Parallel Repetition of $(k_1, \ldots, k_\mu)$-Special-Sound Multi-Round Interactive Proofs

Thomas Attema[1,2,3,*] and Serge Fehr[1,2,**]

[1] CWI, Cryptology Group, Amsterdam, The Netherlands
[2] Leiden University, Mathematical Institute, Leiden, The Netherlands
[3] TNO, Cyber Security and Robustness, The Hague, The Netherlands

Abstract. In many occasions, the knowledge error κ of an interactive proof is not small enough, and thus needs to be reduced. This can be done generically by repeating the interactive proof in parallel. While there have been many works studying the effect of parallel repetition on the *soundness error* of interactive proofs and arguments, the effect of parallel repetition on the *knowledge error* has largely remained unstudied. Only recently it was shown that the t-fold parallel repetition of *any* interactive protocol reduces the knowledge error from κ down to $\kappa^t + \nu$ for any non-negligible term ν. This generic result is suboptimal in that it does not give the knowledge error κ^t that one would expect for typical protocols, and, worse, the knowledge error remains non-negligible.

In this work we show that indeed the t-fold parallel repetition of any $(k_1, \ldots, k_\mu)$-special-sound multi-round public-coin interactive proof optimally reduces the knowledge error from κ down to κ^t. At the core of our results is an alternative, in some sense more fine-grained, measure of quality of a dishonest prover than its success probability, for which we show that it characterizes when knowledge extraction is possible. This new measure then turns out to be very convenient when it comes to analyzing the parallel repetition of such interactive proofs.

While parallel repetition reduces the knowledge error, it is easily seen to *increase* the *completeness error*. For this reason, we generalize our result to the case of s-out-of-t threshold parallel repetition, where the verifier accepts a claim if s-out-of-t of the parallel instances are accepting. An appropriately chosen threshold s allows both the knowledge error and completeness error to be reduced simultaneously.

Keywords: Proofs of Knowledge, Knowledge Soundness, Special-Soundness, Knowledge Extractor, Parallel Repetition, Threshold Parallel Repetition.

1 Introduction

1.1 Background

Proofs of Knowledge. Proofs of Knowledge (PoKs) are essential building blocks in many cryptographic primitives. They allow a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that it knows a (secret) string $w \in \{0,1\}^*$, called a *witness*, satisfying some public constraint. Typically a prover wishes to do this either in (honest-verifier) *zero-knowledge*, i.e., without revealing any information about the witness w beyond the veracity of the claim, or with communication costs smaller than the size of the witness w. Both these requirements prevent the prover from simply revealing the witness w.

A key property of PoKs is *knowledge soundness*. Informally, a protocol is said to be knowledge sound if a dishonest prover that does not know the secret witness can only succeed in convincing a verifier with some small probability κ called the *knowledge error*. This is formalized by requiring the existence of an efficient *extractor* so that for any dishonest prover that succeeds with probability $\epsilon > \kappa$, the extractor outputs a witness w with probability at least $\epsilon - \kappa$, up to a multiplicative polynomial loss (in the security parameter), when given black-box access to the prover [Gol01].

Typical 3-round public-coin protocols satisfy the conceptually simpler notion called *special-soundness.* A 3-round protocol is said to be special-sound if there exists an efficient algorithm that given two valid prover-verifier conversations (transcripts) (a, c, z) and (a, c', z'), with common first message a and distinct second messages (challenges) $c \neq c'$, outputs a witness w. More generally, a 3-round protocol is called k-special-sound if the algorithm requires k transcripts, instead of 2, to compute w. If k is polynomial in the size of the input x, the property k-special-soundness tightly implies the standard notion of knowledge soundness by a generic reduction, with $\kappa = (k-1)/N$, where N is the number of challenges [HL10, ACK21].

In recent years, *multi-round* PoKs have gained a lot of attention [BCC$^+$16, BBB$^+$18, MBKM19, BFS20, AC20, BLNS20, ACF21, ACK21, ACR21]. The notion of k-special-soundness, which is tailored to 3-round protocols, extends quite naturally to $(k_1, \ldots, k_\mu)$-special-soundness for $(2\mu + 1)$-round protocols (see Definition 7 for the formal definition). Many of the considered multi-round protocols satisfy this multi-round version of the special-soundness property. Surprisingly, only recently it was shown that also this generalization tightly implies knowledge soundness [ACK21].

Parallel Repetition. In certain occasions, the knowledge error κ of a "basic" PoK (and thereby the cheating probability of a dishonest prover) is not small enough, and thus needs to be reduced. This is particularly the case for lattice-based PoKs, where typically challenge sets are only of polynomial size resulting in non-negligible knowledge errors [LS18, ACX21]. Reducing the knowledge error can be done generically by repeating the PoK. Indeed, repeating a PoK t times *sequentially*, i.e., one after the other, is known to reduce the knowledge error from κ down to κ^t [Gol01]. However, this approach also increases the number of communication rounds by a factor t. This is often undesirable, and sometimes even insufficient, e.g., because the security loss of the Fiat-Shamir transformation, transforming interactive into non-interactive protocols, is oftentimes exponential in the number of rounds.

Therefore, it is much more attractive to try to reduce the knowledge error by *parallel* repetition. In the case of special-sound protocols, i.e., k-special-sound protocols with $k = 2$, such a parallel repetition is easy to analyze: the t-fold parallel repetition of a special-sound protocol with challenge space of cardinality N is again special-sound protocol, but now with a challenge space of size N^t, and so knowledge-soundness with $\kappa = 1/N^t$ follows immediately from the generic reduction. Unfortunately, this reasoning does not extend to k-special-sound protocols with $k > 2$: even though we still have that the t-fold parallel repetition of a k-special-sound protocol is k'-special-sound, but now with $k' = (k-1)^t + 1$, this large increase in the special-soundness parameter renders the extractor, obtained via the generic reduction, inefficient. More precisely, the run-time of a k'-special-sound protocol scales linearly in k', and therefore exponentially in t for $k' = (k-1)^t + 1$, unless $k = 2$. In case of multi-round protocols, it is not even clear that the t-fold parallel repetition of a $(k_1, \ldots, k_\mu)$-special-sound $(2\mu + 1)$-round protocol satisfies any meaningful notion of special-soundness.

Somewhat surprisingly, so far the only way to analyze the parallel repetition of k-special-sound protocols with $k > 2$, or of $(k_1, \ldots, k_\mu)$-special-sound multi-round protocols, is by means of suboptimal *generic* parallel-repetition results — or by considering weaker notions of knowledge soundness (see the discussion below). Concretely, based on a result from [CP15], it was recently shown that the t-fold parallel repetition of *any* PoK reduces the knowledge error from κ down to $\kappa^t + \nu$ for any non-negligible term ν [ACK21]. This generic result is suboptimal in that, when applied to a k-special-sound protocol for instance, it does not give the knowledge error κ^t that one expects (and that one should get when $k = 2$), and, worse, the knowledge error remains non-negligible.

Even though this generic parallel-repetition result was shown to be tight, in that there are protocols for which parallel repetition does not allow the knowledge error to be reduced down to a negligible function, we can well hope for a stronger result for certain classes of protocols. In particular, it is not too absurd to expect *strong* parallel repetition for k-special-sound protocols, and possibly for $(k_1, \ldots, k_\mu)$-special-sound protocols in the multi-round case. Here, as usual in the general context of parallel repetition, the term "strong" means that the figure of merit κ, here the knowledge error, drops from κ to κ^t under a t-fold parallel repetition.

Other Notions of a PoK. Due to the difficulty in proving the original definition in certain contexts, it has become quite customary to consider modified and/or relaxed notions of a PoK that make it then feasible to obtain positive or stronger results; it is then typically argued that the considered notion is still meaningful and useful (in the considered context).

For example, many works on multi-round protocols consider the weaker notion of *witness-extended emulation* rather than the standard notion of knowledge soundness [BCC+16, BBB+18]. In the context of quantum security, the knowledge extractor is typically allowed to have success probability $(\epsilon - \kappa)^c$ (up to a multiplicative polynomial loss) for an arbitrary constant c, instead of $\epsilon - \kappa$ [Unr12]. Moreover, recently, tighter security guarantees for discrete logarithm based Σ-protocols were obtained under a relaxed notion of knowledge soundness in which the knowledge extractor is not only given black-box access to the (possibly dishonest) prover $\mathcal{P}^*$, but is also given the success probability ϵ of $\mathcal{P}^*$ as input [RS21].

In our work here, we instead insist on the original standard definition of a PoK, and we aim for strong parallel repetition results nevertheless.

1.2 Contributions

In short, we show a strong parallel repetition theorem for the knowledge error of $(k_1, \ldots, k_\mu)$-special-sound $(2\mu + 1)$-round protocols, for $k_1, \ldots, k_\mu$ such that their product $K = k_1 \cdots k_\mu$ is polynomial in the size of the input statement.[4] This in particular implies strong parallel repetition for k-special-sound protocols for arbitrary polynomial k.[5] Strong parallel repetition means that if the original protocol has knowledge error κ then the t-fold parallel repetition has soundness error κ^t, which is optimal, matching the success probability of a dishonest prover that attacks each instance in the parallel repetition independently (and thus succeeds with independent probability κ in each instance).

We also consider a threshold parallel repetition, where the verifier accepts as soon as s out of the t parallel repetitions succeed, and we show also here that the knowledge error is what one would expect, matching the attack where the dishonest prover cheats in each of the t instances independently and hopes that he is successful in at least s instances.

The starting point of our (threshold) parallel repetition results is the following observation, considering (a single execution of) a k-special-sound protocol. The default measure of quality of a dishonest prover $\mathcal{P}^*$ is its success probability $\epsilon = \epsilon(\mathcal{P}^*)$. For instance, if ϵ is below the knowledge soundness κ then we cannot expect the extraction of a witness w to work in general. However, the crucial observation is that for a given dishonest prover $\mathcal{P}^*$, its success probability ϵ does actually not characterize (very well) whether extraction is possible or not. For instance, fixing $\mathcal{P}^*$'s first message, if $\mathcal{P}^*$ then answers correctly with probability ϵ (and fails to do so with probability $1 - \epsilon$) independently *for each* possible choice of the challenge (where the randomness is over P^*'s randomness used for computing the response), then extraction is still possible even when $\epsilon < \kappa$ (yet noticeable): simply try sufficiently many times for k distinct challenges, and after an expected number of k/ϵ trials, we have k correct responses to distinct challenges, from which a witness can then be computed.

At the core of our results is a novel knowledge extractor for k-special-sound protocols, whose success probability can be expressed in terms of an alternative, in some sense more fine-grained, measure of quality of $\mathcal{P}^*$. This new measure $\delta = \delta(\mathcal{P}^*)$ then turns out to be very convenient to work with when it comes to more complicated settings, like a *parallel repetition*, or a *multi-round* protocol, or, ultimately, a *parallel repetition* of a *multi-round* protocol.

More precisely, we ultimately construct a knowledge extractor for the t-fold parallel repetition of a $(k_1, \ldots, k_\mu)$-special-sound protocol. The extractor requires an expected number of at most $t \cdot 2^\mu \cdot K$ queries to $\mathcal{P}^*$ and succeeds with probability at least $(\epsilon - \kappa^t)/(2K)$, where $K = k_1 \cdots k_\mu$ and κ is the knowledge error of the basic protocol. Therefore, we prove that this t-fold parallel repetition has knowledge error κ^t.

