Counterexamples to Hardness Amplification beyond Negligible

Yevgeniy Dodis[1], Abhishek Jain[2], Tal Moran[3], and Daniel Wichs[4]

[1] New York University (NYU)
[2] University of California, Los Angeles (UCLA)
[3] Interdisciplinary Center (IDC) Herzliya
[4] IBM Research, T.J. Watson

Abstract. If we have a problem that is mildly hard, can we create a problem that is significantly harder? A natural approach to *hardness amplification* is the "direct product"; instead of asking an attacker to solve a single instance of a problem, we ask the attacker to solve several independently generated ones. Interestingly, proving that the direct product amplifies hardness is often highly non-trivial, and in some cases may be false. For example, it is known that the direct product (i.e. "parallel repetition") of general interactive games may not amplify hardness at all. On the other hand, positive results show that the direct product does amplify hardness for many basic primitives such as one-way functions, weakly-verifiable puzzles, and signatures.

Even when positive direct product theorems are shown to hold for some primitive, the parameters are surprisingly weaker than what we may have expected. For example, if we start with a weak one-way function that no poly-time attacker can break with probability $> \frac{1}{2}$, then the direct product provably amplifies hardness to *some negligible* probability. Naturally, we would expect that we can amplify hardness exponentially, all the way to 2^{-n} probability, or at least to some fixed/known negligible such as $n^{-\log n}$ in the security parameter n, just by taking sufficiently many instances of the weak primitive. Although it is known that such parameters cannot be proven via black-box reductions, they may seem like reasonable conjectures, and, to the best of our knowledge, are widely believed to hold. In fact, a conjecture along these lines was introduced in a survey of Goldreich, Nisan and Wigderson (ECCC '95). In this work, we show that such conjectures are *false* by providing simple but surprising counterexamples. In particular, we construct weakly secure *signatures* and *one-way functions*, for which standard hardness amplification results are known to hold, but for which hardness does not amplify *beyond just negligible.* That is, for *any negligible* function $\varepsilon(n)$, we instantiate these primitives so that the direct product can always be broken with probability $\varepsilon(n)$, *no matter how many copies we take.*

1 Introduction

Hardness amplification is a fundamental cryptographic problem: given a "weakly secure" construction of some cryptographic primitive, can we use it to build a

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 476–493, 2012.

"strongly secure" construction? The first result in this domain is a classical conversion from weak one-way functions to strong one-way function by Yao [32] (see also [13]). This result starts with a function f which is assumed to be *weakly one-way*, meaning that it can be inverted on at most (say) a *half* of its inputs. It shows that the *direct-product* function $F(x_1, \ldots, x_k) = (f(x_1), \ldots, f(x_k))$, for an appropriately chosen polynomial k, is one-way in the standard sense, meaning that it can be inverted on only a *negligible* fraction of its inputs. The above result is an example of what is called the *direct product theorem*, which, when true, roughly asserts that simultaneously solving many independent repetitions of a mildly hard task yields a much harder "combined task".[1] Since the result of Yao, such direct product theorems have been successfully used to argue security amplification of many other important cryptographic primitives, such as collision-resistant hash functions [8], encryption schemes [12], weakly verifiable puzzles [7,20,22], signatures schemes/MACs [11], commitment schemes [18,9], pseudorandom functions/generators [11,26], block ciphers [24,27,25,30], and various classes of interactive protocols [5,28,19,17].

Direct product theorems are surprisingly non-trivial to prove. In fact, in some settings, such as general interactive protocols [5,29], they are simply false and hardness does not amplify at all, irrespective of the number of repetitions. Even for primitives such as one-way functions, for which we do have "direct product theorems", the parameters of these results are surprisingly weaker than what we may have expected. Let us say that a cryptographic construction is *weakly secure* if no poly-time attacker can break it with probability greater than $\frac{1}{2}$. Known theorems tell us that the direct product of $k = \Theta(n)$ independent instances of a weakly secure construction will become secure in the standard sense, meaning that no poly-time attacker can succeed in breaking security with better than *some negligible probability in the security parameter n*. However, we could naturally expect the direct product of k instances will amplify hardness exponentially, ensuring that no poly-time attacker can break security with more than 2^{-k} probability. Or, we would at least expect that a sufficiently large number of $k = \mathsf{poly}(n)$ repetitions can amplify hardness to some fixed/known negligible probability such as $\varepsilon(n) = 2^{-n^{\delta}}$ for some constant $\delta > 0$, or even less ambitiously, $\varepsilon(n) = n^{-\log n}$. We call such expected behavior *amplification beyond negligible.*

Limitation of Existing Proofs. One intuitive reason that the positive results are weaker than what we expect is the limitation of our reduction-based proof techniques. In particular, assume we wanted to show that the k-wise direct product amplifies hardness down to some very small probability ε. Then we would need an *efficient reduction* that uses an adversary $\mathcal{A}$ breaking the security of the k-wise direct product with probability ε, to break the security of a single instance with a much larger probability, say one half. Unfortunately, the

[1] A related approach to amplifying the hardness of decisional problems is the "XOR Lemma" which roughly asserts the hardness of predicting an XOR of the challenge bits of many independent instances of a decisional problem will amplify. In this work, we will focus of "search" problems such as one-way functions and signatures and therefore only consider amplification via direct product.

reduction cannot get "anything useful" from the attacker $\mathcal{A}$ until it succeeds at least once. And since $\mathcal{A}$ only succeeds with small probability ε, the reduction is forced to run $\mathcal{A}$ at least (and usually much more than) $1/\varepsilon$ times, since otherwise $\mathcal{A}$ might never succeed. In other words, the reduction is only efficient as long as ε is an *inverse polynomial.* This may already be enough to show that the direct product amplifies hardness *to some negligible probability*, since the success probability of $\mathcal{A}$ must be smaller than *every* inverse polynomial ε. But it also tells us that black-box reductions cannot prove any stronger bounds *beyond negligible*, since the reduction would necessarily become inefficient.[2] For example, we cannot even prove that the k-wise direct product of a weak one-way function will amplify hardness to $n^{-\log n}$ security (where n is the security parameter), no matter how many repetitions k we take.

Our Question. The main goal of this work is to examine whether the limitations of current hardness amplification results are just an artifact our proof technique, or whether they reflect reality. Indeed, we may be tempted to ignore the lack of formal proofs and nevertheless make the seemingly believable conjecture that *hardness does amplify beyond negligible.* In more detail, we may make the following conjecture:

Conjecture (Informal): *For all primitives for which standard direct product theorems hold (e.g., one-way functions, signatures etc.), the k-wise direct product of* any *weakly secure instantiation will amplify hardness all the way down to some fixed negligible bound $\varepsilon(n)$, such as $\varepsilon(n) = 2^{-\Omega(n)}$, or, less ambitiously, $\varepsilon(n) = n^{-\log n}$, when $k = \mathsf{poly}(n)$ is sufficiently large.*

To the best of our knowledge, such a conjecture is widely believed to hold. The survey of Goldreich et al. [14] explicitly introduced a variant of the above conjecture in the (slightly different) context of the XOR Lemma and termed it a *"dream version"* of hardness amplification which, although seemingly highly reasonable, happens to elude a formal proof.

