Round-optimal Honest-majority MPC in Minicrypt and with Everlasting Security
(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 study the round complexity of secure multiparty computation (MPC) in the challenging model where full security, including guaranteed output delivery, should be achieved at the presence of an active rushing adversary who corrupts up to half of parties. It is known that 2 rounds are insufficient in this model (Gennaro et al., Crypto 2002), and that 3 round protocols can achieve computational security under public-key assumptions (Gordon et al., Crypto 2015; Ananth et al., Crypto 2018; and Badrinarayanan et al., Asiacrypt 2020). However, despite much effort, it is unknown whether public-key assumptions are inherently needed for such protocols, and whether one can achieve similar results with security against computationally-unbounded adversaries.

In this paper, we use Minicrypt-type assumptions to realize 3-round MPC with full and active security. Our protocols come in two flavors: for a small (logarithmic) number of parties n, we achieve an optimal resiliency threshold of $t \leq \lfloor (n-1)/2 \rfloor$, and for a large (polynomial) number of parties we achieve an almost-optimal resiliency threshold of $t \leq 0.5n(1-\epsilon)$ for an arbitrarily small constant $\epsilon > 0$. Both protocols can be based on sub-exponentially hard injective one-way functions in the plain model.

If the parties have an access to a collision resistance hash function, we can derive *statistical everlasting security* for every NC1 functionality, i.e., the protocol is secure against adversaries that are computationally bounded during the execution of the protocol and become computationally unlimited after the protocol execution.

As a secondary contribution, we show that in the strong honest-majority setting ($t < n/3$), every NC1 functionality can be computed in 3 rounds with everlasting security and complexity polynomial in n based on one-way functions. Previously, such a result was only known based on collision-resistance hash function.

1 Introduction

Interaction is a valuable and expensive resource in cryptography and distributed computation. Consequently, a huge amount of research has been devoted towards characterizing the amount of interaction, typically measured via round

* A full version of this paper appears in [AKP21]

complexity, that is needed for various distributed tasks (e.g., Byzantine agreement [LF82, DR85, FM85], coin flipping [Cle86, MNS16], and zero-knowledge proofs [GK96, CKPR01]) under different security models. In this paper, we focus on the problem of general secure-multiparty-computation (MPC) in the challenging setting of *full security* (including guaranteed output delivery) with *maximal resiliency*. That is, even an active (aka Byzantine or malicious) adversary that controls a minority (up to half) of the parties should not be able to violate privacy or to prevent the honest parties from receiving a valid output. In this setting, originally presented in the classical work of Rabin and Ben-Or [RB89], we assume that each pair of parties is connected by a secure and authenticated point-to-point channel and that all parties have access to a common broadcast channel, which allows each party to send a message to all parties and ensures that the received message is identical.

The round complexity of honest-majority fully-secure MPC protocols was extensively studied. The lower-bound of [GIKR02, GLS15] shows that two rounds are insufficient for this task even when the parties are given access to a common reference string (CRS). In [AJL+12], a 5-round protocol was constructed based on Threshold Fully-Homomorphic Encryption (TFHE) and Non-Interactive Zero-Knowledge proofs (NIZK). An optimal round complexity of three, was later obtained by [GLS15] in the CRS model by relying on a stronger variant of TFHE that can be based on the learning with errors (LWE) assumption. Later in [BJMS20] the CRS was removed, and in [ACGJ18] LWE was replaced by weaker public-key primitives like general public-key encryption (PKE) and two-round witness indistinguishable proofs (Zaps). (The latter can be based on primitives like trapdoor permutations [DN07] and indistinguishability obfuscation [BP15], or on intractability assumptions related to bilinear groups [GOS12] and LWE [BFJ+20, GJJM20].)

The above results may give the impression that public-key assumptions are essential for honest-majority fully-secure MPC. However, if one puts no restriction on the round complexity, then, as shown by Rabin and Ben-Or [RB89], one can obtain unconditional results and no assumptions are needed at all! Specifically, every efficiently computable function can be securely computed with statistical security against computationally-unbounded adversaries.[3] Constant-round versions of this protocol are known either with an exponential dependency in the circuit-depth (or space-complexity) of the underlying function [IK00], or with computational security under the weakest-known cryptographic assumption: the existence of one-way functions [BMR90, DI05]. Moreover, for the special case of 3 parties (and single corruption), 3-round protocols were constructed by [PR18] based on injective one-way functions.

This leaves an intriguing *gap* between general-purpose *optimal-round* protocols to protocols with larger round complexity, both in terms of the underlying assumptions and with respect to the resulting security notion. We therefore ask:

[3] Interestingly, perfect security is impossible to achieve in this setting as it requires a strong honest-majority of $2n/3$ [BGW88].

Q1: Are public-key assumptions inherently needed for 3-round fully-secure honest-majority MPC? Is it possible to replace these assumptions with symmetric-key assumptions?
Q2: Is it possible to obtain 3-round fully-secure honest-majority MPC with some form of unconditional security against computationally-unbounded adversaries?

We answer these questions to the affirmative. We show that 3-round MPC with full security at the presence of honest-majority can be realized based on Minicrypt-type assumptions without relying on PKE, and present variants of our protocol that achieve *statistical everlasting security*. To the best of our knowledge, this is the first construction of everlasting-secure protocol in this setting *regardless of the underlying assumptions*. We continue with a detailed description of our results.