---

[4] The $(k_1, \ldots, k_\mu)$-special-soundness property states that there exists an *efficient* algorithm (i.e., polynomial in the input size) that, on input a set of $K = k_1 \cdots k_\mu$ accepting protocol transcripts (with certain properties), outputs a secret witness. Therefore, this property is only useful when K is polynomial.

[5] For didactical reasons, we actually first treat the case of k-sound protocols, and then consider the more general case of multi-round protocols.

1.3 Related Work

Reducing the Soundness Error. A related question problem is that of reducing the (ordinary) *soundness error* of an interactive proof (or argument) by parallel repetition. It is well known that the t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$ of an interactive *proof* (i.e., not argument) $(\mathcal{P}, \mathcal{V})$ reduces the *soundness error* from σ down to σ^t [Gol98]. Namely, it is relatively easy to reduce an arbitrary prover against the t-fold parallel repetition, and which has success probability ϵ, into a *computationally unbounded* prover that successfully attacks a single instantiation with probability at least $\epsilon^{1/t}$.

The situation is trickier for interactive *arguments*, where the prover is required to be *efficient* and thus where this reduction no longer works. Various parallel repetition theorems for interactive arguments have been established [BIN97, PV07, Hai09, HPWP10, CL10, CP15]. In these theorems, there is an unavoidable trade-off between the success probability and the run-time of the prover $\mathcal{P}^*$. For instance, the reduction in [CL10] results in a prover $\mathcal{P}^*$ with success probability $\epsilon^{1/t} \cdot (1 - \xi)$ and run-time polynomial in $1/\xi$ for arbitrary $\xi > 0$,

Knowledge soundness is a strictly stronger notion than soundness, and thus the above results do not imply corresponding parallel-repetition results for the knowledge soundness of PoKs. In this light, it is then surprising that little is known about the parallel repetition of PoKs beyond the two extreme cases: the 2-special-sound protocols, for which strong parallel repetition holds via a simple argument [HL10], and general protocols, for which only very recently a weak parallel repetition was shown [ACK21], which reduces the knowledge error from κ down to $\kappa^t + \nu$ for any non-negligible term ν.

The Case $t = 1$. The starting point of our parallel repetition result is a new knowledge extractor for a single invocation of $(k_1, \ldots, k_\mu)$-special-sound protocols; we briefly compare this extractor with other knowledge extractors proposed for such protocols.

For instance, considering a different notion of knowledge soundness, [AC20] proposed an extractor for $(k_1, \ldots, k_\mu)$-special-sound protocols that has a *strict* polynomial run-time, yet a success probability that degrades exponentially in $K = k_1 \cdots k_\mu$. Thus, this notion is meaningful only when K is constant in the input size.

Full fledged and tight knowledge soundness for $(k_1, \ldots, k_\mu)$-special-sound protocols was only very recently shown in [ACK21]. In that work, the proposed extractor runs in expected polynomial time and succeeds with probability $\epsilon - \kappa$. As shown in Table 1, our extractor behaves somewhat worse in the (expected) polynomial run time, and also in the success probability when the newly introduced measure δ is bounded by $\epsilon - \kappa$; however, by exploiting the definition of δ, as we show in the technical part, we can obtain an extractor for a *parallel repetition* of the considered protocol by running the extractor individually on each instance of the parallel repetition. Thus, our extractor is well suited to show the claimed (threshold) parallel repetition results. Nevertheless, it remains an interesting problem whether our extractor can be improved to match up with the extractor from [ACK21] while still giving rise to our parallel-repetition results.

Extractor	Number of $\mathcal{P}^*$-queries Q	Success probability P
[AC20]	$Q \leq K$	$P \geq (\epsilon - \kappa)^K$
[ACK21]	$\mathbb{E}[Q] \leq K$	$P \geq \epsilon - \kappa$
This work	$\mathbb{E}[Q] \leq 2^\mu \cdot K \leq K^2$	$P \geq \frac{1}{K}\delta \geq \frac{1}{K}(\epsilon - \kappa)$

Table 1. Different knowledge extractors for $(k_1, \ldots, k_\mu)$-special-sound protocols. Here, $\epsilon = \epsilon(\mathcal{P}^*)$ denotes the success probability of the prover $\mathcal{P}^*$, N_i is size of the i-th challenge set, $\kappa = 1 - \prod_{i=1}^\mu \frac{N_i - k_i + 1}{N_i}$ is the knowledge error, and $K = k_1 \cdots k_\mu$. The refined quality measure $\delta = \delta(\mathcal{P}^*)$ will be defined in Section 3 and Section 4.

1.4 Organization of the Paper

In Section 2, we introduce notation and recall standard definitions regarding interactive proofs. In Section 3, we show the parallel repetition result for k-special-sound 3-round protocols: we first construct a new knowledge extractor for (the single execution of) a k-special-sound protocol, and then handle the parallel repetition of these protocols in a second step. In Section 4, we generalize the aforementioned results to multi-round protocols. Finally, in Section 5, we treat the s-out-of-t threshold parallel repetition of $(k_1, \dots, k_\mu)$-special-sound protocols.

2 Preliminaries

2.1 Interactive Proofs

Following standard terminology, given a binary relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$, a string $w \in \{0,1\}^*$ is called a *witness* for the *instance* $x \in \{0,1\}^*$ if $(x; w) \in R$. The set of valid witnesses for a statement x is denoted by $R(x)$, i.e., $R(x) = \{w : (x; w) \in R\}$. A statement that admits a witness is said to be a *true* or *valid* statement. The set of true statements is denoted by L_R, i.e., $L_R = \{x : \exists\, w \text{ s.t. } (x; w) \in R\}$. A binary relation is said to be an NP relation if the validity of a witness w can be verified in time polynomial in the size $|x|$ of the statement x. From now on we assume all relations to be NP relations.

An interactive proof for a relation R aims for a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that a statement x admits a witness, or even that the prover *knows* a witness $w \in R(x)$. We recall the following definitions.

Definition 1 (Interactive Proof). *An* interactive proof $(\mathcal{P}, \mathcal{V})$ *for relation R is an interactive protocol between two probabilistic machines, a prover $\mathcal{P}$ and a polynomial time verifier $\mathcal{V}$. Both $\mathcal{P}$ and $\mathcal{V}$ take as public input a statement x and, additionally, $\mathcal{P}$ takes as private input a witness $w \in R(x)$, which is denoted as $(\mathcal{P}(w), \mathcal{V})(x)$. As the output of the protocol, $\mathcal{V}$ either accepts or rejects. Accordingly, we say the corresponding transcript (i.e., the set of all messages exchanged in the protocol execution) is* accepting *or* rejecting.

An interactive proof $(\mathcal{P}, \mathcal{V})$ is *complete* if the verifier $\mathcal{V}$ accepts honest executions with a public-private input pair $(x; w) \in R$ with large probability. It is *sound* if the verifier rejects false statements $x \notin L_R$ with large probability. Originally interactive proofs were defined to be complete and sound [GMR85]. By contrast, we do not require interactive protocols to satisfy these properties by definition, but consider them as desirable additional security properties.

Definition 2 (Completeness). *An interactive proof $(\mathcal{P}, \mathcal{V})$ for relation R is* complete *with completeness error $\rho \colon \{0,1\}^* \to [0,1]$ if for every $(x; w) \in R$,*

$$\Pr((\mathcal{P}(w), \mathcal{V})(x) = \mathsf{reject}) \leq \rho(x)\,.$$

If $\rho(x) = 0$ for all x, $(\mathcal{P}, \mathcal{V})$ is said to be perfectly complete.

Definition 3 (Soundness). *An interactive proof $(\mathcal{P}, \mathcal{V})$ for relation R is* sound *with soundness error $\sigma \colon \{0,1\}^* \to [0,1]$, if for every $x \notin L_R$ and every prover $\mathcal{P}^*$,*

$$\Pr((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}) \leq \sigma(x)\,.$$

If an interactive proof is complete and sound, it "merely" allows a prover to convince a verifier that a statement x is true, i.e., $x \in L_R$. It does not necessarily convince a verifier that the prover "knows" a witness $w \in R(x)$. This stronger property is captured by the notion *knowledge soundness*. Informally, an interactive proof $(\mathcal{P}, \mathcal{V})$ is knowledge sound if any prover $\mathcal{P}^*$ with $\Pr((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept})$ large enough is able to compute a witness $w \in R(x)$.

Definition 4 (Knowledge Soundness). *An interactive proof* $(\mathcal{P}, \mathcal{V})$ *for relation R is knowledge sound with knowledge error* $\kappa \colon \{0,1\}^* \to [0,1]$ *if there exists a positive polynomial q and an algorithm $\mathcal{E}$, called a* knowledge extractor, *with the following properties: The extractor $\mathcal{E}$, given input x and rewindable oracle access to a (potentially dishonest) prover $\mathcal{P}^*$, runs in an expected number of steps that is polynomial in $|x|$ and outputs a witness $w \in R(x)$ with probability* $\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in R\big) \geq (\epsilon(x, \mathcal{P}^*) - \kappa(x))/q(|x|)$, *where* $\epsilon(x, \mathcal{P}^*) := \Pr((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept})$.

Remark 1. It is straightforward to verify that in order to satisfy Definition 4 it is sufficient to show that the required property holds for *deterministic* provers $\mathcal{P}^*$. Let $\mathcal{P}^*$ be an arbitrary randomized dishonest prover, and let $\mathcal{P}^*[r]$ be the deterministic prover obtained by fixing $\mathcal{P}^*$'s randomness to r. Then $\epsilon(x, \mathcal{P}^*) = \mathbb{E}[\epsilon(x, \mathcal{P}^*[r])]$, where $\mathbb{E}$ denotes the expectation over the random choice of r. Furthermore, if $\mathcal{E}^{\mathcal{P}^*}$ is declared to run $\mathcal{E}^{\mathcal{P}^*[r]}$ for a random choice of r then the same holds for the success probability of the extractor: $\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in R\big) = \mathbb{E}\big[\Pr\big((x; \mathcal{E}^{\mathcal{P}^*[r]}(x)) \in R\big)\big]$. It follows that in order to satisfy Definition 4 it is sufficient to show that the required property holds for *deterministic* provers $\mathcal{P}^*$. For this reason, we may assume provers to be deterministic, in particular, we will consider the prover's first message to be deterministic. This will significantly simplify our analysis.

If $\epsilon(x, \mathcal{P}^*) = \Pr((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}) > \kappa(x)$, then the success probability of the knowledge extractor of Definition 4 is positive. Hence, $\epsilon(x, \mathcal{P}^*) > \kappa(x)$ implies that x admits a witness, i.e., $x \in L_R$. It therefore follows that knowledge soundness implies soundness.

Remark 2. Sometimes a slightly weaker definition for knowledge soundness is used [BG93, Gol01, HL10]. This weaker definition decouples knowledge soundness from soundness by only requiring the extractor to run in expected polynomial time on inputs $x \in L_R$, i.e., it does not require the protocol to be sound. It can be shown that a *sound* protocol satisfying this weaker version of knowledge soundness is also knowledge sound in the stronger sense of Definition 4.

