J. Cryptol. (2012) 25: 116–135
DOI: 10.1007/s00145-010-9090-x

Journal of
CRYPTOLOGY

Parallel Repetition of Computationally Sound Protocols Revisited*

Krzysztof Pietrzak
CWI, Amsterdam, Netherlands
pietrzak@cwi.nl

Douglas Wikström
KTH, Stockholm, Sweden
dog@kth.se

Communicated by Oded Goldreich

Received 3 May 2010
Online publication 3 November 2010

Abstract. We prove a negative result concerning error reduction by parallel repetition for computationally sound protocols, e.g., interactive arguments. Our main result is a complete and computationally sound eight round interactive argument for which k-fold parallel repetition does not reduce the error below a constant for any polynomial k.

The starting point for our construction is the work of Bellare, Impagliazzo and Naor (FOCS'97). For any fixed k, they construct a four round protocol for which k-fold parallel repetition does not lower the soundness error. The communication complexity of this protocol is linear in k. By using universal arguments due to Barak and Goldreich (CCC 2002), we turn this protocol into an eight-round protocol whose complexity is basically independent of k.

1. Interactive Protocols

1.1. *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 [13]. 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

[*] An extended abstract has appeared in the *4th Theory of Cryptography Conference*, pp. 86–102, 2007.

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.[1]

In general, parallel repetition is more problematic than sequential repetition. For example, parallel repetition does not preserve the zero-knowledge property of a protocol [12], and there are two-prover proofs where running the proof twice in parallel does not decrease the error at all [9]. On the positive side, Raz [22] 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. Clearly $\alpha \leq 1$, a better upper bound is proven in [23].

1.2. *Computational Soundness*

Interactive *arguments*, introduced in [3], are defined like interactive proofs, but the soundness of the protocol only holds against computationally bounded provers. Interactive protocols where the verifier is only secure against *efficient* cheating provers are called *computationally sound* protocols. Sequential composition lowers the error probability of arguments at an exponential rate [7], but also here, the case of parallel repetition is much more intricate.

Parallel Repetition for Arguments Bellare, Impagliazzo and Naor [2] show that, somewhat surprisingly, parallel repetition does not in general reduce the soundness of interactive arguments. More precisely, the authors of [2] construct—for any k—a four round protocol such that k-fold parallel repetition does not decrease the error at all.

The communication complexity of their 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, since the above-mentioned constant α in Raz's theorem is inverse in the communication complexity of the protocol, and this dependence is necessary [10,23]. 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. But it could be that parallel repetition does always reduce the error, and the reason why there is no proof of this is that such a proof would require non-black-box techniques. In this paper, we show that under standard assumptions the interpretation of Bellare et al. is indeed correct for protocols with eight rounds or more, and we strengthen the evidence that this is also true for protocols with four rounds. Before we explain our contribution in more detail, we first summarize the known positive results on parallel repetition of computationally sound protocols.

[1] The parallel case was proven in Appendix C of [11].

Positive Results Complementing their negative results for four-round protocols, Bellare et al. [2] show that parallel repetition reduces the error of computationally sound protocols with three rounds or less at an exponential rate. Impagliazzo et al. [16] give a more general "Chernoff-type" direct product theorem, where they not only show that the prover is unlikely to cheat in all k parallel repetitions, but also that it is even unlikely to cheat in significantly more than in an ϵ fraction of the repetitions, where ϵ is the soundness of a single execution. Canetti et al. [4] give a quantitatively much better reduction for two-round protocols. Pass and Venkitasubramaniam prove a parallel-repetition theorem for *constant-round* public-coin protocols. Håstad et al. [15] give a parallel repetition for a class of computationally sound protocols which as important special cases includes (not necessarily constant-round) public-coin protocols[2] and three-round protocols. Chung and Liu [5] improve upon [15] and prove a tight bound: k-fold repetition reduces the error from δ to δ^k. Haitner [14] considers a different way of doing parallel repetition. He first changes the verifier in protocol at hand: In each round, the verifier can (with some noticeable probability) terminate and accept. Although this increases the error probability of the protocol, Haitner shows that parallel repetition of this modified protocol always reduces the soundness error at an exponential rate.

Verifiers with a Secret Usually, the verifier in an interactive protocol is not supposed to hold any secret information, and so its strategy is efficiently computable. Bellare et al. [2] observe that when considering protocols where the verifier can hold secret information, there exist "trivial" protocols where parallel repetition does not decrease the error. We revisit and strengthen their observation in Sect. 4.

