to the sharing of the dealer, we let every party P_i broadcast a random non-zero field element α_i.

In the second round, our goal is to compute $A(\alpha_i), B(\alpha_i), C(\alpha_i)$ for all $i \in \{1, \ldots, n\}$ and "disqualify the dealer" if for some α_i the test $A(\alpha_i) \cdot B(\alpha_i) = C(\alpha_i)$ fails. Recall that $A(x)$ and $B(x)$ are random polynomials of degree n conditioned on $A(0) = a$ and $B(0) = b$, and therefore one can safely release all these α_i evaluations without revealing any information on a, b and c. The actual computation of $A(\alpha_i), B(\alpha_i), C(\alpha_i)$ makes use of the linear-homomorphism of the secret-sharing. Specifically, observe that $A(\alpha)$ is just a linear function of $A^0, \ldots, A^n$ with coefficients $(\alpha^0, \ldots, \alpha^n)$ (and similarly for $B(\alpha)$ and $C(\alpha)$), and therefore each party can reveal in the second round its share of $A(\alpha_i)$ (resp., $B(\alpha_i), C(\alpha_i)$). The binding property of the VSS guarantees that a corrupted party cannot lie about its shares and the existence of $t + 1$ honest parties guarantees successful reconstruction. The protocol follows the standard commit-challenge-response template with a minor tweak: many challenges are generated (one for each "verifier") concurrently to the commitment stage, and each of the responses is being computed collectively by the "verifiers".

Coping with a rushing adversary. The above protocol is insecure against a rushing adversary since such an adversary can wait to see the selected challenges and then share triples that do not satisfy the product relation and yet pass the tests. We solve this problem by hiding at least some of the challenges from the adversary while revealing them to enough parties so that the response (via reconstruction) can be computed in the second round. Details follow.

Consider all the possible $(t + 1)$-subsets of the parties, $Q_1, \ldots, Q_N$ where $N = \binom{n}{t+1}$. In the first round, we let each subset Q_i generate a secret challenge α_i that is known only to the members of Q_i. Specifically, we define some canonical "leader" for Q_i (e.g., the party with the smallest index) and let her sample a random non-zero α_i and send it to the other members of Q_i over private channels. Concurrently, the dealer shares the coefficients of the polynomials A, B, C among the n parties as before, except that now the degree of A and B is taken to be $d = N(t + 1)$ and the degree of C is taken to be $2d$. In the second round, each party P_j in Q_i broadcasts the value α_i and uses local linear operations to reveal to all the parties the jth share of $A(\alpha_i), B(\alpha_i)$ and $C(\alpha_i)$. After the second

round, for each i, each party P (possibly outside Q_i) verifies that all the parties in Q_i broadcast the same point α_i and that their shares are valid. If one of these checks fail, we refer to the ith test as *bad* and ignore it; Otherwise, the i-th test is called *good*, and P can recover the points $A(\alpha_i), B(\alpha_i)$ and $C(\alpha_i)$. If these values satisfy the product relation, we say that the (good) test *passes*. Finally, P accepts the triple if all the good tests pass, and disqualifies the dealer otherwise.

The analysis is fairly simple. For a corrupt D, we note that there exists (at least) one set Q_i in which all the parties are honest, and that a corrupt dealer has no information about α_i in the first round. The parties in Q_i provide in the second round $t+1$ shares of $A(\alpha_i), B(\alpha_i)$ and $C(\alpha_i)$ and so these values can be publicly recovered, and the probability that $C(x) \neq A(x) \cdot B(x)$ and $C(\alpha_i) = A(\alpha_i) \cdot B(\alpha_i)$ is at most $2d/(|\mathbb{F}| - 1) = 2N(t+1)/(|\mathbb{F}| - 1) = \mathsf{negl}(\kappa)$. Thus, except with negligible probability, there will be at least one good test that fails to pass. On the other hand, an honest dealer will never be disqualified since, by the binding property of the secret sharing, even a fully corrupted set of verifiers Q_i cannot reveal incorrect shares. As for privacy, there are N sets, and from each set the adversary can learn information about at most $(t+1)$ points of $A(x), B(x)$ and $C(x)$ (a corrupt leader in a set Q can send different evaluation points to the parties in Q). Since the degree of $A(x)$ and $B(x)$ is d, and the adversary can learn information about at most $N(t+1) = d$ points, we conclude that the adversary learns no information about $A(0), B(0)$ and $C(0)$, as required. The complexity of the protocol is exponential in $t = \lceil n/2 \rceil - 1$ and so the protocol is efficient (polynomial in the security parameter κ) only when the number of parties n is logarithmic in κ. Indeed, this is the only place where the assumption $n = O(\log \kappa)$ is really necessary.