Definition 5 (Proof of Knowledge). *An interactive proof that is both complete with completeness error $\rho(\cdot)$ and knowledge sound with knowledge error $\kappa(\cdot)$ is a* Proof of Knowledge *(PoK) if there exists a polynomial q such that $1 - \rho(x) \geq \kappa(x) + 1/q(|x|)$ for all x.*

Let us consider some additional (desirable) properties of proofs of knowledge. We assume that the prover $\mathcal{P}$ sends the first and the last message in any interactive proof $(\mathcal{P}, \mathcal{V})$. If this is not the case, the interactive proof can be appended with an empty message. Hence, the number of communication rounds $2\mu + 1$ is always odd. We also say $(\mathcal{P}, \mathcal{V})$ is a $(2\mu + 1)$-round protocol. We will refer to *multi-round* protocols as a way of emphasizing that we are not restricting to 3-round protocols.

Definition 6 (Public-Coin). *An interactive proof* $(\mathcal{P}, \mathcal{V})$ *is public-coin if all of $\mathcal{V}$'s random choices are made public.*

If a protocol is public-coin, the verifier only needs to send its random choices to the prover. In this case, $\mathcal{V}$'s message are also referred to as *challenges* and the set from which $\mathcal{V}$ samples its messages uniformly at random is called the challenge set.

We recall the notion of (general) *special-soundness*. It is typically easier to prove that an interactive proof is special-sound than to prove that it is knowledge sound. Note that we require special-sound protocols to be public-coin.

Definition 7 (k-out-of-N Special-Soundness). *Let $k, N \in \mathbb{N}$. A 3-round public-coin protocol $(\mathcal{P}, \mathcal{V})$ for relation R, with challenge set of cardinality $N \geq k$, is k-out-of-N special-sound if there exists a polynomial time algorithm that, on input a statement x and k accepting transcripts $(a, c_1, z_1), \ldots (a, c_k, z_k)$ with common first message a and pairwise distinct challenges $c_1, \ldots, c_k$, outputs a witness $w \in R(x)$. We also say $(\mathcal{P}, \mathcal{V})$ is k-special-sound and, if $k = 2$, it is simply said to be special-sound.*

We refer to a 3-round public-coin interactive proof as a *Σ-protocol*. Note that often a Σ-protocol is required to be (perfectly) complete, special-sound and special honest-verifier zero-knowledge (SHVZK) by definition. However, we do not require a Σ-protocol to have these additional properties.
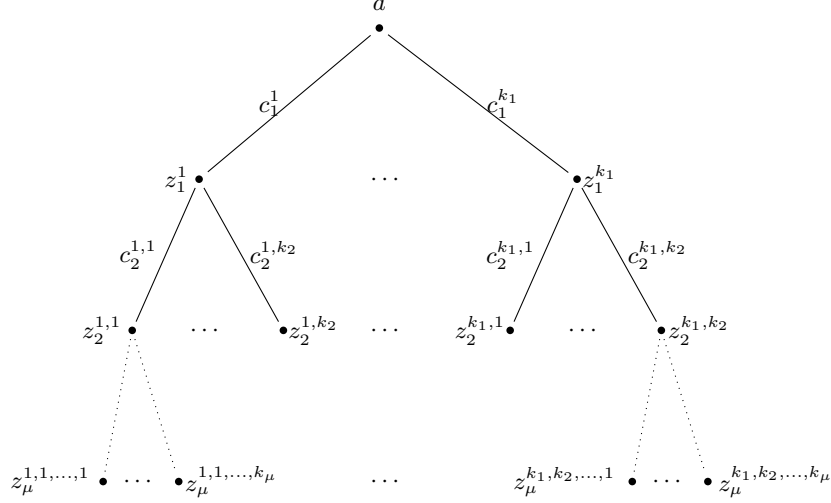
Fig. 1. $(k_1, \ldots, k_\mu)$-tree of transcripts of a $(2\mu + 1)$-round public-coin protocol [ACK21].

Definition 8 (Σ-Protocol). *A Σ-protocol is a 3-round public-coin interactive proof.*

In order to generalize k-special-soundness to multi-round protocols we introduce the notion of a tree of transcripts. We follow the definition of [ACK21].

Definition 9 (Tree of Transcripts). *Let $k_1, \ldots, k_\mu \in \mathbb{N}$. A $(k_1, \ldots, k_\mu)$-tree of transcripts for a $(2\mu+1)$-round public-coin protocol $(\mathcal{P}, \mathcal{V})$ is a set of $K = \prod_{i=1}^{\mu} k_i$ transcripts arranged in the following tree structure. The nodes in this tree correspond to the prover's messages and the edges to the verifier's challenges. Every node at depth i has precisely k_i children corresponding to k_i pairwise distinct challenges. Every transcript corresponds to exactly one path from the root node to a leaf node. For a graphical representation we refer to Figure 1. We refer to the corresponding tree of challenges as a $(k_1, \ldots, k_\mu)$-tree of challenges. The set of all $(k_1, \ldots, k_\mu)$-trees of challenges is denoted by $\textsc{Tree}(k_1, \ldots, k_\mu)$.*

We will also write $\mathbf{k} = (k_1, \ldots, k_\mu) \in \mathbb{N}^\mu$ and refer to a $\mathbf{k}$-tree of transcripts.

Definition 10 ($(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ Special-Soundness). *Let $k_1, \ldots, k_\mu, N_1, \ldots, N_\mu \in \mathbb{N}$. A $(2\mu + 1)$-round public-coin protocol $(\mathcal{P}, \mathcal{V})$ for relation R, where $\mathcal{V}$ samples the i-th challenge from a set of cardinality $N_i \geq k_i$ for $1 \leq i \leq \mu$, is $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound if there exists a polynomial time algorithm that, on input a statement x and a $(k_1, \ldots, k_\mu)$-tree of accepting transcripts outputs a witness $w \in R(x)$. We also say $(\mathcal{P}, \mathcal{V})$ is $(k_1, \ldots, k_\mu)$-special-sound.*

It is well known that, for 3-round protocols, k-special-soundness implies knowledge soundness, but only recently it was shown that more generally, for public-coin $(2\mu + 1)$-round protocols, $(k_1, \ldots, k_\mu)$-special-soundness tightly implies knowledge soundness [ACK21].

2.2 Geometric Distribution

A random variable X with support $\{0, 1\}$ is geometrically distributed with parameter $p \in [0, 1]$ if $\Pr(X = 1) = p$. In this case we write $X \sim \text{Geo}(p)$.

Lemma 1. *Let $X \sim \text{Geo}(p)$ and $Y \sim \text{Geo}(q)$ be independently distributed. Then,*

$$\Pr(X \leq Y) = \frac{p}{p + q - pq} \geq \frac{p}{p + q}.$$

Proof. It holds that

$$\Pr(X \leq Y) = \sum_{x=1}^{\infty} \Pr(X = x) \Pr(Y \geq x) = \sum_{x=1}^{\infty} (1-p)^{x-1} p \cdot (1-q)^{x-1}$$

$$= \frac{p}{1-(1-p)(1-q)} \sum_{x=1}^{\infty} \left((1-p)(1-q)\right)^{x-1} \left(1 - (1-p)(1-q)\right)$$

$$= \frac{p}{1-(1-p)(1-q)} \sum_{x=1}^{\infty} \Pr(Z = x)$$

$$= \frac{p}{1-(1-p)(1-q)} = \frac{p}{p+q-pq} \geq \frac{p}{p+q},$$

where $Z \sim \mathrm{Geo}(p + q - pq)$. This completes the proof of the lemma. $\square$

3 Parallel Repetition of k-Special-Sound Σ-Protocols

To simplify the exposition, we start with the simpler case of Σ-protocols; the general case of multi-round protocols will then be treated in the subsequent section. Thus, for the remainder of this section, we consider a k-special-sound public-coin interactive proof $(\mathcal{P}, \mathcal{V})$ with challenge set $\mathcal{C}$ of cardinality $N \geq k$. It is well known that such an interactive proof is a proof of knowledge with knowledge error $\kappa = (k-1)/N$. We write $(\mathcal{P}^t, \mathcal{V}^t)$ for the t-fold parallel repetition of $(\mathcal{P}, \mathcal{V})$, which runs t instances of $(\mathcal{P}, \mathcal{V})$ in parallel and the verifier $\mathcal{V}^t$ accepts if all the parallel instances are accepted. In this section, we prove that $(\mathcal{P}^t, \mathcal{V}^t)$ is then again a proof of knowledge, but now with knowledge error κ^t, which is optimal. Thus, we show what is sometimes referred to as *strong* parallel repetition, meaning that the figure of merit decreases with power t under parallel repetition. This is well known to hold for special-sound Σ-protocols, i.e., for $k = 2$, but was open for general k.

The standard way to reason about parallel repetition for the special case $k = 2$ uses the fact that $(\mathcal{P}^t, \mathcal{V}^t)$ is ℓ-special-sound with $\ell = (k-1)^t + 1$. However, this reasoning does not apply in general, because ℓ grows exponentially in t for $k > 2$. Instead, our result crucially depends on the fact that $(\mathcal{P}^t, \mathcal{V}^t)$ is the t-fold parallel repetition of a k-special-sound protocol $(\mathcal{P}, \mathcal{V})$. In Section 3.1, we first construct a novel extraction algorithm for k-special-sound protocols $(\mathcal{P}, \mathcal{V})$ thereby reproving that k-special-soundness implies knowledge soundness [ACK21]. In Section 3.2, we show how this extraction algorithm can be used to deduce a strong parallel repetition result for $(\mathcal{P}^t, \mathcal{V}^t)$. In Section 4, we then extend our results to multi-round protocols.

On a high level, the crucial ingredient in our analyses is to introduce and work with a more "fine-grained" notion of success probability of a dishonest prover, as we introduce it below.

3.1 Knowledge Soundness of a Single Invocation

Consider a dishonest prover $\mathcal{P}^*$ against the considered k-special-sound interactive proof $(\mathcal{P}, \mathcal{V})$. The goal of the extractor is to run $\mathcal{P}^*$ and rewind it sufficiently many times so as to obtain a first message a together with k correct answers $z_1, \ldots, z_k$ for k pairwise distinct challenges $c_1, \ldots, c_k \in \mathcal{C}$. The crucial question is how often $\mathcal{P}^*$ needs to be rewound, and thus what is the (expected) running time of the extractor. Alternatively, towards satisfying Definition 4, we would like to have an extractor that runs in a fixed (expected) polynomial time, but may fail with some probability. It is quite clear that in both cases the figure of merit (i.e., the running time in the former and the success probability in the latter) depends on the *success probability* ϵ of $\mathcal{P}^*$; for instance, if ϵ is below the knowledge soundness κ then we cannot expect extraction to work in general. However, a crucial observation is that for a given dishonest prover $\mathcal{P}^*$, its success probability ϵ does actually not characterize (very well) whether extraction is possible or not: if in a special-sound Σ-protocol $\mathcal{P}^*$ provides the correct response with probability ϵ (and fails to do so with probability $1 - \epsilon$) *for every* possible choice of the challenge, then extraction is still possible even when $\epsilon < \kappa$ (but not negligible), simply by trying

sufficiently many times for two distinct challenges. Below, we will identify an alternative, in some sense more fine-grained, "quality measure" of $\mathcal{P}^*$, and we show that this measure does characterize when extraction is possible. This will be helpful when it comes to more complicated settings, like a *parallel repetition*, or a *multi-round* protocol, or, ultimately, a *parallel repetition* of a *multi-round* protocol.