2. Our Contribution

Let n be 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 so 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 $k = k(n)$). In this protocol, the length of the second message from the prover to the verifier is linear in k, and so we are forced to allow the verifier V_{super} to run 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 universal arguments due to Barak and Goldreich [1]. Loosely speaking, the long message is replaced by a hash value of it, which is followed by an interactive proof to V_{poly} that proves that V_{super} would have accepted the message.

This transformed protocol is a computationally sound protocol relative to some public-parameters that must be honestly sampled (the precise result is stated in Theorem 3, Sect. 5). However, an interactive argument [3] does not accept any public parameter as input, and our protocol does not even take a statement as input. In Sect. 5.2, we first eliminate the public parameter (we discuss the issue of public-parameters in more

[2] The counterexamples in [2] and this paper are private-coin protocols. The positive results for public-coin protocols just mentioned show that this is inherent.

detail in Sect. 2.1 below), and then show how to turn the protocol into an interactive argument of any language.

The basic idea for the latter step is to combine our protocol (let us call it Σ_1) with any interactive proof or argument Σ_2. Here we let the prover choose if he wants to run Σ_1 or Σ_2. If Σ_1 is complete, then the combined protocol is a complete argument, and its soundness (of a single run or when run in parallel) is basically the soundness of Σ_2. We get the following theorem.

Theorem 1 (Main theorem). *There exists a complete eight-round interactive argument 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 a collision-free family of hash functions and a non-interactive non-malleable* (*with respect to XOR*) *commitment-schemes with respect to superpolynomial adversaries exist.*

Unfortunately, the use of a universal argument increases the round complexity of the protocol from the optimal four to eight. The question whether parallel repetition lowers the error probability for r-round protocols with $3 < r < 8$ remains open.

Towards answering this question, in Sect. 6 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 $k(\cdot)$. An oracle with this property has already been constructed in [2]. This oracle is quite sophisticated, whereas our oracle is just a generic group and 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). Presently, we are not aware of any suitable candidate group.

More precisely, let $p \in [2^n, 2^{n+1}]$ be a randomly chosen prime, let $\phi' : \mathbb{Z}_p \to [0, 2^{2n} - 1]$ be a randomly chosen injection and $\phi(x) \overset{\text{def}}{=} \phi'(x \bmod p)$ 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 a complete four-round protocol relative to 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* $k(\cdot)$.

2.1. *About the Public Parameter*

As explained above, we first construct a protocol (cf. Theorem 3 in Sect. 5) which requires that a public parameter is available to all parties, and the negative result on parallel repetition only applies if all the instantiations that are run in parallel use the same parameter. This parameter is the public-key of a CCA2-secure cryptosystem, and the protocol is complete in the sense that a prover who knows the corresponding secret-key can make the verifier accept with probability one.

This situation is not quite satisfying as most positive results on error reduction by parallel repetition, including the parallel repetition theorems for three-round protocols of Bellare et al. [2] and for public-coin protocols of Pass et al. [20], do not consider such a public parameter. And, in fact, parallel repetition does not lower the soundness error if we allow such a parameter even for "zero round" protocols: Just consider a random variable Λ, which denotes the initial random parameter such that $\Pr[\Lambda = \lambda] = \epsilon$ for some λ. Now consider a verifier which accepts if $\Lambda = \lambda$. The soundness error here is ϵ, and it does not decrease even if we repeat the "protocol" any number of times (using the same public parameter).

There are two ways to close this conceptual gap. For one thing, we can try to prove the positive results relative to a public parameter. As illustrated by the trivial example above, this is not possible in general. To exclude such pathological counterexamples where it is obvious that repetition (sequential or parallel) is of no use to lower the error, one can state the theorem relative to a fixed public parameter λ, i.e., prove a statement of the form: "if the soundness error for the protocol is $\epsilon = \epsilon(\lambda)$ conditioned on the public parameter being $\Lambda = \lambda$, then k fold parallel repetition (with public parameter λ) lowers the error to…". This then gives a meaningful parallel-repetition theorem for protocols, assuming the distribution of the public-parameter Λ is such that the soundness error ϵ is small (i.e., bounded away from 1) with high probability over the choice of Λ. This clearly is the case for the protocol we construct to prove Theorem 3, where the public parameter is a public-key for a CCA2-secure cryptosystem. Håstad et al. [15] prove a parallel repetition theorem of this kind for public-coin protocols.