From TSS to public SIF. By the standard NP-completeness of quadratic equations, public SIF non-interactively reduces to public SIF where f computes a vector of degree-2 polynomials over an arbitrary finite field [34] and the same output is given to all the parties. One can easily adopt the TSS protocol to the case of general degree-2 SIF functionality (e.g., share the input vector $\mathbf{z}$ and the output vector $\mathbf{y}$, prove that they satisfy a degree-2 relation and ask the parties to publicly reconstruct $\mathbf{y}$.) However, this will not lead to an offline/online protocol. Instead, we use Beaver's trick [10] to transform random triple sharing (realized by TSS) into a degree-2 SIF. The standard transformation has an overhead of 2 additional rounds, and we avoid it by exploiting the SIF setting, i.e., the fact that a single dealer knows *all* the secrets. A reduction from general SIF to public SIF appears in the full version [6].

3.2 SIF for any Number of Parties

We move on to the case where the number of parties, n, is large (polynomial in κ) and the resiliency threshold t is almost optimal, i.e., $n = (2+\epsilon)t$ for some constant $\epsilon > 0$. Our goal is to construct a 2-round offline/online protocol Π for some public SIF functionality $\mathcal{F}$ that takes an input $\mathbf{z}$ from the dealer D and delivers the same output $\mathbf{y} = f(\mathbf{z})$ to all the parties.

We will handle this case by composing two protocols: (1) The aforementioned 2-round SIF protocol Π_s ("s" for small) that achieves an optimal resiliency for a small (logarithmic) number of parties; and (2) a perfectly-secure SIF protocol Π_b ("b" for big) with constant resiliency of, say 1/3, that works efficiently for polynomially many parties. The latter protocol can have many rounds and can be instantiated, for example, by the classical protocol of Ben-Or, Goldwasser and Wigderson (BGW) [12]. We will combine the 2 protocols into a single SIF protocol with almost-optimal threshold and poly(n) complexity via *player virtualization* technique. This idea goes back to the work of Bracha [18] in the context of Byzantine Agreement, and since then has been used several times in the MPC literature [32, 40, 26] culminating in the celebrated MPC-in-the head paradigm [42, 43]. Here we show how to apply this idea in the context of SIF. Unlike other contexts, we show that the combined protocol inherits the round complexity of the first ("internal") protocol, and therefore can be executed in 2 rounds! Details follow.

Let us partition the n parties to $M = \text{poly}(n)$ committees $A_1, \ldots, A_M$ each of size n' for some constant n' that depends on the constant ϵ. Call a committee *good* if it contains at least $(n'+1)/2$ honest parties, and *bad* otherwise. We will make sure that the fraction of bad sets is at most $M/10$ no matter which subset of t parties the adversary decides to corrupt. Such a property can be guaranteed by taking all n' multisets or, more efficiently, based on expander graphs (see, e.g., [26, Lemma 5]).[9] Let Π_b be the BGW protocol that realizes the SIF f among the dealer D and M "virtual" parties $Q_1, \ldots, Q_M$.

In our new protocol, Π, the dealer D executes the BGW protocol Π_b in her "head" with the input $\mathbf{z}$ and then broadcasts a commitment to the transcript. That is, D samples random tapes $r_1, \ldots, r_M$ for the virtual parties $Q_1, \ldots, Q_M$ and computes all the messages that are sent in Π_b, both over private channels and over broadcast channels. Then, D commits to each of these messages and to the randomness r_i of each party Q_i, and broadcasts the tuple of commitments G. We emphasize that every message from Q_i to Q_j has only *one* commitment, that belongs both to the view of Q_i and the view of Q_j. In addition, D broadcasts the value $\mathbf{y} = f(\mathbf{z})$. Now, we let each committee A_i verify, with the aid of the small protocol Π_s, that the view of Q_i is *self-consistent*, i.e., that the (committed) randomness and incoming messages of Q_i yield the (committed) outgoing messages of Q_i and that the final output is indeed $\mathbf{y}$. More precisely, the committee A_i together with D, compute the following public-SIF functionality $\mathcal{G}_{\mathsf{zk}}$:

- (Dealer's input:) An index $i \in \{1, \ldots, M\}$, a vector of commitments G_i, supposedly to the randomness of Q_i and his incoming and outgoing messages, and the corresponding openings.

