Zero-Knowledge Arguments for Subverted RSA Groups

Dimitris Kolonelos[⋆1,2], Mary Maller[3], Mikhail Volkhov[⋆4]

[1] IMDEA Software Institute, Madrid, Spain
dimitris.kolonelos@imdea.org
[2] Universidad Politecnica de Madrid, Spain
[3] Ethereum Foundation, UK
mary.maller@ethereum.org
[4] The University of Edinburgh, UK
mikhail.volkhov@ed.ac.uk

Abstract. This work investigates zero-knowledge protocols in subverted RSA groups where the prover can choose the modulus and where the verifier does not know the group order. We introduce a novel technique for extracting the witness from a general homomorphism over a group of unknown order that does not require parallel repetitions. We then present a NIZK range proof for general homomorphisms as Paillier encryptions in the designated verifier model that works under a subverted setup. The key ingredient of our proof is a constant sized NIZK proof of knowledge for a plaintext. Security is proven in the ROM assuming an IND-CPA additively homomorphic encryption scheme. The verifier's public key can be maliciously generated and is reusable and linear in the number of proofs to be verified.

1 Introduction

A zero-knowledge proof consists of a prover that demonstrates to a verifier that a statement is true while revealing no information about the witness. Sigma protocols [58, 28] are a special type of zero knowledge proof that avoid expensive NP encodings and work naturally with many popular non-general relations. Sigma protocols enjoy negligible soundness-error in groups of known order. The story is different in groups of hidden order where negligible soundness can only be achieved by running $O(\lambda)$ sigma protocols in parallel [6, 60], thus multiplying the prover, proof size, and verifier costs by $O(\lambda)$.

In the common reference string model [11], a negligible soundness-error of hidden order group sigma protocols can be directly linked to hardness assumptions such as the strong-RSA [9, 40, 34, 27]. However, relying on hardness assumptions introduces an avenue for subversion: we can make no guarantees about any hardness assumption when a malicious prover corrupts the parameters of the

⋆ Most of the work was done while the first and third authors were interns at Ethereum Foundation.

hidden order group. For the prominent case of RSA-groups, i.e., multiplicative groups over the ring $\mathbb{Z}_N$ with $N = p \times q$, subversion is easy because one can compute the order of the group given the factorization, p and q.

To date, no natural[5] protocol for general homomorphism-languages with hidden order co-domain has negligible soundness-error (without repetitions), and at the same time does not rely on computational assumptions over the co-domain. Indeed, the task of constructing zero-knowledge proofs over subverted RSA-groups is exceedingly challenging; strictly more so than over traditional hidden order groups that are correctly formed. One can make no guarantees about how the modulus was generated and the Fiat-Shamir challenges can be continuously sampled until one from a malicious distribution is found.

Our question. We thus put forward the question:

Can one build a generalised sigma-protocol in subverted RSA-groups achieving negligible soundness-error without repetitions?

Our answer to this question is affirmative assuming a designated-verifier; we provide and prove secure a construction in the designated verifier model [33, 55]. This is excellent news because currently the only known method to construct RSA-groups is via a trusted setup [45]. Generating secure RSA parameters with a MPC is an extremely challenging task to realise in practice and to date no large scale RSA-MPCs have ever been completed. Our work thus provides an exciting avenue for numerous results in RSA-groups to remain applicable in subverted settings.

Subverted RSA groups are primarily interesting because they are a rare instantiation for groups of unknown order. The only known alternative for building hidden order groups is class groups, that can also be used to build ZKPs (e.g. [26]). In high contrast to RSA groups, cryptanalysists have only recently started focusing on class groups and we are still learning the best practices for choosing the parameters for implementation [38, 47, 50].

Further, the potential for N to be subverted is a delicacy which is rarely considered when using the additively homomorphic Paillier [54] encryption scheme. Here subverted parameters should be considered the default because participants can choose their encryption modulus N. Nonetheless, the handling of subverted parameters is a detail that is often overlooked in protocols that use Paillier. For example, in the influential paper by Hazay et al. [45], we see that they require a subversion resistant zero-knowledge range proof to realise their multiparty MPC but that none of their suggestions are subversion resistant. For more detail see the full version of the paper . As a second example, in the Damgard-Jurik voting scheme [36], they assume that a modulus N is generated by a trusted third party. If it were instead chosen by an election authority — which is a likelihood in real world systems — then this modulus could certainly be subverted. By colluding

[5] By 'natural' we mean a protocol that works directly for the underlying language and does not involve NP-reductions.

with just a single voter, the authority could provide verifying proofs of faulty encryptions and thus entirely decide the election result.

1.1 Our Contributions

In this paper we investigate zero-knowledge proofs under subverted RSA parameters. This is an extremely adversarial setting where the modulus N can be factorised by the prover but not by the verifier. We make no assumptions about ideal properties of the modulus: for example we can have that N is smooth or even that the prover knows the factorisation of N.

Our first contribution is a *new extraction method* for extracting a witness inside general homomorphisms. This extraction technique is completely new to the literature. We reify this technique through a designated-verifier protocol, named $\mathsf{DV}_{\mathsf{Prot}}$, which answers affirmatively the main question of this work introduced in the previous section. A substantial caveat for our extractor is that the challenges used by the sigma protocol are encrypted (under the designated verifiers secret key which importantly is independent from the potentially subverted N). Our extractor should fail if the adversary could decrypt the challenges, thus we describe the general extraction method and reduce the probability of the extractor failing to an adversary's advantage against IND-CPA. At the heart of our extraction method is an information-theoretical lemma about the distribution of the challenges extracted, which we prove to hold unconditionally. Exemplifying the extraction method, and as a stepping stone towards the second contribution, we explain how to make the $\mathsf{DV}_{\mathsf{Prot}}$ protocol practical, with reusable and potentially maliciously generated verifier's public key. Our main results are in the random oracle model however we also provide an optimised version in the generic group model.

Using our extraction technique we arrive at our second contribution, namely a zero-knowledge designated verifier *range proof* for Paillier encryptions under subverted modulus with negligible soundness, which we call $\mathsf{DVRange}_{\mathsf{Prot}}$. The protocol prevents a prover from encrypting a value outside the range even if the prover chooses the encryption key. Our proof is non-interactive (in the random oracle model) and has negligible soundness error without parallel repetitions. Security is proven in the RO model under the assumption that Paillier is IND-CPA. Our techniques for proving security are potentially of independent interest and described in more detail in Section 1.3. In the full version we show how our range proof can be applied for non-injective homomorphisms.

The verifier's public key has size $\mathcal{O}((\lambda + Q) \log N)$ for N a Paillier modulus, λ the security parameter, and Q the number of proofs the verifier will respond to. Our protocol does not require a common reference string; being DV the (designated) verifier inherently runs a setup to generate their potentially malicious key. To ensure zero-knowledge holds against all verifier keys we describe a non-interactive publicly verifiable key generation algorithm. In more detail, the verifier runs a publicly verifiable range proof to demonstrate that the verification public key (VPK) contains ciphertexts in the correct range. We apply amortisation techniques by Cramer et al. [30] (in Section 4.3) to minimise the cost of

this range proof. The key generation process is relatively expensive and can be avoided in scenarios where the verifier only needs to retrospectively prove honest behaviour by revealing the secrets behind their public key. Such scenarios are common in applications such as MPC with identifiable abort (ID-MPC, [46]).

1.2 Related Work

In composite order groups the standard Σ-protocol has knowledge error of only $1/2$ [6]. For a negligibly small extraction error one needs to run the protocol λ times in parallel (for λ the security parameter). This induces an $O(\lambda)$ multiplicative overhead. There are many different approaches in the literature to proving composite group statements more efficiently which we summarise here.

Proofs over groups of unknown order. An intensive line of work focuses on constructring efficient zero knowledge proofs for relations over groups where the order is unknown to all parties, however none would fit our context. The Fujisaki-Okamoto solution [40, 34, 27], the protocols of [18, 8] and the solution by Boneh et. al. [13] being computationally-sound are not sound in subverted RSA groups because having known (to the prover) group order prevents the underlying computational assumptions from holding. The protocol of [7] considers a model where the verifier has extra information about the witness[6]. The protocol from [5] was later cryptanalized [49]. For specific relations, [36, 35] present efficient protocols where the prover knows the order of the group, however they are sound only when the RSA group is correctly formed. The work of Cramer et. al. [29, 30] presents a transformation that allows the protocol to have negligible soundness error, yet only when proving λ statements simultaneously. For a single proof it cannot be applied. Finally, Bangerter et al. [6] and Terelius et al. [60] show a lower bound on soundness error for constant round sigma-like protocols in the standard model (no CRS, no RO), that translates to $1/2$ for common parameters.

Proving RSA relations with zk-SNARKs. Many zk-SNARK proof systems are both general enough to encode any NP circuit and efficient enough to be used in practice. Thus we can prove relations about subverted RSA groups by representing them with an arithmetic circuit or similar. Ozdemir et al. implement an RSA based accumulator inside a SNARK [53]. Their work improves upon xJsnark [48]. Using Ozdemir et al.'s BigNat library[7] we compute the size of the Paillier knowledge-of-plaintext circuit at 80 million gates for 2048 bit N. This is towards the upper end of what can feasibly be computed with a SNARK. To the best of our knowledge the biggest circuits currently in production have about 100-million constraints and take minutes to compute even on specialist hardware[8]. Our work does not require a reduction to NP and therefore we avoid this

[6] For some relations (e.g. Paillier Encryptions) this can lead to fully reconstructing the witness.

[7] https://github.com/alex-ozdemir/bellman-bignat

[8] https://research.protocol.ai/sites/snarks/

prover overhead. Our approach also avoids the significant challenge of auditing an 80 million gate circuit.

Range proofs in the RSA setting. In this work we present range proofs for RSA-like relations (e.g Paillier encryption), or generally (additive) homomorphisms with unknown co-domain. Variations of basic Schnorr-like Σ-protocol exist for RSA-like range relations [39, 25, 20, 34, 18, 12, 27]. Boudot [14] presents the first range proof for general range $[L, R]$ with slackness 1 (i.e. the message lies exactly in $m \in [0 \dots R]$ as opposed to some extended range $m \in [0 \dots \delta R]$). Further [14] uses a so-called four-squares integer decomposition property, a technique which is later used and improved in [51, 44, 62]. None of these works consider a subverted modulus. In fact they are computationally sound and make assumptions about the RSA group, thus they do not work in subverted settings.