For multiple reasons, we will state and prove our core technical results in a more abstract language. One reason is that this allows us to focus on the important aspects; another reason is that we will actually exploit the considered abstraction, and thus generalization, of the considered problem. In our abstraction, we consider an arbitrary function $V \colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$, $(c,y) \mapsto V(c,y)$, and we consider an arbitrary (possibly probabilistic) algorithm $\mathcal{A}$ that takes as input an element $c \in \mathcal{C}$ and outputs a string $y \leftarrow \mathcal{A}(c)$. The *success probability* of $\mathcal{A}$ is then naturally defined as

$$\epsilon^V(\mathcal{A}) := \Pr\bigl(V(C, \mathcal{A}(C)) = 1\bigr),$$

where, here and below, the probability space is defined by means of the randomness of $\mathcal{A}$ and the random variable C being uniformly random in $\mathcal{C}$. If V is clear from context, we simply write $\epsilon(\mathcal{A})$.

The obvious instantiation of $\mathcal{A}$ is given by a *deterministic*[6] dishonest prover $\mathcal{P}^*$ attacking the considered k-special-sound interactive proof $(\mathcal{P}, \mathcal{V})$ on input x. More precisely, on input c, $\mathcal{A}$ runs $\mathcal{P}^*(x)$ sending c as the challenge, and outputs $\mathcal{P}^*(x)$'s (fixed) first message a and its response z, and the function V is defined as the verification check that $\mathcal{V}$ performs. We point out that this instantiation gives rise to a deterministic $\mathcal{A}$; however, later on it will be crucial that in our abstract treatment, $\mathcal{A}$ may be an arbitrary *randomized* algorithm that decides on its output y in a randomized manner given the input c, and that V is arbitrary.

Motivated by the k-special-soundness of the considered protocol, given x and (oracle access to) $\mathcal{A}$ the goal will be to find correct responses $y_1, \ldots, y_k$ for k pairwise distinct challenges $c_1, \ldots, c_k \in \mathcal{C}$, i.e., such that $V(c_i, y_i) = 1$ for all i. As we show below, the measure that captures how well this can be done is

$$\delta_k^V(\mathcal{A}) := \min_{S \subset \mathcal{C} : |S| < k} \Pr\bigl(V(C, \mathcal{A}(C)) = 1 \mid C \notin S\bigr).$$

More precisely, we argue existence of an extraction algorithm $\mathcal{E}^{\mathcal{A}}$ with oracle access to $\mathcal{A}$, that runs in expected polynomial time and succeeds with probability at least $\delta_k^V(\mathcal{A})/k$. As before, if V is clear from context, we write $\delta_k(\mathcal{A})$.

Lemma 2 (Extraction Algorithm). *Let $k \in \mathbb{N}$, $\mathcal{C}$ a finite set with cardinality $N \geq k$ and let $V \colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$.*

Then there exists an algorithm $\mathcal{E}^{\mathcal{A}}$ with the following properties: The algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} \colon \mathcal{C} \to \{0,1\}^*$, requires an expected number of at most $2k$ queries to $\mathcal{A}$ and, with probability at least $\delta_k^V(\mathcal{A})/k$, it outputs k pairs $(c_1, y_1), (c_2, y_2), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

Proof. The extraction algorithm is defined recursively over k. For this reason, we add a subscript k and write $\mathcal{E}_k^{\mathcal{A}}$ for the extraction algorithm that aims to output k pairs (c_i, y_i). In this proof, we also make the set $\mathcal{D} \subset \mathcal{C}$, from which the extractor samples the challenges c_i, explicit by writing $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$. This allows the extractor to be deployed on subsets $\mathcal{D}$ of the full challenge set $\mathcal{C}$, i.e., extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ aims to output k pairs (c_i, y_i) with pairwise distinct challenges $c_i \in \mathcal{D}$ and $V(c_i, y_i) = 1$ for all i. When writing $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ we will always implicitly assume that $|\mathcal{D}| \geq k$. Accordingly, we also write

$$\epsilon^V(\mathcal{A}, \mathcal{D}) := \Pr\bigl(V(C, \mathcal{A}(C)) = 1\bigr),$$
$$\delta_k^V(\mathcal{A}, \mathcal{D}) := \min_{S \subset \mathcal{D} : |S| < k} \Pr\bigl(V(C, \mathcal{A}(C)) = 1 \mid C \notin S\bigr).$$

where the probability space is defined by means of the randomness of $\mathcal{A}$ and the random variable C being uniformly random in $\mathcal{D} \subset \mathcal{C}$. Note that for all $k \geq 1$, $\delta_{k+1}^V(\mathcal{A}, \mathcal{D}) \leq \delta_k^V(\mathcal{A}, \mathcal{D}) \leq \epsilon^V(\mathcal{A}, \mathcal{D})$.

---

[6] Recall that, in order to prove knowledge soundness, it is sufficient to consider deterministic provers (Remark 1).

Let us now define the extraction algorithm. Its pseudocode is presented in Algorithm 1. Let $\mathcal{D} \subset \mathcal{C}$ be an arbitrary subset with cardinality at least k. For $k = 1$, the extractor $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ simply samples a challenge $c_1 \in \mathcal{D}$ uniformly at random and computes $y_1 \leftarrow \mathcal{A}(c_1)$. If $V(c_1, y_1) = 0$, it outputs $\perp$ and aborts. Otherwise, if $V(c_1, y_1) = 1$, it successfully outputs (c_1, y_1). This extractor queries $\mathcal{A}$ once and it succeeds with probability $\epsilon^V(\mathcal{A}, \mathcal{D}) = \delta_1^V(\mathcal{A}, \mathcal{D})$.

For $k > 1$, the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ first runs the extractor $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$. If $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ fails, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ outputs $\perp$ and aborts. Otherwise, if $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ succeeds to output a pair (c_1, y_1), $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ proceeds as follows. It defines the set $\mathcal{D}' = \mathcal{D} \setminus \{c_1\}$ and runs $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$. If $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$ succeeds to output $k-1$ pairs $(c_2, y_2), \ldots (c_k, y_k)$, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ successfully outputs k pairs $(c_1, y_1), \ldots, (c_k, y_k)$. If $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$ fails, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ tosses a coin that returns heads with probability $\epsilon^V(\mathcal{A}, \mathcal{D})$. This coin can be implemented by running $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$, i.e., sampling a random challenge $c \leftarrow \mathcal{D}$ and evaluating $V(c, \mathcal{A}(c))$. If the coin returns heads, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ outputs $\perp$ and aborts. If the coin returns tails, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ runs $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$ once more and performs the same steps as before. The algorithm proceeds in this manner until either it has successfully found a k pairs (c_i, y_i) or until the coin returns heads.

Let us now analyze the success probability and the expected number of $\mathcal{A}$-queries of this algorithm.

Success Probability. The analysis goes by induction. The induction hypothesis is that, for all $k < K$, the success probability $\Delta_k(\mathcal{D})$ of the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ satisfies

$$\Delta_k(\mathcal{D}) \geq \delta_k^V(\mathcal{A}, \mathcal{D})/k.$$

Since $\Delta_1(\mathcal{D}) = \epsilon^V(\mathcal{A}, \mathcal{D}) = \delta_1^V(\mathcal{A}, \mathcal{D})/1$, the induction hypothesis is satisfied for the case $k = 1$.

Let us now consider the case $k = K$. Then if, in its first step, $\mathcal{E}_K^{\mathcal{A}}(\mathcal{D})$ successfully runs extractor $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ (outputting a pair (c_1, y_1) with $V(c_1, y_1) = 1$), it starts running two geometric experiments until one of them finishes. In the first geometric experiment the extractor aims to find an additional set of $K - 1$ pairs (c_i, y_i) by running $\mathcal{E}_{K-1}^{\mathcal{A}}(\mathcal{D}')$, where $\mathcal{D}' = \mathcal{D} \setminus \{c_1\}$. By the induction hypothesis, the parameter p of this geometric distribution satisfies

$$p = \Delta_{K-1}(\mathcal{D}') \geq \delta_{K-1}^V(\mathcal{A}, \mathcal{D}')/(K-1) \geq \delta_K^V(\mathcal{A}, \mathcal{D})/(K-1).$$

In the second geometric experiment the extractor tosses a coin that returns heads with probability

$$q := \epsilon^V(\mathcal{A}, \mathcal{D}) \geq \delta_K^V(\mathcal{A}, \mathcal{D}).$$

The second step of the extractor succeeds if the second geometric experiment does not finish before the first, i.e., it succeeds with probability

$$\Pr\big(\mathrm{Geo}(p) \leq \mathrm{Geo}(q)\big) \geq \frac{\Delta_{K-1}(\mathcal{D}')}{\Delta_{K-1}(\mathcal{D}') + \epsilon^V(\mathcal{A}, \mathcal{D})} \geq \frac{\delta_K^V(\mathcal{A}, \mathcal{D})/(K-1)}{\delta_K^V(\mathcal{A}, \mathcal{D})/(K-1) + \epsilon^V(\mathcal{A}, \mathcal{D})}$$
$$\geq \frac{\delta_K^V(\mathcal{A}, \mathcal{D})/(K-1)}{\epsilon^V(\mathcal{A}, \mathcal{D})/(K-1) + \epsilon^V(\mathcal{A}, \mathcal{D})} = \frac{\delta_K^V(\mathcal{A}, \mathcal{D})}{K \cdot \epsilon^V(\mathcal{A}, \mathcal{D})},$$

where the first inequality follows from Lemma 1 and the second from the monotonicity of the function $x \mapsto \frac{x}{x+q}$. Since the first step of the extractor succeeds with probability $\epsilon^V(\mathcal{A}, \mathcal{D})$, it follows that $\mathcal{E}_K^{\mathcal{A}}(\mathcal{D})$ succeeds with probability at least $\delta_K^V(\mathcal{A}, \mathcal{D})/K$.

Therefore, by induction it follows that for all k, the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ succeeds with probability at least $\delta_k^V(\mathcal{A}, \mathcal{D})/k$. In particular, the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{C})$ succeeds with probability at least $\delta_k^V(\mathcal{A})/k$, which proves that this extractor has the desired success probability.

Expected Number of $\mathcal{A}$-Queries. Let $Q_k(\mathcal{D})$ be the expected number of $\mathcal{A}$-queries made by $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ and let $Q_k = \max_{\mathcal{D} \subset \mathcal{C}} Q_k(\mathcal{D})$. We will prove that $Q_k \leq 2k$ for all $k \in \mathbb{N}$. The proof of this claim goes by induction. First note that, since $Q_1(\mathcal{D}) = 1$ for all $\mathcal{D} \subset \mathcal{C}$, this claim is clearly satisfied for $k = 1$. Let us assume the claim is satisfied for all $k < K$.

Then, $\mathcal{E}_K^{\mathcal{A}}(\mathcal{D})$ first runs $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ which requires exactly $Q_1(\mathcal{D}) = 1$ query. Then with probability $\epsilon^V(\mathcal{A}, \mathcal{D})$ it continues to the second step. In each iteration of the second step $\mathcal{E}_K^{\mathcal{A}}(\mathcal{D})$ runs $\mathcal{E}_{K-1}^{\mathcal{A}}(\mathcal{D}')$, for some $D' \subset \mathcal{C}$, and it tosses a coin by running $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$. Therefore, each iterations requires an expected number of at most

$Q_{K-1}(\mathcal{D}') + 1 \le Q_{K-1} + 1 \le 2K - 1$ queries. Moreover, the expected number of tosses until the coin returns heads is $1/\epsilon^V(\mathcal{A}, \mathcal{D})$. Hence, the expected number of iterations in the second step of this extraction algorithm is at most $1/\epsilon^V(\mathcal{A}, \mathcal{D})$. It follows that, for all $\mathcal{D} \subset \mathcal{C}$,

$$Q_K(\mathcal{D}) \le 1 + \epsilon^V(\mathcal{A}, \mathcal{D}) \frac{1}{\epsilon^V(\mathcal{A}, \mathcal{D})} (2K - 1) = 2K \,,$$

which proves the claimed upper bound on the expected number of $\mathcal{A}$-queries and completes the proof of the lemma.