1.1 Our Contribution

1.1.1 Round-Optimal MPC in Minicrypt We present the first 3-round general MPC protocol under Minicrypt assumptions. In fact, our protocol consists of 1 offline (input-independent) round, and 2 online rounds. To obtain our main result, we reveal a strong connection between round-optimal MPC and round-optimal protocols for functionalities whose output depends on the input of a single party, aka *single input functionalities* (SIF). In particular, we prove the following theorem.

Theorem 1. *Assuming the existence of non-interactive commitment scheme, there exists a compiler that takes a protocol* sif *with 1 offline round and 1 online round for single input functionalities, and outputs a protocol with 1 offline round and 2 online rounds for general MPC, with the same resiliency as* sif*.*

In a recent result by the same authors [AKP22], a round-optimal SIF protocol was presented based on the existence of injective one-way functions with sub-exponential hardness. The protocol has optimal resiliency when the number of parties n is logarithmic in the security parameter, and almost-optimal resiliency when the number of parties is polynomial in the security parameter. Since injective one-way function implies the existence of perfectly-binding non-interactive commitment scheme [Blu81, Yao82, GL89], we obtain the following theorem by plugging the protocol of [AKP22] in Theorem 1.

Theorem 2. *Assuming the existence of injective one-way functions with sub-exponential hardness, for* every $\epsilon > 0$*, every efficiently-computable functionality can be realized in 1 offline round and 2 online rounds in the plain model, with full security against an active rushing adversary, under one of the following conditions.*

- (Optimal resiliency for small number of parties) *The number of parties n is at most logarithmic in the security parameter, and the adversary corrupts less than $n/2$ parties.*

– (Almost-optimal resiliency for polynomially-many parties) *The number of parties n is allowed to be polynomial in the security parameter, and the adversary corrupts less than $n \cdot (\frac{1}{2} - \epsilon)$ parties.*

In concrete terms, for an n-party functionality given by a boolean circuit C, and for security parameter κ, we derive (a) an honest majority protocol with complexity $\mathrm{poly}(|C|, \kappa)2^{O(n)}$ which is $\mathrm{poly}(\kappa)$ when $n = O(\log \kappa)$ and $|C| = \mathrm{poly}(\kappa)$; and (b) $t = n \cdot (\frac{1}{2} - \epsilon)$ resilient protocol of complexity $\mathrm{poly}(n, \kappa, |C|, 2^{1/\epsilon^2})$ which simplifies to $\mathrm{poly}(\kappa)$ when $|C| = \mathrm{poly}(\kappa)$ and $\epsilon > 0$ is an arbitrarily small constant. In fact, even if ϵ mildly *decreases* with κ, e.g., $\epsilon = \Omega(\frac{1}{\sqrt{\log \kappa}})$, the overall complexity remains polynomial. (See also the discussion in [AKP22].)

Let us further mention that two-round SIF protocols with optimal resiliency and polynomially many parties can be obtained if one is willing to make stronger assumptions (e.g., random oracle or correlation intractable functions), or if the adversary is non-rushing [AKP22]. These results extend to the MPC setting via Theorem 1.

1.1.2 Round-Optimal MPC with Everlasting Security in Minicrypt

The notion of statistical everlasting security [MU10] 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 order to extract information on the inputs of the honest parties. This feature is one of the main advantages of information-theoretic protocols: after-the-fact secrecy holds regardless of technological advances and regardless of the time invested by the adversary.

We show that Theorem 1 yields a round-optimal MPC protocol with everlasting security when it is instantiated with statistically-hiding commitments and everlasting secure round-optimal SIF protocol. Such a SIF protocol was also realized in [AKP22] based on collision-resistant hash functions. Since the latter are known to imply statistically-hiding commitments [DPP98, HM96], we derive the following theorem.

Theorem 3. *Given access to a collision resistant hash function, every* $\mathbf{NC}^1$ *functionality can be realized in 1 offline round and 2 online rounds, with full* everlasting *security against an active rushing adversary, under the same conditions of Theorem 2.*

Remark 1 (On the use of hash function). Similarly to the everlasting SIF protocol from [AKP22], our protocol assumes that all parties are given an access to a collision resistance hash function h, and we (implicitly) prove that any adversary that violates the security of the protocol can be efficiently compiled into an adversary that finds collisions in the 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.[4] One may assume that h is chosen *"once and for all"* by some simple set-up mechanism. In particular, this set-up mechanism can be realized distributively by a single round of public-coin messages by letting each party sample randomness r_i that specifies a hash function h_i and then taking h to be the concatenated hash function [Her09]. This simple set-up protocol remains secure even against an active rushing adversary that may corrupt all the participants except for a single one. Alternatively, the choice of the hash function can be abstracted by a CRS functionality, or even, using the multi-string model of [GO14] with a single honestly-generated string. 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. Finally, even if one counts this extra set-up step as an additional round, to the best of our knowledge, our protocol remains the only known solution that achieves everlasting security, regardless of the underlying assumptions.

Remark 2 (On $\mathbf{NC}^1$ *functionalities).* All our everlasting-security protocols are restricted to $\mathbf{NC}^1$. More generally, the computational complexity of these protocols grows exponentially with the depth or space of the underlying function. This is expected since even for strictly-weaker notions of security (e.g., passive statistical security against a single corrupted party), it is unknown how to construct *efficient constant-round* protocols for functions beyond $\mathbf{NC}^1$ and log-space. (In fact, this is a well-known open problem that goes back to [BFKR90].)

