Constructing Non-Malleable Commitments: A Black-Box Approach

Vipul Goyal
Microsoft Research
INDIA
vipul@microsoft.com

Chen-Kuei Lee
UCLA
USA
jcklee@ucla.edu

Rafail Ostrovsky
UCLA
USA
rafail@cs.ucla.edu

Ivan Visconti
Università di Salerno
ITALY
visconti@dia.unisa.it

***Abstract*—We propose the first black-box construction of non-malleable commitments according to the standard notion of non-malleability with respect to commitment. Our construction additionally only requires a constant number of rounds and is based only on (black-box use of) one-way functions. Prior to our work, no black-box construction of non-malleable commitments was known (except for relaxed notions of security) in any (polynomial) number of rounds based on any cryptographic assumption. This closes the wide gap existent between black-box and non-black-box constructions for the problem of non-malleable commitments.**

Our construction relies on (and can be seen as a generalization of) the recent non-malleable commitment scheme of Goyal (STOC 2011). We also show how to get black-box constructions for a host of other cryptographic primitives. We extend our construction to get constant-round concurrent non-malleable commitments, constant-round multi-party coin tossing, and non-malleable statistically hiding commitments (satisfying the notion of non-malleability with respect to opening). All of the mentioned results make only a black-box use of one-way functions.

Our primary technical contribution is a novel way of implementing the proof of consistency typically required in the constructions of non-malleable commitments (and other related primitives). We do this by relying on ideas from the "zero-knowledge from secure multi-party computation" paradigm of Ishai, Kushilevitz, Ostrovsky, and Sahai (STOC 2007). We extend in a novel way this "computation in the head" paradigm (which can be though of as bringing powerful error-correcting codes into purely computational setting). To construct a non-malleable commitment scheme, we apply our computation in the head techniques to the recent (constant-round) construction of Goyal. Along the way, we also present a simplification of the construction of Goyal where a part of the protocol is implemented in an information theoretic manner. Such a simplification is crucial for getting a black-box construction. This is done by making use of pairwise-independent hash functions and strong randomness extractors.

We show that our techniques have multiple applications, as elaborated in the paper. Hence, we believe our techniques might be useful in other settings in future.

***Keywords*-non-malleable commitments; black-box use of cryptographic primitives; computation in the head paradigm;**

I. Introduction

The notion of non-malleable commitments was introduced in the seminal work of Dolev, Dwork and Naor [1] and has been widely studied since then. Non-malleable commitments (and related primitives like non-malleable zero-knowledge) form the foundations of modern techniques for dealing with man-in-the-middle attacks in cryptographic protocols. Man-in-the-middle attacks could be of concern either if there is a single protocol execution with multiple parties (e.g., non-malleable commitments have been useful in constructing round-efficient multi-party computation protocols [2], [3], [4], [5], [6], [7]), or, when there are several executions. There has been a large body of literature on constructing protocols in the concurrent setting (c.f., the lines of works on getting concurrent security in the plain model [8], [9], [10], [11], [12], [13], [14], [15], [16] and on getting universally composable protocols in various settings [17], [18], [19], [20]). Many of these works use non-malleable protocols in some form as a crucial technical tool.

After the initial feasibility results by Dolev et. al., a fruitful line of research has focused on efficiency. Round complexity, a natural measure of efficiency has been studied in several works. Barak, in a breakthrough work [2] gave the first constant-round construction of non-malleable commitments using the so called non-black-box simulation techniques [21]. Since then, a number of works have investigated the round complexity of non-malleable protocols. There have been super-constant-round protocols based on one-way functions [5], [6]. Constant-round protocols using non-standard or sub-exponential hardness assumptions were proposed in [22], [23]. Constant-round protocols using *non-black-box simulation* techniques can be found in [2], [24], [25], [26], [27]. Very recently, constant-round constructions based only on one-way functions (OWF) (with black-box simulation techniques) were proposed independently by Goyal [7] and Lin and Pass [28]. In all of these works, constructions according the the traditional security notion (of non-malleability with respect to commitment) make a non-black-box use of underlying cryptographic primitives.

While round complexity is an important measure of efficiency, a fundamental step in obtaining efficient protocols is to obtain a *black-box construction* (i.e., one where the underlying cryptographic primitives is used only as an oracle). Construction making use of the underlying primitive in a non-black-box way can typically only be seen as a feasibility result (regardless of the round complexity).

Obtaining black-box constructions for various cryptographic primitives has been an active line of research in recent years (c.f., [29], [30], [6]). However the state of art on constructing non-malleable commitments making a black-box use of cryptographic primitives is far from satisfactory. There have only been results according to new and relaxed notions of security [30], [6], [7] (see the end of this section for a more detailed discussion). To summarize, there is sharp contrast in what is known using non-black-box construction (constant-round protocols using only one-way functions) and black-box constructions (no construction known as per the traditional definition) for the problem of non-malleable commitments. This raises the following natural question:

Does there exist a black-box construction of non-malleable commitments following the traditional security notion [1], [25], [31] from any cryptographic assumption with any round complexity?

The main difficult in resolving the above question seems to be in developing a cut and choose technique having the appropriate coding theoretic properties [30], [6].

A. Our Results

We solve the above question in the affirmative by providing a black-box construction of non-malleable commitments. Our construction follows the traditional notion of non-malleability with respect to commitment [1], [25], [31]. Our construction uses only a constant number of rounds and is based only on (a black-box use of) one-way functions. This completely closes the wide gap between the state of knowledge between black-box and non-black-box constructions for non-malleable commitments. Our construction relies on (and can be seen as an instantiation of) the recent non-malleable commitment scheme of Goyal [7]. Our key technical contribution relates to the construction of a commitment scheme which allows one to prove any arbitrary relation over the committed values in zero-knowledge in a black-box manner (which we use in the commitment scheme of Goyal [7]).

Once we obtain such a construction, black-box constructions for several other primitives can be obtained in a natural way. We generalize our construction to get *concurrent* non-malleable commitments. This construction is constant-round as well and is based on one-way functions. We obtain constant-round multi-party coin tossing (with a broadcast channel) based only on a black-box use of one-way functions. This is a direct improvement over the work of Pass and Wee [30] which provided such a construction only for the two-party case (indeed, for the case of two parties, one does not run into issues of man-in-the-middle attacks). We also provide a black-box construction of non-malleable statistically hiding commitments (satisfying the notion of non-malleability with respect to opening [24][1]). Our construction builds on a stand-alone statistically hiding commitment and converts it into a non-malleable statistically hiding commitment. This allows us to get a non-malleable statistically hiding commitment in constant rounds based on (a black-box use of) collision resistant hash functions. Furthermore, one can also have a construction based only on one-way functions in $O(n/\log(n))$-rounds. To our knowledge, this is the first black-box construction of non-malleable statistically hiding commitments.

