Cryptography from Sunspots:
How to Use an Imperfect Reference String

Ran Canetti[*][†] IBM

Rafael Pass[†][‡] Cornell

abhi shelat U. Virginia

Abstract

The Common Reference String (CRS) model enables otherwise-impossible cryptographic goals such as removing interaction from protocols and guaranteeing composable security. However, the CRS model guarantees that the reference string is sampled from a fixed and known distribution (say, the uniform distribution); indeed, security analyses of all current protocols fail when the actual distribution of the reference string is allowed to differ from the specified one even by a small amount. This fact rules out a large class of potential implementations of the CRS model such as measurements of physical phenomena (like sunspots), or alternatively using random sources that might be adversarially influenced.

We study the possibility of obtaining universally composable (UC) security in a relaxed variant of the CRS model, where the reference string it taken from an adversarially specified distribution that's unknown to the protocol. On the positive side, we demonstrate that UC general secure computation is obtainable even when the reference string is taken from an arbitrary, adversarially chosen *distribution, as long as (a) this distribution has some minimal min-entropy, (b) it has not too long a description, (c) it is efficiently samplable, and (d) the sampling algorithm is known to the adversary (and simulator). On the negative side, we show that if any one of these four conditions is removed then general UC secure computation becomes essentially impossible.*

Keywords: UC Security, Setup Models, Common Reference String, Non black-box constructions, Entropy

1 Introduction

The Common Reference String (CRS) model [BFM88] is a useful model for designing and analyzing cryptographic protocols. One prevalent use, for instance, is for the construction of non-interactive zero knowledge (NIZK) proofs which are impossible to realize in the plain model of computation, e.g. [BFM88, BDMP91, FLS90]. Another use, which is the focus of this paper, is for constructing protocols with general secure composability guarantees. This too is known to be impossible in the plain model, e.g. [CF01, CLOS02, Lin04].

Informally, the CRS model assumes that the parties executing the protocol have access to a common string that is guaranteed to be taken from some pre-defined distribution, and that no "side information" on that string is known to anyone. This is a conceptually clean and intuitively appealing model. However, coming up with real-life instantiations of the CRS abstraction that are practically viable and yet preserve the security guarantees provided by the abstract model involves a number of difficulties. One difficulty is that the model requires all the participants in the protocol to trust a single "source of randomness." Another related difficulty with implementing the CRS model is that the model implicitly—but inherently—assumes that the reference string is used only for a single execution of a protocol (or alternatively by a set of well-coordinated executions); thus an implementation of the model needs to provide a new reference string for each new instance of a protocol that uses it. This limitation may make implementations unwieldy.

Consequently, efforts have been made in recent years to find relaxations and alternatives for the CRS model. A first relaxation is provided in [Pas04], where it is shown how to securely realize any multi-party functionality with universally composable (UC) security, using only UC zero-knowledge protocols between each pair of parties. This means that there is no need to use a single reference string that everyone trusts; it is enough to have each pair of parties obtain a string that the two of them trust. An alternative relaxation is to replace the global reference string with a mechanism where each participant registers with some trusted authority and obtains a public key [BCNP04]; here each participant needs to put full trust only in the authority it registers with; only a mild form of trust in the "well structuredness" of the public keys of other parties is needed.

[*] Supported by NSF grants CT-0430450 and CFF-0635297.

[†] Supported by US-Israel Binational Science Foundation Grant 2006317.

[‡] Supported by an I3P Identity Management and Privacy grant.

Yet another relaxation of the CRS model is to allow for a reference string (or registration process) that can be used by multiple protocol executions, and where security of each execution is guaranteed regardless of the makeup and behavior of the other protocol instances that use the same reference string (or public key) [CDPW07]. This allows the reference string (or public key) to be chosen "once and for all" and be used throughout the lifetime of the system.

However, the following property of the original CRS model is inherited by all the above relaxations: The model allows the protocol to precisely specify the distribution of the reference string. Indeed, some protocols in this model specify distributions that require a non-trivial sampling process, whereas others can do with relatively simple distributions such as the uniform distribution. Still, all existing protocols are very particular about the distributions they need in the sense that the security analysis quickly falls apart as soon as distribution of the reference string is changed even slightly.

This property is quite limiting. In particular, it seems to rule out "physical implementations" where the reference string is taken to be the result of joint measurement of some physical phenomenon such as an astronomic measurement (say, the size and shape of sunspots), fluctuations of the stock market, or even network delays across the Internet. Indeed, while it is reasonable to believe that such phenomena are largely unpredictable and uncontrollable, namely they have "high entropy", it is a stretch of the imagination to believe that they are taken from a distribution that is known to and useful for the protocol designer. (We remark that early works proposing the CRS model and its counterparts in the Game Theory community give such physical measurements, and sunspots in particular, as a main justification [BFM88, For88].)

This work addresses the following question: What security properties can be achieved given if only *imperfect* reference strings are available? More specifically, we concentrate on the feasibility of composable secure computation in a relaxed variant of the CRS model where the reference string might be adversarially controlled to some extent.

A first indication that this might not be an easy task is the result of Dodis et. al. [DOPS04] that demonstrates the impossibility of NIZK in a relaxed variant of the CRS model in which the distribution of the reference string can be arbitrary subject to having some minimal min-entropy. However, this result does not rule rule out composable protocols; more importantly, it does not consider the case where the reference string is guaranteed to be taken from an efficiently samplable distribution. Indeed, for such distributions deterministic extractors are known to exist (under computational assumptions) [TV00]. Thus, one might expect it to be possible to "compile" any protocol in the CRS model (or at least protocols that can do with a uniformly distributed reference string) into a protocol that uses a reference string that is taken from any efficiently samplable distribution that has sufficient min-entropy: First have the parties use a deterministic extractor to transform the reference string into a string that is almost uniformly distributed. Next, run the original protocol. Since the extracted string is almost uniform, one might expect the original analysis to work in the same way.

However, deterministic extractability turns out to be insufficient for this purpose. We prove impossibility of universally composable general secure computation in the presence of a reference string that is taken from an adversarially controlled distribution. (Specifically, we prove impossibility of UC commitment protocols.) Impossibility holds even if the chosen distribution is guaranteed to have full min-entropy minus a polynomially vanishing fraction, even if the distribution is guaranteed to be sampled via an *algorithmic* process, namely via a sampling process that has a relatively succinct description, and even when this process is guaranteed to be computationally efficient. Using standard techniques [Lin04, CKL03, Can01], this bound extends to rule out also UC protocols for other primitives, and also protocols that guarantee only general composability.

As a recourse, we restrict attention to the case where the algorithm for sampling the reference string is known to the adversaries involved. (Still, it is of course unknown to the protocol.) Here we show that general feasibility results for UC computation can indeed be recovered, as long as the reference string is taken from a distribution that is guaranteed to have a polynomial time sampling algorithm, a short description, and super-logarithmic min-entropy. Furthermore, we show that all three conditions are simultaneously necessary, in the sense that impossibility holds as soon as any one of the conditions is relaxed.

