Efficient Proofs Of Knowledge of Discrete Logarithms and Representations in Groups with Hidden Order

Endre Bangerter[1], Jan Camenisch[1], and Ueli Maurer[2]

[1] IBM Research, Zurich Research Lab, CH-8803 Rueschlikon, Switzerland
{eba|jca}@zurich.ibm.com
[2] Departement of Computer Science, ETH Zurich, CH-8092 Zurich, Switzerland
maurer@inf.ethz.ch

Abstract. For many one-way homomorphisms used in cryptography, there exist efficient zero-knowledge proofs of knowledge of a preimage. Examples of such homomorphisms are the ones underlying the Schnorr or the Guillou-Quisquater identification protocols.
In this paper we present, for the first time, efficient zero-knowledge proofs of knowledge for exponentiation $\psi(x_1) \doteq h_1^{x_1}$ and multi-exponentiation homomorphisms $\psi(x_1, \ldots, x_l) \doteq h_1^{x_1} \cdot \ldots \cdot h_l^{x_l}$ with $h_1, \ldots, h_l \in H$ (i.e., proofs of knowledge of discrete logarithms and representations) where H is a group of hidden order, e.g., an RSA group.

1 Introduction

Consider mappings $\psi : G \to H$, where the domain is the group $(G, +)$ and the codomain is $(H, \cdot)$. A mapping ψ is called a *homomorphism* if for all g and g' from G it holds that $\psi(g+g') = \psi(g) \cdot \psi(g')$. A *proof of knowledge of a preimage under a homomorphism* is a two party protocol between a prover and a verifier. The parties' common input is a homomorphism ψ and an element $y \in H$. As a result of the protocol the verifier either accepts or rejects. Informally speaking, a proof of knowledge has the property that if a prover succeeds in getting the verifier to accept with a probability larger than some threshold probability (*knowledge error*), then the prover must "know" a preimage x' of y, i.e., an element $x' \in G$ such that $y = \psi(x')$. That is, there exists an algorithm (*knowledge extractor*) for the protocol that can compute a preimage x' of y given rewinding oracle access to such a prover.

For all (computable) homomorphisms there exists a proof of knowledge: the well known commitment-challenge-response protocol, often called Σ-protocol [17, 18], with binary challenges. Due to the binary challenges, the protocol has a knowledge error of 1/2 and therefore it needs to be repeated sequentially sufficiently many times to achieve a reasonably small knowledge error (i.e., a small success probability for a cheating prover). However, some homomorphisms allow one to use the Σ-protocol with larger challenges, which results in a smaller knowledge error. Thus, the protocol needs to be repeated only a few times or just once, which is an order of magnitude more efficient. Examples of homomorphisms that allow for this are the ones underlying the Schnorr and the Guillou-Quisquater identification schemes [32, 27]. In fact, Cramer [17] remarks that all the homomorphisms for which this is the case allow one to compute some information (e.g., the order of the group) from their description that enables the knowledge extractor, together with the information extracted from a convincing prover, to compute a preimage. Cramer calls such homomorphisms *special*.

Unfortunately, many homomorphisms widely used in cryptographic protocols are not known to be special and hence the most efficient proofs of knowledge known for them is the Σ-protocol with binary challenges. Prominent examples of such homomorphisms are exponentiations $\psi(x_1) \doteq h_1^{x_1}$ and multi-exponentiations $\psi(x_1, \ldots, x_l) \doteq h_1^{x_1} \cdot \ldots \cdot h_l^{x_l}$ with $h_1, \ldots, h_l \in H$ in hidden order groups H, e.g., where H is a class group [7, 22] or an RSA group. Such homomorphisms are for instance the basis of recent group signature and identity escrow schemes, credential systems, and fair exchange protocols [2, 1, 8–10, 28, 5, 25, 11]. In fact in these schemes, the authors often employ the Σ-protocol with non-binary challenges, sometimes wrongly relying on them to be proofs of knowledge in this setting as well.

Existing Work. Girault [26] suggests an efficient proof of knowledge for discrete logarithms in the RSA group based on the Σ-protocol. His approach is to publish the order of the sub-group in which the images lies. This requires, on the one hand, that the RSA modulus has a special form and, on the other hand, the non-standard assumption that giving away the order of the sub-group does not allow one to factor the RSA modulus. Also, one can no longer make use of the RSA-trapdoor for this subgroup with this approach.

Poupard and Stern [30] describe an identification scheme based on the Σ-protocol, where the private key is a discrete logarithm of a generator of a subgroup of the RSA group. They show that from an adversary that brakes the identification scheme, a discrete logarithm can be extracted. While their construction is appropriate to prove the security of their identification scheme, their protocol is not a proof of knowledge of a discrete logarithm in the RSA group.

The most relevant work in the field is the one by Damgaard and Fujisaki [21] (based on work by Fujisaki and Okamoto [24]). They show that the Σ-protocol can be used in certain cases to demonstrate knowledge of a discrete logarithm (or representation) in hidden order groups provided that the prover is not given the group's order. Let us refer in the following to the Damgaard and Fujisaki scheme as the DF scheme. As pointed out and explained in detail by its authors[21], the DF scheme is *not* a (computational) proof of knowledge according to the standard definition [4]. Rather, it only works in a stronger definitional setting resulting in "weak proofs of knowledge". Technically, the DF scheme demonstrates knowledge only over a suitable probability distribution of (multi-) exponentiations. This distribution is enforced in a setup phase prior to the proof protocol. While for some applications this is appropriate, it often leads to complicated and error prone proofs of security as one can no longer consider each proof protocol separately (as one could with standard proofs of knowledge) but has to analyze all of them in conjunction with each other. In fact, many authors seem not to be aware of this fact and correspondingly the security analysis of their applications using the DF scheme are incomplete or false.

Our Results. In this paper we provide two new methods to obtain, for the first time, efficient zero-knowledge proofs of knowledge for (multi-) exponentiations in hidden order groups H, where the order H is not known to at least the verifier.

Our first method is based on a new protocol. It yields efficient proofs of knowledge for any (multi-) exponentiation homomorphism in groups H with hidden order. The efficiency of the proof depends on the smallest factor of the order of the homomorphism's image. Thus, we obtain for instance efficient proofs for discrete logarithm based homomorphisms in RSA groups whose modulus is a product of two safe primes. Technically, we apply the ideas underlying the DF scheme and extend them to obtain standard proofs of knowledge. As a consequence one can *always* use our protocol instead of the DF scheme to obtain standard proofs of knowledge. Yet, compared to the DF scheme, our protocol is applicable to a wider number of settings and is also more efficient in certain application scenarios.

Our second method is based on the Σ-protocol. It is relies on the new idea to provide auxiliary information to the verifier (and thus to the knowledge extractor) to obtain proofs of knowledge for homomorphisms for which the Σ-protocol is not known to work otherwise. The method applies to (multi-) exponentiation homomorphisms in hidden order groups H, provided that the prover (but not the verifier) knows the order H. The method relies on the hardness of a new computational problem, which we call the *pseudo preimage problem.* We prove the pseudo preimage problem to be hard under standard assumptions, e.g., the RSA assumption. This result is potentially of independent interest for the construction new cryptographic schemes.

A Remark on the Presentation. We formulate all our new results for multi-exponentiation $\psi_M : \mathbb{Z}^l \to H$ in hidden order groups H. We would like to emphasize that the results (trivially) specialize to the practically relevant cases such as when H is an RSA group or a class group and also to simple exponentiations $\psi_M(x) = h^x$. We recall that what we call a proof of knowledge for a multi-exponentiation homomorphism is often referred to as a proof of knowledge of a representation

2 Preliminaries

Let M be an algorithm. By $y \leftarrow M(x)$, we denote that y was obtained by running M on input x. If M is deterministic, then this y is unique; if M is probabilistic, then y is a random variable.