Our Results. In this work, we show that, surprisingly, the above conjecture does *not* hold, and give strong counterexamples to the conjectured hardness amplification beyond negligible. We do so in the case of signature schemes and one-way functions for which we have standard direct-product theorems showing that hardness amplifies to negligible [32,11]. Our result for the signature case, explained in Section 3, relies on techniques from the area of *stateless (resettably-secure) multiparty computation* [6,3,10,16,15]. On a high level, we manage to embed an execution of a stateless mutliparty protocol Π into the design of our signature scheme, where Π generates a random instance of a hard relation $\mathcal{R}$, and the signer will output its secret key if the message contains a witness for $\mathcal{R}$. The execution of Π can be driven via carefully designed signing queries. Since Π is secure and $\mathcal{R}$ is hard, the resulting signature scheme is still secure by itself.

[2] This "folklore" observation has been attributed to Steven Rudich in [14].

However, our embedding is done in a way so as to allow us to attack the direct product of *many independent* schemes by forcing them to execute a *single correlated* execution of Π resulting in a common instance of the hard relation $\mathcal{R}$. This allows us to break all of the schemes simultaneously by breaking a single instance of $\mathcal{R}$, and thus with some negligible probability $\varepsilon(n)$, which is independent of the number of copies k. Indeed, we can make $\varepsilon(n)$ an arbitrarily large negligible quantity (say, $n^{-\log n}$) by choosing the parameters for the relation $\mathcal{R}$ appropriately.

One may wonder whether such counterexamples are particular to signature schemes. More specifically, our above counterexample seems to crucially rely on the fact that the security game for signatures is highly interactive (allowing us to embed an interactive MPC computation) and that the communication complexity between the challenger and attacker in the security game can be arbitrarily high (allowing us to embed data from all of the independent copies of the scheme into the attack on each individual one). Perhaps hardness still amplifies beyond negligible for simpler problems, such as one-way functions, where the security game is not interactive and has an a-priori bounded communication complexity. Our second result gives strong evidence that this too is unlikely, by giving a counterexmaple for one-way functions. The counterexample relies on a new assumption on a hash functions called *Extended Second Preimage Resistance* (ESPR), which we introduce in this paper. Essentially, this assumption says that given a random challenge x, it is hard to find a bounded-length *Merkle path* that starts at x, along with a collision on it. To break many independent copies of this problem, the attacker takes the independent challenges $x_1, \ldots, x_k$ and builds a *Merkle tree* with them as leaves. If it manages to find a *single* collision at the root of tree (which occurs with some probability independent of k), it will be able to find a witness (a Merkle path starting at x_i with a collision) for *each* of the challenges x_i. So far, this gives us an amplification counterexample for a *hard relation* based on the ESPR problem (which is already interesting), but, with a little more work, we can also convert it into a counterexample for a *one-way function* based on this problem. For the counterexample to go through, we need the ESPR assumption to hold for some fixed hash function (not a family), and so we cannot rely on collision resistance. Nevertheless, we argue that the ESPR assumption for a fixed hash function is quite reasonable and is likely satisfied by existing (fixed) cryptographic hash functions, by showing that it holds in a variant of the random oracle model introduced by Unruh [31], where an attacker gets arbitrary "oracle-dependent auxiliary input". As argued by [31], such model is useful for determining which security properties can be satisfies by a single hash function rather than a family.

Overall, our work gives strong indications that the limitations of our reductionist proofs for the direct product theorems might actually translate to real attacks for some schemes.

Related Work. Interestingly, a large area of related work comes from a seemingly different question of *leakage amplification* [2,1,23,21]. These works ask the following: given a primitive P which is resilient to ℓ bits of leakage on its secret key, is it true that breaking k independent copies of P is resilient to almost $L = \ell k$ bits of leakage? At first sight this seems to be a completely unrelated question. However, there is a nice connection between hardness and leakage-resilience: if a primitive (such as a signature or one-way function) is hard to break with probability ε, then it is resilient to $\log(1/\varepsilon)$ bits of leakage. This means that if some counter-example shows that the leakage bound L does not amplify with k, then neither does the security. Therefore, although this observation was never made, the counterexamples to leakage amplification from [23,21] seem to already imply some counterexample for hardness. Unfortunately, both works concentrate on a modified version of parallel repetition, where some *common public parameters* are reused by all of the instances and, thus, they are *not truly independent.* Indeed, although showing counterexamples for (the harder question of) leakage amplification is still interesting in this scenario, constructing ones for hardness amplification becomes trivial.[3] However, the work of [21] also proposed that a variant of their counterexample for leakage amplification may extend to the setting without common parameters under a highly non-standard assumption about computationally sound (CS) proofs. Indeed, this suggestion led us to re-examine our initial belief that such counterexamples should not exist, and eventually resulted in this work. We also notice that our counterexample for signature schemes (but not one-way functions) can be easily extended to give a counterexample for leakage amplification without common parameters.

2 Hardness Amplification Definitions and Conjectures

In this work, we will consider a non-uniform model of computation. We equate entities such as challengers and attackers with circuit families, or equivalently, Turing Machines with advice. We let n denote the *security parameter.* We say that a function $\varepsilon(n)$ is *negligible* if $\varepsilon(n) = n^{-\omega(1)}$.

We begin by defining a general notion of (single prover) cryptographic games, which captures the security of the vast majority of cryptographic primitives, such as one-way functions, signatures, etc.

Definition 1 (Games). *A* game *is defined by a probabilistic interactive* challenger $\mathcal{C}$. *On security parameter n, the challenger $\mathcal{C}(1^n)$ interacts with some attacker $\mathcal{A}(1^n)$ and may output a special symbol* win. *If this occurs, we say that $\mathcal{A}(1^n)$* wins *$\mathcal{C}(1^n)$.*

We can also define a *class* $\mathbb{C}$ of cryptographic games $\mathcal{C} \in \mathbb{C}$. For example the factoring problem fixes a particular game with the challenger $\mathcal{C}_{FACTOR}$ that chooses two random n-bit primes p, q, sends $N = p \cdot q$ to $\mathcal{A}$, and outputs win iff it gets back p, q. On the other hand, *one-way functions* can be thought of as a class of games $\mathbb{C}_{OWF}$, where each candidate one-way function f defines

[3] E.g., the hard problem could ask to break either the actual instance or the common parameter. While such an example does not necessarily contradict leakage amplification, it clearly violates hardness amplification.

a particular game $\mathcal{C}_f \in \mathbb{C}_{OWF}$. So far, this definition of games and classes of games such as one-way function is purely syntactic and we now define what it means for a game to be hard.

Definition 2 (Hardness). *We say that the game $\mathcal{C}$ is $(s(n), \varepsilon(n))$-hard if, for all sufficiently large $n \in \mathbb{N}$ and all $\mathcal{A}(1^n)$ of size $s(n)$, we have*

$$\Pr[\mathcal{A}(1^n) \text{ wins } \mathcal{C}(1^n)] < \varepsilon(n).$$

We say that the game $\mathcal{C}$ is $(\mathsf{poly}, \varepsilon(n))$-hard if it is $(s(n), \varepsilon(n))$-hard for all polynomial $s(n)$. We say that the game $\mathcal{C}$ is $(\mathsf{poly}, \mathsf{negl})$-hard if it is $(s(n), 1/p(n))$-hard for all polynomials $s(n)$, $p(n)$.

Definition 3 (Direct Product). *For a cryptographic game $\mathcal{C}$ we define the* k-wise direct-product game *$\mathcal{C}^k$, which initializes k independent copies of $\mathcal{C}$ and outputs the* win *symbol if and only if all k copies individually output* win.

Finally, we are ready to formally define what we mean by hardness amplification. Since we focus on negative results, we will distinguish between several broad levels of hardness amplification and ignore exact parameters. For example, we do not pay attention to the number of repetitions k needed to reach a certain level of hardness (an important parameter for positive results), but are more concerned with which levels of hardness are or are not reachable altogether.