$\square$

In the context of a deterministic dishonest prover $\mathcal{P}^*$ attacking a k-special-sound protocol, we make the following observation. First, by basic probability theory,

$$\begin{aligned}
\Pr\big(V(C, \mathcal{A}(C)) = 1 \mid C \notin S\big) &= \frac{\Pr\big(V(C, \mathcal{A}(C)) = 1 \wedge C \notin S\big)}{\Pr\big(C \notin S\big)} \\
&\ge \frac{\Pr\big(V(C, \mathcal{A}(C)) = 1\big) - \Pr\big(C \in S\big)}{\Pr\big(C \notin S\big)} \,.
\end{aligned} \tag{1}$$

Thus, extractor $\mathcal{E}^{\mathcal{A}}$ succeeds with positive probability as soon as $\epsilon(\mathcal{A}) > \Pr\big(C \in S\big)$ for every $S \subset \mathcal{C}$ with $|S| < k$. More precisely,

$$\Pr\big(\mathcal{E}^{\mathcal{A}} \text{ succeeds}\big) \ge \delta_k(\mathcal{A})/k \ge \frac{\epsilon(\mathcal{A}) - \kappa}{k(1 - \kappa)} \,,$$

where $\kappa = (k-1)/N$.

This observations confirms that k-special-soundness implies knowledge soundness with knowledge error κ (see also [ACK21] for an alternative proof). This result is summarized in Theorem 1.

Theorem 1. *Let $(\mathcal{P}, \mathcal{V})$ be a k-out-of-N special-sound Σ-protocol for relation R. Then $(\mathcal{P}, \mathcal{V})$ is knowledge sound with knowledge error $\kappa = (k-1)/N$.*

Note that this is the best we can hope for, since it may be — and for typical schemes this is the case — that for any $S \subset \mathcal{C}$ with $|S| < k$, $\mathcal{P}^*$ can prepare a first message a for which he can correctly answer any challenge $c \in S$. Thus, $\kappa = (k-1)/N$ is the trivial cheating probability, confirming the tightness of the theorem.

3.2 Knowledge-Soundness of the Parallel Repetition

When moving to the t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$ of the k-special-sound public-coin protocol $(\mathcal{P}, \mathcal{V})$, we consider an algorithm $\mathcal{A}$ that takes as input a row $(c_1, \ldots, c_t) \in \mathcal{C}^t$ of challenges[7] and outputs a string y, and the *success probability* of $\mathcal{A}$ is then defined as

$$\epsilon^V(\mathcal{A}) = \Pr\big(V(C_1, \ldots, C_t, \mathcal{A}(C_1, \ldots, C_t)) = 1\big) \,,$$

for some given $V : \mathcal{C}^t \times \{0,1\}^* \to \{0,1\}$ and where the C_i are understood to be independently and uniformly distributed over $\mathcal{C}$.

The obvious instantiation of $\mathcal{A}$ is given by a deterministic prover P^* attacking the considered t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$ of $(\mathcal{P}, \mathcal{V})$. More precisely, on input $(c_1, \ldots, c_t)$, $\mathcal{A}$ runs $\mathcal{P}^*(x)$ sending $(c_1, \ldots, c_t)$ as the challenges for the t repetitions of $(\mathcal{P}, \mathcal{V})$, and outputs $\mathcal{P}^*(x)$'s (fixed) first messages $(a_1, \ldots, a_t)$ and its responses $(z_1, \ldots, z_t)$, and the function V is defined as the verification procedure of $\mathcal{V}^t$, which checks each repetition independently and accepts only if all are correct.

---

[7] There is no rigorous meaning in the list of challenges forming a *row*; it is merely that later we will also consider a *column* of challenges, which will then play a different *contextual* role.

Such an $\mathcal{A}$ naturally induces t algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$ as considered above in the context of a single execution of a k-special-sound protocol, taking *one* challenge as input: on input c_i, the algorithm $\mathcal{A}_i$ runs $y \leftarrow \mathcal{A}(c_1, \ldots, c_t)$ with c_j chosen uniformly at random from $\mathcal{C}$ for $j \neq i$, and outputs y along with the c_j's for $j \neq i$. We can thus run the extractor from above on all of the $\mathcal{A}_i$'s individually, with the goal being that at least one of them succeeds. We know that for each $\mathcal{A}_i$ individually, the extraction succeeds with probability

$$\delta_k^V(\mathcal{A}_i) = \min_{S_i \subset \mathcal{C}: |S_i| < k} \Pr\big(V(C_i, \mathcal{A}_i(C_i)) = 1 \mid C_i \notin S_i\big), \tag{2}$$

where V is understood to appropriately reorder its inputs. The following lemma allows us to bound the probability that at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ succeeds to produce either k challenge-response pairs $((c_1, \ldots, c_t), y)$ that all verify V and for which the k choices of c_i are all distinct for the considered i.

Lemma 3. *Let* $x \in \{0,1\}^*$, $k, t \in \mathbb{N}$, $\mathcal{C}$ *a finite set with cardinality* $N \geq k$, $V : \mathcal{C}^t \times \{0,1\}^* \to \{0,1\}$, *and* $\mathcal{A}$ *a (probabilistic) algorithm that takes as input a vector* $(c_1, \ldots, c_t) \in \mathcal{C}^t$ *and outputs a string* $y \in \{0,1\}^*$. *Then*

$$\sum_{i=1}^{t} \delta_k^V(\mathcal{A}_i) \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^t}{1 - \kappa},$$

where $\kappa = (k-1)/N$.

Proof. Let Λ denote the event $V(C_1, \ldots, C_t, \mathcal{A}(C_1, \ldots, C_t)) = 1$ and, for $1 \leq i \leq t$, let S_i be such that it minimizes Equation 2. Moreover, let Γ_i denote the event $C_i \notin S_i$.

Without loss of generality, we may assume that $|S_i| = k - 1$ for all i. Then, for all i,

$$\Pr(\Gamma_i) = \Pr(c_i \notin S_i) = 1 - \kappa.$$

Moreover, using elementary probability theory,

$$\sum_{i=1}^{t} \delta_k^V(\mathcal{A}_i) = \sum_{i=1}^{t} \Pr\big(V(C_i, \mathcal{A}_i(C_i)) = 1 \mid C_i \notin S_i\big) = \sum_{i=1}^{t} \Pr\big(\Lambda \mid \Gamma_i\big) = \sum_{i=1}^{t} \frac{\Pr\big(\Lambda \wedge \Gamma_i\big)}{\Pr\big(\Gamma_i\big)}$$

$$= \sum_{i=1}^{t} \frac{\Pr\big(\Lambda \wedge \Gamma_i\big)}{1 - \kappa} \geq \frac{\Pr\big(\Lambda \wedge \exists i : \Gamma_i\big)}{1 - \kappa} \geq \frac{\Pr\big(\Lambda\big) - \Pr\big(\neg \Gamma_i \; \forall i\big)}{1 - \kappa} = \frac{\epsilon^V(\mathcal{A}) - \kappa^t}{1 - \kappa},$$

which completes the proof. $\square$

Now let $\Delta = \min\big(1, \sum_{i=1}^{t} \delta_k^V(\mathcal{A}_i)/k\big)$. Then the probability that at least one extractor $\mathcal{E}^{\mathcal{A}_i}$ succeeds equals

$$1 - \prod_{i=1}^{t}(1 - \delta_k^V(\mathcal{A}_i)/k) \geq 1 - \left(1 - \frac{\Delta}{t}\right)^t \geq 1 - e^{-\Delta} \geq (1 - e^{-1})\Delta \geq \frac{1}{2}\Delta, \tag{3}$$

where the first inequality follows from the inequality of arithmetic and geometric means and the third inequality follows since $0 \leq \Delta \leq 1$. Hence, by Lemma 3, the probability of at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ being successful is at least

$$\frac{\Delta}{2} \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^t}{2k(1 - \kappa)}.$$

From this it follows that the t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$ of a k-special-sound protocol $(\mathcal{P}, \mathcal{V})$ is knowledge sound with knowledge error κ^t, where $\kappa = (k-1)/N$ is the knowledge error of a single execution of $(\mathcal{P}, \mathcal{V})$. This parallel repetition result for k-special-sound Σ-protocols is formalized in Theorem 2.

Theorem 2 (Parallel Repetition of k-Special-Sound Σ-Protocols). *Let* $(\mathcal{P}, \mathcal{V})$ *be a k-out-of-N special-sound Σ-protocol. Let* $(\mathcal{P}^t, \mathcal{V}^t)$ *be the t-fold parallel repetition of protocol* $(\mathcal{P}, \mathcal{V})$. *Then* $(\mathcal{P}^t, \mathcal{V}^t)$ *is knowledge sound with knowledge error* κ^t *for* $\kappa = (k-1)/N$.

Also here we have that the knowledge error κ^t matches the trivial cheating probability, which succeeds if in each instance of the parallel repetition the challenge falls into a given set of size $k-1$.

Remark 3. The above parallel repetition result (and also the generalization of Section 4), directly generalize to the parallel composition of t *different* protocols or to the parallel composition of t different instances of the same protocol. In this case, the knowledge error will be the product of the individual knowledge errors.

4 Parallel Repetition of Multi-Round Protocols

We now consider the general case of multi-round protocols. The line of reasoning is quite similar to that of 3-round protocols, but with an appropriately adjusted definition of δ. So, for the remainder of this section, we consider a $(k_1, \ldots, k_\mu)$-special-sound $(2\mu + 1)$-round public-coin interactive proof $(\mathcal{P}, \mathcal{V})$, where the verifier samples its i-th challenge uniformly at random from a finite set $\mathcal{C}^{[j]}$ for $1 \leq j \leq \mu$. We denote the superscript j with square brackets to distinguish the set $\mathcal{C}^{[j]}$ from the j-fold Cartesian product $\mathcal{C}^j$. Eventually, we want to analyze its t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$, but again we first consider a single invocation.

4.1 Knowledge Soundness of a Single Invocation

Here, we consider a (possibly randomized) algorithm $\mathcal{A}$ that takes as input a column $(c^1, \ldots, c^\mu) \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ of challenges and outputs a string y, and we consider a function

$$V \colon \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]} \times \{0,1\}^* \to \{0,1\}.$$

The obvious instantiation is a deterministic prover $\mathcal{P}^*$ attacking the considered protocol. Formally, on input $(c^1, \ldots, c^\mu)$, $\mathcal{A}$ runs $\mathcal{P}^*(x)$, sending c^1 in the first challenge round, c^2 in the second, etc., and eventually $\mathcal{A}$ outputs all of $\mathcal{P}^*(x)$'s messages. Then $V \colon \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]} \times \{0,1\}^* \to \{0,1\}$ captures the verification procedure of $\mathcal{V}$, i.e., $V(c^1, \ldots, c^\mu, y) = 1$ if and only if the corresponding transcript is accepting. This instantiation results in a deterministic algorithm $\mathcal{A}$. However, again it is crucial that in general $\mathcal{A}$ may be probabilistic, i.e., its output y is not necessarily uniquely determined by its input $(c^1, \ldots, c^\mu)$.