Before describing our results in more detail, we propose a high level motivation for the positive result. First, we stress that the proposed protocol works for a large class of distributions and requires no knowledge of the actual distribution in use. Still, it may appear over-optimistic to assume that the physical (or man-made) phenomena used to generate the reference string are governed by distributions where the sampling algorithm is computable in polynomial time. Indeed, why should Nature be governed by efficient algorithms? However, beyond the technical fact that these restrictions are necessary, one can view our analysis as a proof that any successful attack against the proposed protocols demonstrates that either the underlying hardness assumptions are violated, *or else that the process for choosing the reference string is not efficiently computable, or has long description.* This might be an interesting revelation in itself.

Another interpretation of the positive result is that it addresses situations where the process of choosing the reference string is influenced by an actual attacker. Here the guarantee that the distribution has some min-entropy represents the fact that the attacker's influence on the sampling process is limited.

Our results in more detail. We formulate our results within the generalized version of universally composable (GUC) security framework [Can01, CDPW07]. This allows us to formulate the set-up assumptions under consideration in a more precise manner. Recall that in this framework the CRS model is captured via an ideal functionality, $\mathcal{F}_{\text{CRS}}$, that is parameterized by a public, pre-specified distribution Δ. Moreover, a sampling algorithm D for Δ is specified, namely Δ is the distribution of $D(\rho)$ where ρ is a sufficiently long sequence of random bits. $\mathcal{F}_{\text{CRS}}$ sets $r = D(\rho)$, where ρ is the local random input of $\mathcal{F}_{\text{CRS}}$, and makes r available to the adversary and the parties in a given protocol instance. We say that protocol π UC-realizes some functionality $\mathcal{F}$ in the CRS model if for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{S}$ (called *simulator*) such that no environment can tell whether it interacts with $\mathcal{A}$ and π or with $\mathcal{S}$ and $\mathcal{F}$. In *both cases* the participants have access to $\mathcal{F}_{\text{CRS}}$. Having $\mathcal{F}_{\text{CRS}}$ give r to $\mathcal{A}$ and $\mathcal{S}$ represents the fact that r is not secret; Having $\mathcal{F}_{\text{CRS}}$ give r only to the parties in a specific protocol instance (and, in particular, not to the environment) represents the fact that only these parties, in this execution of their protocol, can trust that the reference string comes from the specified source.

Our first relaxation of the CRS is captured via the following ideal functionality, called $\mathcal{F}_{\text{BBSUN}}$. (Here SUN is reminder of the sunspot observation, and BB stands for "black-box"). Instead of treating the distribution Δ as a fixed, public parameter, we now let the environment determine the distribution by providing a description of a sampling algorithm D. Then, $\mathcal{F}_{\text{BBSUN}}$ chooses a sufficiently long random string ρ and computes the reference string $r = D(\rho)$. In addition, $\mathcal{F}_{\text{BBSUN}}$ lets the adversary (and simulator) obtain additional independent samples from the distribution "on the side". These samples are not seen by the environment or the parties running the protocol.

We consider the following three parameters of $\mathcal{F}_{\text{BBSUN}}$. First is the min-entropy, or "amount of randomness" of the reference string (measured over the random choices of both the environment and the sunspot functionality). Next is the runtime, or computational complexity of the sampling algorithm D. Last is the description-size of D (namely, the number of bits in its representation as a string); this quantity essentially measures the amount of randomness in the reference string that comes from the random choices of the environment. We measure all quantities as a function of the length n of the reference string; that is, we treat n as the security parameter. We can now informally state our first negative result:

Theorem 1 (informal): There exist no two-party protocols that UC-realize the ideal commitment functionality $\mathcal{F}_{\text{COM}}$ when given access to two instances of $\mathcal{F}_{\text{BBSUN}}$. This holds even if the distribution of the reference string is guaranteed to have min-entropy greater than $n - n^{\epsilon}$, and even if both the description size and the computational complexity of the provided sampling algorithm are guaranteed to be at most n^{ϵ}, for any $\epsilon > 0$.

Next we consider a more restricted setting, where the adversary has access to the "code", or description of the sampling algorithm D. This is modeled by having the set-up functionality explicitly send the description of D to the adversary. (Note that this relaxation is meaningful only for sampling algorithms that can be described in $poly(n)$ bits, else the adversary cannot read the description.) We call this functionality $\mathcal{F}_{\text{GBSUN}}$. Finally, we consider yet another variant, called $\mathcal{F}_{\text{SUN}}$, which gives to the adversary also the local random choices used to generate the reference string. It turns out that this variant provides an incomparable setup guarantee to that of $\mathcal{F}_{\text{GBSUN}}$. We can now informally state our second negative result:

Theorem 2 (informal): There exist no two-party protocols that UC-realize the ideal commitment functionality $\mathcal{F}_{\text{COM}}$ when given access to $O(1)$ instances of either $\mathcal{F}_{\text{GBSUN}}$ or $\mathcal{F}_{\text{SUN}}$. This holds even if either one of the following holds

1. The computational complexity of the algorithm can be super-polynomial in n, as long as the distribution of the reference string is guaranteed to have min-entropy $n - \text{poly}\log n$, and the description size of the provided sampling algorithm is guaranteed to be at most $\text{poly}\log n$ (assuming one-way functions with sub-exponential hardness).

2. The description size is at least $\mu(n) - \log n$, as long as the distribution of the reference string is guaranteed to have min-entropy $\mu(n) = n$ and the computational complexity is guaranteed to be at most $O(n)$.

3. The distribution of the reference has min-entropy at most $\log n$, as long as the description length is $O(1)$ and the computational complexity is $O(n)$.

In fact, in all the above cases we actually prove a slightly stronger result: We demonstrate a single distribution for the reference string which fails all candidate protocols for realizing $\mathcal{F}_{\text{COM}}$. Also, a slightly weaker impossibility result in terms of computational complexity holds unconditionally.

Next we turn to the positive result. We show:

Theorem 3 (informal): Assume there exist collision-resistant hash functions, dense crypto-systems and one-way functions with sub-exponential hardness. Then there exists a two-party protocol that UC-realizes the commitment functionality, $\mathcal{F}_{\text{COM}}$, when given access to $O(1)$ instances of either $\mathcal{F}_{\text{GBSUN}}$ or $\mathcal{F}_{\text{SUN}}$, as long as it is guaranteed that the min-entropy of the reference string is at least $\mu(n) = \text{poly}\log n$ the computational complexity of the provided sampling algorithm is at most $poly(n)$ and its description size is at most $\mu(n) - \text{poly}\log n$.

Furthermore, the protocol from Theorem 3 withstands even adaptive party corruptions, with no data erasure, whereas Theorems 1 and 2 apply even to protocols that only withstand static corruptions. Also, the protocol UC-realizes a *multi-instance* version of $\mathcal{F}_{\text{COM}}$—denoted $\mathcal{F}_{\text{MCOM}}$—where two parties can use the same reference string to exchange polynomially many commitments. A slightly weaker result with respect to min-entropy and description size, holds without assuming sub-exponential hardness.

