# Efficiency Limitations for $\Sigma$-Protocols for Group Homomorphisms *

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, 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, $\Sigma$-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, while 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 $\Sigma$-protocols not using common reference strings or the like.
We prove our results for a comprehensive class of $\Sigma$-protocols in the generic group model, and further strengthen them by investigating certain instantiations in the plain model.

**Keywords.** Generic Group Model; $\Sigma$-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, while no malicious prover (not knowing the secret) can do so with a probability larger than some threshold value (the *knowledge error*).

Fundamental results show that there are ZK-PoK for all languages in $\mathcal{NP}$ [2]. Yet, the respective protocols are of theoretical interest only, because executing

* This work was in part funded by the European Community's Seventh Framework Programme (FP7) under grant agreements no. 216499 and 216483. We also refer to the full version of this paper [1].

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 $\kappa$ result in a protocol with knowledge error $\kappa^r$), 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 $\Sigma^\phi$*-protocol* (i.e., the Schnorr [3] or Guillou/Quisquater [4] protocol generalized to arbitrary homomorphisms [5–7]). 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 [8] allow one to transform this protocol into non-interactive versions or so called signatures of knowledge.

The $\Sigma^\phi$-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 $\Sigma^\phi$-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., [9–16], including *Direct Anonymous Attestation (DAA)* [17], and the *identity mixer (idemix)* anonymous credential system [18]. In this case, the $\Sigma^\phi$-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.

A number of authors have tried to overcome the above problem by proposing alternative protocols for exponentiation homomorphisms with hidden order co-domain [5, 19–23]. All these protocols build on a basic idea put forth by Fujisaki and Okamoto [22], and we thus call them *FO-based* henceforth. Unfortunately, none of these FO-based protocols is fully satisfactory, neither from a practical nor from a theoretical point of view:

- One run of any FO-based protocol is much more expensive than running the $\Sigma^\phi$-protocol once. Moreover, if only standard complexity assumptions (i.e. the Strong RSA Assumption [22]) are made, a recent analysis has revealed that in many cases FO-based protocols are even more expensive than the sequential repetition of the $\Sigma^\phi$-protocol with knowledge error $1/2$ [20].
- The FO-based protocols in [5, 19–22] make use of a common reference string, which is either issued by a trusted third party or generated in an expensive in-

teractive 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 [23–25]). The security proofs for the protocols in [5, 19] 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 $\Sigma$-protocols have not yet been explored thoroughly, and it could be possible that a novel, cleverly designed $\Sigma$-protocol or even the existing $\Sigma^\phi$-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 $\Sigma^\phi$-protocol with a suitably chosen challenge set that allows one 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 $\Sigma^\phi$-protocol cannot be underrun, neither by a better knowledge extractor for the $\Sigma^\phi$-protocol nor by any other $\Sigma$-protocol. In particular, our results indicate that using $\Sigma$-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 $\Sigma$-protocols in the generic group model. That is, $\Sigma$-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 $\Sigma$-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 be 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 $\Sigma^\phi$-protocol is *optimal* and there cannot be another, more efficient $\Sigma$-protocol.

We further complement our results by proving lower bounds on the knowledge error of the $\Sigma^\phi$-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 divisor of $e$. 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.

Finally, we note that our results do not rule out entirely the possibility to obtain efficient PoK using $\Sigma$-protocols. On the one hand, we describe a large number of cases (i.e., instances of $\Sigma$-protocols) where this is indeed impossible,

[4] For completeness, we note that while the protocol in [23] yields ZK-PoK in the plain model, it is by far too inefficient for practical usage.

indicating that there are inherent efficiency limitations for $\Sigma$-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 $\Sigma$-protocols that overcome current efficiency limitations.

**Related Work.** Given the abundant usage of $\Sigma$-protocols, very little work on their theoretical foundations has been done. Shoup [26] 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 [5]. Apart from this we are not aware of any other work on efficiency limitations of $\Sigma$-protocols. Yet, technically we make use of generic group proof techniques devised by Shoup [26] as well as the extension of these techniques to groups of hidden order by Damgård/Koprowski [27].