The other way to close the conceptual gap is to construct counterexamples without using a public parameter, which is what we do in this work. It is possible to adapt our main Theorem 3 so that no public parameter is required by using a commitment scheme instead of an encryption scheme in the protocol proving our counterexample. To prove the soundness (of a single run of) the protocol, the commitment scheme must be non-malleable with respect to XOR.[3] Panday et al. [19] construct non-interactive non-malleable (with respect to any relation, not just XOR as we need it) commitments, albeit under somewhat non-standard assumptions. Very recently Pass and Wee construct such commitments from any sub-exponentially hard one-way function [21].

3. Preliminaries

3.1. *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 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 $\nu : \mathbb{N} \to \mathbb{N}$ is a function we denote by PT^*_ν the set of non-uniform Turing machines that execute in time $\nu(n)p(n)$ for some polynomial p. We say

[3] More precisely, no efficient adversary A should be able to win the following game with probability more than $1/2 + \mathsf{negl}$. Given a commitment $C = \mathsf{Commit}(b, r)$ for a random bit b, A outputs commitments $C_1, \dots, C_\ell$ with $C_i \neq C$ for all $i = 1, \dots, \ell$. Then A gets b and r, opens $C_i, \dots, C_\ell$ to $b_1, \dots, b_\ell$, and we say that A wins if $b = b_1 \oplus b_2 \oplus \cdots \oplus b_\ell$.

that ν is polynomial-time computable if there exists a Turing machine M_ν that on input $x \in \{0,1\}^*$ outputs $\nu(x)$ in at most $p(|x|)$ steps, for a 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.

3.2. *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.

3.3. *Universal Arguments*

Barak and Goldreich [1] introduce the notion of universal arguments as a special variant of Micali's computationally sound proofs [17]. 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: (i) the complexity of the verifier depends only on the size of the common input and not on the size of the witness, and (ii) 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 $\Pr[\langle V, P(w)\rangle(M,x,t) = 1] = 1$. 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 $(M,x,t) \in \{0,1\}^n \setminus \mathcal{R}_U$: $\Pr[\langle V, P^*_n\rangle(M,x,t) = 1] < \mu(n)$ 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 $\Pr[\langle V, P_n^*\rangle(y) = 1] > \frac{1}{p(|y|)}$, then

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

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

4. When the Verifier Holds a Secret

In this section, we show that parallel repetition does not in general decrease the error probability of computationally sound protocols when the verifier is given 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 decreases the error probability of computationally sound protocols where the verifier holds a secret, for all protocols where sequential repetition reduces 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 $b = b'$.

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

Proof. 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' equals the content of B. □

Proposition 3 (Sequential Repetition of Protocol 1). *If the cryptosystem $\mathcal{CS}$ is CCA2-secure, then for every polynomially bounded $k(\cdot)$ and every prover $P^* \in \mathrm{PT}^*$: $\Pr_{(pk,sk),s}[\langle kV_s(sk, pk), P^*(pk)\rangle = 1] < (\frac{1}{2})^k + \mathsf{negl}(n)$.*

Proof. 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 $\Pr[E_i] = \Pr[b'_i = b_i] = \frac{1}{2} + \frac{1}{2}(\Pr[b'_i = 1 \mid b_i = 1] - \Pr[b'_i = 1 \mid b_i = 0])$.

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

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

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, $|\Pr[b'_l = 1 \mid b_l = 0 \wedge \bigwedge_{i=1}^{l-1} E_i] - \Pr[b'_l = 1 \mid b_l = 1 \wedge \bigwedge_{i=1}^{l-1} E_i]| \geq n^{-c}/2$. 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.

We clearly have $\Pr[\bigwedge_{i=1}^{l-1} E_i] \geq n^{-c}$. Thus, the probability that A completes the interaction with the experiment is non-negligible. By construction, A never queries its decryption oracle on $B_l = B$ except with negligible probability. Thus, it follows that the CCA2-security of $\mathcal{CS}$ is broken. □

Proposition 4 (Parallel Repetition of Protocol 1). *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. □

5. 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 (relative to some public-parameter) where parallel repetition does not decrease the error.