Thus, under computational assumptions, Theorem 2 and 3 provide an essentially tight characterization of the feasibility of UC protocols, in terms of the min-entropy, computational complexity and description length of the reference string. Informally,

> *UC-security of non-trivial tasks is possible if and only if the reference string has min-entropy at least* $\mu(n) = \text{poly}\log n$*, and is generated by a computationally-efficient sampling algorithm with description length at most* $\mu(n) - \text{poly}\log n$*.*

Protocol techniques. To explain the main idea behind our protocol, we first sketch a simpler protocol that is only secure with respect to static corruptions. Also, the protocol aims to realize the ideal zero-knowledge functionality, $\mathcal{F}_{\text{ZK}}$, rather than $\mathcal{F}_{\text{MCOM}}$. (The last distinction is of lesser importance since either one suffices for general feasibility [CLOS02, Pas04].) The idea is to use a variation on Barak's protocol [Bar01]: Let L be an NP language and assume that a prover P wishes to prove to a verifier V that $x \in L$, having access to a reference string r that is taken from an unknown distribution with min-entropy at least $\mu = n^{\epsilon}$. Then, P and V will engage in a witness-indistinguishable proof that *"either $x \in L$ or the reference string r has a description of size $\mu/2$"*. (As in Barak's protocol, the description size is measured in terms of the Kolmogorov complexity, namely existence of a Turing machine M with description size $\mu/2$ that outputs r on empty input. Also, in order to guarantee that the protocol is simulatable in polynomial-time we require that M is polynomial time.) Soundness holds because in a real execution of the protocol, r is taken from a distribution with min-entropy at least μ, so the second part of the "or" statement is false with high probability. To demonstrate zero-knowledge, the simulator generates a simulated reference string $\tilde{r}$ by running the sampling algorithm D for the distribution on a *pseudo*random random-input. That is, the simulator chooses a random string $\tilde{\rho}$ of length, say, $\mu/2 - |D|$ (where $|D|$ denotes the description size of D) and computes $\tilde{r} = D(G(\tilde{\rho}))$, where G is some length-tripling pseudo-random generator. Now, $\tilde{r}$ indeed has description of size $\mu/2$ (namely, $\tilde{\rho}$ plus $|D|$ plus the constant-size description of G); furthermore, the simulator knows this description. Also, since both D and the environment are polynomial time, the simulated string $\tilde{r}$ is indistinguishable from the real string r.

The above protocol allows for straight-line simulation. It is not yet straight-line extractable, but it can be modified to be so using the techniques of [BL04]. Still, it is only secure against *static* corruptions of parties. In order to come up with a protocol that withstands *adaptive* corruptions we use a somewhat different technique, which combines the above idea with techniques from [CDPW07]. First, we move to realizing $\mathcal{F}_{\text{MCOM}}$. We then proceed in several steps: The first step is to construct a commitment scheme that is *equivocal* and adaptively secure. This is done using Feige and Shamir's technique [FS89] for constructing equivocal commitments from Zero-Knowledge protocols such as the one described above. Next, we use the constructed equivocal commitment scheme in a special type of a coin-tossing protocol, and use the obtained coin tosses as a reference string for a standard UC commitment protocol such as [CF01].

The protocol allows *two* parties to perform *multiple* commitment and decommitment operations between them, using only *two* reference strings —one for the commitments by each party. This means that in a multi-party setting it is possible to realize any ideal functionality using one reference string for each (ordered) pair of parties, regardless of the number of commitments and decommitment performed. Furthermore, each reference string needs to be trusted only by the two parties who use it.

Dealing with noisy measurements. Another caveat with implementing the CRS model via joint measurements of physical phenomena is that different parties may obtain somewhat different measurement values. In contrast, protocols in the CRS model, including the above protocol, need the parties in a protocol to have exactly the same value of the reference string.

A first attempt to get around this problem might be to use standard encoding mechanisms: Have the parties first apply the decoding procedure of an error-correcting code to the reference string, with the hope that if the Hamming distance between the two measured strings is small enough then the error-corrected strings will be identical. This approach, however, has a number of drawbacks. First, standard error-correcting codes bear no guarantee when applying the decoding procedure to arbitrary strings rather than to perturbed codewords. Second, applying error-correcting codes may result in a linear loss in min-entropy. This is not good enough when we only have super-logarithmic min-entropy to begin with.

We thus use a different technique, which is tied to our specific protocol. That is, we modify the protocol as follows: Assume we want to withstand measurement errors in up to δ bit locations of the reference string. Then, we modify the statement to be proven in zero-knowledge to: *"either $x \in L$ or I know a value r' that has a short description and is at most δ-far from your value of the reference string"*. Zero-knowledge holds in exactly the same way (the simulator can choose $r' = r$). Soundness holds using a similar argument, though with an error that depends on δ.

A remark on global set-up and plausible deniability. We emphasize that our reference string functionalities are explicitly modeled as a "local" set-up; namely, only the two parties sharing a reference string and the adversary have access to it. This stands in contrast with a "global" set-up which can be accessed by the environment and thus by other honest protocols. Indeed, by the results of [CDPW07] such a restriction is necessary. As a consequence, when instantiating our ideal functionalities with a real-world source, "plausible deniability" is only guaranteed as long as *honest* parties cannot (or choose not to) observe the reference string used by other parties or protocols. (Still, adversarial parties are, of course, allowed to see the reference strings even for protocols in which they do not participate.) Aside from the technical fact that such a restriction is necessary, this modeling arguably better captures the process of joint measurements of physical phenomena that are only accessible to the protocol instance that performs the measurement.

Organization. Section 2 defines our relaxed variants of the CRS model; §3 presents the impossibility results, §4 presents the basic UC commitment protocol; §5 present results concerning noisy measurements.

2 Model

We use the *generalized* version of the universally composable (UC) security framework [Can01, CDPW07]. Using the generalized version allows for a clearer and simpler modeling of the proposed relaxations of the CRS model while still providing general composable security guarantees via the universal composition theorem. Specifically, it allows modeling the fact that the set-up does not come as part of the protocol, but rather as part of the general execution environment. This allows capturing the basic premise that the protocol does not know exactly which set-up it is using. It also allows the complexity of the set-up to be unrelated to the complexity of the protocol. We assume familiarity with the modeling of [Can01, CDPW07]. See motivation and more details there and in [Can06].

The common high-entropy source ("sunspot") set-ups. The high-entropy source set-up model (or, **sunspot model** for short) provides the participating parties with a common string, along with the guarantee that the string is taken from a distribution that satisfies some basic properties. These properties include having sufficient amount of min-entropy, and having a sampling algorithm that is both computationally efficient and has a limited description length. The protocol does not know which particular distribution is used, and must function properly for *any* distribution for the reference string, as long as this distribution satisfies the stated properties. We formalize this model via an ideal functionality (with some variants) that captures the process of generating the reference string.

As mentioned in the Introduction, the common reference string functionality, $\mathcal{F}_{\text{CRS}}$, is parameterized by a sampling algorithm D, and some protocol instance with session identifier sid. In its first activation, $\mathcal{F}_{\text{CRS}}$ sets $r = D(\rho)$ where ρ is the local random input of $\mathcal{F}_{\text{CRS}}$. Next, to each query by either the adversary or a party with the session identifier sid, $\mathcal{F}_{\text{CRS}}$ responds with r.