[9] In principle, n' should be taken to be $\Omega(1/\epsilon^2)$. Thus, in order to keep n' small (e.g., logarithmic in the security parameter), one has to assume that ϵ is not too small, e.g., at least $\Omega(1/\sqrt{\log \kappa})$. We limit the discussion to a constant ϵ only for the sake of simplicity.

– (Public output:) the tuple $(v_i, \mathbf{y}_i, G_i, i)$ where v_i is a consistency bit that indicates whether the committed values are self-consistent, and the value $\mathbf{y}_i$ is the output that the virtual party Q_i outputs given the committed view.[10]

We realize this sub-computation by running the small SIF protocol Π_s among D and the sub-committee A_i while making sure that the final output is available to all parties including ones that do not belong to A_i. This can be done (without an extra round of communication) by passing all the broadcast messages of the small protocol Π_s over the external n-party broadcast channel. Indeed, we note that, for public-output SIF, the public output of our protocol Π_s can be fully recovered based on its broadcast messages. Getting back to Π, we conclude the protocol, by letting each party P_i accept the output $\mathbf{y}$ if at least $0.9M$ of the committees approve this output (i.e., if the output of the ith committee is $(1, \mathbf{y}, G_i, i)$ where G_i is consistent with G), and disqualify the dealer otherwise.

The protocol Π can be executed in 2 rounds where the first round is devoted to the offline round of all the instances of the Π_s protocol, and the second round is devoted to the commitment generation and to the second online-round of the Π_s instances. Note that the first round of Π remains input-independent. Let us briefly analyze the security of Π.

For an honest dealer, the verification Π_s succeeds for every good committee Q_i that contains an honest majority, and may fail for a bad committee Q_i that contains a dishonest majority. We conclude that at most $M/10$ of the verifications fail, and so an honest dealer will never be disqualified. As for privacy, a bad committee Q_i may completely learn the input of the dealer D in the corresponding SIF $\mathcal{G}_{\mathsf{zk}}$. This leakage is equivalent to learning the internal state of the virtual party Q_i in the external protocol Π_b. Since there are at most $M/10$ bad committees, the adversary can learn the state of at most $M/10$ parties of Π_b. The privacy of Π_b therefore protects us against such a leakage. (In fact, for this part we only use the privacy of Π_b against a passive corruption.)

A corrupt dealer can commit to an illegal transcript while being approved by all bad committees. So, in order to be approved, such a dealer must still get the votes of at least $0.8M$ good committees. Hence, cheating in Π reduces to cheating in Π_b while actively controlling at most $0.2M$ of the virtual parties, and while controlling the randomness of the honest virtual parties. Since Π_b is *perfectly correct* against $0.2M$ active corruptions, a cheating dealer will always be caught. (For this part, no privacy is needed and Π_b is only required to achieve "perfect correctness with abort" against an active adversary.)

Remark 3 (Comparison to the MPC-to-ZK transformation of [42]). It is instructive to consider the following variant of the protocol. First, the dealer secret-shares its input $\mathbf{z}$ to $(\mathbf{z}_1, \ldots, \mathbf{z}_M)$ via some robust $M/3$-out-of-M secret sharing then it virtually runs an MPC protocol among the parties $Q_1, \ldots, Q_M$ for the public SIF $\mathcal{F}'$ that takes $(\mathbf{z}_1, \ldots, \mathbf{z}_M)$ from the parties, recovers $\mathbf{z}$ via robust reconstruction, and delivers the output $f(\mathbf{z})$. The dealer commits to the views and

[10] The circuit that realizes $\mathcal{G}_{\mathsf{zk}}$ depends on the code of the NICOM, consequently, our final construction makes a non-black-box use of the NICOM.

transcript and the committees $A_1, \ldots, A_M$ use the small SIF protocol to verify consistency for each virtual party. This description be can viewed as a special case of the protocol Π in which Π_b is realized by sharing $\mathbf{z}$ and computing $\mathcal{F}'$.