Proofs of correct form of moduli. An orthogonal to the above line of work intends to prove that the group itself is not subverted [61, 41, 19, 10, 3, 42], meaning that the modulus N of the RSA group has some beneficial property; for example is square-free, a product of two primes, a product of equally-sized primes, a Blum integer or a product of two safe primes, etc. Other works consider proving that moduli are correctly formed in the context of specific applications as password-based key agreement [23] or threshold ECDSA signatures [21]. All these solutions require repetitions to reach a negligible soundness-error. Furthermore, to apply computationally-sound protocols for general homomorphisms (such as Fujisaki-Okamoto) over the group afterwards, one needs to prove that the RSA group is a product of two safe primes. Only [19] ensures this, however it has high costs and does not avoid the $O(\lambda)$ parallel repetitions.

1.3 Overview of Techniques

In this work we design efficient designated-verifier ZK protocols for knowledge and range of RSA group homomorphisms, which have negligible soundness error without repetitions even when the group is maliciously chosen. The main unifying ideas of all our techniques are (1) an alternative approach to Σ-protocols' witness extraction and (2) a careful realisation through homomorphic encryption with respect to (also potentially subverted) verifier's modulus, which allows hiding protocol challenges from the prover in a way that prevents lower-bound attacks of [6, 60].

Let $\psi : \mathcal{D} \to \mathbb{H}$ be a group homomorphism where $\mathbb{H}$ is an RSA-related group, such as exponentatiations $w \mapsto g^w$ over $\mathbb{H} = \mathbb{Z}_N^*$ (or multiexponentations), or Paillier encryption $(w, r) \mapsto (N+1)^w h^r$. We wish to design an efficient argument of knowledge of w such that $Y = \psi(w)$, and $w \in \{0 \dots R\}$ for $R \in \mathcal{D} \subset \mathbb{Z}$.

Σ-protocol soundness. The classic Σ-protocol for proving knowledge of w such that $Y = \psi(w)$, described in Fig. 1, is only secure if elements from $\mathcal{D}$ are

invertible. The standard special-soundness extractor behaves as follows: given two successful transcripts with the same first message $(a, c, s), (a, c', s')$ such that $aY^c = \psi(s)$ and $aY^{c'} = \psi(s')$ and $c \neq c'$ it combines the two:

$$aY^c = \psi(s) \quad aY^{c'} = \psi(s')$$

from which it gets $Y = \psi(s-s')^{(c-c')^{-1}} = \psi((s-s')(c-c')^{-1})$. When $\mathbb{H}$ is a group of public prime order p, as in case of the Schnorr protocol, this strategy always succeeds, because $(c-c')^{-1} \mod p$ is efficiently computable. However, when $\mathbb{H}$ is a maliciously chosen RSA group, the extractor has two problems. First, it does not know the order of the group and thus can only compute $(c-c')^{-1}$ when $c - c' = 1$ (in this trivial case $Y^1 = \psi(s - s')$, and $s - s'$ is the witness). This limitation is similar to the hardness of taking roots in groups of unknown order. Second, some inverses $(c-c')^{-1}$ do not exist because it is possible that $\gcd(c - c', \mathrm{ord}(\mathcal{D})) \neq 1$ for a maliciously chosen N.

In fact the impossibility results of [6, 60] show that the above extractor fails for any group $\mathbb{H}$ whose order is not publicly known, such as RSA groups.

A generalized extraction lemma. Towards constructing an efficient protocol with negligible soundness error, our starting point is a generalized extraction approach. Assume that our extractor has $M \geq 3$ successful transcripts[9] $\{(a, c_i, s_i)\}_{i=1}^{M}$ such that:

$$aY^{c_1} = \psi(s_1) \quad aY^{c_2} = \psi(s_2) \quad \ldots \quad aY^{c_M} = \psi(s_M)$$

then combining the first with the rest we get the equivalent:

$$Y^{c_2-c_1} = \psi(s_2 - s_1) \quad \ldots \quad Y^{c_M-c_1} = \psi(s_M - s_1)$$

Now if $\gcd(c_2 - c_1, \ldots, c_M - c_1) = 1$ then we can always compute coefficients $\gamma_2, \ldots, \gamma_M$ such that $\gamma_2(c_2 - c_m) + \ldots + \gamma_M(c_2 - c_M) = 1$, which means:

$$Y^1 = Y^{\gamma_2(c_2-c_1)+\ldots+\gamma_2(c_M-c_1)} = \psi(\gamma_2(s_2 - s_1) + \ldots + \gamma_M(s_M - s_1))$$

so $s^* = \gamma_2(s_2 - s_1) + \ldots + \gamma_M(s_M - s_1)$ is a valid pre-image.

This extraction technique succeeds as long as $\gcd(c_2 - c_1, \ldots, c_M - c_1) = 1$. If we had an honest prover and the c_i challenges were truly random and independent, then well-known results from mathematics show that this happens with probability $1/\zeta(M)$, for ζ being the zeta Riemann function. This probability is overwhelming (negligibly close to 1) as a function of M.

However, a malicious prover may choose not to respond upon receiving certain challenges c, so that $\gcd(c_2 - c_1, \ldots, c_M - c_1) \neq 1$. As an example they can choose only to answer even challenges. The natural conclusion is that for this generalized extraction to work we need the (adversarial) prover to be oblivious to the challenges it answers.

[9] Extracting k successful transcripts is no harder than extracting 2 [1].

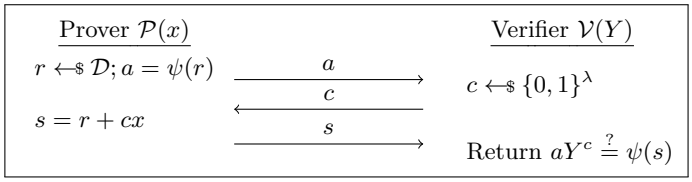

Fig. 1. A Σ-protocol for the relation containing elements (Y, w) such that $Y = \psi(w)$, where ψ is a general homomorphism. This protocol is only knowledge sound if elements from $\mathcal{D}$ are invertible.

Designated verifier techniques. We bootstrap the protocol of Fig. 1 to a secure one (with negligible soundness error) in the Designated-Verifier model.

One of our key observations is that in the Designated-Verifier setting we can hide the challenge c from the malicious prover by encrypting it with a homomorphic encryption scheme for verifier's public key. Then the prover computes the response to the challenge "blindly", using additive homomorphism of the encryption scheme. The verifier, who possesses the secret key of the encryption, decrypts the response normally in order to retrieve the plaintext response of the Σ-protocol. For this we need the verifier to hold the corresponding secret key, which must be kept secret from the prover. The public key of the designated verifier (VPK) is merely the pk of the encryption scheme and the ciphertext ct of the encrypted challenge. The idea of encrypting a (single) challenge in the designated-verifier public key appears in previous DV protocols [33, 24]

To prove the existence of an extractor we require M answers with different challenges from the prover. This is clearly not possible when we encrypt just a single challenge; but we also cannot do it even when we encrypt M challenges — the prover can potentially choose only to answer with respect to the first challenge. What we require is an exponential sized challenge space. For this, we encrypt λ sub-challenges that are chosen uniformly at random: $\mathsf{ct}_1 = \mathsf{Enc}(c_1), \ldots, \mathsf{ct}_\lambda = \mathsf{Enc}(c_\lambda)$ and add them to the public key. Then the value $\mathcal{P}$ responds to is a random $(0, 1)$ linear combination of $\{c_i\}$: $c = \sum_{i=1}^{\lambda} b_i c_i$ where $\boldsymbol{b} = (b_1, \ldots, b_\lambda)$ a random bitstring-challenge sampled by the verifier, which gives rise to exponential $\mathcal{C}$.

To prove soundness, the core of our security proof is an information-theoretical lemma showing that after $M = \mathsf{poly}(\lambda)$ linear combinations have been extracted, the probability of $\{\boldsymbol{b}_i \boldsymbol{c}^\top\}_{i=1}^{M}$ being coprime is overwhelming (assuming that c_i's were uniformly sampled and independent during the setup).

DV with a reusable VPK. A common issue in the Designated-Verifier model is that a prover, after seeing whether some proofs of its choice verify or not, can learn information about the VPK's structure and break soundness. This is the analogue of IND-CCA security of encryption schemes. Intuitively, the verification oracle behaves in a similar manner to a decryption oracle. Additive homomorphic encryption schemes cannot be IND-CCA and thus an attacker

could use a verification oracle to learn information about vpk. We overcome this by adding $Q = \mathsf{poly}(\lambda)$ statistical blinding factors $e_1, \ldots, e_Q$ encrypted in the VPK. At each proof one of these factors is added to the linear combination and thus statistically blinds it; thus Q is maximum number of verification queries the prover can ask. The CRS size is thus $O(1)$ per proof.

1.4 Comparison with Alternative Approaches

To the best of our knowledge, this work is the first that deals with the problem of constructing zero-knowledge proofs in subverted RSA groups. On the other hand, the literature provides numerous techniques on constructing zero-knowledge proofs in non-subverted RSA groups. It is challenging to compare the efficiency of our scheme directly against the state-of-the-art for non-subverted solutions because this would require fully researching how to convert multiple solutions into the subverted setting. Instead we here briefly justify our techniques against two possible alternative approaches that provide partial solutions to the problem.

Combine with an auxiliary group of unknown order. A possible approach to constructing a sound *proof of knowledge* in the subverted RSA setting would be to combine the simple protocol of Fig. 1 with a proof of a preimage in an established group of unknown order. That is, generate an unknown order group $\mathbb{G}$, commit to the same preimage $\mathsf{Commit}(w)$ and send the commitment to the verifier. Then compose in parallel a proof of knowledge for $\mathsf{Commit}(w)$ (over $\mathbb{G}$) and the protocol of Fig. 1 (over the subverted RSA group). The Fujisaki-Okamoto extraction technique [40, 34, 27] gives negligible knowledge error and avoids the need for λ repetitions. However, this solution either requires a private-coin trusted setup in case an RSA group is used as the auxiliary group of unknown order, or must rely on class groups [16]. Solutions relying on class groups are outside the scope of this work (see Introduction).