Before introducing our relaxed formulation, we first recall how $\mathcal{F}_{\text{CRS}}$ is being used in the model of computation. Let π be an $\mathcal{F}_{\text{CRS}}$-hybrid protocol (namely, a protocol that makes use of $\mathcal{F}_{\text{CRS}}$), such that π is geared towards realizing some ideal functionality $\mathcal{F}$. In the basic UC modeling of [Can01], $\mathcal{F}_{\text{CRS}}$ is accessed only by π, and is formally treated as a "subroutine" of π. In contrast, the generalized UC modeling of [CDPW07] allows $\mathcal{F}_{\text{CRS}}$ to interact directly with the environment, and still be used by π. We adopt this modeling, since it allows capturing the basic premise that $\mathcal{F}_{\text{CRS}}$ represents a construct that is external to the execution of π. In particular, it may be influenced by the environment; furthermore, it exists even when π is replaced by $\mathcal{F}$. In addition, this formulation facilitates handling the case where the reference string might be sampled via a non-efficient process, hence $\mathcal{F}_{\text{CRS}}$ may not be polytime.[1]

To formally capture the fact that the distribution of the reference string can be unknown or adversarially controlled, we let the environment determine the sampling algorithm, hence the distribution. The first relaxation of $\mathcal{F}_{\text{CRS}}$, denoted $\mathcal{F}_{\text{BBSUN}}$, is used to model the case where the only way to access the distribution of the reference string to obtain samples from it; this holds not only for the honest parties but also the adversary. Whereas honest parties obtain only a single sample, the adversary is allowed to obtain multiple independent samples from the distribution. In addition, we formulate ideal functionalities that provide the adversary with the description of the sampling algorithm of the dis-

[1] We note that the main reason that motivates [CDPW07] to introduce the generalized formalism is to be able to allow $\mathcal{F}_{\text{CRS}}$ to give the reference string directly to entities external to the protocol execution, and in particular to the environment. Here we use the generalized formalism for a different purpose; in particular, we concentrate on the case where only the parties that actually participate in the protocol can directly "measure" the reference string.

tribution. Here we distinguish two variants. The first one, called $\mathcal{F}_{\text{GBSUN}}$ (here GB stands for "gray box") discloses the description of the sampling algorithm to the adversary, but keeps the local random choices hidden. The second variant, called $\mathcal{F}_{\text{SUN}}$, discloses also the local random choices to the adversary. Functionality $\mathcal{F}_{\text{SUN}}$ is presented in the figure below. Functionalities $\mathcal{F}_{\text{GBSUN}}$ and $\mathcal{F}_{\text{BBSUN}}$ are identical except that in $\mathcal{F}_{\text{GBSUN}}$, ρ is not sent to the adversary, whereas in $\mathcal{F}_{\text{BBSUN}}$ neither ρ nor D is sent to the adversary.

We remark that functionalities $\mathcal{F}_{\text{BBSUN}}$, $\mathcal{F}_{\text{GBSUN}}$ and $\mathcal{F}_{\text{SUN}}$ provide set-ups that are a-priori incomparable in strength. This is so since the corresponding setup functionality exists both in the real and the ideal executions, thus any additional advantages given to the adversary in the real world can also be helpful in the simulation.

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

1. Upon activation with session id sid proceed as follows. Send the message $(\texttt{Activated-Sun}, sid)$ to the environment, and wait to receive back a message $(\texttt{Distribution}, sid, D)$. Run the sampling algorithm D on a uniformly distributed random input ρ to obtain a reference string $r = D(\rho)$. Store D, ρ, r and send $(\texttt{CRS}, sid, D, r, \rho)$ to the adversary.
2. When receiving input ($\texttt{CRS}$,sid) from some party P with session id sid', send ($\texttt{CRS}$,sid, r) to that party if $sid = sid'$; otherwise ignore the message.
3. When receiving a request $(\texttt{NewSample}, sid)$ from the adversary proceed as follows. Run D on a fresh uniformly distributed random input ρ to obtain a reference string $r = D(\rho)$. and send $(\texttt{CRS}, sid, D, r, \rho)$ to the adversary.

In of themselves, $\mathcal{F}_{\text{BBSUN}}$, $\mathcal{F}_{\text{GBSUN}}$ and $\mathcal{F}_{\text{SUN}}$ do not provide any guarantees regarding the properties of the distribution of the reference string. Such guarantees are given by way of restricting the set of environments considered. (We do it this way so as to sidestep questions such as hoe to efficiently verify properties of a distribution.) Specifically, we concentrate on the following three quantities. The first quantity is the min-entropy of the distribution, which measures the "amount of unpredictability" in the reference string:

Definition 1 *A distribution Δ has min-entropy μ if* $\max_r \Pr_\Delta[r] < 2^{-\mu}$.

The second quantity is the runtime, or computational complexity of the sampling algorithm D. Third is the description-size of D, or number of bits in D's representation as a string. To simplify the notation, we measure all quantities as a function of the length n of the reference string. We also equate n with the security parameter.

The above three quantities are unrelated to each other and measure different aspects in which the reference string is "skewed away" from a perfect, uniformly distributed reference string. Indeed, the "ideal" sampling algorithm for the reference string would simply output its random input. Here a reference string of length n has min-entropy n, the complexity of the sampling algorithm is linear in n, and the description size is constant. In accordance, the guarantees one would need to provide regarding a reference string are that its distribution has high min-entropy, and that the sampling algorithm has small description size and can run in a small number of steps. This is formalized as follows.

Definition 2 *An environment machine $\mathcal{Z}$ is called (μ, d, t)*-**conforming** *if the following conditions hold:*

1. *Given security parameter 1^n, and upon receiving a message* $(\texttt{Activated-CRS}, sid)$ *from $\mathcal{F}_{\text{SUN}}$ (resp., $\mathcal{F}_{\text{BBSUN}}, \mathcal{F}_{\text{GBSUN}}$), $\mathcal{Z}$* directly *replies by sending back a message* $(\texttt{Distribution}, sid, D)$*, where the sampling algorithm D outputs reference strings of length n, has description size at most $d(n)$, and generates an output within $t(n)$ steps.*
2. *The distribution induced by the output of $\mathcal{F}_{\text{SUN}}$ (resp., $\mathcal{F}_{\text{BBSUN}}, \mathcal{F}_{\text{GBSUN}}$) in the execution by $\mathcal{Z}$ (on input 1^n) has min-entropy at least $\mu(n)$ (over the random choices of both $\mathcal{Z}$ and $\mathcal{F}_{\text{SUN}}$).*

Let π be an $\mathcal{F}_{\text{SUN}}$ (resp., $\mathcal{F}_{\text{BBSUN}}, \mathcal{F}_{\text{GBSUN}}$)-hybrid protocol. Then, π (μ, d, t)**-UC-realizes** *an ideal functionality $\mathcal{F}$ if for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{S}$ such that (μ, d, t)-conforming environments $\mathcal{Z}$ can distinguish between an interaction with π and $\mathcal{A}$ and an interaction with $\mathcal{F}$ and $\mathcal{S}$ only with probability that is negligible in n.*

A remark on our use of min-entropy Note that Definition 2 considers the min-entropy of an entire execution by $\mathcal{Z}$, and not just the min-entropy of the output of D. This only makes the set-up weaker as it imposes less restrictions on $\mathcal{F}_{\text{SUN}}$ (resp., $\mathcal{F}_{\text{BBSUN}}, \mathcal{F}_{\text{GBSUN}}$).