Definition 4 (Hardness Amplification). *For a fixed game $\mathcal{C}$, we say that* hardness amplifies to *$\varepsilon = \varepsilon(n)$ if there exists some polynomial $k = k(n)$ such that $\mathcal{C}^k$ is $(\mathsf{poly}, \varepsilon)$-hard. We say that hardness amplifies* to negligible *if there exists some polynomial $k = k(n)$ such that $\mathcal{C}^k$ is $(\mathsf{poly}, \mathsf{negl})$-hard. For a* class $\mathbb{C}$ *of games, we say that:*

1. *The hardness of a class $\mathbb{C}$ amplifies* to negligible *if, for every game $\mathcal{C} \in \mathbb{C}$ which is $(\mathsf{poly}, \frac{1}{2})$-hard, the hardness of $\mathcal{C}$ amplifies to negligible.*
2. *The hardness of a class $\mathbb{C}$ amplifies to $\varepsilon(n)$ if, for every game $\mathcal{C} \in \mathbb{C}$ which is $(\mathsf{poly}, \frac{1}{2})$-hard, the hardness of $\mathcal{C}$ amplifies to $\varepsilon(n)$.*
3. *The hardness of a class $\mathbb{C}$ amplifies* beyond negligible *if there exists some global negligible function $\varepsilon(n)$ for the entire class, such that the hardness of $\mathbb{C}$ amplifies to $\varepsilon(n)$.*

Remarks on Definition. The standard "direct product theorems" for classes such as one-way functions/relations and signatures show that the hardness of the corresponding class amplifies *to negligible* (bullet 1). For example, if we take *any* $(\mathsf{poly}, 1/2)$-hard function f, then a sufficiently large direct product f^k will be $(\mathsf{poly}, \mathsf{negl})$-hard.[4] However, what "negligible" security can we actually get? The result does not say and it may depend on the function f that we start with.[5] One

[4] The choice of 1/2 is arbitrary and can be replaced with any constant or even any function bounded-away-from 1. We stick with 1/2 for concreteness and simplicity.

[5] It also seemingly depends on the exact polynomial size $s(n)$ of the attackers we are trying to protect against. However, using a result of Bellare [4], the dependence on $s(n)$ can always be removed.

could conjecture that there is *some* fixed negligible $\varepsilon(n)$ such that a sufficiently large direct product of *any* weak instantiation will amplify its hardness to $\varepsilon(n)$. This is amplification *beyond negligible* (bullet 3). More ambitiously, we could expect that this negligible $\varepsilon(n)$ is very small such as $\varepsilon(n) = 2^{-n^{\Omega(1)}}$ or even $2^{-\Omega(n)}$. We explicitly state these conjectures below.

Dream Conjecture (Weaker): For any class of cryptographic games $\mathbb{C}$ for which hardness amplifies *to negligible*, it also amplifies *beyond negligible.*

Dream Conjecture (Stronger): For any class of cryptographic games $\mathbb{C}$ for which hardness amplifies *to negligible*, it also amplifies to $\varepsilon(n) = 2^{-n^{\Omega(1)}}$.

Our work gives counterexamples to both conjectures. We give two very different types counterexamples: one for the classes of *signature schemes* (Section 3) and one for the class of *one-way functions* (Section 4). Our counterexamples naturally require that some hard instantiations of these primitives exist to begin with, and our counterexamples for the weaker versions of the dream conjecture will actually require the existence of *exponentially hard* versions of these primitives. In particular, under strong enough assumptions, we will show that for *every* negligible function $\varepsilon(n)$ there is stand-alone scheme which is *already* $(\mathsf{poly}, \mathsf{negl})$-hard, but whose k-wise direct product is *not* $(\mathsf{poly}, \varepsilon(n))$-hard, no matter how large k is.

2.1 Hard and One-Way Relations

As a component of both counterexamples, we will rely on the following definition of hard relations phrased in the framework of cryptographic games:

Definition 5 (Hard Relations). *Let $R \subseteq \bigcup_{n \in \mathbb{N}} \{0,1\}^n \times \{0,1\}^{p(n)}$ be an NP relation consisting of pairs (y, w) with* instances y *and* wintesses w *of polynomial size $p(|y|)$. Let $L = \{\, y \;:\; \exists\, w \text{ s.t. } (y, w) \in R\}$ be the corresponding NP language. Let $y \leftarrow \textsc{SamL}(1^n)$ be a PPT algorithm that samples values $y \in L$. For a relation $\mathcal{R} = (R, \textsc{SamL})$, we define the corresponding security game where the challenger $\mathcal{C}(1^n)$ samples $y \leftarrow \textsc{SamL}(1^n)$ and the adversary wins if it outputs w s.t. $(y, w) \in R$. By default, we consider $(\mathsf{poly}, \mathsf{negl})$-hard relations, but we can also talk about $(s(n), \varepsilon(n))$-hard relations.*

Note that, for hard relations, we only require that there is an efficient algorithm for sampling hard instances y. Often in cryptography we care about a sub-class of hard relations, which we call *one-way relations*, where is it also feasible to efficiently sample a hard instance y *along with* a witness w. We define this below.

Definition 6 (One-Way Relation). *Let R be an NP relation and L be the corresponding language. Let $(y, w) \leftarrow \textsc{SamR}(1^n)$ be a PPT algorithm that samples values $(y, w) \in R$, and define $y \leftarrow \textsc{SamL}(1^n)$ to be a restriction of* $\textsc{SamR}$ *to its first output. We say that $(R, \textsc{SamR})$ is a* one-way relation *if $(R, \textsc{SamL})$ is a hard relation.*

3 Counterexample for Signature Schemes

3.1 Overview

The work of [11] shows that the direct product of any *stateless* signature scheme amplifies hardness *to negligible.* We now show that it does not (in general) amplify hardness beyond negligible. In fact, we will give a transformation from any standard signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ into a new signature scheme $(\textsc{Gen}, \textsc{Sign}, \textsc{Verify})$ whose hardness does not amplify (via a direct product) beyond negligible. We start by giving an informal description of the transformation to illustrate our main ideas. In order to convey the intuition clearly, we will first consider a simplified case where the signing algorithm $\textsc{Sign}$ of the (modified) scheme $(\textsc{Gen}, \textsc{Sign}, \textsc{Verify})$ is *stateful,* and will then discuss how to convert the stateful signing algorithm into one that is *stateless.*[6]

Embedding MPC in Signatures. Let $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be any standard signature scheme. Let $\mathcal{F} = \{\mathcal{F}_k\}_{k\in\mathbb{N}}$ be a randomized k-party "ideal functionality" that takes no inputs and generates a random instance y of a hard relation $\mathcal{R} = (R, \textsc{SamL})$ according to the distribution $\textsc{SamL}$. Further, let $\Pi = \{\Pi_k\}_{k\in\mathbb{N}}$ be a multi-party computation protocol that securely realizes the functionality $\mathcal{F}$ for any number of parties k. Then, the new signature scheme $(\textsc{Gen}, \textsc{Sign}, \textsc{Verify})$ works as follows.