Along the way we also give several results of independent interest most notably a black-box non-malleability amplification preserving security against general (non-synchronizing) adversaries. This is an improvement over the analogous result of Wee [6] which required non-black-box access to a one-way function.

B. Technical Overview

Traditionally, constructions of non-malleable commitment schemes have relied on executing a basic protocol block somehow several times and then proving consistency among all of them. The proof of consistency typically makes use of underlying cryptographic primitives in a non-black-box way. The question of constructing non-malleable commitments in a black-box way has been raised in a number of previous works [5], [30], [6], [7]. The main difficulty encountered in previous works is in coming up with a cut-and-choose technique having the right properties to replace the zero-knowledge proof of consistency.

Our main technical construction of this work is a novel way of implementing the zero-knowledge proof of consistency that is typically required in non-malleable commitment protocols. Our technique is based on ideas from the "zero-knowledge from secure multi-party computation" paradigm of Ishai, Kushilevitz, Ostrovsky, and Sahai [32]. In this paradigm, we have a prover who runs a multi-party computation protocol "in his head" and proves the correctness of the result to the verifier. This form of computation in the head approach was proposed in [32] in the context of improving the communication complexity of zero-knowledge protocols. Our goal and the way we use these ideas are somewhat different. Our basic idea will be as follows.

Suppose one needs to commit to a set of strings $S = (s_1, \ldots, s_n)$ (and prove a statement about these strings later on). The committer starts to emulate k virtual players "in his head". Each player is given as input a share of S. Secret sharing is done using a verifiable secret sharing scheme (see, e.g., [33]). Let the view of the players so far be $view_1^0, \ldots, view_k^0$ respectively. The committer commits to these views using a regular computationally secure commitment scheme.

At a later point in the interaction, suppose the committer needs to perform some computation f on the committed strings, reveal the result $f(S)$ to the verifier and prove its correctness. This can now be done as follows.

- The committer continues to emulate the k virtual player in his head. The players will now compute the following

[1]For statistically hiding commitments, the notion of non-malleability with respect to commitment is meaningless as the committed value is not well defined. To analyze security in such a setting, the standard notion of non-malleability is with respect to opening as studied for instance in [24].

functionality: the functionality will take the share of each player, reconstruct the set S and output $f(S)$ to each player.

- The players jointly run a secure computation protocol (starting with the views already committed to) to compute this functionality. The secure computation protocol being used is information theoretically secure tolerating up to a constant fraction of corrupted parties such as [34].
- Let the new views of the players up to this point be $view_1^1, \ldots, view_k^1$. The committer now reveals $f(S)$ and commits to these new views.
- The receiver chooses a constant fraction of the players at random. The prover decommits to both the views for the selected players. This includes the initial views $view_i^0$ as well as the new views $view_1^1$.
- The receiver checks if the players behaved honestly during the entire computation and that their views are "consistent" with each other (see Section II for the precise notion of consistency). If this check is successful, "most" of the virtual players were correctly emulated by the committer. Hence the output of the computation must be correct (since the protocol anyway tolerates a constant fraction of corrupted players).
- The security of the committer is also preserved since revealing views for a constant fraction of players does not reveal anything about the set of strings S that he started with (other than of course the output $f(S)$).

The key difference from [32] is that in our setting, the statement we are proving actually inherently involve a non-black-box use of the commitment scheme: "the evaluation of f on the set of committed values results in $f(S)$". We are able to resolve this issue by using an extension of the computation in the head paradigm to multiple "consistent" execution [35]. Our technique can also be seen as a way of proving a secret but committed statement (in a way that does not involve the circuit of the commitment scheme in a non-black-box way).

We believe our technique to be of independent interest. The above technique might allow us to obtain black-box constructions by eliminating zero-knowledge proofs of consistency in other settings as well. Notice that we use a non-constant-round multi-party computation protocol such as [34] in our construction. Indeed, obtaining constant-round information theoretically secure multi-party computation is currently a major open problem connected to the existence of short locally decodable codes [36]. However our final protocol is still constant-round since this computation needs to be only done "in the head" of the committer.

To construct non-malleable commitments in a black-box manner, our starting point is the recent constant-round protocol of Goyal [7] (which makes use of one-way functions in a non-black-box manner). Goyal's protocol has a zero-knowledge proof of consistency (on committed strings) which we implement using the above secure computation in the head approach. However we note that the protocol of Goyal, very informally, is still too "non-black-box" to admit a simple application of this idea. The protocol of Goyal uses a proof of complex statements involving the randomness with which a commitment is constructed. We present a simplification of the non-malleable commitment scheme of Goyal. Our simplification involves making a part of the protocol *purely information theoretic* using pairwise-independent hash functions and strong randomness extractors. This is done in a way such that the proof of non-malleability (with respect to commitment) still goes through.

We are finally left with a protocol where the only computational part is an initial commitment to a set of random strings such that the consistency proof only needs to prove a statement above the committed strings. Our MPC in the head technique discussed about is powerful enough to handle such a scenario.

C. Related Work

Di Crescenzo, Ishai and Ostrovsky presented in [39] the first non-interactive non-malleable commitment scheme in the common reference string model. Pass and Wee [30] gave a construction of a non-malleable commitment scheme with respect to commitment in $O(\log(n))$ rounds (and in $O(n)$ rounds for concurrent non-malleable commitments) making a black-box use of one-way functions in the standard model. Their construction is according to a relaxed security notion called non-malleability with respect to extraction (which they introduce). Wee [6] gave a $O(\log^*(n))$ round construction following the same notion of security. A limited black-box constant-round construction was given by Goyal [7] for an even weaker notion called non-malleability with respect to replacement. The construction of Goyal was restricted to providing security only against synchronizing adversaries[2] (as opposed to general adversary). This makes it useful in stand-alone settings only. However in settings where there are more than one (uncoordinated) executions, the construction of Goyal does not provide any security.