By k we denote an integer *security parameter.* A *negligible function* is a function that, asymptotically in k, is smaller than one divided by any polynomial in k. A *non-negligible function* is a function that is asymptotically lower bounded by any polynomial in k.

We call a computational problem *hard* if there is a probability ensemble $\mathcal{D}(k)$ on problem instances such that for any probabilistic polynomial-time algorithm, the probability of solving the problem over choices according to $\mathcal{D}(k)$ is negligible. If there is a probabilistic polynomial-time algorithm that is successful over choices $\mathcal{D}(k)$ with probability $1 - \nu(k)$, where $\nu(k)$ is a negligible function, we call the problem *easy*.

Let $(G, +)$ and $(H, \cdot)$ be abelian groups, with their identity elements denoted 0 and 1, respectively. By $|H|$ we denote the order of the group H, and by $|h|$ the order of the element $h \in H$. We say that a group H has *hidden order*, if there is a description of H such that it is hard to compute a non-zero multiple of $|H|$. A *(group) homomorphism* ψ is a mapping $\psi : G \to H$ such that $\psi(g_1 + g_2) = \psi(g_1) \cdot \psi(g_2)$ for all $g_1, g_2 \in G$. We recall that the image of a homomorphism image(ψ) is a subgroup of its co-domain H. In the following we assume that H is a finite group. Throughout the paper ψ_M stands for a multi-exponentiation homomorphism $\psi_M : \mathbb{Z}^l \to H$, where $\psi_M(x_1, \ldots, x_l) = h_1^{x_1} \cdot \ldots \cdot h_l^{x_l}$ and $h_1, \ldots, h_l \in H$. We always assume that groups and homomorphisms are computationally tractable. That is, there shall be descriptions of groups and homomorphisms such that in (probabilistic) polynomial-time one can evaluate the group operation, invert group elements, test membership in the group, uniformly choose an element from the group (for finite groups), and evaluate a homomorphism.

By a *collection of homomorphisms* Ψ we refer to a (finite or infinite) set of homomorphisms together with a probability ensemble $\mathcal{D}_\Psi(k)$ on Ψ. We assume that there is a probabilistic polynomial-time algorithm that allows one to chose homomorphisms ψ according to $\mathcal{D}_\Psi(k)$. Also, we consider families of sets $\Psi(k) \doteq \{\psi : \psi \in \Psi, \psi : G \to H, \text{ and } |H| = k\}$. The notion of a collection of homomorphisms Ψ comprises as special cases families of homomorphisms, where Ψ is an infinite set of homomorphisms indexed by the security parameter, and single homomorphisms, i.e., $\Psi = \{\psi\}$.

Given a binary relation $\mathcal{R}$, we denote the corresponding language by $\mathcal{L}_\mathcal{R}$. Homomorphism collections give rise to what we call a *homomorphism relation*

$$\mathcal{R}[\Psi] \doteq \{((\psi, y), x) : \psi \in \Psi, \psi : G \to H, x \in G, y \doteq \psi(x)\} \quad \text{or}$$

$$\mathcal{R}[\Psi(k)] \doteq \{((\psi, y), x) : \psi :\in \Psi(k), \psi : G \to H, x \in G, y \doteq \psi(x)\} \ .$$

Our results on (computational) proofs of knowledge are formulated with respect to the corresponding definitions put forth by Bellare and Goldreich [4].

3 The Pseudo Preimage Problem

In this section we introduce the notion of a *pseudo preimage* of a homomorphism and a related computational problem termed the *pseudo preimage problem.* While the pseudo preimage problem has not been explicitly considered in existing work, it is implicit in the construction of all known knowledge extractors for the Σ-protocol. In fact, these constructions crucially rely on the existence of easy instances of the pseudo preimage problem. In the following we prove that for a large class of multi-exponentiation homomorphisms the pseudo preimage problem is hard.

Definition 1 (Pseudo Preimage). *Let be a homomorphism $\psi : G \to H$ and $y \in H$. A* pseudo preimage *of y under ψ is a pair (v, w) such that $y^v = \psi(w)$, where v is a non-zero integer and $w \in G$. We refer to v as the* exponent *of the pseudo preimage (v, w).*

Note that when ψ is not surjective, then there are pseudo preimages (v, w) of $y \in H$ under ψ even for elements $y \notin \text{image}(\psi)$.

Definition 2 (Pseudo Preimage Problem). *The* pseudo preimage (PP) problem for a homomorphism ψ *is to compute a preimage x of y under ψ given a pseudo preimage (v, w) of y under ψ.*

For homomorphisms that are easy to invert, the PP problem trivially is easy. More interestingly, the PP problem is also easy for certain one-way homomorphisms. In fact, as we will see in § 4.1, the existence of easy instances of the PP problem for one-way homomorphisms is key for the construction of knowledge extractors for the Σ-protocol. Examples of such one-way homomorphisms are the ones underlying the Schnorr and the Guillou-Quisquater schemes.

In the following we show that the PP problem is hard for multi-exponentiations in groups for which the ROOT problem, i.e., computing roots, is hard. Let us introduce concepts and notation used for the formulation of this result. We recall the ROOT problem for an arbitrary abelian group H. It is to compute a $h \in H$ such that $h^e = u$ given an integer $e > 1$ and a group element $u \in H$. Next, we define generators $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{D}_{\mathcal{P}}$ for the ROOT and PP problem, respectively. Let H be an arbitrary multiplicative abelian group and l be an arbitrary integer parameter. The generator $\mathcal{D}_{\mathcal{R}}(H)$ works as follows: 1) Choose $u \in_U H$ and an integer $e > 1$ such that $\gcd(|H|, e) = 1$, whereas the distribution of e may be arbitrary. 2) Output the ROOT problem instance (u, e).

In the definition of the generator for the PP problem we use as a subroutine a probabilistic polynomial-time algorithm $\tilde{\mathcal{D}}(H, l)$ with the following properties. The algorithm $\tilde{\mathcal{D}}(H, l)$ outputs tuples $(v, (w_1, \ldots, w_l), (e_1, \ldots, e_l))$, where v is an integer and $(w_1, \ldots, w_l)$ and $(e_1, \ldots, e_l)$ are elements of $\mathbb{Z}^l$, such that $v \nmid (e_1 w_1 + \ldots + e_l w_l)$ and $\gcd(|H|, v) = 1$. Apart from this, the tuples may be distributed arbitrarily. Note that the latter condition can be fulfilled by $\tilde{\mathcal{D}}(H, l)$ without being given $|H|$. It suffices if one can compute a $\lambda^+ \geq |H|$ from the description of H. Then one can, for instance, choose a v as a prime $\geq \lambda^+$.

