New Notions of Security: Achieving Universal Composability without Trusted Setup

Manoj Prabhakaran*
Princeton University

Amit Sahai†
Princeton University

ABSTRACT

We propose a modification to the framework of Universally Composable (UC) security [3]. Our new notion involves comparing the real protocol execution with an ideal execution involving ideal functionalities (just as in UC-security), but allowing the environment and adversary access to some super-polynomial computational power. We argue the meaningfulness of the new notion, which in particular subsumes many of the traditional notions of security.

We generalize the Universal Composition theorem of [3] to the new setting. Then under new computational assumptions, we realize secure multi-party computation (for static adversaries) without a common reference string or any other set-up assumptions, in the new framework. This is known to be impossible under the UC framework.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Security,Theory

Keywords

Secure Multi-Party Computation, Universal Composability, General Composition, Environmental Security, Generalized Environmental Security, secure protocols, simulation

1. INTRODUCTION

Over the last two decades, there has been tremendous success in placing cryptography on a sound theoretical foundation, and building an amazingly successful theory out of it. The key elements in this Modern Cryptographic Theory are the definitions capturing the intuitive, yet elusive notions of security in the various cryptographic settings. The definitions of the early 80's proved to be extremely successful in this regard. But with time, as the theory started addressing more and more complex concerns, further notions of security had to be introduced. One of the most important concerns theory ventured into is of complex environments where different parties are communicating with each other concurrently in many different protocols. A series of efforts in extending security definitions culminated in the paradigm of Universally Composable (UC) Security [3], which along with modeling a general complex network of parties and providing definitions of security in that framework, provided powerful tools for building protocols satisfying such definitions.[1]

The Background: Universally Composable Security. The basic underlying notion of security in the UC framework and its predecessors is based on *simulation*. An "ideal" world is described, where all requisite tasks get accomplished securely, as if by magic. The goal of the protocol designer is to find a way to accomplish these tasks in the "real" world, so that no malicious adversary can take advantage of this substitution of ideal magic by real protocols. To formalize this, we say that for every malicious adversary $\mathcal{A}$ that tries to take advantage of the real world, there is an adversary $\mathcal{S}$ that can achieve *essentially the same results* in the ideal world. The "results" are reflected in the behavior of an *environment*. In this paper we shall refer to this notion of security as *"Environmental Security."* If a real-life protocol "Environmentally Securely realizes" a task, it ensures us that replacing the magic by reality does not open up new unforeseen threats to the system. (There may already be threats to the system even in the ideal world. But employing cryptographic primitives cannot offer a solution if the ideal system itself is badly conceived.) The ideal-world adversary $\mathcal{S}$ is called a *simulator* as it simulates the real-world behavior of $\mathcal{A}$, in the ideal world.

The advantage of Environmentally Secure (ES) protocols, as shown in [3], is that they are "Universally Composable," i.e., roughly, if multiple copies of an ES-protocol are present in the system (in fact they could be copies of different protocols), then they collectively ES-realize the collection of the

*Email: mp@cs.princeton.edu.

†Supported by grants from the Alfred P. Sloan Foundation and the NSF ITR program. Email: sahai@cs.princeton.edu.

STOC'04, June 13–15, 2004, Chicago, Illinois, USA.

[1] A similar framework to UC Security was independently proposed by Pfitzmann and Waidner [22, 23]. These two frameworks are conceptually very similar, although there are a number of technical differences. We choose to use the UC framework in this paper.

tasks they individually ES-realize. Hence we shall often refer to the framework in [3] as the ES/UC framework, or simply ES-framework or UC-framework.

Unfortunately, this notion of security turns out to be too strong to be achievable in standard settings. It has been shown that much of the interesting cryptographic tasks (including *e.g.* commitment, zero knowledge and secure multiparty computation) *cannot* be ES-realized when the adversary can control at least half the parties [3, 4, 6]. On the other hand, under a trusted set-up assumption – that there is a public reference string chosen by a completely trusted party – it is known how to build protocols for the most ambitious of cryptographic tasks (general secure multiparty computation with dishonest majority) satisfying the Environmental Security definition.[2] In this work we seek to develop such protocols in the plain model (without any trusted set-up assumptions), by modifying the notion of security, while still retaining the composability.

This Work: New Ideas. This work seeks to modify the ES/UC framework. Our starting point is the observation that in the ideal-world used by the ES/UC framework, even if the adversary has unlimited computational powers, the ideal-world captures a strong notion of security in most cases of interest.

Generalizing Environmental Security: Consider the ideal-world version of a commitment protocol between two parties. There is a trusted third party functionality which has secure channels with the two parties. In the commitment phase, the functionality receives a bit from the sender, and informs the receiver that it received a bit (without telling it which bit, or anything else). Later, in the reveal phase, the sender can request the functionality to reveal the bit, and it will send the bit it originally received to the receiver.

Now we observe that the computational power of an adversary is irrelevant in this ideal world. This is because the security in the ideal-world is "information-theoretic" – additional computational power will not aid the advsersary because it has no relevant information to compute with! Indeed, in most applications, the functionality is so defined as to capture the notion of security with no reference to the power of the adversary. It is *legitimate* for a computationally unbounded adversary to interact with the honest parties in the ideal-world.

Overcoming Impossibility Results: Allowing *the simulator to be more powerful than the real-world adversary* would have helped us overcome the impossibility results from [3, 4, 6]. Similar motivation is behind previous works which explored super-polynomial or quasi-polynomial simulation in the context of simpler compositions (*e.g.* [20]). However, to prove the Universal Composition theorem *we do require that the real-world adversary be as powerful as the ideal-world adversary.* On the other hand, unfortunately, if we provide *both* real and ideal adversaries with same computational power, no matter how large, the impossibility results continue to hold (see later for an explanation).

[2]It is also known how to achieve this when the majority of the parties are honest, even without trusted setup. But we stress that this result needs the assumption that a majority of parties are honest in *every* execution of various protocols. This assumption, to us, seems against the "spirit" of universal composability, which seeks to assure us of security in arbitrary adversarial network scenarios.

We get out of this apparent deadlock by introducing the notion of *"Imaginary Angels."* An Imaginary Angel is essentially a super-polynomial time oracle imagined to be available to the environment and adversary in the real or ideal world. We use the name "Angel" to highlight a key feature of the imaginary oracles that we consider: an Imaginary Angel answers queries selectively, so as not to hurt honest parties. An Imaginary Angel will be designed so that it will answer queries which will allow breaking the security of already corrupted parties (and thus will be of good use to the ideal world adversary in carrying out the simulation), but will be unhelpful in breaking the security of honest parties.

We stress that an Imaginary Angel, considered available to the environment and the adversary, is only for the purpose of defining and analyzing security; the actual parties in protocols do not have access to the Imaginary Angel.

Generalized Environmental Security. We shall refer to the framework introduced here as the *generalized Environmental Security* (gES) framework, and a model in this framework using Imaginary Angel Γ will be called the Γ-ES-model. In the Γ-ES-model, the set of honest protocols is the same as in the original ES/UC model. However, the environment, the real-world adversary and the ideal-world adversary are allowed access to the Imaginary Angel Γ. All that an Imaginary Angel Γ needs to know about the state of the system is the set of corrupted parties. We shall say a protocol π Γ-ES-realizes a functionality $\mathcal{F}$ against a class of adversaries $\mathcal{C}$ if no environment with access to the Imaginary Angel Γ can distinguish between the real and ideal worlds, where the adversaries also have access to Γ. It turns out that, like in the ES/UC model, the Universal Composition theorem still holds in the Γ-ES-model, for a fixed Imaginary Angel Γ.

Meaningfulness of the New Notion. As discussed above, usually an ideal-world employing the ideal functionality captures the security requirements even when the adversary has unbounded powers (and in particular, access to the Imaginary Angel Γ). This is usually the case in most interesting applications: like e-commerce or database transactions, secure communication, and generally various multi-party computation tasks.