Range proof with an auxiliary prime order group. For the *range proof* problem for the preimage w of a homomorphism, $Y = \psi(w)$ with $0 < w < R$, one possible approach is the following. Generate an auxiliary prime order group $\mathbb{G}$ and commit to the preimage, $\mathsf{Commit}(w)$ over this group (e.g. via Pedersen commitment). Then run in parallel the protocol of Fig. 1 for $\psi(w)$ in the subverted RSA group and a simple Schnorr protocol for the commitment on $\mathbb{G}$, to prove that $\mathsf{Commit}(w)$ and $\psi(w)$ contain the same value. Afterwards one can use a range proof protocol in the prime order group [17, 26] to prove the range of w. The main benefit here is that due to progress on range proofs over prime order groups, the actual range proof block is concretely efficient.

This solution, however, inherits the soundness-error (and thus the required iterations) of the protocol of Fig. 1. That is $1/2$ for general homomorphisms $1/\mathsf{poly}(\lambda)$ for some specific special homomorphisms such as the (original) Paillier

Encryption [6]. This leads to an overhead of $O(\lambda)$ and $O(\lambda/\log(\lambda))$ respectively, due to the repetitions needed.

Our work concerns with the former category, general non-special homomorphisms (such as ElGamal-Paillier) where the overhead is $O(\lambda)$, and provides a a truly unique perspective on how to decrease their asymptotic efficiency to $O(1)$ which was not previously known to be possible. We achieve this by providing and proving secure an alternative extraction technique together with an information theoretical lemma that have no dependence on parallel executions.

2 Preliminaries

2.1 Notation

We denote the security parameter with λ; $\mathsf{poly}(\lambda)$ is any positive $f(n) = O(\mathsf{poly}(n))$, and $\mathsf{negl}(\lambda)$ is a negligible positive function. With $[a, b]$ we denote the set $\{a, a+1, \ldots, b\}$, and with $[n]$ we denote $[1, n]$. Similarly with $[\![n]\!]$ we denote the set $[-\lfloor \frac{n}{2} \rfloor \ldots \lfloor \frac{n}{2} \rfloor]$. Adversaries are assumed to be stateful unless stated otherwise.

$\mathbb{Z}_n$ is the additive group of order n. We often explicitly consider interval $[\![n]\!]$ as the integer encoding for $\mathbb{Z}_n$. $\mathbb{Z}_n^*$ is the multiplicative group of all integers in $[\![n]\!]$ coprime with n. With $\phi(\cdot)$ we denote the Euler's totient function. $\mathcal{U}_S$ stands for uniform distribution on S as a finite set (e.g. $\mathcal{U}_{\mathbb{Z}_p}$); $\mathcal{U}_{[L,R]}$ is a uniform distribution on $[L, R]$, and $\mathcal{U}_R$ is a shorthand for $\mathcal{U}_{[0,R]}$. In general we denote with capital letters, e.g. Y, elements of the RSA group. In bold we denote vectors (e.g. $\boldsymbol{s}$) and matrices (e.g. $\boldsymbol{A}$).

2.2 Homomorphic Encryption Schemes

In this work we engage public-key encryption schemes that have additively homomorphic properties. That is an encryption scheme is called additively homomorphic if for every $\mathsf{pk} \in \mathcal{PK}$ and $m_1, m_2 \in \mathcal{M}$, $\mathsf{Enc}_{\mathsf{pk}}(m_1) \cdot \mathsf{Enc}_{\mathsf{pk}}(m_2) = \mathsf{Enc}_{\mathsf{pk}}(m_1 + m_2)$, where '$\cdot$' is a ciphertext space operation. In the rest we assume that the message space $\mathcal{M}$ of the additively homomorphic schemes we refer to forms a ring. Some known examples of additively homomorphic encryption are the Paillier cryptosystem and its variants [54, 36, 31, 15] in the RSA setting, the Castagnos-Laguillaumie cryptosystem over class groups [22] and schemes from lattices [43, 56]. Notably, no additively homomorphic public-key cryptosystems from groups of prime order exist.[10]

Paillier encryption scheme. We briefly recall the Paillier public key encryption scheme [54], and refer the reader to our full version for more details .

[10] Although the lifted ElGamal cryptosystem (alike ElGamal but the message is lifted in the exponent) is additively homomorphic, the decryption is not polynomial-time, unless one restricts the message space to polynomial size. This makes it unsuitable for most applications.

$\mathsf{KeyGen}(1^\lambda)$: sample p, q primes of the size λ and set $N = p \cdot q$. Compute $d = \phi(N)^{-1} \mod N^2$. Output $\mathsf{pk} = N$ and $\mathsf{sk} = (d, \phi(N))$.
$\mathsf{Enc}_{\mathsf{pk}}(m)$: sample uniformly $r \leftarrow\$ \mathbb{Z}_N^*$ and output $\mathsf{ct} = (N+1)^m r^N \mod N^2$.
$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$: compute $c = (\mathsf{ct}^{\phi(N)} - 1)d \mod N^2$ and return $m = \frac{c}{N}$.

2.3 Homomorphisms and Efficient Σ-protocols

Let $\psi : \mathcal{D} \to \mathbb{H}$ be a homomorphism between a domain $\mathcal{D}$ (group or ring), and an output group $\mathbb{H}$ (e.g. RSA). When $Y = \psi(w)$, we call w a witness, and Y an instance.

A pair $(v, u) \in \mathbb{Z} \times \mathcal{D}$ is called a *pseudo-preimage* (PP) for instance $Y = \psi(x)$, if $Y^v = \psi(u)$ holds [7, 5], where v is called a degree of a given PP. Pseudo-preimages naturally occur in Σ-protocols: the extractor usually transforms two transcripts for the same commitment a ($Y^{c_i} a = \psi(s_i)$, $i \in 1, 2$) into a single PP by dividing the equations: $Y^{c_1 - c_2} = \psi(s_1 - s_2)$, thus $(c_1 - c_2, s_1 - s_2)$ is a PP.

In prime-order groups ($|\mathbb{H}| = p$) knowledge of PP implies knowledge of preimage, since inverses in $\mathbb{Z}_p$ are efficiently computable. In groups where the order is not prime or even unknown to $\mathcal{V}$ (e.g. in Paillier $\mathbb{H} = \mathbb{Z}_{N^2}^*$) there is another way to extract a proper preimage, but from *two* pseudo-preimages: given (v_1, u_1), (v_2, u_2) with $\gcd(v_1, v_2) = 1$ for any Y we can use the so-called called "Shamir's trick". Given $(v_1, u_1), (v_2, u_2)$ s.t. $Y^{v_i} = \psi(u_i), i \in \{1, 2\}$, it first checks if $\gcd(v_1, v_2) \neq 1$ and aborts if not. Then it computes Bezout coefficients — integers γ, δ such that $\gamma v_1 + \delta v_2 = 1$, and returns $u := \gamma u_1 + \delta u_2$. This extractor succeeds, since given $Y^{v_i} = \psi(u_i)$, $Y = Y^{\gamma v_1 + \delta v_2} = \psi(u_1 \gamma + u_2 \delta) = \psi(u)$.

Special homomorphisms. In [7], following Cramer [28], the homomorphism $\psi : \mathcal{D} \to \mathbb{H}$ is called *special* if for any instance Y one can easily find a *non-trivial* PP $(\hat{v}, \hat{u})$ of Y (non-trivial means $\hat{v} \neq 0 \mod |\mathbb{H}|$). Examples of special homomorphisms include Schnorr-like homomorphism[11] $\psi : \mathbb{Z}_q \to \mathbb{Z}_p^*$, $\psi : x \mapsto h^x$ with $\mathsf{ord}(h) = q$, $q \mid (p-1)$ and Paillier homomorphism[12].

For special homomorphisms it is sometimes possible to build Σ-protocols with non-binary challenge spaces (and thus small soundness error) by applying Shamir's trick to just one extracted PP, and the special PP. This is the best known method of extraction for Paillier in the honest setting. However, in the subverted N scenario it does not work, and binary challenges are still optimal. This is because of the GCD condition in Shamir's trick: $\mathcal{A}$ can choose N to maximize $\Pr[\gcd(c_1 - c_2, N) \neq 1]$ (N is a degree of Paillier special PP); with binary challenges $c_1 - c_2 = 1$, and GCD is always 1. Other variants of Paillier (e.g. ElGamal-Paillier [31, 15]), are not known to be special, thus even the above extraction technique fails unless challenges are binary ($c_1 - c_2 = 1$).

[11] Its special PP is $(q, 0)$, since $Y^q = \psi(0)$; and the PP is non-trivial: $q \neq 0 \mod p$.
[12] From $Y = G^m r^N$ we can derive $Y^N = (G^m r^N)^N = G^0 (G^m r^N)^N$, so $(N, (0, Y))$ is a pseudo-preimage of degree N (and $N \neq 0 \mod \phi(N^2)$).

2.4 Designated-Verifier Arguments of Knowledge

We assume some familiarity with the notion of interactive arguments of knowledge and their standard security properties (completeness, knowledge-soundness, and zero-knowledge). In the *designated verifier* (DV) model, additionally to $\mathcal{P}, \mathcal{V}$ programs we claim existence of a KeyGen routine that the verifier uses to create verifier's public key (VPK). This public key is then used to interact with this verifier only, and can potentially be reused multiple times. The formal definitions of completeness, soundness with reusable VPK, and honest verifier zero-knowledge under a malicious VPK are deferred to the full version.

3 Our Extraction Technique

In this section we state and prove two lemmas about our novel extraction method. The first is a generalised extraction lemma, Lemma 1, that describes how to extract a witness given M accepting transcripts such that the gcd of the challenges is 1. Our second lemma, Lemma 2, is the core information-theoretical lemma behind the security of our construction, which argues about this probability of random challenges being coprime.

3.1 The Generalized Extraction Lemma

We consider the three-round public-coin protocol of Figure 1 where transcripts have the form (a, c, s). In Lemma 1 we design an extractor that, given M valid transcripts on the same first message, always succeeds provided that $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$. The following is proven in the full version.

Lemma 1. *Let $\mathcal{T} = \left\{(a, c^{(i)}, s^{(i)}\right\}_{i=1}^{M}$ be a collection of $M \geq 3$ successful transcripts for the relation $\mathcal{R}_{\mathsf{Hom}}$ and input Y, $aY^{c^{(i)}} = \psi(s^{(i)})$, such that $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$. Then there exists a PPT extractor Ext that outputs w such that $Y = \psi(w)$ with probability 1.*

3.2 Our Core Coprimality Lemma

The above generalized extraction technique is effective conditioned on the fact that differences of the challenges in the extracted transcripts are coprime, $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$. However, this cannot be guaranteed for any malicious prover. This stems from the fact that an adversarial prover can manipulate the $c^{(i)}$'s by selectively choosing to answer successfully or not, after receiving $c^{(i)}$.