Now, the generator $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$ is as follows: 1) Choose $(v, (w_1, \ldots, w_l), (e_1, \ldots, e_l)) \leftarrow \tilde{\mathcal{D}}(H, l)$ and an element $h \in_U H$. 2) Set $h_1 \doteq h^{e_1}, \ldots, h_l \doteq h^{e_l}$ and define the homomorphism $\psi_M : \mathbb{Z}^l \to H$ by $\psi_M(x_1, \ldots, x_l) \doteq h_1^{x_1} \cdot \ldots \cdot h_l^{x_l}$. 3) Set $(z_1, \ldots, z_l) \doteq (w_1 v^{-1} \pmod{|H|}, \ldots, w_l v^{-1} \pmod{|H|})$ and let $y \doteq \psi_M(z_1, \ldots, z_l) = h_1^{z_1} \cdot \ldots \cdot z_l^{x_l}$. (Note that by construction $(v, (w_1, \ldots, w_l))$ is a pseudo preimage of y under ψ_M.) 4) Output the PP problem instance $((v, (w_1, \ldots, w_l)), y, \psi_M)$.

Let be given computational problems P_1 and P_2 and the respective generators $\mathcal{D}_1$ and $\mathcal{D}_2$. We say P_2 is reducible to P_1, if given a probabilistic polynomial-time solver M with non-negligible success probability for P_1 over choices of $\mathcal{D}_1$, one can construct a probabilistic polynomial-time solver given black box access to M that has non-negligible success probability for P_2 over choices of $\mathcal{D}_2$. We denote this by $P_1[\mathcal{D}_1] \geq P_2[\mathcal{D}_2]$.

Theorem 1. *For the generators $\mathcal{D}_{\mathcal{R}}(H)$ and $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$ (as defined above) it is $PP[\mathcal{D}_{\mathcal{P}}(H, |H|, l)] \geq ROOT[\mathcal{D}_{\mathcal{R}}(H)]$.*

Proof. Given an instance of the ROOT problem $(u, e) \leftarrow \mathcal{D}_{\mathcal{R}}(H)$ we construct an instance of the PP problem as follows. Choose $(v', (w'_1, \ldots, w'_l), (e'_1, \ldots, e'_l)) \leftarrow \tilde{\mathcal{D}}(H, l)$. Then we set $h' \doteq u^v$, $h'_1 \doteq h'^{e_1}, \ldots, h'_l \doteq h'^{e'_l}$ and define the homomorphism $\psi'_M : \mathbb{Z}^l \rightarrow H$ as $\psi'_M(x_1, \ldots, x_l) \doteq {h'_1}^{x_1} \cdot \ldots \cdot {h'_l}^{x_l}$. We set $y' \doteq u^{(e'_1 w'_1 + \ldots + e'_l w'_l)}$. It is easy to see that we have constructed an instance $(v', (w'_1, \ldots, w'_l), y', \psi'_M)$ of the PP problem.

Let us verify the claim that the distribution of such instances $(v', (w'_1, \ldots, w'_l), y', \psi'_M)$ equals the distribution of instances generated as by $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$. From $y^v = \psi_M(w_1, \ldots, w_l)$ and $\gcd(v, |H|) = 1$ we have that the image element y is uniquely determined by ψ_M and $(v, (w_1, \ldots, w_l))$ and the same is true for $((v', (w'_1, \ldots, w'_l), y', \psi'_M)$. Hence, it suffices to show that the distribution of v', $(w'_1, \ldots, w'_l)$, and ψ'_M is indistinguishable from the distribution of the corresponding quantities chosen by $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$. By construction the distribution of tuples $(v', (w'_1, \ldots, w'_l), (e'_1, \ldots, e'_l))$ chosen above is the same as the one of those output by $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$. It remains to see that ψ'_M and ψ_M have the same distribution. To this end, note that $h' = u^v$, where u is a uniform random element of H. From $\gcd(v, |H|) = 1$ it follows that h' is uniformly distributed in H, and thus has the same distribution as the element h chosen by the generator $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$. The claim now follows immediately, as the homomorphism ψ'_M is constructed from h' in the same way as is ψ_M from h by the generator $\mathcal{D}_{\mathcal{P}}(H, |H|, l)$.

Let M we denote a probabilistic polynomial-time solver of the PP problem that is successful with non-negligible probability over choices of $\mathcal{D}_{\mathcal{P}}$. We invoke M on an instance of the PP problem (v', w), y, and ψ_M (constructed from an instance of the ROOT problem $(u, e) \leftarrow \mathcal{D}_{\mathcal{R}}(H)$). With non-negligible probability, M outputs a preimage $(z_1, \ldots, z_l)$ of y' under ψ'_M. Thus we have ${y'}^{v'} = ({h'}_1^{z_1} \cdot \ldots \cdot {h'}_l^{z_l})^{v'} = h'^{(e'_1 z_1 + \ldots + e'_l z_l) v'}$ and $y'^{v'} = {h'_1}^{w'_1} \cdot \ldots \cdot {h'_l}^{w'_l} = h'^{(e'_1 w'_1 + \ldots + e'_l w'_l)}$. Using $\lambda \doteq (e'_1 w'_1 + \ldots + e'_l w'_l) - (e'_1 z_1 + e'_2 z_2 + \ldots + e'_l z_l) v$ we have $h'^{\lambda} = 1$. By assumption $v \nmid (e'_1 w'_1 + \ldots + e'_l w'_l)$ and thus $\lambda \neq 0$, i.e., λ is a non-zero multiple of the order of h'. As h' and u have the same order, λ is also a multiple of the order of u. This allows us to compute the e-th root of u as follows. We note that λ is not necessarily co-prime to e. However, we have by assumption $\gcd(e, |H|) = 1$. Thus we can easily find a multiple λ' of $|u|$ that is co-prime to e, if we set $\lambda' \doteq \lambda$ and compute $\lambda' \doteq \lambda' / \gcd(e, \lambda')$ until $\gcd(e, \lambda') = 1$. Finally we compute $1/e$ modulo λ' to obtain $u^{1/e}$. $\square$

Theorem 1 implies that the PP problem is hard for multi-exponentiations in groups for which the ROOT problem is hard. This is widely assumed to be the case for RSA groups [31] and class groups [7]. Moreover, Damgaard and Koprowski [22] have shown that if a group has hidden order and if the order of that group contains a large prime factor, then the ROOT problem is hard for generic algorithms.

Corollary 1. *There is a probabilistic polynomial-time algorithm M such that the probability distributions $((v,(w_1,\ldots,w_l)),y,\psi_M) \leftarrow \mathcal{D}_{\mathcal{P}}(H,|H|,l)$ and $((v,(w_1,\ldots,w_l)),y,\psi_M) \leftarrow M(H,l)$ are equal.*

Corollary 1 follows from the proof of Theorem 1. It implies that instances of the PP problem as output by $\mathcal{D}_{\mathcal{P}}(H,|H|,l))$ do not reveal any computational information on the order of H.

4 Efficient Proofs of Knowledge Using Auxiliary Pseudo Preimages

This section presents a new technique that uses the hardness of the pseudo preimage problem to yields proofs of knowledge for multi-exponentiations ψ_M in groups for which the ROOT problem is hard (e.g., RSA and class groups). The proofs are based on the Σ-protocol. The technique requires that the honest prover is given the order of H, while it ensures that the verifier does not learn the order of H. The resulting proofs are efficient, they achieve in fact an arbitrarily small knowledge error in a single execution of the Σ-protocol.

4.1 Preliminaries: The Σ-Protocol and its Properties

In this section we review known properties of the Σ-protocol. For a detailed discussion we refer to Cramer [17] and Damgaard [20].

