Efficiency Limitations for Σ-Protocols for Group Homomorphisms *
(Full Version)

Endre Bangerter[1], Jan Camenisch[2], and Stephan Krenn[3]

[1] Bern University of Applied Sciences, Biel-Bienne, Switzerland
endre.bangerter@bfh.ch
[2] IBM Research, Zurich Research Laboratory, Rüschlikon, Switzerland
jca@zurich.ibm.com
[3] Bern University of Applied Sciences, Biel-Bienne, Switzerland, and
University of Fribourg, Fribourg, Switzerland
stephan.krenn@bfh.ch

Abstract. Efficient zero-knowledge proofs of knowledge for group homomorphisms are essential for numerous systems in applied cryptography. Especially, Σ-protocols for proving knowledge of discrete logarithms in known and hidden order groups are of prime importance. Yet, while these proofs can be performed very efficiently within groups of known order, for hidden order groups the respective proofs are far less efficient. This paper shows strong evidence that this efficiency gap cannot be bridged. Namely, whilst there are efficient protocols allowing a prover to cheat only with negligibly small probability in the case of known order groups, we provide strong evidence that for hidden order groups this probability is bounded below by $1/2$ for all efficient Σ-protocols not using common reference strings or the like.
We prove our results for a comprehensive class of Σ-protocols in the generic group model, and further strengthen them by investigating certain instantiations in the plain model.

Keywords. Generic Group Model; Σ-Protocols; Proofs of Knowledge; Error Bounds;

1 Introduction

A *Zero-Knowledge Proof of Knowledge (ZK-PoK)* is a two party protocol between a prover and a verifier enabling the prover to convince the verifier that he knows some secret value, without the verifier being able to learn anything about it. More precisely, in a ZK-PoK an honest prover can always convince the verifier, whilst no malicious prover (i.e., a prover not knowing the secret) can do so with a probability larger than some threshold value, which is called the *knowledge error*.

* This work was partly funded by the European Community's Seventh Framework Programme (FP7) under grant agreement no. 216499.

Fundamental results show that there are ZK-PoK for all languages in $\mathcal{NP}$ [30]. Yet, the respective protocols are of theoretical interest only, because executing them once is either computationally and communicationally too expensive for real world use, or enables the prover to cheat with a high probability. In the latter case, the protocols have to be repeated numerous times to reduce the knowledge error (remember that r repetitions of a ZK-PoK with knowledge error κ result in a protocol with knowledge error κ^r), and thus they become inefficient again.

A (group) homomorphism is a mapping between two groups $\mathcal{G}$ and $\mathcal{H}$ satisfying $\phi(a+b) = \phi(a) \cdot \phi(b)$ for all $a, b \in \mathcal{G}$. Proving knowledge of a preimage under a homomorphism (i.e., of w satisfying $x = \phi(w)$) can often be done very efficiently by using the so-called *Σ^ϕ-protocol* (i.e., the Schnorr [40] or Guillou/Quisquater [31] protocol generalized to arbitrary homomorphisms [4, 18, 34]). This protocol consists of three messages being exchanged: the prover chooses r at random from the domain of the homomorphism, and sends the *commitment* $t := \phi(r)$ to the verifier. The verifier then chooses a random *challenge* c from a predefined challenge set $\mathcal{C}$, and sends it to the prover, who computes its response $s := r + c \cdot w$. The verifier now accepts the proof, if and only if $\phi(s) = x^c \cdot t$. Standard techniques [25] allow to transform this protocol into non-interactive versions or so called signatures of knowledge.

The Σ^ϕ-protocol is a very efficient proof of knowledge for many proof goals existing in cryptography (e.g., knowledge of a discrete logarithm in a known order group, or of the plaintext encrypted in a Paillier ciphertext). The reason is that for the respective homomorphisms, a negligibly small knowledge error can be obtained in a *single run* of the Σ^ϕ-protocol. Yet, the situation is different for the important class of exponentiation homomorphisms with hidden order co-domain (e.g., $\phi(\cdot) : \mathbb{Z} \to \mathbb{Z}_n^* : a \mapsto g^a$, where g is a generator of the quadratic residues modulo n). Such homomorphisms play an important role for many cryptographic applications, e.g. [9, 12, 15, 16, 32, 46, 48, 49], including *Direct Anonymous Attestation (DAA)* [10], and the *identity mixer (idemix)* anonymous credential system [13]. In this case, the Σ^ϕ-protocol is only known to be a PoK with knowledge error $1/2$, and hence must be repeated sequentially to get a sufficiently small knowledge error (e.g., 80 sequential repetitions are required to obtain a knowledge error of $1/2^{80}$). The resulting computational and communicational costs are much too high for many practical applications.

In the recent past a branch of research has tried to overcome the above problem by proposing alternative protocols for exponentiation homomorphisms with hidden order co-domain [4–6, 14, 20, 27]. All these protocols build on a basic idea put forth by Fujisaki and Okamoto [27], and we thus call them *FO-based* henceforth. Unfortunately, none of these FO-based protocols is fully satisfactory - either from a practical or from a theoretical point of view:

- One run of any FO-based protocol is much more expensive than running the Σ^ϕ-protocol once. Moreover, if only standard complexity assumptions (i.e. the Strong RSA Assumption [27]) are made, a recent analysis has revealed that in many cases FO-based protocols are even more expensive than the sequential repetition of the Σ^ϕ-protocol with knowledge error $1/2$ [6].

– The FO-based protocols in [4–6, 20, 27] make use of a common reference string, which is either issued by a trusted third party or generated in an expensive interactive setup phase. Yet, the presence of common reference strings reduces the modularity, and thus increases the complexity of the security analysis of larger applications (as discussed, e.g., in [14, 17, 38]). The security proofs for the protocols in [4, 5] additionally assume the existence of ideal hash functions, and thus only hold true in the random oracle model[4].

Because of these disadvantages, the natural question arises *whether it is necessary to use FO-based protocols at all*? After all, the possibilities of Σ-protocols have not yet been explored thoroughly, and it could be possible that a novel, cleverly designed Σ-protocol or even the existing Σ^ϕ-protocol could be used to overcome the current efficiency limitations. (We note that the latter could be quite possible, if one could find a new knowledge extractor working for the Σ^ϕ-protocol with a suitably chosen challenge set that allows to obtain a small knowledge error in a single execution of the protocol.).

Contribution and Results. In this paper we are aiming at answering this question. We provide ample evidence suggesting that the known minimal knowledge error of the Σ^ϕ-protocol cannot be underrun, neither by a better knowledge extractor for the Σ^ϕ-protocol nor by any other Σ-protocol. In particular, our results indicate that using Σ-protocols the knowledge error of $1/2$ cannot be decreased for exponentiation homomorphisms with hidden order co-domain.

More precisely, we first consider PoK based on Σ-protocols in the generic group model. That is, Σ-protocols where prover, verifier, and knowledge extractor are generic algorithms that can only access the homomorphism and its domain and co-domain through an oracle. We then show that there are lower bounds on the knowledge error for (almost) arbitrary Σ-protocols. These lower bounds on the knowledge error in turn imply efficiency limitations for most possible protocol instances. Roughly, these follow by the fact that a PoK with a large knowledge error needs to repeated sequentially to reduce the knowledge error, which results in a high computational and communicational overhead. Within the generic group model our efficiency analysis shows that the existing Σ^ϕ-protocol is *optimal* and there cannot be another, more efficient Σ-protocol.

We further complement our results by proving lower bounds on the knowledge error of the Σ^ϕ-protocol in the plain model. First, for homomorphisms of the form $w \mapsto w^e$ in RSA groups we show that $1/d$ is a lower bound on the knowledge error, where d is the smallest prime dividing e. Then, we show that for exponentiation homomorphisms with hidden order co-domain, $1/2$ is a lower bound on the knowledge error for all knowledge extractors structurally related to the only one currently known. These results are in accord with those in the generic model and again suggest that the knowledge error that is currently known to be achievable and the associated efficiency limitations cannot be underrun.