The difference between everlasting and computational security is *fundamental* and is analogous to the difference between statistical commitments and computational commitments or statistical ZK arguments vs. computational ZK arguments (see, e.g., the discussions in [BCC88, NOVY98]). In both the former cases, we get computational security against "online cheating" and statistical security against after-the-fact attacks.

We note that all previous protocols inherently fail to achieve everlasting security. Indeed, for technical reasons (that will be discussed later in Section 2), previous constructions emulate private channels over a broadcast channel via the use of PKE. Furthermore, the (encrypted) information that is delivered over this channel fully determines the inputs. Thus, an analyst that collects the broadcast messages and later breaks the secrecy of the PKE (e.g., via brute-force) can learn all the private inputs of the parties.

1.1.3 Round-Optimal MPC for $t < n/3$ with Everlasting Security from OWF

For strong honest-majority, where $t < n/3$, we provide a 3-round protocol for general MPC with everlasting security *in the plain model* based on the existence of one-way functions. This protocol is round-optimal by the lower bound of [GIKR02].

[4] In a uniform setting, one could use a keyless hash function; see also the discussion of Rogaway [Rog06].

Theorem 4. *Assuming the existence of one-way functions, every* $\mathbf{NC}^1$ *functionality can be realized in the plain model by a 3-round protocol that provides* everlasting security *against an active rushing adversary corrupting* $t < n/3$ *of the parties. If we are willing to compromise to computational security, we obtain a secure protocol for every efficiently computable functionality.*

Known round-optimal protocols in this regime, all appear in [AKP20], either achieve (1) statistical security but with running time exponential in n, or (2) everlasting security from collision resistant hash-functions and a CRS as a trusted setup, or (3) computational security from injective one-way function in the plain model. Therefore, our construction can be seen as the first round-optimal construction that efficiently achieves some form of security against unbounded adversaries in the plain model. Moreover, it does so only based on one-way functions. As a primary tool, we design a verifiable secret sharing (VSS) with everlasting security in 2 rounds from OWFs. Known VSS protocols in this regime either achieve (1) statistical security but with running time exponential in n with $t < n/3$ [AKP20], (2) everlasting security from collision resistant hash-functions and a CRS as a trusted setup with $t < n/2$ [BKP11], or (3) computational security from non-interactive commitments schemes with $t < n/2$ [BKP11].

1.1.4 Summary of the Results We summarize our results in the honest-majority regime in Table 1 and compare them to the existing results. In Table 2 we summarize our results in the strong honest-majority regime, and compare them to the existing results.

Ref.	Rounds	Threshold	Setup Plain / CRS	Security it / es / cs†	Cryptographic Assumptions
[RB89]	circuit-depth	$t < n/2$	Plain	it	–
[IK00]⋆	constant > 3	$t < n/2$	Plain	it	–
[BMR90, DI05]	constant > 3	$t < n/2$	Plain	cs	OWF
[PR18]	3	$n = 3, t = 1$	Plain	cs	injective OWF
[GLS15]	3	$t < n/2$	CRS	cs	threshold multi-key FHE
[BJMS20]	3	$t < n/2$	Plain	cs	LWE
[ACGJ18]	3	$t < n/2$	Plain	cs	PKE, Zaps
This	3	$t < n(\frac{1}{2} - \epsilon)$§	Plain	cs	sub-exponential injective OWF
This⋆	3	$t < n(\frac{1}{2} - \epsilon)$§	CRS	es	collision resistant hash function

† it: information-theoretic, es: everlasting security, cs: computational security.
⋆ For $\mathbf{NC}^1$ circuits
§ We achieve $t < n/2$ when n is logarithmic in the security parameter.

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

Ref.	Rounds	Threshold	Setup Plain / CRS	Security it / es / cs†	Cryptographic Assumptions	Complexity in terms of n
[AKP20]⋆	3	$t < n/3$	Plain	it	–	Exponential
[AKP20]	3	$t < n/3$	Plain	cs	injective OWF	polynomial
[AKP20]⋆	3	$t < n/3$	CRS	es	collision-resistant hash-function	polynomial
This⋆	3	$t < n/3$	Plain	es	OWF	polynomial
This	3	$t < n/3$	Plain	cs	OWF	polynomial

† it: information-theoretic, es: everlasting security, cs: computational security.
⋆ For $\mathbf{NC}^1$ circuits

Table 2: Comparison of our work with the state-of-the-art relevant results for $t < n/3$

Previous unpublished version and a sibling paper. A previous version of this paper contained a weak form of some of the current results together with 2-round SIF protocols based on the Fiat-Shamir heuristic. The SIF protocols were strengthened and were fully moved to [AKP22], and the derivation of the 3-round MPC protocols was significantly changed and modularized, leading to the new compiler (Theorem 1). Theorem 4 is also new and did not appear in previous versions. Overall, the current version of this writeup and [AKP22] contain a disjoint sets of results that together fully subsume the previous versions of this paper.

2 Technical Overview

In this section, we give a detailed overview of our constructions while emphasizing the main novelties. Section 2.1 is devoted to the proof of the main theorem (Theorem 1) and Section 2.2 is devoted to the strong honest-majority result (Theorem 4). Throughout, we assume that there are n parties, $P_1, \ldots, P_n$, of which at most t are corrupt, where we assume two settings: $t < n/2$ for Section 2.1 and $t < n/3$ for Section 2.2. We assume that the parties communicate over secure point-to-point channels and over a broadcast channel.