Intuitively, we would like the adversary to answer independently of $c^{(i)}$. Then for sufficiently large $M = \mathsf{poly}(\lambda)$, $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$ would hold. To this end we let the challenges consist of two factors: the challenge is $e = \boldsymbol{b}\boldsymbol{c}^T$ where $\boldsymbol{b}$ is sampled during the protocol execution and $\boldsymbol{c}$ is a vector that is uniformly random from the point of view of the adversary. The adversary can manipulate $\boldsymbol{b}$ because $\boldsymbol{b}$ is chosen during the protocol, but $\boldsymbol{c}$ cannot

be manipulated. Looking ahead, in Section 4 we realize this technique in the designated-verifier setting.

In Lemma 2 we prove an information-theoretical statement which is at the core of our construction. The distribution of values output by our extractor depend nontrivially on some adversarial matrix $\boldsymbol{B}$: the matrix of all $\boldsymbol{b}$ that the adversary chooses to answer successfully. Because there are no computational restrictions on how an adversary might choose $\boldsymbol{B}$, we require that for any $\boldsymbol{B}$ the extractor will succeed with high probability. Lemma 2 is new to this work and as far as we are aware there are no similar results in the literature.

How to interpret the lemma. As previously noted, Lemma 2 aims to information-theoretically prove that M extracted accepting transcripts (on the same first message) have coprime challenges where each challenge is $\boldsymbol{b}^{(i)}\boldsymbol{c}^T$. From the point of view of the adversary $\boldsymbol{b}$ is known but $\boldsymbol{c}$ is not, and assumed uniformly random.

To make the applicability of the lemma more clear we briefly recall (omitting the non-relevant details) the extractor of [2] (that generalizes [32]) which obtains M accepting transcripts, with the same first message, for any Σ-protocol.

Let $\boldsymbol{H}$ be the binary matrix where the rows represent the first messages $\alpha_1 = \psi(r_1), \alpha_2 = \psi(r_2), \ldots, \alpha_{|\mathcal{D}|} = \psi(r_{|\mathcal{D}|})$ and the columns represent the different challenges $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{2^\lambda}$. The position $\boldsymbol{H}_{i,j}$ is 1 if the adversary can answer successfully on $\alpha_i, \boldsymbol{b}_j$ and 0 otherwise. The extractor works as follows:

- Probes different positions of $\boldsymbol{H}$ until it finds a 1.
- If it finds a first 1 it continues sampling uniformly in the same row until it finds $M-1$ more 1's (or terminates with some specific probability).

Attema et. al. [2] show that this extraction strategy outputs M accepting transcripts in expected polynomial time.

Assume that the extractor succeeds in outputting the M transcripts from some row i. Then $\boldsymbol{B}$ (in matrix form) represents all the $\boldsymbol{b}_j$'s of this row that have 1. Similarly, $\boldsymbol{B'}$ (also in matrix form) represents all the $\boldsymbol{b}^{(j)}$'s of the row that were sampled (uniformly) by the extractor, contained 1 and thus gave an accepting transcript. Lastly, for the lemma to be applied we need that $\boldsymbol{B}$ has exponentially large number of rows $> 2^\lambda/\mathsf{poly}(\lambda)$. Conditioned on the fact that the extractor terminates in (expected) polynomial time this holds, otherwise the probability of the extractor to find M 1's in the row (in poly-time) would be negligible. Clearly then, $\boldsymbol{B'}$ is a polynomially sized sub-matrix of $\boldsymbol{B}$.

We highlight that the matrix $\boldsymbol{H}$ represents the malicious prover's strategy and it is clearly adversarially chosen, thus so is $\boldsymbol{B}$. For this it is important that the lemma holds for any arbitrary $\boldsymbol{B}$. This makes the lemma and its proof highly non-trivial.

Lemma statement. Lemma 2 proves the following. Assume *any* exponentially-large $(2^\lambda/\mathsf{poly}(\lambda))$ space $\boldsymbol{B}$ of binary vectors with λ coordinates. Then if we sample uniformly $M = \mathsf{poly}(\lambda)$ vectors from this space $\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(M)} \xleftarrow{\$} \boldsymbol{B}$ and λ

uniformly random values (from an exponentially large space) $\boldsymbol{c} := (c_1, \ldots, c_\lambda) \leftarrow\!\!\$ \left(\llbracket 2^\lambda \rrbracket\right)^\lambda$ we get that their inner products $\boldsymbol{b}^{(1)}\boldsymbol{c}^T, \ldots, \boldsymbol{b}^{(M)}\boldsymbol{c}^T$ are coprime, except with negligible probability. This then generalizes to our final result that concerns with the differences $\{\boldsymbol{b}^{(i)}\boldsymbol{c}^T - \boldsymbol{b}^{(1)}\boldsymbol{c}^T\}_{i=2}^{M}$ being coprime.

Crucially, this holds for any space $\boldsymbol{B}$ as long as it is sufficiently large.

Lemma 2. *Let* $\boldsymbol{B}$ be any $(\epsilon' 2^\lambda) \times \lambda$ *binary matrix consisting of* $\epsilon' 2^\lambda$ *distinct binary rows, with* $\epsilon' > 1/\mathsf{poly}(\lambda)$*. Sample:*

- $M = \mathsf{poly}(\lambda)$ *rows of* $\boldsymbol{B}$*,* $i_k \leftarrow\!\!\$\, [1, \epsilon' 2^\lambda]$ *for* $k = 1, \ldots, M$*, and set*

$$\boldsymbol{B'} = (\boldsymbol{b}^{(1)}\ \boldsymbol{b}^{(2)} \ldots \boldsymbol{b}^{(M)})^T := (\boldsymbol{b}_{i_1}\ \boldsymbol{b}_{i_2} \ldots \boldsymbol{b}_{i_M})^T$$

- λ *uniformly random values,* $c_i \leftarrow\!\!\$\, \llbracket 2^\lambda \rrbracket$ *for* $i = 1, \ldots, \lambda$*, and set*

$$\boldsymbol{c} = (c_1\ c_2\ \ldots\ c_\lambda)$$

and set $(e^{(1)} \ldots e^{(M)})^T = \boldsymbol{B'c}$*. Then:*

$$\Pr[\gcd(e^{(2)} - e^{(1)}, \ldots, e^{(M)} - e^{(1)}) = 1] = 1 - \mathsf{negl}(\lambda)$$

the probability is over the choices of $\boldsymbol{c}, \boldsymbol{B'}$*.*

Due to space limitations the full proof is deferred to full version.

4 Designated Verifier Proofs of Knowledge for General Homomorphisms

In this section we design a designated verifier argument of knowledge for an opening to a general homomorphisms. We prove that there is a negligible soundness error assuming an additively homomorphic encryption scheme that is CPA secure. Zero-knowledge holds even under subverted parameters and it does not require a common reference string. Our proofs consist of 6 elements and can be made non-interactive using the Fiat-Shamir transform.

We show in Section 5 that knowledge of an opening for a general homomorphism is powerful enough to build range proofs for ciphertexts over a subverted encryption key. For now we focus on the simpler general relation

$$\mathcal{R}_{\mathsf{Hom}} = \{\ \psi, A \,\big|\, w : Y = \psi(w)\ \}$$

where $\psi : \mathcal{D} \to \mathbb{H}$ and $\mathbb{H}$ is a group parametrized by a maliciously generated RSA modulus N (for example $\mathbb{Z}_N^*$ or $\mathbb{Z}_{N^2}^*$). Although not directly in our scope, the techniques of this sections also apply to any group of unknown order.

4.1 The Designated-Verifier Protocol

We are now ready to present our designated verifier zero-knowledge proof system for $\mathcal{R}_{\mathsf{Hom}}$ where ψ is any additive group homomorphism.

The public-coin interactive DV protocol for $\mathcal{R}_{\mathsf{Hom}}$ is run between a prover and the verifier. The protocol is a modification of the sigma protocol in Fig. 1 to ensure soundness even for subverted RSA groups. One of the key observations is that in the Designated-Verifier setting we can hide the challenge from the malicious prover. We can thus assume that *all* the challenges answered are independent, provided that they are sampled independently by the verifier. In order to hide the challenges from the prover they are encrypted with a public key homomorphic encryption scheme. These encrypted challenges are provided in advance inside the verifier's public key.

Then if these encrypted challenges are linearly combined with fresh (binary) challenges, sampled during the actual execution one can directly apply the extraction techniques of Section 3 (Lemma 1 and Lemma 2). The linear combination is performed homomorphically through the ciphertexts.

The full protocol is presented in $\mathsf{DV}_{\mathsf{Prot}}$. For ease of presentation, we first describe our protocol incrementally: with respect to a trusted setup that always outputs $(\mathsf{vpk}, \mathsf{vsk})$ honestly and without allowing any reusability of it; then in the next sections we incrementally present how to achieve these properties.

Our construction makes use of any additive additively homomorphic encryption scheme with message space $\mathcal{M}$, randomness space $\mathcal{R}$, and ciphertext space $\mathcal{CT}$ such that $\mathcal{CT}$ forms a multiplicative group. For simplicity we will assume AHE to be standard Paillier w.r.t. N_{pk}, and $\mathcal{M}$ to be the ring $\mathbb{Z}_{N_{\mathsf{pk}}}$ for an integer N_{pk}, although our scheme works with any AHE and ring $\mathcal{M}$.[13]

First the key generation algorithm creates a verification key: it chooses an encryption key pair $(\mathsf{pk}, \mathsf{sk})$ and sets the verifier's secret key to $\mathsf{vsk} = \mathsf{sk}$. It then samples uniformly λ values, $c_1, \ldots, c_\lambda \xleftarrow{\$} [\![2^\lambda]\!]$ (denote $\boldsymbol{c} = (c_1, \ldots, c_\lambda)$) and encrypts them under pk, $\mathsf{ct}_1 = \mathsf{Enc}_{\mathsf{pk}}(c_1), \ldots, \mathsf{ct}_\lambda = \mathsf{Enc}_{\mathsf{pk}}(c_\lambda)$. In Section 4.3 we describe a protocol by which the verifier proves that their vpk is well formed, ensuring that we achieve zero-knowledge under subverted vpk (hence without trusting the designated verifier for the key setup).