Both these weaker notions of security have been useful in constructing secure protocols for (stand-alone) multi-party computation in a black-box manner. In both of these notions, the adversary can indeed correlate (in a limited way) the value it commits to in the right execution to the one in the left execution: in particular, if the value on left is 0, adversary may be able to commit to 0, while if the value on left is 1, adversary commits to $\perp$. Such a situation raises the possibility of selective abort attacks. Even in settings where these notions have been useful, the analysis is more complex than if one were using the standard notion of non-malleability with respect to commitment. Using the standard

[2]Roughly, this means that the man-in-the-middle $\mathcal{M}$ sends the i-th round message on the right immediately after getting the i-th round message in the left interaction.

security notion allows us to construct commitment scheme which can be useful in a wider range of settings as well as obtain simpler and cleaner proofs of security.

II. Definitions and Tools

For lack of space, we will not include the detailed background of each of all the cryptographic primitives that are used in our constructions and of some facts from information theory.

Basic notation: Throughout this paper, we let $\mathbb{N}$ denote the set of all natural numbers and $[m]$ be the set $\{1, 2, \ldots, m\}$ for any $m \in \mathbb{N}$. Unless stated otherwise, we denote by $k \in \mathbb{N}$ the security parameter and all quantities that are polynomial in k will be denoted by $\texttt{poly}(k)$. For any $x \in \{0,1\}^*$, we denote the length of x (in bits) by $|x|$. We denote by (A, B) a pair of interactive Turing machines A and B, and denote by $\langle A, B \rangle$ the random variable that represents the interaction between two interactive Turing machines A and B. More precisely, we denote by $\tau = \langle A(x), B(y) \rangle$ the interactive execution of (A, B) invoked with inputs x for A, y for B, and producing τ as the transcript of the execution. We now give the formal definition of a commitment scheme.

A. Commitment Schemes

We will use Naor's statistically binding commitment scheme [37] in our construction. Naor's scheme only requires a black-box use of a pseudo-random generator (which can be based on the black-box use of any one-way function [38]), and we will use Naor's commitment scheme in a black-box manner. We denote by $\texttt{CS} = (\mathrm{Com}, \mathrm{Rec})$, Naor's commitment scheme executed by a sender Com and a receiver Rec with the following notation: $c = \mathrm{Com}_\sigma(b; \omega)$ denotes a commitment to a bit b computed using randomness ω, where σ is the first message generated by Rec to construct the commitment. To decommit and verify the commitment, Com sends (b, ω) and Rec verifies that $c = \mathrm{Com}_\sigma(b; \omega)$. We stress that Naor's commitment scheme can be used to commit to strings by iterating it for each bit of the string. Moreover, we will use it with non-interactive opening.

Extractable commitment schemes: Informally, a commitment scheme is said to be extractable if there exists an efficient extractor that having black-box access to any efficient malicious sender ExCom^* that successfully performs the commitment phase, is able to efficiently extract the committed string. One can construct an extractable commitment scheme $\texttt{ExCS} = (\mathrm{ExCom}, \mathrm{ExRec})$ with non-interactive opening from any commitment scheme $\texttt{CS} = (\mathrm{Com}, \mathrm{Rec})$ with non-interactive opening in a black-box as described in [30]. The extractor described in [30] produces over extraction, which means that the extractor can output a value different from $\perp$ when the transcript has no valid opening. In our constructions, we will also need an extractable commit scheme without over extraction, but tolerating extraction failure.

Definition 1: A weakly extractable commitment scheme $\texttt{WExCS} = (\mathrm{WExCom}, \mathrm{WExRec})$ is a commitment scheme such that given oracle access to any PPT malicious sender WExCom^*, committing to a string, there exists an expected PPT extractor $\texttt{Ext}$ that outputs a pair (τ, σ^*) such that the following properties hold:

- **Simulatability:** the simulated view τ is identically distributed to the view of WExCom^* (when interacting with an honest WExRec) in the commitment phase.
- **Extractability:** the probability that τ is accepting and σ^* correspond to $\perp$ is at most $1/2$. Moreover if $\sigma^* \neq \perp$ then the probability that ExCom^* opens τ to a value different than σ^* is negligible.

A construction that satisfies our definition on top of any commitment scheme $\texttt{CS} = (\mathrm{Com}, \mathrm{Rec})$ is as follows:

Commitment phase:

1) WExCom on input a message σ, generates a random strings r^0 of the same length as σ, and computes $r^1 = \sigma \oplus r^0$. That is, WExCom sets $\sigma = r^0 \oplus r^1$. Then WExCom uses $\texttt{CS}$ to commit to a pair of values (r^0, r^1). That is, WExCom and WExRec produce $c^0 = \langle \mathrm{Com}(r^0, \omega^0), \mathrm{Rec} \rangle, c^1 = \langle \mathrm{Com}(r^1, \omega^1), \mathrm{Rec} \rangle$.
2) WExRec responses to WExCom by sending a random bit challenge string b.
3) WExCom decommits c^b (i.e., non-interactively opens one of previous commitments).
4) WExRec verifies that c^b has been opened correctly.

Decommitment phase:

1) WExCom sends σ and non-interactively decommits the other commitment $c^{\bar{b}}$, where $\bar{b} = 1 - b$.
2) WExRec checks that $\sigma = r^0 \oplus r^1$. If so, WExRec takes the value committed to be σ and $\perp$ otherwise.

The proof of binding and hiding of $\texttt{WExCS}$ are even simpler than the one given in [30]. Moreover, there is no issue of over-extraction, rather there is an issue of under-extraction. Since the malicious sender might refuse to open another commitment during rewinds, with probability $1/2$, a cheating sender commits successfully but the extractor fails. We will show later that this weak notion of extractability suffices to prove the security of our main theorem.