[4] For completeness, we note that while the protocol in [14] yields ZK-PoK in the plain model, it is by far too inefficient for practical usage.

Finally, we note that our results do not rule out entirely the possibility to obtain efficient PoK using Σ-protocols. On the one hand, we clearly describe a large number of cases (i.e., instances of Σ-protocols) where this is indeed impossible, indicating that there are inherent efficiency limitations for Σ-protocols. On the other hand, the cases that are not covered by our results also seem to be valuable, since they provide cues for protocol designers on how it could be possible to conceive novel Σ-protocols that overcome current efficiency limitations.

Related Work. Given the abundant usage of Σ-protocols, very little work on their theoretical foundations has been done. Shoup [42] shows that the knowledge error of $1/2$ for homomorphisms of the form $\phi(w) = w^{2^t}$ in RSA groups cannot be improved. One of our results in the plain model extends this to arbitrary exponents. Further, parts of our results are based on unpublished results of one of the authors [4]. Apart from this we are not aware of any other work on efficiency limitations of Σ-protocols. Yet, technically we make use of generic group proof techniques devised by Shoup [42] as well as the extension of these techniques to groups of hidden order by Damgård/Koprowski [22].

The generic group model goes back to [36, 43]. It has been extensively used since then to provide evidence for the security of various cryptographic systems, e.g., [1, 2, 8, 11, 22, 24, 33, 35, 36, 41, 43, 45]. The model is often criticized, because of the risk of lulling a user in a false sense of security. Indeed, there are cases where information only available in the non-generic model (i.e., obtained from encoding specific properties of the group) can be used to break a system which was proved secure in the generic model [23, 26]. Yet, the implications of these observations are different for all the systems cited above, and our results. All these proofs are used to give evidence for the security of a cryptographic system. Thus, if any of them does not hold true in the plain model, the security of the according system can be flawed, resulting in dire consequences for all applications using the respective scheme. In contrast to this, we use the generic group model in a more conservative way. Namely, we show efficiency limitations on the efficiency of a cryptographic primitive. Thus, if our results do not hold true in the non-generic model this means that the efficiency of the scheme can be increased, but the security of the scheme is not affected by any means.

We finally remark that our results do not conflict with those in [19]. The authors there show how to build efficient Σ-protocols for certain exponentiation homomorphisms with hidden order co-domain. Yet, their approach is not generic, but rather uses certain properties of the homomorphism at hand. Further, only very view proofs of practical interest can be performed with their technique.

Structure of this Document. In §2 we recap the basic definitions, and introduce the notion of lower bounds and the class of Σ-protocols for which our results hold true. In §3 we then formulate our main result in the generic group model. This result is strengthened in §4, where we give results in the plain model. We finally conclude and point out some open problems in §5.

2 Preliminaries

In §2.1 we give a short introduction to ZK-PoK and briefly discuss the Σ^ϕ-protocol in §2.2. Then, in §2.3 we introduce the notion of lower bounds on the knowledge error of a protocol. In §2.4 we recap the generic group model we are working in, and finally describe the class of protocols for which our results in the generic group model hold true in §2.5.

2.1 Zero-Knowledge Proofs of Knowledge

After having defined ZK-PoK informally in §1, we next define them formally. We therefore use the widely accepted standard definition of [7, 29], where $(\mathsf{P}(w), \mathsf{V})(x)$ denotes a two party protocol between a prover P and a verifier V with common input x and private input w to P.

Definition 1 (Computational Proof of Knowledge [7, 29]). *A* computational proof of knowledge *for a binary relation $\mathcal{R}$ with knowledge error $\kappa(\cdot) : \mathbb{N} \to [0,1]$ is a two party protocol $(\mathsf{P}(w), \mathsf{V})(x)$, satisfying the following two conditions:*

Completeness: *The verifier always accepts the proof, if $(x, w) \in \mathcal{R}$.*

Soundness: *There exists a polynomial* poly$(\cdot)$*, and a probabilistic algorithm* M *(the* knowledge extractor*) with input x and rewindable black-box access to the prover, such that the following holds true. For every probabilistic polynomial-time (PPT) prover P^* that can make V accept the proof with probability $\varepsilon(x) > \kappa(x)$, M outputs w' satisfying $(x, w') \in R$ in expected time at most*

$$t^+(\varepsilon, \kappa, x) := \frac{\text{poly}(\|x\|)}{\varepsilon(x) - \kappa(\|x\|)},$$

where access to P^* counts as one step only.

The computational aspect of this definition, i.e., the restriction of P^* to be a PPT algorithm, is of importance for our results, as it (almost) allows us to stay in the standard complexity class of PPT algorithms. This issue will also be discussed in §2.3.

A proof of knowledge (PoK) is called *honest verifier zero knowledge (HVZK)*, if no verifier following the protocol is able to gain any information about the secret value w except that it satisfies the stated relation. For a formal description we refer to [29]. There are well known techniques to transform HVZK protocols into protocols which are zero-knowledge also against maliciously behaving verifiers [25].

2.2 The Σ^ϕ-Protocol in Hidden-Order Groups

Most practical applications using ZK-PoK make use of the Σ^ϕ-protocol explained in §1. This allows to prove knowledge of a preimage w of a public value x under

some group homomorphism $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$. If $\phi(\cdot)$ is an exponentiation homomorphism with hidden order co-domain, e.g. $\phi(\cdot) : \mathbb{Z} \to \mathbb{Z}_n^* : a \mapsto g^a$ for some RSA modulus n, the domain of the homomorphism is infinite. To circumvent the problem of drawing random values from an infinite set in P's first step, the random choice $r \in_R \mathcal{G} = \mathbb{Z}$ is substituted by $r \in_R \mathcal{G}'$, where $\mathcal{G}' = \{-\Delta w, \ldots, \Delta w\}$, such that $ord\ \mathcal{H}/\Delta w$ is negligibly small. The rest of the protocol remains unchanged. This approach can be generalized also to the case $\mathcal{G} = \mathbb{Z}^u$ for some integer u. For more details see, e.g., [4, 14].

It is well known that the Σ^ϕ-protocol is a PoK with knowledge error $1/2$ for exponentiation homomorphisms with hidden order co-domain. For homomorphisms with a co-domain of known order v, and power homomorphisms $(w_1, w_2) \mapsto \psi(w_1) \cdot w_2^e$, it is known to have a knowledge error of $1/d$, where d is the smallest prime dividing v, respectively e [18].

2.3 Lower Bounds of the Knowledge Error

Let us now introduce the notion of *lower bounds*, which is a key to our results stated in the following. Intuitively, β is a lower bound of the knowledge error of a protocol, if for this protocol it is not possible to achieve any knowledge error *smaller than or equal to* β:

Definition 2 (Lower Bound). *A function $\beta(\cdot) : \mathbb{N} \to [0, 1]$ is called a* lower bound *on the knowledge error of the protocol (P, V) for a binary relation $\mathcal{R}$, if (P, V) is not a computational proof of knowledge for $\mathcal{R}$ for any $\kappa'(\cdot) : \mathbb{N} \to [0, 1]$ with $\kappa'(\cdot) \leq \beta(\cdot)$.*

An alternative but equivalent characterization is, that $\beta(\cdot)$ is a lower bound, if and only if (P, V) is not a computational PoK with knowledge error $\beta(\cdot)$ for the given relation.

All our results on lower bounds are proven by showing that the conditions of the following theorem are satisfied.