Theorem 3. *There exists a 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 a collision-free family of hash functions and a CCA*2*-secure cryptosystem with respect to superpolynomial adversaries exist.*

The protocol is relative to a public parameter, *namely an honestly sampled public-key of a cryptosystem is known to all parties* (*the protocol can be seen as* "*proof of knowledge*" *of the corresponding secret-key*).

In order to prove Theorem 2, in Sect. 5.2 we turn this protocol into an interactive argument, as explained in the introduction.

The Example of Bellare et al. Before we give our counterexample, we recall the counterexample 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 $i = 1, \dots, k-1$ 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 $(C_1, \dots, C_{k-1})$ to V.
3. V hands (b, r) to P.
4. P hands $((b'_1, r'_1), \dots, (b'_{k-1}, r'_{k-1}))$ 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 counterexample 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 Is the Example 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, whereby one can show that the error probability is lowered by the parallel repetition of a protocol. They prove that there exists no such method that treats the prover interacting with the repeated verifier as a *black-box*. 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.

5.1. *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 cannot 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 [24] 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 [8] or Sahai [25], 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

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

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.

Private input to prover: Private key sk.

1. V samples a collision-resistant hash function H. It chooses $b \in \mathbb{Z}_{17}$ and $r \in \{0,1\}^n$ randomly, computes $B = \mathsf{Enc}_{pk}(b, r)$, and sends B, H 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 cannot guarantee that a polynomial time adversary cannot use implicit such values, which could give a superpolynomial witness during extraction.

Proposition 5 (Single Instance of Protocol 3). *Protocol 3 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: $\Pr_{(pk,sk),s}[\langle V_s(pk), P^*(pk)\rangle = 1] < \frac{3}{4}$.

We postpone the proof of the above proposition for a second. By the following proposition, sequential repetition does lower the error probability of Protocol 3. We only state this proposition to emphasize that the reason that parallel repetition does not lower the error probability of the protocol is not due to some pathological behavior—where the success probability of the prover depends significantly on the choice of the public parameter—as explained in Sect. 2.1.

Proposition 6 (Sequential Repetition of Protocol 3). *Let $\nu : \mathbb{N} \rightarrow \mathbb{N}$ 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 polynomially bounded* $k(\cdot)$ *and every prover* $P^* \in \mathrm{PT}^*$: $\Pr_{(pk,sk),s}[\langle kV_s(pk), P^*(pk)\rangle = 1] < (\frac{1}{2})^k + \mathsf{negl}(n)$.

We omit the proof of this proposition as it is very similar to the proof of Proposition 3.

Proposition 7 (Parallel Repetition of Protocol 3). *For every polynomially bounded* $k(\cdot)$ *there is a prover* $P^* \in \mathrm{PT}$ *such that* $\Pr_{(pk,sk),s}[\langle V_s^k(pk), P^*(pk)\rangle = 1] > \frac{1}{17} - \mathsf{negl}(n)$.

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 $(C_{i,1}, \dots, C_{i,k-1}) = (B_1, \dots, B_{i-1}, B_{i+1} \dots, B_k)$. Then it executes the first universal argument honestly. When it gets (b_i, r_i) from V_i it defines

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

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 of Proposition 5. 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 $C_1, \dots, C_{k-1}$ such that $((B, H, h, k), (C_1, \dots, C_{k-1})) \in \mathcal{R}_h$. More precisely, it tries a random r and computes $(C_1, \dots, C_{k-1}) = (E_r^{P^*}(y_h, 1), \dots, E_r^{P^*}(y_h, k'))$, where k' is the number of bits in $k - 1$ ciphertexts, $y_h = (M_{\mathcal{R}_h}, (B, H, h, k), t_h)$, 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 ciphertexts 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 four parts: s_H is used to sample the hash functions H, 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 (s_H, pk, sk, s_B) such that

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

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

Claim 1. For every constant $f > 0$ 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 $(s_H, pk, sk, s_B) \in S_{\text{good}}$ 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

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

for common inputs y_h induced by $(s_H, pk, sk, s_B) \in S_{\text{good}}$. 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 $(s_H, pk, sk, s_B) \in S_{\text{good}}$, 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 by choosing $f > 0$ appropriately we 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 (s_H, pk, sk, s_B, s_h) such that $(s_H, pk, sk, s_B) \in S_{\text{good}}$ and

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

An averaging argument implies that

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

Claim 2. For every constant $f' > 0$ 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 $(s_H, pk, sk, s_B, s_h) \in S'_{\text{good}}$ is bounded by $\delta 2^{-f'}$.

