Parallel Repetition of Computationally Sound Protocols Revisited

Krzysztof Pietrzak[1] and Douglas Wikström[2]

[1] Ecole Normale Supérieure, Département d'informatique, pietrzak@di.ens.fr
[2] ETH Zürich, Department of Computer Science, douglas@inf.ethz.ch

Abstract. Parallel repetition is well known to reduce the error probability at an exponential rate for single- and multi-prover interactive *proofs*.
Bellare, Impagliazzo and Naor (1997) show that this is also true for protocols where the soundness only holds against computationally bounded provers (e.g. interactive arguments) if the protocol has at most three rounds.
On the other hand, for four rounds they give a protocol where this is no longer the case: the error probability does not decrease below some constant even if the protocol is repeated a polynomial number of times. Unfortunately, this protocol is not very convincing as the communication complexity of each instance of the protocol grows linearly with the number of repetitions, and for such protocols the error does not even decrease for some types of interactive *proofs*. Noticing this, Bellare et al. construct (a quite artificial) oracle relative to which a four round protocol exists whose communication complexity does not depend on the number of parallel repetitions. This shows that there is no "black-box" error reduction theorem for four round protocols.
In this paper we give the first computationally sound protocol where k-fold parallel repetition does not decrease the error probability below some constant for any polynomial k (and where the communication complexity does not depend on k). The protocol has eight rounds and uses the universal arguments of Barak and Goldreich (2001). We also give another *four* round protocol relative to an oracle, unlike the artificial oracle of Bellare et al., we just need a generic group. This group can then potentially be instantiated with some real group satisfying some well defined hardness assumptions (we do not know of any candidate for such a group at the moment).

1 Introduction

Interactive Proofs. In a (single prover) interactive proof a prover P tries to convince a computationally bounded verifier V that their common input x is in a language L. The soundness of such a protocol is an upper bound on the error probability of V, i.e. the probability that V accepts P's claim, even though $x \notin L$. In order to lower the error probability one can repeat the interactive proof k times, where V accepts the claim if it accepts in all k runs. The protocol can be repeated either *sequentially*, here V and P start the ith run of the protocol only after finishing the $(i-1)$th, or *in parallel*. Although the computational and communication complexity of parallel and sequential repetition is the same, parallel repetition has the big advantage of not increasing the round complexity. For single prover interactive proofs, sequential and parallel repetition reduce the error at an exponential rate: if a protocol with soundness ϵ is repeated k times sequentially or in parallel, the error probability drops to ϵ^k.

In general, parallel repetition is more problematic than sequential repetition. For example: parallel repetition does not preserve the zero-knowledge property of a protocol [8], and there are two-prover proofs where running the proof twice in parallel does not decrease the error at all [6]. On the positive side, Raz [12] shows that k-fold parallel repetition of a two-prover two-round proof system with soundness ϵ does decrease the error to $\epsilon^{\alpha k}$ where $\alpha > 0$ is some constant depending only on the proof system.

Computational Soundness. Interactive *arguments* are defined like interactive proofs, but where the soundness of the protocol only holds against computationally bounded provers. Damgård and Pfitzmann [4] show that sequential repetition lowers the error probability of arguments at an exponential rate.

Bellare et al. [2] show that parallel repetition reduces the error of computationally sound protocols with three rounds or less at an exponential rate. On the negative side, they give, for any k, a four round protocol where k-fold parallel repetition does not decrease the error at all. The communication complexity of this protocol is linear in k, which leaves open the possibility that parallel repetition does reduce the error if the communication complexity is not allowed to depend on the number of repetitions. This is a possibility one should consider, as the before-mentioned constant α in Raz's theorem is inverse in the communication complexity of the protocol, and this dependence is necessary [7]. So for protocols where the communication complexity grows linearly in k, parallel repetition does not imply error reduction at all for two-prover two-round proofs. Observing that the four-round protocol of Bellare et al. can be restated as a two-round two-prover protocol (without loosing the property that parallel repetition does not decrease the error), makes the possibility that unbounded communication complexity is necessary here even more likely.

Noticing this possibility, Bellare et al. propose another four-round protocol with fixed communication complexity, which has the property that *relative to an oracle* repeating the protocol any polynomial number of times in parallel, does not decrease the error. This shows that there is no "black-box" error reduction theorem for this protocol. Bellare et al. see this result as evidence that parallel repetition does not decrease the error of computationally sound protocols. Another interpretation of this result could be that parallel repetition does always reduce the error, and the reason why there's no proof of this is that such a proof would require non black-box techniques. We show that under standard assumptions the interpretation of Bellare et al. is indeed correct for protocols with eight rounds or more, and we give much stronger evidence that this is also true for protocols with four rounds.

THE VERIFIER'S SECRET. Except for Section 3, throughout we consider protocols where the verifier holds no secret and thus its strategy is efficiently computable. The reason is that otherwise there are trivial protocols where parallel repetition does not decrease the error as observed by Bellare et al. [2], we extend their observation in Section 3.

1.1 Our Contribution

For n a security parameter, we present the first computationally sound protocol where $k(n)$-fold parallel repetition does not decrease the error for any polynomial $k(\cdot)$. To achieve this we start with the protocol of Bellare et al. whose k-fold parallel repetition does not decrease the error, but we modify it such that k is chosen by the prover (in particular, if the prover has to run the protocol $k(n)$ times in parallel, he can set $k = k(n)$). As in this protocol the the length of the second message from the prover to the verifier is linear in k, we must allow a verifier V_{super} which runs in super-polynomial time, in order for the protocol to work for *any* polynomial $k(\cdot)$. We then transform this protocol into one with a fixed polynomial time verifier V_{poly} using the universal arguments due to Barak and Goldreich [1]. Loosely speaking, the long message is replaced by a hash value, which then is followed by an interactive proof to V_{poly} which shows that V_{super} would have accepted the message. We get the following theorem.

Theorem 1. *There exists an overwhelmingly complete eight round protocol with error probability* $3/4$ *such that* $k(\cdot)$*-fold parallel repetition does not reduce its error probability below* $1/17$ *for any polynomially bounded* $k(\cdot)$*, under the assumption that collision-free family of hash functions and CCA2-secure cryptosystem with respect to superpolynomial adversaries exists.*

Unfortunately, the use of an universal argument increases the round complexity of the protocol from the optimal four to eight.

In Section 5 we propose a new four round protocol relative to an oracle, where $k(n)$-fold parallel repetition does not decrease the error for any polynomial $k(\cdot)$. Unlike the artificial oracle used by Bellare et al., we only need a generic group which potentially can be instantiated with a concrete group satisfying some clearly defined hardness assumptions (basically, it must be hard to compute the inverse of a random element).