Under this choice, our transformation can be viewed as a multi-verifier version of the MPC-to-ZK transformation of [42]. The two versions differ with respect to the underlying secret sharing (M-out-of-M in [42] vs. $M/3$-out-of-M in our case), and, more importantly, with respect to the verification part. In [42] a single verifier opens few views (for soundness) while keeping other views unopened (for zero-knowledge), whereas in our case multiple verifiers distributively open (all) the views in a way that preserves soundness "globally", and secrecy for bounded-size coalitions. Furthermore, we show that verification can be realized with low round complexity based on an "internal" SIF protocol.

3.3 Replacing the Idealised VSS with 1.5-Round Protocols

In the previous section, TSS and public SIF for logarithmic number of parties are the direct consumers of the idealized VSS. In both, the scenario is as follows: D has m inputs $s_1, \ldots, s_m$ and the parties want to compute a linear combination of the inputs. The coefficients of the linear combination may be chosen by some other party, and the output should be delivered by the end of second round. For simplicity, we consider the somewhat degenerate case where the goal is to compute $z := s_1 + \ldots + s_m$. As mentioned earlier, two challenges arise: (a) VSS sharing itself requires 2 rounds, whereas our requirement is to complete sharing and reconstruction within 2 rounds and (b) the known 2-round VSS from Minicrypt-like assumptions is not homomorphic. In a nutshell, we solve the first issue by noting that the VSS of [7] is a "1.5-round" VSS in the sense that "tentative shares" are distributed already in the first round, and any update that may occur in the second round is *publicly known* to all parties. To solve the second issue, we construct a novel protocol that allows a party to reveal a "certified" linear combination of its shares. This protocol, glinear, has 2 rounds where the first round is an offline round. Since our protocols employ linear homomorphism during their second round, glinear forms a viable substitute. Related tools have been developed in [4] for a smaller resiliency threshold (e.g., $n \geq 3t + 1$), and we extend them to the challenging setting of $n = 2t + 1$ while maintaining efficiency for polynomially many parties $n = \text{poly}(\kappa)$. Before describing our solutions in more detail, we present some background on the underlying secret sharing scheme.

The underlying secret sharing scheme. The secret sharing scheme is essentially the classical t-out-of-n Shamir-like scheme (extended to bivariate polynomials as in [12]) accompanied with public commitments to all the shares. To (honestly) share a secret $s \in \mathbb{F}$, one samples a random symmetric bivariate polynomial $F(x, y)$ of degree at most t in each variable conditioned on $F(0, 0) = s$, and hands to each party P_i the vector $(F(i, 0), \ldots, F(i, n))$ which fully defines the degree-t univariate polynomial $f_i(x) = F(i, x)$. We embed these elements in an $(n + 1)$-by-$(n+1)$ matrix $\mathbf{F} = (F(i, j))_{i,j \in \{0,\ldots,n\}}$, and note that this matrix is symmetric

since $F(i,j) = F(j,i)$. The 0th row of this matrix is referred to as the *main* row and its ith entry $F(0,i) = F(i,0)$ is referred to as the main share of party P_i. (The main row corresponds to the univariate polynomial $f_0(x) = F(0,x)$ which forms a standard Shamir sharing of s.) As part of the secret sharing, we publish a symmetric matrix, $\mathbf{C} = (C_{ij})_{i,j\in\{0,\ldots,n\}}$ of commitments to each entry of $\mathbf{F}$, and hand the openings, $\mathbf{O}_i = (o_{ij})_{j\in\{0,\ldots,n\}}$, of the ith row to party P_i. We let $\mathbf{O}$ denote the matrix of openings $(o_{ij})_{i,j\in\{0,\ldots,n\}}$. It is well-known that this scheme is t-out-of-n secret sharing scheme. The commitment layer makes it impossible for a corrupted party to lie about its share (the scheme is "binding"), and so it enables robust reconstruction.[11] We point out that a statistically-hiding computationally-binding commitment leads to a secret sharing scheme with statistical privacy whose robustness holds only against computationally-bounded adversaries whereas a computationally-hiding statistically-binding commitment scheme yields a secret sharing scheme with computational privacy and robustness against computationally-unbounded adversaries. Let us record the fact that the "polynomial part" of the secret sharing is linearly homomorphic but the "commitment part" is not.