Non-malleable commitment schemes: For the non-malleability of commitments, we follow the definition introduced by Pass and Rosen and by Lin et al. [25], [31]. Let $\mathcal{M}$ be the man-in-the-middle adversary running on auxiliary input z, and $\texttt{NMCS} = (\mathcal{C}, \mathcal{R})$ denote a non-malleable commitment scheme executed by a sender $\mathcal{C}$ and a receiver $\mathcal{R}$. We use the notion of non-malleability with respect to commitment from [1] for statistically binding non-malleable commitment schemes. In this setting, the adversary $\mathcal{M}$ is said to succeed in the experiment if $\mathcal{M}$ can commit to a message $\tilde{\sigma}$ that is related to the message σ committed by the honest committer. Formally, let $mim^{\mathcal{M}}_{\text{NMCS}}(\sigma, z, tag)$ denote a random variable that describes the value σ that $\mathcal{M}$ commits to in the right execution and the view of $\mathcal{M}$ in the full experiment. In the simulated experiment, a simulator $\mathcal{S}$ directly interacts with $\mathcal{R}$. Let $sim^{\mathcal{S}}_{\text{NMCS}}(z, tag)$ denote the random variable describing the value σ' committed to by $\mathcal{S}$ and the output of $\mathcal{S}$. Notice that both in $mim^{\mathcal{M}}_{\text{NMCS}}(\sigma, z, tag)$ and in $sim^{\mathcal{S}}_{\text{NMCS}}(z, tag)$ the values σ and σ' are well defined since the commitment scheme is statistically binding.

We will consider tag-based commitments, where an additional string referred to as tag is received in input by both sender and receiver. The goal of $\mathcal{M}$ receiving a commitment of σ in an execution with tag tag, consists in committing to a related σ' in an execution with a tag $\tilde{tag}$ such that $tag \neq \tilde{tag}$. Therefore in $mim^{\mathcal{M}}_{\text{NMCS}}(\sigma, z, tag)$ we will assume that when the tag used in the right-hand execution is equal to the one used in left-hand execution, then the message committed in $mim^{\mathcal{M}}_{\text{NMCS}}(\sigma, z, tag)$ is always defined as $\perp$. It is well known that tag-based non-malleable commitments imply plain non-malleable commitments since one can use any signature scheme for this implication. Since it is known how to construct signature schemes by using a one-way function in a black-box manner, we have that the sole notion to care about in this work is that of tag-based non-malleable commitments.

Definition 2: A tag-based commitment scheme NMCS is said to be non-malleable if for every PPT man-in-the-middle adversary $\mathcal{M}$, there exists a (expected) PPT simulator $\mathcal{S}$ such that the following ensembles are computationally indistinguishable: $\{mim^{\mathcal{M}}_{\text{NMCS}}(\sigma, z, tag)\}_{tag\in\{0,1\}^k, \sigma\in\{0,1\}^k, k\in\mathbb{N}, z\in\{0,1\}^*}$, $\{sim^{\mathcal{S}}_{\text{NMCS}}(z, tag)\}_{tag\in\{0,1\}^k, k\in\mathbb{N}, z\in\{0,1\}^*}$.

Similarly, one can define the one-many (resp., many-many) variant of the above definition where the view of $\mathcal{M}$ along with the tuple of values it commits to is required to be indistinguishable regardless of the value (resp., values) committed to in the left interaction (resp., interactions) by the honest sender. We refer the reader to [31] for more details. We also define the notion of one-sided non-malleable commitment scheme where we only consider interactions where the players of the left execution use a common value tag that is smaller than any value $\tilde{tag}$ used in any right interaction[3].

In the statistically hiding case, the previous definition of non-malleability (with respect to commitment) does not make sense, because the committed value is not necessary well defined. To analyze the non-malleability in such a setting, the standard notion of non-malleability is with respect to opening and was studied in [39], [24]. Briefly, in the notion of non-malleability with respect to opening, the adversary is considered successful if after the commitment phase (where $\mathcal{M}$ commits to a message σ), and after observing the decommitment to σ from a honest committer, $\mathcal{M}$ can decommit a message $\tilde{\sigma}$ that is related to σ.

[3]If there exists a right interaction with $\tilde{tag} < tag$, the value b committed to in that right interaction is defined to be $\perp$.

B. Statistically Secure Multi-Party Computation (MPC)

Informally, a secure multi-party computation (MPC) [34] scheme allows n players to jointly and correctly compute an n-ary function based on their private inputs, even in the presence of t corrupted players. More precisely, let n be the number of players and t denotes the number of corrupted players. Under the assumption that there exists a synchronous network over secure point-to-point channels, in [34] it is shown that for every n-ary function $f : (\{0,1\}^*)^n \to (\{0,1\}^*)^n$, there exists a t-secure MPC protocol π_f that securely computes f in the semi-honest model for any $t < n/2$, and in the malicious model for any $t < n/3$, with perfect completeness and security. That is, given the private input w_i of player i, after running the protocol π_f, each honest player i receives in output the i-th component of the result of the function f applied to the inputs of the players, as long as the adversary corrupts less than t players. In addition, nothing is learnt by the adversary from the execution of π_f other than the output. We will later use MPC protocols with perfect completeness and statistical security. We will refer to such a protocol π_f mentioned in the above theorem as an (n, t)-perfectly secure MPC protocol for f. Notice that all the above communication requirements to run the MPC protocol will not result in communication requirements for our commitment scheme, since we will use virtual executions of MPC that will be run only locally by players.

C. Consistency of Views

In an MPC protocol, the view of a player includes all messages received by that player during the execution of the protocol, the private inputs given to the player and the randomness used by the player. We further denote by $view_i$ the view of player P_i. For a honest player P_i, the final output and all messages sent by that player can be inferred from $view_i$ by running a virtual execution of the protocol. Next, we recall the following definition of view consistency adapted from [40].

Definition 3: A $view_i$ of an honest player during an MPC computation π contains input and randomness used in the computation, and all messages received/sent from/to the communication tapes. We have that a pair of views $(view_i, view_j)$ are consistent with each other if, (a) both the players P_i and P_j individually computed each outgoing message honestly by using the random tapes, inputs and incoming messages specified in $view_i$ and $view_j$ respectively, and, (b) all output messages of P_i to P_j appearing in $view_i$ are consistent with incoming messages of P_j received from P_i appearing in $view_j$, and vice versa.

D. Verifiable Secret Sharing (VSS)

Informally, a verifiable secret sharing (VSS) [33] scheme is a two-stage secret sharing protocol for implementing the following functionality. In the first stage, a special player referred to as dealer shares a secret among the other players referred to as shareholders in the presence of at most t corrupted players. In the second stage, players reconstruct the secret shared by the dealer. The functionality ensures that when the dealer is honest, before the second stage begins, all corrupted players have no information about the secret. Moreover, when the dealer is dishonest, at the end of the share phase the honest players would have realized it through an accusation mechanism that disqualifies the dealer. In contrast to Shamir's Secret Sharing scheme [41], a VSS scheme can tolerate errors on malicious dealer and players on distributing inconsistent or incorrect shares, indeed the critical property is that even in case the dealer is dishonest but has not been disqualified, still the second stage always reconstruct the same bit among the honest players.