A remark on multiple sources Note that a protocol π in the $\mathcal{F}_{\text{BBSUN}}$ (resp. $\mathcal{F}_{\text{GBSUN}}$ and $\mathcal{F}_{\text{SUN}}$) hybrid model might invoke multiple instances of $\mathcal{F}_{\text{BBSUN}}$ (resp. $\mathcal{F}_{\text{GBSUN}}$ and$\mathcal{F}_{\text{SUN}}$). This corresponds to a scenario in which a protocol has access to multiple sources that each have some conditional min-entropy (but might otherwise be dependent); this follows since the environment can provide the description of, say, the second source after having seen the output (or even random coins) of the first source. In this paper we restrict our attention to protocols that only invoke a *constant* number of instances of $\mathcal{F}_{\text{BBSUN}}$ (resp $\mathcal{F}_{\text{GBSUN}}$ and$\mathcal{F}_{\text{SUN}}$).[2]

[2] In particular, our lower bounds hold for any constant number of instances, whereas our upper bounds requires only two instances.

A remark on the UC theorem. Definition 2 slightly modifies the definition of UC-emulation from [Can01] by restricting the set of environments under consideration. Consequently, one may wonder whether the universal composition theorem of [Can01] holds also for this definition. We note that this is indeed the case; in fact the proof remains unchanged, and one only needs to notice that the environments constructed in that proof remain conforming.

A remark on modeling run-times. In the next section we consider settings where the runtime $t(n)$ of the sampling algorithm is super-polynomial in n, whereas all other entities, including the environment, are polynomial in n. However, in the UC framework it is the invoker of each ITM that provides it with its runtime quota t (and then the amount t is deducted from the invoker's quota). This creates a technical problem, since in order for the environment to allow the functionality to run in time t, the environment itself has to have runtime t in the first place. To get around this technicality, we consider only environments that consist of a polynomial-time module plus a separate module that has (potentially super-polynomial) runtime quota t for the sole purpose of transferring this quota to the setup functionality.

3 Impossibility Results

This section presents impossibility results for composable protocols that use $\mathcal{F}_{\text{BBSUN}}$, $\mathcal{F}_{\text{GBSUN}}$ and $\mathcal{F}_{\text{SUN}}$. These results extend the impossibility results for two-party protocols that UC-realize $\mathcal{F}_{\text{COM}}$ in the plain model, where the only setup available is authenticated communication channels. We first recall $\mathcal{F}_{\text{COM}}$ below. (For simplicity, we give here the formulation for non-adaptive corruptions. In the case of adaptive corruptions one must slightly modify the formulation; see [Can01] for details.)

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

1. Upon receiving input $(\texttt{Commit}, sid, P_j, b)$ from P_i where $b \in \{0,1\}$, internally record the tuple (P_i, P_j, b) and send the message (sid, P_i, P_j) to the adversary; When receiving $(\texttt{ok})$ from the adversary, output $(\texttt{Receipt}, sid, P_i)$ to P_j. Ignore all subsequent $(\texttt{Commit}, ...)$ inputs.
2. Upon receiving a value $(\texttt{Open}, sid)$ from P_i, where a tuple (P_i, P_j, b) is recorded, send (b) to the adversary; When receiving $(\texttt{ok})$ from the adversary, output $(\texttt{Open}, sid, b)$ to P_j.

Black-box distributions

We show that, as long as the distribution of the reference string can be accessed by the adversary only in a "black-box" way, even relatively mild relaxations of the original CRS model render the reference string practically useless—at least from the point of view of obtaining composable security. That is, we extend the impossibility results for the plain model and show that there exist no two-party $\mathcal{F}_{\text{BBSUN}}$-hybrid protocols that UC-realize $\mathcal{F}_{\text{COM}}$, even when the distribution of the reference string is guaranteed to have min-entropy at least $n - \log^\epsilon n$ for some $\epsilon > 1$. Furthermore, this holds even if it is guaranteed that the sampling algorithm provided to $\mathcal{F}_{\text{BBSUN}}$ has description size at most n^ϵ, and runs in time at most $poly(n)$.

Theorem 1 *There exists a $c > 0$ such that for all $\epsilon > 0$ there does not exist any two-party protocols in the $\mathcal{F}_{\text{BBSUN}}$-hybrid model which invoke at most $O(1)$ instances of $\mathcal{F}_{\text{BBSUN}}$ that $(n - n^\epsilon, n^\epsilon, n^c)$-UC-realizes $\mathcal{F}_{\text{COM}}$.*

At high level, the proof proceeds as follows. Recall that in order to show that some protocol π UC-realizes $\mathcal{F}_{\text{COM}}$ using at most $m = O(1)$ invocations of $\mathcal{F}_{\text{BBSUN}}$, one has to describe a simulator that generates for the environment a view that is indistinguishable from its view of an interaction with π. In particular, this view includes values for the (at most m) reference strings used by π. We then proceed in two steps:

First, we show that the impossibility proof of [CF01] for realizing $\mathcal{F}_{\text{COM}}$ in the plain model can be extended to protocols that use $\mathcal{F}_{\text{BBSUN}}$, as long as it is guaranteed that the reference strings provided to the environment by the simulator are one of the values that were actually generated by $\mathcal{F}_{\text{BBSUN}}$ in that execution. To show this we rely on techniques from the black-box lower bounds of [GK96]. Briefly, recall that to show that a protocol UC-implements $\mathcal{F}_{\text{COM}}$ we need to show that the protocol is *equivocal* and *extractable*. Now, if the simulator uses an honestly generated sample of $\mathcal{F}_{\text{BBSUN}}$ as a reference string, but is still able to extract commitments, this simulator can also be used to violate the hiding property of the commitment. The simulator has of course the advantage of repeatedly asking for new reference strings until it finds ones that it likes, which is something a malicious receiver in the real protocol execution cannot; however, as the simulator is polynomial-time (and thus can ask at most a polynomial number of questions), this only improves its success probability by a polynomial fraction. To show this we here rely on the fact that protocol π only uses a constant number of invocations of $\mathcal{F}_{\text{BBSUN}}$.

Next, we make sure that the simulator only outputs reference strings that have been generated by $\mathcal{F}_{\text{BBSUN}}$. We achieve this by having $\mathcal{F}_{\text{BBSUN}}$ "sign" all the samples it gives to the simulator in a way that is verifiable by the environment. More specifically, the environment will first generate a key pair (sk, vk) for a signature scheme, and then feed $\mathcal{F}_{\text{BBSUN}}$ with a sampling algorithm D that includes a description of the signing key sk. Each run of D will then generate a pair (d, σ) where d is taken uniformly from $\{0,1\}^{n-|\sigma|}$ and $\sigma \leftarrow \text{sign}(d, sk)$. (It is stressed that sk is fixed throughout the lifetime of $\mathcal{F}_{\text{BBSUN}}$.)