More precisely, let $p \in [2^n, 2^{n+1}]$ be a randomly chosen prime, let $\phi' : \mathbb{Z}_p \to [0, 2K-1]$ be a randomly chosen injection and $\phi(x) \stackrel{\text{def}}{=} \phi'(x \bmod p)$ its natural extension to the whole of $\mathbb{Z}$. Then denote by $\mathcal{O}$ the oracle defined by $\mathcal{O}(x) = \phi(x)$ and $\mathcal{O}(X, Y) = \phi(\phi^{-1}(X) + \phi^{-1}(Y))$ if $X, Y \in \phi(\mathbb{Z}_p)$ and $\perp$ otherwise. We prove the following theorem.

Theorem 2. *There exists an overwhelmingly complete four round protocol relative the oracle* $\mathcal{O}$ *with error probability* $1/2 + \mathsf{negl}(n)$ *such that* $k(\cdot)$*-fold parallel repetition does not reduce its error probability below* $1/2 + \mathsf{negl}(n)$ *for any polynomially bounded* $k(\cdot)$*.*

2 Preliminaries

2.1 Notation

We use $\mathbb{Z}$ to denote the integers and $\mathbb{Z}_p$ to denote the integers modulo p. We use log to denote the logarithm in base two. We denote by TM the set of Turing machines. We denote by PT and PT^* the set of uniform and non-uniform polynomial time Turing machines respectively. The corresponding sets of oracle machines are denoted by adding a superscript, e.g. $\mathrm{PT}^{\mathcal{O}}$. We use n to denote the security parameter, and say that a function $\epsilon(n)$ is negligible if for every constant c there exists a constant n_0 such that $\epsilon(n) < n^{-c}$ for $n > n_0$. We use $\mathsf{negl}(n)$ to denote a fixed but unspecified non-negative negligible function. A function $f(n)$ is overwhelming if $1 - f(n)$ is negligible. If $\nu : \mathbb{N} \rightarrow \mathbb{N}$ is a function we denote by PT^*_ν the set of non-uniform Turing machines that executes in time $\nu(n)p(n)$ for some polynomial p. We say that ν is polynomial-time computable if there exists a Turing machine M_ν that on input $x \in \{0,1\}^n$ outputs $\nu(x)$ using at most $p(n)$ steps, for some polynomial p.

We say that a family of hash functions is PT^*_ν-collision-free if it is collision-free with respect to adversaries in PT^*_ν. Similarly, we say that a cryptosystem is PT^*_ν-CCA2-secure, if it is CCA2-secure with respect to adversaries in PT^*_ν.

We denote by $\langle V(x), P(y)\rangle(z)$ the output of V on private input x and common input z after interacting with P on private input y and common input z. We denote by kV the sequential repetition of k copies of V and we denote by V^k the parallel repetition of k copies of V. In both cases identical private and common inputs are given to each instance and the combined verifier accepts if and only if all instances accept.

2.2 Computationally Sound Protocols

We consider the setting introduced in [2]. Two parties, a prover P and a verifier V, are communicating. They are both given an initial context $\lambda \in \{0,1\}^*$ and the length of this string serves as the security parameter. The initial context could be the output of another protocol or some string in a set-up assumption. Since we do not mention λ explicitly below, we replace it by the security parameter in unary representation 1^n, but our results hold in the more general setting.

Both parties are also given a common input x which is generated together with some secret information w by a probabilistic polynomial time instance generator I that is given input 1^n. The secret information w is given to P at the start of the protocol.

2.3 Universal Arguments

Barak and Goldreich [1] introduce the notion of universal arguments as a special variant of Micali's computationally sound proofs [9]. They define the relation $\mathcal{R}_U$ as the set of pairs $((M, x, t), w)$ such that the Turing machine M outputs 1 on input (x, w) within t steps. Denote by $T_M(x, w)$ the number of steps made by M on input (x, w). A key property of their definition is that t is given in binary. We are mainly interested in two properties of universal arguments: (1) the complexity of the verifier depends only on the size of the common input and not on the size of the witness, and (2) the witness used by the prover can be extracted in a weak sense. The actual definition given by Barak and Goldreich [1] is duplicated below.

Definition 1 (Universal Argument). *A universal-argument system is a pair of strategies, denoted (P, V) that satisfies the following properties:*

1. ***Efficient verification.*** *There exists a polynomial p such that for any $y = (M, x, t)$, the total time spent by the probabilistic verifier strategy V, on common input y, is at most $p(|y|)$. (In particular, all messages exchanged in the protocol have length smaller than $p(|y|)$.)*
2. ***Completeness by a relatively efficient prover.*** *For every $((M, x, t), w)$ in $\mathcal{R}_U$ we have $\Pr[(P(w), V)(M, x, t) = 1] = 1$. Furthermore, there exists a polynomial p such that the total time spent by $P(w)$ on common input (M, x, t) is at most $p(T_M(x, w)) \leq p(t)$.*
3. ***Computational soundness.*** *For every polynomial-size circuit family $\{P_n^*\}_{n\in\mathbb{N}}$, and every $(M, x, t) \in \{0,1\}^n \setminus \mathcal{R}_U$ $\Pr[(P_n^*, V)(M, x, t) = 1] < \mu(n)$ for some negligible function $\mu(n)$.*
4. ***Weak proof of knowledge.*** *For every positive polynomial p there exists a positive polynomial p' and a probabilistic polynomial-time oracle machine E such that for every polynomial-size circuit family $\{P_n^*\}_{n\in\mathbb{N}}$, and every sufficiently long $y = (M, x, t) \in \{0,1\}^*$, if $\Pr[(P_n^*, V)(y) = 1] > \frac{1}{p(|y|)}$, then*

$$\Pr_r[\exists w \cap \{0,1\}^t \;\; \forall i \in \{1, \ldots, t\} : (x, w) \in \mathcal{R}_U \wedge E_r^{P_n^*}(y, i) = w_i] > \frac{1}{p'(|y|)} \;\; .$$

Theorem 3 ([1]). *If there exists a family of collision-free hash functions, then there exists universal arguments with 4 rounds.*

3 When the Verifier Holds a Secret

In this section we show that parallel repetition does not decrease the error probability of computationally sound protocols when the verifier gets any private information.

Bellare et al. [2] give the following simple example of such a protocol: The common input is an RSA modulus $N = pq$ and the secret of the verifier is the factors p and q. The verifier flips a coin. If it is heads it gives the factors to the prover and otherwise not. It accepts if the prover's reply is (p, q). An even simpler example is the following one-round protocol: The verifier has a secret bit b, and accepts if the message from the prover is b.