2.1 Main Theorem

Our goal is to prove our main Theorem 1, that states that assuming the existence of non-interactive commitments we can transform any sif protocol with 1 offline round and 1 online round into a 3 round protocol for general MPC with the same resiliency as sif. Following previous works [GLS15, ACGJ18], we prove Theorem 1 by using the following outline: (1) We start with a 2 round protocol Π^{sm} with security against *semi-malicious* adversary that is allowed to choose its input and randomness, but other than that plays honestly; (2) We upgrade the security of the protocol to hold against a *first-round fail-stop* adversary that, in addition to choosing its input and randomness, is allowed to abort a corrupted

party during the first round of the protocol; (3) We compile the protocol to a new protocol with an extra offline round that achieves security against a *fully fail-stop* adversary that is allowed to abort a corrupted party at any round; (4) We transform the protocol for fail-stop adversaries to a protocol for malicious adversaries. Jumping ahead, previous constructions employed Zaps/NIZK for the last step and PKE/threshold homomorphic encryption both for steps (3) and (4). We will show how to relax these assumptions.

The initial protocol Π^{sm}. Our starting point is a 2-round protocol Π^{sm} that is secure against a rushing semi-malicious adversary that corrupts a minority of the parties. For concreteness, we use the protocol of [ABT18], though any other protocol could be used. This protocol provides *perfect security* for $\mathbf{NC}^1$ functionalities and *computational security* for $\mathbf{P}$/poly functionalities, assuming the existence of one-way functions. The protocol is fully describe in the full version of this paper [AKP21]. The first round of the protocol consists only of private messages, and the second round consists of broadcast messages. (In fact, using standard techniques we can transform any 2-round protocol to a protocol that satisfies this property, see e.g., [GIKR01].) We denote the first-round private message from P_i to P_j by a_{ij}, and the second-round broadcast of P_i by b_i.

2.1.1 Coping with First-Round Aborts Roughly speaking, when an adversary aborts, we let the other parties emulate her role for the remaining rounds. The emulation is relatively simple when the abort happens in the first round of Π^{sm} since the parties have a chance to respond to the abort in the second round. Specifically, suppose that P_i aborts in the first round. Then the other parties face 2 problems: (1) P_i did not send her first round messages; and (2) the first-round messages that were *directed* to P_i were lost and will be missing later during the reconstruction of output. The first issue is solved by letting each party to locally generate the outgoing messages of P_i by running P_i on the all-zero input and the all-zero random tape.[5] To solve the second issue, we modify the protocol so that each first round message from P_j to P_i is also being shared among all other parties. That is, in the first round, every P_j shares each of its first-round outgoing messages $a_{j1}, \ldots, a_{jn}$ via Shamir's secret sharing, using degree-t polynomials. If P_i aborts during the first round then in the second round, the parties reconstruct all the 1st round incoming messages of P_i. After the second round, the parties have enough information to locally continue the emulation of P_i (with respect to the all-zero inputs) and generate her second round broadcast messages. We note that in previous works (e.g., [ACGJ18]) first-round aborts are handled differently by adding an additional "function-delayed" requirement on the initial protocol Π^{sm}, and that this property is not required for our compiler.

2.1.2 Coping with Second-Round Aborts Second-round aborts are trickier to handle: When the honest parties send their second-round messages, they

[5] Here, among other places, we use the fact that Π^{sm} is secure against a semi-malicious adversary.

do not know which other parties are about to abort. Accordingly, one has to support "silent emulation", that is, any subset of $n - t$ second-round messages should suffice for emulating all other second-round messages. In previous works, the implementation of this mechanism employs heavy tools (threshold homomorphic encryption in [GLS15] and PKE plus garbled circuits in [ACGJ18]) and requires an additional offline round. We review these ideas and present an information-theoretic variant of them.

Ananth et al. [ACGJ18] (ACGJ) first use PKE to ensure that all the communication between the parties will be over the broadcast channel. That is, in a preprocessing round (denoted Round 0), every P_i generates keys $(\mathsf{pk}_i, \mathsf{sk}_i)$ for PKE, and broadcasts pk_i. In the following rounds, the private channel from P_j to P_i is emulated by letting P_j broadcast her message encrypted under the public key pk_i of P_i. After this modification, we can write the second-round message of party P_i as a function f_i that given

(1) the encrypted messages $(A_{ji})_{j\in\{1,\ldots,n\}}$ that P_i receives in Round 1,
(2) the input $\mathbf{x}(i)$ and randomness r_i of P_i in the simulation of Π^{sm}, and
(3) the secret key sk_i,

outputs the public broadcast message b_i that P_i sends in the second round. (That is, f_i decrypts the messages A_{ji} using sk_i in order to obtain a_{ji}, and then computes the second round broadcast b_i of P_i in Π^{sm} based on $(\mathbf{x}(i), r_i, (a_{ji})_{j\in\{1,\ldots,n\}})$.) Observe that f_i depends on private inputs (items 2, 3) and on some public values (item 1) that will be broadcasted during the first round. The key observation is that the private inputs are already known before the first round begins. This fact will be exploited to delegate the computation of f_i.