Syntactically identical to the previous section, the *success probability* of $\mathcal{A}$ is defined as

$$\epsilon^V(\mathcal{A}) := \Pr\big(V(C, \mathcal{A}(C)) = 1\big),$$

where $C = (C^1, \ldots, C^\mu)$ is uniformly random in $\mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$. However, here the goal of the extractor is different: the goal is to find correct responses for a $\mathbf{k}$-tree of challenges, where $\mathbf{k} = (k_1, \ldots, k_\mu)$. Generalizing the case of ordinary 3-round protocols, the figure of merit here is

$$\delta_{\mathbf{k}}^V(\mathcal{A}) := \min_{S^{[1]}, S^{[2]}(\cdot), \ldots, S^{[\mu]}(\cdot)} \Pr\left(V(C, \mathcal{A}(C)) = 1 \,\middle|\, \begin{matrix} C^1 \notin S^{[1]} \wedge C^2 \notin S^{[2]}(C^1) \wedge \cdots \\ \cdots \wedge C^\mu \notin S^{[\mu]}(C^1, \ldots, C^{\mu-1}) \end{matrix} \right), \tag{4}$$

where the minimum is over all sets $S^{[1]} \in \mathcal{C}^{[1]}|_{<k_1}$, and over all functions $S^{[2]} \colon \mathcal{C}^{[1]} \to \mathcal{C}^{[2]}|_{<k_2}$, $S^{[3]} \colon \mathcal{C}^{[1]} \times \mathcal{C}^{[2]} \to \mathcal{C}^{[3]}|_{<k_3}$, etc. Here for any set $\mathcal{C}$ and $k \in \mathbb{N}$, $\mathcal{C}|_{<k}$ denotes the set of subsets of $\mathcal{C}$ with cardinality smaller than k.

Indeed, the following lemma shows that there exists an expected polynomial time extractor $\mathcal{E}^\mathcal{A}$ with oracle access to $\mathcal{A}$ that, with probability $\delta_{\mathbf{k}}^V(\mathcal{A})/\prod_{i=1}^\mu k_i$, succeeds to extract correct responses for a $\mathbf{k}$-tree of challenges. Exploiting the abstract notation of Lemma 2, the proof of this lemma follows by induction over the number of challenges μ send by the verifier.

Lemma 4 (Multi-Round Extraction Algorithm). *Let $\mathbf{k} = (k_1, \ldots, k_\mu) \in \mathbb{N}^\mu$, $K = \prod_{i=1}^\mu k_i$, $\mathcal{C}^{[1]}, \ldots, \mathcal{C}^{[\mu]}$ finite sets $\mathcal{C}^{[j]}$ with cardinality $N_j \geq k_j$ and let $V \colon \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]} \times \{0,1\}^* \to \{0,1\}$.*

Then there exists an algorithm $\mathcal{E}^\mathcal{A}$ with the following properties: The algorithm $\mathcal{E}^\mathcal{A}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} \colon \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]} \to \{0,1\}^*$, requires an expected number of at most $2^\mu \cdot K$ queries to $\mathcal{A}$ and, with probability at least $\delta_{\mathbf{k}}^V(\mathcal{A})/K$, outputs K pairs $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_K, y_K) \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]} \times \{0,1\}^*$ with $V(\mathbf{c}_i, y_i) = 1$ for all i and such that the vectors $\mathbf{c}_i \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ form a $\mathbf{k}$-tree.

Proof. The proof goes by induction on μ. For $\mu = 1$, the lemma directly follow from Lemma 2. So let us assume the lemma holds for all $\mu \leq M$ and let us consider the case $\mu = M + 1$.

For any $c \in \mathcal{C}^{[1]}$, let $\mathcal{A}_c$ be the algorithm that takes as input a vector $(c^2, \ldots, c^\mu) \in \mathcal{C}^{[2]} \times \cdots \times \mathcal{C}^{[\mu]}$ and runs $\mathcal{A}(c, c^2, \ldots, c^\mu)$. The function V_c is defined accordingly, i.e.,

$$V_c \colon \mathcal{C}^{[2]} \times \cdots \times \mathcal{C}^{[\mu]} \times \{0,1\}^* \to \{0,1\}, \quad (\mathbf{c}, y) \mapsto V(c, \mathbf{c}, y).$$

Moreover, let $\mathbf{k}' = (k_2, \ldots, k_\mu) \in \mathbb{N}^{\mu-1}$ and $K' = \prod_{i=2}^{\mu} k_i$.

By the induction hypothesis there exists an algorithm $\mathcal{E}^{\mathcal{A}_c}$ that outputs a set $\mathcal{Y} = \{(c_i^2, \ldots, c_i^\mu, y_i)\}_{1 \leq i \leq K'}$ with

$$V(c, c_i^2, \ldots, c_i^\mu, y_i) = 1 \, \forall i \quad \text{and} \quad \{(c_i^2, \ldots, c_i^\mu)\}_i \in \text{Tree}(k_2, \ldots, k_\mu).$$

Moreover, $\mathcal{E}^{\mathcal{A}_c}$ requires an expected number of at most $2^{\mu-1} \cdot K'$ queries to $\mathcal{A}$ and succeeds with probability at least $\delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c)/K'$. We define $W \colon \mathcal{C}^{[1]} \times \{0,1\}^* \to \{0,1\}$, by setting $W(c, \mathcal{Y}) = 1$ if and only if $\mathcal{Y}$ is a set satisfying the above properties.

Now let $\mathcal{B} \colon \mathcal{C}^{[1]} \to \{0,1\}^*$ be the algorithm that takes as input an element $c \in \mathcal{C}^{[1]}$ and runs $\mathcal{E}^{\mathcal{A}_c}$. By Lemma 2, there exists an expected polynomial time algorithm $\mathcal{E}^{\mathcal{B}}$ that aims to output k_1 pairs $(c_1, \mathcal{Y}_1), \ldots, (c_{k_1}, \mathcal{Y}_{k_1}) \in \mathcal{C}^{[1]} \times \{0,1\}^*$ with $W(c_i, \mathcal{Y}_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$. The extractor $\mathcal{E}^{\mathcal{A}}$ simply runs $\mathcal{E}^{\mathcal{B}}$.

Let us now analyze the success probability and the expected number of $\mathcal{A}$-queries of the algorithm $\mathcal{E}^{\mathcal{B}}$ and therefore of $\mathcal{E}^{\mathcal{A}}$.

Success Probability. Again by Lemma 2, it follows that $\mathcal{E}^{\mathcal{B}}$ succeeds with probability at least

$$
\begin{aligned}
\delta_{k_1}^W(\mathcal{B})/k_1 &= \min_{S^{[1]} \subset \mathcal{C}^{[1]}, |S^{[1]}| < k_1} \frac{\Pr\big(W(C, \mathcal{B}(C)) = 1 \mid C \notin S^{[1]}\big)}{k_1} \\
&= \min_{S^{[1]} \subset \mathcal{C}^{[1]}, |S^{[1]}| < k_1} \frac{\Pr\big(W(C, \mathcal{B}(C)) = 1 \wedge C \notin S^{[1]}\big)}{k_1 \cdot \Pr(C \notin S^{[1]})} \\
&\geq \min_{S^{[1]} \subset \mathcal{C}^{[1]}, |S^{[1]}| < k_1} \frac{\sum_{c \notin S^{[1]}} \Pr(C = c) \cdot \Pr\big(W(c, \mathcal{B}(c)) = 1\big)}{k_1 \cdot \Pr(C \notin S^{[1]})} \\
&\geq \min_{S^{[1]} \subset \mathcal{C}^{[1]}, |S^{[1]}| < k_1} \frac{\sum_{c \notin S^{[1]}} \Pr(C = c) \cdot \delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c)}{k_1 \cdot K' \cdot \Pr(C \notin S^{[1]})} \\
&= \min_{S^{[1]} \subset \mathcal{C}^{[1]}, |S^{[1]}| < k_1} \frac{\sum_{c \notin S^{[1]}} \Pr(C = c) \cdot \delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c)}{K \cdot \Pr(C \notin S^{[1]})}
\end{aligned}
\tag{5}
$$

Now note that,

$$\delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c) = \min_{S^{[2]}(\cdot), \ldots, S^{[\mu]}(\cdot)} \Pr\left(\Lambda \, \middle| \, \begin{array}{l} C^1 = c \wedge C^2 \notin S^{[2]}(C^1) \wedge \cdots \\ \cdots \wedge C^\mu \notin S^{[\mu]}(C^1, \ldots, C^{\mu-1}) \end{array} \right),$$

where Λ denotes the event $V(C, \mathcal{A}(C)) = 1$. Hence,

$$\sum_{c \notin S^{[1]}} \Pr(C = c) \cdot \delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c) =$$

$$\min_{S^{[2]}(\cdot), \ldots, S^{[\mu]}(\cdot)} \Pr\left(\Lambda \wedge C^1 \notin S^{[1]} \, \middle| \, \begin{array}{l} C^2 \notin S^{[2]}(C^1) \wedge \cdots \\ \cdots \wedge C^\mu \notin S^{[\mu]}(C^1, \ldots, C^{\mu-1}) \end{array} \right),$$

Plugging this equality in Equation 5, shows that

$$\delta_{k_1}^W(\mathcal{B})/k_1 \geq \frac{\delta_{\mathbf{k}}^V(\mathcal{A})}{K},$$

14

which shows that $\mathcal{E}^{\mathcal{B}}$ has the desired success probability.

Expected Number of $\mathcal{A}$-Queries. By Lemma 2, it follows that $\mathcal{E}^{B}$ requires an expected number of at most $2k_1$ queries to $\mathcal{B}$. By the induction hypothesis it follows that $\mathcal{B}$ requires an expected number of at most $2^{\mu-1} \cdot K'$ queries to $\mathcal{A}$. Hence, $\mathcal{E}^{\mathcal{B}}$ requires an expected number of at most $2^{\mu} \cdot K$ queries to $\mathcal{A}$, which completes the proof of the lemma. $\qquad\square$

Let $S^{[1]}, S^{[2]}(\cdot), \ldots, S^{[\mu]}(\cdot)$ be the arguments minimizing Equation 4. Further, let Λ denote the event $V(C, \mathcal{A}(C)) = 1$ and let Γ denote the event

$$\Gamma = C^1 \notin S^{[1]} \wedge C^2 \notin S^{[2]}(C^1) \wedge \cdots \wedge C^\mu \notin S^{[\mu]}(C^1, \ldots, C^{\mu-1}).$$

Then, using the same kind of reasoning as in Equation 1, we have

$$\delta_{\mathbf{k}}^V(\mathcal{A}) = \Pr(\Lambda \mid \Gamma) = \frac{\Pr(\Lambda \wedge \Gamma)}{\Pr(\Gamma)} \geq \frac{\Pr(\Lambda) - \Pr(\neg\Gamma)}{\Pr(\Gamma)} = \frac{\epsilon^V(\mathcal{A}) - \kappa}{1 - \kappa},$$

where

$$\kappa = \Pr(\neg\Gamma) = 1 - \prod_{j=1}^{\mu} \frac{N_j - k_j + 1}{N_j}.$$

This confirms that a $(k_1, \ldots, k_\mu)$-special-sound protocol is knowledge sound with knowledge error κ. See [ACK21] for an alternative and the original proof of this statement. This result is formalized in Theorem 3.