Direct implementations of $(n+1, \lfloor n/3 \rfloor)$-perfectly secure VSS schemes can be found in [34], [42]. However since we are interested in a deterministic reconstruction procedure, we will use the scheme of [43]that implements an $(n+1, \lfloor n/4 \rfloor)$-perfectly secure VSS scheme.We will denote by $\Pi_{VSSshare}$ the execution of an $(n+1, \lfloor n/4 \rfloor)$-perfectly secure protocol that implements the Share stage of the above VSS functionality. We will denote by Π_{recon} the corresponding protocol executed by shareholders to implement the deterministic Recon stage.

III. Construction of Non-Malleable Commitments

We now describe a simplified version of our protocol, which considers "short" tags with one-sided no n-malleability. Moreover security is guaranteed against synchronized adversaries only. A synchronized adversary is a restricted adversary that plays the main-in-the-middle attack by playing exactly one message on the right execution after a message is received from the left execution, and playing exactly one message on the left execution after a message is received from the right execution. In our scheme we will use an extractable commitment scheme ExCS as already used in previous work [30]. Such a commitment scheme suffers of "over ex traction", which means that the extractor can output a value different than $\perp$ even when the committed message is not well formed (therefore the committed message is undefined and can not be opened anymore). In addition, we use the commitment scheme WExCS which instead suffers from "under extraction" as described in Section II. We assume that each execution has a session identifier $tag \in [2n]$, where n is the length of party identity in bits. Let k be the security parameter and $\ell = \ell(k) = k \cdot tag$. The commitment scheme $\texttt{NMCS} = (\mathcal{C}, \mathcal{R})$ between a committer $\mathcal{C}$ and a receiver $\mathcal{R}$ proceeds as follows to commit to a k-bit string σ. We assume that $\lambda = \lfloor k/4 \rfloor$. In the description below, we have included some intuition in *italics*.

Commitment phase:

0. **Initial setup.** $\mathcal{R}$ picks λ out of k players (which will be later emulated by the committer) at random. That is, it randomly selects λ distinct indices $\Lambda = \{r_1, \ldots, r_\lambda\}$ where $r_i \in [k]$ for any $i \in [\lambda]$. For each r_i, $\mathcal{R}$ sends an extractable commitment c_i of r_i using ExCS.

1) **Primary slot.** Let $\Pi_{VSSshare}$ be a protocol implementing the Share phase of a $(k+1, \lambda)$-perfectly secure VSS scheme. We require the VSS protocol to have a deterministic reconstruction phase. The committer $\mathcal{C}$ is given a k-bit string σ to commit.

   1.1. **Commit:** $\mathcal{C}$ first generates ℓ pairs of random strings $\{\alpha_i^0, \alpha_i^1\}_{i \in [\ell]}$ of length $4k$ each, and a k-bit random string s. *Here the strings are such that the knowledge of both strings $\{\alpha_i^0, \alpha_i^1\}$ for any pair will allow an extractor to extract the committed value. The string s is meant to serve as a seed of a strong extractor used later on in the protocol. The purpose of the next two stages (1.2 and 1.3) is simply to produce a specialized commitment to the strings $\{\alpha_i^0, \alpha_i^1\}_{i \in [\ell]}, s$ and σ.*

   1.2. The committer $\mathcal{C}$ now starts emulating $k+1$ (virtual) players locally "in his head". $\mathcal{C}$ sets the input of P_{k+1} (i.e., the Dealer) to the concatenation of σ, s, and $\{\alpha_i^0, \alpha_i^1\}_{i \in [\ell]}$, while each other player has no input. Then $\mathcal{C}$ runs $\Pi_{VSSshare}$ and each player P_i obtains shares w_i, for any $i \in [k]$.

   1.3. Let $view_1^1, \ldots, view_{k+1}^1$ be the views of the $k+1$ players describing the execution of $\Pi_{VSSshare}$. $\mathcal{C}$ uses WExCS to send a commitment V_i^1 of $view_i^1$ to $\mathcal{R}$, in parallel for any $i \in [k]$. *At this stage, the committer is now committed to σ, s and $\{\alpha_i^0, \alpha_i^1\}_{i \in [\ell]}$.*

   1.4. **Challenge:** $\mathcal{R}$ sends a random ℓ-bit challenge string $ch = (ch_1, \ldots, ch_\ell)$.

   1.5. **Response:** $\mathcal{C}$ sends $\{\alpha_i^{ch_i}\}_{i \in [\ell]}$ to $\mathcal{R}$. *The goal of the extractor would be to rewind and learn a pair $\{\alpha_i^0, \alpha_i^1\}$. To ensure non-malleability, this would be done without rewinding the (interleaved) left interaction.*

2) **Verification message.** Let $\mathcal{H}$ be a family of pairwise-independent hash functions with domain $\{0,1\}^{4k}$ and range $\{0,1\}^k$, and $\texttt{Ext} : \{0,1\}^{4k} \times \{0,1\}^k \to \{0,1\}^k$ be a strong randomness $(3k, 2^{-k})$-extractor.

   2.1. $\mathcal{R}$ picks a function h at random from $\mathcal{H}$ and sends it to $\mathcal{C}$.

   2.2. $\mathcal{C}$ sends $s, \{h(\alpha_i^0), h(\alpha_i^1), B_i = \sigma \oplus \texttt{Ext}(\alpha_i^0, s) \oplus \texttt{Ext}(\alpha_i^1, s)\}_{i \in [\ell]}$ to $\mathcal{R}$.