Proof. This follows mutatis mutandis from the proof of Claim 1. □

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

Proof. If this were 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. □

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

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

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. □

5.2. *Making the Counterexample an Interactive Argument*

Protocol 2 (and thus also Protocol 3 derived from it) can be seen as an interactive proof of knowledge of the secret-key sk corresponding to the public-parameter pk. As discussed in Sect. 2, we would rather have a counterexample which is an interactive argument. To achieve this, we first remove the public parameter by replacing the encryption scheme with a commitment scheme. We then show how to turn the resulting protocol into an interactive argument with basically the same soundness.

Protocol 4 below is basically Protocol 2, but where the encryptions are replaced with commitments.

Protocol 4 (Don't Do in k-Parallel Without Public Parameter).

1. V chooses $b \in \{0,1\}$ and $r \in \{0,1\}^n$ randomly, computes $B = \mathsf{Commit}(b, r)$, and sends B to P.
2. P chooses random $\tilde{b} \in \{0,1\}$, and $b'_i \in \{0,1\}$ and $r'_i \in \{0,1\}^n$ for $i = 1, \ldots, k-1$ randomly under the restriction that $\tilde{b} = \bigoplus_{i=1}^{k-1} b'_i$, computes $C_i = \mathsf{Commit}(b'_i, r'_i)$, 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 $C_i = \mathsf{Commit}(b'_i, r'_i)$, $B \neq C_i$, and $\bigoplus_{i=1}^{k-1} b'_i = b$.

One can show (similarly as we did for Protocol 2 in Sect. 5.1) that the soundness error of a single instance is $1/2 + \mathsf{negl}(n)$ if the commitment scheme is non-malleable. Such commitment schemes have very recently been constructed by Pandey et al. [19], albeit under non-standard assumptions. A cheating prover can win with probability $1/2$ even if this protocol is repeated k times in parallel, again this can be seen exactly as we did for Protocol 2 in Sect. 5.1.

We can derive from Protocol 4 an eight-round protocol—which we denote by $\langle V_T, P_T \rangle$—whose communication complexity is independent of k, exactly as we constructed Protocol 3 from Protocol 4 in the previous section. Analogously to Proposition 5, one can show that for any efficient cheating prover P_T^* we have $\Pr_s[\langle V_T, P_T^* \rangle = 1] < 3/4$, and analogously to Proposition 7, for any polynomial $k(\cdot)$ there exists a prover P_T^* which can make V_T accept with probability $1/17 - \mathsf{negl}(n)$.

In order to prove Theorem 1, we now must explain how to construct an interactive argument whose soundness (of a single run or when run in parallel) is basically the same as of $\langle V_T, P_T \rangle$. For this, let $\langle V_P, P_P(w) \rangle(x)$ be any complete interactive proof (or argument) with an efficient prover P_P and negligible soundness error. Recall that this means that $P_P(w)$, given as input a witness w for $x \in L$, can convince V_P of the fact that $x \in L$. Let $\langle V_C, P_C \rangle(x)$ be the "combined" protocol, where the prover can with the first message decide if he wants to run the protocol $\langle V_T, P_T \rangle$ (i.e., interact with V_T) or rather the protocol $\langle V_P, P_P \rangle(x)$. Here we let the first message from V_C contain the first message from V_T and one from V_P, this way we do not need an extra round where the prover communicates with which verifier he would like to interact. The honest prover $P(w)$ always chooses to run the (complete) proof $\langle V_P, P_P(w) \rangle(x)$; clearly, then also $\langle V_C, P_C(w) \rangle(x)$ is complete. Further, as the soundness error of $\langle V_P, P_P \rangle(x)$ is negligible, the soundness error of $\langle V_C, P_C \rangle(x)$ is the same (up to a negligible additive term) as the soundness error of $\langle V_T, P_T \rangle$.

6. 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 an optimal four rounds relative to a generic group oracle.