Theorem 3 (Sufficient Conditions for Lower Bounds). *Let (P, V) be a two-party protocol, let $\mathcal{R}$ be a binary relation, and let $\beta(\cdot) : \mathbb{N} \to [0, 1]$ be a function. Then $\beta(\cdot)$ is a lower bound on the knowledge error of (P, V) for $\mathcal{R}$, if the following two conditions are satisfied:*

Uniformity: *There are a polynomial* $\mathrm{poly}(\cdot)$ *and PPT algorithms P^* and D, such that $\varepsilon(x) - \beta(\|x\|) \geq 1/\mathrm{poly}(\|x\|)$ holds for all sufficiently long x generated by D, where $\varepsilon(x)$ is the probability that P^* makes V accept on common input x.*

Hardness: *For all expected PPT algorithms M having rewindable black-box access to P^*, the probability that M outputs a w' with $(x, w') \in \mathcal{R}$ is negligible.*

Proof. By the remark after Def. 2 we need to show that if the conditions hold, then the protocol (P, V) is not a computational PoK for $\mathcal{R}$ with knowledge error $\beta(\cdot)$. We show this by contrapositive, and thus assume that (P, V) is a computational PoK for $\mathcal{R}$ with knowledge error $\beta(\cdot)$.

Hence, by Def. 1, there is a knowledge extractor M for the protocol, which runs in expected time at most $t^+(\varepsilon, \beta, x) := \frac{\text{poly}(\|x\|)}{\varepsilon(x)-\beta(\|x\|)}$, where $\varepsilon(x)$ is the probability that P^* makes V accept on common input x, and poly$(\cdot)$ is an arbitrary but fixed polynomial.

By the uniformity condition there is a P^* such that when x is generated by D, we have $\varepsilon(x) - \beta(\|x\|) \geq 1/(\|x\|)$. This implies that the upper bound $t^+(\varepsilon, \beta, x)$ on the expected running time of M given rewindable black-box access to P^* is a polynomial function in $\|x\|$, i.e., the knowledge extractor M is an expected PPT algorithm. We now immediately see that the hardness property cannot be fulfilled, which concludes our argument by contrapositive. □

From the uniformity condition and Def. 1 it follows that any hypothetical knowledge extractor must be an expected PPT algorithm. This is important, as in our results we show that the hardness condition has to be satisfied by showing that otherwise the respective knowledge extractor could be used to break a cryptographic standard assumption, which are typically defined against PPT attackers. Still, we will have to adopt these assumptions in a natural way. As the standard definition of PoK allows the knowledge extractor to be an *expected* time algorithm [7, 29], we have to generalize the class of attackers the cryptographic assumption holds against to *expected* PPT algorithms as well. Yet, we believe that this generalization is reasonable as by Markov's inequality we see that an expected PPT algorithm may only run super-polynomially long for a small fraction of its executions.

2.4 The Generic Group Model and Groups of Hidden Order

Our main result holds in the generic group model, which we briefly recap next.

The *generic group model* is used to analyze the complexity of problems by considering algorithms in groups whose representation does not reveal any information to the algorithm. That is, such an algorithm must not exploit encoding dependent properties of the group, but may only use operations which are available in arbitrary groups. The hardness of a problem in the generic model is a necessary but not sufficient condition for a problem to be hard in the plain model [23, 26].

Various formalizations of this model have been proposed [3, 33, 36, 43]. They all have in common that an algorithm does not get the concrete group description, but only handles to group elements (e.g., via random encodings [43] or indices to elements [33]). Further, the algorithm gets access to an oracle. To evaluate a group operation, the algorithm gives the handles of elements and the operation to perform to the oracle, which then returns the handle of the result. Similarly, a homomorphism $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$ has to be evaluated by calling an oracle.

We call an algorithm a *generic homomorphism algorithm* for $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$, if, through an oracle $\mathcal{O}^{\phi(\cdot)}$, it might perform the following operations.

$+$: Evaluation of the group operation within $\mathcal{G}$ or $\mathcal{H}$,
$-$: inverting an element within $\mathcal{G}$ or $\mathcal{H}$,
$\stackrel{?}{=}$: testing the equality of two elements from the same group,
$\in_{\mathbf{R}}$: choosing a group element uniformly at random within $\mathcal{G}$ and $\mathcal{H}$, and
$\phi(\cdot)$: evaluating the homomorphism on arbitrary elements $a \in \mathcal{G}$.

When proving our results, we show that any generic algorithm, acting as hypothetical knowledge extractor for a knowledge error smaller than the stated lower bounds, must fail with overwhelming probability. We therefore describe next which operations such an algorithm may perform.

Definition 4 (Generic Black-Box Algorithm). *A* generic black-box algorithm *is a generic homomorphism algorithm for $\phi(\cdot)$ with oracle $\mathcal{O}^{\phi(\cdot)}$, which additionally has rewindable black-box access to P^*. That is, it can (i) execute P^*, (ii) choose the random inputs of P^*, and (iii) repeatedly reset P^*. Resetting P^* does not reset $\mathcal{O}^{\phi(\cdot)}$.*

We remark that the black-box property of such an algorithm is exactly the same as for a knowledge extractor according to Def. 1.

Groups of Hidden Order. In the following we will be interested in group homomorphisms with hidden order co-domain. Intuitively this means that the order of the image of $\phi(\cdot)$ (denoted by Im $\phi(\cdot)$) cannot be computed with non-negligible probability. More precisely, using the formalization of Damgård/Koprowski [22], we let π be the largest prime dividing the order of Im $\phi(\cdot)$, and let $\alpha(\pi)$ denote the maximal probability that π occurs when $\phi(\cdot)$ is chosen randomly from a predefined finite set of homomorphisms. Then $\phi(\cdot)$ is said to have a *hidden order co-domain*, if $\alpha(\pi)$ is negligibly small.

2.5 Generic Σ-Protocols

We call the class of protocols for which our results hold true generic Σ-protocols. Informally, this class consists of almost all HVZK Σ-protocols of the following form. The prover is allowed to compute and send arbitrary elements obtained from generic homomorphism algorithms in both moves. The verifier may send multiple randomly chosen challenges in its first move, and use an arbitrary generic algorithm to decide whether to accept or to reject the proof.

Definition 5 (Generic (Group) Σ-Protocols). *$a_{ij}, b_{ij}, d_i, e_i, f_i, g_i$ be integer coefficients, let $\{(b_{11}, \ldots, b_{1l}), \ldots, (b_{n1}, \ldots, b_{nl})\}$ be linearly independent over the integers, and let $\mathcal{C}_1, \ldots, \mathcal{C}_p \subseteq \mathbb{Z}$ be arbitrary finite sets. Let further $\mathsf{Verify}(\cdot, \ldots, \cdot)$ be a generic homomorphism algorithm, and let the verifier always accept for an honest prover. We then call an HVZK two party protocol a* generic (group) Σ-protocol *for a homomorphism $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$, if it has the form depicted in Fig. 1.*

It can easily be seen that this class covers the existing Σ^{ϕ}-protocol as well as the parallel execution of multiple instantiations thereof. Yet, a much broader set of protocols is covered by the class of generic Σ-protocols.

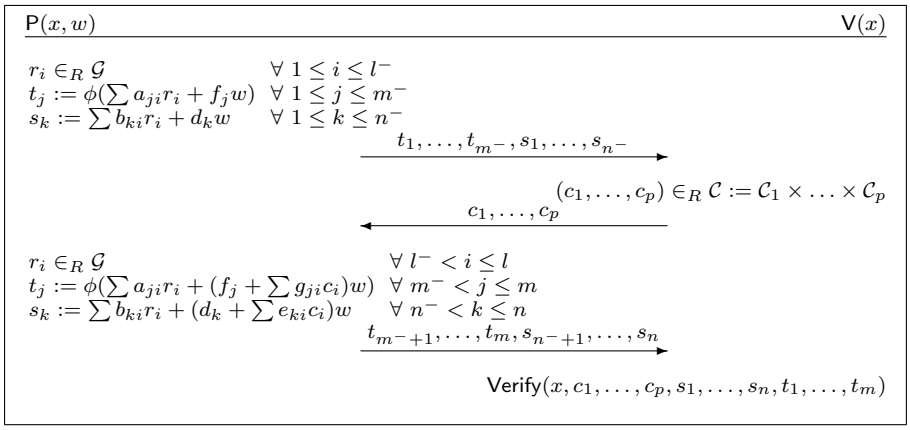

Fig. 1. Structure of a generic Σ-protocol for a homomorphism $\phi : \mathcal{G} \to \mathcal{H}$.