1.5-round VSS. Backes et al. [7] describe a 2-round protocol for securely distributing a secret according to the above secret sharing scheme. We note that this protocol has the following structure. After the first ("sharing") round, the commitment matrix $\mathbf{C}$ is delivered to all the parties and each party holds a private *tentative share* that may be invalid. During the second ("verification") round of the protocol, each party P_i who may be "unhappy" for some reason, can form a "complaint" against the dealer D. At the end of this round, either some complaint turns to be "justified", or all the complaints are rejected as being "unjustified". In the former case, the dealer is being publicly disqualified, and in the latter case, the private shares of all unhappy parties are publicly revealed. (That is, all parties learn the openings $(\mathbf{O}_i)_{i\in\mathsf{W}}$ where W is the set of all unhappy parties.) By design, an honest party never complains about an honest dealer. We will make use of the fact that a tentative share either remains unchanged during the second round, or becomes publicly available to all parties.

We formalize these properties via a new 2-phase functionality $\mathcal{F}_{\mathsf{vss}}$ (a refined version of VSS), and prove that the protocol UC-realizes it. The choice of being unhappy is captured by an input $\mathsf{flag}_i \in \{0,1\}$ that is given to P_i at the beginning of the verification phase. As a result P_i can ask to publicly reveal $\mathbf{O}_i$ even it is unhappy with D due to some external reason, that does not depend on the VSS execution (say, P_i thinks that D is corrupt in the outer-protocol).

[11] We, in fact, consider a weak variant of this sharing in which for a pair of corrupted parties, (P_i, P_j), the share $f_i(j)$ may be inconsistent with the commitment C_{ij}. Still, it can be shown that P_i and P_j cannot lie about their *main shares* and so this scheme still allows robust reconstruction. For details, refer to the full version of this paper [6].

3.3.1 Supporting Linear Operations Let us now go back to our goal of computing $z := s_1 + \ldots + s_m$ in two rounds where the secrets $s_1, \ldots, s_m$ are given to D as inputs. We start by running the first round of the VSS to distribute tentative shares for $s_1, \ldots, s_m$ via the polynomials $F^1, \ldots, F^m$ and the commitments $\mathbf{C}^1, \ldots, \mathbf{C}^m$. Our goal now is to publicly reveal the value $z := s_1 + \ldots + s_m$ by using a single round of communication that will be carried in parallel to the verification phase of the VSS. Denote by $F^z(x, y)$ the bivariate polynomial $F^1(x, y) + \ldots + F^m(x, y)$. Observe that it suffices to design a single-round protocol that allows to each party P_i to publish the univariate polynomial $F^z(i, \cdot)$ while providing a certificate for correctness (and while hiding the original shares). Formally, for every "guide" P_i the parties engage in a subprotocol glinear ("guided linear computation") so that (1) if P_i is honest then all parties output $F^z(i, x)$, and (2) if P_i is corrupt then all parties output either $F^z(i, x)$ or an erasure $\perp$. Since there are $n - t \geq t + 1$ honest parties, and all non-$\perp$ shares are consistent with $F^z(x, y)$, the parties can recover the polynomial $F^z(x, y)$ and output $z = F^z(0, 0)$. Observe that we can restrict our attention to the case where the guide is "happy" with the dealer D, since the shares of a non-happy guide will be publicly released anyway in the end of the second round by the verification phase of the secret sharing.