The generic group model goes back to Nechaev and Shoup [28, 29]. It has been extensively used since then to provide evidence for the security of various cryptographic systems, e.g., [27–38]. 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 plain 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 [39, 40]. Yet, the implications of these observations are different for all the systems cited above than for our results. All the proofs in the former case 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 plain 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 [41]. The authors there show how to build efficient $\Sigma$-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 few 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 $\Sigma$-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 $\Sigma^\phi$-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 We recall the widely accepted definition of zero-knowledge proofs of knowledge (ZK-PoK) [42, 43]. We by $(\mathsf{P}(w), \mathsf{V})(x)$ denote a two party protocol between a prover $\mathsf{P}$ and a verifier $\mathsf{V}$ with common input $x$ and private input $w$ to $\mathsf{P}$.

**Definition 1 (Computational Proof of Knowledge [42, 43]).** *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* $\mathrm{poly}(\cdot)$*, and a probabilistic algorithm* $\mathsf{M}$ *(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 $\mathsf{P}^*$ that can make $\mathsf{V}$ accept the proof with probability $\varepsilon(x) > \kappa(x)$, $\mathsf{M}$ outputs $w'$ satisfying $(x, w') \in R$ in expected time at most*

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

*where access to $\mathsf{P}^*$ counts as one step only.*

The computational aspect of this definition, i.e., the restriction of $\mathsf{P}^*$ 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 [43]. There are well known techniques to transform HVZK protocols into protocols which are zero-knowledge also against maliciously behaving verifiers [8].

### 2.2 The $\Sigma^\phi$-Protocol in Hidden-Order Groups

Most practical applications using ZK-PoK make use of the $\Sigma^\phi$-protocol explained in §1. This allows one 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 $\mathsf{P}$'s first step, the random choice $r \in_R \mathcal{G} = \mathbb{Z}$ is substituted by $r \in_R \mathcal{G}' = \{-\Delta w, \ldots, \Delta w\}$

with $(\Delta w - \operatorname{ord} \mathcal{G})/\operatorname{ord} \mathcal{G}$ being 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., [5, 23].

It is well known that the $\Sigma^\phi$-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 $(w_1, w_2) \mapsto \psi(w_1) \cdot w_2^e$, the protocol is known to have a knowledge error of $1/d$, where $d$ is the smallest prime dividing $v$ in the former, respectively $e$ in the latter case [6].

### 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, $\beta$ 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* $\beta$:

**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 $(\mathsf{P}, \mathsf{V})$ for a binary relation $\mathcal{R}$, if $(\mathsf{P}, \mathsf{V})$ 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 of $\beta(\cdot)$ being a lower bound if and only if $(\mathsf{P}, \mathsf{V})$ 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 $(\mathsf{P}, \mathsf{V})$ 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 $(\mathsf{P}, \mathsf{V})$ for $\mathcal{R}$, if the following two conditions are satisfied:*

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

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

From the uniformity condition and Definition 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 is 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 [42, 43], 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 is restricted to only use group operations. 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 [39, 40].

Various formalizations of this model have been proposed [28, 29, 35, 44]. 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 [29] or indices to elements [35]). Further, the algorithm gets access to an oracle. To evaluate a group operation, the algorithm inputs 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 through 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 $\mathsf{P}^*$. That is, it can (i) execute $\mathsf{P}^*$, (ii) choose the random inputs of $\mathsf{P}^*$, and (iii) repeatedly reset $\mathsf{P}^*$. Resetting $\mathsf{P}^*$ 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 Definition 1.

**Groups of Hidden Order.** In the following we will be interested in group homomorphisms with hidden order co-domain (resp., image). Intuitively this means

that the order of the co-domain (image of $\phi(\cdot)$, denoted by Im $\phi(\cdot)$) cannot be computed with non-negligible probability. More precisely, using the formalization of Damgård/Koprowski [27], we let $\pi$ be the largest prime dividing the order of the co-domain (Im $\phi(\cdot)$), and let $\alpha(\pi)$ denote the maximal probability that $\pi$ 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 (image)*, if $\alpha(\pi)$ is negligibly small.