However there may be some situations where the extra power for the adversary is not entirely "ideal"– for instance consider playing online poker against human players in the Γ-ES-model, using (in the ideal-world) an ideal functionality which interacts with the players. In the ideal-world the players have access to an Imaginary Angel Γ, and they may, in principle, find that useful in finding a good strategy for the game. However typically an Imaginary Angel is designed to break some specific cryptographic problem (as will be apparent with the Imaginary Angel Ψ we will use in this work) and access to it is presumably not useful in a game of poker. Thus, even in many of these situations, where it is not entirely ideal to allow unlimited power to the adversary in the ideal-world, the security guarantee provided by Γ-ES-model may be considered good for all practical purposes. We further stress that, of course, in the real world, no parties actually gain any super-polynomial powers.

It is instructive to consider what the new notion yields in terms of more familiar notions of security. We note that under the more traditional measures of security, in many cases Γ-ES security implies security somewhat *stronger* than

that implied by ES/UC-security. For instance consider non-malleability of commitments: Any commitment scheme which Γ-ES-realizes the commitment functionality , is in fact non-malleable[3] (with respect to itself), even for PPT adversaries with access to Γ. But on the other hand, the traditional definition of Zero Knowledge proofs depends on simulation; so it may not be true that a protocol which Γ-ES-realizes the zero-knowledge proof functionality is a Zero-Knowledge proof under the traditional definition. Nevertheless the Witness Indistinguishability property of that protocol does get translated to (a stronger) Witness Indistinguishability property under the traditional definition (stronger, because it holds against adversaries with access to Γ). Note that the above observations hold for *any* protocol and *any* Imaginary Angel Γ– not just the specific one we introduce in this work.

A Less Technical Interpretation of Security in the Γ-ES-model. As we derive them, our results are stated in a technical setting – where the environment, and the real-world and ideal-world adversaries are modeled to have access to the Imaginary Angel. To understand this result from a less technical viewpoint, with no reference to the particular Imaginary Angel, a simple corollary of this result is useful.

Security in our setting implies the following: For every *polynomial-time* environment and real-world adversary, there exists a super-polynomial-time ideal-world adversary, such that the environment cannot distinguish the ideal world from the real world. This is essentially the situation that we first proposed in our discussion. Note, however, that Universal Composability does not necessarily hold for the security in this corollary (that is, once we state the security only with reference to a polynomial-time real-world). Hence this corollary is used to interpret the result *at the end*, after all the compositions are applied. Still, if this notion of security is convincing, then our framework suffices for the application at hand.

Avoiding the Impossibility Results. It is interesting to observe how this work manages to evade the impossibility results from [3, 4, 6]. First, let us briefly recall the result showing that under the ES/UC-framework, commitment functionality cannot be securely realized in the plain model (impossibility for other functionalities are similar in spirit). Suppose, for contradiction, there is indeed such a protocol between the sender C and receiver R. The proof proceeds by considering two "real-world" situations A and B. In situation A, the adversary corrupts C and directs it to act transparently between the environment and R. The environment will run an honest commitment protocol (on behalf of C), and so the receiver will accept the commitment (and later a reveal) . Since the protocol is UC-secure, there exists a simulator $\mathcal{S}_A$ which can effect the same commitment and reveal in the "ideal-world." In other words $\mathcal{S}_A$ can *extract* the committed bit from the protocol messages (so that it can send it to the ideal commitment functionality). Now consider situation B, where the receiver R is corrupted. The contradiction is achieved by considering an adversary $\mathcal{A}_B$ which directs R to act honestly, but sends all the messages also to an internal copy of $\mathcal{S}_A$. Now $\mathcal{S}_A$ is essentially in the same position as in situation A and can extract the committed bit, from the honest sender's commitment. However this violates the secrecy property of the commitment protocol, leading to the contradiction.

We again note that just allowing the adversary (real and ideal) access to more computational resources does not by itself stop the above proof from going through. $\mathcal{A}_B$ can still run $\mathcal{S}_A$ internally and violate the protocol's security, as it has the same computational powers as $\mathcal{S}_A$. So we would like to make sure that $\mathcal{A}_B$ cannot run $\mathcal{S}_A$, presumably because $\mathcal{S}_A$ has more computational powers than $\mathcal{A}_B$. But on the other hand, for the UC theorem to hold, the environment (and hence the adversary) should be able to internally run the simulators. In other words, the composition property holds only when the protocol is secure in environments which can be as powerful as the simulators. Thus we need to give the environment all the power of the simulator. So it would seem that we cannot prevent $\mathcal{A}_B$ from being able to run $\mathcal{S}_A$.

We manage to get out of this apparent dead-lock as we allow the power of the environment/simulator to *depend on the set of corrupted parties*. The key factor is that the Imaginary Angel (used only in the proof of security), to which the environment/simulator have access, will base its answers to queries *on the set of corrupted parties*. Note that above, in situations A and B, the set of corrupted parties are different. This lets us make sure that $\mathcal{A}_B$ in situation B cannot run $\mathcal{S}_A$ (which expects to be in situation B), because the Imaginary Angel behaves differently in the two situations. This prevents the impossibility proof from going through. Indeed, as our results show, the new model prevents not just the proof, but also the impossibility.

Our Assumptions. The protocols we construct are proven secure in the Ψ-ES model, where Ψ is a specific Imaginary Angel related to a hash function $\mathcal{H}$ that we assume[4] to exist. While our assumptions are new and therefore not standard, we believe they are quite likely to be true with respect to hash functions often used in practice. (For further discussion, see next section.) As a demonstration of the plausibility of this assumption, we show how to implement $\mathcal{H}$ and Ψ in the standard UC model with Common Reference String, *assuming only that one-way functions exist*. In particular, this also shows that our protocols give rise to UC-secure protocols in the CRS model, when the hash function is instantiated according to the construction we suggest. In this sense, our protocols are "no worse" than CRS UC protocols (this is omitted from this version).

Motivations, Our Work, and Previous Work. Soon after the UC framework was defined, it was observed that many important cryptographic tasks including commitment and zero knowledge, were impossible in the standard model [3, 4, 6]. Furthermore, it was recently shown that *any* model (with polynomial-time adversaries) seeking general composability in an "ideal" world / "real" world framework would suffer from the same impossibility results as the UC model [18]. Thus, if one seeks general composability in the plain setting with no trusted setup assumptions, the definitions must be changed in some significant manner. In our Γ-ES model, where Γ is allowed to base it answers to queries on the set of corrupted parties, these impossibility results no longer

[3]This follows from the fact that the ideal commitment functionality provides unconditional secrecy and binding.

[4]We stress that our assumptions are specific computational assumptions, for which a mathematical proof or refutation could exist. We are not assuming the existence of random oracles, or any other such "mythical" object.

hold.[5] Indeed, based on the assumptions outlined above, in Theorem 2 we show how to use the new framework to securely realize any multi-party computation with dishonest majority (for static adversaries), arguably the Holy Grail of modern cryptography, *without any trusted set-up assumptions*. (However we do this only for the case of *static adversaries*. Extending this to adaptive adversaries is left as an open problem here.)

We stress that prior to our work, under *any* kinds of computational assumptions, in the plain model very little was known regarding composability. Essentially, all results only deal with *self-composability* of 2-party protocols, not general composability. This work started with a sequence of work on Concurrent Zero Knowledge [11, 28, 16, 24], where an arbitrary polynomial number of concurrent executions can be handled. For general 2-party computations, recently it was shown that in the plain model self-composition for a *bounded* number of concurrent executions can be handled [17, 21]. We stress that our result is for *general* composition of general *multi*-party computation protocols for an *unbounded* number of concurrent executions. This result was only known previously in the presence of a trusted common reference string [7].

Finally, we point out that our protocols are conceptually simpler than the corresponding ones in [7] (and of course, do not use the common reference string). We believe that the new framework will lead to considerably more efficient and intuitive protocols.