We make two minor remarks on this definition. First, the required linear independence can often be inferred from the HVZK property. Namely, if the vectors were not linearly independent the verifier could compute a multiple of w, and using Shamir's trick [4] could thus often compute the secret. Second, the definition of generic homomorphism algorithms also allows to draw random choices in the co-domain of the homomorphism. The above definition allows to draw random choices in the image by drawing $r \in_R \mathcal{G}$ and computing $\phi(r)$.

3 Efficiency Limitations in the Generic Group Model

In this section we describe lower bounds on the knowledge error for generic Σ-protocols with generic black-box algorithms as knowledge extractors. From these lower bounds we infer efficiency limitations for ZK-PoK using Σ-protocols.

In the statement of our results we refer to the notion of *expected PPT pseudo random functions*. These are defined just as pseudo random functions (cf., e.g., [29]), except for one minor modification. Namely, we require that no *expected* PPT algorithm can distinguish such a function from a truly random function (whereas usually only *strict* PPT distinguishers are considered). See §2.3 for a brief discussion why we resort to expected PPT time assumptions.

We are now ready to formulate our main result in the generic group model.

Theorem 6 (Lower Bounds in the Generic Group Model). *Let be given an arbitrary but fixed polynomial* $\mathrm{poly}(\cdot)$*, a homomorphism* $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$ *with hidden order image, and* $x \in \mathcal{H}$*, for which knowledge of a preimage under* $\phi(\cdot)$ *shall be proven. Consider a generic* Σ*-protocol as in Def. 5, and let* q *equal the number of responses sent by the prover in its second step, i.e.,* $q := n - n^- + m - m^-$*. Assuming that expected PPT pseudo random functions exist, the knowledge*

error of this protocol in the generic group model is lower bounded by

$$\frac{1}{2^{\min(p,q)}} - \frac{1}{\mathrm{poly}(\|x\|)}.$$

Let us briefly discuss the relevance and implications of this result.

- Our result above holds for homomorphisms with a hidden order co-domain, which can be identified in practice, for instance, with exponentiations in hidden RSA groups. As discussed at the onset of this paper, the Σ^ϕ-protocol is known to be a PoK for such homomorphisms with a knowledge error of $1/2$. Our results indicate that this is an inherent limitation for the Σ^ϕ-protocol.
- The best currently known technique to decrease the knowledge error is to sequentially or parallely repeat the Σ^ϕ-protocol. In either case, the number of elements sent by the prover and the verifier in each move increases by the number of repetitions. Our results show that at least for the second and third move, i.e., the challenges sent by V and the responses sent by P, this increase cannot be avoided.
  Put differently, Th. 6 shows that the number p of challenges, and the number q of responses are the key parameters determining the size of the knowledge error. This implies that the strategy of repeating the Σ^ϕ-protocol parallely is optimal concerning the second and third move of the protocol.
- Finally, a protocol designer can deduce from Th. 6 how an alternative for the Σ^ϕ-protocol must not look like. Namely, it must either not be a generic Σ-protocol, or the protocol must have a non-generic knowledge extractor, which uses particulars of the homomorphism.

3.1 Generalization to Other Classes of Homomorphisms

For some classes of homomorphisms used in cryptography there is one more operation that can be performed efficiently. Namely, one can efficiently compute a pseudo-preimage for $\phi(\cdot)$ and x (i.e. a pair $(u, v) \in \mathcal{G} \times \mathbb{Z} \setminus \{0\}$ satisfying $\phi(u) = x^v$). Especially this is the case for power homomorphisms, and homomorphisms with known order co-domain. Such homomorphisms are called *special*.

By allowing the generic homomorphism algorithm to request the oracle to output a pseudo preimage for x under $\phi(\cdot)$, the lower bounds stated above can potentially be decreased. This situation is covered by the following lemma, the proof of which is a straightforward adoption of the proof of Th. 6.

Lemma 7 (Lower Bounds for Special Homomorphisms).

(i) For power homomorphisms $(w_1, w_2) \mapsto \psi(w_1) \cdot w_2^e$ with hidden order co-domain, Th. 6 can be generalized with a lower bound of $\frac{1}{d^{\min(p,q)}} - \frac{1}{\mathrm{poly}(\|x\|)}$, where d is the smallest prime dividing e.

(ii) For arbitrary homomorphisms with a co-domain of known order v, Th. 6 generalizes literally with a lower bound of $\frac{1}{d^{\min(p,q)}} - \frac{1}{\mathrm{poly}(\|x\|)}$, where d is the smallest prime dividing v, if v has a super-polynomially large prime factor.

Note that no such generalization is suitable for exponentiation homomorphisms with hidden order co-domain, as in this case no efficient method to compute a pseudo-preimage is known. Further, analogue observations as for Th. 6 on the implications of this lemma hold.

Examples for homomorphisms falling into Case (i) are the homomorphisms used in the RSA- and Paillier respectively Damgård/Jurik encryption schemes [21, 37, 39, 47]. The according protocol was introduced by Guillou/Quisquater [31]. Case (ii) covers the ElGamal encryption scheme [28], or the protocol proposed by Schnorr [40].

3.2 Proof of Theorem 6

The remainder of this section is now dedicated to proving the theorem. We therefore recap the following lemma introduced by Damgård/Koprowski.

Lemma 8 (Lemma 3 of [22]). *Let $E := a_1X_1 + \ldots + a_uX_u \in \mathbb{Z}[X_1, \ldots, X_u]$ be a non-zero polynomial, and let $z \geq |a_i|$ for all i. Let further $\mathcal{G}$ be a group of hidden order, and $x_1, \ldots, x_u \in_R \mathcal{G}$. For any positive A, we then have*

$$Pr[a_1x_1 + \ldots + a_ux_u = 0] \leq \frac{1}{A} + (\log_2 z + A)\alpha(\pi).$$

Proof (of Theorem 6 – Sketch). The proof is structured as follows. We describe a prover P^* for which we show that it satisfies the conditions of Th. 3. We will see that the uniformity condition holds true by definition. For the hardness condition we simulate the behavior of P^* in the additive subgroup of a suitable polynomial ring. We then estimate the success probability of this simulated game and the error made when making this simulation.

We start with describing a malicious prover P^*.This cheating prover essentially behaves like the honest prover, but it does not answer all challenges, but only certain ones. Depending on whether $p \leq q$ or not, this set (called $\mathcal{C}'$ in the following) is defined as follows:

$p \leq q$: For $i = 1, \ldots, p$, let $\bar{c_i} \in \{0, 1\}$ such that at least half of the elements of $\mathcal{C}_i$ have the same parity as $\bar{c_i}$. Then $\mathcal{C}' := \{(c_1, \ldots, c_p) \in \mathcal{C} | c_i \equiv \bar{c_i} \bmod 2\}$.

$q < p$: We define $\mathcal{C}'$ as a subset of $\mathcal{C}$, which has a cardinality of at least $\#\mathcal{C}/2^q$, and all $(c_1, \ldots, c_p)$, $(c'_1, \ldots, c'_p) \in \mathcal{C}'$ satisfy the following q equations for all $j = m^- + 1, \ldots, m$ and all $k = n^- + 1, \ldots, n$:

$$\sum g_{ji}c_i \equiv \sum g_{ji}c'_i \bmod 2 \quad \text{and} \quad \sum e_{ki}c_i \equiv \sum e_{ki}c'_i \bmod 2$$

Note that such a $\mathcal{C}'$ can easily be constructed from $\mathcal{C}$ by letting $\mathcal{S}_1$ be a subset of cardinality at least $\#\mathcal{C}/2$ for which all tuples yield the same parity when evaluating the first equation. Then, inductively, let $\mathcal{S}_i$ be a subset of $\mathcal{S}_{i-1}$ of cardinality at least $\#\mathcal{S}_{i-1}/2$ such that all tuples have the same cardinality when evaluating the i^{th} equation. Finally, set $\mathcal{C}' := \mathcal{S}_q$.