6.1. *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 is provided by a public oracle $\mathcal{O}$. This model was put forward by Nechaev [18] and extended by Shoup [26] 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 [6] 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.[4] 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 $\phi' : \mathbb{Z}_p \to [0, N^2 - 1]$ are chosen. Let $\phi(x) \overset{\text{def}}{=} \phi'(x \bmod p)$ 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 [6] 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 use a polynomial time computable predicate $\tau : [0, N^2 - 1] \to \{0, 1\}$ such that $|\Pr_{X \in \phi(\mathbb{Z})}[\tau(X) = 1] - 1/2|$ is negligible. A simple way[5] 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.

6.2. *Our Counterexample*

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.

[4] There is no efficient generic algorithm to find the inverse of an element if a large prime divides the (unknown) group order. In [6], the oracle explicitly provides the operation of inverting elements, the reason is that the authors of [6] 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).

[5] 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.

Protocol 5 (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 $x \in [0, N^2 - 1]$ 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.

Proposition 8 (Single Instance). *The protocol is overwhelmingly complete and has* 4 *rounds. For every prover* $P^{*\mathcal{O}} \in \mathrm{PT}^{*\mathcal{O}}$ *with total query complexity polynomially bounded in* n *we have* $\Pr[\langle V^{\mathcal{O}}, P^{*\mathcal{O}}\rangle(\tau) = 1] < \frac{1}{2} + \mathsf{negl}(n)$, *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 9 (Parallel Repetition). *For every polynomially bounded* $k(\cdot)$ *there is a prover* $P^{*\mathcal{O}} \in \mathrm{PT}^{\mathcal{O}}$ *such that* $\Pr[\langle (V^{\mathcal{O}})^k, P^{*\mathcal{O}}\rangle(\tau) = 1] > \frac{1}{2} - \mathsf{negl}(n)$, *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 $y_1 = \cdots = y_k = \sum_{l=1}^{k} x_l$. 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 of Proposition 8. Without loss of generality, we can forbid queries $\mathcal{O}(X, Y)$ to the oracle using inputs $X, Y \in [0, N^2 - 1]$ that have not been previously received as output from $\mathcal{O}$. The reason is that only a $p/N^2 < 2/N$ fraction of $[0, N^2 - 1]$ is in the range of ϕ, and thus $\mathcal{O}(X, Y)$ will almost certainly output $\perp$ if X and/or Y have not been received as output from $\mathcal{O}$.

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 $Q^i = \{Q_0, \ldots, Q_i\}$.

Note that now each oracle output Q_i is of the form $\phi(a_i + b_i x)$ where $P^{*\mathcal{O}}$ knows $a_i, b_i \in \mathbb{Z}$.[6] Denote by $\ell = \ell(n)$ the polynomial number of oracle queries made by the prover. Without loss of generality, we assume that $(a_i, b_i) \neq (a_j, b_j)$ for $i \neq j$, and that

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

$Z \in Q^\ell$. (Otherwise, Z is most likely not even in the range of ϕ, and thus the verifier who checks $\phi(y-x) = Z$ will certainly reject.) 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 $\Pr[(M^{\mathcal{O}} = v) \wedge (p \mid v)] \in O(m^3/N)$ (which is negligible for a polynomially bounded m).

Proof. Denote by $\mathcal{P}(N)$ the set of primes in $[N, 2N-1]$. By the prime number theorem $|\mathcal{P}(N)| = \Theta(N/n)$.

The ith oracle query of M is either of the form $T_i = \mathcal{O}(t_i)$ (for some $t_i \in \mathbb{Z}$, $\log(t_i) \leq m$) or $T_i = \mathcal{O}(T_j, T_k)$ for $j, k < i$ (then $t_i = t_j + t_k$). So M can learn $T_1 = \mathcal{O}(t_1), \dots, T_{m-1} = \mathcal{O}(t_{m-1})$ where either $\log(t_i) \leq m$ or $t_i = t_j + t_k$ for $j, k < i$. Moreover, we let M do two additional queries, the first query $T_0 = \mathcal{O}(0)$ and the last query $T_m = \mathcal{O}(v)$ must be the output $t_m = v$. We call a sequence $t_0, \dots, t_m$ as above valid.