The protocol then proceeds in 5 moves which we detail in Fig. 2. The prover essentially proves that $Y = \psi(w)$ by sending $a = \psi(r)$; an encryption S of $(r + cw)$; and a proof (T, u_1, u_2, u_3) that the prover knows the contents of S. The additional steps 4 and 5 that prove knowledge of the preimage of S are there so that we can technically avoid passing vsk to the extractor to compute s. Instead they can extract s from the additional protocol of these steps. This explains why d is sampled from the exponentially big challenge space – the modulus in question (chosen by the verifier and extractor) is trusted for soundness.

As usual in public-coin protocols, the interactive $\mathsf{DV}_{\mathsf{Prot}}$ can be transformed into a non-interactive one applying the Fiat-Shamir transformation (in the random oracle model).

[13] As long as all elements in $[\![2^{\lambda+1}]\!]$ have a multiplicative inverse in $\mathcal{M}$.

$\mathcal{V}.\mathsf{KeyGen}(1^\lambda)$**:** Generate a VPK:

- Sample a key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{AHE.KeyGen}(1^\lambda)$ with $|\mathcal{M}| > 2^{2\lambda+\log\lambda}|\mathcal{D}|$.
- Sample challenges uniformly: $c_1, \ldots, c_\lambda \xleftarrow{\$} [\![2^\lambda]\!]$
- Encrypt them: $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(c_i)$ for each $i \in [1, \lambda]$.
- Return $\mathsf{vpk} = (\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\lambda)$, $\mathsf{vsk} = \mathsf{sk}$.

$\mathcal{P} \leftrightarrow \mathcal{V}$**:** The prover and the verifier interact as follows.

$\mathcal{P}(\mathsf{vpk}, \psi, Y, w)$		$\mathcal{V}(\mathsf{vsk}, \mathsf{vpk}, \psi, Y)$
$r_1 \xleftarrow{\$} [\![2^{2\lambda+\log\lambda}\lvert\mathcal{D}\rvert]\!]$		
$a = \psi(r_1)$	$\xrightarrow{a}$	
	$\xleftarrow{b}$	$b \xleftarrow{\$} \{0,1\}^\lambda$
$C = \prod_{i=1}^{\lambda} \mathsf{ct}_i^{b_i}$		
$r_2 \xleftarrow{\$} \mathcal{R}$		
$S = C^w \cdot \mathsf{Enc}_{\mathsf{pk}}(r_1; r_2)$		
$t_1 \xleftarrow{\$} [\![2^{2^\lambda}\lvert\mathcal{D}\rvert]\!]$		
$t_2 \xleftarrow{\$} \mathcal{M}, t_3 \xleftarrow{\$} \mathcal{R}$		
$T = C^{t_1} \cdot \mathsf{Enc}_{\mathsf{pk}}(t_2; t_3)$	$\xrightarrow{S, T}$	
	$\xleftarrow{d}$	$d \xleftarrow{\$} [\![2^\lambda]\!]$
$u_1 = t_1 + dw \in \mathbb{Z}$		
$u_2 = t_2 + dr_1 \in \mathcal{M}$		
$u_3 = t_3 \cdot r_2^d \in \mathcal{R}$	$\xrightarrow{u_1, u_2, u_3}$	$s = \mathsf{Dec}_{\mathsf{sk}}(S)$
		$c = \sum c_i b_i; C = \prod \mathsf{ct}_i^{b_i}$
		$aY^c \stackrel{?}{=} \psi(s)$
		$TS^d \stackrel{?}{=} C^{u_1} \cdot \mathsf{Enc}_{\mathsf{pk}}(u_2; u_3)$

Fig. 2. $\mathsf{DV}_{\mathsf{Prot}}$: The designated-verifier Σ-protocol for $\mathcal{R}_{\mathsf{Hom}}$ demonstrating knowledge of a preimage of $\psi(\cdot)$. The additively homomorphic encryption scheme is instantiated with Paillier with $|\mathcal{M}| = |N_{\mathsf{pk}}|$. This scheme is knowledge sound for subverted RSA groups provided that the outputs of $\mathsf{KeyGen}(1^\lambda)$ are well-formed.

4.2 Security

We now argue the security of our $\mathsf{DV}_{\mathsf{Prot}}$. For correctness, we only need to make sure that the message space $\mathcal{M}$ of AHE is large enough to fit the largest possible $s = r_1 + cw$. That is we require an additively homomorphic IND-CPA secure Encryption Scheme with message space $|\mathcal{M}| > 2^{2\lambda+\log\lambda}|\mathcal{D}|$.

Knowledge soundness. To demonstrate knowledge soundness we first describe an extractor that can rewind a malicious prover and aims to output the prover's witness. This extractor obtains $M(\lambda) = \mathsf{poly}(\lambda)$ different verifying transcripts from the prover and succeeds if the gcd of the challenges of these transcripts is equal to 1. We then describe a reduction $\mathcal{B}$ that succeeds at IND-CPA whenever the extractor fails at obtaining a valid witness. The reduction queries an encryption oracle to determine the vpk and therefore does not know the contents of the encryptions. It runs the prover and decides whether a transcript verifies or not based on whether the transcript verifies with *both* possible contents. We argue that if it verifies with one of the possible contents but not the other, then provided the domain space of $\psi()$ is bigger than 2^λ, then $\mathcal{B}$ can guess the contents of the ciphertexts with overwhelming probability. We further argue that the gcd of the challenges the prover does not see must equal 1 with overwhelming probability. Thus if the extractor fails then $\mathcal{B}$ can guess which challenges the ciphertexts contain based on whether the gcd is 1 or not.

The protocol and theorem currently do not give the prover oracle access to the verifier. In Section 4.4 we will describe an extension of our DV protocol that can give the prover this access.

Theorem 1 (Knowledge Soundness). *The* $\mathsf{DV}_{\mathsf{Prot}}$ *protocol is knowledge-sound in the designated verifier model, provided that the* AHE *is IND-CPA secure.*[14]

Proof. Suppose that $(\mathsf{vpk}, \mathsf{vsk}, \tau) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$, where $\tau = \{c_1, \ldots, c_\lambda\}$ contains the challenges encrypted in vpk but not the secret key sk of AHE. Assume that $\mathcal{P}^*(\mathsf{vpk}, \psi, Y; \mathsf{coin})$ is a malicious prover that is run on random coins coin. We first describe an extractor Ext, that has rewindable black-box access to the prover $\mathcal{P}^*$, such that whenever $\mathcal{P}^*$ outputs verifying $(Y; (a, S, T, u_1, u_2, u_3))$ $\mathsf{Ext}^{\mathcal{P}^*}(\tau, \mathsf{vpk}, \psi, Y)$ outputs a witness w such that $Y = \psi(w)$. The Ext algorithm depends on two subalgorithms, Ext_0 and Ext_1 where Ext_0 is the extractor from Lemma 1, and Ext_1 we present below.

Ext_1, on input τ, vpk, ψ and Y, runs $\mathcal{P}^*(\mathsf{vpk}, \psi, Y; \mathsf{coin})$ (on challenges b, d of its choice) until it obtains a full $(M, 2)$-tree of accepting transcripts, for the same first message a. That is:

$$\mathcal{T} = \left\{ \left(a, b^{(j)}, S^{(j)}, T^{(j)}, d^{(j,k)}, u_1^{(j,k)}, u_2^{(j,k)}, u_3^{(j,k)} \right) \right\}_{j \in [M], k \in [2]}$$

[14] We further assume that if $\mathbb{Z}_N$ is the message space, then the largest factor of N is larger than $2^{\lambda+1}$, which is the case for example in Paillier.

and outputs $\mathcal{T}$. For Ext_1 we use the generic $(M, 2)$-special soundness extractor (see [2]), that efficiently finds such a tree. As we argue later we set $M = \mathsf{poly}(\lambda)$.

More specifically, Ext_1 proceeds as follows. It probes $\mathcal{P}^*$ on randomly sampled coin, b, d until it obtains $\left(a, b^{(1)}, S^{(1)}, T^{(1)}, d^{(1,1)}, u_1^{(1,1)}, u_2^{(1,1)}, u_3^{(1,1)}\right)$ such that $T^{(1)}(S^{(1)})^{d^{(1,1)}} = (C^{(1)})^{u_1^{(1,1)}}\mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,1)}; u_3^{(1,1)})$, where $C^{(1)} = \prod_{i=1}^{\lambda} \mathsf{ct}_i^{b_i^{(1)}}$. Since it does not have vsk it cannot directly decrypt $S^{(1)}$ to $s^{(1)}$ and check whether $aY^{c^{(1)}} = \psi(s^{(1)})$. For this it continues probing $\mathcal{P}^*$ on the same coin and $b^{(1)}$ until it obtains a second $\left(a, b^{(1)}, S^{(1)}, T^{(1)}, d^{(1,2)}, u_1^{(1,2)}, u_2^{(1,2)}, u_3^{(1,2)}\right)$ such that $T^{(1)}(S^{(1)})^{d^{(1,2)}} = (C^{(1)})^{u_1^{(1,2)}}\mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,2)}; u_3^{(1,2)})$. So we have:

$$T^{(1)}(S^{(1)})^{d^{(1,1)}} = (C^{(1)})^{u_1^{(1,1)}}\mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,1)}; u_3^{(1,1)})$$
$$T^{(1)}(S^{(1)})^{d^{(1,2)}} = (C^{(1)})^{u_1^{(1,2)}}\mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,2)}; u_3^{(1,2)})$$

or

$$(S^{(1)})^{d^{(1,1)}-d^{(1,2)}} = \mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,1)} + c^{(1)}u_1^{(1,1)} - u_2^{(1,2)} - c^{(1)}u_1^{(1,2)})$$