   Say that in the primary slot phase, the extractor rewinds the adversary and receives a value α_i^j. This *phase enables checking such a received value for correctness (and for subsequent recovery of the string σ). This phase is purely information theoretic but still provides for the right binding properties. The corresponding mechanism in the construction of Goyal was implemented using complex computations involving random tapes used to generate various commitments.*

3) **Consistency proof.** *Now the sender needs to prove the correctness of the values revealed in stage 1.5 and 2.2.*
   3.1. Let Π_{ch} be a (k, λ)-perfectly secure MPC protocol such that given ch as a public input and w_i as the private input of P_i for any $i \in [k]$, at the end of the computation $\{\alpha_i^{ch_i}\}_{i\in[\ell]}$ is received in output by P_i for any $i \in [k]$. $\mathcal{C}$ runs internally Π_{ch} and sends a commitment V_i^2 of the view $view_i^2$ of P_i when executing Π_{ch} using `WExCS` in parallel for any $i \in [k]$ to $\mathcal{R}$.
   3.2. Let Π_h be a (k, λ)-perfectly secure MPC protocol such that given a hash function h as a public input and w_i as the private input of P_i for any $i \in [k]$, at the end of the computation $(s, \{h(\alpha_i^0), h(\alpha_i^1)\}_{i\in[\ell]}, \{B_i = \sigma \oplus \mathtt{Ext}(\alpha_i^0, s) \oplus \mathtt{Ext}(\alpha_i^1, s)\}_{i\in[\ell]})$ is received in output by P_i for any $i \in [k]$. $\mathcal{C}$ runs internally Π_h and sends a commitment V_i^3 of the view $view_i^3$ of P_i when executing Π_h using `WExCS` in parallel for any $i \in [k]$.
   3.3. $\mathcal{R}$ decommits $\{c_i\}_{i\in[\lambda]}$.
   3.4. $\mathcal{C}$ decommits $\{V_{r_i}^1, V_{r_i}^2, V_{r_i}^3\}_{i\in[\lambda]}$ (i.e., it decommits the subset of views $\{view_{r_i}^1, view_{r_i}^2, view_{r_i}^3\}_{i\in[\lambda]}$.)
   3.5. for $j = 1, 2, 3$, $\mathcal{R}$ verifies that all pairs of views in $\{view_{r_i}^j\}_{i\in[\lambda]}$ are consistent (according to Definition 3) and that the dealer P_{k+1} has not been disqualified by any player, otherwise $\mathcal{R}$ aborts; moreover for $j = 1, 2$ and $i = 1, \ldots, \lambda$, $\mathcal{R}$ checks that $view_{r_i}^j$ is a prefix of $view_{r_i}^{j+1}$, otherwise $\mathcal{R}$ aborts.

Decommitment phase:

1) $\mathcal{C}$ decommits $\{V_i^1\}_{i\in[k]}$ as $\{view_i^1\}_{i\in[k]}$.
2) $\mathcal{R}$ checks that all commitments to the views are opened correctly in the previous step. If a commitment is opened incorrectly, $\mathcal{R}$ sets the corresponding revealed view to 0^k (instead of just aborting).
3) Let $\Pi_{VSSrecon}$ be a protocol implementing the `Recon` phase corresponding to the $(k+1, \lambda)$-perfectly secure VSS `Share` phase (which includes the string σ) used in the commitment phase. $\mathcal{R}$ runs $\Pi_{VSSrecon}$ using $view_1^1, \ldots, view_{k+1}^1$ as input to reconstruct and output the first substring of the value that the majority of the players would output in the reconstruction. If there is no majority, then consider the committer to have aborted during the commitment phase (and output $\perp$).

   We stress that the receiver does not perform any additional checks. In particular, even if it detects that some of the views are not correctly constructed (and hence the committer behaved in a dishonest way), it still accepts the decommitment phase as long as a majority of the players agree on a value during reconstruction. This is crucial for getting security as per the non-malleability with respect to commitment notion.

Theorem 1: The scheme `NMCS` is a one-sided non-malleable commitment scheme with short tags secure against synchronized adversaries.

Hiding: To prove the hiding property, we claim that any adversary $\mathcal{A}$ that breaks the hiding property of `NMCS` can be used to break the hiding property of `WExCS`. The proof goes through hybrid arguments starting with $\mathcal{S}$ that runs $\mathcal{C}$ on input m_0. In the next hybrid $\mathcal{S}$ retrieves all the indices r_i selected by $\mathcal{A}$. In the next hybrid for all commitments of the views that will not be opened to $\mathcal{A}$, $\mathcal{S}$ will gradually replace the committed values (corresponding to views that are consistent with the computations) by commitments to random strings. In the next hybrid $\mathcal{S}$ uses simulators $\mathcal{S}_{ch}$ and $\mathcal{S}_h$ of underlying MPC protocols Π_{ch} and Π_h and for every player $i \in \Lambda$, our $\mathcal{S}$ will commit the views being generated by $\mathcal{S}_{ch}$ and $\mathcal{S}_h$. In the next experiment the simulator assumes that all honest players play with shares of m_1. In the next experiment for all commitments that will not be opened to $\mathcal{A}$, $\mathcal{S}$ will gradually changes back the committed values from the random strings to views that are consistent with committed value m_1. Finally, the simulator honestly executes the protocol by committing to the value m_1 and outputs the corresponding views.

Binding: The statistical binding property follows in a straightforward manner from the statistical binding of `WExCS`. Indeed the only step performed by the unbounded adversarial sender in the decommitment phase, consists in decommitting all the statistically binding commitments $\{V_i^1\}_{i\in[k+1]}$ sent in the commitment phase through the extractable commitment scheme. The decommitted string is then derived from running the reconstruction phase of the VSS using these views. Then since the reconstruction is deterministic, regardless of how these views are constructed, there is a unique string which will be reconstructed. Therefore to violate the binding of our scheme one has to violate the statistical binding of the extractable commitment scheme.

Non-malleability: As one could expect, we borrow several ideas from the analysis of Goyal [7][4] with some crucial modifications. First of all, we simplified part of his construction by using pairwise-independent hash functions and strong randomness extractors. Therefore these tools will be used in the proof when showing an extractor that

[4]Parts of the text in this section is borrowed verbatim from [7].

breaks the hiding as a consequence of a successful $\mathcal{M}$. Secondly, the protocol of [7] uses a zero-knowledge proof of consistency for (at least) two different crucial purposes: 1) in the proof of non-malleability, the soundness of the zero-knowledge proof guarantees that when the commitment phase is completed successfully, there exists a unique committed string not equal to $\perp$; 2) in the proof of hiding the zero-knowledge property of the zero-knowledge proof guarantees that no information on the committed string is leaked. Since we replaced the zero-knowledge proof with our extensions of the computation in the head paradigm, we will use some of the arguments of [40] to prove that the same holds also in our scheme.