Note that if $p \mid v$ then $T_m = \phi(v \bmod p) = \phi(0) = T_0$. Thus we can generously upper-bound the probability that $p \mid v$ by the probability that for any $i, j, 0 \leq i < j \leq m$ we have $T_i = T_j$. The reason for upper-bounding the probability of the more general event that M finds a collision (and not just that p divides v) is that now we can without loss of generality assume that M chooses the queries non-adaptively, i.e., M initially outputs a valid sequence $t_0, \dots, t_m$, meaning it wants to make the queries $T_0, \dots, T_m$ where $T_i = \phi(t_i)$.[7] The size of any t_i can be upper-bounded by $\log(t_i) \leq 2m$ as follows: If the ith query is of the form $\mathcal{O}(t_i)$ then $\log(t_i) \leq m$ (as we do not allow queries that are 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 $i \leq m$, $\log(t_i) \leq m + i \leq 2m$.

For any $0 \leq i < j \leq m$ let $d_{ij} = |t_i - t_j|$. Note that $\log(d_{ij}) \leq \max\{t_i, t_j\} \leq 2m$, thus at most $2m/n$ primes from $\mathcal{P}(N)$ divide d_{ij}. The probability that p is one of those primes is at most $(2m/n)/|\mathcal{P}(N)| = \Theta(m/N)$. By the union bound, the probability that p divides any of the d_{ij} (or equivalently, we have a collision $T_i = T_j$) is at most $\Theta(m^3/N)$. □

The following claim is very similar to Theorem 1 in [26].

Claim 5 (x close to uniform). Let γ denote the view of the prover $P^{*\mathcal{O}}$ after Step 2 of Protocol 5. Then with overwhelming probability over γ, the distribution of x conditioned on γ is statistically close to uniform over $[0, N^2 - 1]$.

Proof. The prover initially gets $Q_0 = \phi(x)$, and then can make queries $Q_i = \phi(q_i) = \phi(a_i + xb_i)$. We will say that the prover wins (and in this case just give him x) if

[7] The reason adaptivity does not help is that all M sees, as long as there is no collision, is a sequence $T_0, T_1, \dots$ of uniformly random elements without repetition. To prove that adaptivity does no help, we construct a non-adaptive M' from an adaptive M which has exactly the same probability of finding a collision: M' first runs M answering its oracle queries with random (without repetition) $T_0', T_1', \dots, T_m'$. This run defines a sequence $t_0, \dots, t_m$, which M now uses as its (non-adaptively chosen) queries.

$Q_i = Q_j$ (equivalently, $q_i = q_j \mod p$ or $a_i + b_i x = a_j + b_j x \mod p$) for any $i \neq j$. As in the proof of the previous claim, we can now without loss of generality assume that the queries are chosen non-adaptively and defined by a sequence $(a_1, b_1), \ldots, (a_s, b_s)$ (where each (a_i, b_i) either satisfies $b_i = 0$ or $a_i = a_j + a_k, b_i = b_j + b_k$ for some $0 \leq j \leq k < i$.) As shown in the proof of the previous claim, the probability of a collision where

$$q_i = q_j \mod p \quad \text{but } q_i \neq q_j$$

is negligible (and this is true even if the a_i, b_i can be chosen as functions of x). It remains to prove that a collision where

$$q_i = q_j \quad (\text{or equivalently, } a_i + b_i x = a_j + b_j x)$$

is unlikely. For any fixed $(a_i, b_i) \neq (a_j, b_j)$ the above equation has at most one solution x. As x is uniform over $[0, N^2 - 1]$, the collision probability is bounded by $1/N^2$. Taking the union bound over all pairs of queries $0 \leq i < j \leq m$ the collision probability is at most m^2/N^2. □

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

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

Let $Z = \phi(a_i + b_i x)$ for some a_i, b_i. Then $y = a_i + (b_i + 1)x \mod p$ whenever $\phi(y - x) = Z$. By Claim 4, we can assume that $p \nmid (b_i + 1)$. By Claim 5, we know that x is close to uniformly random for the prover at the point where he must choose a_i, b_i, thus $a_i + (b_i + 1)x \mod p$ 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 $[0, N^2 - 1]$. □

Acknowledgements

We thank Thomas Holenstein for fruitful discussions.