Definition 3 (Σ-Protocol). *Let Ψ be a collection of homomorphisms with a finite domain and $((\psi,y),x) \in \mathcal{R}[\Psi(k)]$. Let (P,V) be a pair of interactive Turing machines with common input (ψ,y), the private input of P being x, and challenge set $\mathcal{C} \doteq \{0,\ldots,c^+(k)\}$. A Σ-protocol is (P,V) performing the following joint computation.*

1. *P: Choose $r \in_U G$, compute $t \doteq \psi(r)$, and send t to V.*
2. *V: Choose $c \in_U \mathcal{C}$ and send c to P.*
3. *P: Set $s \doteq r + cx$ and send s to V.*
4. *V: If $\psi(s) = ty^c$ output 1; otherwise output 0.*

The Σ-protocol is honest-verifier zero-knowledge, but not known to be zero-knowledge unless the cardinality of $\mathcal{C}$ is polynomially bounded in k. In case one requires real zero-knowledge or the even stronger notion of concurrent zero-knowledge, one can apply one of numerous constructions, e.g., [19, 23, 15]. Most notably, the technique by Damgaard [19] achieves concurrent zero-knowledge at almost no computational and communicational overhead. In Definition 3, the Σ-protocol is only defined for homomorphisms with a finite domain. However, there is a standard variant of the Σ-protocol that is defined for multi-exponentiations $\psi_M : \mathbb{Z}^l \to H$ (which have an infinite domain). That variant of the protocol is statistical zero-knowledge instead of perfect zero-knowledge; apart from this, the above comments and results stated in the following are valid for both variants of the Σ-protocol.

We call Ψ a (collection of) *special homomorphisms*, if there is a probabilistic polynomial-time algorithm M that on input any $(\psi, y) \in \mathcal{L}_{\mathcal{R}[\Psi]}$ outputs a pseudo preimage (v, w) of y under ψ. The algorithm M is called a *pseudo preimage finder (for Ψ)*. An example of a special homomorphism is the one used in the Schnorr protocol, i.e., the mapping $\psi : \mathbb{Z}_q \to \mathbb{Z}_p^*$ defined by $\psi(x) \doteq h^x$ with $q \mid p$ and $|h| = q$. From the description of this mapping, the pseudo preimage finder can derive $(q, 0)$. Now $y^q = 1 = \psi(0)$ for all $y \in \text{image}(\psi)$ and therefore the pair $(q, 0)$ is a pseudo preimage of y under ψ. More generally, homomorphisms $\psi : G \to H$ for which a multiple of the order image(ψ) can be efficiently computed from the description of ψ are easily seen to be special. An example of a special homomorphism with hidden order co-domain is the mapping $\psi : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ given by $\psi(x) \doteq x^e$, where e is an integer, which is used in the Guillou and Quisquater [27] scheme. For such mappings we have $y^e = \psi(y)$ and hence (y, e) is a pseudo preimage of y under ψ.

To simplify the subsequent discussion we make the following assumption on collections Ψ and pseudo preimage finders M. For $(\psi, y) \in \mathcal{L}_{\mathcal{R}[\Psi(k)]}$ and $(v, w) \leftarrow M(\psi, y)$ we assume that the exponents v are all equal for a given value of the security parameter k, i.e., that $v = v(k)$. It is straightforward to generalize our discussion and results to the setting where this assumption is not made. Moreover, all known examples of (collections of) special homomorphisms fulfill this assumption.

Theorem 2. *The Σ-protocol with challenge set $\mathcal{C} = \{0, \ldots, c^+(k)\}$ is a proof of knowledge for $\mathcal{R}[\Psi]$,*

(a) with knowledge error $1/2$ if $c^+(k) = 1$.

(b) with knowledge error $1/(c^+ + 1)$ if Ψ is a collection of special homomorphism and $c^+(k) < p(k)$, where $p(k)$ is the smallest prime dividing the pseudo preimage exponent $v(k)$ output by a pseudo preimage finder M for Ψ.

Pseudo preimages have the property that given two (appropriate) pseudo preimages of y under ψ one can compute a preimage of y as follows.

Lemma 1 (Shamir's Trick). *Let be given two pseudo preimages (v_1, w_1) and (v_2, w_2) of y for ψ. If $\gcd(v_1, v_2) = 1$, then $x = aw_1 + bw_2$ is a preimage of y under ψ, where a and b are integers such that $av_1 + bv_2 = 1$ as they can for instance be computed using the extended Euclidean algorithm.*

Proof (Theorem 2). Let us describe a knowledge extractor for the Σ-protocol. Let P^* be an arbitrary prover that is successful in the Σ-protocol on common input $(\psi, y) \in \mathcal{L}_{\mathcal{R}[\Psi]}$ and arbitrary private input with probability $\epsilon > \kappa \doteq 1/(c^+ + 1)$. It is well known that given rewinding access to P^*, one can obtain a pair of tuples (t, c, s) and (t', c', s') that fulfill the verification equation in step 4 of the Σ-protocol, with $t = t'$ and $c \neq c'$. We refer to this property of the Σ-protocol as the *collision extractibility* property. For a detailed analysis of this property we refer to Damgaard [20]. Now, using $\triangle c \doteq c' - c$ and $\triangle s \doteq s - s'$, where wlog we assume $\triangle c > 0$, one gets

$$y^{\triangle c} = \psi(\triangle s). \tag{1}$$

In the case where the challenge set is $\mathcal{C} = \{0, 1\}$ we have $\triangle c = 1$ and thus $y = \psi(\triangle s)$. This proofs part (a) of the theorem. To prove part (b) we may assume that ψ is special. Now, we in invoke a pseudo preimage finder for ψ on input (ψ, y) to obtain a pseudo preimage (v, w) of ψ under y. Using that $\triangle c \leq c^+$ and the assumption $c^+(k) < p(k)$, it follows that $\gcd(v, \triangle c) = 1$, and by Lemma 1 we can compute a preimage of y under ψ. □

We call the knowledge extractor described in the proof of Theorem 2 the *standard knowledge extractor (for the Σ-protocol).* This is only known the standard knowledge extractor is the *only* knowledge extractor that is known for the Σ-protocol. In fact, Cramer [17] points out that all existing knowledge extractors for the Σ-protocol with a challenge set of cardinality larger than two use the collision extractability property, the existence of pseudo preimage finders for special homomorphisms, and Shamir's trick to compute a preimage.

It is worthwhile to note that the standard knowledge extractor is only successful when the instances (1) of the PP problem (obtained from the prover P^*) are easy to solve. In fact, we can distinguish two classes of PP instances that are easy to solve. One class consists of PP problem instances $((v, w), y, \psi)$ with $v = 1$, where w is a preimage of y under ψ, in which case the PP problem is trivial to solve. The other class consists of easy PP problem instances for special homomorphisms. In fact, let ψ be a special homomorphism, $y \in \text{image}(\psi)$, and (v, w) be the pseudo preimage output by a pseudo preimage finder for ψ. Then by Lemma 1 all instances $((v', w'), y, \psi)$ of the PP problem with $\gcd(v, v') = 1$ are easy. The former class of easy instances underlies the proof of part (a) and the latter the proof of part (b) of Theorem 2.

For non-special homomorphisms, such as multi-exponentiations in groups with hidden order, the PP problem instances (1) extracted from the Σ-protocol with non-binary challenge set are not known to be easy. Hence, the standard knowledge extractor does not work for non-special homomorphisms.

4.2 Basic Idea Underlying Our Scheme

Our idea in the following is to enhance the common input of the Σ-protocol by a pseudo preimage. That is, we consider the Σ-protocol on common input (ψ, y) and a pseudo preimage (v, w) (of y under ψ). The prover's private input remains to be a preimage x (of y under ψ). This allows us to obtain proofs of knowledge for non-special homomorphisms using the Σ-protocol with challenge sets of cardinality larger than two.