New Tools and Techniques. We introduce some interesting techniques on the way to developing our final protocol for secure multi-party computation. We characterize the security of certain simple intermediate protocols (BCOM and BZK) in terms of non-standard functionalities that we introduce, tailor-made to suit these protocols. This is in contrast to the standard role of functionalities in the ES/UC framework. Indeed we suggest such non-standard functionalities as a way to demonstrate some level of security and composability in natural or simple protocols, a line further explored in [26]. A somewhat similar idea appears in [5] also, in the context of secure Key-Exchange. Our non-standard functionalities are designed to capture the secrecy requirements; but the correctness requirements need to be proven separately, "stepping outside" the gES framework. We point out that this is in contrast with the treatment of correctness and secrecy requirements in the ES/UC framework.

Finally, for our Ψ-ES model with Imaginary Angel Ψ, we show how to "implement" the Angel and related assumptions in the CRS model, assuming only one-way functions. (This result is omitted from the Proceedings version of this paper.) This may be of independent interest.

Going forward with the New Model. The new gES framework opens up a whole range of exciting possibilities. However we point out that one needs to be careful to understand the subtleties while working in this framework.

Note that in the Γ-ES-model, Γ is a single Imaginary Angel that defines the security model. If a protocol is shown secure in the Γ'-ES-model for another Imaginary Angel Γ', it may not compose with a Γ-ES-protocol. We point out that this is usually not a big problem because the specific nature of the Imaginary Angel will be used only for basic primitives and all other functionalities are built on top of it. For instance, in this work we use an Imaginary Angel Ψ only to realize a basic commitment functionality, which the other protocols build on. However it is the case that computational assumptions will typically need to be made relative to the Imaginary Angel. So it is desirable to have a standard Imaginary Angel model (or at most a few), relative to which the usual assumptions (one-way functions, trap-door permutations etc.) are well studied.

A user of this framework must keep in mind its main feature: Suppose a particular Imaginary Oracle Γ is fixed, and the user is considering adding a new protocol/functionality to the mix of Γ-ES-secure protocols. Then, the user must keep in mind that, in the security analysis of the protocol and security assessment of the ideal functionality, the adversary must be considered to have the specific super-polynomial computing resources given by Γ, even though in reality this is not the case.

Though candidates for our current assumptions may be instantiated by using some popular cryptographic hash function used in practice, the assumptions we make about them are non-standard. The main problem left open by this work is to find a way to base our result on more standard and better studied assumptions. Indeed, it will be interesting to come up with entirely new constructions and Imaginary Angels, for which the corresponding assumptions are better understood. Another possibility is to use *"complexity leveraging"* techniques to reduce some of the assumptions to more standard ones. See Section 2.2 for a discussion.

2. PRELIMINARIES

Notation. For two distributions $\mathcal{X}$ and $\mathcal{Y}$ with security parameter k, we write $\mathcal{X} \approx \mathcal{Y}$ to mean that $\mathcal{X}$ and $\mathcal{Y}$ are indistinguishable by probabilistic polynomial (in k) size circuits. We denote the distribution $f(\mathcal{X})$ by the set notation $\{f(x)|x \leftarrow \mathcal{X}\}$.

2.1 The Γ-ES Model

The Γ-ES model is the same as the ES/UC model in [3], except that the adversary and the environment are given access to an "Imaginary Angel" Γ. This is the case in the real, ideal and hybrid executions, as defined in the ES/UC model (see below). We stress, however, that **all protocols and honest parties** *are still polynomial-time, without any Imaginary Angels.* The Imaginary Angel is merely a means of defining and analyzing security. We allow the Imaginary Angel to base it answers on the set of corrupted parties. An Imaginary Angel Γ takes in a query q and returns an answer $\Gamma(q, \mathfrak{X})$, where $\mathfrak{X}$ is the set of corrupted parties at the time the query is made. We point out that there is a single Imaginary Angel Γ throughout the Γ-ES model.

Real, Ideal and Hybrid executions with an Imaginary Angel. We define REAL^{Γ} execution (with parties $P_1, \ldots, P_n$ running protocol π, an adversary $\mathcal{A}^{\Gamma}$, and an environment $\mathcal{Z}^{\Gamma}$) just like the REAL execution in [3],[6] except that now the

[5] If Γ is a fixed function (which does not depend on the set of corrupted parties), results of [3, 4, 6] will still imply impossibility of securely realizing the functionalities even if the adversary has access to Γ.

[6] Figure 1. in [3]

adversary $\mathcal{A}^\Gamma$ and environment $\mathcal{Z}^\Gamma$ have access to the Imaginary Angel Γ, which they may query any number of times. Analogous to $\text{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ in [3], we define $\text{REAL}^\Gamma_{\pi,\mathcal{A}^\Gamma,\mathcal{Z}^\Gamma}$ as the distribution ensemble (one distribution for each choice of security parameter and input to the parties) on the output produced by $\mathcal{Z}^\Gamma$ on interacting with the parties running protocol π and the adversary $\mathcal{A}^\Gamma$.

Similarly the IDEAL^Γ execution is defined exactly like the IDEAL execution in [3],[7] except that the environment $\mathcal{Z}^\Gamma$ and the ideal-execution adversary $\mathcal{S}^\Gamma$ have access to the Imaginary Angel Γ. Analogous to $\text{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$, we define $\text{IDEAL}^\Gamma_{\mathcal{F},\mathcal{S}^\Gamma,\mathcal{Z}^\Gamma}$ as the distribution ensemble of the output of $\mathcal{Z}^\Gamma$ on interacting with the "dummy" parties, the ideal functionality $\mathcal{F}$ and the ideal adversary (simulator) $\mathcal{S}^\Gamma$.

Finally, the HYB^Γ execution is defined as the hybrid execution in [3],[8] except that the environment $\mathcal{Z}^\Gamma$ and the hybrid-execution adversary $\mathcal{H}^\Gamma$ have access to the Imaginary Angel Γ. Analogous to $\text{HYB}^{\mathcal{F}}_{\pi,\mathcal{H},\mathcal{Z}}$, $\text{HYB}^{\Gamma,\mathcal{F}}_{\pi,\mathcal{H}^\Gamma,\mathcal{Z}^\Gamma}$ denotes the distribution ensemble on the output of $\mathcal{Z}^\Gamma$ on interacting with the parties running protocol π in the $\mathcal{F}$-hybrid model (with multiple copies of $\mathcal{F}$) and the hybrid-execution adversary $\mathcal{H}^\Gamma$. [9]

Note that above, if Γ is a polynomial time computable function (in particular if it is a trivial function which returns $\perp$ on all input), then the Γ-ES model is *identical* to the original ES/UC model.

DEFINITION 1. *A protocol π is said to Γ-ES-realize the functionality $\mathcal{F}$ against the class $\mathcal{C}$ of adversaries. if $\forall \mathcal{A}^\Gamma \in \mathcal{C}$, $\exists \mathcal{S}^\Gamma$ such that $\forall \mathcal{Z}^\Gamma$, $\text{IDEAL}^\Gamma_{\mathcal{F},\mathcal{S}^\Gamma,\mathcal{Z}^\Gamma} \approx \text{REAL}^\Gamma_{\pi,\mathcal{A}^\Gamma,\mathcal{Z}^\Gamma}$.*

The following is a restatement of the UC theorem in [3], where we replace the REAL and HYBRID executions by REAL^Γ and HYB^Γ executions respectively. The theorem holds for adaptive adversaries as well. The proof (as well as an extension to the *specialized simulator* case and simple generalizations) appears in the extended version of this paper [25].

THEOREM 1. **(Extended Universal Composition Theorem)** *Let $\mathcal{C}$ be a class of adversaries with access to the Imaginary Angel Γ, and $\mathcal{F}$ be an ideal functionality. Let π be an n-party protocol in the $\mathcal{F}$-hybrid model and let ρ be an n-party protocol that Γ-ES-realizes $\mathcal{F}$ against adversaries of class $\mathcal{C}$. Then, $\forall \mathcal{A}^\Gamma \in \mathcal{C}$, $\exists$ (a hybrid-model adversary) $\mathcal{H}^\Gamma \in \mathcal{C}$ such that $\forall \mathcal{Z}^\Gamma$ we have:*

$$\text{REAL}^\Gamma_{\pi^\rho,\mathcal{A}^\Gamma,\mathcal{Z}^\Gamma} \approx \text{HYB}^{\Gamma,\mathcal{F}}_{\pi,\mathcal{H}^\Gamma,\mathcal{Z}^\Gamma}$$

All the parties are assumed to have unique IDs, but possibly chosen adversarially. Like in previous works on Universally Composable multi-party computation we work in the authenticated channels model.