Algorithms $\textsc{Gen}$ and $\textsc{Verify}$ are identical to Gen and Verify respectively. The signing algorithm $\textsc{Sign}$ is essentially the same as Sign, except that, on receiving a signing queries of a "special form", $\textsc{Sign}$ interprets these as "protocol messages" for Π_k and (in addition to generating a signature of them under $\textsc{Sign}$) also executes the *next message function* of the protocol and outputs its response as part of the new signature. A special initialization query specifies the number of parties k involved in the protocol and the role P_i in which the signing algorithm should act. The signing algorithm then always acts as the honest party P_i while the user submitting signing queries can essentially play the role of the remaining $k - 1$ parties. When Π_k is completed yielding some output y (interpreted as the instance of a hard relation $\mathcal{R}$) the signing algorithm $\textsc{Sign}$ will look for a signing query that contains a corresponding witness w, and, if it receives one, will respond to it by simply outputting its entire *secret key* in the signature. The security of the transformed signature $(\textsc{Gen}, \textsc{Sign}, \textsc{Verify})$ immediately follows from the security of the MPC protocol Π_k against all-but-one corruptions, the hardness of the relation $\mathcal{R}$ and the security of the original signature scheme.

Attacking the Direct-Product. Let us briefly demonstrate an adversary $\mathcal{A}$ for the k-wise direct product. Very roughly, $\mathcal{A}$ carefully chooses his signing queries

[6] We note that in the setting of stateful signatures, hardness fails to amplify even *to* negligible since we can embed the counterexamples of [5,29] into the signature scheme. Nevertheless our initial description of our counterexample for the stateful setting will clarify the main new result, which is a counterexample for the stateless setting.

so as to force $\textsc{Sign}_1, \ldots, \textsc{Sign}_k$ to engage in a *single* execution of the protocol Π_k, where each $\textsc{Sign}_i$ plays the role of a different party P_i, while $\mathcal{A}$ simply acts as the "communication link" between them. This results in all component schemes $\textsc{Sign}_i$ generating a common instance y of the hard relation. Finally, $\mathcal{A}$ simply "guesses" a witness w for y at random and, if it succeeds, submits w as a signing query, thereby learns the secret key of *each* component signature scheme thereby breaking all k of them! Note that the probability of guessing w is bounded by some negligible function in n and is *independent* of the number of parallel repetitions k.

Stateful to Stateless. While the above gives us a counterexample for the case where $\textsc{Sign}$ is a *stateful* algorithm, (as stated above) we are mainly interested in the (standard) case where $\textsc{Sign}$ is *stateless.* In order to make $\textsc{Sign}$ a stateless algorithm, we can consider a natural approach where we use a modified version Π'_k of protocol Π_k: each party P_i in Π'_k computes an outgoing message in essentially the same manner as in Π_k, except that it also attaches an *authenticated encryption* of its current protocol state, as well as the previous protocol message. This allows each (stateless) party P_i to "recover" its state from the previous round to compute its protocol message in the next round. Unfortunately, this approach is insufficient, and in fact insecure, since an adversarial user can *reset* the (stateless) signing algorithm at any point and achieve the effect of *rewinding* the honest party (played by the signing algorithm) during the protocol Π_k. To overcome this problem, we leverage techniques from the notion of resettably-secure computation. Specifically, instead of using a standard MPC protocol in the above construction, we use a recent result of Goyal and Maji [15] which constructs an MPC protocol that is secure against reset attacks and works for stateless parties for a large class of functionalities, including "inputless" randomized functionalities (that we will use in this paper).

The above intuitive description hides many details of how the user can actually "drive" the MPC execution between the k signers within the direct-product game where all signers respond to a single common message. We proceed to make this formal in the following section.

3.2 Our Signature Scheme

We now give our transformation from any standard signature scheme into one whose hardness does not amplify beyond negligible. We first establish some notation.

Notation. Let n be the security parameter. Let $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be any standard signature scheme. Further, let $(\mathcal{R}, \textsc{SamL})$ be a hard relation as per Definition 5. Let $\{PRF_K : \{0,1\}^{\mathsf{poly}(n)} \rightarrow \{0,1\}^{\mathsf{poly}(n)}\}_{K \in \{0,1\}^n}\}$ be a pseudo-random function family.

Stateless MPC. We consider a randomized k-party functionality $\mathcal{F} = \{\mathcal{F}\}_{k \in \mathbb{N}}$ that does not take any inputs; $\mathcal{F}$ simply samples a random pair $y \leftarrow \textsc{SamL}(1^n)$

and outputs y to all parties. Let $\{\Pi_k\}_{k\in\mathsf{poly}(n)}$ be a family of protocols, where each $\Pi_k = \{P_1, \ldots, P_k\}$ is a k-party MPC protocol for computing the functionality $\mathcal{F}$ in the *public state model*. This model is described formally in the full version, and we only give a quick overview here. Each party P_i is completely described by the next message function NM_i, which takes the following four values as input: (a) a string π_{j-1} that consists of all the messages sent in any round $j-1$ of the protocol, (b) the *public state* state_i of party P_i, and (c) the secret randomness r_i. On receiving an input of the form $\pi_{j-1}\|\mathsf{state}_i\|r_i$, NM_i outputs P_i's message in round j along with the updated value of state_i. We assume that an attacker corrupts (exactly) $k-1$ of the parties. In the real-world execution, the attacker can arbitrarily call the next-message function NM_i of the honest party P_i with arbitrarily chosen values of the public state state_i and arbitrary message π_{j-1} (but with an honestly chosen and secret randomness r_i). Nevertheless, the final output of P_i and the view of the attacker can be simulated in the ideal world where the simulator can "reset" the ideal functionality. In our case, that means that the attacker can adaptively choose one of polynomially many honestly chosen instances $y_1, \ldots, y_q$ of the hard relation which P_i will then accept as output.

The Construction. We describe our signature scheme $(\textsc{Gen}, \textsc{Sign}, \textsc{Verify})$.

$\textsc{Gen}(1^n)$: Compute $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$. Also, sample a random tape $K \leftarrow \{0,1\}^{\mathsf{poly}(n)}$ and a random identity $\mathsf{id} \in \{0,1\}^n$. Output $PK = (pk, \mathsf{id})$ and $SK = (sk, K, \mathsf{id})$.

$\textsc{Sign}(SK, m)$: To sign a message m using secret key $SK = (sk, K, \mathsf{id})$, the signer outputs a signature $\sigma = (\sigma^1, \sigma^2)$ where $\sigma^1 \leftarrow \mathsf{Sign}(sk, m)$. Next, if m does not contain the prefix "`prot`", then simply set $\sigma^2 = \{0\}$. Otherwise, parse $m = (\text{"}\mathtt{prot}\text{"}\|\mathsf{IM}\|\pi_j\|\mathsf{state}\|w)$, where $\mathsf{IM} = k\|\mathsf{id}_1\|\ldots\|\mathsf{id}_k$ such that $\mathsf{state} = \mathsf{state}_1\|\ldots\|\mathsf{state}_k$, then do the following:

- Let $i \in [k]$ be such that $\mathsf{id} = \mathsf{id}_i$. Compute $r_i = PRF_K(\mathsf{IM})$. Then, apply the next message function NM_i of (stateless) party P_i in protocol Π_k over the string $\pi_j\|\mathsf{state}_i\|r_i$ and set σ^2 to the output value.[7]
- Now, if σ^2 contains the output y of protocol Π_k,[8] then further check whether $(y, w) \in \mathcal{R}$. If the check succeeds, set $\sigma^2 = SK$.

$\textsc{Verify}(PK, m, \sigma)$: Given a signature $\sigma = (\sigma^1, \sigma^2)$ on message m with respect to the public key $PK = (pk, \mathsf{id})$, output 1 iff $\textsc{Verify}(pk, m, \sigma^1) = 1$.

[7] Note that here σ^2 consists of party P_i's protocol message in round $j+1$, and its updated public state state_i.