Let us refer to the pseudo preimage in the common input as an "auxiliary pseudo preimage". In fact, auxiliary pseudo preimages enable us to use the standard knowledge extractor for non-special homomorphisms. This claim is easy to verify: The common input and thus the auxiliary pseudo preimage is by definition given to the knowledge extractor [4]. We recall that the standard knowledge extractor (described in §4.1) first computes a pseudo preimage $(\triangle c, \triangle s)$ from the prover P^*. It then uses a second pseudo preimage to compute the desired preimage using Shamir's trick. For special homomorphisms the second pseudo preimage can be obtained using a corresponding pseudo preimage finder. In our approach, this second preimage is the auxiliary preimage contained in the common input. In the following we formalize this idea and discuss under what conditions it can be used to obtain practically useful proofs of knowledge.

Definition 4. *Let $v(k)$ be an arbitrary integer parameter and Ψ a collection of homomorphisms. We call $\mathcal{R}^{(v)}[\Psi] \doteq \{((\psi, y, (v(k), w)), x) : \psi \in \Psi(k), \psi : G \to H, x \in G,\ y = \psi(x),$ and $(v(k), w)$ is a pseudo preimage of y under $\psi\}$ a* pseudo preimage relation*.*

Definition 3 specifies the Σ-protocol only for homomorphism relations. However, the Σ-protocol cleary also applies to pseudo preimage relations $\mathcal{R}^{(v)}[\Psi]$ with the common input being $(\psi, y, (v, w)) \in \mathcal{L}_{\mathcal{R}^{(v)}[\Psi]}$.

Corollary 2. *The Σ-protocol with challenge set $\mathcal{C} \doteq \{0, \ldots, c^+(k)\}$ is a proof of knowledge for the pseudo preimage relation $\mathcal{R}^{(v)}[\Psi]$ if the smallest prime factor of $v(k)$ is larger than $c^+(k)$. The knowledge error is $1/(c^+(k) + 1)$.*

Corollary 2 follows from the proof of Theorem 2. Let us consider a collection of homomorphisms Ψ, a homomorphism relation $\mathcal{R}[\Psi]$, and the pseudo preimage relation $\mathcal{R}^{(v)}[\Psi]$. We observe that a proof of knowledge for a $((\psi, y), x) \in \mathcal{R}[\Psi]$ and a proof of knowledge for $((\psi, y, (v, w)), x) \in \mathcal{R}^{(v)}[\Psi]$ both are proofs of knowledge of a preimage x of y under ψ (possibly with different knowledge errors). Thus to prove knowledge of a preimage under a homomorphism one can also use proofs for pseudo preimage relations. In the following we pursue this idea of using pseudo preimage relations for proving knowledge of a preimage of a homomorphism. We refer to a proof of knowledge for a collection of homomorphisms Ψ using a pseudo preimage relation $\mathcal{R}^{(v)}[\Psi]$ as a *proof of knowledge in the auxiliary setting* and call $\mathcal{R}^{(v)}[\Psi]$ an*auxiliary relation*.

A desirable property of the auxiliary setting is that it allows one to obtain very efficient proofs of knowledge for *any* homomorphism collection Ψ. In fact, using the Σ-protocol in the auxiliary setting, we can achieve an arbitrary small knowledge error for any Ψ. Therefore, we use the auxiliary relation $\mathcal{R}^{(v)}[\Psi]$, where $v(k)$ is prime, and the Σ-protocol with the challenge set $\mathcal{C} \doteq \{0, \ldots, (v(k) - 1)\}$. By Corollary 2, the resulting knowledge error is $1/v(k)$, which can be made arbitrarily small by choosing $v(k)$ appropriately. This is in contrast to existing proofs of knowledge for homomorphisms obtained the Σ-protocol in the ordinary way, where the knowledge error can not be made arbitrarily small and is in fact often quite large (see Theorem 2).

Our discussion so far was focused on obtaining proofs of knowledge of a preimage. We have seen that within this focus proofs in the auxiliary setting and conventional proofs (i.e., proofs for homomorphism relations without using auxiliary pseudo preimages) are equivalent and thus one can use the former instead of the latter. However, if we widen our focus, then the auxiliary setting is in general not equivalent to the conventional setting. The reason is that by providing auxiliary pseudo preimages might reveal information that is not available otherwise. For instance, the auxiliary pseudo preimage could suddenly allow a verifier to compute a preimage from the common input to the Σ-protocol Thus, in the following we need to additionally consider what (computational) information the prover and the verifier obtain from an auxiliary pseudo preimage.

4.3 Proofs of Knowledge in the Auxiliary Setting for Multi-Exponentiations

In the following we look at proofs of knowledge in the auxiliary setting for multi-exponentiations $\psi_M : \mathbb{Z}^l \to H$ in groups H for which the ROOT problem is hard. In particular, we consider the information the prover and the verifier can derive from an auxiliary pseudo preimage. It turns out that, on the one hand, the verifier does not get any additional (computational) information on $|H|$ and the preimage x. On the other hand, we see that $|H|$ is required by the (honest) prover.

Let us first consider a (possibly dishonest) verifier in the auxiliary setting. The results from §3 allow us to exclude that the verifier can either compute a preimage or information about the order of H from an auxiliary pseudo preimage. In fact, by Theorem 1 (under the ROOT assumption) it is impossible for the verifier to compute a preimage from a pseudo preimage, i.e., to solve the PP problem. Concerning the order of H, Corollary 1 implies that instances of the PP problem for multi-exponentiations in a group H, and thus the common input to the Σ-protocol in the auxiliary setting, can be generated without knowing the order of H. Hence, an auxiliary pseudo preimage gives the verifier no advantage in computing the order of H neither. Finally, if the Σ-protocol is used such that it is zero-knowledge, the verifier does not get an advantage in computing a preimage or information on the order of H from running the protocol with the prover.

Next, we consider the information the (honest) prover learns on $|H|$ in the auxiliary setting. We note that the prover is in addition to the common input also given a preimage as private input. It is easy to see that from this input $(\psi_M, y, (v, w), x) \in \mathcal{R}^{(v)}$, where $\psi_M(z) = h^z$ is a simple exponentiation homomorphism, one can derive the order of h (and assuming that $v \nmid w$). In certain groups, such as RSA groups with moduli being a safe-prime product, this allows one to factor the modulus and to obtain the group's order. For the case where ψ_M is a multi-exponentiation, we don't know how to show that the (honest) prover obtains information on $|H|$. But neither can we prove that it does not get information on $|H|$. Thus, unless we want to put forth a corresponding (and "rather questionable") computational assumption, we should expect that the prover can compute $|H|$. Moreover, we only know how to generate the protocol's input in the auxiliary setting, i.e., $(\psi_M, y, (v, w), x)$ when the order of H is given. (For a possible way to generate the input we refer to the description of the PP instance generator $\mathcal{D}_{\mathcal{P}}$ in §3.) Thus, in the context of an application where the input to the Σ-protocol protocol in the auxiliary setting is generated by the (honest) prover, then the (honest) prover explicitly needs to be privy to $|H|$.

Finally, we note that if one uses our auxiliary setting to obtain a proof of knowledge as a sub-protocol in some application, one needs to consider the information an auxiliary pseudo preimage reveals in the context of the whole system– in the same way one has to do this for the image y itself. Such an analysis, however, must be outside the scope of this paper.