References

[1] B. Barak, O. Goldreich, Universal arguments and their applications. *SIAM J. Comput.* **38**(5), 1661–1694 (2008)

[2] M. Bellare, R. Impagliazzo, M. Naor, Does parallel repetition lower the error in computationally sound protocols? in *38th IEEE Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society, Los Alamitos, 1997), pp. 374–383

[3] G. Brassard, D. Chaum, C. Crépeau, Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.* **37**(2), 156–189 (1988)

[4] R. Canetti, S. Halevi, M. Steiner, Hardness amplification of weakly verifiable puzzles, in *TCC* (2005), pp. 17–33

[5] K.-M. Chung, F.-H. Liu, Parallel repetition theorems for interactive arguments, in *TCC 2010: 7th Theory of Cryptography Conference*, ed. by Daniele Micciancio. Zurich, Switzerland, February 9–11, 2010. Lecture Notes in Computer Science, vol. 5978 (Springer, Berlin, 2010), pp. 19–36

[6] I. Damgård, M. Koprowski, Generic lower bounds for root extraction and signature schemes in general groups, in *Advances in Cryptology—Eurocrypt 2002*. Lecture Notes in Computer Science, vol. 2332 (Springer, Berlin, 2002), pp. 256–271

[7] I. Damgård, 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)* (1998), pp. 772–783

[8] D. Dolev, C. Dwork, M. Naor, Non-malleable cryptography, in *23rd ACM Symposium on the Theory of Computing (STOC)* (ACM, New York, 1991), pp. 542–552

[9] U. Feige, On the success probability of the two provers in one-round proof systems, in *Structure in Complexity Theory Conference* (1991), pp. 116–123

[10] U. Feige, O. Verbitsky, Error reduction by parallel repetition—a negative result. *Combinatorica* **22**(4), 461–478 (2002)

[11] O. Goldreich, *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness* (Springer, New York, 1998)

[12] O. Goldreich, H. Krawczyk, On the composition of zero-knowledge proof systems. *SIAM J. Comput.* **25**(1), 169–192 (1996)

[13] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems. *SIAM J. Comput.* **18**(1), 186–208 (1989)

[14] I. Haitner, A parallel repetition theorem for any interactive argument, in *50th IEEE Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society, Los Alamitos, 2009), pp. 241–250

[15] J. Håstad, R. Pass, D. Wikström, K. Pietrzak, An efficient parallel repetition theorem, in *TCC 2010: 7th Theory of Cryptography Conference*, ed. by D. Micciancio. Zurich, Switzerland, February 9–11, 2010. Lecture Notes in Computer Science, vol. 5978 (Springer, Berlin, 2010), pp. 1–18

[16] R. Impagliazzo, R. Jaiswal, V. Kabanets, Chernoff-type direct product theorems, in *Advances in Cryptology—Crypto 2007* (2007), pp. 500–516

[17] S. Micali, Computationally sound proofs. *SIAM J. Comput.* **30**(4), 1253–1298 (2000)

[18] V.I. Nechaev, Complexity of a determinate algorithm for the discrete logarithm. *Math. Not.* **55**(2), 165–172 (1994)

[19] O. Pandey, R. Pass, V. Vaikuntanathan, Adaptive one-way functions and applications, in *Advances in Cryptology-Crypto 2008*. Lecture Notes in Computer Science, vol. 5157 (Springer, Berlin, 2008), pp. 57–74

[20] R. Pass, M. Venkitasubramaniam, An efficient parallel repetition theorem for Arthur-Merlin games, in *STOC*2007, pp. 420–429

[21] R. Pass, H. Wee, Constant-round non-malleable commitments from sub-exponential one-way functions, in *Advances in Cryptology—Eurocrypt'10*. Lecture Notes in Computer Science (Springer, Berlin, 2010), pp. 638–655

[22] R. Raz, A parallel repetition theorem. *SIAM J. Comput.* **27**(3), 763–803 (1998)

[23] R. Raz, A counterexample to strong parallel repetition, in *49th Annual Symposium on Foundations of Computer Science*. Philadelphia, Pennsylvania, USA, October 25–28, 2008 (IEEE Computer Society, Los Alamitos, 2008), pp. 369–373

[24] C. Rackoff, D. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, in *Advances in Cryptology—Crypto'91*. Lecture Notes in Computer Science, vol. 576 (Springer, Berlin, 1991), pp. 433–444

[25] A. Sahai, Non-malleable non-interactive zero-knowledge and adaptive chosen-ciphertext security, in *40th IEEE Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society, Los Alamitos, 1999), pp. 543–553

[26] V. Shoup, Lower bounds for discrete logarithms and related problems, in *Advances in Cryptology—Eurocrypt'97*. Lecture Notes in Computer Science, vol. 1233 (Springer, Berlin, 1997), pp. 256–266