Specifically, at the beginning of the first round, we let every P_i generate a *garbled circuit* for a function f_i. During the first round, P_i broadcasts the garbled circuit together with the labels of $(\mathbf{x}(i), r_i)$ and sk_i. In addition, P_i secret-shares all the labels that correspond to *every potential* ciphertext value $(A_{ji})_{j\in[n]}$. The actual ciphertexts, $(A_{ji})_{j\in\{1,\ldots,n\}}$, are broadcasted concurrently during the first round by the corresponding parties, and so, in the second round, all the non-aborted parties publish the shares of the corresponding labels. Consequently, after this round, everyone can recover the correct labels via secret reconstruction of the secret sharing, and hence obtain the broadcast b_i of P_i. To make the proof go through, ACGJ assume that the garbled circuit is *adaptively* private [BHR12] in the sense that privacy holds even if the adversary first gets to see the garbled circuit, and only then chooses the inputs to the circuit and receive the corresponding labels.

We note that the same approach can be applied without relying on any computational assumptions. First, instead of using PKE, we let the parties exchange one-time pads during the offline round. That is, in Round 0 we let every P_i sample random pads $\eta_i = (\eta_{i1}, \ldots, \eta_{in})$ and send the pad ("key") η_{ij} to P_j by using a *private channel*. Now a first-round message a_{ji} from P_j to P_i can be broadcasted in an encrypted form $A_{ji} := a_{ji} + \eta_{ij}$. (For technical reasons that will be explained later, we encrypt the message under the receiver's key.) The garbled

circuits can also be instantiated with an information-theoretic garbled circuits, aka perfect randomized encodings. (The second-message function of Π^{sm} is now "simple enough" to allow such a realization.) Furthermore, we avoid the need for adaptive garbled circuits, by sharing the garbled circuit together with the labels of $(\mathbf{x}(i), r_i)$ and η_i among all the other parties; these shares are later revealed during the second round.[6] We note that the above description is over-simplified and, in order to handle second-round aborts *together* with first-round aborts, we need to slightly modify the function f_i. (See the full version of this paper [AKP21] for full details.)

2.1.3 From Fail-Stop to Malicious Adversary To obtain a protocol with security against a malicious adversary, we follow the GMW paradigm and ask each party to prove in zero-knowledge that she followed the protocol. Ignoring for now the exact details of the zero-knowledge proof, the basic idea is that a malicious deviation from the protocol will be caught due to the soundness properties of the proof, and will be treated as if the cheater aborted the computation. Crucially, here too one must assume that the underlying protocol works over a *broadcast* channel. As discussed in [ACGJ18], if the underlying semi-malicious protocol uses private channels, then a party may need to prove different statements to different parties in order to establish honest behavior, which may lead to inconsistent views regarding her "abort" status. Indeed, [GLS15, ACGJ18] make here another use of PKE in order to make sure that the protocol's messages are delivered over a broadcast channel. In fact, this usage of PKE dates back to the GMW compiler [GMW87].

Generating public committing transcript. We can use the previous maneuver to shift all private messages to Round 0 via one-time pads, however, the resulting protocol is still not ready for "zero-knowledge compilation". Indeed, even if we add a zero-knowledge layer, the adversary can cheat either by "claiming that she received different messages" (i.e., changing the keys that correspond to her incoming messages) or by "claiming that she sent different messages". Intuitively, the problem is that our information-theoretic solution is non-committing. We solve this problem via the use of non-interactive commitment (NICOM). Details follow.

In the preprocessing round (Round 0), we let each party P_i broadcast a vector of commitments, $(C_{i1}, \ldots, C_{in})$ to all her private keys, $(\eta_{i1}, \ldots, \eta_{in})$, for the one-time pads, and send o_{ij}, the opening of C_{ij}, to P_j over the private channel. In addition, we let all parties commit to their inputs and randomness for the fail-stop protocol in Round 1 just like in the standard GMW transform. (We

[6] We note that [ACGJ18] implicitly shared the garbled circuit as well. Indeed, recall that they (a) shared the "input labels" and (b) employed the adaptively secure garbled circuit from [BHR12]. The latter is obtained by taking a standard garbled circuit and encrypting the offline part under a one-time pad that is released as part of the online input. The combination of these two steps, (a) and (b), indirectly induces (a somewhat complicated) secret sharing of the garbled circuit and the input labels.

emphasize that Round 0 is still input-independent.) Next, we employ some zero-knowledge primitive (to be discussed below) to prove that a party P_i computes a message properly with respect to the public commitments. Specifically, in the first round party P_i can prove that the garbled circuit for f_i was generated properly with respect to her committed randomness, committed input, and with respect to the one-time keys, $\eta_{1i}, \ldots, \eta_{ni}$, that he received from all other parties in the preprocessing round. For the last part we exploit the fact that P_i also received a witness, o_{ji}, that connects the keys to their commitments.

This approach almost works. The only problem is that a party P_j may cheat in Round 0 by sending to P_i a "bad" pair of key/opening (η_{ji}, o_{ji}) that are inconsistent with the public commitment C_{ij}. Fortunately, there is a simple round-efficient solution: If the key is malformed, we simply send the messages from P_i to P_j in the clear un-encrypted. Formally, in Round 1, P_i broadcasts a list L_i of all parties that sent *invalid* openings in Round 0. If P_i needs to send a private message a_{ij} to a party P_j according to Π^{sm}, for $P_j \notin L_i$, then P_i simply sends the encrypted message $a_{ij} + \eta_{ji}$ over the broadcast channel. For a party $P_j \in L_i$, we simply let P_i send the message a_{ij} *unencrypted* over the broadcast channel. We also use the same mechanism for additional private messages that the parties have to exchange, that are not necessarily a part of the protocol Π^{sm} (e.g., sending private shares for the garbled circuit). As before, we only use encryption in Round 1, while Round 2 consists only of public unencrypted messages. This modification does not violate privacy since messages from P_i to P_j will be sent unencrypted only if one of these parties is corrupted, which means that the adversary is supposed to learn the message anyway.