We next describe P^*. We therefore make the random input $\zeta = (\zeta_1, \ldots, \zeta_l)$ to the prover explicit, and let $\rho(\cdot)$ be a pseudo random function.

(i) It sets $r'_i := \rho(\zeta_i)$ for $i = 1, \ldots, l$, and using these random elements, it behaves just as an honest prover.
(ii) If $c_i \in \mathcal{C}'$, P^* behaves like an honest prover, using $(r'_{l^-+1}, \ldots, r'_l)$ as random elements. Otherwise it halts.

The **uniformity property** of Th. 3 is obviously satisfied, as the prover answers a fraction of at least $1/2^{\min(p,q)}$ of all challenges, and makes the verifier accept (because the verifier would accept for an honest prover).

Let us now turn towards the **hardness property**. We say that a generic black-box algorithm succeeds, if after v steps it outputs the handle corresponding to a preimage of x under $\phi(\cdot)$. Now, instead of letting the knowledge extractor interact with P^* and the oracle $\mathcal{O}^{\phi(\cdot)}$, we play the following game. We substitute $\mathcal{G}, \mathcal{H}$ by the following subgroups of the polynomial rings over the indeterminantes $W, O_{ij}, R_{ij}, T_{ij}$:

$$\mathcal{G}' := \langle W, O_{11}, \ldots, O_{1l}, \ldots, O_{v1}, \ldots, O_{vl}, R_{11}, \ldots, R_{1m}, \ldots, R_{v1}, \ldots, R_{vm} \rangle$$
$$\mathcal{H}' := \langle \mathcal{G}', T_{11}, \ldots, T_{1n}, \ldots, T_{v1}, \ldots, T_{vn} \rangle.$$

Accordingly, the oracle $\mathcal{O}'^{\phi(\cdot)}$ performs its computations within $\mathcal{G}', \mathcal{H}'$.

The prover P^* is adopted as described next. It maintains a list L, which is initially empty, and sets $u := 0$. On random input ζ, it performs the following steps:

(i) For each ζ_i, it checks whether there is a pair $(\zeta_{ji}, \bar{R_{ji}})$ with $\zeta_i = \zeta_{ji}$ in L. If so, it sets $\hat{R}_i := \bar{R_{ji}}$. Otherwise, it increases u by 1 (but at most once in each run), sets $\hat{R}_i := R_{ui}$, and adds $(\zeta_i, \hat{R}_i)$ to L. Then it sends

$$\Big((\sum a_{ji} \cdot \hat{R}_i + f_j \cdot W)_{j=1}^{m^-}, (\sum b_{ki} \cdot \hat{R}_i + d_k \cdot W)_{k=1}^{n^-} \Big)$$

to V. Former are marked as elements of $\mathcal{G}'$, latter as elements of $\mathcal{H}'$.
(ii) If $c_i \in \mathcal{C}'$, P^* analogously computes its response according to the protocol. Otherwise, if $c \notin \mathcal{C}'$, P^* halts.

By $\mathbf{r}$ we denote an element from the set of random choices done by the oracle, and by the generator of the input to the protocol, i.e.,

$$\mathbf{r} \in \Big\{ (\phi(\cdot), x, w, \rho, o, t) \;\Big|\; \phi(\cdot) : \mathcal{G} \to \mathcal{H} \text{ has hidden order co-domain,} \\ x = \phi(w), \rho(\cdot) \text{ pseudo random}, o \in \mathcal{G}^{v \times l}, t \in \mathcal{H}^{v \times m} \Big\}$$

We then define the following two mappings. By $\iota^{\mathbf{r}}_{\mathcal{G}'}(\cdot)$ we denote the evaluation homomorphism from $\mathcal{G}'$ into $\mathcal{G}$. That is, by $\iota^{\mathbf{r}}_{\mathcal{G}'}(E)$ we denote the element in $\mathcal{G}$ which results when all indeterminantes in E are substituted in the following way:

$$W \mapsto w \quad O_{ij} \mapsto o_{ij} \quad R_{ij} \mapsto r'_{ij}.$$

In absolute analogy we let $\iota^{\mathbf{r}}_{\mathcal{H}'}(\cdot)$ be the evaluation homomorphism from $\mathcal{H}'$ into $\mathcal{H}$. That is, the substitution is given by:

$$W \mapsto \phi(w) \quad O_{ij} \mapsto \phi(o_{ij}) \quad R_{ij} \mapsto \phi(r'_{ij}) \quad T_{ij} \mapsto t_{ij}.$$

We observe that for all $E \in \mathcal{G}'$ we have $\phi(\iota^{\mathbf{r}}_{\mathcal{G}'}(E)) = \iota^{\mathbf{r}}_{\mathcal{H}'}(E)$.

During its computation the generic black-box algorithm maintains a list of elements $E_i \in \mathcal{G}'$ respectively $F_i \in \mathcal{H}'$. We say that the algorithm wins this modified game, if one of the following to cases occurs. In case (a), the algorithm finds a preimage of x under $\phi(\cdot)$, while in case (b) there is a pair $i \neq j$ satisfying the following. For a randomly chosen $\mathbf{r}$, we either have $E_i \neq E_j$ and $\iota^{\mathbf{r}}_{\mathcal{G}'}(E_i - E_j) = 0$, or $F_i \neq F_j$ and $\iota^{\mathbf{r}}_{\mathcal{H}'}(F_i - F_j) = 0$.

Observing that the behavior of this game and the actual interaction between the algorithm and the real oracle are indistinguishable as long as the above game is not won, we get that the success probability of the generic black-box algorithm is upper bounded by the probability that the algorithm wins the game [44].

Case (a). Finding a preimage means to compute E_i such that $\phi(\iota^{\mathbf{r}}_{\mathcal{G}'}(E_i)) = x$. Using the observation that we always have $\iota^{\mathbf{r}}_{\mathcal{H}'}(W) = x$ this means to find an E_i such that $\iota^{\mathbf{r}}_{\mathcal{H}'}(E_i - X) = 0$. By introspection of how the E_i are computed, and by using the linear independency of the vectors $\{(b_{11}, \ldots, b_{1l}), \ldots, (b_{n1}, \ldots, b_{nl})\}$, one can show that $W \neq E_i$ for all i. A proof of this claim can be found in App. A.

Let $K := K(\mathcal{C}, a_{ji}, b_{ki}, g_{ji}, e_{ji}, f_j, d_k)$ be an integer such that K is larger than the absolute values of all coefficients occurring in the definition of the examined generic Σ-protocol. Using that $E_i \neq W$ and Lemma 8, and noting that after v oracle queries for every E_i, F_j all coefficients are smaller than $2^v \cdot K$, we get that the probability for (a) is upper bounded by

$$Pr[(a)] \leq \frac{1}{A} + (v + \log_2 K + A)\alpha(\pi) \qquad \text{for all} \qquad A \in \mathbb{Z}.$$

Case (b). Using K as before, and observing that there are at most v different E_i, F_i, we get by a similar argument that the probability for (b) is bounded by

$$Pr[(b)] \leq v^2 \left(\frac{1}{A} + (v + \log_2 K + A)\alpha(\pi) \right) \qquad \text{for all} \qquad A \in \mathbb{Z}.$$

We here assumed that $\phi(\cdot)$ is surjective, and that $\rho(\cdot)$ is a truly random function. The former can easily be seen to be just a technical issue to ease presentation, and the latter yields only a negligible error as $\rho(\cdot)$ is pseudo random by definition.