Note that to implement step 2 we are required to assume the existence of signature schemes. However, this assumption is without loss of generality; standard techniques can be used to show that the existence of an implementation of $\mathcal{F}_{\text{COM}}$ in the $\mathcal{F}_{\text{BBSUN}}$-hybrid model (or even the $\mathcal{F}_{\text{CRS}}$ hybrid-model) implies the existence of one-way functions, and thus the existence of signatures [Rom90].

If one is willing to assume existence of one-way functions with sub-exponential hardness then it is possible to "scale down" the signature scheme even further, and thus demonstrate impossibility even with respect to distributions with entropy $n - \text{poly}\log n$ and sampling algorithms with description $\text{poly}\log n$.

Theorem 2 *Assume the existence of one-way functions with sub-exponential hardness. Then, there exist some $c > 0$ for which there does not exist any two-party $\mathcal{F}_{\text{BBSUN}}$-hybrid protocol, invoking at most $O(1)$ instances of $\mathcal{F}_{\text{BBSUN}}$, that $(n - \log^c n, \log^c n, n^c)$-UC-realizes $\mathcal{F}_{\text{COM}}$.*

Non-efficiently samplable distributions

We extend the impossibility result for protocols using $\mathcal{F}_{\text{BBSUN}}$ to protocols that use $\mathcal{F}_{\text{SUN}}$ and $\mathcal{F}_{\text{GBSUN}}$, as long as the sampling algorithm may run in (sub) exponential time. This demonstrates that, for such complex sampling algorithms, providing the simulator with the code of the sampling algorithm does not help, even if the description size of the algorithm is guaranteed to be polynomial.

Theorem 3 *There exist no $\epsilon > 0$ for which there exists a two-party $\mathcal{F}_{\text{SUN}}$-hybrid or $\mathcal{F}_{\text{GBSUN}}$-hybrid protocol, invoking at most $O(1)$ instances of $\mathcal{F}_{\text{SUN}}$ (resp $\mathcal{F}_{\text{GBSUN}}$), that $(n - n^\epsilon, n^\epsilon, 2^{n^\epsilon})$-UC-realizes $\mathcal{F}_{\text{COM}}$.*

Proof Idea: Consider the same distribution D described in the prior section, except that now D has only the *verification* key, vk. Instead of using the signing algorithm to sign the random string s, D uses its 2^{n^ϵ} runtime to forge a length-n^ϵ signature σ on s that passes the verification with vk. Note that the signatures verifiable by vk remain unforgeable for the simulator, even after seeing the code of D. Also, the only random choice made by D are in the choice of s; thus revealing these random choices to the adversary does not give it any additional information. The rest of the argument thus remains the same as in the proof of Thm. 2. □

We also observe that, if assuming the existence of one-way functions with sub-exponential hardness, we rule out even the case where the description length is of poly-logarithmic length and D runs in quasi-polynomial time. (This is done by using signatures of poly-logarithmic length.)

Theorem 4 *Assume the existence of one-way functions with sub-exponential hardness. Then, there exist values of $c > 0$ for which there does not exist any two-party $\mathcal{F}_{\text{SUN}}$-hybrid or $\mathcal{F}_{\text{GBSUN}}$-hybrid protocol, invoking at most $O(1)$ instances of $\mathcal{F}_{\text{SUN}}$ (resp $\mathcal{F}_{\text{GBSUN}}$), that $(n - \log^c n, \log^c n, n^{\log^c n})$-UC-realizes $\mathcal{F}_{\text{COM}}$.*

Distributions with long description

We extend the impossibility results to the case where the sampling algorithm has long description. In fact, we show that the description of the sampling algorithm must be smaller than $\mu(n) - \log n$ where μ denotes the min-entropy of the distribution.

Theorem 5 *Assume there exists a two-party $\mathcal{F}_{\text{SUN}}$-hybrid or $\mathcal{F}_{\text{GBSUN}}$-hybrid protocol, invoking at most $O(1)$ instances of $\mathcal{F}_{\text{SUN}}$ (resp $\mathcal{F}_{\text{GBSUN}}$), that $(\mu(n), d(n), O(n))$-UC-realizes $\mathcal{F}_{\text{COM}}$. Then $d(n) \leq \mu(n) - \log n$.*

Proof Idea: Consider an environment that picks an $n - O(\log n)$ bit random string r_1 and lets the sunspot functionality pick only an $O(\log n)$ bit random string r_2 and finally outputs $r_1 || r_2$. Clearly, such an environment is $(n, n - O(\log n), O(n))$ conforming. Additionally, it follows that an ideal-model simulator only gets a polynomial advantage over a real world adversary, as intuitively its only advantage is in picking r_2 (for a constant number of sources). □

Distributions with small min-entropy

We finally note that the same proof as for Theorem 5 can be used to rule out the case where the min-entropy of the sunspot is only $\log n$.

Theorem 6 *Assume there exists a two-party $\mathcal{F}_{\text{SUN}}$-hybrid or $\mathcal{F}_{\text{GBSUN}}$-hybrid protocol, invoking at most $O(1)$ instances of $\mathcal{F}_{\text{SUN}}$ (resp $\mathcal{F}_{\text{GBSUN}}$), that $(\mu, O(1), O(n))$-UC-realizes $\mathcal{F}_{\text{COM}}$. Then $\mu \geq \log n$.*

Proof Idea: The proof is essentially identical to the proof of Thm. 5. The only difference is that we here consider an environment $\mathcal{Z}$ that simply lets D be a machine that picks a $\log n$-bit long random string r and outputs it. □

4 Protocols in the $\mathcal{F}_{\text{SUN}}$ Model

This section shows that if the min-entropy of the distribution is not too low and the sampling algorithm is efficient and has not too long a description then it is possible to UC-realize $\mathcal{F}_{\text{COM}}$ in both the $\mathcal{F}_{\text{SUN}}$ and $\mathcal{F}_{\text{GBSUN}}$ hybrid models. In fact, it is possible to UC-realize the "multi-commitment" functionality $\mathcal{F}_{\text{MCOM}}$ using only one instance of either $\mathcal{F}_{\text{SUN}}$ or $\mathcal{F}_{\text{GBSUN}}$ per ordered pair of parties. Recall that $\mathcal{F}_{\text{MCOM}}$

is defined identically to $\mathcal{F}_{\text{COM}}$ with the exception that it allows a *single* pair of parties (one acting as a committer and one acting as a receiver) to exchange polynomially many commitments.[3] Recall that $\mathcal{F}_{\text{MCOM}}$ suffices to realize the two-party zero-knowledge functionality $\mathcal{F}_{\text{ZK}}$ [CF01] which in turn suffices to realize general multi-party computation [CLOS02, Pas04].