From assumption $\gcd(d^{(1,1)} - d^{(1,2)}, N) = 1$ (given that the largest prime factor of N is larger that $|d^{(1,1)} - d^{(1,2)}|$) so the inverse $\left(d^{(1,1)} - d^{(1,2)}\right)^{-1}$ exists in $\mathcal{M}$ and Ext_1 extracts $s^{(1)} = s_2^{(1)} + c^{(1)}s_1^{(1)}$ such that $S^{(1)}$ encrypts $s^{(1)}$ (under some randomness unknown to the extractor) where

$$s_1^{(1)} = \left(u_1^{(1,1)} - u_1^{(1,2)}\right)\left(d^{(1,1)} - d^{(1,2)}\right)^{-1} \mod N$$
$$s_2^{(1)} = \left(u_2^{(1,1)} - u_2^{(1,2)}\right)\left(d^{(1,1)} - d^{(1,2)}\right)^{-1} \mod N$$

From here Ext_1 can verify $aY^{c^{(1)}} = \psi(s^{(1)})$ to confirm if the two transcripts are accepting or not. It continues in a similar manner until it obtains a full $(M, 2)$-tree of accepting transcripts $\mathcal{T}$. Whenever $\mathcal{P}^*$ convinces $\mathcal{V}$ with non-negligible probability Ext_1 computes the decryption of $S^{(1)}$ in polynomial time thus the probability that Ext_1 accepts a false transcript is negligible.[15]

Now, the extractor Ext behaves as follows. It runs $\mathcal{T} \leftarrow \mathsf{Ext}_1^{\mathcal{P}^*}(\tau, \mathsf{vpk}, \psi, Y)$ and computes $c^{(j)} = \boldsymbol{b}^{(j)}\boldsymbol{c}^T = \sum_{i=1}^{\lambda} c_i b_i^{(j)}$. If $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(\lambda)} - c^{(1)}) \neq 1$ it aborts. Else it computes $s^{(j)}$ as shown above (where it holds that $s^{(j)} = \mathsf{Dec}_{\mathsf{sk}}(S^{(j)})$) for each $j \in [M]$ and runs $w \leftarrow \mathsf{Ext}_0(\psi, Y; (a, c^{(1)}, s^{(1)}), \ldots, (a, c^{(M)}, s^{(M)}))$ and returns w.

We first see that Ext runs in polynomial time provided that the adversary $\mathcal{P}^*$ has non-negligible probability of success. So either $\epsilon(\lambda)$ is polynomial in λ or $\mathcal{P}^*$ only convinces $\mathcal{V}$ with negligible probability. Let $\epsilon(\lambda) > 1/\mathsf{poly}(\lambda)$ denote the probability that $\mathcal{P}^*$ convinces an honest verifier on input (ψ, Y). By Lemma 1

[15] For ease of exposition we keep the description simple. We omit the technical details of special soundness extractors related to aborting senarios, that ensure termination in polynomial time(see lemma 5, [2]).

we have that Ext_0 runs in polynomial time. For the runtime of Ext_1 we rely on [2, Lemma 5] which shows that Ext_1 runs in expected time $O(\frac{\lambda}{\epsilon-(M-1)/2^\lambda})$, which is polynomial (since we assumed that ϵ is non-negligible).

We must now show that Ext only aborts with negligible probability. This occurs if and only if $\gcd(c^{(2)}-c^{(1)},\ldots,c^{(M)}-c^{(1)}) \neq 1$ with non-negligible probability. In order to show this, we design an adversary $\mathcal{B}$ against IND-CPA that, using Ext, wins the IND-CPA game:

$\underline{\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}}(\mathsf{pk})}$
$c_1, z_1, \ldots, c_\lambda, z_\lambda \xleftarrow{\$} [\![2^\lambda]\!]$
$\mathsf{ct}_i \xleftarrow{\$} \mathcal{O}_{\mathsf{Enc}}(c_i, z_i) \quad$ for $i \in [\lambda]$;
$\mathsf{vpk} \leftarrow (\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\lambda)$
$\mathsf{coin} \xleftarrow{\$} [1, 2^\lambda]$; $j \leftarrow 1$
while $j < M$: $\quad (\mathsf{trans}_{j,1}, \mathsf{trans}_{j,2}) \leftarrow \mathcal{P}^*(\mathsf{vpk}, \psi, Y; \mathsf{coin})$
  if $aY^{c^{(j)}} = \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ and $aY^{z^{(j)}} \neq \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$ return 0
  if $aY^{c^{(j)}} \neq \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ and $aY^{z^{(j)}} = \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$ return 1
  if $aY^{c^{(j)}} = \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ and $aY^{z^{(j)}} = \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$ $j \leftarrow j+1$
if $\gcd(c^{(2)}-c^{(1)},\ldots,c^{(M)}-c^{(1)}) \neq 1$ return 0
if $\gcd(z^{(2)}-z^{(1)},\ldots,z^{(M)}-z^{(1)}) \neq 1$ return 1

where we denote $c^{(j)} = \boldsymbol{b}^{(j)}\boldsymbol{c}^T$ and $z^{(j)} = \boldsymbol{b}^{(j)}\boldsymbol{z}^T$.

Case 1. First we show that if $aY^{c^{(j)}} = \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ and $aY^{z^{(j)}} \neq \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$, then with overwhelming probability the encryptions contain $c_1, \ldots, c_\lambda$ and $\mathcal{B}$ succeeds.

The fact that $aY^{c^{(j)}} = \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ can be rewritten as:

$$\left(a\psi(-s_2^{(j)})\right) = \left(\psi(s_1^{(j)})Y^{-1}\right)^{c^{(j)}}$$

Assume that $\mathsf{ct}_i \neq \mathsf{Enc}_{\mathsf{pk}}(c_i)$ then $\mathcal{P}^*$ gets no information about $c_1, \ldots, c_\lambda$, so they are perfectly hidden. This means that from the point of view of $\mathcal{P}^*$ these are uniformly random over $[\![2^\lambda]\!]$, which makes the above happen with probability $2^{-\lambda}$ (considering also that $|\mathbb{H}| > 2^\lambda$), unless $a\psi(-s_2^{(j)}) = \psi(s_1^{(j)})Y^{-1} = 1$. Now, since $aY^{z^{(j)}} \neq \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$ then $a \neq \psi(s_2^{(j)})$ or $Y \neq \psi(s_1^{(j)})$.

We conclude then that, except with negligible probability $2^{-\lambda}$, $\{\mathsf{ct}_i\}_i$ contain encryptions of c_i.

Case 2. Second, we use the same argument as in the previous case to claim that if $aY^{c^{(j)}} \neq \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ and $aY^{z^{(j)}} = \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$, then with overwhelming probability the encryptions contain $z_1, \ldots, z_\lambda$ and $\mathcal{B}$ succeeds.

Case 3. Third we argue that if the extractor Ext fails then $\mathcal{B}$ succeeds. Indeed we have from the first two cases that transcripts only verify if both $aY^{c^{(j)}} = \psi(s_2^{(j)} + c^{(j)} s_1^{(j)})$ and $aY^{z^{(j)}} = \psi(s_2^{(j)} + z^{(j)} s_1^{(j)})$. If the encryptions contain $c_1, \ldots, c_\lambda$ then Ext only fails if $\gcd(c^{(2)}-c^{(1)},\ldots,c^{(M)}-c^{(1)}) \neq 1$. In this case $\mathcal{B}$ correctly guesses.

If the encryptions instead contain $z_1, \ldots, z_\lambda$ then Ext only fails if $\gcd(z^{(2)} - z^{(1)}, \ldots, z^{(M)} - z^{(1)}) \neq 1$. In this case $\mathcal{B}$ guesses correctly unless $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) \neq 1$. The $(c_1, \ldots, c_\lambda)$ are uniformly distributed values that are perfectly hidden from the prover and the extractor. Indeed, the encryptions contain no information and, by the first two cases, the behaviour of the extractor is entirely determined by the verification with respect to $z_1, \ldots, z_\lambda$. So the probability that $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$ is overwhelming (see Lemma 2). We thus argue that if Ext fails then $\mathcal{B}$ succeeds with overwhelming probability.

Indeed Lemma 2 shows that $\Pr[\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1] = 1 - \mathsf{negl}(\lambda)$.

To see why Lemma 2 applies in our case, $\boldsymbol{B}$ corresponds to the matrix containing all the challenges b which the adversary can successfully answer, when the first message is a. Since the extractor was able to obtain M such challenges in (expected) polynomial time, this means that $\boldsymbol{B}$ is at most polynomially smaller than 2^λ: there exists $\epsilon' > 1/\mathsf{poly}(\lambda)$ such that $|\boldsymbol{B}| = \epsilon' 2^\lambda$. We can show this by contradiction, assume that $\epsilon' = 1/\omega(\mathsf{poly}(\lambda))$, then the expected time for Ext to find a successful answer would be non-polynomial $\omega(\mathsf{poly}(\lambda))$. Finally, $\boldsymbol{B'}$ corresponds to the matrix consisting of the challenges in $\mathcal{T}$.

□

Zero-knowledge. To demonstrate zero-knowledge we will provide a simulator and argue that the simulators outputs are indistinguishable from the honest provers. We make use of a standard blinding lemma.

The main HVZK result is as follows (due to space limitations the proof is deferred to the full version of the paper):

Theorem 2 (Honest Verifier Zero Knowledge). $\mathsf{DV}_{\mathsf{Prot}}$ *is statistical honest-verifier zero-knowledge for the relation* $\mathcal{R}_{\mathsf{Hom}}$.

Since our DV protocol is essentially Schnorr-like, the simulator is almost as usual: it samples response values uniformly (since they are properly blinded in the honest protocol), and generates (encrypted) challenges using verifier's equations. The only difference is that one challenge is an encryption value. Also the proof assumes honest CRS setup.

4.3 Malicious VPK Generation

The $\mathsf{DV}_{\mathsf{Prot}}$ protocol in the previous section assumes that the verifier's public key is trusted. In particular, zero-knowledge only holds on the condition that ct_i contains plaintexts $c_i \in [\![2^\lambda]\!]$ for all i. In this section we explain how to generate a vpk in a way that prevents dishonest verifiers from breaking zero-knowledge of our DV construction.

For lack of space we defer the formal description of the malicious-verifier alternative key generation procedure is presented to the full version. We edit the setup algorithm such that the verifier must provide a range proof on the ciphertexts it generates for vpk.