Demonstration of Hardness Condition. The overall probability that the algorithm wins the game described above is hence limited by

$$Pr[(a)] + Pr[(b)] \leq (v^2 + 1) \left(\frac{1}{A} + (v + \log_2 K + A)\alpha(\pi) \right) \qquad \text{for all} \qquad A \in \mathbb{Z}.$$

for a fixed choice of $\mathbf{r}$. We now set the so far arbitrary value of A to $A := \sqrt{1/\alpha(\pi)}$, such that both, $1/A$ and $A \cdot \alpha(\pi)$ are negligible, and observe that K and $\alpha(\pi)$ are independent from $\mathbf{r}$. Using now that for the hardness condition to be satisfied we only need to consider generic black-box algorithms the expected number v of steps of which is polynomially bounded, and computing the expectation value over all choices of $\mathbf{r}$, we get that the success probability of the generic black-box algorithm is negligible. □

4 Results in the Plain Model

As a special case we have seen in the previous section that there cannot be a generic knowledge extractor underrunning the currently known limitations on the knowledge error of the Σ^ϕ-protocol. This bound is given by $1/2$ for exponentiation homomorphisms with hidden order co-domain, and by $1/d$ for homomorphisms with known order co-domain and power homomorphisms, where d is the smallest divisor of the group order respectively the exponent, cf. §3.1.

Yet, as pointed out in [23, 26], there still is a possibility that these restrictions can be overcome in the plain model. That is, there could be non-generic knowledge extractors, i.e. extractors making use of the concrete description of the homomorphism $\phi(\cdot)$, that improve over the lower bounds shown in §3. Thus, in the following we give further evidence that this is not the case.

The following results are based on the following variant of the Root Assumption [39].

Definition 9 (Root Problem and Expected Root Assumption). *Given a group $\mathcal{H}$ and a uniformly randomly chosen $h \in_R \mathcal{H}$, as well as an integer $e \geq 2$ with $\gcd(\text{ord}\ \mathcal{H}, e) = 1$, the* Root Problem *is to find an $r \in \mathcal{H}$ such that $r^e = h$ holds true. The* Expected Root Assumption *for a group $\mathcal{H}$ says that there is no expected PPT algorithm that solves the Root problem with non-negligible probability.*

In contrast to the standard formulation of the Root Assumption we also require that no *expected* PPT algorithm has a noticeable success probability. Again this requirement naturally arises from the fact that the definition of PoK only restricts the *expected* running time of the knowledge extractor, cf. §2.3.

4.1 Lower Bounds for Power Homomorphisms

We first consider the Σ^ϕ-protocol for power homomorphisms in the plain model. Such homomorphisms underlie, e.g., the RSA encryption scheme [39]. The according instantiation of the Σ^ϕ-protocol was proposed by Guillou/Quisquater [31]. We show that (if possible at all) underrunning the lower bound stated in Lemma 7(i) for such homomorphisms requires a challenge set which is not an integer interval.

In the following result we use the following notation. For a set $\mathcal{S}$ and an integer r, we define $Div\ (\mathcal{S}, r)$ to be all multiples of r within $\mathcal{S}$, i.e. $Div\ (\mathcal{S}, r) := \{s : s \in \mathcal{S}, r|s\}$.

Theorem 10 (Bounds for Power Homomorphisms). *Let $\text{poly}(\cdot)$ be an arbitrary but fixed polynomial. Then for every power homomorphism $\phi_P(\cdot) : \mathcal{H} \to \mathcal{H} : w \mapsto w^e$ with $e \geq 2$, the knowledge error of the Σ^ϕ-protocol for $\phi_P(\cdot)$ is lower bounded by*

$$\max_{2 \leq r \leq e, r|e} \frac{\#\text{Div}\ (\mathcal{C}, r)}{\#\mathcal{C}} - \frac{1}{\text{poly}(\|x\|)},$$

if the Expected Root Assumption is satisfied for $\mathcal{H}$ and $\gcd(e, \text{ord}\ \mathcal{H}) = 1$.

Especially if the challenge set is an integer interval, (i.e. $\mathcal{C} = \{a, \ldots, b\}$ for some $a, b \in \mathbb{Z}$), and d is the smallest prime dividing e, the knowledge error is bounded below by

$$\frac{1}{d} - \frac{1}{\text{poly}(\|x\|)}.$$

Note here that, if $\mathcal{H}$ is an RSA group, i.e. $\mathcal{H} = \mathbb{Z}_n^*$ for a composite modulus n of unknown factorization, the condition $\gcd(e, ord\ \mathcal{H}) = 1$ is always satisfied.

Proof. We prove the theorem by defining a prover and showing that this satisfies the conditions of Th. 3.

Let $2 \leq f \leq e$ be such that $\#Div\ (\mathcal{C}, f) = \max \#Div\ (\mathcal{C}, r)$, where the maximum is taken over all r satisfying $2 \leq r \leq e$ and $r|e$. Let further $x := u^{e/f}$ for an arbitrary element u. The malicious prover P^* now takes u as private input, and performs the following steps:

(i) It chooses $r \in_R \mathcal{H}$, sets $t := \phi_P(r) = r^e$ and sends t to V.
(ii) If the received challenge c satisfies $c \in Div\ (\mathcal{C}, f)$, the prover sets $s := r \cdot u^{c/f}$ and sends s to V. Otherwise P^* halts.

For the **uniformity condition** see that for all $c \in Div\ (\mathcal{C}, f)$ we have $\phi_P(s) = \phi_P(r \cdot u^{c/f}) = t \cdot u^{ce/f} = t \cdot x^c$. Thus, with the maximum taken over the same values of r as before, P^* succeeds with probability $\max \#Div\ (\mathcal{C}, r)/\#\mathcal{C}$.

For the **hardness condition**, assume by contradiction, that there is an expected PPT algorithm M with black-box access to P^* that succeeds in computing an w such that $\phi_P(w) = w^e = x$ with non-negligible probability. Then, by definition of x, we also have $u^{e/f} = w^e$. Because of $\gcd(e, ord\ \mathcal{H}) = 1$ this implies $u = w^f$, and we have computed a f^{th} root of u with non-negligible probability. This contradicts the expected Root assumption, cf. Def. 9. □

Our result is a generalization of that by Shoup [42], where only the case $e = 2^t$ is considered. On the one hand, if the challenge set $\mathcal{C}$ is an integer interval, the theorem implies a lower bound which is equal to the smallest knowledge error that is currently known to be achievable. Albeit, if all elements of $\mathcal{C}$ are co-prime (e.g., if all elements of $\mathcal{C}$ are primes), the theorem becomes meaningless, as it implies a lower bound of 0. Though, the result is still relevant when seen in connection with Th. 6. Namely, whilst latter states that any hypothetical knowledge extractor has to use encoding specific properties of the homomorphism $\phi_P(\cdot)$, the above theorem further restricts the situations where the generic result could potentially be violated in the plain model. In total, the existence of an extractor underrunning the limitation of $1/d$ seems unlikely.

4.2 Lower Bounds for Exponentiation Homomorphisms

For exponentiation homomorphisms $\phi_E(\cdot) : \mathcal{G} \to \mathcal{H} : w \mapsto h^w$, with hidden order co-domain $\mathcal{H}$, the Σ^ϕ-protocol is only known to be a PoK with knowledge error $1/2$. In this section we show that any knowledge extractor achieving a smaller

knowledge error in this case would have to be structurally different from the only knowledge extractor currently known.

This standard knowledge extractor works as described next. In a first phase, it is given black-box access to the prover, and extracts a pseudo preimage (u, v), i.e. a pair satisfying $v \neq 0$ and $x^v = \phi_E(u)$, cf. §3.1, [18]. Then, in a second phase in which the extractor does not have access to the prover any more, it computes a preimage of x from this pseudo preimage. We call knowledge extractors working this way *pseudo preimage based*.

In the formalization of the next result we use the following notation: for a set $\mathcal{S}$ of integers, we write *Diff* $(\mathcal{S})$ for the set of all possible absolute values of differences between different elements of $\mathcal{S}$, i.e. *Diff* $(\mathcal{S}) := \{|s_1 - s_2| : s_1 \neq s_2 \in \mathcal{S}\}$. We further say that an integer d and a set $\mathcal{S}$ are co-prime, if $\gcd(d, s) = 1$ for all $s \in \mathcal{S}$.