2.2 The Hash Function, the Imaginary Angel and the Assumptions

In this work, we use hash functions with concrete assumptions. Below we sketch the assumptions we use in this work.

[7]Figure 2. in [3]

[8]Figure 3. in [3]

[9]By abuse of notation, sometimes we will use $\text{REAL}^\Gamma_{\pi,\mathcal{A}^\Gamma,\mathcal{Z}^\Gamma}$, $\text{IDEAL}^\Gamma_{\mathcal{F},\mathcal{S}^\Gamma,\mathcal{Z}^\Gamma}$ and $\text{HYB}^{\Gamma,\mathcal{F}}_{\pi,\mathcal{H}^\Gamma,\mathcal{Z}^\Gamma}$ to denote (the distribution of) the entire view of the environment $\mathcal{Z}^\Gamma$, instead of (the distribution of) just its output.

We assume a hash function $\mathcal{H} : \{0,1\}^k \to \{0,1\}^\ell$, with the following properties: The k-bit input to $\mathcal{H}$ is considered to be an element $(\mu, r, x, b) \in \mathcal{I} \times \{0,1\}^{k_1} \times \{0,1\}^{k_2} \times \{0,1\}$, where $\mathcal{I}$ is the set of IDs used for the parties, and k_1, k_2, ℓ are all polynomially related to k. Then,

A1 (Collisions and Indistinguishability): For every $\mu \in \mathcal{I}$ and $r \in \{0,1\}^{k_1}$, there is a distribution $\mathcal{D}^\mu_r$ over $\{(x,y,z)|\mathcal{H}(\mu,r,x,0) = \mathcal{H}(\mu,r,y,1) = z\} \neq \phi$, such that

$$\{(x,z)|(x,y,z) \leftarrow \mathcal{D}^\mu_r\} \approx \{(x,z)|x \leftarrow \{0,1\}^{k_2}, z = \mathcal{H}(\mu,r,x,0)\}$$

$$\{(y,z)|(x,y,z) \leftarrow \mathcal{D}^\mu_r\} \approx \{(y,z)|y \leftarrow \{0,1\}^{k_2}, z = \mathcal{H}(\mu,r,y,1)\}$$

Further, even if the distinguisher is given sampling access to the set of distributions $\{\mathcal{D}^{\mu'}_{r'}|\mu' \in \mathcal{I}, r' \in \{0,1\}^{k_1}\}$, these distributions still remain indistinguishable.

A2 (Difficult to find collisions with same prefix): For all PPT circuits M and every id $\mu \in \mathcal{I}$, for a random $r \leftarrow \{0,1\}^{k_1}$, probability that $M(r)$ outputs (x,y) such that $\mathcal{H}(\mu,r,x,0) = \mathcal{H}(\mu,r,y,1)$ is negligible. This remains true even when M is given sampling access to the set of distributions $\{\mathcal{D}^{\mu'}_{r'}|\mu' \neq \mu, r' \in \{0,1\}^{k_1}\}$.

The first assumption simply states that there are collisions in the hash function, which are indistinguishable from a random hash of 0 or 1. Note that this assumption implies that for every $\mu \in \mathcal{I}$ and every $r \in \{0,1\}^{k_1}$ $\mathcal{H}(\mu,r,\{0,1\}^{k_2},0)$ and $\mathcal{H}(\mu,r,\{0,1\}^{k_2},1)$ are indistinguishable (because they are indistinguishable from $\{z|(x,y,z) \leftarrow \mathcal{D}^\mu_r\}$).

We make one more cryptographic assumption for our constructions:

A3 There exists a family of trapdoor permutations $\mathcal{T}$ over $\{0,1\}^n$, which remains secure even if the adversary has sampling access to $\mathcal{D}^\mu_r$ for all μ and r.

We use the notation $(f, f^{-1}) \leftarrow \mathcal{T}$ to specify generating a permutation $f : \{0,1\}^n \to \{0,1\}^n$ and its inverse (trapdoor) f^{-1}. We let $B(\cdot)$ denote a hardcore predicate associated with this permutation, which retains its security even if the adversary has sampling access to $\mathcal{D}^\mu_r$ for all μ and r, (for instance, it is easy to see that the Goldreich-Levin bit [14] continues to be a hardcore predicate as required, under Assumption A3). We will also need a perfectly binding (non-interactive) commitment scheme C, whose hiding property (in a stand-alone setting) holds against PPT adversaries with access to the distributions $\mathcal{D}^\mu_r$ for all μ and r. C can be readily constructed from $\mathcal{T}$ and B.

Plausibility of Our Assumptions.. Our set of assumptions on our hash function essentially give it the nature of a kind of non-malleable commitment (NMC). We make several observations here. NMC in the standard model is something that has been known to exist for over a decade [10], and recently even constant-round NMC has been realized in the standard model [1]. Further work on realizing simple NMC under standard complexity assumptions remains an important and exciting research area, partly because, tantalizingly, NMC is essentially something we know most functions satisfy (in the sense that a random oracle realizes it immediately), and it is something we expect any "sufficiently unstructured" hash function (such as something like SHA) to satisfy; indeed

we know that just one-way functions suffice to implement NMC in the CRS model [9, 8]. We make these observations to highlight two points: First, assuming that some hash function has NMC-like properties is not at all unreasonable. Second, since NMC is already known to exist, but known NMC protocols do not (and indeed *cannot*) yield the results we want, what we are doing is not just trivial given NMC – *i.e.* we are not making an assumption which "obviously" implies the goal we want to achieve.

As further evidence of the plausibility of our assumptions, we show that our hash functions, our assumptions, and the Imaginary Angel below can be realized in the CRS model *assuming only that one-way functions exist.* (This result is omitted from this version of the paper.) The fact that only one-way functions are needed to realize our assumptions in the CRS model gives further evidence that Assumption A3 is valid, since it is about *trapdoor* primitives, as opposed to merely *one-way* primitives.

Complexity Leveraging to Reduce Assumptions. By choosing parameters appropriately, at least one of our assumptions can be reduced to a more standard one. Specifically, Assumption A3, which assumes trapdoor permutations secure against adversaries with sampling access to $\mathcal{D}_r^\mu$ can be replaced with an assumption of trapdoor permutations secure against super-polynomial adversaries.

Consider choosing the domain of the trapdoor permutation $\{0,1\}^n$ such that the input size of the hash function $k = n^\epsilon$, for some constant $0 < \epsilon < 1$. Then we can safely replace Assumption A3 by the assumption that the trapdoor permutations are secure against circuits of size 2^{n^ϵ}. This implies Assumption A3 (given Assumptions A1 and A2) because a circuit of size $2^k = 2^{n^\epsilon}$ can represent the distributions $\mathcal{D}_r^\mu$ for all (μ, r). Note that this is only to change the assumption to a more standard one (trapdoor permutations secure against sub-exponential circuits), and has no effect on the model. In particular, we are not changing the power of the real or ideal adversaries.

The Imaginary Angel Ψ. Suppose $\mathfrak{X}$ is the set of corrupted parties. (Since we are dealing with static adversaries, this is a fixed set). On query (μ, r) the Imaginary Angel Ψ checks if $\mu \in \mathfrak{X}$, i.e., if the party with ID μ is corrupted or not. If it is, Ψ draws a sample from $\mathcal{D}_r^\mu$ described above and returns it; else it returns $\perp$. The results in this work are in the Ψ-ES-model.

2.3 Conventions

We point out a few conventions we follow in this work. All parties and functionalities referred to in the Γ-ES-model are (uniform or non-uniform) probabilistic polynomial time machines. Adversaries and environments are *non-uniform* PPT machines. The functionalities do not have access to any information about the system other than what the honest parties would have– in particular, a functionality would not know the set of corrupted parties. (In [7] such functionalities are referred to as "well-formed.")