Clearly, parallel repetition does not decrease the error probability for the two protocols above (in fact, for the first protocol it increases), but neither does sequential repetition. This leaves open the interesting possibility that parallel repetition does always decrease the error probability of computationally sound protocols where the verifier can hold a secret, for all protocols where sequential repetition does reduce the error. Below we show that this is not the case by giving a natural (four-round) protocol that when repeated sequentially lowers the error probability, but if repeated in parallel gives error probability essentially one. Here $\mathcal{CS} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ denotes a public key cryptosystem.

Protocol 1 (Don't Do In Parallel (Verifier Holds a Secret)).
Common input: Public key pk.
Private input to both prover and verifier: Private key sk.

1. V chooses $b \in \{0, 1\}$ randomly, computes $B = \mathsf{Enc}_{pk}(b)$, and hands B to P.
2. P chooses $c \in \{0, 1\}$ randomly, computes $C = \mathsf{Enc}_{pk}(c)$, and hands C to V.
3. If $C \neq B$, then V hands $c = \mathsf{Dec}_{sk}(C)$ to P and otherwise $\perp$.
4. P computes $b' = \mathsf{Dec}_{sk}(B)$ and hands b' to V.
5. V accepts if and only if $b = b'$.

The next two propositions are proved in Appendix A for completeness.

Proposition 1 (Single Instance). *The protocol is overwhelmingly complete and has 4 rounds. If the cryptosystem $\mathcal{CS}$ is CCA2-secure, then for every prover $P^* \in \mathrm{PT}^*$:* $\Pr_{(pk,sk),s}[\langle V_s(sk, pk), P^*(pk)\rangle = 1] < \frac{1}{2} + \mathsf{negl}(n)$.

Proposition 2 (Sequential Repetition). *If the cryptosystem $\mathcal{CS}$ is CCA2-secure, then for every polynomially bounded $k(\cdot)$ and every prover $P^* \in \mathrm{PT}^*$:* $\Pr_{(pk,sk),s}[\langle kV_s(sk, pk), P^*(pk)\rangle = 1] < (\frac{1}{2})^k + \mathsf{negl}(n)$.

Proposition 3 (Parallel Repetition). *For every polynomially bounded $k(\cdot)$ there exists a prover $P^* \in \mathrm{PT}$ such that* $\Pr_{(pk,sk),s}[\langle V_s^k(sk, pk), P^*(pk)\rangle = 1] \geq 1 - \mathsf{negl}(n)$.

Proof. The prover P^* does the following. It waits for B_i from V_i. Then it defines $C_i = B_{i+1 \bmod k}$ and hands it to V_i. With overwhelming probability $C_i \neq B_i$, so it is given $b'_{i+1 \bmod k} = \mathsf{Dec}_{sk}(C_i)$ from V_i. Then it returns b'_i to V_i. Thus, with overwhelming probability $b_i = b'_i$, each V_i accepts, and V^k accepts with overwhelming probability as well, since k is polynomial. $\square$

4 When the Verifier Holds No Secret

From now on we consider computationally sound protocols where the verifier holds no secret. In this section we give an eight-round computationally sound protocol where parallel repetition does not decrease the error.

The Example of Bellare et al. Before we give our counter example we recall the counter example given by Bellare et al. [2] on which our example is based. The idea of the protocol is to explicitly allow the prover to make several instances of it dependent if run in parallel.

Protocol 2 (Don't Do In k-Parallel, [2]).
Common input: Public key pk.
Private input to prover: Private key sk.

1. V chooses $b \in \{0,1\}$ and $r \in \{0,1\}^n$ randomly, computes $B = \mathsf{Enc}_{pk}(b,r)$, and sends B to P.
2. P computes $b = \mathsf{Dec}_{sk}(B)$. Then it chooses $b'_i \in \{0,1\}$ and $r'_i \in \{0,1\}^n$ for $i = 1,\ldots,k-1$ randomly under the restriction that $b = \bigoplus_{i=1}^{k-1} b'_i$, computes $C_i = \mathsf{Enc}_{pk}(b'_i, r'_i)$, and hands $(C_1,\ldots,C_{k-1})$ to V.
3. V hands (b,r) to P.
4. P hands $((b'_1,r'_1),\ldots,(b'_{k-1},r'_{k-1}))$ to V.
5. V accepts if $C_i = \mathsf{Enc}_{pk}(b'_i,r'_i)$, $B \neq C_i$, and $\bigoplus_{i=1}^{k-1} b'_i = b$.

We have modified the protocol slightly to be more consistent with our counter example below. In the original the test is $b \neq \bigoplus_{i=1}^{k-1} b'_i$ and this is needed if given a ciphertext the cryptosystem allows construction of a new ciphertext of an identical plaintext. If we require that the cryptosystem used in the protocol is CCA2-secure this is not an issue.

Intuitively, if a single instance of the protocol is run, then a prover without access to sk can only convince the honest verifier with probability $1/2$, since it must commit itself to a guess $\bigoplus_{i=1}^{k-1} b'_i$ of b before receiving (b,r) and the cryptosystem is non-malleable (recall that CCA2-security implies non-malleability). On the other hand, if k instances of the protocol are run in parallel, then the prover can send the tuple $(C_{i,1},\ldots,C_{i,k-1}) = (B_1,\ldots,B_{i-1},B_{i+1},\ldots,B_k)$ to V_i and then either all verifier instances accept or all verifier instances fail, the first event occurring with probability at least $1/2$. If there are fewer than k instances the remaining C_i's can be defined as ciphertexts of zero.

Why the Example is Unsatisfactory. The example requires that the complexity of the verifier in each instance grows linearly with the number of instances. In other words, the example does not imply that k'-parallel repetition of the protocol for $k' > k$ does not lower the error probability.

This deficiency motivated Bellare et al. [2] to consider if there exists any analytical method, i.e, an error-reduction procedure, whereby one can show that the error probability is lowered by the parallel repetition of a protocol. They prove that there exists no such *black-box* error-reduction procedure. Although we agree that this result is a strong indication that there exists no error-reduction procedure at all, it does not preclude the possibility of a non-black-box error-reduction procedure.

4.1 Our Counter Example

The idea of our counter example is to reduce the complexity of the verifier by making the long messages submitted by the prover in Bellare et al's protocol implicit. More precisely, we let the prover choose k on the fly, and hand a hash value of the list of ciphertext $(C_1,\ldots,C_{k-1})$ instead of sending them explicitly.

It also sends a hash value of $(b'_1, r'_1), \ldots, (b'_{k-1}, r'_{k-1})$ instead of sending them explicitly. The problem with this is of course that now the verifier can not perform the original verification. To solve this problem without increasing the complexity of any instance of the verifier the prover proves using universal arguments [1] that it knows correct preimages of the hash values. For technical reasons we replace addition modulo 2 by addition modulo 17. The reader may think of 17 as some constant to be defined in the proof such that the theorem holds.