In the full version we present a protocol proving range of the VPK ciphertext efficiently, together with a security proof. The protocol follows the transformation by Cramer et al. [29, 30] allowing to increase performance when proving multiple instances simultaneously; however our instantiation has a number of differences from the original transformation. The range proof comes with a slack: a verifying π on the prover's side guarantees that when $c_i \in [\![2^\lambda]\!]$, the resulting messages in the ciphertexts ct_i of vpk are in the extended interval $[\![2^{3\lambda+\log\lambda-1}]\!]$ (the slack is $2^{2\lambda+\log\lambda-1}$). Therefore the encrypted sum-challenge $\mathcal{P}$ replies to is in $[\![2^{3\lambda+2\log\lambda-1}]\!]$. To preserve zero-knowledge we must increase the blinding parameter r_1 on the prover's side to this value, multiplied by $|\mathcal{D}|$. This in turn requires us to increase AHE $|\mathcal{M}|$ to $|\mathcal{D}|2^{3\lambda+2\log\lambda}$, to be enough to fit the new $s = r_1 + cw \overset{s}{\approx} r_1$.

In addition to this, we also must prove that verifier's public key N_{pk} gives rise to an *injective* Paillier instantiation, since otherwise the statement of the range proof is not useful . For this we use [42, Protocol $\mathcal{P}_{\mathsf{Paillier-N}}$, Sec. 3.2] — it is public-coin, so can be executed non-interactively (using FS); it proves $\gcd(N_{\mathsf{pk}}, \phi(N_{\mathsf{pk}})) = 1$, which is enough to achieve injectivity of Paillier; and it is quite efficient, only taking a few percent of all KeyGen computations.

Theorem 3. *Protocol* $\mathsf{DV}_{\mathsf{Prot}}$*, augmented with* KeyGen *and* $\mathcal{P}$ *from* **??***, is statistical honest-verifier zero-knowledge under malicious VPK for the relation* $\mathcal{R}_{\mathsf{Hom}}$*.*

4.4 Reusable VPK

In this section we present $\mathsf{DVReusable}_{\mathsf{Prot}}$, a modification of $\mathsf{DV}_{\mathsf{Prot}}$, in which vpk is reusable $Q = \mathsf{poly}(\lambda)$ number of times. This means the prover can query the verifier to learn whether their response verifies up to Q times. We achieve this by adding Q encrypted challenges to the vpk. The result is that both the communication and the computation complexity related to vpk generation and verification can be amortized down to $O(1)$ per query.

For the basic $\mathsf{DV}_{\mathsf{Prot}}$ it is possible to show an attack in which an adversarial prover, interacting with the verifier many times, uses the information of whether a (malicious) proof of their choice verifies or not in order to learn plaintext challenges c_i in the vpk. This in turn defeats the purpose of hiding the challenges, and prevents extraction, breaking soundness.

To overcome this we introduce additional challenge blinders. First, we sample $\hat{c}_\kappa$ of size at least $\lambda 2^{2\lambda}$ per query, encrypt them to $\hat{\mathsf{ct}}_\kappa$, and add them all to the VPK. Then we use $\hat{\mathsf{ct}}_\kappa$ in the final challenge $C = \hat{\mathsf{ct}}_\kappa \prod_i \mathsf{ct}_i^{b_i}$ (for a challenge bit-vector b) so that $\hat{c}_\kappa$ statistically hides $\sum c_i b_i$ since $\hat{c}_\kappa$ is at least 2^λ larger. This means that the adversary statistically learns no information about $\{c_i\}$, but only about $\hat{c}_\kappa$. Each challenge $\hat{c}_\kappa$ must be used exactly once, which is enforced by $\mathcal{V}$.

The final challenge size now grows to $\lambda 2^{2\lambda}$, which means r_1 must be sampled from $[\![\lambda 2^{3\lambda}|\mathcal{D}|]\!]$, and $|\mathcal{M}|$ of verifier's AHE must be bigger than this value.

Theorem 4. $\mathsf{DVReusable}_{\mathsf{Prot}}$ *is a complete, honest-verifier zero-knowledge protocol in the designated-verifier setting, that has knowledge-soundness with Q-times reusable VPK for any polynomial $Q(\lambda)$.*

Due to space limitations the proof is deffered to the full version.

4.5 Malicious and Reusable VPK

Techniques from the two previous sections can be combined. The *reusable* VPK from Section 4.4 can also be generated *maliciously* with the same technique from Section 4.3.

The batched range proof now must also cover new "bigger" challenges introduced for reusability. From the perspective of efficiency of amortized $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$ it is optimal to batch exactly $n = \lambda$ instances together. Thus we will prove challenge ranges of c_i in batches of size λ, where first batch uses range bound $R_1 = 2^\lambda$ (corresponding to small ciphertexts), and the following Q/λ batches use $R_2 = \lambda 2^{2\lambda}$. When $\lambda \nmid Q$, $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$ instance can be padded with dummy values.

Given $2^{\lambda+\log\lambda-1}$ slack of the range proof, we must sample $r_1 \in [\![2^{5\lambda+2\log\lambda}|\mathcal{D}|]\!]$; and $|\mathcal{M}|$ must be chosen to be bigger than this r_1.

4.6 Efficiency Optimization in the Generic Group Model

Here we describe a variant of the $\mathsf{DV}_{\mathsf{Prot}}$ protocol that consists of 3 rounds (instead of 5) and thus saves 4 elements from the proof size. The protocol transcript simply consists of (a, b, S) omitting T, d, u_1, u_2, u_3 together with the last two rounds.

In $\mathsf{DV}_{\mathsf{Prot}}$ the last three messages T, d and (u_1, u_2, u_3) are used to prove that S is a well-formed ciphertext. Namely, the extractor of Theorem 1, at each accepting transcript should be able to obtain an $s^{(j)}$ such that $S^{(i)} = \mathsf{Enc}_{\mathsf{pk}}(s^{(j)})$. We observe that if we instantiate the encryption scheme with the Paillier-with-randomness-in-the-exponent cryptosystem, $S^{(j)} = (N+1)^{s^{(j)}} h^r$ then our extractor can obtain $s^{(j)}$ for free in the generic group model [59, 52] (GGM).

GGM for unknown order groups has been established [37, 13] in a similar manner to the original model. For this optimization we make use of this model. For knowledge-soundness we assume that the group generated for the Paillier encryption is honest (it's part of VPK), thus the model applies normally.

The following proof is almost identical to that of Theorem 1 except that the extractor now uses whitebox access to the prover instead of the rewinding argument to find a representation for S.

Theorem 5 (Knowledge Soundness). *The optimised* $\mathsf{DV}_{\mathsf{Prot}}$ *described above is knowledge-sound in the generic group model provided that the* AHE *is IND-CPA secure.*

Due to space limitations the proof is deffered to the full version.

5 Designated Verifier Range Proof

In this section we construct $\mathsf{DVRange}_{\mathsf{Prot}}$ — a zero-knowledge argument of knowledge for the range of the pre-image of general homomorphisms. Formally, we are interested in the relation:

$$\mathcal{R}_{\mathsf{HomRange}} = \left\{(\psi, Y, R); x : Y = \psi(x) \ \wedge \ x \in [0, R]\right\}$$

where $\psi : \mathcal{D} \to \mathbb{G}$ and $\mathbb{G}$ is a group parameterised by a (possibly subverted) RSA modulus N. We use our designated-verifier protocol of Section 4, that is able to extract the witness using the extraction strategy of Lemma 1. On top of that, we use the range proof from [27] for RSA groups.

The protocol from [27] works over an integer commitment [40, 34] $c = g^x h^r$ in an RSA group for which the order is unknown to the prover. Since we cannot assume that $\mathbb{G}$ is such a group (recall that the prover might know the order of $\mathbb{G}$) we let the verifier generate an RSA modulus N_{cm} together with the bases of the commitment g, h, which are included in the verification key. The prover first commits to the pre-image x in $\mathbb{Z}_{N_{\mathsf{cm}}}$, $c = g^x h^r$ and sends c to the verifier. Then it performs the two protocols, the opening of ψ (section 4.1) and the range proof of [27] (compiled with the same Designated-Verifier technique), in parallel.

For completeness, we recall the aforementioned integer commitment scheme used. It works over any group of unknown (to the committer) order such as an RSA group or a class group. In our case, we focus on the RSA instantiation, thus the underlying group is $\mathbb{Z}_{N_{\mathsf{cm}}}$, where N_{cm} is an RSA modulus. The commitment key consists of two random elements $g, h \in \mathbb{Z}_{N_{\mathsf{cm}}}$ such that $g \in \langle h \rangle$. In the key generation phase we sample uniformly $g \leftarrow\!\!\$\ \mathbb{Z}_{N_{\mathsf{cm}}}$ and $f \leftarrow\!\!\$\ \phi(N_{\mathsf{cm}})$[16] and output $(g, h) = (h^f, h)$. A commitment to x is merely $c = g^x h^r$ for a random $r \leftarrow\!\!\$\ [\![\frac{N_{\mathsf{cm}}}{2}]\!]$. The opening values are (x, r) and the verification is $c = \pm g^x h^r$.[17] The scheme is binding under the factoring assumption for N_{cm} and statistically hiding.

We present $\mathsf{DVRange}_{\mathsf{Prot}}$ in Fig. 3,(for lack of space we describe its key generation in the full version).For ease of presentation parts related to the range proof and the opening of ψ are visually separated, denoted as (1) and (2) respectively. We directly present our protocol with reusable and maliciously generated vpk, similarly to how these were presented for $\mathsf{DV}_{\mathsf{Prot}}$ in Sections 4.3 and 4.4.

For the key generation, except for a secret/public key of the additively homomorphic encryption scheme (Paillier cryptosystem), we further need an RSA modulus N_{cm} and the group elements g, h to instantiate the integer commitment scheme. For zero-knowledge to hold even under maliciously generated vpk it is important that $g = h^f$ holds. Therefore we additionally include a zero-knowledge proof ensuring it.

[16] In case $\phi(N_{\mathsf{cm}})$ is unknown, sampling $f \leftarrow\!\!\$\ [\![\frac{N_{\mathsf{cm}}}{2}]\!]$ is statistically close.

[17] The $\pm$ relaxation is artificially added in order to achieve a sound zero-knowledge proof of opening of c, which however does not affect the binding of the commitment scheme.