When we say a protocol Γ-ES-realizes a functionality against static adversaries, we require that it be a *non-trivial* protocol (as defined in [7]): i.e., if the real world adversary corrupts no parties and forwards all messages promptly, the ideal world adversary (simulating the real-world execution with the protocol) is required do the same.

The following restrictions of the class of adversaries are standard. A *static* adversary can corrupt the parties only at the onset of computation. A *semi-honest* (or passive) adversary has read-only access to the internal state of the corrupted parties, but cannot modify the program run by the parties.

The following notation is also standard: if Π is a protocol in the $\mathcal{F}$-hybrid model (with Imaginary Angel Γ) and π is a protocol which securely realizes $\mathcal{F}$ (with respect to Γ) in $\mathcal{F}'$-hybrid model, then the protocol Π^π is a protocol in the $\mathcal{F}'$-hybrid model obtained from Π by replacing interaction with $\mathcal{F}$ by interaction with programs implementing the protocol π.

3. SECURE MULTI-PARTY COMPUTATION IN THE Ψ-ES MODEL

In this section we present our main result: for any multi-party computation (MPC) functionality $\mathcal{F}$, a protocol which Ψ-ES-realizes $\mathcal{F}$ against *static* adversaries. The overall structure of our Secure multi-party computation protocol follows that in [7], which in turn follows [15, 13]. But we differ from [7] in a very crucial manner: we introduce basic tools and protocols which allow us to achieve security (in the Ψ-ES model), *without a Common Random String.*

3.1 One-to-many Commit-and-prove

Following [7], first we construct a protocol which Ψ-ES-realizes $\mathcal{F}_{\text{CP}}^{1:\text{M}}$ against static adversaries, where $\mathcal{F}_{\text{CP}}^{1:\text{M}}$ is the "One-to-Many Commit-and-Prove" functionality shown in Figure 4 (see Section 6).

LEMMA 1. *Under assumptions A1, A2 and A3, there is a protocol which Ψ-ES-realizes $\mathcal{F}_{\text{CP}}^{1:M}$ against static adversaries.*

Lemma 1 contains the central contribution of this work. In Sections 4 and 5 we build tools for proving it, and in Section 6 we give the proof.

3.2 MPC from Commit-and-prove

Given Lemma 1, the rest of the construction closely follows that in [7]. First we begin with a protocol which Ψ-ES-realizes $\mathcal{F}$ against static *semi-honest* adversaries. A semi-honest adversary is one which does not alter the behavior of the parties it corrupts (see [7] for more details). Then using Lemma 1 we construct a *protocol compiler* which can take a protocol secure against semi-honest (static) adversaries and generates a protocol secure against general (static) adversaries, thereby completing the proof. These two steps are further elaborated below. For full details we refer the reader to [7].

MPC for Semi-Honest Parties. In general, all the proofs for the semi-honest case from [7] are information-theoretic, and immediately imply their Ψ-ES analogs. First, we observe that the Oblivious Transfer functionality (denoted by $\mathcal{F}_{\text{OT}}$) is realized by the same protocol as in [15, 13, 7]. The proof as given in [7] that the protocol securely realizes $\mathcal{F}_{\text{OT}}$ with respect to semi-honest static adversaries carries over directly to the Ψ-ES model, under Assumption A3.

This allows us to work in the $\mathcal{F}_{\text{OT}}$-hybrid model. Again, the protocols for semi-honest parties, in the $\mathcal{F}_{\text{OT}}$-hybrid model carry over exactly as they are given in [7]. As observed there,

there is no assumption on the computational power of the adversary and environment in the proof of security (under the $\mathcal{F}_{\text{OT}}$-hybrid model). Thus, using the secure realization of $\mathcal{F}_{\text{OT}}$ with respect to Ψ above, we get a secure multi-party computation protocol for semi-honest parties in the Ψ-ES model.

From the above, we conclude following:

LEMMA 2. *(Following [7]): Under Assumption A3, for any multi-party functionality $\mathcal{F}$, there exists a protocol which Ψ-ES-realizes $\mathcal{F}$ against* semi-honest *static adversaries.*

Protocol Compiler. As mentioned above, to complete the construction, we need to show how to convert the above protocol for semi-honest parties into one secure against malicious parties. We note that the compiler given in [7] under the $\mathcal{F}_{\text{CP}}^{1:\text{M}}$-hybrid model works in the Ψ-ES model as well. The proof in [7] that this compiler works in the $\mathcal{F}_{\text{CP}}^{1:\text{M}}$-hybrid model is information-theoretic, and holds for all classes of adversaries and environments; hence it is easily verified that the proof carries over to the Ψ-ES model.

LEMMA 3. *(Following [7]): There exists a protocol compiler* Comp *which takes a multi-party protocol Π, and outputs a protocol* $\mathsf{Comp}(\Pi)$ *in the $\mathcal{F}_{\text{CP}}^{1:M}$-hybrid model such that, for every protocol Π and static adversary $\mathcal{A}^{\Psi}$, there exists a* semi-honest *static adversary $\mathcal{A}'^{\Psi}$ such that for every environment $\mathcal{Z}^{\Psi}$,*

$$\text{REAL}^{\Psi}_{\Pi,\mathcal{A}'^{\Psi},\mathcal{Z}^{\Psi}} \equiv \text{HYB}^{\Psi,\mathcal{F}_{\text{CP}}^{1:M}}_{\mathsf{Comp}(\Pi),\mathcal{A}^{\Psi},\mathcal{Z}^{\Psi}}$$

Our main theorem readily follows.

THEOREM 2. *Under assumptions A1, A2 and A3, there is a protocol which Ψ-ES-realizes any multi-party functionality*[10] *against static adversaries.*

PROOF (SKETCH). Consider any multi-party functionality $\mathcal{F}$. By Lemma 2 there is a protocol Π which Ψ-ES-realizes $\mathcal{F}$ against semi-honest static adversaries. Applying Lemma 3, we obtain a protocol $\Pi' = \mathsf{Comp}(\Pi)$ in the $\mathcal{F}_{\text{CP}}^{1:\text{M}}$-hybrid model against (possibly malicious) static adversaries. Finally using Lemma 1 and the composition theorem, we get that $\Pi'^{\text{OM-CP}}$ Ψ-ES-realizes $\mathcal{F}$ against (possibly malicious) static adversaries. □

The rest of the paper is devoted to proving Lemma 1. Most of the proofs have been omitted (or replaced with short sketches) due to lack of space. The detailed proofs appear in the extended version of this paper [25].

4. BASIC BUILDING BLOCKS

In this section, we build the basic functionalities we need to achieve the result of secure (static) multi-party computation in the Ψ-ES model. Because we are not availing of any common reference string, our path is a bit more complicated than it would be otherwise. We introduce a new modeling and proof technique based on intermediate non-standard functionalities. In some cases, to establish our results, we need to "step outside" the Ψ-ES model, because our intermediate functionalities do not fully capture the security properties we need from our protocols for their later application. This section develops all the tools we'll need to realize the *commitment functionality* in the Ψ-ES model, which we'll do in the next section.

A note about session-ID's. In the ES/UC framework, and similarly in our gES framework, every functionality should be instantiated with a unique *session-ID* in order to distinguish it from other instantiations. This is an important part of the modeling, but it can be distracting in (often already complicated) protocols and functionality specifications. For sake of ease of reading, we omit session-ID's from our description, but they are implicit[11]. The extended version of this paper [25] explains the conventions underlying this implicit notation.

4.1 Basic Commitment Protocol

In Figure 1(a) we give a protocol BCOM for commitment, in the $\mathcal{F}_{\text{ENC}}$-hybrid. $\mathcal{F}_{\text{ENC}}$ is the encryption functionality, which receives a message from a party and delivers it to the destination party, publishing the length of the message to the adversary.

We will use protocol BCOM as a component in later protocols. Thus we would like to show some sort of composable security for this protocol. But note that this protocol cannot be a Ψ-ES secure commitment protocol (in particular, it does not provide a way for a simulator to *extract* the values committed to by a corrupted sender). So we introduce a novel technique to formalize and analyze the security of this protocol.