We assume that there exists a cryptosystem that is chosen ciphertext secure in the sense of Rackoff and Simon [11] against adversaries in PT^*_ν where $\nu(\cdot)$ is a polynomially computable superpolynomial function (the reader can think of $\nu(n)$ as $n^{\log n}$). It should be possible to construct such a scheme from any family of trap-door permutations secure against adversaries in PT^*_ν following Dolev, Dwork, and Naor [5] or Sahai [13], but we are not aware of any explicit proof of this. We also assume the existence of a family of hash functions that is collision-free against adversaries in PT^*_ν.

Denote by $\mathcal{R}_h$ the relation consisting of pairs $((B, H, h, k), (C_1, \ldots, C_{k-1}))$ such that $h = H(C_1, \ldots, C_{k-1})$ and $B \neq C_i$ for $i = 1, \ldots, k-1$. Denote by $\mathcal{R}_a$ the relation consisting of pairs $((pk, H, h, b, a, k), ((b'_1, r'_1), \ldots, (b'_{k-1}, r'_{k-1})))$ such that $b = -\sum_{i=1}^{k-1} b'_i \bmod 17$, $a = H((b'_1, r'_1), \ldots, (b'_{k-1}, r'_{k-1}))$, and

$$h = H(\mathsf{Enc}_{pk}(b'_1, r'_1), \ldots, \mathsf{Enc}_{pk}(b'_{k-1}, r'_{k-1})).$$

Denote by $M_{\mathcal{R}_h}$ a canonical Turing machine that decides $\mathcal{R}_h$ in polynomial time in n and k and correspondingly for $M_{\mathcal{R}_a}$.

Protocol 3 (Don't Do In Parallel).
Common input: Public key pk and collision-free hash function H.
Private input to prover: Private key sk.

1. V chooses $b \in \mathbb{Z}_{17}$ and $r \in \{0,1\}^n$ randomly, computes $B = \mathsf{Enc}_{pk}(b, r)$, and sends B to P.
2. P computes $b' = \mathsf{Dec}_{sk}(B)$. Then it chooses $r' \in \{0,1\}^n$ randomly, computes $C = \mathsf{Enc}_{pk}(b', r')$ and $h = H(C)$, and hands (h, k, t_h) to V, where $k = 1$ and $t_h = T_{M_{\mathcal{R}_h}}((B, H, h, k), C)$.
3. If $k > \nu(n)$ or $t_h > \nu(n)$, then V outputs 0. Otherwise P and V execute a universal argument on common input $y_h = (M_{\mathcal{R}_h}, (B, H, h, k), t_h)$ and private input $w_h = C$ to the prover.
4. If V accepts the universal argument, then it hands (b, r) to P. Otherwise it outputs 0.
5. P computes $a = H(b', r')$ and $t_a = T_{M_{\mathcal{R}_a}}((pk, H, h, b, a, k), (b', r'))$ and hands (a, t_a) to V.
6. If $t_a > \nu(n)$, then V outputs 0. Otherwise P and V execute a universal argument on common input $y_a = (M_{\mathcal{R}_a}, (pk, H, h, b, a, k), t_a)$ and private input $w_a = (b', r')$.
7. If V accepts the universal argument it outputs 1 and otherwise 0.

We stress that k, t_h, and t_a are encoded in binary. Thus, even though the adversary can choose t_h and t_a larger than any polynomial (as they only have to be smaller than the superpolynomial $\nu(n)$), the complexity of the verifier can still be bounded by some fixed polynomial in n as it is polynomial only in n and $\log(\nu(n))$. This means that also k can be larger than any polynomial. This freedom is needed since we do not want to put any fixed polynomial bound on the "width" of the parallel repetition. On the other hand this is what forces us to consider superpolynomial adversaries. The problem is that when reducing soundness of the protocol to breaking the cryptosystem or the collision-freeness of the hash function we need to extract the ciphertexts $C_1, \ldots, C_{k-1}$, but we can not guarantee that a polynomial time adversary can not use implicit such values, which could give a superpolynomial witness during extraction.

Proposition 4 (Single Instance). *The protocol is overwhelmingly complete and has 8 rounds. Let $\nu : \mathbb{N} \to \mathbb{N}$ be a fixed superpolynomial and polynomial-time computable function, let the hash function be PT^*_ν-collision-free, and let $\mathcal{CS}$ be PT^*_ν-CCA2-secure. Then for every prover $P^* \in \mathrm{PT}^*_\nu$ for all sufficiently large n: $\Pr_{(pk,sk),s}[\langle V_s(pk), P^*(pk)\rangle = 1] < \frac{3}{4}$.*

The relation between the constants 3/4 and 17 is essentially that in the reduction we need to "split" the success probability of the adversary twice, giving a factor 1/8, and we need to extract, giving a factor $(3/4)^2$. Thus, the resulting adversary has success probability at least 1/16, which is bigger than 1/17.

Before we prove the above theorem we show that its error probability does not decrease if repeated in parallel. We stress that each instance V_i of the verifier V^k has the same complexity both in terms of computation and communication independently of k.

Proposition 5 (Parallel Repetition). *For every polynomially bounded $k(\cdot)$ there is a prover $P^* \in \mathrm{PT}$ such that $\Pr_{(pk,sk),s}[\langle V^k_s(pk), P^*(pk)\rangle = 1] > \frac{1}{17} - \mathsf{negl}(n)$.*

Proof. We define the prover P^* interacting with V^k, i.e., the parallel repetition of k instances of V, as follows. Given the cryptotexts B_i from all V_i it defines $(C_{i,1}, \ldots, C_{i,k-1}) = (B_1, \ldots, B_{i-1}, B_{i+1} \ldots, B_k)$. Then it executes the first universal argument honestly. When it gets (b_i, r_i) from V_i it defines

$$\begin{aligned}&((b'_{i,1}, r'_{i,1}), \ldots, (b'_{i,k-1}, r'_{i,k-1}))\\ &= ((b_1, r_1), \ldots, (b_{i-1}, r_{i-1}), (b_{i+1}, r_{i+1}), \ldots, (b_k, r_k)) \enspace .\end{aligned}$$

If $\sum_{i=1}^k b_i \neq 0 \bmod 17$ it fails and stops. Otherwise it executes the rest of the protocol honestly. With probability $\frac{1}{17}$ we have $\sum_{i=1}^k b_i = 0 \bmod 17$ and the probability that $B_i = B_j$ for some $i \neq j$ is negligible. Thus, it follows that the prover succeeds at least with probability $\frac{1}{17} - \mathsf{negl}(n)$. □

Proof (Proposition 4). Completeness follows by inspection. Although the naive implementation of the protocol has more than eight rounds, it is easy to see that one can combine the rounds of the universal argument with the main protocol and achieve eight rounds.