Instantiating the zero-knowledge layer. Finally, we have to instantiate the zero-knowledge layer in a round-preserving way. Previous works either make use of NIZK at the expense of adding a CRS [AJL+12, GLS15] or exploited the offline round to set-up some multi-party variant of ZK [GOS12, ACGJ18]. In terms of assumptions both approaches rely on NIZK/Zaps which are known to be equivalent assuming one-way functions [DN07]. We strongly exploit the existence of honest majority, and observe that these primitives can be replaced by a SIF protocol. Given a relation R, define the single input functionality that (1) takes the statement x and witness w from the prover, and (2) if $R(x, w) = 1$ it returns x to all parties, and if not, it returns a failure symbol $\perp$ to all parties. We can therefore realize a round-efficient variant of multi-verifier zero-knowledge proof (MVZK) based on SIF with 1 offline round and 1 online round. We emphasize that the security of SIF protocols is formulated via an MPC-based definition by relating the protocol to an *ideal SIF functionality*. This leads to security guarantees that are stronger than those achieved by standalone versions of the MVZK primitive (e.g., the SIF protocol provides *knowledge-extraction*).

Summary. Overall, the SIF is being employed as follows. In Round 0, the parties execute the offline round of the SIF protocol, exchange one-time pads and publish their commitments. In Round 1, we let every P_i commit to its input and randomness, and let P_i prove via SIF that (1) for every $P_j \notin L_i$, the public

encrypted message from P_i to P_j is consistent with the committed input and randomness of P_i, and it is encrypted with the committed random pad η_{ji}; (2) for every $P_j \in L_i$, the public unencrypted message from P_i to P_j is consistent with the committed input and randomness of P_i. Similarly, in Round 2 every P_i proves via SIF that its public broadcast is consistent with (1) its committed input and randomness; (2) the unencrypted public incoming message from P_j, for every P_j for which $P_i \in L_j$; and (3) the decrypted incoming message from P_j, where the decryption used the committed random pad η_{ij}, for every P_j for which $P_i \notin L_j$.

Remark 3 (Everlasting security). All the components, except for the NICOM and SIF, are information-theoretic. As a result, we derive the everlasting security version of the protocol by plugging-in NICOM and SIF with everlasting security guarantees. The protocol remains the same and the proof of security is given in a unified way.

Remark 4 (Reusing the preprocessing round). Recall that the preprocessing round consists of exchanging committed one-time pads, and initializing the SIF protocol. If one does not care about everlasting security, the one-time pads can be replaced with (committed) pairwise private-keys for a symmetric encryption scheme, and in this case the same keys can be used for many invocations of the protocol. Under this modification, we can reuse the preprocessing step (Round 0) or even treat it as a private-key infrastructure provided that the preprocessing step of the SIF is also reusable. While the construction from [AKP22] does not satisfy this property, other SIF constructions (e.g., based on NIZK) can be used to achieve this property. We remark that, even if one employs NIZK-based SIF, our approach is beneficial since it bypasses the need for PKE. Indeed, the Fiat-Shamir heuristic [FS86] suggests that NIZK can be based on strong symmetric-key assumptions like correlated robust hash functions [CGH04], and may not require PKE-based assumptions. (See [CCH+19] for further discussion and references).

Remark 5 (On non black-box use of the commitment scheme). Observe that our compiler uses the underlying commitment scheme in a non black-box way. This is a common characteristic of GMW-type compilers, where the zero-knowledge proofs use the underlying cryptographic primitives in a non black-box way, and it occurs in previous round-optimal protocols as well, including [ACGJ18].

2.2 Strong Honest-majority MPC with Everlasting Security from OWF

We continue with an overview of the 3-round MPC protocol that provides everlasting security in the plain model for strong honest-majority, $t < n/3$. In [AKP20] it is shown that such a protocol follows from a 2-round protocol for *verifiable secret sharing* (VSS) that provides everlasting security. We design such a protocol based on digital signatures whose existence is equivalent to the existence of one-way functions [Rom90].

The VSS functionality. We will need the following variant of VSS.[7] The functionality receives a symmetric bivariate polynomial $F(x, y)$ of degree at most t in each variable from a distinguished party D, called the *dealer*, and delivers to each party P_i the univariate polynomial $f_i(x) := F(x, i)$. The use of symmetric bivariate polynomials can be seen as an extension of the standard Shamir's t-out-of-n secret sharing, that allow us to make a consistency-check between any pair of parties P_i and P_j, since $f_i(j) = F(j, i) = F(i, j) = f_j(i)$.