[8] Note that this is the case when j is the final round in Π_k. Here we use the property that the last round of Π_k is the *output delivery round*, and that when NM_i is computed over the protocol messages of this round, it outputs the protocol output.

This completes the description of our signature scheme. In the full version, we prove the following theorem showing that the signature scheme satisfies basic signature security.

Theorem 1. *If* (Gen, Sign, Verify) *is a secure signature scheme,* $\{PRF_K\}$ *is a PRF family,* $\mathcal{R}$ *is a hard relation, and* Π_k *is a stateless MPC protocol for functionality* $\mathcal{F}$*, then the proposed scheme* ($\textsc{Gen}, \textsc{Sign}, \textsc{Verify}$) *is a secure signature scheme.*

3.3 Attack on the Direct Product

Theorem 2. *Let* ($\textsc{Gen}, \textsc{Sign}, \textsc{Verify}$) *be the described signature scheme and let* $\mathcal{R} = (\textsc{SamL}, R)$ *be the hard relation used in the construction. Assume that for any* $y \overset{\$}{\leftarrow} \textsc{SamL}(1^n)$ *, the size of the corresponding witness* w *is bounded by* $|w| = p(n)$*. Then, for any polynomial* $k = k(n)$*, there is an attack against the* k*-wise direct product running in time* $\mathsf{poly}(n)$ *with success probability* $\varepsilon(n) = 2^{-p(n)}$.

We will prove Theorem 2 by constructing an adversary $\mathcal{A}$ that mounts a key-recovery attack on any k-wise direct product of the signature scheme ($\textsc{Gen}$, $\textsc{Sign}$, $\textsc{Verify}$).

k-wise Direct Product. Let (**Gen**, **Sign**, **Verify**) denote the k-wise direct product of the signature scheme ($\textsc{Gen}$, $\textsc{Sign}$, $\textsc{Verify}$), described as follows. The algorithm **Gen** runs $\textsc{Gen}$ k-times to generate (PK_1, SK_1),...,(PK_k, SK_k). To sign a message m, **Sign** computes $\sigma_i \leftarrow \textsc{Sign}(SK_i, m)$ for every $i \in k$ and outputs $\sigma = (\sigma_1, \ldots, \sigma_k)$. Finally, on input a signature $\sigma = (\sigma_1, \ldots, \sigma_k)$ on message m, **Verify** outputs 1 iff $\forall i \in k$, $\textsc{Verify}(PK_i, m, \sigma_i) = 1$.

Description of $\mathcal{A}$. We now describe the adversary $\mathcal{A}$ for (**Gen**, **Sign**, **Verify**). Let $(PK_1, \ldots, PK_k)$ denote the public key that $\mathcal{A}$ receives from the challenger of the signature scheme (**Gen**, **Sign**, **Verify**), where each $PK_i = (pk_i, \mathsf{id}_i)$. The adversary $\mathcal{A}$ first sends a signing query m_0 of the form "prot"$\|\mathsf{IM}\|\pi_0\|\mathsf{state}\|w$, where $\mathsf{IM} = k\|\mathsf{id}_1\| \ldots \|\mathsf{id}_k$, and $\pi_0 = \mathsf{state} = w = \{0\}$. Let $\sigma = (\sigma_1, \ldots, \sigma_k)$ be the response it receives, where each $\sigma_i = \sigma_i^1, \sigma_i^2$. $\mathcal{A}$ now parses each σ_i^2 as a first round protocol message π_1^i from party P_i followed by the public state state_i of P_i (at the end of the first round) in protocol Π_k.

$\mathcal{A}$ now prepares a new signing query m_1 of the form "prot"$\|\mathsf{IM}\|\pi_1\|\mathsf{state}\|w$, where IM and w are the same as before, but $\pi_1 = \pi_1^1\| \ldots \|\pi_1^k$, and $\mathsf{state} = \mathsf{state}_1\| \ldots \|\mathsf{state}_k$. On receiving the response, $\mathcal{A}$ repeats the same process as above to produce signing queries $m_2, \ldots, m_{t-1}$, where t is the total number of rounds in protocol Π_k. (That is, each signing query $m_2, \ldots, m_{t-1}$ is prepared in the same manner as m_1.)

Finally, let $\sigma = (\sigma_1, \ldots, \sigma_k)$ be the response to the signing query m_{t-1}. $\mathcal{A}$ now parses each σ_i^2 as the round t protocol message π_t^i from party P_i followed by the state state_i of P_i. Now, since the final round (i.e., round t) of protocol Π_k is the *output delivery round*, and further, Π_k satisfies the *publicly computable output* property, $\mathcal{A}$ simply computes the protocol output y from the messages $\pi_t^1, \ldots, \pi_t^k$.

Now, $\mathcal{A}$ guesses a $p(n)$-sized witness $w^* \xleftarrow{\$} \{0,1\}^{p(n)}$ at random and, if $(y, w^*) \in \mathcal{R}(x)$, it now sends the final signing query $m_t =$"prot"$\|\mathsf{IM}\|\pi_t\|\mathsf{state}\|w$, where IM is the same as before, $\pi_t = \pi_t^1\|\ldots\|\pi_t^k$, $\mathsf{state} = \mathsf{state}_1\|\ldots\|\mathsf{state}_k$, and $w = w^*$. Thus, $\mathcal{A}$ obtains $SK_1, \ldots, SK_k$ from the challenger and can forge arbitrary signatures for the direct product scheme. It's clear that its success probability is at least $2^{-p(n)}$.

Corollary 1. *Assuming the existence hard relations and a general stateless MPC compilers, the hardness of signature schemes does* not *amplify to any $\varepsilon(n) = 2^{-n^{\Omega(1)}}$. This gives a counterexample to the strong dream conjecture. If we, in addition, assume the existence of $(2^{\Omega(n)}, 2^{-\Omega(n)})$-hard relations with witness size $p(n) = O(n)$, then there exist signature schemes whose hardness does not amplify* beyond negligible*. This gives a counterexample to the weak dream conjecture.*

Proof. For the first result, assume that the witness size of the relation $\mathcal{R}$ is bounded by $p(n) = O(n^c)$ for some constant c. Given any constant $\delta > 0$, we can simply instantiate the signature scheme $(\textsc{Gen}, \textsc{Sign}, \textsc{Verify})$ used in our counterexample with the hard relation $\mathcal{R}$' that uses security parameter $m(n) = n^{\delta/c}$ so that its witness size is $p'(n) = p(m) = O(n^\delta)$. It's clear that $\mathcal{R}$' is still $(\mathsf{poly}(n), \mathsf{negl}(n))$-secure but, by Theorem 2, the k-wise direct product can be broken in $\mathsf{poly}(n)$ time with probability $\varepsilon(n) = 2^{-O(n^\delta)}$. Therefore security does not amplify 2^{n^δ} for any $\delta > 0$. The second part of the theorem follows in the same way, except that, for *any fixed* negligible function $\delta(n)$ we set $m(n) = -\log(\delta(n))$.

4 Counterexample for One-Way Relations and Functions

In Section 3, we proved that there exist signature schemes whose hardness does not amplify. This already rules out the general conjecture that "for *any* game for which hardness amplifies to negligible, hardness will also amplify to exponential (or at least beyond negligible)". Nevertheless, one might still think that the conjecture hold for more restricted classes of games. Perhaps the simplest such class to consider is one-way functions. Note that, unlike the case for signature schemes, the one-wayness game does not allow interaction and has bounded communication between attacker and challenger. Thus, the general strategy we employed in Section 3 of embedding a multiparty computation inside signature queries, will no longer work. In this section, we propose an alternate strategy for showing that one-way relation hardness does not amplify beyond negligible.