Suppose there exists a prover $P^* \in \mathrm{PT}_\nu^*$ with $\Pr_{(pk,sk),s}[\langle V_s(pk), P^*(pk)\rangle = 1] = \delta \geq \frac{3}{4}$ for n in some infinite index set $\mathcal{N}$. Consider the following experiment. The adversary is given a public key pk and a challenge ciphertext $B = \mathsf{Enc}_{pk}(b, r)$ where b is chosen randomly in $\mathbb{Z}_{17}$. Then it may ask any decryption queries except B and then output a guess b' of b. A simple averaging argument implies that if $|\Pr[b' = b] - 1/17|$ is non-negligible, then the cryptosystem is not CCA2-secure.

The CCA2-Adversary. We define an adversary $A \in \mathrm{PT}_\nu^*$ against the above experiment run with the cryptosystem $\mathcal{CS}$ as follows. It accepts a public key pk and a challenge $B = \mathsf{Enc}_{pk}(b, r)$, where b is chosen randomly in $\mathbb{Z}_{17}$. Then it generates a collision-free hash function H and simulates the honest verifier V except that it instructs it to use B instead of generating this ciphertext as in the protocol. If t_h is too large and V outputs 0, then A outputs 0. The simulation proceeds until the first universal argument has been executed. Then A invokes the knowledge extractors of the universal argument to extract $C_1, \ldots, C_{k-1}$ such that $((B, H, h, k), (C_1, \ldots, C_{k-1})) \in \mathcal{R}_h$. More precisely, it tries a random r and computes $(C_1, \ldots, C_{k-1}) = (E_r^{P^*}(y_h, 1), \ldots, E_r^{P^*}(y_h, k-1))$, where $y_h = (M_{\mathcal{R}_h}, (B, H, h, k), t_h)$ and $E_r^{P^*}$ is the extraction algorithm guaranteed by the weak proof of knowledge property of universal arguments. If $w_h = (C_1, \ldots, C_{k-1})$ does not satisfy $(y_h, w_h) \in \mathcal{R}_U$ it tries again with a fresh r. This procedure is repeated at most $g_h(n)$ times, where $g_h(n)$ is a polynomial to be determined in the analysis below. If extraction fails it outputs 0. Otherwise it asks its decryption oracle for $b'_i = \mathsf{Dec}_{sk}(C_i)$ for $i = 1, \ldots, k-1$ and outputs as its guess of b the value $b' = -\sum_{i=1}^{k-1} b'_i \bmod 17$.

We want to show that the CCA2-security of $\mathcal{CS}$ is broken by A, since this contradicts the security of $\mathcal{CS}$. To do that we must argue that extraction succeeds from the first universal argument, but this is not sufficient. The problem is that it is conceivable that the adversary uses one set of ciphertext as a preimage of h in the first universal argument and another set in the second. Intuitively, the collision-freeness of the hash function prohibits this, but we must prove that this is so.

Divide the randomness s used by the verifier into three parts: s_B is used to form B, s_h is used in the first universal argument, and s_a is used in the second universal argument. Denote by S_{good} the set of tuples (H, pk, sk, s_B) such that

$$\Pr_{s_h, s_a}[\langle V_{(s_B, s_h, s_a)}(H, pk), P^*(H, pk)\rangle = 1] \geq \delta/2 \ .$$

An averaging argument implies $\Pr[(H, pk, sk, s_B) \in S_{good}] \geq \delta/2$. Note that the common input $y_h = (M_{\mathcal{R}_h}, (B, H, h, k), t_h)$ is defined by (H, pk, sk, s_B).

Claim 1. For every $f > 0$ there is a polynomial $g_h(n)$ such that the probability that A fails to extract w_h such that $(y_h, w_h) \in \mathcal{R}_U$ on a common input y_h induced by $(H, pk, sk, s_B) \in S_{good}$ is bounded by $\delta 2^{-f}$.

Proof. From the weak proof of knowledge property of a universal argument follows that there exists a positive polynomial $p'(\cdot)$ such that

$$\Pr_r[\exists w_h \cap \{0,1\}^t \ \ \forall i \in \{1,\ldots,t\} : (y_h, w_h) \in \mathcal{R}_U \wedge E_r^{P_n^*}(y_h, i) = w_{h,i}] > \frac{1}{p'(|y_h|)} \ .$$

for common inputs y_h induced by $(H, pk, sk, s_B) \in S_{good}$. Thus, for such common inputs the expected number of repetitions needed to extract a witness is bounded by $p'(|y_h|)$. If we define $g_h(n) = (2^f/\delta)p'(|y_h|)$ it follows from Markov's inequality that extraction fails with probability bounded by $\delta 2^{-f}$ for such inputs. □

We conclude from the union bound that the probability that $(H, pk, sk, s_B) \in S_{good}$ and A succeeds to extract w_h such that $(y_h, w_h) \in \mathcal{R}_U$ is at least $(1/2 - 2^{-f})\delta$. Then we set $c_1 = 1/2 - 2^{-f}$ and note that we by choosing $f > 0$ appropriately may set $c_1 < 1/2$ arbitrarily close to $1/2$.

A Hypothetical Machine. Unfortunately, the above claim says nothing about the probability that the negative sum (modulo 17) of the plaintexts of the extracted $C_1, \ldots, C_{k-1}$ equal the plaintext of B. Intuitively, the problem is that the prover could use one H-preimage of h in the first universal argument and another one in the second, but this should of course never happen due to the collision-freeness of H.

Denote by A_C the machine that simulates A until $C_1, \ldots, C_{k-1}$ are extracted from the first universal argument, or until it outputs 0. Then it chooses s_a randomly and continues the simulation of the interaction of V and P^* until P^* hands (a, t_a) to V. Then it repeatedly, at most $g_a(n)$ times, invokes the extractors of the second universal argument with fresh randomness in the hope to extract $w_a = ((b'_1, r'_1), \ldots, (b'_{k-1}, r'_{k-1}))$ such that $(y_a, w_a) \in \mathcal{R}_U$, and then outputs (w_h, w_a). Otherwise it outputs 0.

Denote by S'_{good} the set of tuples (H, pk, sk, s_B, s_h) such that $(H, pk, sk, s_B) \in S_{good}$ and

$$\Pr_{s_a}[\langle V_{(s_B, s_h, s_a)}(H, pk), P^*(H, pk)\rangle = 1] \geq \delta/4 \ .$$

An averaging argument implies that

$$\Pr_{s_h}[(H, pk, sk, s_B, s_h) \in S'_{good} \mid (H, pk, sk, s_B) \in S_{good}] \geq \delta/4 \ .$$

Claim 2. For every $f' > 0$ there is a polynomial $g_a(n)$ such that the probability that A_C fails to extract w_a such that $(y_a, w_a) \in \mathcal{R}_U$ on a common input y_a induced by $(H, pk, sk, s_B, s_h) \in S'_{good}$ is bounded by $\delta 2^{-f'}$.