### 2.5 Generic $\Sigma$-Protocols

We call the class of protocols for which our results hold true generic $\Sigma$-protocols. Informally, this class consists of almost all HVZK $\Sigma$-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) $\Sigma$-Protocols).** *Let $a_{ij}, b_{ij}, d_i, e_i, f_i, g_i$ 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) $\Sigma$-protocol *for a homomorphism $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$, if it has the form depicted in Fig. 1.*

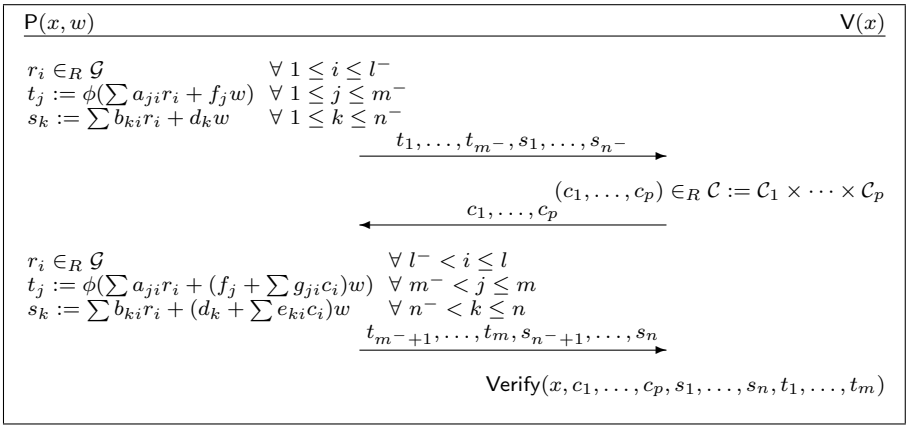


**Fig. 1.** Structure of a generic $\Sigma$-protocol for a homomorphism $\phi : \mathcal{G} \to \mathcal{H}$.

It can easily be seen that this class covers the existing $\Sigma^\phi$-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 $\Sigma$-protocols.

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 [5] 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 $\Sigma$-protocols with generic black-box algorithms as knowledge extractors. From these lower bounds we infer efficiency limitations for ZK-PoK using $\Sigma$-protocols.

In the statement of our results we refer to the notion of *expected PPT pseudo random functions*. Such functions are defined just as pseudo random functions (cf., e.g., [43]), except for one minor modification. Namely, we require that no *expected* PPT algorithm can distinguish such a function from a truly random one (usually one considers only *strict* PPT distinguishers). 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*[5]*, and* $x \in \mathcal{H}$*, for which knowledge of a preimage under* $\phi(\cdot)$ *shall be proven. Consider a generic* $\Sigma$*-protocol as in Definition 5, and let* $q$ *be 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 results indicates that a knowledge error of $1/2$ is an inherent limitation of the $\Sigma^\phi$-protocol for homomorphisms with hidden order co-domain, which especially cover exponentiation homomorphisms in RSA groups.
- The best known technique to decrease the knowledge error is to repeat the $\Sigma^\phi$-protocol, sequentially or in parallel. In either case, the number of elements sent by the prover and the verifier increases by the number of repetitions. Our results show that at least for the second and third move, i.e., the challenges sent by $\mathsf{V}$ and the responses sent by $\mathsf{P}$, this growth cannot be avoided.

[5] Note that this is a stronger requirement than the requirement that the co-domain has hidden order. Yet, typically these two properties accompany each other.

Put differently, Theorem 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 $\Sigma^\phi$-protocol parallely is optimal concerning the second and third move of the protocol.
– Finally, a protocol designer can deduce from Theorem 6 how an alternative for the $\Sigma^\phi$-protocol must not look like. Namely, it must either not be a generic $\Sigma$-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

So far we have considered homomorphisms with hidden order co-domain. Yet, in practice this information is sometimes available and could potentially be used to decrease the lower bounds on the knowledge error.

More generally, we thus consider the class of *special* homomorphisms next. A homomorphisms $\phi(\cdot) : \mathcal{G} \to \mathcal{H}$ is called special, if for every $x \in \mathcal{H}$ a pair $(u, v) \in \mathcal{G} \times \mathbb{Z} \setminus \{0\}$ satisfying $\phi(u) = x^v$ can be computed efficiently. The pair $(u, v)$ is called *pseudo-preimage* of $x$ under $\phi(\cdot)$. Besides homomorphisms with known order co-domain, also power homomorphisms are known to be special.

We model this property by adding one more query to the oracle $\mathcal{O}^{\phi(\cdot)}$, i.e., we allow a generic homomorphism algorithm to request a pseudo-preimage under $\phi(\cdot)$ for arbitrary elements from the co-domain of the homomorphism. We then obtain the following lemma:

**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, Theorem 6 can be generalized to 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 $v$, Theorem 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 they are not known to be special. Analogue observations as for Theorem 6 on the implications of this lemma hold. Especially, in the generic group model the known knowledge error of the $\Sigma^\phi$-protocol cannot be underrun for special homomorphisms.