4.1 Our Construction

We begin by giving a counterexample for *hard relations*. We then extend it to counterexamples for one-way relations and and one-way functions. Our constructions are based on a new (non-standard) cryptographic security assumption on hash functions. Let $h : \{0,1\}^{2n} \mapsto \{0,1\}^n$ be a hash function. We define a *Merkle path of length* ℓ to be a tuple of the form

$$p_\ell = (x_0, (b_1, x_1), \ldots, (b_\ell, x_\ell)) \quad : \quad b_i \in \{0,1\}, x_i \in \{0,1\}^n.$$

Intuitively, x_0 could be the *leaf* of some Merkle tree of height ℓ, and the values $x_1, \ldots, x_\ell$ are the siblings along the path from the leaf to the *root*, where the bits b_i indicate whether the sibling x_i is a left or right sibling. However, we can also talk about a path p_ℓ on its own, without thinking of it as part of a larger tree. Formally, if p_ℓ is a Merkle path as above, let $p_{\ell-1}$ be the path with the last component (b_ℓ, x_ℓ) removed. The *value* of a Merkle path p_ℓ as above is defined iteratively via:

$$\bar{h}(p_\ell) = \begin{cases} h(\bar{h}(p_{\ell-1}), x_\ell) & \ell > 0, b_\ell = 1 \\ h(x_\ell, \bar{h}(p_{\ell-1})) & \ell > 0, b_\ell = 0 \\ x_0 & \ell = 0 \end{cases}$$

We call x_0 the *leaf* of the path p_ℓ, and $z = \bar{h}(p_\ell)$ is its *root*. We say that $y = (x_L, x_R) \in \{0,1\}^{2n}$ is the *known preimage of the path* p_ℓ if x_L, x_R are the values under the root, so that either $x_L = x_\ell$, $x_R = \bar{h}(p_{\ell-1})$ if $b_\ell = 0$, or $x_L = \bar{h}(p_{\ell-1}), x_R = x_\ell$ if $b_\ell = 1$. Note that this implies $h(y) = \bar{h}(p_\ell)$. We say that $y' \in \{0,1\}^{2n}$ is a *second preimage of the path* p_ℓ if $y' \neq y$ is *not* the known preimage of p_ℓ, and $h(y') = \bar{h}(p_\ell)$. We are now ready to define the *extended second-preimage resistance (ESPR)* assumption. This assumption says that, given a random challenge $x_0 \in \{0,1\}^n$, it is hard to find a (short) path p_ℓ containing x_0 as a leaf, and a second-preimage y' of p_ℓ.

Definition 7 (ESPR). *Let* $h : \{\{0,1\}^{2n} \mapsto \{0,1\}^n\}_{n\in\mathbb{N}}$ *be a poly-time computable hash function. We define the* Extended Second Preimage Resistance *(ESPR) assumption on* h *via the following security game between a challenger and an adversary* $\mathcal{A}(1^n)$*:*

1. *The challenger chooses* $x_0 \xleftarrow{\$} \{0,1\}^n$ *at random and gives it to* $\mathcal{A}$.
2. $\mathcal{A}$ *wins if it outputs a tuple* (p_ℓ, y')*, where* p_ℓ *is a Merkle path of length* $\ell \leq n$ *containing* x_0 *as a leaf, and* y' *is a second-preimage of* p_ℓ.

Discussion. In the above definition, we want h to be a single *fixed* hash function and not a function family. The notion of ESPR security seems to lie somewhere in between second-preimage resistance (SPR) and collision resistance (CR), implying the former and being implied by the latter.[9] Unfortunately, collision resistance *cannot* be achieved by any fixed hash function (at least w.r.t non-uniform attackers), since the attacker can always know a single hard-coded collision as auxiliary input. Fortunately, there does not appear to be any such trivial non-uniform attack against ESPR security, since the attacker is forced to "incorporate" a random leaf x_0 into the Merkle path on which it finds a collision. Therefore, in this regard, it seems that ESPR security may be closer to SPR

[9] A hash function is SPR if, given a uniformly random y, it's hard to find any $y' \neq y$ such that $h(y) = h(y')$. It is CR if it is hard to find any $y \neq y'$ s.t. $h(y) = h(y')$.

security, which can be achieved by a fixed hash function (if one-way functions exist). Indeed, in Section 4.2, we give a heuristic argument that modern (fixed) cryptographic hash functions already satisfy the ESPR property, even against non-uniform attackers. We do so by analyzing ESPR security in a variant of the random-oracle model, where the attacker may observe some "oracle-dependent auxiliary input". This model, proposed by Unruh [31], is intended to capture the properties of hash functions that can be achieved by fixed hash functions, rather than function families.

A Hard Relation from ESPR. Given a hash function h we can define the NP relation R_h with *statements* $x \in \{0,1\}^n$ and *witnesses* $w = (p_\ell, y')$ where p_ℓ is a Merkle path of length $\ell \leq n$ containing x as leaf, and y' is a second-preimage of p_ℓ. The corresponding NP language is defined as $L_h \overset{\text{def}}{=} \{x \ : \ \exists\ w \text{ s.t. } (x,w) \in R_h \ \}$. We say that h is *slightly regular*, if for every $z \in \{0,1\}^n$ there exist at least two distinct pre-images $y \neq y'$ such that $h(y) = h(y') = z$. If this is the case, then $L_h = \{0,1\}^*$ is just the language consisting of all bit strings. Now, we can define the distribution $x \leftarrow \textsc{SamL}(1^n)$ which just samples $x \overset{\$}{\leftarrow} \{0,1\}^n$ uniformly at random. It is easy to see that, if h is an $(s(n), \varepsilon(n))$-hard ESPR hash function, then $\mathcal{R}_h = (R_h, \textsc{SamL})$ is an $(s(n), \varepsilon(n))$-*hard relation.*

Hardness Non-Amplification. We now show our counterexample to the hardness amplification for the hard relation $\mathcal{R}_h$. The main idea is that, given k random and independent challenges $x^{(1)}, \ldots, x^{(k)}$, the attacker builds a Merkle tree with the challenges as leaves. Let z be the value at the top of the Merkle tree. Then the attack just guesses some value $y' \in \{0,1\}^{2n}$ at random and, with probability $\geq 2^{-2n}$, y' will be a second-preimage of z (i.e. $h(y') = z$ and y' is distinct from the known preimage y containing the values under the root). Now, for each leaf $x^{(i)}$, let p_ℓ^i be the Merkle path for the leaf $x^{(i)}$. Then the witness $w_i = (y', p_\ell^i)$ is good witness for $x^{(i)}$. So, with probability $\geq 2^{-2n}$ with which the attack correctly guessed y', it breaks *all* k independent instances of the relation $\mathcal{R}_h$, no matter how large k is! By changing the relation $\mathcal{R}_h = (R_h, \textsc{SamL})$ so that, on security parameter n, the sampling algorithm $\textsc{SamL}(1^n)$ chooses $x \overset{\$}{\leftarrow} \{0,1\}^m$ with $m = m(n)$ being some smaller function of n such as $m(n) = n^\delta$ for a constant $\delta > 0$ or even $m(n) = \log^2(n)$, we can get more dramatic counterexamples where hardness does not amplify beyond $\varepsilon(n) = 2^{-n^\delta}$ or even $\varepsilon(n) = n^{-\log n}$. We now summarize the above discussion with a formal theorem.