A property of practical interest of proofs of knowledge in the auxiliary setting is that one can use techniques from groups with known order for proving relations among preimages of different multi-exponentiations [6, 14]. As an example one can prove knowledge of two discrete logarithms of two different group elements with respect to different bases and also that the discrete logarithms are equal. That is, using notation introduced by Camenisch and Stadler [13], on can realize a proof $\mathrm{PK}(\{\alpha_1, \alpha_2\} : y_1 = h_1^{\alpha_1} \wedge y_2 = h_2^{\alpha_2} \wedge \alpha_1 = \alpha_2\}$. The approach to obtain such an equality proof in the auxiliary setting is to choose the auxiliary pseudo preimage (v, w) to be the *same* for $(y_1, \psi_{M,1}(x_1) = {h_1}^{x_1})$ and for $(y_2, \psi_{M,2}(x_2) = {h_2}^{x_2})$. Using this approach it is straightforward to verify that the knowledge extractor indeed is able to find a value $x = \log_{h_1} y_1 = \log_{h_2} y_2$.

5 The Σ^+-Protocol

In this section we introduce a new protocol that we call the Σ^+-protocol. The Σ^+-protocol is a an efficient zero-knowledge (computational) proof of knowledge for of multi-exponentiations ψ_M in arbitrary groups H and, in particular, in groups with hidden order. The knowledge error of the Σ^+-protocol is governed by the smallest prime in $|\operatorname{image}(\psi_M)|$. The computational validity property of the Σ^+-protocol holds under the Strong RSA assumption [3, 24] and under the computational binding property of the commitment scheme used in the protocol. The Σ^+-protocol is a proof of knowledge regardless of whether the prover or the verifier knows the order of H.

Technically, the construction of the Σ^+-protocol takes up and extends ideas underlying the DF scheme (c.f. § 1) to obtain standard proofs of knowledge according to [4]. In fact, the Σ^+-protocol can *always* be used to replace the DF scheme to obtain standard proofs of knowledge. Yet, compared to the DF scheme, the Σ^+-protocol works under weaker conditions and hence can be used more broadly. In fact, when applied to a multi-exponentiation $\psi_M : \mathbb{Z}^l \to H$, the DF scheme requires that H is a group (with hidden order) for which the generalized root assumption[3] holds, and that the prover must not know the order of H. The Σ^{+}-protocol needs neither of these requirements. Additionally, in certain applications scenarios the Σ^{+}-protocol is more efficient than the DF scheme. We recall that the DF scheme consists of two parts. A rather inefficient setup part that is run once and an efficient proof of knowledge part using the Σ-protocol, which is typically executed several times. The computational cost of the Σ^{+}-protocol, which is an atomic protocol, is roughly three times the cost of the Σ-protocol. As a consequence, the Σ^{+}-protocol is more efficient than the DF protocol in settings when a few proofs of knowledge are required, while the DF scheme is more efficient when one requires many proofs of knowledge.

5.1 Preliminaries

The Strong RSA assumption [3, 24] states that there is a generator $\mathcal{D}_{\mathcal{S}}(k)$ such that given $(n, g) \leftarrow \mathcal{D}_{\mathcal{S}}(k)$, with $g \in \mathbb{Z}_n^*$, it is hard to compute a $u \in \mathbb{Z}_n^*$ and an integer $e > 1$ fulfilling $u^e = g$. In the following we assume that $n = (2p+1)(2q+1)$ with p, q, $(2p+1)$, and $(2q+1)$ being primes, and that $g \in \mathrm{QR}_n$, where QR_n is the subgroup of quadratic residues of $\mathbb{Z}_n^*$.

We define a generator $\mathcal{D}_{\vartheta}(l, k)$ of multi-exponentiations $\vartheta : \mathbb{Z}^l \rightarrow \mathrm{QR}_n$ as follows. Choose $(n, g) \leftarrow \mathcal{D}_{\mathcal{S}}(k)$, then for $i = 1, \ldots, (l-1)$ choose $\rho_i \in_U [0, 2^k \lfloor n/4 \rfloor]$ and set $g_i \doteq g^{\rho_i}$. Using these choices define a multi-exponentiation $\vartheta : \mathbb{Z}^l \rightarrow \mathrm{QR}_n$ by $\vartheta(x_1, \ldots, x_l) \doteq g_1^{x_1} \cdot \ldots \cdot g_{l-1}^{x_{l-1}} \cdot g^{x_l}$ and output (ϑ, n).

Theorem 3. *Under the Strong RSA assumption, it is hard given $(n, \vartheta) \leftarrow D_{\vartheta}(l, k)$ to compute a $y \in \mathbb{Z}_n^*$ and a pseudo preimage $(v, (w_1, \ldots, w_l))$ of y under ϑ such that $v \neq 1$ and $v \nmid w_i$ for some $i \in \{1, \ldots, l\}$.*

Theorem 3 underlies the construction of the knowledge extractor of the DF scheme as well as the one for our Σ^{+}-protocol. A similar statement was recently proved by Camenisch and Shoup [12].

Let $\mathrm{commit}(\cdot, \cdot)$ be a computationally binding and statistically hiding commitment scheme such as the one by Pedersen [29]. To commit to value γ, one computes $C \leftarrow \mathrm{commit}(\gamma, r)$, where r is a random value. To open the commitment C, one reveals γ and r to a verifier, who checks that $C = \mathrm{commit}(\gamma, r)$.

5.2 The Σ^{+}-Protocol and its Properties

In this section we define the Σ^{+}-protocol. For simplicity, we describe the protocol only for simple-exponentiations $\psi_M(x) \doteq h^x$ with $h \in H$. This allows us to focus on the key ideas underlying the protocol construction. It is a straightforward exercise to extend the definition of the protocol and the results given below to multi-exponentiations $\psi_M(x_1, \ldots, x_l) \doteq h_1^{x_1} \cdot \ldots \cdot h_l^{x_l}$ with $h_1, \ldots, h_l \in H$. Let $\triangle x \doteq \triangle x(k)$ and $l_z \doteq l_z(k)$ denote integer parameters.

---

[3] The assumption is that given $h \in_{\mathrm{U}} H$ it is hard to compute an integer $e \neq 1$ and $u \in H$ such that $u^e = h$.

Definition 5 (Σ^+-Protocol). *Let Ψ be a collection of simple-exponentiation homomorphisms and $((\psi_M, y), x) \in \mathcal{R}[\Psi]$ with $x \in [-\triangle x, +\triangle x]$. Let (P, V) be a pair of interactive Turing machines with common input (ψ_M, y), the private input of P being x, and $\mathcal{C} \doteq \{0, \ldots, c^+\}$ be the challenge set. A Σ^+-protocol is (P, V) performing the joint computation described in Fig. 1.*

Note that $x \in [-\triangle x, +\triangle x]$ in Definition 5 is required for the Σ^+-protocol to become statistical zero-knowledge, i.e., one needs to know how large x can be to blind x in the messages sent by the prover. The statistical zero-knowledgeness of the Σ^+-protocol is controlled by the parameter l_z.

Next, we sketch the key features underlying the proof of knowledge and zero-knowledge property of the Σ^+-protocol. Let us therefore consider the Σ^+-protocol on input $((\psi_M, y), x)$.