LEMMA 4. *Protocol* BCOM *Ψ-ES-realizes $\mathcal{F}_{\widetilde{\text{COM}}}$ shown in Figure 1(b) against static adversaries, in the $\mathcal{F}_{\text{ENC}}$-hybrid model.*

PROOF (SKETCH). For every PPT adversary $\mathcal{A}^{\Psi}$ we demonstrate a PPT simulator $\mathcal{S}^{\Psi}$ such that no PPT environment $\mathcal{Z}^{\Psi}$ can distinguish between interacting with the parties and $\mathcal{A}^{\Psi}$ in the REAL$^{\Psi}$ world, and interacting with the parties and $\mathcal{S}^{\Psi}$ in the IDEAL$^{\Psi}$ world.

$\mathcal{S}^{\Psi}$ internally runs $\mathcal{A}^{\Psi}$ (which expects to work in the $\mathcal{F}_{\text{ENC}}$-hybrid with the parties running the BCOM protocol), and works as an interface between $\mathcal{A}^{\Psi}$ and the parties. When $\mathcal{A}^{\Psi}$ starts the BCOM protocol, $\mathcal{S}^{\Psi}$ initiates a session with the IDEAL functionality. If $\mathcal{A}^{\Psi}$ corrupts both parties, $\mathcal{S}^{\Psi}$ allows it to directly interact with them. Below we consider the other three possible cases.

If $\mathcal{A}^{\Psi}$ corrupts neither of the two parties C and R, then all it sees are the random string r from R to C, and the message from $\mathcal{F}_{\text{ENC}}$ giving the length of the commit and reveal messages from C, all of which can be perfectly simulated. If $\mathcal{A}^{\Psi}$ corrupts the sender C alone, the simulation is straightforward, because the functionality and the protocol are identical on the sender's side.

Finally, suppose that the adversary corrupts the receiver alone. When $\mathcal{A}^{\Psi}$ sends out the first message r in the protocol, $\mathcal{S}^{\Psi}$ sends a query (μ_R, r) to the Imaginary Angel Ψ and (since R is corrupted), receives $(x, y, z) \leftarrow \mathcal{D}_r^{\mu_R}$, where $\mathcal{D}_r^{\mu_R}$ is the distribution over $\{(x, y, z) | \mathcal{H}(\mu_R, r, x, 0) =$

[10] see Section 2.3.

[11] Because there is no "joint state" represented by a CRS, we are in the lucky and relatively simple situation of only having to associate a *single* session-ID to each functionality (as opposed to a session-ID and a *sub-session-ID*). So almost all of the "complications" of dealing with multiple session-ID's that arise in [7] do not arise for us. This is one reason we feel comfortable omitting them from the protocol description, to avoid clutter.

Protocol BCOM

The parties are a sender or committer C, and a receiver R. The security parameter is k, and k_1, k_2 are polynomial in k. The sender C gets as input a bit b, which it wants to commit to.

COMMIT PHASE:

1. R picks $r \leftarrow \{0,1\}^{k_1}$ and sends it to C.
2. C chooses $r' \leftarrow \{0,1\}^{k_2}$ and computes $c = \mathcal{H}(\mu_R, r, r', b)$. C requests $\mathcal{F}_{\text{ENC}}$ to send c to R.
3. R receives c from $\mathcal{F}_{\text{ENC}}$ and accepts the commitment.

REVEAL PHASE:

1. C requests $\mathcal{F}_{\text{ENC}}$ to send (b, r') to R, which the receiver R receives.
2. R checks if $\mathcal{H}(\mu_R, r, r', b) = c$. If so it accepts b as revealed.

(a) The Basic Commitment Protocol (BCOM)

Functionality $\mathcal{F}_{\widetilde{\text{COM}}}$

The parties are sender C and receiver R, with an adversary $\mathcal{S}^{\Psi}$. The security parameter is k, and k_1, k_2 are polynomial in k.

COMMIT PHASE:

1. $\mathcal{F}_{\widetilde{\text{COM}}}$ picks $r \leftarrow \{0,1\}^{k_1}$ and sends it to C.
2. $\mathcal{F}_{\widetilde{\text{COM}}}$ receives c from C.
3. $\mathcal{F}_{\widetilde{\text{COM}}}$ sends the message COMMIT to R

REVEAL PHASE:

1. $\mathcal{F}_{\widetilde{\text{COM}}}$ receives (b, r') from C
2. $\mathcal{F}_{\widetilde{\text{COM}}}$ checks if $\mathcal{H}(\mu_R, r, r', b) = c$. If so it sends the message (REVEAL, b) to R and the adversary $\mathcal{S}^{\Psi}$.

(b) A functionality realized by the protocol BCOM

Figure 1: The Basic Commitment Protocol and a Functionality it realizes.

$\mathcal{H}(\mu_R, r, y, 1) = z\}$ as specified in the assumption on $\mathcal{H}$. Then, when $\mathcal{F}_{\widetilde{\text{COM}}}$ gives the COMMIT message, $\mathcal{S}^{\Psi}$ sends z to $\mathcal{A}^{\Psi}$ as a message from the REAL sender. Later if $\mathcal{F}_{\widetilde{\text{COM}}}$ gives the message $(\text{REVEAL}, 0)$, then $\mathcal{S}^{\Psi}$ sends $(0, x)$ to $\mathcal{A}^{\Psi}$, and if $\mathcal{F}_{\widetilde{\text{COM}}}$ gives the message $(\text{REVEAL}, 1)$, then $\mathcal{S}^{\Psi}$ sends $(1, y)$ to $\mathcal{A}^{\Psi}$. Under the assumption A1 on $\mathcal{D}_r^{\mu}$, we have that $\mathcal{Z}^{\Psi}$ cannot distinguish between the real execution and the simulation. □

A priori the functionality $\mathcal{F}_{\widetilde{\text{COM}}}$ does not offer any guarantee that the commitment is binding on a corrupt sender. The following lemma formulates the binding property outside the gES-framework (i.e., we do not give a functionality reflecting the binding property). A proof appears in the extended version of this paper [25].

LEMMA 5. *Consider a copy of $\mathcal{F}_{\widetilde{\text{COM}}}$ interacting with a corrupt sender C and an honest receiver R, in a system with environment $\mathcal{Z}^{\Psi}$ and multiple other copies of the same or other functionalities as well as one or more protocols and adversary $\mathcal{A}^{\Psi}$. Then, after finishing the commit phase, there is a fixed bit b^* (determined by the entire system state), such that C can make $\mathcal{F}_{\widetilde{\text{COM}}}$ accept a reveal to $1-b^*$ with only negligible probability.*

4.2 Basic Zero Knowledge Proof

Consider a proto-typical 3-round Zero Knowledge Proof protocol (a Σ-protocol) for proving membership in an NP-complete language (like 3-colorability or Hamiltonicity), in which the prover uses the basic commitment functionality $\mathcal{F}_{\widetilde{\text{COM}}}$ from above, to carry out the commitments (first round) and the reveals (last round). Let us denote this protocol by BZK. Then, like we defined $\mathcal{F}_{\widetilde{\text{COM}}}$ from BCOM, we can define a basic Zero Knowledge Proof functionality $\mathcal{F}_{\widetilde{\text{ZK}}}$ from BZK. The description of the functionality is simple: $\mathcal{F}_{\widetilde{\text{ZK}}}$ interacts with the prover according to the protocol BZK, playing the verifiers role. If the prover completes the proof according to the protocol, $\mathcal{F}_{\widetilde{\text{ZK}}}$ sends a message PROVEN to the verifier. Note that both BZK and $\mathcal{F}_{\widetilde{\text{ZK}}}$ are defined in the $\mathcal{F}_{\widetilde{\text{COM}}}$-hybrid model. Their exact specifications appear in the extended version of this paper [25]. There the following lemmas are proven.