Guided linear computation from SCG. To explain how glinear is implemented, let us focus, for concreteness, on the case where the guide is P_1. After the input sharing, the guide P_1 holds all the information regarding the first rows $F^1(1, x), \ldots, F^m(1, x)$, including the openings to the corresponding commitments. In addition, every P_j holds all the information regarding the j-th share of each first-row, $F^1(1, j), \ldots, F^m(1, j)$. The idea now is to let the guide P_1 and every P_j engage in a subprotocol for the computation of $F^z(1, j)$ where the role of P_j is to *guard* the computation, i.e., to make sure that P_1 uses the "correct" values as inputs. Formally, we construct such a subprotocol, called *secure computation with a guard* and denoted scg, that has essentially the following "patrial security" guarantees:

- If both, P_1 and P_j, are honest then the value $F^z(1, j)$ is given to all parties while the values, $\vec{F}(1, j) := (F^1(1, j), \ldots, F^m(1, j))$, remain hidden.
- If P_1 and P_j are both corrupt, there are no correctness or privacy guarantees.
- If exactly one party is corrupt (either P_1 or P_j) then there are no privacy guarantees and the public output is either $F^z(1, j)$ or an identifiable abort (i.e., $\perp$ symbol accompanied with the identity of the corrupt party).

We postpone the description of the scg protocol. For now, let us mention that the protocol is *publicly decodable* (all honest parties receive the same output that is computed based on broadcasted values), and has 2 rounds in the offline/online model. Since the first round is input-independent we can execute it in parallel to the first round of VSS. Now glinear can be reduced to n executions of scg between P_1 and each of the parties $P_1, \ldots, P_n$, where each P_j acts as the guard of the computation of $F^z(1, j)$. Given the scg outputs, we output a degree-t polynomial

$f_1(\cdot)$ if and only if (1) P_1 was not disqualified by any of the scg calls, and (2) $f_1(\cdot)$ is consistent with all the revealed points. Otherwise, we disqualify P_1. The analysis is straightforward. If P_1 is honest, for every honest guard P_j all the parties learn $F^z(1,j)$ (without leaking information on $\vec{F}(1,j)$), while for every corrupt P_j the parties either learn $F^z(1,j)$ or an erasure $\perp$ (since the adversary already knows $\vec{F}(1,j)$ we do not care about leakage in this case). Since there are $n-t \geq t+1$ honest parties, the parties recover uniquely the polynomial $F^z(1,x)$. If P_1 is corrupt, then it is either being disqualified by one of the honest guards, or release at least $n-t \geq t+1$ points that are consistent with $F^z(1,\cdot)$. This means that the final outcome is either $F^z(1,\cdot)$ or $\perp$. Before delving into the scg construction, we mention that the VSS together with the guided linear computation lead to a protocol for general linear function evaluation in 3 rounds which is optimal by [34].

Realizing scg. Roughly speaking, in an scg protocol, the guide Alice is given as an input a vector b^A and the guard Bob receives a copy, b^B, of this vector that supposedly agrees with b^A. Alice wishes to publicly reveal the value $f(b^A)$, for some public function f, and the guard Bob should make sure that f is computed consistently with respect to his input. This notion was introduced by [3] who constructed a 2-round offline/online protocol that statistically realizes the partial security properties defined above. However, their protocol works with a designated receiver, and so multiple invocations of this protocol (with different receivers) may lead to inconsistent outputs. (Such inconsistencies were tolerated in [3] by leveraging the existence of a strong honest majority, i.e., $t < n/3$.) We present a publicly decodable scg by exploiting the fact that all parties are given *external commitments* C to the input b^A and that the corresponding openings, o, are given to Alice as certificates. Moreover, we make use of NICOM internally in the scg itself, and so get only computational security. Details follow.

Thanks to the external commitments, it suffices to securely compute the functionality $\mathcal{F}$ that takes $x = (b^A, o)$ from Alice and $y = b^B$ from Bob, and outputs

$$y = \begin{cases} f(b^A), & \text{if } b^A = b^B, \\ (b^A, o) & \text{otherwise.} \end{cases}$$

Indeed, if Alice and Bob are honest the output will be $f(b^A)$. If the parties disagree (due to a single cheater) then the output reveals Alice's certified input, and one can check whether the released values (b^A, o) are consistent with the external commitments or not. In the former case, we can decode the output $f(b^A)$, and in the latter case, we conclude that Alice aborted the computation. While we will not be able to realize $\mathcal{F}$ with full security, we provide an instantiation that suffices for "partial security".

Our starting point is the following variant of private simultaneous message (PSM) protocol of [29]. Bob samples a random string r and sends it to Alice privately during the offline phase. Then, in the online phase, given the inputs, x and y, Alice and Bob publish messages, $A(x,r)$ and $B(y,r)$, that publicly

reveal $\mathcal{F}$ and nothing else. Unfortunately, the standard PSM realization only works when both parties are honest, and a dishonest party, say Alice, can violate correctness by sending an invalid message a' that does not correspond to any input x (with respect to the chosen r).