Theorem 7 *There exists a two-party protocol invoking only two instances of $\mathcal{F}_{\text{SUN}}$ (or $\mathcal{F}_{\text{GBSUN}}$) that (μ, d, t)-UC-realizes $\mathcal{F}_{\text{MCOM}}$, if dense cryptosystems exist, $t(n)$ is a polynomial and:*

1. *collision-resistant hash functions exist and $\mu(n) - d(n) > n^{\epsilon}$ for $\epsilon > 0$, or*
2. *one-way functions with sub-exponential hardness and collision-resistant hash functions exist and $\mu(n) - d(n) - \delta(n) > \log^{c}(n)$ for some specific $c > 1$ related to the (sub-exponential) hardness of the one-way function.*

We prove part (1) of the theorem; the second part follows a similar argument—the only difference is that sub-exponentially hard one-way functions imply the existence of pseudo-random generators that can expand $\text{poly}\log(n)$ bits into a random tape. These generators can then be used in place of the random generator used in part (1) of proof. in the construction of the equivocal scheme.

Overview of the construction

For ease of presentation we present our protocol and analysis in the $\mathcal{F}_{\text{SUN}}$-hybrid model, but the same analysis goes through also in the $\mathcal{F}_{\text{GBSUN}}$-hybrid model. Our construction proceeds in the two steps. In the first step, we construct an equivocal commitment given an imperfect reference string. In the second step, we use our equivocal commitment protocol to (μ, d, t)-realize $\mathcal{F}_{\text{MCOM}}$ in the $\mathcal{F}_{\text{SUN}}$-hybrid model.

Strong Equivocal Commitment In our first step we construct an equivocal (i.e. trapdoor) commitment scheme in the Common Reference String Model. In contrast to prior equivocal commitments schemes which remain secure only when then the reference string is sampled for a pre-specified distribution, our commitment scheme retains its security properties as long as the reference string satisfies the following 2 properties for some specified values α, β, where $\alpha - \beta > n^{\epsilon}$:

- With high probability, the reference string s has Kolmogorov complexity greater than α (i.e., the length of the shortest deterministic program that outputs s is greater than α).
- The reference string is generated by applying a deterministic and efficient program F, whose description length is at most β, on input a random string $r \in \{0,1\}^{\infty}$.

The high-level idea of our commitment protocol is to construct a commitment whose binding property holds as long as the Kolmogorov complexity of the reference string s is high, but can be totally violated whenever the Kolmogorov complexity is low. Then, the simulator can easily set-up a reference string that allows it to equivocate commitments: simply generate s by applying F to a pseudo-random string instead of a truly random string.

More precisely, we implement the above by relying on a variant of Feige-Shamir's trapdoor commitment. Let ϕ denote the statement that the reference string s has Kolmogorov complexity smaller than k. The sender commits to bit b by running the honest-verifier simulator for Blum's Hamiltonian Circuit protocol [Blu86] on input the statement ϕ and the verifier message b, generating the transcript (a, b, z), and finally outputting a as its commitment. In the decommitment phase, the sender reveals the bit b by providing both b, z. As in Feige-Shamir [FS89], binding follows from the (special)-soundness property of Blum's protocols, and Hiding follows from its zero-knowledge property. To equivocate commitments, the simulator sets up s by applying F on input r such that $r = g(r')$ where $|r'| \leq \delta$ and g is a pseudo-random generator. Since the statement ϕ is true it follows from the perfect completeness property of Blum's protocol that the simulator can provide valid decommitments to both $b = 0$ and $b = 1$. It additionally follows from the pseudo-random property of g and the honest-verifier ZK property of the Blum's protocols that the commitments created by the simulator are indistinguishable from real commitments.

The problem with the above description is that the statement ϕ above is not in $\mathcal{NP}$ (since there is no fixed polynomial upper-bound on the program F). As in [Bar01], we circumvent this problem, by instead letting the sender and receiver exchange 4 random strings (v_1, c_1, v_2, c_2), and then letting ϕ denote the statement that c_1, c_2 are commitments (using a commitment scheme with pseudo-random commitments) to messages p_1, p_2 such that (v_1, p_1, v_2, p_2) is an accepting transcript of a Universal argument [Mic94, Kil92, BG02] of the statement that the reference string s has Kolmogorov complexity at most k.[4]

An important feature of our commitment scheme is that in addition to the traditional equivocality property, our equivocation algorithm can also generate random coins for an honest committer which are consistent with the generated commitment. This property, which we denote *strong*

[3] We mention that previous definitions of $\mathcal{F}_{\text{MCOM}}$ [CF01] also allow multiple sets of participants. Here we do not need this extra generality.

[4] Whereas in [Bar01] it is enough that c_1 and c_2 are arbitrary commitments, we here require them to be random strings; this property will be useful next.

equivocality, will be critical in the next step of our construction. It follows from 1) the fact that the messages sent by the honest committer in the pre-amble phase are simply random strings, and 2) from a special property of the Blum protocol, called reverse-state generation in [CDPW07]: given the random coins used by the Blum protocol prover, it is possible to generate coins which when used by the simulator algorithm would produce the same transcript as the prover.

Realizing $\mathcal{F}_{\text{MCOM}}$ in the $\mathcal{F}_{\text{SUN}}$-Hybrid Model We show how to transform any strongly equivocal commitment scheme withstanding imperfect reference strings and whose decommitment phase is only a single message, into a (μ, d, t)-implementation of $\mathcal{F}_{\text{MCOM}}$ in the $\mathcal{F}_{\text{SUN}}$-Hybrid Model. We first note that (by relying our Compressibility Lemma stating that the output of a source with high min-entropy almost always has high Kolmogorov complexity) it follows that whenever $t = poly(n)$, $\mu(n) - d(n) > n^{\epsilon}$, it holds that for every (μ, d, t)-conforming environment, the "sun-spot" output of $\mathcal{F}_{\text{SUN}}$ has high Kolmogorov complexity but is generated by applying an efficient program F with short description on input a random string. Thus, the security of our equivocal commitment scheme holds with respect to (μ, d, t)-conforming environment. However, recall that to implement $\mathcal{F}_{\text{MCOM}}$ we require a commitment that is both equivocal and extractable. Towards also achieving extractability we employ a coin-flipping technique from [CDPW07]. The committer and receiver use coin-flipping (with the receiver moving first) in order to create a public key for an encryption scheme. Under the assumption of dense crypto-systems, this amounts to taking the xor of two random strings—one chosen by the receiver, and the other by the committer. The final commitment consists of a commitment to the bit b, and *either*—depending on the value of b—a random string or an encryption of the decommitment information under the key resulting from the coin-tossing.

In order to extract, the simulated receiver first uses the equivocal commitment procedure in the coin-tossing in order to force the coin-tossing to result in a special key for which it knows the secret decryption key. Given this key, the simulator will be able to straight-line extract the committed bit b. As a final note, by using an encryption scheme with pseudo-random keys and ciphertexts and by allowing the additional encryption to be occasionally just a random string (instead of always an encryption of the opening information), we can show the scheme enjoys adaptive security. This follows because when an honest party is corrupted, reasonable state information can be easily generated for either a "0" or a "1" commitment since the simulator can always pretend that the ciphertext generated for the 0 case was just a randomly chosen string. We here additionally rely on the *strong* equivocality property of the commitment scheme.