Theorem 3. *Let h be a slightly regular, ESPR-secure hash function and let $\mathcal{R}_h = (R_h, \textsc{SamL})$ be the corresponding $(\mathsf{poly}, \mathsf{negl})$-hard relation. Then, for any polynomial $k = \mathsf{poly}(n)$, the k-wise direct product of $\mathcal{R}_h$ is* not $(\mathsf{poly}, 2^{-2n})$ *secure. That is, for any polynomial k, there is a poly-time attack against the k-wise direct product of $\mathcal{R}_h$ having success probability 2^{-2n}.*

Proof. We first describe the attack. The attacker gets k independently generated challenges $x^{(1)}, \ldots, x^{(k)}$. Let ℓ be the unique value such that $2^{\ell-1} < k \leq 2^\ell$, and let $k^* = 2^\ell$ be the smallest power-of-2 which is larger than k. Let us define additional "dummy values" $x^{(k+1)} = \ldots = x^{(k^*)} := 0^n$. The attack constructs a *Merkle Tree*, which is a full binary tree of height ℓ, whose k^* leaves are associated with the values $x^{(1)}, \ldots, x^{(k^*)}$. The value of any non-leaf node v is defined recursively as $val(v) = h(val(v_L), val(v_R))$ where v_L, v_R are the left and right children of v respectively. For any leaf $v^{(i)}$ associated with the value $x^{(i)}$, let $(v_1 = v^{(i)}, v_2, \ldots, v_\ell, r)$ be the nodes on the path from the leaf v_1 to the root r in the Merkle tree. The Merkle path associated with the value $x^{(i)}$ is then defined by $p_\ell^{(i)} = (x^{(i)}, (x_1, b_1), \ldots, (x_\ell, b_\ell))$ where each x_j is the value associated with the *sibling* of v_j, and $b_j = 0$ if v_j is a right child and 1 otherwise. Note that, if r is the root of the tree and $z = val(r)$ is the value associated with it, then $\bar{h}(p_\ell^{(i)}) = z$ for all paths $p_\ell^{(i)}$ with $i \in \{1, \ldots, k^*\}$. Furthermore let us label the nodes v_L, v_R to be the children of the root r, the values x_L, x_R be the values associated with them, and set $y := (x_L, x_R)$. Then y is the known preimage such that $h(y) = z$, associated with each one of the paths $p_\ell^{(i)}$.

The attack guesses a value $y' \xleftarrow{\$} \{0,1\}^{2n}$ at random and, outputs the k-tuple of witnesses $(w_1, \ldots, w_k)$ where $w_i = (p_\ell^{(i)}, y')$. With probability at least 2^{-2n}, y' is a second-preimage of z with $h(y') = z$ and $y' \neq y$ (since h is slightly regular, such second preimage always exists). If this is the case, then y' is also a second preimage of *every* path $p_\ell^{(i)}$. Therefore, with probability $\geq 2^{-2n}$ the attack finds a witness for each of the k instances and wins the hard relation game for the direct product relation $\mathcal{R}_h^k$.

Corollary 2. *Assuming the existence of a slightly regular* $(\mathsf{poly}, \mathsf{negl})$*-hard ESPR hash functions, the hardness of hard relations does* not *amplify to* $2^{-n^{\Omega(1)}}$*, giving a counterexample to the stronger dream conjecture. If we instead assume the existence of* $(2^{\Omega(n)}, 2^{-\Omega(n)})$*-hard ESPR hash functions, then the hardness of hard relations does* not *amplify beyond negligible, giving a counterexample to the weaker dream conjecture.*

Proof. Let h be the ESPR hash-function. We define a modified relation $\mathcal{R}_h^m = (R_h, \textsc{SamL})$ where the sampling algorithm $\textsc{SamL}(1^n)$ samples an instance $x \xleftarrow{\$} \{0,1\}^m$ where $m = m(n)$ is some function of n. For the first part of the corollary, let $\delta > 0$ be any constant, and set $m(n) = n^\delta/2$. Then $\mathcal{R}_h^m$ is still a $(\mathsf{poly}, \mathsf{negl})$-hard relation. However, by appplying Theorem 3 with m replacing n, we see that for any $k = \mathsf{poly}(m) = \mathsf{poly}(n)$, there is an attack against the k-wise direct product which succeeds with probability $\geq 2^{-2m} = 2^{-n^\delta}$. In other word, for any $\delta > 0$, there is a $(\mathsf{poly}, \mathsf{negl})$-hard relation whose direct product is *not* $(\mathsf{poly}, 2^{-n^\delta})$-hard, no matter how large k is. This proces the first part of the corollary. The second part of the corollary works the same way as the first part but, for *any fixed* negligible function $\delta(n)$ we set $m(n) = -\frac{1}{2}\log(\delta(n))$. Assuming that h is a $(2^{\Omega(n)}, 2^{-\Omega(n)})$-hard ESPR hash function, the relation $\mathcal{R}_h^m$ is then still $(\mathsf{poly}, \mathsf{negl})$-hard, but it's direct product is *not* $(\mathsf{poly}, \delta(n))$-hard. This proves the second part of the corollary.

Extension to One-Way Relations. We can get essentially the same results as above for *one-way* relations rather than just *hard* relations. Assume that $\mathcal{R}_{ow} = (R_{ow}, \textsc{SamR}_{ow})$ is *any one-way* relation, and $\mathcal{R}_h = (R_h, \textsc{SamL}_H)$ is the hard relation used in our counterexample. Define the OR relation $\mathcal{R}_{or} = (R_{or}, \textsc{SamR}_{or})$ via:

$$\begin{aligned} R_{or} &\overset{\text{def}}{=} \{(y_1, y_2), (w_1, w_2) \ : \ (y_1, w_1) \in R_H \text{ or } (y_2, w_2) \in R_{ow}\} \\ \textsc{SamR}_{or}(1^n) \ &: \ \text{Sample } \ y_1 \leftarrow \textsc{SamL}_h(1^n), (y_2, w_2) \leftarrow \textsc{SamR}_{ow}(1^n) \\ &\quad\ \text{Output: } ((y_1, y_2), (0, w_2)). \end{aligned}$$

Then Theorem 3 applies as-is to the one-way relation $\mathcal{R}_{or}$ replacing $\mathcal{R}_H$ and Corollary 2 applies to *one-way relations* as well.

Extension to One-Way Functions. We can also extend the above counterexample to one-way functions. Let $i(n) \geq n$ be a polynomial and $f \ : \ \{\{0,1\}^{i(n)} \rightarrow \{0,1\}^n \ \}_{n \in \mathbb{N}}$ be a *regular one-way function* so that, for $x \overset{\$}{\leftarrow} \{0,1\}^{i(n)}$, the output $f(x)$ is uniformly random over $\{0,1\}^n$. Let $\mathcal{R} = (R, \textsc{SamL})$ be the hard relation for which we have a counterexample, with witness-size bounded by $u(n)$. We define $F \ : \ (\{0,1\}^{i(n)} \times \{0,1\}^n \times \{0,1\}^{u(n)} \times \{0,1\}^n) \rightarrow \{0,1\}^n$ via:

$$F(x, y, w, z) \overset{\text{def}}{=} \begin{cases} y & \text{If } (y, w) \in R \wedge z = 0^n \\ f(x) & \text{Otherwise.} \end{cases}$$

Note that the distribution of $F(x, y, w, z)$ is statistically close to that of $f(x)$ since the probability of $z = 0^n$ is negligible. The preimage of any $y \in \{0,1\}^n$ is either of the form $(\cdot, y, w, \cdot)$ where $(y, w) \in R$ or of the form $(x, \cdot, \cdot)$ where $f(x) = y$, and hence breaking the one-wayness of F is no easier then breaking that of f or breaking the hard relation $\mathcal{R}$. On the other hand, it is possible to break the k-wise direct product of F just by breaking the k-wise direct product of the hard relation $\mathcal{R}$. Therefore, the results of Corollary 2 apply to *one-way relations* as well, if we also assume the existence of a (fixed) regular one-way function f (and an exponentially secure one for the counterexample to the weaker conjecture). In the full version of this work, we also show how to instantiate f using the ESPR function h so as to get the results of Corollary 2 for one-way functions, without needing any additional assumptions.