Focusing on the case of corrupt Alice, we modify the protocol as follows. At the offline round, Bob broadcasts *internal commitments* to all the possible PSM online-messages. That is, for every possible Alice-input x (resp., every possible Bob-input y), Bob computes a commitment C'_x to the PSM message $A(x,r)$ (resp., C'_y to the PSM message $B(y,r)$). At the offline round, Bob broadcasts the (randomly permuted) list of commitments $(C'_x)_x$ and $(C'_y)_y$ and privately sends to Alice all the information: the PSM randomness r together with the corresponding openings $(o'_x)_x$ and $(o'_y)_y$. At the online round, Alice and Bob compute the PSM messages that correspond to their inputs, and certify them by opening the corresponding internal commitments. Now, assuming that Bob is honest, Alice is forced to behave honestly in the PSM and must send a "valid" PSM message that corresponds to an actual input x. This protocol achieves a similar guarantee against a cheating Bob and honest Alice, provided that Bob behaves *honestly* in the offline round. We handle the case where Bob misbehaves in the offline round (e.g., by committing to bad values or sending to Alice bad openings) by letting Alice fully expose her certified input. That is, if Alice sees that Bob misbehaved in the offline round, she simply broadcasts her inputs together with the external openings as certificates while ignoring the PSM execution. Here we exploit the fact that no privacy is required at the presence of a cheating Bob.

The above description is somewhat simplified and yields a solution whose complexity is linear in the domain of $\mathcal{F}$ which is too expensive. Moreover, when scg is modelled as a reactive functionality, simulation becomes somewhat subtle and the commitments should satisfy some level of security under a selective-openning attack. More details (including an efficient version based on multiparty PSM protocols and a refined definition of scg) appear in the full version [6].

Acknowledgements. B. Applebaum and E. Kachlon are supported by the Israel Science Foundation grant no. 2805/21. A. Patra is supported by DST National Mission on Interdisciplinary Cyber-Physical Systems (NM-CPS) 2020-2025 and SERB MATRICS (Theoretical Sciences) Grant 2020-2023.

References

1. Abe, M., Cramer, R., Fehr, S.: Non-interactive distributed-verifier proofs and proving relations among commitments. In: Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings. pp. 206–223 (2002)
2. Ananth, P., Choudhuri, A.R., Goel, A., Jain, A.: Round-optimal secure multiparty computation with honest majority. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. pp. 395–424 (2018)

3. Applebaum, B., Kachlon, E., Patra, A.: The resiliency of MPC with low interaction: The benefit of making errors (extended abstract). In: Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II. pp. 562–594 (2020)
4. Applebaum, B., Kachlon, E., Patra, A.: The round complexity of perfect MPC with active security and optimal resiliency. In: 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020. pp. 1277–1284 (2020)
5. Applebaum, B., Kachlon, E., Patra, A.: Round-optimal honest-majority MPC in minicrypt and with everlasting security. IACR Cryptol. ePrint Arch. **2021**, 346 (2021)
6. Applebaum, B., Kachlon, E., Patra, A.: Verifiable relation sharing and multi-verifier zero-knowledge in two rounds: Trading NIZKs with honest majority. Cryptology ePrint Archive (2022), https://eprint.iacr.org/2022/167
7. Backes, M., Kate, A., Patra, A.: Computational verifiable secret sharing revisited. In: Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. pp. 590–609 (2011)
8. Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. In: Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. pp. 299–315 (2003)
9. Baum, C., Jadoul, R., Orsini, E., Scholl, P., Smart, N.P.: Feta: Efficient threshold designated-verifier zero-knowledge proofs. Cryptology ePrint Archive (2022)
10. Beaver, D.: Efficient Multiparty Protocols Using Circuit Randomization. In: Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15. pp. 420–432 (1991)
11. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security. pp. 62–73 (1993)
12. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 1–10 (1988)
13. Bitansky, N., Paneth, O.: Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II. pp. 401–427 (2015)
14. Blum, M.: Coin flipping by telephone. In: Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981. pp. 11–15 (1981)
15. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 103–112 (1988)
16. Boneh, D., Boyle, E., Corrigan-Gibbs, H., Gilboa, N., Ishai, Y.: Zero-knowledge proofs on secret-shared data via fully linear PCPs. In: Annual International Cryptology Conference. pp. 67–97 (2019)
17. Boyle, E., Gilboa, N., Ishai, Y.: Function secret sharing: Improvements and extensions. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1292–1303 (2016)