Examples for homomorphisms of Case (i) are those used in the RSA, Paillier, and Damgård/Jurik encryption schemes [45–48]. The ZK-PoK for these homomorphisms was introduced by Guillou/Quisquater [4]. Case (ii) covers the homomorphisms underlying the ElGamal encryption scheme [49] and the ZK-PoK for it was proposed by Schnorr [3].

### 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 [27]).** *Let $E := a_1X_1 + \cdots + a_uX_u \in \mathbb{Z}[X_1, \ldots, X_u]$ 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*

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

*Proof (of Theorem 6 – Sketch).* The proof is structured as follows. We describe a prover $\mathsf{P}^*$ for which we show that it satisfies the conditions of Theorem 3. We will see that the uniformity condition holds true by definition. For the hardness condition we simulate the behavior of $\mathsf{P}^*$ 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 $\mathsf{P}^*$. This cheating prover essentially behaves like the honest prover, except that it does not answer all challenges but only certain ones. Depending on whether $p \leq q$ or not, the set $\mathcal{C}'$ of answered challenges 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 $\mathcal{C}' := \{(c_1, \ldots, c_p) \in \mathcal{C} | c_i \equiv \bar{c_i} \mod 2\}$.

**$q < p$:** 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 $k = n^- + 1, \ldots, n$:

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

We next describe $\mathsf{P}^*$. 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 $c_i \in \mathcal{C}'$, $\mathsf{P}^*$ behaves like an honest prover, using $(r'_{l^-+1}, \ldots, r'_l)$ as random elements. Otherwise it halts.

The **uniformity property** of Theorem 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 $\mathsf{P}^*$ and the oracle $\mathcal{O}^{\phi(\cdot)}$, we play the following game. We substitute $\mathcal{G}$ and $\mathcal{H}$ 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 $\mathcal{O}'^{\phi(\cdot)}$ now performs its computations within $\mathcal{G}'$ and $\mathcal{H}'$.

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

(i) For each $\zeta_i$, it checks whether there is a pair $(\zeta_{ji}, \bar{R_{ji}})$ with $\zeta_i = \zeta_{ji}$ in $\mathsf{L}$. 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 $\mathsf{L}$. 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 $\mathsf{V}$. Former are marked as elements of $\mathcal{G}'$, latter as elements of $\mathcal{H}'$.

(ii) If $c_i \in \mathcal{C}'$, $\mathsf{P}^*$ analogously computes its response according to the protocol. Otherwise, if $c \notin \mathcal{C}'$, $\mathsf{P}^*$ halts.

By $\mathbf{r}$ we denote an element from the set of from which the oracle and by the generator of the input to the protocol draw their random choices, 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 [50].

**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 $E_i$'s 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 $i$.

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 $\Sigma$-protocol. Using that $E_i \neq W$ and Lemma 8, and noting that after $v$ oracle queries for $E_i$'s and $F_j$'s each, all coefficients are smaller than $2^v \cdot K$, we get

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

**Case (b).** Using $K$ as before, and observing that there are at most $v$ different $E_i$'s and $F_j$'s each, 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 Lower Bounds for the $\Sigma^\phi$-Protocol in the Plain Model

As pointed out by Dent and Fischlin [39, 40], restrictions proven in the generic model do not necessarily hold true in the plain model as well. In this section we thus confirm our results obtained in the generic model by showing the existence of lower bounds in the plain model. That is, we provide evidence that for exponentiation homomorphisms with hidden order co-domain, and for power homomorphisms of the form $\phi(\cdot) : \mathcal{H} \to \mathcal{H} : w \mapsto w^e$, no smaller knowledge error than in the generic model can be reached in the plain model. The results only hold for the $\Sigma^\phi$-protocol, and not for the entire class of generic $\Sigma$-protocols.