While our approach is similar to the one presented by Canetti, et.al. [CDPW07], the major difference from the prior work is that we do not make any reference to "identity-based" commitments, and our theorem holds for also for interactive equivocal commitments. Moreover, the prior analysis only handled a single commitment (i.e., they considered only $\mathcal{F}_{\text{COM}}$).[5]

Commitment Protocol SCOM

GEN$_n$**:** Produces a reference string O.

Let ℓ be the size of the non-interactive commitment with security parameter 1^n.

$S \longrightarrow R$ Send $(scom1, cid, S, R)$ to R.

$S \xleftarrow{m_1} R$ Upon receipt of $(scom1, cid, S, R)$ from S, run $m_1 \leftarrow V_1(O, n)$ and send to S.

$S \xrightarrow{c_1} R$ Pick $c_1 \leftarrow_r \{0,1\}^{\ell}$ and send to R.

$S \xleftarrow{m_2} R$ Run $m_2 \leftarrow V_2(O, k)$ and send to S.

$S \xrightarrow{c_2} R$ Pick $c_2 \leftarrow_r \{0,1\}^{\ell}$ and send to R.

$S \xrightarrow{a} R$ Run the Blum-protocol simulator on the theorem statement $(m_1, c_1, m_2, c_2) \in \pi(O, n, \alpha)$ using the challenge bit b:

$$(a, b, z) \leftarrow S_{\pi(O,n,\alpha)}((m_1, c_1, m_2, c_2), b)$$

Send a to R and store private state $(scom, cid, S, R, b, z)$.

R Record the commitment as $(scom, cid, S, R, x, a)$ where $x = (m_1, c_1, m_2, c_2)$.

DECOMMITMENT PROTOCOL SOPEN

$S \xrightarrow{b,z} R$ To open a commitment, S sends (b, z) to R.

R runs the Blum-protocol verifier on the triple $V_{\pi}(x, a, b, z)$ and accepts the decommitment to bit b if the verifier accepts the triple.

5 Dealing with noisy measurements

This section extends the construction of Section 4 to deal with the case where the participants in a protocol execution obtain somewhat different versions (or, "measurements") of the reference string. As a first step, we modify the set-up functionality $\mathcal{F}_{\text{SUN}}$ to capture this case. The modified functionality, $\mathcal{F}_{\text{MSUN}}$ (for "noisy sunspots"), is parameterized by a "closeness relation" R. It behaves just like $\mathcal{F}_{\text{SUN}}$ except that, instead of providing some party with the reference string r, $\mathcal{F}_{\text{MSUN}}$ obtains a "perturbed string" r' from the

[5] This was without loss of generality in their setting as they assumed a "globally" shared reference string. In contrast, since our reference strings are "local", we are instead required to directly implement $\mathcal{F}_{\text{MCOM}}$.

adversary; then, as long as (r, r') holds, $\mathcal{F}_{\text{MSUN}}$ returns r' as the value of the reference string.

To state our results we define the following measure on relations R over $\{0,1\}^* \times \{0,1\}^*$: R is said to have *radius* $\delta(\cdot)$ if for every $r \in \{0,1\}^*$ there exists at most $2^{\delta(|r|)}$ strings r' such that $R(r, r')$ holds.

Theorem 8 *There exists a two-party protocol invoking only two instances of the $\mathcal{F}_{\text{MSUN}}$ with respect to any efficient relation R with radius δ, that (μ, d, t)-UC-realizes $\mathcal{F}_{\text{MCOM}}$ if dense cryptosystems exist, $t(n)$ a polynomial and:*

1. *collision-resistant hash functions exist and $\mu(n) - d(n) - \delta(n) \ > n^\epsilon$ for $\epsilon > 0$, or*
2. *one-way functions with sub-exponential hardness and collision-resistant hash functions exist and $\mu(n) - d(n) - \delta(n) > \log^c(n)$ for some specific $c > 1$ related to the (sub-exponential) hardness of the one-way function.*

We complement the above positive result by showing that the extra description-length requirement is necessary.

Theorem 9 *Assume there exists a two-party $\mathcal{F}_{\text{MSUN}}$-hybrid protocol, invoking at most $O(1)$ instances of $\mathcal{F}_{\text{MSUN}}$, that $(\mu(n), d(n), O(n))$-UC-realizes $\mathcal{F}_{\text{COM}}$ with respect to every relation R with radius δ. Then $d(n) \leq \mu(n) - \delta(n) - \log n$.*

References

[Bar01] B. Barak. How to go beyond the black-box simulation barrier. In *FOCS 01*, pages 106–115, 2001.

[BCNP04] B. Barak, R. Canetti, J. B. Nielson, and R. Pass. Universally composable protocols with relaxed setup assumptions. In *FOCS 04*, pages 186–195, 2004.

[BDMP91] M. Blum, A. De Santis, S. Micali, and G. Persiano. Noninteractive zero-knowledge. *SIAM J. Computing*, 20(6):1084–1118, 1991.

[BFM88] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC 88*, pages 103–112, 1988.

[BG02] B. Barak and Goldreich. Universal arguments and their applications. In *CCC*, pages 194–203, 2002.

[BL04] B. Barak and Y. Lindell. Strict polynomial-time in simulation and extraction. *SIAM J. Comput*, 33(4):738–818, 2004.

[Blu86] M. Blum. How to prove a theorem so no one can claim it. In *Proc. of The International Congress of Mathematicians*, pages 1444–1451, 1986.

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS 01*, pages 136–145, 2001.

[Can06] Ran Canetti. Security and composition of cryptographic protocols: A tutorial. Cryptology ePrint Archive, Report 2006/465, 2006.

[CDPW07] R. Canetti, Y. Dodis, R. Pass, and S. Walfish. Universally composable security with global setup. In *TCC*, pages 61–85, 2007.

[CF01] R. Canetti and M. Fischlin. Universally composable commitments. In *CRYPTO*, 2001.

[CKL03] R. Canetti, E. Kushilevitz, and Y. Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In *EUROCRYPT 03*, 2003.

[CLOS02] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *STOC 02*, pages 494–503, 2002.

[DOPS04] Y. Dodis, S. Ong, M. Prabhakaran, and A. Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS 04*, pages 196–205, 2004.

[FLS90] U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero knowledge proofs based on a single random string. In *FOCS 90*, pages 308–317, 1990.

[For88] F. Forges. Can sunspots replace a mediator? *J. of Math. Ec.*, 17(4):347–368, 1988.

[FS89] U. Feige and A. Shamir. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO*, pages 526–544, 1989.

[GK96] O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Jour. on Computing*, 25:169–192, 1996.

[Kil92] J. Kilian. A note on efficient zero-knowledge proofs and arguments. In *STOC 92*, pages 723–732, 1992.

[Lin04] Y. Lindell. Lower bounds for concurrent self composition. In *TCC*, pages 203–222, 2004.

[Mic94] S. Micali. CS (computationally-sound) proofs. In *FOCS 94*, pages 436–453, 1994.

[Pas04] R. Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC 04*, pages 232–241, 2004.

[Rom90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC 90*, pages 387–394, 1990.

[TV00] L. Trevisan and S. Vadhan. Extracting randomness from samplable distributions. In *FOCS 00*, pages 32–42, 2000.