18. Bracha, G.: An o(log n) expected rounds randomized byzantine generals protocol. J. ACM **34**(4), 910–920 (1987)
19. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988)
20. Burmester, M., Desmedt, Y.: Broadcast interactive proofs (extended abstract). In: Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings. pp. 81–95 (1991)
21. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA. pp. 136–145 (2001)
22. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 68–86 (2003)
23. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In: 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985. pp. 383–395 (1985)
24. Corrigan-Gibbs, H., Boneh, D.: Prio: Private, robust, and scalable computation of aggregate statistics. In: 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17). pp. 259–282 (2017)
25. Corrigan-Gibbs, H., Boneh, D., Mazières, D.: Riposte: An anonymous messaging system handling millions of users. In: Proceedings of the 2015 IEEE Symposium on Security and Privacy. pp. 321–338 (2015)
26. Damgård, I., Ishai, Y., Krøigaard, M., Nielsen, J.B., Smith, A.D.: Scalable multiparty computation with nearly optimal work and resilience. In: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. pp. 241–261 (2008)
27. Damgård, I., Pedersen, T.P., Pfitzmann, B.: Statistical secrecy and multibit commitments. IEEE Trans. Inf. Theory **44**(3), 1143–1151 (1998)
28. Dwork, C., Naor, M.: Zaps and their applications. SIAM J. Comput. **36**(6), 1513–1543 (2007)
29. Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation (extended abstract). In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada. pp. 554–563 (1994)
30. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM Journal on computing **29**(1), 1–28 (1999)
31. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. pp. 186–194 (1986)
32. Fitzi, M., Franklin, M.K., Garay, J.A., Simhadri, H.V.: Towards optimal and efficient perfectly secure message transmission. In: Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings. pp. 311–322 (2007)
33. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: Proceedings of the thirty-third annual ACM symposium on Theory of computing. pp. 580–589 (2001)
34. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: On 2-round secure multiparty computation. In: Advances in Cryptology - CRYPTO 2002, 22nd Annual Inter-

national Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. pp. 178–193 (2002)
35. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA. pp. 25–32 (1989)
36. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. Journal of Cryptology **7**(1), 1–32 (1994)
37. Groth, J., Ostrovsky, R.: Cryptography in the multi-string model. Journal of cryptology **27**(3), 506–543 (2014)
38. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. Journal of the ACM (JACM) **59**(3), 1–35 (2012)
39. Halevi, S., Micali, S.: Practical and provably-secure commitment schemes from collision-free hashing. In: Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. pp. 201–215 (1996)
40. Harnik, D., Ishai, Y., Kushilevitz, E.: How many oblivious transfers are needed for secure multiparty computation? In: Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings. pp. 284–302 (2007)
41. Herzberg, A.: Folklore, practice and theory of robust combiners. Journal of Computer Security **17**(2), 159–189 (2009)
42. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007. pp. 21–30 (2007)
43. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. pp. 572–591 (2008)
44. Müller-Quade, J., Unruh, D.: Long-term security and universal composability. J. Cryptol. **23**(4), 594–671 (2010)
45. Naor, M.: Bit commitment using pseudorandomness. J. Cryptology **4**(2), 151–158 (1991)
46. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for *NP* using any one-way permutation. J. Cryptol. **11**(2), 87–108 (1998)
47. Patra, A., Ravi, D.: On the exact round complexity of secure three-party computation. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. pp. 425–458 (2018)
48. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Annual International Cryptology Conference. pp. 89–114 (2019)
49. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA. pp. 73–85 (1989)
50. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the forty-sixth annual ACM symposium on Theory of computing. pp. 475–484 (2014)
51. Yang, K., Wang, X.: Non-interactive zero-knowledge proofs to multiple verifiers. Cryptology ePrint Archive (2022)

52. Yao, A.C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982. pp. 80–91 (1982)