Proof. This follows mutatis mutandi from the proof of the previous claim. □

We conclude that the probability that A_C succeeds to extract w_a where $(y_a, w_a) \in \mathcal{R}_U$ conditioned on $(H, pk, sk, s_B) \in S_{good}$ is at least $(1/4 - 2^{-f'})\delta$. We define $c_2 = 1/4 - 2^{-f'}$ and note that we by choosing $f' > 0$ appropriately can set $0 < c_2 < 1/4$ arbitrarily close to $1/4$.

Claim 3. The probability that the output (w_h, w_a) contains a collision for H, i.e., it satisfies $(C_1, \ldots, C_{k-1}) \neq (\mathsf{Enc}_{pk}(b'_1, r'_1), \ldots, \mathsf{Enc}_{pk}(b'_{k-1}, r'_{k-1}))$, conditioned on $(H, pk, sk, s_B) \in S_{good}$ is negligible.

Proof. If this was not the case we could define A'_C as the adversary that takes a description H of a hash function as input and simply simulates A_C and outputs $(C_1, \ldots, C_{k-1})$ and $(\mathsf{Enc}_{pk}(b'_1, r'_1), \ldots, \mathsf{Enc}_{pk}(b'_{k-1}, r'_{k-1}))$. It would break the collision-freeness of H with non-negligible probability. $\square$

Conclusion of Proof of Proposition. From our claims follow that the probability that A_C outputs (w_h, w_a) such that

$$(C_1, \ldots, C_{k-1}) = (\mathsf{Enc}_{pk}(b'_1, r'_1), \ldots, \mathsf{Enc}_{pk}(b'_{k-1}, r'_{k-1}))$$

and $b = -\sum_{i=1}^{k-1} b'_i \bmod 17$ is at least $(c_1\delta)(c_2\delta) - \mathsf{negl}(n) \geq c_3\delta^2 > \frac{1}{16}$, where the constant $0 < c_3 < 1/8$ may be chosen arbitrarily close to $1/8$. This concludes the proof. $\square$

5 Parallel Repetition Relative to a Generic Group

In the previous section we gave – under standard assumptions – an eight-round protocol with constant communication complexity where parallel repetitions does not decrease the error. In this section we give such a protocol with optimal four rounds relative to a generic group oracle.

5.1 The Model

A generic group is a group where the group elements are encoded by random strings. Access to the encoding and the group operation are provided by a public oracle $\mathcal{O}$. This model was put forward by Nechaev [10] and extended by Shoup [14] to prove lower bounds on the running time of the best generic algorithms to solve the discrete logarithm and related problems. An algorithm is called generic, if it does not use the representation of the group elements, for example the baby-step giant-step algorithm for the discrete logarithm problem is generic, but index-calculus is not. Damgård and Koprowski [3] extend this model to groups of unknown order, our model is very similar to theirs, the main difference is that our group oracle does not provide any efficient way to invert elements.[3] For ease of notation we write $N = 2^n$.

The distribution of the group oracle is defined as follows. A random prime p in the range $N < p < 2N$ and a random injection $\phi' : \mathbb{Z}_p \to [0, 2N-1]$

[3] There is no efficient generic algorithm to find the inverse of an element if a large prime divides the (unknown) group order. In [3] the oracle explicitly provides the operation of inverting elements, the reason is that [3] wanted to prove lower bounds on the hardness of a problem in the RSA-group, where there exists an efficient (non-generic) algorithm for inversion (Extended Euclid).

are chosen. Let $\phi(x) \stackrel{\text{def}}{=} \phi'(x \bmod p)$ denote the natural extension of ϕ' to the whole of $\mathbb{Z}$. To find the encoding of an element the oracle is called with a single argument, i.e., we define $\mathcal{O}(x) = \phi(x)$. In addition to providing encodings, the oracle can be called with two arguments from $\phi(\mathbb{Z})$ to find their product, i.e, we define $\mathcal{O}(X, Y) = \phi(\phi^{-1}(X) + \phi^{-1}(Y))$ if $X, Y \in \phi(\mathbb{Z})$ and $\perp$ otherwise. As mentioned above, unlike [3] our oracle does not provide the inverse operation $\phi(-x \bmod p)$ from $\phi(x)$, in fact, for our proof it is necessary that computing $\phi(-x \bmod p)$ given $\phi(x)$ is hard.

We will often have to sample a random element from the range of $\phi(\mathbb{Z})$, unfortunately we cannot efficiently sample a uniformly random one, as we do not know p. We thus use the following observation.

Observation 1. If x is uniformly distributed over $[0, N^2]$, then $\phi(x)$ is statistically close to the uniform distribution over $\phi(\mathbb{Z})$ for every $\mathcal{O}$ with $N < p < 2N$.

We use a polynomial time computable predicate $\tau : [0, 2N-1] \to \{0, 1\}$ such that $|\Pr_{X \in \phi(\mathbb{Z})}[\tau(X) = 1] - 1/2|$ is negligible. A simple way[4] to construct such a predicate is to set $\tau(x) = 1 \iff x > N$. Due to the random choice of ϕ it is not hard to see that it has the required property with overwhelming probability over the choice of ϕ. Below we assume that $\Pr_{X \in \phi(\mathbb{Z})}[\tau(X) = 1] = 1/2$ to simplify the exposition.

5.2 Our Counter Example

We present a protocol which can be seen as an interactive proof that the prover P "knows" the group order p of the group oracle $\mathcal{O}$. If P indeed knows p, he can make the verifier V accept with probability 1.

Protocol 4 (Don't Do In Parallel (Generic Group)).
Common input: A predicate τ.
Private input to prover: A predicate τ and a group order p.

1. $V^{\mathcal{O}}$ chooses $x \in [0, N^2]$ randomly and sends $X = \phi(x)$ to $P^{\mathcal{O}}$.
2. $P^{\mathcal{O}}$ chooses any $y \in [0, 2N - 1]$ which satisfies $\tau(\phi(y)) = 1$, computes $Z = \phi(y - x)$, and sends Z to $V^{\mathcal{O}}$.
3. $V^{\mathcal{O}}$ sends x to $P^{\mathcal{O}}$.
4. $P^{\mathcal{O}}$ sends y to $V^{\mathcal{O}}$.
5. $V^{\mathcal{O}}$ accepts if and only if $\phi(y - x) = Z$ and $\tau(\phi(y)) = 1$.