First, we look at the proof of knowledge property, i.e., the features that allow us to construct a knowledge extractor. In step 1, the verifier chooses a multi-exponentiation $\vartheta(\cdot, \cdot)$ by executing the steps of the generator $\mathcal{D}_\vartheta(2, k)$ as defined in the previous section. The description of $\vartheta(\cdot, \cdot)$ is sent to the prover. In step 2, the prover first computes $\mathsf{y} \doteq \vartheta(x, \mathsf{x})$, where x is the preimage of y under ψ_M and x is random value to ensure that y does not reveal information about x. Now, we observe that the remainder of step 2 and steps 3, 4, and 7 essentially correspond to a Σ-protocol for each the homomorphisms ψ and ϑ. (For the matter of this observation, we may forget about the commitment $\mathrm{commit}(\cdot, \cdot)$ used in steps 2 and 7, and assume that the message sent at the end of step 2 is (t, t).) These two Σ-protocols are run in parallel as one would do in a proof of equality in groups of known order to demonstrate that the preimage of y equals the first component of the preimage of y (cf. [16]). In fact, in all evaluations of ψ_M and ϑ (see steps 2 and 7) the argument of ψ_M and the first argument of ϑ are equal. This allows us to obtain a knowledge extractor for the Σ^+-protocol as follows. As the Σ-protocol uses essentially the same verification equations (step 7) as the Σ^+-protocol, the knowledge extractor can retrieve from a convincing prover a pseudo preimage $(\triangle c, \triangle s)$ of y under ψ_M and a pseudo preimage $(\triangle c, (\triangle s, \triangle \mathsf{s}))$ of y under ϑ. That is, we have

$$y^{\triangle c} = \psi_M(\triangle s) = h^{\triangle s} \tag{2}$$

$$\mathsf{y}^{\triangle c} = \vartheta(\triangle s, \triangle \mathsf{s}) = g^{\triangle s} g_1^{\triangle \mathsf{s}}. \tag{3}$$

As we run the two Σ-protocols in parallel as described above, the same integers $\triangle c$ and $\triangle s$ occur in (2) and (3). Now, as ϑ was chosen according to $\mathcal{D}_\vartheta(2, k)$, Theorem 3 implies that in (3) we must have $\triangle c \mid \triangle s$ and $\triangle c \mid \triangle \mathsf{s}$. Thus, (if we, e.g., additionally assert that $\gcd(\triangle c, |\,\mathrm{image}(\psi_M)|) = 1$) the knowledge extractor can compute a preimage $x \doteq \triangle s / \triangle c$ of y under ψ_M. Finally we note, that the Σ^+-protocol is not a proof of knowledge for the multi-exponentiation ϑ; the role of ϑ is just to enable the construction of the knowledge extractor for ψ_M.

It remains to discuss the statistical zero-knowledge property of the Σ^+-protocol. We have seen that for the knowledge extractor to work, the prover needs to provide to the verifier the values (t, s) and $(\mathsf{t}, (s, \mathsf{s}))$ that fulfill the verification equations in step 7. As these are the same verification equations as for the

$P((\psi_M, y), x)$		$V(\psi_M, y)$
		$\boxed{1.}$
		$(n, g) \leftarrow \mathcal{D}_{\mathrm{S}}(k)$
		$\rho \in_U [0, 2^k \lfloor n/4 \rfloor]$; $g_1 \doteq g^\rho \bmod n$
		$\vartheta(x_1, x_2) \doteq g_1^{x_1} g^{x_2} \bmod n$
$\boxed{2.}$	$\xleftarrow{(\vartheta, n)}$	
$\mathsf{x} \in_U [0, \lfloor n/4 \rfloor]$; $\mathsf{y} \doteq \vartheta(x, \mathsf{x})$		
$r \in_U [-2^{l_z} c^+ \triangle x, 2^{l_z} c^+ \triangle x]$; $t \doteq \psi_M(r)$		
$\mathsf{r} \in_U [-2^{l_z} c^+ \lfloor n/4 \rfloor], 2^{l_z} c^+ \lfloor n/4 \rfloor]$; $\mathsf{t} \doteq \vartheta(r, \mathsf{r})$		
Choose r_{y}; $\bar{\mathsf{y}} \doteq \mathrm{commit}(\mathsf{y}, r_{\mathsf{y}})$		
Choose r_{t}; $\bar{\mathsf{t}} \doteq \mathrm{commit}(\mathsf{t}, r_{\mathsf{t}})$		
	$\xrightarrow{(\bar{\mathsf{y}}, \bar{\mathsf{t}}, t)}$	$\boxed{3.}$
		$c \in_U \mathcal{C} = \{0, \ldots, c^+\}$
$\boxed{4.}$	$\xleftarrow{c}$	
$s \doteq r + cx$		
$\mathsf{s} \doteq \mathsf{r} + c\mathsf{x}$		
	$\xrightarrow{(s, \mathsf{s})}$	$\boxed{5.}$
$\boxed{6.}$	$\xleftarrow{\rho}$	
If $g_1 \not\equiv g^\rho \pmod{n}$, then halt.		
	$\xrightarrow{((\mathsf{t}, r_{\mathsf{t}}), (\mathsf{y}, r_{\mathsf{y}}))}$	$\boxed{7.}$
		If the equalities
		$\bar{\mathsf{y}} = \mathrm{commit}(\mathsf{y}, r_{\mathsf{y}})$; $\bar{\mathsf{t}} = \mathrm{commit}(\mathsf{t}, r_{\mathsf{t}})$
		$\psi_M(s) = ty^c$; $\vartheta(s, \mathsf{s}) \equiv \mathsf{t}\mathsf{y}^c \pmod{n}$
		hold, then output 1; else output 0

Fig. 1. Description Σ^+-Protocol

Σ-protocol, we can use the standard zero-knowledge simulation technique for the Σ-protocol. That is, given (ψ_M, y) and (ϑ, y) we can simulate tuples (t, c, s) and $(\mathsf{t}, c, (s, \mathsf{s}))$ fulfilling the respective verification equations. This approach works fine for *given* (ψ_M, y) and (ϑ, y), respectively. However, in the Σ^+-protocol (ϑ, y) are chosen within the protocol. Thus, for the Σ^+-protocol to be zero-knowledge, we additionally need to simulate the choices of y. Choices of y can be easily simulated when $\vartheta(x_1, x_2) = g_1^{x_1} g^{x_2}$ is formed correctly, i.e., $g_1 \in \langle g \rangle$. Then, over the choices of x, $\mathsf{y} = \vartheta(x, \mathsf{x}) = g_1^x g^{\mathsf{x}}$ is a uniform random element in $\langle g \rangle$ (we recall that $\mathsf{x} \in_U [0, \lfloor n/4 \rfloor]$ is statistically close to uniform on $\mathbb{Z}_{|g|}$). However, a dishonest verifier could choose a malformed ϑ such that $\mathsf{y} = \vartheta(x, \mathsf{x})$ would leak information about the preimage x and thus ruin the zero-knowledge property of the Σ^+-protocol. To overcome this problem, we use the commitment scheme $\mathrm{commit}(\cdot, \cdot)$ as follows. In step 2, the prover does not know whether ϑ is correctly chosen, and thus only sends the commitments to t and y instead of these values themselves. Then, in steps 5 and 6 the verifier convinces the prover that ϑ is correctly formed., i.e., that $g_1 \in \langle g \rangle$. To this ends, it sends the discrete logarithm ρ of g_1 with respect to g to the prover. Finally, when the prover is convinced of the correctness of ϑ, it opens the commitments from step 2 and reveals the values t and y. It is important that the verifier reveals the discrete logarithm ρ only after the prover has answered the challenge (steps 3 and 4). This is because for Theorem 3 to be applicable in the construction of the knowledge extractor, $\triangle s$ and $\triangle \mathsf{s}$ in (3) and thus s and s in step 4 of the protocol need to be computed by the prover without being given the discrete logarithm ρ. (In fact, Theorem 3 does not hold when one is given the discrete logarithms (with respect to some base element) of the g_i defining ϑ .) The simulator sketched above only works when the cardinality of $\mathcal{C}$ is bounded by some polynomial in the security parameter. This is because the simulator needs to be able to guess the challenge value for which it computes the simulated view. However, applying Damgaard's technique [19], we can make the Σ^+-protocol concurrent-zero knowledge just by additionally committing to t in step 2 and correspondingly open the commitment in step 6.