Theorem 11 (Bounds for Exponentiation Homomorphisms). *Let* $\mathrm{poly}(\cdot)$ *be an arbitrary but fixed polynomial. Then for every exponentiation homomorphisms* $\phi_E(\cdot) : \mathbb{Z} \to \mathcal{H}' : w \mapsto h^w$*, with* $h \in \mathcal{H}'$*, the knowledge error of the* Σ^ϕ*-protocol for* $\phi_E(\cdot)$ *is lower bounded by*

$$\frac{1}{2} - \frac{1}{\mathrm{poly}(\|x\|)},$$

against pseudo preimage based knowledge extractors, if the following conditions are satisfied. The co-domain $\mathcal{H}'$ *is a large subgroup of* $\mathcal{H}$ *(i.e.* $\#\mathcal{H}'/\#\mathcal{H}$ *is not negligible), the Expected Root Assumption is satisfied for* $\mathcal{H}$*, and* $\mathrm{ord}\ \mathcal{H}'$ *and* $\mathrm{Diff}\ (\mathcal{C})$ *are co-prime.*

We remark that this result can straightforwardly be generalized to homomorphisms of the form $\phi_M(\cdot) : \mathcal{G}^r \to \mathcal{H} : (w_1, \ldots, w_r) \mapsto h_1^{w_1} \ldots h_r^{w_r}$.

Proof (Sketch). The proof works by showing that the conditions of Th. 3 are satisfied. Because of space limitations, we here only describe the required malicious prover, and the main idea how the proof works. For describing the prover P^*, let $\bar{c} \in \{0, 1\}$ be such that at least half of the elements in $\mathcal{C}$ have the same parity as $\bar{c}$, and $\mathcal{G}' = \{-\Delta w, \ldots, \Delta w\}$, such that *ord* $\mathcal{H}'/\Delta w$ is negligible. Then P^*, upon input x, w and *ord* h, performs the following steps.

(i) It chooses $r \in_R \mathcal{G}'$, sets $t := \phi_E(r) = h^r$ and sends t to V.
(ii) If the received challenge c has the same parity as $\bar{c}$, the prover sets $s := r + c \cdot x + \frac{c+\bar{c}}{2} \cdot \mathit{ord}\ h$ and sends s to V. Otherwise, P^* halts.

The **uniformity condition** can directly be seen by noting that $h^s = h^{r+c \cdot x}$ for all challenges answered by P^*.

For the **hardness condition**, the proof follows the following reasoning. Under the expected Root assumption it is hard to compute a preimage of x from any pseudo preimage (u, v) not satisfying $v \mid u$ [5, 16, 20] (note here that we assumed the domain of $\phi_E(\cdot)$ to be $\mathbb{Z}$). We show that all pseudo preimages that can be extracted from P^* satisfy $(c_1 - c_2) \nmid (s_1 - s_2)$, and hence the knowledge extractor fails in its second phase. □

In practice the conditions of this theorem will always be satisfied. If, e.g., $\mathcal{H} = \mathbb{Z}_n^*$ for a safe RSA modulus n, i.e. $n = (2p+1)\cdot(2q+1)$, where $p, q, (2p+1), (2q+1)$ are prime, then $\mathcal{H}'$ is usually given by the set of quadratic residues modulo n. We thus have $\#\mathcal{H}'/\#\mathcal{H} = 1/4$. Further, *ord* $\mathcal{H}' = p \cdot q$, and hence any challenge set $\mathcal{C}$ only containing elements smaller than p, q will satisfy the condition of *Diff* $(\mathcal{C})$ and *ord* $\mathcal{H}'$ being co-prime.

Although this result only considers pseudo preimage based knowledge extractors, it is still relevant for the following reason. Together with the results in the generic group model in §3, Th. 11 implies that a knowledge extractor for exponentiation homomorphisms with hidden order co-domain must neither be generic nor pseudo preimage based. Thus, if possible at all, substantially new insights were required to underrun the restriction of $1/2$ in this case. We seriously doubt the existence of such an extractor, and consequently also that for reaching a small knowledge error in the case of exponentiation homomorphisms with hidden order co-domain, either running the Σ^ϕ-protocol sequentially, or employing an FO-based protocol can be avoided.

5 Conclusion

We have introduced the class of generic Σ-protocols, and have shown that for exponentiation homomorphisms with hidden order co-domain a knowledge error of $1/2^n$ (where n is the minimum of the number of challenges and responses sent in the protocol) is inherent to any of these protocols in the generic group model. We further generalized this result to other classes of homomorphisms as well, especially covering the homomorphisms underlying various crypto systems such as [21, 28, 37, 39, 47].

Besides showing that the currently known limitations on the knowledge error for the Σ^ϕ-protocol are tight in the generic model, our results also give new insights in how these restrictions could be overcome. Namely, any Σ-protocol overcoming these bounds must either be substantially different from the Σ^ϕ-protocol (i.e., it must not be a generic Σ-protocol), or it must have a non-generic knowledge extractor. To us, the former seems to be hard to achieve without using auxiliary constructions resulting from a common reference string as done in [4, 6, 20], because the class of generic Σ-protocols does not leave much design options for other Σ-protocols to look like. On the other hand, the latter also is unlikely, because our results in the generic model were further strengthened by results in the plain model. Thus, although being riddled with various limitations from a theoretical points of view, FO-based protocols [4–6, 14, 20, 27] using common reference strings seem to be inevitable for many real systems.

References

1. M. Abe and S. Fehr. Perfect NIZK with adaptive soundness. In S. Vadhan, editor, *Theory of Cryptography – TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 118–136. Springer Verlag, 2007.

2. D. Aggarwal and U. Maurer. Breaking RSA generically is equivalent to factoring. In A. Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 36–53. Springer Verlag, 2009.
3. L. Babai and E. Szemerédi. On the complexity of matrix group problems I. *IEEE Symposium on Foundations of Computer Science – FOCS 84*, pages 229–240, 1984.
4. E. Bangerter. *Efficient Zero-Knowledge Proofs of Knowledge for Homomorphisms*. PhD thesis, Ruhr-University Bochum, 2005.
5. E. Bangerter, J. Camenisch, and U. Maurer. Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In S. Vaudenay, editor, *International Workshop on Practice and Theory in Public-Key Cryptography – PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 154–171. Springer Verlag, 2005.
6. E. Bangerter, S. Krenn, A.-R. Sadeghi, T. Schneider, and J.-K. Tsay. Design and implementation of efficient zero-knowledge proofs of knowledge. In *SPEED-CC 2009*, 2009.
7. M. Bellare and O. Goldreich. On defining proofs of knowledge. In E. F. Brickell, editor, *Advances in Cryptology – CRYPTO 92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer Verlag, 1993.
8. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer Verlag, 2005.
9. F. Boudot. Efficient proofs that a committed number lies in an interval. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer Verlag, 2000.
10. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In V. Atluri, M. Backes, D. A. Basin, and M. Waidner, editors, *ACM Conference on Computer and Communications Security – CCS 2004*, pages 132–145. ACM Press, 2004.
11. D. Brown. Generic groups, collision resistance, and ECDSA. Cryptology ePrint Archive, Report 2002/026, 2002.
12. J. Camenisch. Better privacy for trusted computing platforms: (extended abstract). In P. Samarati, P. Ryan, D. Gollmann, and R. Molva, editors, *European Symposium on Research in Computer Security – ESORICS 2007*, volume 3193 of *Lecture Notes in Computer Science*, pages 73–88. Springer Verlag, 2004.
13. J. Camenisch and E. V. Herreweghen. Design and implementation of the idemix anonymous credential system. In V. Atluri, editor, *ACM Conference on Computer and Communications Security – CCS 2002*, pages 21–30. ACM Press, 2002.
14. J. Camenisch, A. Kiayias, and M. Yung. On the portability of Generalized Schnorr Proofs. In A. Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 425–442. Springer Verlag, 2009.
15. J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122. Springer Verlag, 1999.
16. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer Verlag, 2003.
17. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004. Preliminary version in STOC, 1998.