Note that if the prover computes the messages Z and y as shown in the protocol, then the verifier accepts. In Step 2 the prover can compute $\phi(-x \bmod p) = \phi((p-1)x)$ from X in polynomial time using his knowledge of p.

[4] Here we are using the fact that the representation is random, i.e., our argument is not purely generic. A simple way to avoid this is to use the predicate $\tau'(x) = \tau(\mathrm{PRF}_s(x))$ for some pseudo-random function PRF and public seed s.

Proposition 6 (Single Instance). *The protocol is overwhelmingly complete and has 4 rounds. For every prover $P^{\mathcal{O},*} \in \mathrm{TM}^{\mathcal{O}}$ with total query complexity polynomially bounded in n we have $\Pr[\langle V^{\mathcal{O}}(\tau), P^{\mathcal{O},*}(\tau)\rangle = 1] < \frac{1}{2} + \mathsf{negl}(n)$, where the probability is taken over $\mathcal{O}$, τ, and the internal randomness of $V^{\mathcal{O}}$.*

Before we prove the proposition above we show that parallel repetition fails to reduce the error probability.

Proposition 7 (Parallel Repetition). *For every polynomially bounded $k(\cdot)$ there is a prover $P^{\mathcal{O},*} \in \mathrm{PT}^{\mathcal{O}}$ such that $\Pr[\langle (V^{\mathcal{O}})^k(\tau), P^{\mathcal{O},*}(\tau)\rangle = 1] > \frac{1}{2} - \mathsf{negl}(n)$, where the probability is taken over $\mathcal{O}$, τ, and the internal randomness of $V^{\mathcal{O}}$.*

Proof. The prover $P^{\mathcal{O},*}$ after receiving the messages $X_i = \phi(x_i), 1 \leq i \leq k$, simply computes $Z_i = \phi(\sum_{l\in\{1,\ldots,k\}\setminus\{i\}} x_l)$. Then when it receives $x_1, \ldots, x_k$ it computes $y_1 = \ldots = y_k = \sum_{l=1}^{k} x_l$. Note that Z_i can be computed by repeated queries to $\mathcal{O}$ using only $X_1, \ldots, X_k$. By construction we have $\phi(y_i - x_i) = \phi(\sum_{l=1}^{k} x_l - x_i) = \phi(\sum_{l\in\{1,\ldots,k\}\setminus\{i\}} x_l) = Z_i$ for $i = 1, \ldots, k$. The distribution of $\phi(y_1)$ is statistically close to uniform, and thus $\tau(\phi(y_1)) = 1$ with probability at least $1/2 - \mathsf{negl}(n)$. □

Proof (Proposition 6). Let $Q_0 = X = \phi(x)$ and for $i > 0$ we denote by Q_i the answer to the ith oracle query $P^{\mathcal{O},*}$ makes to $\mathcal{O}$. We define $Q^i = \{Q_0, \ldots, Q_i\}$. Without loss of generality we assume that the replies received by $P^{\mathcal{O},*}$ are either of the form $Q_i = \mathcal{O}(q_i) = \phi(q_i)$ for some query $q_i \in \mathbb{Z}$ or $Q_i = \mathcal{O}(Q_j, Q_k) = \phi(\phi^{-1}(Q_j), \phi^{-1}(Q_k))$ for $j, k < i$. Note that then each reply Q_i is of the form $\phi(a_i + b_i x)$ where $P^{\mathcal{O},*}$ knows $a_i, b_i \in \mathbb{Z}$.[5] Denote by $\ell = \ell(n)$ the polynomial number of oracle queries made by the prover. Without loss we assume that $(a_i, b_i) \neq (a_j, b_j)$ for $i \neq j$, and that $Z \in Q^\ell$. The latter holds, since the probability that $\phi(y - x) = Z$ conditioned on $Z \notin Q^\ell$ is easily bounded by $1/(N - \ell)$. We now prove two claims from which the proposition follows.

Claim 4 (Hard to find multiple of p). For any algorithm $M \in TM^{\mathcal{O}}$ which makes at most $m-1$ oracle queries, each of length at most m bits and where the output is of length at most m bits, we have $\Pr[M^{\mathcal{O}} = v \wedge p \mid v] \in O\left(m^2/N\right)$ (which is negligible for a polynomially bounded m).

Proof. Denote by $\mathcal{P}(N)$ the set of primes in $[N, 2N]$, by the prime number theorem $|\mathcal{P}(N)| = \Theta(N/n)$.

The machine M can choose a sequence $t_1, t_2, \ldots, t_{m-1}$ of values in $\mathbb{Z}$ and ask the oracle for $T_1, T_2, \ldots, T_{m-1}$ where $T_i = \phi(t_i)$. Moreover we allow M an additional mth query which must be its output, i.e. $v = t_m$. The ith oracle query can be either of the form $T_i = \mathcal{O}(t_i)$ or $T_i = \mathcal{O}(T_j, T_k)$ for $j, k < i$ (then $t_i = t_j + t_k$). We can upper bound the size of any t_i as $\log(t_i) \leq 2m$ as

[5] Here "knows" means that one can efficiently extract a_i, b_i given the queries that $P^{\mathcal{O},*}$ makes to $\mathcal{O}$.

follows: if the ith query is of the form $\mathcal{O}(t_i)$ then $\log(t_i) \leq m$ (as no query can be longer than m bits). If the query is of the form $\mathcal{O}(T_j, T_k)$, then $\log(t_i) \leq 1 + \max\{\log(t_j), \log(t_k)\}$, so for any $i \leq m, \log(t_i) \leq m + i \leq 2m$.

Let $t = \prod_{i=1}^{m} t_i$. Then we have $\log(t) \leq \sum_{i=1}^{m} \log(t_i) \leq 2m^2$. So at most $2m^2/n$ primes from $\mathcal{P}(N)$ divide t, and thus also $v = t_m$. The probability that p is one of those primes is at most $(2m^2/n)/|\mathcal{P}(N)| = \Theta(m^2/N)$. □

The following claim is very similar to Theorem 1 in [14].

Claim 5 (x close to uniform). Let γ denote the view of the prover $P^{\mathcal{O}}$ after step 2. Then with overwhelming probability x is statistically close to uniform (over $[0, N^2]$) given γ.

Proof. The view γ contains, for some s, the oracle answers $Q_0, \ldots, Q_s$ and (a_i, b_i) for $i = 1, \ldots, s$. In fact, we prove the slightly stronger statement where p is also contained in γ. Recall that $Q_i = \phi(a_i + xb_i)$ and note that since $Q_0 = \phi(x)$ we have $a_0 = 0, b_0 = 1$. Let $a'_i = a_i \bmod p, b'_i = b_i \bmod p$.