A. Getting Non-Malleable Commitments for Small Tags

We construct a non-malleable commitment scheme for small tags (i.e., $tag \in [2n]$) against a synchronizing adversary. This can be done very similar to the construction by Goyal (which is based on ideas from Pass and Rosen [24]). Denote by $\ell[a]$ the value $k \cdot tag$ and by $\ell[b]$ the value $k\cdot(2n - tag)$. The idea is to have two slots (each representing a rewinding opportunity) such that for exactly one of these slots, the "tag being used on the right" is larger than the one on the left. The extractor will now rewind this slot and extract the value ν. To prove many-many security of the above scheme, we first prove one-many security by simply applying the extractor one by one on all sessions on the right and then resort to a general result of Lin et al [31] (as in Goyal's construction).

B. Security Against Non-Synchronizing Adversaries

Given a one-many non-malleable commitment scheme $sNM = (sCom, sRec)$ for tags of length $\log(n) + 1$, we presents a general and black-box transformation to obtain a one-many non-malleable commitment scheme for tags of length n. This transformation can be sees as a generalization of a previous transformation of [6].

The idea of our construction is similar in spirit to the technique used in [1] to obtain logarithmic round complexity. Each execution of the commitment scheme is associated to an n-bit tag Tag. The committer $\mathcal{C}$ runs a $(k+1, \lfloor k/4 \rfloor)$-perfectly secure VSS scheme with deterministic reconstruction. That is, $\mathcal{C}$ shares the committed value σ and the randomness to the i-th player P_i for $i \in [k]$. Then $\mathcal{C}$ commits to the views of n VSS shareholders using n times the scheme $sCom$ and n different short tags that are derived by Tag.

The key idea is that by applying again the cut and choose techniques on the VSS computation, the adversary is essentially forced in committing to correct views almost in all commitments computed with $sCom$. Then, by noticing that in each commitment of the adversary there are always views of the VSS players that are committed with a tag that has not been used in the commitment received by the adversary, it holds that such views are independent (this comes from non-malleability). Therefore we will be able in the hybrid experiments to change the message committed in the left session while the adversary will still commit to the same message on the right sessions.

Another crucial idea is the fact that the above commitments of the views of the VSS computation must be repeated 3 times. The reason we need this extra technique is that during the hybrid games, we will need to compute commitments of inconsistent views that however will not be detected by the adversary since we will extract first the indexes of the views that the adversary wants to see later. This extraction will require to rewind the adversary, and thus the need of 3 repetitions of the sub-commitment protocol comes from the need of having the guarantee that at least one of such sub-commitments is not disturbed by the above rewind.

IV. Applications

Here we discuss two applications of our constant-round concurrent non-malleable commitment scheme based only on the black-box use of one-way functions.

A. Constant-Round Multi-Party Coin-Tossing

Informally, a coin-tossing protocol allows parties to generate a common unbiased random string. We show a constant-round protocol based on the black-box use of any one-way function. In our protocol the adversary can control up to $n-1$ players and computations are performed in parallel.

Each party P_i selects a random k-bit string σ_i and runs in parallel a sub-protocol $\langle S(\sigma_i), R\rangle$ with each other party, in order to transfer σ_i to the other players. The final outcome of the protocol is $\sigma = \sigma_1 \oplus \cdots \oplus \sigma_n$. The main technical challenge is the design of the above sub-protocol $\langle S(\sigma_i), R\rangle$ by only using one-way functions in a black-box way, still allowing a simulator to bias at its wish the outcome of the coin tossing. In the sub-protocol that we design, each party plays as sender to send its contribution to the coin-tossing protocol, by running a perfectly secure $(k+1, \lambda)$-party VSS protocol where $\lambda = \lfloor k/4 \rfloor$. More precisely, player S will use `NMExCS` to commit to the views of the above VSS players. The above VSS computations and commitments are repeated twice. Then, a (k, λ)-statistically secure MPC protocol is invoked to ensure that the same string σ_i has been shared in the two above VSS executions. Commitments of such resulting views are sent to R and the cut and choose is performed to ensure R that the same string has been shared by S in both above executions. Finally, the views used in the *second* execution of VSS are revealed and the receiver can apply the reconstruction to obtain the string of the sender.

The reason why we will be able to show a simulator that can bias the outcome of the scheme, is that the simulator when playing as receiver can extract from the commitments of the views of the *first* VSS sharing phase, the string that the sender will open later. The simulator can then adjust in the second commitment its string so that the output of the protocol will be the desired string. Obviously the simulator will have to cheat in the MPC protocol since it committed to different strings. This will be again possible by extracting in the first phase of the protocol the set of indexes that the adversary wants to see opened later.

B. Non-Malleable Statistically Hiding Commitments

We also apply our techniques and our concurrent non-malleable commitment scheme to construct the first non-malleable (with respect to opening) statistically hiding commitment scheme `NM-SHCS` from any extractable statistically hiding commitment scheme in a fully black-box way. The idea of the construction is that the message σ to be committed is first shared with a VSS computation and the views of the players are committed through an extractable statistically hiding commitment scheme. In the opening σ is simply revealed and then the VSS reconstruction is performed and the resulting views are committed through our concurrent non-malleable commitment scheme. Then the cut and choose technique certifies that σ was indeed the value shared in the views committed in the commitment phase.

Acknowledgment

The work is supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xer ox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work of the fourth author has been done while visiting UCLA and is supported in part by the Europe an Commission through the FP7 programme under contract 216676 ECRYPT II.

References

[1] D. Dolev, C. Dwork, and M. Naor, "Non-Malleable Cryptography (Extended Abstract)," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, ser. STOC '91, 1991, pp. 542–552.

[2] B. Barak, "Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '02, 2002, pp. 345–355.

[3] J. Katz, R. Ostrovsky, and A. Smith, "Round Efficiency of Multi-party Computation with a Dishonest Majority," in *Advances in Cryptology — EUROCRYPT '03*, ser. Lecture Notes in Computer Science, vol. 2656. Springer, 2003, pp. 578–595.

[4] R. Pass, "Bounded-Concurrent Secure Multi-Party Computation with a Dishonest Majority," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, ser. STOC '04, 2004, pp. 232–241.

[5] H. Lin and R. Pass, "Non-malleability Amplification," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, ser. STOC '09, 2009, pp. 189–198.

[6] H. Wee, "Black-Box, Round-Efficient Secure Computation via Non-malleability Amplification," in *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '10, 2010, pp. 531–540.

[7] V. Goyal, "Constant Round Non-malleable Protocols Using One-way Functions," in *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, ser. STOC '11. ACM, 2011, pp. 695–704.