18. R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and University of Amsterdam, 1997.
19. R. Cramer and I. Damgård. On the amortized complexity of zero-knowledge protocols. In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 177–191. Springer Verlag, 2009.
20. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 77–85. Springer Verlag, 2002.
21. I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *International Workshop on Practice and Theory in Public-Key Cryptography – PKC 2001*, Lecture Notes in Computer Science, pages 119–136. Springer Verlag, 2001.
22. I. Damgård and M. Koprowski. Generic lower bounds for root extraction and signature schemes in general groups. In L. R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 256–271. Springer Verlag, 2002.
23. A. W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 100–109. Springer Verlag, 2002.
24. A. W. Dent. The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156, 2006.
25. A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO 86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Verlag, 1987.
26. M. Fischlin. A note on security proofs in the generic model. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 458–469. Springer Verlag, 2000.
27. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. Kaliski, editor, *Advances in Cryptology – CRYPTO 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer Verlag, 1997.
28. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology – CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer Verlag, 1985.
29. O. Goldreich. *Foundations of Cryptography – Basic Tools*. Cambridge University Press, 2001.
30. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991. Preliminary version in 27th FOCS, 1986.
31. L. Guillou and J.-J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO 88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer Verlag, 1990.
32. H. Lipmaa. On diophantine complexity and statistical zeroknowledge arguments. In C.-S. Laih, editor, *Advances in Cryptology – ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*. Springer Verlag, 2003.

33. U. Maurer. Index search, discrete logarithms, and Diffie-Hellman. In *Number-theoretic cryptography workshop*. Mathematical Sciences Research Institute, Berkeley, October 2000.
34. U. Maurer. Unifying zero-knowledge proofs of knowledge. In B. Preneel, editor, *Progress in Cryptology – AFRICACRYPT 2009*, volume 5580 of *Lecture Notes in Computer Science*, pages 272–286. Springer Verlag, 2009.
35. U. Maurer and S. Wolf. Lower bounds on generic algorithms in groups. In K. Nyberg, editor, *Advances in Cryptology – EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 72–84. Springer Verlag, 1998.
36. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994. Translated from Matematicheskie Zametki, 55(2):91–101, 1994.
37. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Verlag, 1999.
38. R. Pass. On deniability in the common reference string and random oracle model. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 316–337. Springer Verlag, 2003.
39. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
40. C. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
41. C. Schnorr and M. Jakobsson. Security of signed elgamal encryption. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 73–89. Springer Verlag, 2000.
42. V. Shoup. On the security of a practical identification scheme. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 344–353. Springer Verlag, 1996.
43. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT 97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer Verlag, 1997.
44. V. Shoup. OAEP reconsidered. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259. Springer Verlag, 2001.
45. N. P. Smart. The exact security of ECIES in the generic group model. In B. Honary, editor, *8th International Conference on Cryptography and Coding – IMA 2001*, volume 2260 of *Lecture Notes in Computer Science*, pages 73–84. Springer Verlag, 2001.
46. D. X. Song. Practical forward secure group signature schemes. In *ACM Conference on Computer and Communications Security – CCS 2001*, pages 225–234. ACM Press, 2001.
47. T. Takagi. Fast RSA-type cryptosystem modulo $p^k q$. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO 98*, volume 1462 of *Lecture Notes in Computer Science*, pages 318–326. Springer Verlag, 1998.
48. C. Tang, Z. Liu, and M. Wang. A verifiable secret sharing scheme with statistical zero-knowledge. Cryptology ePrint Archive, Report 2003/222, 2003.
49. P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. In C. Cachin and J. Camenisch, editors, *Progress in Cryptology – INDOCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 384–398. Springer Verlag, 2004.

A Proof of Case (a) in Theorem 6

We here only consider the case that $(\zeta_1, \ldots, \zeta_l) \neq (\zeta'_1, \ldots, \zeta'_l)$ always implies that $\zeta_i \neq \zeta'_j$ for all i, j. If this is not the case, the proof can still be adopted. Yet, this is accompanied by technical issues which primarily complicate the presentation of the proof. We hence exclude this case in the following.

We first note that each E_i is a linear combination of all elements received by the verifier by communicating with the oracle, or the malicious prover. Further, we let $c_1, \ldots, c_k$ be the challenges answered by P^*. Using q_{ab} to denote the coefficient of the secret in the computation of s_a for challenge c_b, we have, for some $j \leq v$ and integer coefficients u_{abc}, u'_{ab},

$$\begin{aligned} E_i = \sum_{\iota=1}^{j} \Big(& u_{\iota 11}(\sum b_{1i}R_{\iota i} + q_{11}W) + \ldots + u_{\iota n1}(\sum b_{ni}R_{\iota i} + q_{n1}W) + \ldots + \\ & u_{\iota 1k}(\sum b_{1i}R_{\iota i} + q_{11}W) + \ldots + u_{\iota nk}(\sum b_{ni}R_{\iota i} + q_{n1}W)\Big) + \\ & u'_{11}O_{11} + \ldots + u'_{vl}O_{vl} \end{aligned} \tag{1}$$

For this to be equal to W, the coefficients of all other indeterminantes have to be equal to zero. Thus, we have for each $R_{\iota i}$ that

$$\begin{aligned} 0 &= u_{\iota 11}b_{1i} + \ldots + u_{\iota n1}b_{ni} + \ldots + u_{\iota 1k}b_{1i} + \ldots + u_{\iota nk}b_{ni} \\ &= (u_{\iota 11} + \ldots + u_{\iota 1k})b_{1i} + \ldots + (u_{\iota n1} + \ldots + u_{\iota nk})b_{ni}. \end{aligned}$$

Using the linear independency pf the vectors $\{(b_{11}, \ldots, b_{1l}), \ldots, (b_{n1}, \ldots, b_{nl})\}$, this can only hold true for a fixed ι and all $i = 1, \ldots, l$, if

$$(u_{\iota 11} + \ldots + u_{\iota 1k}) = \ldots = (u_{\iota n1} + \ldots + u_{\iota nk}) = 0. \tag{2}$$

If $W = E_i$, the coefficients of W in (1) had to sum up to one. Thus we had:

$$1 = \sum_{\nu=1}^{n} \Big((u_{1\nu 1}q_{\nu 1} + \ldots + u_{1\nu k}q_{\nu k}) + \ldots + (u_{j\nu 1}q_{\nu 1} + \ldots + u_{j\nu k}q_{\nu k}) \Big). \tag{3}$$

Yet, by (2), we can write (3) in the following way:

$$\begin{aligned} 1 &= \sum_{\nu=1}^{n} \Bigg(\Big((-u_{1\nu 2} - \ldots - u_{1\nu k})q_{\nu 1} + u_{1\nu 2}q_{\nu 2} + \ldots + u_{1\nu k}q_{\nu k} \Big) + \ldots + \\ & \qquad \Big((-u_{j\nu 2} - \ldots - u_{j\nu k})q_{\nu 1} + u_{j\nu 2}q_{\nu 2} + \ldots + u_{j\nu k}q_{\nu k} \Big) \Bigg) \\ &= \sum_{\nu=1}^{n} \Big((u_{1\nu 2} + \ldots + u_{j\nu 2})(q_{\nu 2} - q_{\nu 1}) + \ldots + (u_{1\nu k} + \ldots + u_{j\nu k})(q_{\nu k} - q_{\nu 1}) \Big). \end{aligned}$$

By construction of P^*, or more precisely by the set of challenges which it responses to, we now can infer that $q_{\nu 1} \equiv \ldots \equiv q_{\nu k} \pmod 2$ for all $\nu = 1, \ldots, n$. Thus, the right hand side of the last equation is equal to zero modulo 2, and consequently (3) cannot be satisfied, and we get $W \neq E_i$. □