2-round VSS protocol. In the first round, we let D generate a signature-key and a verification-key for a digital signature scheme, and broadcast the verification-key. In addition, we let D send $f_i(x)$ to P_i, together with a signature on the *tuples* $(i, j, f_i(j))_{j\in\{1,\ldots,n\}}$. At the end of the first round, a party is *happy* with D if all the signatures it received are valid, and it is *unhappy* with D otherwise. Observe that if D is honest then all honest parties are happy. The second round of the protocol consists of (1) consistency check for happy parties, and (2) public recovery of the shares of unhappy parties. We elaborate on these two issues in the next subsections.

2.2.1 Consistency Check The goal of the consistency check is to ensure that (a) there are at least $t + 1$ happy honest parties, and that (b) all of them are consistent with each other, i.e., $f_i(j) = f_j(i)$ for every happy and honest P_i and P_j. Looking forward, this will imply that the shares of the happy honest parties fully determine a symmetric bivariate polynomial $F(x, y)$ of degree at most t in each variable, where for an honest D the polynomial $F(x, y)$ is the input polynomial of D.

It is not hard to achieve (a). In Round 2, each party declares, via broadcast, whether she is happy or not, and we discard the dealer if there are more than t unhappy parties. This guarantees that an honest dealer will never be discarded (since all honest parties are happy) and a corrupt dealer must gain the support of at least $(n-t)-t \geq t+1$ happy honest parties in order to remain undiscarded.

2-wise consistency via Reveal-if-not-equal gadget. Pair-wise consistency (item b) is being handled via a special comparison gadget that takes from each pair of happy parties (P_i, P_j) the points $m_A = f_i(j), m_B = f_j(i)$ and their corresponding signatures s_A, s_B, and broadcasts an equality bit that indicates whether $m_A = m_B$ and in case of inequality releases the points and their signatures (m_A, s_A, m_B, s_B). When P_i and P_j are honest, a disagreement accompanied with valid signatures certifies that D is corrupted. Of course, when $m_A = m_B$, we do not want any information about m_A, m_B to be revealed to the other parties. If 3 rounds are allowed then we can easily realize the gadget by letting P_i and P_j compare their values privately on the second round (by exchanging messages

[7] Previous works on VSS [CGMA85] usually define VSS as a standalone primitive that satisfies a set of requirements (see, e.g., [KKK09, BKP11]). Following [AKP20] (see also [AL17]) we consider VSS as an *ideal functionality*. We mention that any VSS that satisfies the ideal-functionality definition also satisfies the standalone definition.

over the private channel) and then announcing the result at the next round. We avoid this overhead by making an additional observation: When one of the parties, say P_i, is corrupt we do not care about the privacy nor the correctness of the gadget. Privacy does not matter since the adversary already knows $m_B = f_j(i)$. As for correctness, even if the "gadget misbehaves", an honest dealer is protected against a disqualification by the security of the signatures.

We realize the gadget with the aid of garbled circuits (or perfect randomized encodings). Let g be a function that takes (m_A, m_B, s_A, s_B), returns 1 if $m_A = m_B$, and returns (m_A, m_B, s_A, s_B) otherwise. In the first round, we let Alice (P_i) generate a garbled circuit G for g, and send the randomness used to generate G to Bob (P_j). Conveniently, g is "simple enough" (i.e., an $\mathbf{NC}^1$ function) so we can obtain an *information-theoretic* garbled circuit G. In the second round, Alice broadcasts G, together with the labels corresponding to her inputs in G, and Bob broadcasts the labels corresponding to his inputs in G. It is not hard to see that the properties of the protocol follow directly from the correctness and security of the garbled circuit. Based on this gadget, after the second round everyone learns whether Alice and Bob are in agreement, and, in case they disagree, whether the dealer should be discarded due to a conflicting pair of valid signatures. If the dealer was not discarded in any consistency check of a pair (P_i, P_j), we conclude that all happy honest parties are consistent.

2.2.2 Handling Unhappy Parties It remains to explain how to help unhappy (honest) parties to recover a share that is consistent with all the happy honest parties. The main idea is to let every unhappy P_i ask from every other P_j to publicly reveal all the common information, i.e., the value $f_j(i)$ and the corresponding signature. Since we have only 1 additional round, we design an additional gadget with 1 offline round and 1 online round similarly to the reveal-if-not-equal gadget.[8] In this gadget, Alice inputs a bit flag_A, while Bob inputs some secret s_B. When Alice and Bob are honest, if $\mathsf{flag}_A = 0$ then the listeners learn no information about s_B, while if $\mathsf{flag}_A = 1$ they learn s_B. As before, when one of the parties is corrupt there are no security guarantees.

We use this mechanism for every pair (P_i, P_j), where P_i takes the role of Alice and P_j takes the role of Bob. We let P_i input $\mathsf{flag}_A = 1$ if P_i is unhappy, and $\mathsf{flag}_i = 0$ otherwise; in addition, P_j sets s_B to be the share $f_j(i)$ together with the corresponding signature. Observe that if both P_i and P_j are honest and happy, then the adversary learns no information about their common point; however, if P_i is unhappy and P_j is happy, then *all the parties* learn the point $f_j(i)$ together with a valid signature.

An honest unhappy P_i will be able to reveal all evaluations $f_j(i)$ from happy honest parties P_j, together with valid signatures. We let *all* parties interpolate over all values whose corresponding signatures were valid, in order to obtain $f_i(x)$. Since there are at least $t + 1$ happy honest parties, we are promised that $f_i(x)$ is either consistent with the polynomial $F(x, y)$ defined by the shares of the happy honest parties, or has degree more than t, in which case *all* the parties

[8] In fact, in our construction we merge the two gadgets.

reject the dealer. Finally, for an honest D and a corrupt unhappy P_i, the values that are revealed with valid signatures must be consistent with $F(x, y)$, so the interpolated polynomial will have degree at most t, and D will not be discarded.