LEMMA 6. *Protocol* BZK *Ψ-ES-realizes $\mathcal{F}_{\widetilde{\text{ZK}}}$ against static adversaries, in the $\mathcal{F}_{\widetilde{\text{COM}}}$-hybrid model.*

The functionality $\mathcal{F}_{\widetilde{\text{ZK}}}$ does not make any guarantees of soundness, *a priori*. But as with $\mathcal{F}_{\widetilde{\text{COM}}}$, this property can be established outside the gES-framework.

LEMMA 7. *Consider a corrupt prover P interacting with a copy of $\mathcal{F}_{\widetilde{\text{ZK}}}$ and an honest verifier V, in a system with environment $\mathcal{Z}^{\Psi}$ and multiple other copies of the same or other functionalities as well as one or more protocols (which can all be w.l.o.g considered part of the environment) and adversary $\mathcal{A}^{\Psi}$. Then $\mathcal{F}_{\widetilde{\text{ZK}}}$ accepts the proof to a false statement with negligible probability.*

5. COMMITMENT

The basic protocols and non-standard functionalities given in the previous section now allow us to achieve the "fully" ideal Ψ-ES commitment functionality $\mathcal{F}_{\text{COM}}$ given below. Since for the sake of simplicity in describing our protocols we allowed the session IDs to be implicit, we do the same in specifying the functionality.

Functionality $\mathcal{F}_{\text{COM}}$

The parties are a sender C and a receiver R, with adversary $\mathcal{S}^{\Psi}$.

COMMIT PHASE:

$C \rightarrow \mathcal{F}_{\text{COM}}$: b

$\mathcal{F}_{\text{COM}} \rightarrow R$: COMMIT

REVEAL PHASE:

$C \rightarrow \mathcal{F}_{\text{COM}}$: REVEAL

$\mathcal{F}_{\text{COM}} \rightarrow R, \mathcal{S}^{\Psi}$: (REVEAL, b)

Figure 2: The Commitment Functionality

Let C be a perfectly binding commitment scheme. Let $\mathcal{T}_k$ be a family of trapdoor-permutations (f, f^{-1}) on $\{0,1\}^k$. B stands for a hardcore predicate for the family of trapdoor permutations used. These primitives are assumed to be secure against adversaries with access to Ψ (see Section 2.2). The protocol is based on the commit-with-extract protocol from [2].

THEOREM 3. *Protocol* COM *Ψ-ES-realizes $\mathcal{F}_{\text{COM}}$ against static adversaries, in the $\mathcal{F}_{\widetilde{\text{ZK}}}$-hybrid model.*

Protocol COM

The parties are a sender C and a receiver R. k is the security parameter.

COMMIT PHASE:

1. R draws $r_R \leftarrow \{0,1\}^k$ and sends $c = \mathsf{C}(r_R; r)$ where r is also drawn at random.
2. C draws $(f, f^{-1}) \leftarrow \mathcal{T}_k$ and sends f to R. C interacts with $\mathcal{F}_{\widetilde{\text{ZK}}}$ to prove to R that $(\exists r', g : (f, g) \leftarrow \mathcal{T}_k(r'))$. R receives the message PROVEN from $\mathcal{F}_{\widetilde{\text{ZK}}}$.
3. C draws $r_C \leftarrow \{0,1\}^k$ and sends it to R.
4. R sends r_R to C.
5. R interacts with $\mathcal{F}_{\widetilde{\text{ZK}}}$ to prove to C that $(\exists r' : c = \mathsf{C}(r_R; r'))$ C receives the message PROVEN from $\mathcal{F}_{\widetilde{\text{ZK}}}$.
6. Let b be the bit C wants to commit to. C compute $b' = B(f^{-1}(r_R \oplus r_C)) \oplus b$ and sends b' to R. R accepts the commitment.

REVEAL PHASE:

1. C sends the bit b to R.
2. C interacts with $\mathcal{F}_{\widetilde{\text{ZK}}}$ to prove to R that $(\exists t : f(t) = r_R \oplus r_C \bigwedge b' = B(t) \oplus b)$.
3. Up on receiving the message PROVEN from $\mathcal{F}_{\widetilde{\text{ZK}}}$, R accepts b as the revealed bit.

Figure 3: Protocol COM which Ψ-ES-realizes $\mathcal{F}_{\text{COM}}$ against static adversaries

PROOF (SKETCH). Given an adversary $\mathcal{A}^\Psi$, we need to construct a simulator $\mathcal{S}^\Psi$ such that for all environments $\mathcal{Z}^\Psi$, we have $\text{HYB}^{\Psi,\mathcal{F}_{\widetilde{\text{ZK}}}}_{\text{COM},\mathcal{A}^\Psi,\mathcal{Z}^\Psi} \approx \text{IDEAL}^{\Psi}_{\mathcal{F}_{\text{COM}},\mathcal{S}^\Psi,\mathcal{Z}^\Psi}$. As is usual, $\mathcal{S}^\Psi$ internally simulates $\mathcal{A}^\Psi$. If both the sender C and receiver R running the protocol COM are corrupted, $\mathcal{S}^\Psi$ lets $\mathcal{A}^\Psi$ interact with them directly. We briefly discuss the other three cases below.

When both C and R are honest, $\mathcal{S}^\Psi$ simulates the protocol exactly until the step where the bit b is used (Step 7). At this step, it sends out a random bit as b'. In the reveal phase $\mathcal{S}^\Psi$ can easily simulate a proof from $\mathcal{F}_{\widetilde{\text{ZK}}}$ to open it either way. The hiding property of the hard-core bit B can be used to show that the simulation is indistinguishable from an actual execution. When R is corrupt and C is honest, the same simulator works, for the same reasons. However the reduction to the security of B becomes slightly more involved in this case.

When R is honest and C corrupt, $\mathcal{S}^\Psi$ should be able to *extract* the committed bit. The idea here is that $\mathcal{S}^\Psi$ (playing the part of R in the protocol) will cheat in the proof using the simulated $\mathcal{F}_{\widetilde{\text{ZK}}}$ in Step 5 (reveal phase of the coin-flipping part), and have $r_R \oplus r_C$ match a random string r such that it knows $B(r_R \oplus r_C)$. This will allow it to extract the bit b. Soundness of $\mathcal{F}_{\widetilde{\text{ZK}}}$ (Lemma 7) ensures that C cannot feasibly open to a bit other than b. Also it ensures that f is indeed a permutation, which along with the hiding property of the commitment C ensures that the simulation is indistinguishable from an actual execution.

The full proof is somewhat tedious, and appears in the extended version of this paper [25]. □

COROLLARY 4. *Under assumptions A1, A2 and A3, there is a protocol which Ψ-ES-realizes $\mathcal{F}_{\text{COM}}$ against static adversaries.*

PROOF (SKETCH). Employing the composition theorem (Theorem 1) to compose protocols in Theorem 3, and Lemmas 4 and 6, we conclude that there is a protocol in the $\mathcal{F}_{\text{ENC}}$-hybrid model which Ψ-ES-realizes $\mathcal{F}_{\text{COM}}$ against static adversaries. So to complete the proof we need to specify how to Ψ-ES-realize $\mathcal{F}_{\text{ENC}}$ against static adversaries. For this we use the same protocol as in [3], namely a CCA2-secure encryption with the receiving party generating the public-key/secret-key pair afresh for each session. But since we are working in the Ψ-ES model, we need the CCA2-secure encryption scheme to remain secure even when the adversary has access to Ψ. This can be accomplished based on assumption A3, by using any CCA2-secure encryption based on trapdoor permutations, for instance the one from [27]. □

6. ONE-TO-MANY COMMIT-AND-PROVE

In this section we outline the proof of Lemma 1, which completes the proof of our main theorem- Theorem 2. As in [7], we use two other functionalities, namely Zero-Knowledge ($\mathcal{F}_{\text{ZK}}$) and Authenticated Broadcast ($\mathcal{F}_{\text{BC}}$). Canetti and Fischlin [4] show how to UC-securely realize $\mathcal{F}_{\text{ZK}}$ in the $\mathcal{F}_{\text{COM}}$-hybrid model, in an information-theoretic sense: that is, without any computational assumptions, or restrictions on the class of adversaries. It is easy to show that the same protocol Ψ-ES-realizes $\mathcal{F}_{\text{ZK}}$ against static adversaries, in the $\mathcal{F}_{\text{COM}}$-hybrid model. Since we have already shown how to Ψ-ES-realize $\mathcal{F}_{\text{COM}}$ against static adversaries (Theorem 3), from the composition theorem, Theorem 1, it follows that there is a protocol which Ψ-ES-realizes $\mathcal{F}_{\text{ZK}}$ against static adversaries.