4.2 Justifying the ESPR Assumption

We now give some justification that ESPR hash functions may exist by showing how to construct them in in a variant of the random-oracle (RO) model. Of course, constructions in the random-oracle model do not seem to offer any meaningful guarantees for showing that the corresponding primitive may be realized by a *fixed* hash function: indeed the RO model immediately implies collision resistance which cannot be realized by a fixed hash function. Rather, the RO model is usually interpreted as implying that the given primitive is likely to be realizable by a *family of hash functions.* Therefore, we will work with a variant of the RO model in which the attacker is initialized with some arbitrary "oracle-dependent auxiliary input". This model was proposed by Unruh [31] with the explicit motivation of capturing properties the can be satisfied by a fixed hash function. For example, the auxiliary input may include some small number of fixed collisions on the RO and therefore collision-resistance is unachievable in this model. By showing that ESPR security *is* achievable, we provide some justification for this assumption.

Let $\mathcal{O} : \{0,1\}^{2n} \mapsto \{0,1\}^n$ be a fixed length random oracle. Following [31], we define "oracle-dependent auxiliary input" of size $p(n)$ as an arbitrary function $z : \{\{0,1\}^{2n} \mapsto \{0,1\}^n\} \mapsto \{0,1\}^{p(n)}$ which can arbitrarily "compresses" the entire oracle $\mathcal{O}$ into $p(n)$ bits of auxiliary information $z(\mathcal{O})$. When considering security games in the oracle-dependent auxiliary input model, we consider attackers $\mathcal{A}^{\mathcal{O}}(z(\mathcal{O}))$ which are initialized with polynomial-sized oracle-dependent auxiliary input $z(\cdot)$. In the full version, we show that the ESPR security game is hard in the random oracle model with auxiliary input.

Theorem 4. *Let $\mathcal{O}$ be modeled as a random oracle, and consider the ESPR game in which h is replaced with $\mathcal{O}$. Then, for any attacker $\mathcal{A}^{\mathcal{O}}(z(\mathcal{O}))$ with polynomial-sized auxiliary input $z(\cdot)$ and making at most polynomially many queries to $\mathcal{O}$, its probability of winning the ESPR game is at most $\varepsilon = 2^{-\Omega(n)}$.*

References

1. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-Key Encryption in the Bounded-Retrieval Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010)
2. Alwen, J., Dodis, Y., Wichs, D.: Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)
3. Barak, B., Goldreich, O., Goldwasser, S., Lindell, Y.: Resettably-sound zero-knowledge and its applications. In: FOCS, pp. 116–125 (2001)
4. Bellare, M.: A note on negligible functions. J. Cryptology 15(4), 271–284 (2002)
5. Bellare, M., Impagliazzo, R., Naor, M.: Does parallel repetition lower the error in computationally sound protocols? In: FOCS, pp. 374–383 (1997)
6. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: STOC, pp. 235–244 (2000)
7. Canetti, R., Halevi, S., Steiner, M.: Hardness Amplification of Weakly Verifiable Puzzles. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 17–33. Springer, Heidelberg (2005)
8. Canetti, R., Rivest, R., Sudan, M., Trevisan, L., Vadhan, S.P., Wee, H.: Amplifying Collision Resistance: A Complexity-Theoretic Treatment. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 264–283. Springer, Heidelberg (2007)
9. Chung, K.-M., Liu, F.-H., Lu, C.-J., Yang, B.-Y.: Efficient String-Commitment from Weak Bit-Commitment. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 268–282. Springer, Heidelberg (2010)
10. Deng, Y., Goyal, V., Sahai, A.: Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In: FOCS, pp. 251–260 (2009)
11. Dodis, Y., Impagliazzo, R., Jaiswal, R., Kabanets, V.: Security Amplification for *Interactive* Cryptographic Primitives. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 128–145. Springer, Heidelberg (2009)

12. Dwork, C., Naor, M., Reingold, O.: Immunizing Encryption Schemes from Decryption Errors. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 342–360. Springer, Heidelberg (2004)
13. Goldreich, O.: Foundations of Cryptography: Basic Tools. Cambridge University Press (2001)
14. Goldreich, O., Nisan, N., Wigderson, A.: On Yao's XOR-Lemma. Electronic Colloquium on Computational Complexity (ECCC) 2(50) (1995)
15. Goyal, V., Maji, H.K.: Stateless cryptographic protocols. In: FOCS (2011)
16. Goyal, V., Sahai, A.: Resettably Secure Computation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 54–71. Springer, Heidelberg (2009)
17. Haitner, I.: A parallel repetition theorem for any interactive argument. In: FOCS, pp. 241–250 (2009)
18. Halevi, S., Rabin, T.: Degradation and Amplification of Computational Hardness. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 626–643. Springer, Heidelberg (2008)
19. Håstad, J., Pass, R., Wikström, D., Pietrzak, K.: An Efficient Parallel Repetition Theorem. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 1–18. Springer, Heidelberg (2010)
20. Impagliazzo, R., Jaiswal, R., Kabanets, V.: Chernoff-Type Direct Product Theorems. Journal of Cryptology (September 2008); In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 500–516. Springer, Heidelberg (2007)
21. Jain, A., Pietrzak, K.: Parallel Repetition for Leakage Resilience Amplification Revisited. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 58–69. Springer, Heidelberg (2011)
22. Jutla, C.S.: Almost Optimal Bounds for Direct Product Threshold Theorem. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 37–51. Springer, Heidelberg (2010)
23. Lewko, A., Waters, B.: On the insecurity of parallel repetition for leakage resilience. In: FOCS, pp. 521–530 (2010)
24. Luby, M., Rackoff, C.: Pseudo-random permutation generators and cryptographic composition. In: STOC, pp. 356–363 (1986)
25. Maurer, U., Tessaro, S.: Computational Indistinguishability Amplification: Tight Product Theorems for System Composition. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 355–373. Springer, Heidelberg (2009)
26. Maurer, U., Tessaro, S.: A Hardcore Lemma for Computational Indistinguishability: Security Amplification for Arbitrarily Weak PRGs with Optimal Stretch. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 237–254. Springer, Heidelberg (2010)
27. Naor, M., Reingold, O.: On the construction of pseudo-random permutations: Luby-Rackoff revisited. J. of Cryptology 12, 29–66 (1999); Preliminary version in: Proc. STOC 1997
28. Pass, R., Venkitasubramaniam, M.: An efficient parallel repetition theorem for Arthur-Merlin games. In: STOC, pp. 420–429 (2007)
29. Pietrzak, K., Wikström, D.: Parallel Repetition of Computationally Sound Protocols Revisited. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 86–102. Springer, Heidelberg (2007)
30. Tessaro, S.: Security Amplification for the Cascade of Arbitrarily Weak PRPs: Tight Bounds via the Interactive Hardcore Lemma. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 37–54. Springer, Heidelberg (2011)
31. Unruh, D.: Random Oracles and Auxiliary Input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (2007)
32. Yao, A.C.-C.: Theory and applications of trapdoor functions (extended abstract). In: FOCS, pp. 80–91 (1982)