Theorem 3. *Let $(\mathcal{P}, \mathcal{V})$ be a $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound protocol. Then $(\mathcal{P}, \mathcal{V})$ is knowledge sound with knowledge error*

$$\kappa = 1 - \prod_{j=1}^{\mu} \frac{N_j - k_j + 1}{N_j}.$$

Once more, κ matches the trivial cheating probability.

4.2   Knowledge-Soundness of the Parallel Repetition

We finally move towards stating and proving our main general parallel repetition result for multi-round protocols. Thus, consider the t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$ of the given $(k_1, \ldots, k_\mu)$-special-sound $(2\mu+1)$-round public-coin interactive proof $(\mathcal{P}, \mathcal{V})$.

We consider an algorithm $\mathcal{A}$ that takes as input a *row* $(\mathbf{c}_1, \ldots, \mathbf{c}_t)$ of *columns* $\mathbf{c}_i = (c_i^1, \ldots, c_i^\mu) \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ of challenges and outputs a string y. Furthermore, we consider a verification function V, which then defines the *success probability* of $\mathcal{A}$ as

$$\epsilon^V(\mathcal{A}) = \Pr\big(V(C, \mathcal{A}(C)) = 1\big),$$

where $C = (C_1, \ldots, C_t)$ with C_i distributed uniformly over $\mathcal{C}^{[1]} \times \cdots \mathcal{C}^{[\mu]}$ for all $1 \leq i \leq t$.

Again, the obvious instantiation for $\mathcal{A}$ is a deterministic dishonest prover $\mathcal{P}^*$ attacking $(\mathcal{P}^t, \mathcal{V}^t)$. More precisely, on input a row $(\mathbf{c}_1, \ldots, \mathbf{c}_t)$ of columns, $\mathcal{A}$ runs $\mathcal{P}^*(x)$ sending $(\mathbf{c}_1, \ldots, \mathbf{c}_t)$ as the challenges, and outputs all of $\mathcal{P}^*(x)$'s messages, and the function V is defined as the verification check that $\mathcal{V}^t$ performs.

Such an $\mathcal{A}$ naturally induces t algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$ as considered above in the context of a single execution of a multi-round protocol, taking one challenge-column as input and outputting one string: on input $\mathbf{c}_i$, the algorithm $\mathcal{A}_i$ runs $y \leftarrow \mathcal{A}(\mathbf{c}_1, \ldots, \mathbf{c}_\ell)$ with $\mathbf{c}_j$ chosen uniformly at random from $\mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ for $j \neq i$, and outputs y along with the $\mathbf{c}_j$'s for $j \neq i$. Thus, we can run the extractor from Lemma 4 on all of the $\mathcal{A}_i$'s individually, with the goal being that at least one of them succeeds. For each $\mathcal{A}_i$ individually, the extraction succeeds with probability at least

$$\delta_{\mathbf{k}}^V(\mathcal{A}_i)/K = \min_{S_i^{[1]}, S_i^{[2]}(\cdot), \ldots, S_i^{[\mu]}(\cdot)} \Pr\left(V(C_i, \mathcal{A}_i(C_i)) = 1 \,\middle|\, \begin{array}{l} C_i^1 \notin S_i^{[1]} \wedge C_i^2 \notin S_i^{[2]}(C_i^1) \wedge \cdots \\ \cdots \wedge C_i^\mu \notin S_i^{[\mu]}(C_i^1, \ldots, C_i^{\mu-1}) \end{array} \right)/K, \quad (6)$$

where V is understood to appropriately reorder its inputs and $K = \prod_{i=1}^{\mu} k_i$. The following lemma allows us to bound the probability that at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ succeeds.

Lemma 5. *Let $x \in \{0,1\}^*$, $\mathbf{k} \in \mathbb{N}^\mu$, $t \in \mathbb{N}$, $\mathcal{C}^{[1]}, \ldots, \mathcal{C}^{[\mu]}$ finite sets $\mathcal{C}^{[j]}$ with cardinality $N_j \geq k_j$, $V : \left(\mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]} \right)^t \times \{0,1\}^* \to \{0,1\}$, and $\mathcal{A}$ a (probabilistic) algorithm that takes as input a row $(\mathbf{c}_1, \ldots, \mathbf{c}_t)$ of columns $\mathbf{c}_i = (c_i^1, \ldots, c_i^\mu) \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ and outputs a string $y \in \{0,1\}^*$.*
Then

$$\sum_{i=1}^t \delta_{\mathbf{k}}^V(\mathcal{A}_i) \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^t}{1 - \kappa},$$

where

$$\kappa = 1 - \prod_{j=1}^\mu \frac{N_j - k_j + 1}{N_j}.$$

Proof. Let Λ denote the event $V(C, \mathcal{A}(C)) = 1$ and, for $1 \leq i \leq t$, let $S_i^{[1]}, S_i^{[2]}(\cdot) \ldots, S_i^{[\mu]}(\cdot)$ be such that they minimize Equation 6. Moreover, let Γ_i denote the event

$$C_i^1 \notin S_i^{[1]} \wedge C_i^2 \notin S_i^{[2]}(C_i^1) \wedge \cdots \wedge C_i^\mu \notin S_i^{[\mu]}(C_i^1, \ldots, C_i^{\mu-1}).$$

Without loss of generality, we may assume that $|S_i^{[1]}| = k_1 - 1$ and $S_i^{[j]} : \mathcal{C}^{[1]} \times \cdots \mathcal{C}^{[j-1]} \to \{S \subset \mathcal{C}^{[j]} : |S| = k_j - 1\}$ for all $2 \leq j \leq \mu$ and $1 \leq i \leq t$. Then, for all $1 \leq i \leq t$,

$$\Pr(\Gamma_i) = \prod_{j=1}^\mu \frac{N - k_j + 1}{N} = 1 - \kappa.$$

Moreover, using elementary probability theory,

$$\sum_{i=1}^t \delta_{\mathbf{k}}^V(\mathcal{A}_i) = \sum_{i=1}^t \Pr(\Lambda \mid \Gamma_i) = \sum_{i=1}^t \frac{\Pr(\Lambda \wedge \Gamma_i)}{\Pr(\Gamma_i)} = \sum_{i=1}^t \frac{\Pr(\Lambda \wedge \Gamma_i)}{1 - \kappa}$$

$$\geq \frac{\Pr(\Lambda \wedge \exists i : \Gamma_i)}{1 - \kappa} \geq \frac{\Pr(\Lambda) - \Pr(\neg \Gamma_i \ \forall i)}{1 - \kappa} = \frac{\epsilon^V(\mathcal{A}) - \kappa^t}{1 - \kappa},$$

which completes the proof. $\square$

As for the parallel repetition of a 3-round protocol, it follows that the probability of at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ being successful is at least

$$\frac{\Delta}{2} \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^t}{2K(1 - \kappa)},$$

where $\Delta = \min\left(1, \sum_{i=1}^t \delta_{\mathbf{k}}^V(\mathcal{A}_i)/K\right)$ and $K = \prod_{i=1}^\mu k_i$. This gives us the following strong parallel repetition result for $(k_1, \ldots, k_\mu)$-special-sound protocols.

Theorem 4 (Parallel Repetition Theorem for Multi-Round Protocols). *Let $(\mathcal{P}, \mathcal{V})$ be a $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound protocol. Let $(\mathcal{P}^t, \mathcal{V}^t)$ be the t-fold parallel repetition of protocol $(\mathcal{P}, \mathcal{V})$. Then $(\mathcal{P}^t, \mathcal{V}^t)$ is knowledge sound with knowledge error κ^t, where*

$$\kappa = 1 - \prod_{j=1}^\mu \frac{N_j - k_j + 1}{N_j},$$

is the knowledge error of $(\mathcal{P}, \mathcal{V})$.

Also here, the knowledge error κ^t coincides with the trivial cheating probability $\prod_i \Pr(\neg \Gamma_i)$, which is potentially achievable for any $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound protocol.

5 Threshold Parallel Repetition

The knowledge error κ^t of the t-fold parallel repetition $(\mathcal{P}^t, \mathcal{V}^t)$ of a $\mathbf{k}$-special-sound protocol $(\mathcal{P}, \mathcal{V})$ decreases exponentially with t. However, the completeness error of $(\mathcal{P}^t, \mathcal{V}^t)$ equals $\rho' = 1 - (1 - \rho)^t$, where ρ is the completeness error of $(\mathcal{P}, \mathcal{V})$. Hence, if $\rho \notin \{0, 1\}$, the completeness error of $(\mathcal{P}^t, \mathcal{V}^t)$ increases quickly with t. In order to decrease both the knowledge and the completeness error simultaneously, we consider a *threshold parallel repetition*. The s-out-of-t threshold parallel repetition of an interactive protocol $(\mathcal{P}, \mathcal{V})$, denoted by $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$, runs t instances of $(\mathcal{P}, \mathcal{V})$ in parallel and $\mathcal{V}^{s,t}$ accepts if at least s-out-of-t instances are accepted. In particular, it holds that $(\mathcal{P}^{t,t}, \mathcal{V}^{t,t}) = (\mathcal{P}^t, \mathcal{V}^t)$. In this section, we show that if $(\mathcal{P}, \mathcal{V})$ is $\mathbf{k}$-special-sound then $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$ is knowledge sound. We will immediately consider the general case of multi-round protocols.

As in Section 4.2, we consider an algorithm $\mathcal{A}$ that takes as input a row $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_t)$ of columns $\mathbf{c}_i = (c_i^1, \ldots, c_i^\mu) \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ of challenges and outputs a string y. However, this time we consider t different verification functions

$$V_i \colon \left(\mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}\right)^t \times \{0,1\}^* \to \{0,1\},$$

together with one additional *threshold* verification function defined as follows:

$$V(\mathbf{c}, y) = \begin{cases} 1 & \text{if } \sum_{i=1}^t V_i(\mathbf{c}, y) \geq s, \\ 0 & \text{otherwise}. \end{cases}$$

The obvious instantiation for $\mathcal{A}$ is a deterministic dishonest prover $\mathcal{P}^*$ attacking $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$. This instantiation defines V_i as the verification that the i-th instance of $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$ performs. The verification function V then captures the verification that $\mathcal{V}^{s,t}$ performs.

As before, such $\mathcal{A}$ induces t algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$ as considered in the context of a single execution of $(\mathcal{P}, \mathcal{V})$, taking one challenge-column as input and outputting one string: on input $\mathbf{c}_i$, the algorithm $\mathcal{A}_i$ runs $y \leftarrow \mathcal{A}(\mathbf{c}_1, \ldots, \mathbf{c}_\ell)$ with $\mathbf{c}_j$ chosen uniformly at random from $\mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ for $j \neq i$, and outputs y along with the $\mathbf{c}_j$'s for $j \neq i$. For each $\mathcal{A}_i$, we can run the extractor from Lemma 4, which succeeds with probability at least

$$\frac{\delta_\mathbf{k}^{V_i}(\mathcal{A}_i)}{\prod_{i=1}^\mu k_i} = \min_{S_i^{[1]}, S_i^{[2]}(\cdot), \ldots, S_i^{[\mu]}(\cdot)} \frac{\Pr\left(V_i(C_i, \mathcal{A}_i(C_i)) = 1 \,\middle|\, \begin{array}{l} C_i^1 \notin S_i^{[1]} \wedge C_i^2 \notin S_i^{[2]}(C_i^1) \wedge \\ \cdots \wedge C_i^\mu \notin S_i^{[\mu]}(C_i^1, \ldots, C_i^{\mu-1}) \end{array}\right)}{\prod_{i=1}^\mu k_i}, \tag{7}$$

where V_i is understood to appropriately reorder its inputs. The following lemma is a generalization of Lemma 5 and it allows us to bound the probability that at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ succeeds.