The functionality $\mathcal{F}_{\text{BC}}$ ensures that all the parties to which a message is addressed receive the same message (if they do receive the message). Following [7], we use the protocol from [12]. The protocol in [12] securely realizes $\mathcal{F}_{\text{BC}}$ in an information-theoretic manner: it does not require any computational restrictions on the class of adversaries. Thus, in particular, this protocol Ψ-ES-realizes $\mathcal{F}_{\text{BC}}$ against static adversaries.

Functionality $\mathcal{F}_{\text{CP}}^{\mathbf{1:M}}$

The parties are a sender C and a set of possible receivers $P_1, \ldots, P_n$, with an adversary $\mathcal{S}^\Psi$. The functionality is parameterized by a relation R. The security parameter is k.

COMMIT PHASE

- Upon receiving a message $(\text{COMMIT}, \mathcal{P}, w)$ from C where $\mathcal{P}$ is a set of parties and $w \in \{0,1\}^k$, append the value w to the list $\overline{w}$, record $\mathcal{P}$, and send the message $(\text{RECEIPT}, C, \mathcal{P})$ to all parties $P \in \mathcal{P}$ and to $\mathcal{S}^\Psi$. (Initially, the list $\overline{w}$ is empty). But, if a COMMIT message has already been received with a different set of parties $\mathcal{P}' \neq \mathcal{P}$ ignore this message.

PROVE PHASE

- Upon receiving a message (PROVE, x) from C, where $x \in \{0,1\}^{\text{poly}(k)}$, compute $R(x, \overline{w})$. If $R(x, \overline{w}) = 1$, then send the message (PROVEN, x) to all parties $P_i \in \mathcal{P}$ and to $\mathcal{S}^\Psi$. Otherwise, ignore the message.

Figure 4: The One-to-many commit-and-prove functionality

The proof of Lemma 1 easily follows from the following lemma and the observations above, using the composition theorem.

LEMMA 8. *There is a protocol which Ψ-ES-realizes $\mathcal{F}_{\text{CP}}^{1:M}$ against static adversaries in the ($\mathcal{F}_{\text{BC}}$,$\mathcal{F}_{\text{ZK}}$)-hybrid model (under Assumption A3).*

PROOF (SKETCH). To commit to a value w, the sender C computes a commitment c to w under a perfectly binding commitment C obtained from the trapdoor-permutation of Assumption A3 (which remains hiding even to adversaries with sampling access to $\mathcal{D}_r^\mu$ for all μ and r). Then it broadcasts c and proves to each party separately, using the $\mathcal{F}_{\text{ZK}}$ functionality, that c is indeed a valid commitment. Each party on receiving this proof broadcasts this fact. If all parties accept the respective proofs and announce it, they all proceed to accept the commitment by adding c to a list $\overline{c}$. Later, to prove $R(x, \overline{w})$, where x is an input and $\overline{w}$ is the list of all commitments made so far, the C proofs the statement (formulated in terms of x and $\overline{c}$) to each party separately using the $\mathcal{F}_{\text{ZK}}$ functionality. As before, on accepting the proof, each party broadcasts this fact. Finally they all accept the proof if all parties complete this broadcast step. It easily follows from the security of the commitment scheme C that this protocol Ψ-ES-realizes $\mathcal{F}_{\text{CP}}^{1;\text{M}}$ against static adversaries. □

Acknowledgments

We would like to thank Boaz Barak, Ran Canetti, Yuval Ishai, Yehuda Lindell, Oded Goldreich and Rafael Pass for many useful discussions. We also thank Jonathan Katz for suggesting a modification to the protocol in [2] on which our protocol COM (Figure 3) is based, thereby allowing us to simplify the proof of Theorem 3 considerably. We thank the anonymous referees whose detailed reviews were helpful in preparing this version.

7. REFERENCES

[1] Boaz Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. FOCS 2002: 345-355

[2] Boaz Barak and Yehuda Lindell. Strict Polynomial-time in Simulation and Extraction. Electronic Colloquium on Computational Complexity (ECCC)(026): (2002)

[3] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Electronic Colloquium on Computational Complexity (ECCC) (016): (2001) (Preliminary version in *IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2001.)

[4] Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO*, pages 19–40, 2001.

[5] Ran Canetti and Hugo Krawczyk. Universally Composable Notions of Key Exchange and Secure Channels. EUROCRYPT 2002: 337-351.

[6] R. Canetti, E. Kushilevitz, and Y. Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In *EUROCRYPT*, pages 68–86, 2003.

[7] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *ACM Symposium on Theory of Computing*, pages 494–503, 2002.

[8] Ivan Damgard and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. STOC 2003: 426-437

[9] Giovanni Di Crescenzo, Yuval Ishai and Rafail Ostrovsky. Non-Interactive and Non-Malleable Commitment. STOC 1998: 141-150

[10] Danny Dolev, Cynthia Dwork and Moni Naor. Nonmalleable Cryptography. SIAM J. Comput. 30(2): 391-437 (2000)

[11] Cynthia Dwork, Moni Naor and Amit Sahai. Concurrent Zero-Knowledge. STOC 1998. 409-418

[12] S. Goldwasser and Y. Lindell. Secure Computation without Agreement. DISC 2002: 17-32

[13] O. Goldreich. *Secure Multi-Party Computation*. Manuscript. Preliminary version, 1998. Available from `http://www.wisdom.weizmann.ac.il/~oded/pp.html`.

[14] O. Goldreich and L. Levin. A Hard Predicate for All One-Way Functions. In 21*st STOC,* pages 25–32, 1989.

[15] O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In *19th STOC,* pages 218–229, 1987. For details see [13].

[16] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in poly-logarithmic rounds. STOC 2001: 560-569.

[17] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. STOC 2003. 683-692.

[18] Yehuda Lindell. General composition and universal composability in secure multi-party computation. In *IEEE Symposium on Foundations of Computer Science*, pages 394-403, 2003.

[19] Moni Naor. Bit Commitment using Pseudorandom Generators. *Journal of Cryptology*, 4(2):151–158, 1991.

[20] Rafael Pass. Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition. EUROCRYPT 2003: 160-176.

[21] Rafael Pass and Alon Rosen. Bounded-Concurrent Secure Two-Party Computation in a Constant Number of Rounds. FOCS 2003: 404-413.

[22] B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In ACM Conference on Computer and Communications Security (CCS 2000), pp. 245 - 254, 2000.

[23] B. Pfitzmann and M. Waidner. A Model for Asynchronous Reactive Systems and its Application to Secure Message Transmission. In IEEE Symposium on Security and Privacy, 2001.

[24] Manoj Prabhakaran, Alon Rosen and Amit Sahai. Concurrent Zero Knowledge with Logarithmic Round-Complexity. FOCS 2002: 366-375

[25] Manoj Prabhakaran and Amit Sahai. New Notions of Security: Achieving Universal Composability without Trusted Setup. At the Cryptology ePrint Archive `http://eprint.iacr.org/` . 2004.

[26] Manoj Prabhakaran and Amit Sahai. Revisiting Concurrency: Monitored Functionalities and Client-Server Computation. Manuscript under preparation.

[27] Amit Sahai. Non-malleable Non-interactive Zero Knowledge and Adaptive Chosen Ciphertext Security. FOCS 1999: 543-553

[28] Ransom Richardson and Joe Kilian. On the Concurrent Composition of Zero-Knowledge Proofs. EUROCRYPT 1999: 415-431

[29] John Rompel. One-Way Functions are Necessary and Sufficient for Secure Signatures. STOC 1990: 387-394