The following results are based on a generalization of the Root Assumption [47], which we call the *Expected Root Assumption*. We say that the Expected

Root Assumption holds for a group $\mathcal{H}$ if there exists no expected PPT algorithm that on input a random element $h \in_R \mathcal{H}$ and $e \geq 2$ outputs an $e^{th}$ root of $h$ with non-negligible probability. In contrast to the standard Root Assumption, we here also require that no *expected* PPT algorithm has a noticeable success probability. 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 $\Sigma^\phi$-protocol for power homomorphisms of the form $\phi_P(\cdot) : \mathcal{H} \to \mathcal{H} : w \mapsto w^e$. This is a generalization of the protocol proposed Guillou/Quisquater [4]. We generalize the result from Shoup [26] from exponents of the form $e = 2^t$ to arbitrary values of $e$.

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

**Theorem 9 (Bounds for Power Homomorphisms).** *Let* $\mathrm{poly}(\cdot)$ *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* $\Sigma^\phi$*-protocol for* $\phi_P(\cdot)$ *is lower bounded by*

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

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

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 $\gcd(e, \mathrm{ord}\,\mathcal{H}) = 1$ is always satisfied.

We stress that, 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:

**Corollary 10.** *Let the conditions of Theorem 9 be satisfied, and let the challenge set be an integer interval (i.e.,* $\mathcal{C} = \{a, \ldots, b\}$ *for some* $a, b \in \mathbb{Z}$*). Let* $d$ *denote the smallest divisor of* $e$*. Then knowledge error of the* $\Sigma^\phi$*-protocol is bounded from below by*

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

Theorem 9 becomes meaningless if all elements of $\mathcal{C}$ are co-prime (e.g., if all elements of $\mathcal{C}$ are primes), as it then implies a lower bound of 0. Though, the result is still relevant when seen in connection with Theorem 6. Namely, while the latter states that any hypothetical knowledge extractor has to use encoding specific properties of the homomorphism $\phi_P(\cdot)$, the former further restricts the situations where the generic result could potentially be violated in the plain model. In summary, the existence of an extractor underrunning the limitation of $1/d$ seems unlikely.

### 4.2 Lower Bounds for Exponentiation Homomorphisms

For exponentiation homomorphisms $\phi_E(\cdot) : \mathbb{Z} \to \mathcal{H} : w \mapsto h^w$ with hidden order co-domain $\mathcal{H}$, the $\Sigma^\phi$-protocol is only known to be a PoK with knowledge error $1/2$. In this section we show that (if existing at all) any knowledge error achieving a smaller knowledge error in this case would require fundamentally new insights to the $\Sigma^\phi$-protocol.

Although being used for numerous different homomorphisms, essentially only one knowledge extractor is known for the $\Sigma^\phi$-protocol. This standard knowledge extractor works as described next. In a first phase, it is given rewindable 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. 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*. We show that no such knowledge extractor can underrun a knowledge error of $1/2$ for the $\Sigma^\phi$-protocol and exponentiation homomorphisms with hidden order co-domain.

Let us introduce some notation: for a set $\mathcal{S}$ of integers, we write $\mathrm{Diff}(\mathcal{S})$ for the set of all possible absolute values of differences between different elements of $\mathcal{S}$, i.e., $\mathrm{Diff}(\mathcal{S}) := \{|s_1 - s_2| : s_1 \neq s_2 \in \mathcal{S}\}$. 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* $\Sigma^\phi$*-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 $\phi_M(\cdot) : \mathcal{G}^r \to \mathcal{H} : (w_1, \ldots, w_r) \mapsto h_1^{w_1} \ldots h_r^{w_r}$.

In practice the conditions of this theorem are most often satisfied. For instance consider the case where $\mathcal{H} = \mathbb{Z}_n^*$ for a safe RSA modulus $n$, i.e., $n = (2p+1) \cdot (2q+1)$, where $p$, $q$, $(2p+1)$, and $(2q+1)$ are primes. Then $\mathcal{H}'$ is usually given by the set of quadratic residues modulo $n$, and we have $\#\mathcal{H}'/\#\mathcal{H} = 1/4$. Further, $\mathrm{ord}\,\mathcal{H}' = p \cdot q$, and hence any challenge set $\mathcal{C}$ only containing elements smaller than $p, q$ will satisfy the condition of $\mathrm{Diff}(\mathcal{C})$ and $\mathrm{ord}\,\mathcal{H}'$ 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, Theorem 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. According to current knowledge, we doubt the existence of such an extractor. We thus believe that for reaching a small knowledge error in the case of exponentiation homomorphisms with hidden order co-domain, either running the $\Sigma^{\phi}$-protocol repeatedly or employing an FO-based protocol cannot be avoided.