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

References

ABT18. Benny Applebaum, Zvika Brakerski, and Rotem Tsabary. Perfect secure computation in two rounds. In *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part I*, pages 152–174, 2018.

ACGJ18. Prabhanjan Ananth, Arka Rai Choudhuri, Aarushi Goel, and Abhishek Jain. 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*, pages 395–424, 2018.

AJL+12. Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 483–501, 2012.

AKP20. Benny Applebaum, Eliran Kachlon, and Arpita Patra. The resiliency of MPC with low interaction: The benefit of making errors (extended abstract). In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 562–594. Springer, 2020.

AKP21. Benny Applebaum, Eliran Kachlon, and Arpita Patra. Round-optimal honest-majority MPC in minicrypt and with everlasting security. *IACR Cryptol. ePrint Arch.*, 2021:346, 2021. https://eprint.iacr.org/2021/346.

AKP22. Benny Applebaum, Eliran Kachlon, and Arpita Patra. Verifiable relation sharing and multi-verifier zero-knowledge in two rounds: Trading nizks with honest majority. Cryptology ePrint Archive, Report 2022/167, 2022. https://ia.cr/2022/167, To appear in CRYPTO 2022.

AL17. Gilad Asharov and Yehuda Lindell. A full proof of the BGW protocol for perfectly secure multiparty computation. *J. Cryptology*, 30(1):58–151, 2017.

BCC88. Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

BFJ+20. Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on*

the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III, pages 642–667, 2020.

BFKR90. Donald Beaver, Joan Feigenbaum, Joe Kilian, and Phillip Rogaway. Security with low communication overhead. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pages 62–76, 1990.

BGW88. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. 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*, pages 1–10, 1988.

BHR12. Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Adaptively secure garbling with applications to one-time programs and secure outsourcing. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 134–153. Springer, 2012.

BJMS20. Saikrishna Badrinarayanan, Aayush Jain, Nathan Manohar, and Amit Sahai. Secure MPC: laziness leads to GOD. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part III*, volume 12493 of *Lecture Notes in Computer Science*, pages 120–150. Springer, 2020.

BKP11. Michael Backes, Aniket Kate, and Arpita Patra. 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*, pages 590–609, 2011.

Blu81. Manuel Blum. 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.*, pages 11–15, 1981.

BMR90. Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 503–513, 1990.

BP15. Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 401–427. Springer, 2015.

CCH+19. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1082–1090. ACM, 2019.

CGH04. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

CGMA85. Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. 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*, pages 383–395, 1985.

CKPR01. Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires omega˜(log n) rounds. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 570–579, 2001.

Cle86. Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 364–369, 1986.

DI05. Ivan Damgård and Yuval Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, pages 378–394, 2005.

DN07. Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

DPP98. Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Trans. Inf. Theory*, 44(3):1143–1151, 1998.

DR85. Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. *J. ACM*, 32(1):191–204, 1985.

FM85. Paul Feldman and Silvio Micali. Byzantine agreement in constant expected time (and trusting no one). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 267–276, 1985.

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

GIKR01. Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. The round complexity of verifiable secret sharing and secure multicast. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 580–589, 2001.

GIKR02. Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. On 2-round secure multiparty computation. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 178–193, 2002.

GJJM20. Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III*, pages 668–699, 2020.

GK96. Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

GL89. Oded Goldreich and Leonid A. Levin. 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*, pages 25–32, 1989.

GLS15. S. Dov Gordon, Feng-Hao Liu, and Elaine Shi. Constant-round MPC with fairness and guarantee of output delivery. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 63–82, 2015.

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to solve any protocol problem. In *Proc. of STOC*, 1987.

GO14. Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. *Journal of cryptology*, 27(3):506–543, 2014.

GOS12. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.

Her09. Amir Herzberg. Folklore, practice and theory of robust combiners. *Journal of Computer Security*, 17(2):159–189, 2009.

HM96. Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 201–215. Springer, 1996.

IK00. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304, 2000.

KKK09. Jonathan Katz, Chiu-Yuen Koo, and Ranjit Kumaresan. Improving the round complexity of VSS in point-to-point networks. *Inf. Comput.*, 207(8):889–899, 2009.

LF82. Leslie Lamport and Michael Fischer. Byzantine generals and transaction commit protocols. Technical report, Technical Report 62, SRI International, 1982.

MNS16. Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. *J. Cryptology*, 29(3):491–513, 2016.

MU10. Jörn Müller-Quade and Dominique Unruh. Long-term security and universal composability. *J. Cryptol.*, 23(4):594–671, 2010.

NOVY98. Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for *NP* using any one-way permutation. *J. Cryptol.*, 11(2):87–108, 1998.

PR18. Arpita Patra and Divya Ravi. 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*, pages 425–458, 2018.

RB89. Tal Rabin and Michael Ben-Or. 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*, pages 73–85, 1989.

Rog06. Phillip Rogaway. Formalizing human ignorance: Collision-resistant hashing without the keys. *IACR Cryptol. ePrint Arch.*, page 281, 2006.

Rom90. John Rompel. One-way functions are necessary and sufficient for secure signatures. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM*

Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA, pages 387–394. ACM, 1990.

Yao82. Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91, 1982.