Lemma 6. *Let* $x \in \{0,1\}^*$, $\mathbf{k} \in \mathbb{N}^\mu$, $t \in \mathbb{N}$, $\mathcal{C}^{[1]}, \ldots, \mathcal{C}^{[\mu]}$ *finite sets* $\mathcal{C}^{[j]}$ *with cardinality* $N_j \geq k_j$ *and* $\mathcal{A}$ *a (probabilistic) algorithm that takes as input a row* $(\mathbf{c}_1, \ldots, \mathbf{c}_t)$ *of columns* $\mathbf{c}_i = (c_i^1, \ldots, c_i^\mu) \in \mathcal{C}^{[1]} \times \cdots \times \mathcal{C}^{[\mu]}$ *and outputs a string* $y \in \{0,1\}^*$.

Then

$$\sum_{i=1}^t \delta_\mathbf{k}^{V_i}(\mathcal{A}_i) \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^{s,t}}{1 - \kappa},$$

where

$$\kappa^{s,t} = \sum_{\ell=s}^t \binom{t}{\ell} \kappa^\ell (1 - \kappa)^{t-\ell} \quad \text{and} \quad \kappa = 1 - \prod_{j=1}^\mu \frac{N_j - k_j + 1}{N_j}.$$

Note that $\kappa^{s,t}$ is the probability of being successful at least s times when given t trials, when each trial is successful with independent probability κ.

Proof. For $1 \leq i \leq t$, let Λ_i denote the event $V_i(C, \mathcal{A}(C)) = 1$ and let $S_i^{[1]}, S_i^{[2]}(\cdot) \dots, S_i^{[\mu]}(\cdot)$ such that they minimize Equation 7. Moreover, let Γ_i denote the event

$$C_i^1 \notin S_i^{[1]} \wedge C_i^2 \notin S_i^{[2]}(C_i^1) \wedge \cdots \wedge C_i^\mu \notin S_i^{[\mu]}(C_i^1, \dots, C_i^{\mu-1}).$$

Without loss of generality, we may assume that $|S_i^{[1]}| = k_1 - 1$ and $S_i^{[j]} : \mathcal{C}^{[1]} \times \cdots \mathcal{C}^{[j-1]} \to \{S \subset \mathcal{C}^{[j]} : |S| = k_j - 1\}$ for all $2 \leq j \leq \mu$ and $1 \leq i \leq t$. Then, for all $1 \leq i \leq t$,

$$\Pr(\Gamma_i) = \prod_{j=1}^{\mu} \frac{N - k_j + 1}{N} = 1 - \kappa.$$

Moreover, using elementary probability theory,

$$\sum_{i=1}^{t} \delta_{\mathbf{k}}^{V_i}(\mathcal{A}_i) = \sum_{i=1}^{t} \Pr(\Lambda_i \mid \Gamma_i) = \sum_{i=1}^{t} \frac{\Pr(\Lambda_i \wedge \Gamma_i)}{\Pr(\Gamma_i)} = \sum_{i=1}^{t} \frac{\Pr(\Lambda_i \wedge \Gamma_i)}{1 - \kappa}$$
$$\geq \frac{\Pr(\exists i : \Lambda_i \wedge \Gamma_i)}{1 - \kappa} \geq \frac{\Pr(|\{i : \Lambda_i\}| \geq s \wedge |\{i : \Gamma_i\}| \geq t - s + 1)}{1 - \kappa}$$
$$\geq \frac{\Pr(|\{i : \Lambda_i\}| \geq s) - \Pr(|\{i : \Gamma_i\}| \leq t - s)}{1 - \kappa} \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^{s,t}}{1 - \kappa}.$$

which completes the proof. $\square$

As before (see Equation 3), it follows that the probability of at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ being successful is at least

$$\frac{\Delta}{2} \geq \frac{\epsilon^V(\mathcal{A}) - \kappa^{s,t}}{2K(1 - \kappa)},$$

where $\Delta = \min(1, \sum_{i=1}^{t} \delta_{\mathbf{k}}^{V_i}(\mathcal{A}_i)/K)$ and $K = \prod_{i=1}^{\mu} k_i$. This gives us the following threshold parallel repetition result for $(k_1, \dots, k_\mu)$-special-sound protocols.

Theorem 5 (Threshold Parallel Repetition Theorem). *Let $(\mathcal{P}, \mathcal{V})$ be a $(k_1, \dots, k_\mu)$-out-of-$(N_1, \dots, N_\mu)$ special-sound protocol. Let $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$ be the s-out-of-t threshold parallel repetition of protocol $(\mathcal{P}, \mathcal{V})$. Then $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$ is knowledge sound with knowledge error*

$$\kappa^{s,t} = \sum_{\ell=s}^{t} \binom{t}{\ell} \kappa^\ell (1 - \kappa)^{t-\ell},$$

where

$$\kappa = 1 - \prod_{j=1}^{\mu} \frac{N_j - k_j + 1}{N_j},$$

is the knowledge error of $(\mathcal{P}, \mathcal{V})$.

As before, the knowledge error $\kappa^{s,t}$ coincides with the trivial cheating probability for $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$, confirming the tightness of Theorem 5.

Note that the completeness error of $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$ equals

$$\rho^{s,t} = \sum_{\ell=0}^{s-1} \binom{t}{\ell} \rho^{t-\ell} (1 - \rho)^\ell.$$

Hence, the completeness error $\rho^{s,t}$ increases and the knowledge error decreases $\kappa^{s,t}$ in s. Moreover, it is easily seen that for t large enough and $\kappa \cdot t < s < (1-\rho)t$ the threshold parallel repetition $(\mathcal{P}^{s,t}, \mathcal{V}^{s,t})$ has a smaller knowledge and a smaller completeness error than $(\mathcal{P}, \mathcal{V})$, i.e., $\kappa^{s,t} < \kappa$ and $\rho^{s,t} < \rho$. In contrast to standard parallel repetition, threshold parallel repetition therefore allows both these errors to reduced.

6 Acknowledgments

References

AC20.   Thomas Attema and Ronald Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 513–543. Springer, Heidelberg, August 2020.

ACF21.   Thomas Attema, Ronald Cramer, and Serge Fehr. Compressing proofs of k-out-of-n partial knowledge. In *CRYPTO (4)*, volume 12828 of *Lecture Notes in Computer Science*, pages 65–91. Springer, 2021.

ACK21.   Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In *CRYPTO (2)*, volume 12826 of *Lecture Notes in Computer Science*, pages 549–579. Springer, 2021.

ACR21.   Thomas Attema, Ronald Cramer, and Matthieu Rambaud. Compressed Σ-protocols for bilinear group arithmetic circuits and application to logarithmic transparent threshold signatures. To appear in: *ASIACRYPT*, Lecture Notes in Computer Science. Springer, 2021.

ACX21.   Thomas Attema, Ronald Cramer, and Chaoping Xing. A note on short invertible ring elements and applications to cyclotomic and trinomials number fields. *Mathematical Cryptology*, pages 45–70, 2021.

BBB+18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.

BCC+16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016.

BFS20.   Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent SNARKs from DARK compilers. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 677–706. Springer, Heidelberg, May 2020.

BG93.   Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 390–420. Springer, Heidelberg, August 1993.

BIN97.   Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *38th FOCS*, pages 374–383. IEEE Computer Society Press, October 1997.

BLNS20. Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 441–469. Springer, Heidelberg, August 2020.

CL10.   Kai-Min Chung and Feng-Hao Liu. Parallel repetition theorems for interactive arguments. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 19–36. Springer, Heidelberg, February 2010.

CP15.   Kai-Min Chung and Rafael Pass. Tight parallel repetition theorems for public-coin arguments using KL-divergence. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 229–246. Springer, Heidelberg, March 2015.

GMR85.   Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

Gol98.   Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer, 1998.

Gol01.   Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

Hai09.   Iftach Haitner. A parallel repetition theorem for any interactive argument. In *50th FOCS*, pages 241–250. IEEE Computer Society Press, October 2009.

HL10.   Carmit Hazay and Yehuda Lindell. *Efficient Secure Two-Party Protocols - Techniques and Constructions*. ISC. Springer, Heidelberg, 2010.

HPWP10. Johan Håstad, Rafael Pass, Douglas Wikström, and Krzysztof Pietrzak. An efficient parallel repetition theorem. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 1–18. Springer, Heidelberg, February 2010.

LS18.   Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic
           rings and applications to lattice-based zero-knowledge proofs. In Jesper Buus Nielsen and Vincent Rij-
           men, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Heidelberg,
           April / May 2018.
MBKM19. Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs
           from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes
           Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2111–2128. ACM Press,
           November 2019.
PV07.   Rafael Pass and Muthuramakrishnan Venkitasubramaniam. An efficient parallel repetition theorem for
           Arthur-Merlin games. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 420–429.
           ACM Press, June 2007.
RS21.   Lior Rotem and Gil Segev. Tighter security for schnorr identification and signatures: A high-moment
           forking lemma for Σ-protocols. In *CRYPTO (1)*, volume 12825 of *Lecture Notes in Computer Science*,
           pages 222–250. Springer, 2021.
Unr12.   Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors,
           EUROCRYPT 2012, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012.

Appendix

A   Extraction Algorithm for k-Special-Sound Protocols

Algorithm 1: Extraction Algorithm $\mathcal{E}_k(\mathcal{D})$ for k-Special-Sound Protocols

---

1 Input: $k \in \mathbb{N}$ and a set $\mathcal{D}$ of cardinality at least k.

2 Output: $\perp$ or a set $\mathsf{output} = \{(c_1, y_1), \ldots, (c_k, y_k)\}$ with $(c_i, y_i) \in \mathcal{D} \times \{0,1\}^*$ and $V(c_i, y_i) = 1$ and $c_i \neq c_j$ for all i, j.

3 $c_1 \leftarrow_R \mathcal{D}$

4 $y_1 \leftarrow \mathcal{A}(c)$

5 if $V(c_1, y_1) = 0$ **then**

6 $\mathsf{output} = \perp$

7 else if $k = 1$ **then**

8 $\mathsf{output} = \{(c_1, y_1)\}$

9 else

10 $out_{k-1} = \perp$

11 $\mathrm{COIN} = 0$

12   **while** $\neg\mathrm{COIN} \wedge out_{k-1} = \perp$ **do**

13 $out_{k-1} \leftarrow \mathcal{E}_{k-1}(\mathcal{D} \setminus \{c\})$

14 $\mathrm{COIN} \leftarrow_R \mathrm{Ber}\big(\epsilon^V(\mathcal{A}, \mathcal{D})\big)$

15   **end**

16   **if** $out_{k-1} = \perp$ **then**

17 $\mathsf{output} = \perp$

18   **else**

19 $\mathsf{output} = \{(c_1, y_1)\} \cup out_{k-1}$

20   **end**

21 end