## 5 Conclusion

We have introduced the class of generic $\Sigma$-protocols, and have shown that in the generic group model 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 for homomorphisms with hidden order co-domain. We further generalized this result to special homomorphisms as well, covering essentially all homomorphisms being used in cryptography. Especially, those underlying various crypto systems fall into this class [45–49]. We then confirmed our results for the $\Sigma^{\phi}$-protocol and certain homomorphisms in the plain model as well.

Besides pointing out these limitations, our results also give insights in how these restrictions could be overcome. Namely, any $\Sigma$-protocol overcoming these bounds must either be substantially different from the $\Sigma^{\phi}$-protocol (i.e., it must not be a generic $\Sigma$-protocol), or it must have a non-generic knowledge extractor.

The former seems to be hard to achieve without using auxiliary constructions resulting from a common reference string as done in [5, 20, 21], because the class of generic $\Sigma$-protocols does not leave much design options for other $\Sigma$-protocols to look like. Yet, the latter also is unlikely, because of our results in the plain model. Thus, although being riddled with various limitations from a theoretical point of view, FO-based protocols [5, 19–23] using common reference strings seem to be inevitable for many real systems.

## References

1. Bangerter, E., Camenisch, J., Krenn, S.: Efficiency limitations for $\Sigma$-protocols for group homomorphisms. Cryptology ePrint Archive, Report 2009/595 (2009)
2. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM **38** (1991) 691–729 Preliminary version in 27th FOCS, 1986.
3. Schnorr, C.: Efficient signature generation by smart cards. Journal of Cryptology **4** (1991) 161–174
4. Guillou, L., Quisquater, J.J.: A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In: CRYPTO 88. Volume 403 of LNCS., Springer (1990) 216–231
5. Bangerter, E.: Efficient Zero-Knowledge Proofs of Knowledge for Homomorphisms. PhD thesis, Ruhr-University Bochum (2005)
6. Cramer, R.: Modular Design of Secure yet Practical Cryptographic Protocols. PhD thesis, CWI and University of Amsterdam (1997)
7. Maurer, U.: Unifying zero-knowledge proofs of knowledge. In: AFRICACRYPT 2009. Volume 5580 of LNCS., Springer (2009) 272–286

8. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: CRYPTO 86. Volume 263 of LNCS., Springer (1987) 186–194
9. Boudot, F.: Efficient proofs that a committed number lies in an interval. In: EUROCRYPT 2000. Volume 1807 of LNCS., Springer (2000) 431–444
10. Camenisch, J.: Better privacy for trusted computing platforms: (extended abstract). In: ESORICS 2007. Volume 3193 of LNCS., Springer (2004) 73–88
11. Camenisch, J., Michels, M.: Proving in zero-knowledge that a number is the product of two safe primes. In: EUROCRYPT 99. Volume 1592 of LNCS., Springer (1999) 107–122
12. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: CRYPTO 2003. Volume 2729 of LNCS., Springer (2003) 126–144
13. Lipmaa, H.: On diophantine complexity and statistical zeroknowledge arguments. In: ASIACRYPT 2003. Volume 2894 of LNCS., Springer (2003)
14. Song, D.X.: Practical forward secure group signature schemes. In: ACM CCS 2001, ACM (2001) 225–234
15. Tang, C., Liu, Z., Wang, M.: A verifiable secret sharing scheme with statistical zero-knowledge. Cryptology ePrint Archive, Report 2003/222 (2003)
16. Tsang, P.P., Wei, V.K., Chan, T.K., Au, M.H., Liu, J.K., Wong, D.S.: Separable linkable threshold ring signatures. In: INDOCRYPT 2004. Volume 3027 of LNCS., Springer (2004) 384–398
17. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: ACM CCS 2004, ACM (2004) 132–145
18. Camenisch, J., Herreweghen, E.V.: Design and implementation of the idemix anonymous credential system. In: ACM CCS 2002, ACM (2002) 21–30
19. Bangerter, E., Camenisch, J., Maurer, U.: Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In: PKC 2005. Volume 3386 of LNCS., Springer (2005) 154–171
20. Bangerter, E., Krenn, S., Sadeghi, A.R., Schneider, T., Tsay, J.K.: Design and implementation of efficient zero-knowledge proofs of knowledge. In: SPEED-CC 2009. (2009)
21. Damgård, I., Fujisaki, E.: A statistically-hiding integer commitment scheme based on groups with hidden order. In: ASIACRYPT 2002. Volume 2501 of LNCS., Springer (2002) 77–85
22. Fujisaki, E., Okamoto, T.: Statistical zero knowledge protocols to prove modular polynomial relations. In: CRYPTO 97. Volume 1294 of LNCS., Springer (1997) 16–30
23. Camenisch, J., Kiayias, A., Yung, M.: On the portability of Generalized Schnorr Proofs. In: EUROCRYPT 2009. Volume 5479 of LNCS., Springer (2009) 425–442
24. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Journal of the ACM **51** (2004) 557–594 Preliminary version in STOC, 1998.
25. Pass, R.: On deniability in the common reference string and random oracle model. In: CRYPTO 2003. Volume 2729 of LNCS., Springer (2003) 316–337
26. Shoup, V.: On the security of a practical identification scheme. In: EUROCRYPT 96. Volume 1070 of LNCS., Springer (1996) 344–353
27. Damgård, I., Koprowski, M.: Generic lower bounds for root extraction and signature schemes in general groups. In: EUROCRYPT 2002. Volume 2332 of LNCS., Springer (2002) 256–271
28. Nechaev, V.I.: Complexity of a determinate algorithm for the discrete logarithm. Mathematical Notes **55** (1994) 165–172 Translated from Matematicheskie Zametki, 55(2):91–101, 1994.

29. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: EUROCRYPT 97. Volume 1233 of LNCS., Springer (1997) 256–266
30. Abe, M., Fehr, S.: Perfect NIZK with adaptive soundness. In: TCC 2007. Volume 4392 of LNCS., Springer (2007) 118–136
31. Aggarwal, D., Maurer, U.: Breaking RSA generically is equivalent to factoring. In: EUROCRYPT 2009. Volume 5479 of LNCS., Springer (2009) 36–53
32. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: EUROCRYPT 2005. Volume 3494 of LNCS., Springer (2005) 440–456
33. Brown, D.: Generic groups, collision resistance, and ECDSA. Cryptology ePrint Archive, Report 2002/026 (2002)
34. Dent, A.W.: The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156 (2006)
35. Maurer, U.: Index search, discrete logarithms, and Diffie-Hellman. In: Number-theoretic cryptography workshop, Mathematical Sciences Research Institute, Berkeley (2000)
36. Maurer, U., Wolf, S.: Lower bounds on generic algorithms in groups. In: EUROCRYPT 98. Volume 1403 of LNCS., Springer (1998) 72–84
37. Schnorr, C., Jakobsson, M.: Security of signed elgamal encryption. In: ASIACRYPT 2000. Volume 1976 of LNCS., Springer (2000) 73–89
38. Smart, N.P.: The exact security of ECIES in the generic group model. In: 8th International Conference on Cryptography and Coding – IMA 2001. Volume 2260 of LNCS., Springer (2001) 73–84
39. Dent, A.W.: Adapting the weaknesses of the random oracle model to the generic group model. In: ASIACRYPT 2002. Volume 2501 of LNCS., Springer (2002) 100–109
40. Fischlin, M.: A note on security proofs in the generic model. In: ASIACRYPT 2000. Volume 1976 of LNCS., Springer (2000) 458–469
41. Cramer, R., Damgård, I.: On the amortized complexity of zero-knowledge protocols. In: CRYPTO 2009. Volume 5677 of LNCS., Springer (2009) 177–191
42. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: CRYPTO 92. Volume 740 of LNCS., Springer (1993) 390–420
43. Goldreich, O.: Foundations of Cryptography – Basic Tools. Cambridge University Press (2001)
44. Babai, L., Szemerédi, E.: On the complexity of matrix group problems I. IEEE FOCS 84 (1984) 229–240
45. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: PKC 2001. Volume 1992 of LNCS., Springer (2001) 119–136
46. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT 99. Volume 1592 of LNCS., Springer (1999) 223–238
47. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21** (1978) 120–126
48. Takagi, T.: Fast RSA-type cryptosystem modulo $p^k q$. In: CRYPTO 98. Volume 1462 of LNCS., Springer (1998) 318–326
49. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO 84. Volume 196 of LNCS., Springer (1985) 10–18
50. Shoup, V.: OAEP reconsidered. In: CRYPTO 2001. Volume 2139 of LNCS., Springer (2001) 239–259