Theorem 4. *Let Ψ be a collection of simple-exponentiation homomorphisms and $c^+(k)$ be a positive integer parameter such that for any $\psi_M \in \Psi(k)$, $c^+(k)$ is smaller than the smallest prime dividing $|\,\mathrm{image}(\psi_M)|$. Then the Σ^+-protocol with challenge set $\mathcal{C} \doteq \{0, \ldots, c^+(k)\}$ is a computational proof of knowledge for $\mathcal{R}[\Psi]$. The computational validity property holds under the computational binding property of the commitment scheme and the Strong RSA assumption. The knowledge error is $1/|\mathcal{C}| + 1/p(k)$, where $p(\cdot)$ is an arbitrary polynomial.*

Let us conclude with a technical remark. Consider the DF scheme and the Σ^+-protocol computed for, e.g., common input a simple-exponentiation $\psi_M(x) = h^x$ and an image element y. We note that the knowledge extractors of both schemes rely on obtaining a pseudo preimage $(\triangle c, \triangle s)$ of y under ψ_M, i.e., $y^{\triangle c} = h^{\triangle s}$ such that the divisibility $\triangle c \mid \triangle s$ holds, which allows one to compute a preimage of y. (In fact, the Σ^+-protocol can guarantee the divisibility under weaker conditions.) Technically, this is the reason why the Σ^+-protocol works in all cases where the DF scheme is known to work. In particular, the Σ^+-protocol can also be used to obtain so called interval or range proofs [5]. Finally, the DF scheme is often considered under different conditions than formulated in Theorem 4, allowing one, e.g., only to prove that one knows b and z such that $y = bh^z$ with $b^2 = 1$. Given the foregoing observation, it is clear that such proofs can also be obtained using the Σ^+-protocol.

References

1. G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *Proc. 6th ACM Conference on Computer and Communications Security*, pp. 138–146. ACM press, Nov. 1999.
2. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology — CRYPTO 2000*, vol. 1880 of *Lecture Notes in Computer Science*, pp. 255–270. Springer Verlag, 2000.
3. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology — EUROCRYPT '97*, vol. 1233 of *LNCS*, pp. 480–494. Springer Verlag, 1997.
4. M. Bellare and O. Goldreich. On defining proofs of knowledge. In *Advances in Cryptology — CRYPTO '92*, vol. 740 of *Lecture Notes in Computer Science*, pp. 390–420. Springer-Verlag, 1992.
5. F. Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology — EUROCRYPT 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 431–444. Springer Verlag, 2000.
6. S. Brands. Rapid demonstration of linear relations connected by boolean operators. In *Advances in Cryptology — EUROCRYPT '97*, vol. 1233 of *Lecture Notes in Computer Science*, pp. 318–333. Springer Verlag, 1997.
7. J. Buchmann and H. C. Williams. A key-exchange system based on imaginary quadratic fields. *Journal Of Cryptology*, 1(2):107 – 118, 1998.
8. J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *Advances in Cryptology — EUROCRYPT 2001*, vol. 2045 of *LNCS*, pp. 93–118. Springer Verlag, 2001.
9. J. Camenisch and A. Lysyanskaya. An identity escrow scheme with appointed verifiers. In *Advances in Cryptology — CRYPTO 2001*, vol. 2139 of *LNCS*, pp. 388–407. Springer Verlag, 2001.
10. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology — CRYPTO 2002*, vol. 2442 of *LNCS*, pp. 61–76. Springer Verlag, 2002.
11. J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In *Advances in Cryptology — ASIACRYPT '98*, vol. 1514 of *LNCS*, pp. 160–174. Springer Verlag, 1998.
12. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Advances in Cryptology — CRYPTO 2003*, vol. 2729 of *LNCS*, pp. 126–144, 2003.
13. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology — CRYPTO '97*, vol. 1296 of *Lecture Notes in Computer Science*, pp. 410–424. Springer Verlag, 1997.
14. J. L. Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem.* PhD thesis, ETH Zürich, 1998. Diss. ETH No. 12520, Hartung Gorre Verlag, Konstanz.
15. R. Canetti, O. Goldreich, S. Goldwasser, and S. Micali. Resettable zero-knowledge. pp. 235–244. ACM Press, 2000.
16. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Advances in Cryptology — CRYPTO '92*, vol. 740 of *Lecture Notes in Computer Science*, pp. 89–105. Springer-Verlag, 1993.

17. R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocol*. PhD thesis, University of Amsterdam, 1997.
18. R. Cramer and I. Damgård. Zero-knowledge proof for finite field arithmetic, or: Can zero-knowledge be for free? In *Advances in Cryptology — CRYPTO '98*, vol. 1642 of *Lecture Notes in Computer Science*, pp. 424–441, Berlin, 1998. Springer Verlag.
19. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Advances in Cryptology — EUROCRYPT 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 431–444. Springer Verlag, 2000.
20. I. Damgård. On sigma-protocols. *Lecture Notes*, 2002.
21. I. Damgård and E. Fujisaki. An integer commitment scheme based on groups with hidden order. In *Advances in Cryptology — ASIACRYPT 2002*, vol. 2501 of *LNCS*. Springer, 2002.
22. I. Damgård and M. Koprowski. Generic lower bounds for root extraction and signature schemes in general groups. In *Advances in Cryptology — EUROCRYPT'02*, vol. 2332 of *Lecture Notes in Computer Science*, pp. 256–?? Springer Verlag, 2002.
23. C. Dwork, M. Naor, and A. Sahai. Concurrent zero knowledge. In *Proc. 30th Annual ACM Symposium on Theory of Computing (STOC)*, 1998.
24. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — CRYPTO '97*, vol. 1294 of *Lecture Notes in Computer Science*, pp. 16–30. Springer Verlag, 1997.
25. E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *Advances in Cryptology — EUROCRYPT '98*, vol. 1403 of *LNCS*, pp. 32–46. Springer Verlag, 1998.
26. M. Girault. An identity-based identification scheme based on discrete logarihtms modulo a composite number. In *Advances in Cryptology – EUROCRYPT '90*, vol. 473 of *Lecture Notes in Computer Science*, pp. 481–486. Springer-Verlag, 1991.
27. L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *Advances in Cryptology — EUROCRYPT '88*, vol. 330 of *Lecture Notes in Computer Science*, pp. 123–128. Springer Verlag, 1988.
28. P. MacKenize and M. K. Reiter. Two-party generation of DSA signatures. In *Advances in Cryptology — CRYPTO 2001*, vol. 2139 of *LNCS*, pp. 137–154. Springer Verlag, 2001.
29. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO '91*, vol. 576 of *Lecture Notes in Computer Science*, pp. 129–140. Springer Verlag, 1992.
30. G. Poupard and J. Stern. Security analysis of a practical "on the fly" authentication and signature generation. In *Advances in Cryptology — EUROCRYPT '98*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 422–436. Springer Verlag, 1998.
31. 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, Feb. 1978.
32. C. P. Schnorr. Efficient signature generation for smart cards. *Journal Of Cryptology*, 4(3):239–252, 1991.