For $(a'_i, b'_i) \neq (a'_j, b'_j)$ we have $\Pr[Q_i = Q_j] \leq 2/p$ as $a_i + b_i x = a_j + b_j x \bmod p$ for at most one x in each interval $t \leq x \leq t + p - 1$. Thus by the union bound, the probability that there is any nontrivial collision, i.e. $Q_i = Q_j$ for some $(a'_i, b'_i) \neq (a'_j, b'_j)$, is at most $\epsilon = s(s-1)/p$. So with overwhelming probability $1 - \epsilon$ there is no nontrivial collision, and conditioned on this event, x is uniformly random over at least a $1 - \epsilon$ fraction of $[0, N^2]$. □

We can now conclude the proof of the proposition, for this we must show that

$$\Pr[\phi(y - x) = Z \wedge \tau(\phi(y)) = 1] - 1/2 < \mathsf{negl}(n) \ .$$

Let $Z = \phi(a_i + b_i x)$ for some a_i, b_i. Then $y = a_i + (b_i + 1)x \bmod p$ when $\phi(y - x) = Z$. By Claim 4 we can assume that $p \nmid (b_i + 1)$. By Claim 5 x is close to uniformly random for the prover at the point where he must choose a_i, b_i, thus $a_i + (b_i + 1)x \bmod p$ is close to uniformly random over $\mathbb{Z}_p$ (as $b_i + 1$ generates $\mathbb{Z}_p$ additively). This implies that $\Pr[\tau(\phi(y)) = 1] - 1/2$ is negligible, since $\Pr[\tau(\phi(u)) = 1]$ is negligibly close to $1/2$ if u is chosen randomly in $[0, N^2]$. □

6 Acknowledgments

We thank Thomas Holenstein for fruitful discussions.

References

1. B. Barak and O. Goldreich. Universal arguments and their applications. *Electronic Colloquium on Computational Complexity (ECCC)*, (093), 2001.
2. M. Bellare, R. Impagliazzo, and M. Naor. Does parallel repetition lower the error in computationally sound protocols? In *38th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 374–383. IEEE Computer Society Press, 1997.

3. I. Damgård and M. Koprowski. Generic lower bounds for root extraction and signature schemes in general groups. In *Advances in Cryptology – Eurocrypt 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 256–271. Springer Verlag, 2002.
4. I. Damgård and B. Pfitzmann. Sequential iteration of interactive arguments and an efficient zero-knowledge argument for np. In *25th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 772–783, 1998.
5. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *23rd ACM Symposium on the Theory of Computing (STOC)*, pages 542–552. ACM Press, 1991.
6. U. Feige. On the success probability of the two provers in one-round proof systems. In *Structure in Complexity Theory Conference*, pages 116–123, 1991.
7. U. Feige and O. Verbitsky. Error reduction by parallel repetition – a negative result. *Combinatorica*, 22(4):461–478, 2002.
8. O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.
9. Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.
10. V.I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
11. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology – Crypto '91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer Verlag, 1991.
12. R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.
13. A. Sahai. Non-malleable non-interactive zero-knowledge and adaptive chosen-ciphertext security. In *40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 543–553. IEEE Computer Society Press, 1999.
14. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology – Eurocrypt '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer Verlag, 1997.

A Omitted Proofs

Proof (Proposition 1). Completeness is clear and the number of rounds follow by counting. Suppose the claim is false, i.e., there exists a prover P^* that succeeds with probability at least $1/2 + n^{-c}$ for n in some infinite index set $\mathcal{N}$. Denote by A the CCA2-adversary that proceeds as follows. It accepts a public key pk, hands the pair of messages $(0, 1)$ to the experiment, and waits for a challenge ciphertext B. Then it starts a simulation of the interaction between V and P^* on the common input pk and using B. If P^* sends $C \neq B$ to the verifier it invokes its decryption oracle to compute $c = \mathsf{Dec}_{sk}(C)$ and hands it back. Finally, it outputs the reply b' of P^* as its guess of the contents of B. It follows that A breaks the CCA2-security of $\mathcal{CS}$, since when the verifier accepts the guess b' equal the content of B. □

Proof (Proposition 2). We use a subscript i with the elements in the ith sequential execution, i.e., we write $(B_i, C_i, c_i, b'_i, b_i)$ for the values in the ith execution.

Denote by E_i the event that the verifier accepts in the ith instance of the protocol. Thus, we have $\Pr[E_i] = \Pr[b'_i = b_i] = \frac{1}{2} + \frac{1}{2}(\Pr[b'_i = 1 \mid b_i = 0] - \Pr[b'_i = 1 \mid b_i = 1])$.

Suppose there exists a constant c, an infinite index set $\mathcal{N}$, and a prover P^* such that $\Pr_{(pk,sk),s}[\langle kV_s(sk,pk), P^*(pk)\rangle = 1] \geq (\frac{1}{2})^k + n^{-c}$ for $n \in \mathcal{N}$ and fix such a security parameter n. Then we have

$$\Pr[E_1]\Pr[E_2 \mid E_1]\Pr[E_3 \mid E_2 \wedge E_1] \cdots \Pr[E_3 \mid \wedge_{i=1}^{k-1} E_i] \geq (1/2)^k + n^{-c} \ .$$

This implies that there exists a fixed l such that $\Pr[E_l \mid \bigwedge_{i=1}^{l-1} E_i] \geq \frac{1}{2} + n^{-c}$. In other words $|\Pr[b'_l = 1 \mid b_l = 0 \wedge \bigwedge_{i=1}^{l-1} E_i] - \Pr[b'_l = 1 \mid b_l = 1 \wedge \bigwedge_{i=1}^{l-1} E_i]| \geq n^{-c}/2$. We clearly also have $\Pr[\wedge_{i=1}^{l-1} E_i] \geq n^{-c}$. Denote by A the adversary that accepts a public key pk and hands the pair of messages $(0,1)$ to the experiment, and waits for a challenge ciphertext B. Then it proceeds as follows:

1. It simulates the interaction between kV and P^* on common input pk. The verifier V_i for $i = 1, \ldots, l-1$ is simulated honestly except that it invokes the decryption oracle to compute $c_i = \mathsf{Dec}_{sk}(C_i)$ if necessary. If any event $\bar{E}_i$ occur for an $1 \leq i \leq l-1$ it halts with output 0.
2. Then it defines $B_l = B$, continues the simulation computing c_l using the decryption oracle if necessary, and outputs the final message b'_l of P^* in the lth instance of the protocol.

By construction A never queries its decryption oracle on $B_l = B$. Thus, it follows that the CCA2-security of $\mathcal{CS}$ is broken. $\square$