[8] R. Pass, "Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition," in *Advances in Cryptology — EUROCRYPT '03*, 2003, pp. 160–176.

[9] M. Prabhakaran and A. Sahai, "New Notions of Security: Achieving Universal Composability without Trusted Setup," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004, pp. 242–251.

[10] B. Barak and A. Sahai, "How To Play Almost Any Mental Game Over The Net - Concurrent Composition via Super-Polynomial Simulation," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, 2005, pp. 543–552.

[11] S. Micali, R. Pass, and A. Rosen, "Input-Indistinguishable Computation," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '06, 2006, pp. 367–378.

[12] C. Ventre and I. Visconti, "Completely non-malleable encryption revisited," in *Proceedings ofPublic Key Cryptography*, ser. Lecture Notes in Computer Science, vol. 4939. Springer, 2008, pp. 65–84.

[13] R. Ostrovsky, G. Persiano, and I. Visconti, "Constant-round concurrent non-malleable zero knowledge in the bare public-key model," in *Proceedings of ICALP*, ser. Lecture Notes in Computer Science, vol. 5126. Springer, 2008, pp. 548–559.

[14] R. Ostrovsky, O. Pandey, and I. Visconti, "Efficiency Preserving Transformations for Concurrent Non-malleable Zero Knowledge," in *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, 2010, pp. 535–552.

[15] R. Canetti, H. Lin, and R. Pass, "Adaptive Hardness and Composable Security in the Plain Model from Standard Assumptions," in *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '10, 2010, pp. 541–550.

[16] Z. Cao, I. Visconti, and Z. Zhang, "On constant-round concurrent non-malleable proof systems," *Inf. Process. Lett.*, vol. 111, no. 18, pp. 883–890, 2011.

[17] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai, "Universally composable two-party and multi-party secure computation," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, ser. STOC '02, 2002, pp. 494–503.

[18] B. Barak, R. Canetti, J. B. Nielsen, and R. Pass, "Universally Composable Protocols with Relaxed Set-Up Assumptions," in *Proceedings of the 45rd Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '04, 2004, pp. 186–195.

[19] J. Katz, "Universally Composable Multi-party Computation Using Tamper-Proof Hardware," in *Advances in Cryptology — EUROCRYPT '07*, 2007, pp. 115–128.

[20] H. Lin, R. Pass, and M. Venkitasubramaniam, "A Unified Framework for Concurrent Security: Universal Composability from Stand-alone Non-malleability," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, ser. STOC '09, 2009, pp. 179–188.

[21] B. Barak, "How to Go Beyond the Black-Box Simulation Barrier," in *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '01, 2001, pp. 106–115.

[22] O. Pandey, R. Pass, and V. Vaikuntanathan, "Adaptive One-Way Functions and Applications," in *Advances in Cryptology — CRYPTO '08*, 2008, pp. 57–74.

[23] R. Pass and H. Wee, "Constant-Round Non-malleable Commitments from Sub-exponential One-Way Functions," in *Advances in Cryptology — EUROCRYPT '10*, 2010, pp. 638–655.

[24] R. Pass and A. Rosen, "New and improved constructions of non-malleable cryptographic protocols," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, ser. STOC '05, 2005, pp. 533–542.

[25] R. Pass and A. Rosen, "Concurrent non-malleable commitments," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '05, 2005, pp. 563–572.

[26] R. Ostrovsky, G. Persiano, and I. Visconti, "Simulation-based concurrent non-malleable commitments and decommitments," in *Proceedings of TCC*, ser. Lecture Notes in Computer Science, vol. 5444. Springer, 2009, pp. 91–108.

[27] Z. Cao, I. Visconti, and Z. Zhang, "Constant-round concurrent non-malleable statistically binding commitments and decommitments," in *Proceedings of PKC*, ser. Lecture Notes in Computer Science, vol. 6056. Springer, 2010, pp. 193–208.

[28] H. Lin and R. Pass, "Constant-round Non-malleable Commitments from Any One-way Function," in *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, ser. STOC '11, 2011, pp. 705–714.

[29] Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank, "Black-box constructions for secure computation," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, ser. STOC '06, 2006, pp. 99–108.

[30] R. Pass and H. Wee, "Black-Box Constructions of Two-Party Protocols from One-Way Functions," in *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, 2009, pp. 403–418.

[31] H. Lin, R. Pass, and M. Venkitasubramaniam, "Concurrent Non-malleable Commitments from Any One-Way Function," in *Theory of Cryptography, 5th Theory of Cryptography Conference, TCC 2008*, 2008, pp. 571–588.

[32] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Zero-Knowledge Proofs from Secure Multiparty Computation," *SIAM J. Comput.*, vol. 39, no. 3, pp. 1121–1152, 2009.

[33] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract)," in *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '85, 1985, pp. 383–395.

[34] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract)," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, ser. STOC '88, 1988, pp. 1–10.

[35] V. Goyal, Y. Ishai, M. Mahmoody, and A. Sahai, "Interactive locking, zero-knowledge pcps, and unconditional cryptography (full version)," in *CRYPTO*, ser. Lecture Notes in Computer Science, T. Rabin, Ed., vol. 6223. Springer, 2010, pp. 173–190.

[36] Y. Ishai and E. Kushilevitz, "On the Hardness of Information-Theoretic Multiparty Computation," in *Advances in Cryptology — EUROCRYPT '04*, 2004, pp. 439–455.

[37] M. Naor, "Bit Commitment Using Pseudorandomness," *J. Cryptology*, vol. 4, no. 2, pp. 151–158, 1991.

[38] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A Pseudorandom Generator from any One-way Function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[39] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky, "Non-interactive and non-malleable commitment," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, ser. STOC '98. ACM, 1998, pp. 141–150.

[40] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Zero-knowledge from Secure Multiparty Computation," in *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, ser. STOC '07, 2007, pp. 21–30.

[41] A. Shamir, "How to Share a Secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[42] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin, "Efficient Multiparty Computations Secure Against an Adaptive Adversary," in *Advances in Cryptology — EUROCRYPT '99*, ser. Lecture Notes in Computer Science, vol. 1592. Springer, 1999, pp. 311–326.

[43] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin, "The Round Complexity of Verifiable Secret Sharing and Secure Multicast," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, ser. STOC '01. ACM, 2001, pp. 580–589.