$\mathcal{P}(\mathsf{vpk}, \psi, Y, R, \kappa, x) \leftrightarrow \mathcal{V}(\mathsf{vsk}, \mathsf{vpk}, \psi, Y, R, \kappa)$**:**

$\mathcal{P}_1$:
1. Sample $t \leftarrow\!\!\$\ [\![2^{\lambda} \frac{N_{\mathsf{cm}}}{2}]\!]$ and compute $\mathsf{cm} = g^x h^t \mod N_{\mathsf{cm}}$.
2. Sample $r \leftarrow\!\!\$\ [\![2^{5\lambda+2\log\lambda} R]\!], \sigma \leftarrow\!\!\$\ [\![2^{6\lambda+2\log\lambda} \frac{N_{\mathsf{cm}}}{2}]\!]$ and compute $\beta = g^r h^{\sigma}$.
3. Find $x_1, x_2, x_3 \in \mathbb{Z}$ such that $4x(R-x)+1 = \sum_{i=1}^{3} x_i^2$ (using e.g. [57]).
4. Sample $t_i \leftarrow\!\!\$\ [\![2^{\lambda} \frac{N_{\mathsf{cm}}}{2}]\!]$ and compute $\mathsf{cm}_i = g^{x_i} h^{t_i}$, for $i \in [1,3]$.
5. Sample $r_i \leftarrow\!\!\$\ [\![2^{5\lambda+2\log\lambda} R]\!], \sigma_i \leftarrow\!\!\$\ [\![2^{6\lambda+2\log\lambda} \frac{N_{\mathsf{cm}}}{2}]\!]$ and compute $\beta_i = g^{r_i} h^{\sigma_i}$, for $i \in [1,3]$.
6. Sample $\tau \leftarrow\!\!\$\ [\![2^{6\lambda+2\log\lambda+4} \frac{N_{\mathsf{cm}}}{2} R]\!]$ and compute $\beta_4 = h^{\tau} \mathsf{cm}^{4r} \prod_{i=1}^{3} \mathsf{cm}_i^{-r_i}$.
7. Compute $\alpha = \psi(r)$.

$\underline{\mathcal{P} \to \mathcal{V}}$: send $a = \left(\mathsf{cm}, \{\mathsf{cm}_i\}_{i\in[1,3]}, \alpha, \beta, \{\beta_i\}_{i\in[1,4]}\right)$

$\mathcal{V}_1$: Sample $b \xleftarrow{\$} \{0,1\}^{\lambda}$ (denote $(b_1, \ldots, b_\lambda) := b$).
$\underline{\mathcal{V} \to \mathcal{P}}$: send b

$\mathcal{P}_2$:
1. Compute challenge ciphertext $C = \mathsf{ct}_{\lambda+\kappa} \cdot \prod_{i=1}^{\lambda} \mathsf{ct}_i^{b_i}$
2. Compute:
   - $U = \mathsf{Enc}_{\mathsf{pk}}(r) \cdot C^{R-x}$, $V = \mathsf{Enc}_{\mathsf{pk}}(\sigma) \cdot C^{-t}$.
   - $U_i = \mathsf{Enc}_{\mathsf{pk}}(r_i) \cdot C^{x_i}$, $V_i = \mathsf{Enc}_{\mathsf{pk}}(\sigma_i) \cdot C^{t_i}$, for $i \in [1,3]$.
   - $U_4 = \mathsf{Enc}_{\mathsf{pk}}(\tau) \cdot C^{\sum_{i=1}^{3} x_i t_i - 4(R-x)t}$.

$\underline{\mathcal{P} \to \mathcal{V}}$: send $S = (U, V, \{U_i\}_{i\in[1,3]}, \{V_i\}_{i\in[1,3]}, U_4)$

$\mathcal{V}_2$:
1. Compute plaintext challenge $c = c_{\lambda+\kappa} + \sum_{i=1}^{\lambda} c_i b_i$
2. Decrypt $U, V, \{U_i\}_{i\in[1,3]}, \{V_i\}_{i\in[1,3]}, U_4$: $u = \mathsf{Dec}_{\mathsf{sk}}(U)$, $v = \mathsf{Dec}_{\mathsf{sk}}(V)$, $u_i = \mathsf{Dec}_{\mathsf{sk}}(U_i)$, $v_i = \mathsf{Dec}_{\mathsf{sk}}(V_i)$ for $i \in [1,3]$ and $u_4 = \mathsf{Dec}_{\mathsf{sk}}(U_4)$
3. Perform the following checks:
   - $\beta(\mathsf{cm}^{-1} g^R)^c \stackrel{?}{=} g^u h^v$
   - $\beta_i \mathsf{cm}_i^c \stackrel{?}{=} g^{u_i} h^{v_i}$, for $i \in [1,3]$
   - $\beta_4 \prod_{i\in[1,3]} \mathsf{cm}_i^{u_i} \stackrel{?}{=} h^{u_4} g^c \mathsf{cm}^{4u}$
   - $u_i \stackrel{?}{\in} [\![2^{5\lambda+2\log\lambda} R]\!]$, for $i \in [1,3]$
   - $\alpha \left(Y^{-1}\psi(R)\right)^c \stackrel{?}{=} \psi(u)$

$\mathcal{P} \leftrightarrow \mathcal{V}$: (Non-GGM part:) For each ciphertext of the third message S perform a variant of the three-round $\mathsf{Sigma}_{\mathsf{Prot}}$ for the relation $\mathcal{R} = \left\{(S_i, C); (w_1, w_2, w_3) : S_i = \mathsf{Enc}_{\mathsf{pk}}(w_1; w_2) \cdot C^{w_3}\right\}$ with $|\mathcal{C}| = 2^{\lambda}$. This can be done in two extra rounds starting with $\mathcal{P}_2$, as in Fig. 2.

Fig. 3. $\mathsf{DVRange}_{\mathsf{Prot}}$: The designated-verifier range proof of a preimage of ψ. The key generation phase is presented in **??**.

Security. The above protocol consists of two sub-protocols: our protocol of Section 4.1 and the range proof by Couteau et. al. [27] over RSA groups. Thus the security of the protocol can be proven in a straightforward way from the security of these subprotocols. For correctness, again we need to consider the size of the message space $\mathcal{M}$ of the encryption scheme AHE. Indeed $|\mathcal{M}|$ needs to be at least as large as the maximum value encrypted, which equals $\tau + \sum_{i=1}^{3} x_i t_i - 4(R-x)t$, the content of U_4. Knowledge-Soundness follows directly from the knowledge-soundness of the two sub-protocols.

Theorem 6. *Let* AHE *be an IND-CPA secure Encryption Scheme with message space* $|\mathcal{M}| > 2^{6\lambda+2\log\lambda+4} N_{\mathsf{cm}} R$. *Then* $\mathsf{DVRange}_{\mathsf{Prot}}$ *is a designated verifier argument of knowledge for the relation* $\mathcal{R}_{\mathsf{HomRange}}$ *that is: correct, Q-reusable knowledge-sound under the Factoring assumption for* N_{cm} *and IND-CPA security of* AHE *and statistically honest-verifier zero-knowledge under malicious VPK.*

Due to space limitations the proof is deffered to the full version.

$\mathsf{DVRange}_{\mathsf{Prot}}$ can be optimised in the generic group model similarly to how it is done in Section 4.6. In this case we can omit the final interaction between prover and verifier in Fig. 3 that proves knowledge of the plaintext inside S_i.

6 Evaluation and Performance

We implemented[18] and benchmarked our protocols, primarily focusing on evaluating and comparing $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$ (Table 1), proving knowledge of the ciphertext message, and its range correspondingly. As a baseline we also implemented several flavours of the basic Σ-protocol (Table 2). For simplicity here we only present non-interactive (Fiat-Shamir transformed) variants.

The evaluation indicates that our protocols is a strictly better choice for certain types of applications (e.g. ID-MPC such as RSA ceremonies), as they exhibit better verification time and communication size. For *generic* applications, our protocols are comparable to other solutions, providing different performance trade-offs.

Setup and Instantiation Details. We ran our benchmarks on the Intel i5-8500 @ 3.00GHz processor. For illustrative purposes the protocol code runs in the single-core mode only, and no specifically tailored low-level optimisations are implemented. All the evaluations are presented for $\lambda = 128$, and $\log N = 2048$; for the range proof we take $R = 2^{256}$; the maximum query number of VPK reuses is set to $Q = 128$. For Fiat-Shamir transformation we instantiate the random oracle with the $\mathsf{Blake2b}$ [4] hash function.

For $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$ we use Paillier-ElGamal encryption as the target homomorphism (which is additively homomorphic in both message and randomness), and standard Paillier as the AHE scheme on the verifier's side. For each of

[18] The implementation is available publicly on Github: https://github.com/volhovm/rsa-zkps-impl

	VPK Gen	VPK Verify	Prove	Verify	Proof size	VPK size
$\mathsf{DV}_{\mathsf{Prot}}$ M	4754	12310	162	66	5.52 KB	741 KB
$\mathsf{DV}_{\mathsf{Prot}}$ T	836	-	130	56	5.14 KB	159 KB
$\mathsf{DV}_{\mathsf{Prot}}$ M GGM	4754	12310	84	32	2.32 KB	741 KB
$\mathsf{DV}_{\mathsf{Prot}}$ T GGM	836	-	69	28	2.19 KB	159 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ M	13827	25900	1880	1120	34.32 KB	842 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ T	9106	-	1330	782	31.78 KB	188 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ M GGM	13827	25900	689	153	11.05 KB	842 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ T GGM	9106	-	490	111	10.41 KB	188 KB

Table 1. Evaluation of our main protocols. Timings are in ms. "GGM" is GGM optimisation, and "M/T" stand for malicious or trusted setup.

	Prove	Pre-Verify	Verify	Proof size
$\mathsf{Sigma}_{\mathsf{Prot}}$ Paillier, $\lambda = 128$ reps	342	0	1161	134.00 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$ Paillier, 8 reps	21	4	73	8.38 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$ Paillier, 7 reps	19	36	64	7.33 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$ Paillier, 6 reps	16	339	55	6.28 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$ Paillier, 5 reps	14	6535	46	5.23 KB
$\mathsf{SigmaRange}_{\mathsf{Prot}}$ Paillier (with slack)	345	0	1157	108.00 KB

Table 2. Performance for the baseline algorithms. Timings are in milliseconds. $\mathsf{Sigma}_{\mathsf{Prot}}$ is evaluated with different p_{max}/number of repetition parameters. Note that $\mathsf{SigmaRange}_{\mathsf{Prot}}$ has range slack while $\mathsf{DVRange}_{\mathsf{Prot}}$ is tight.

our two protocols we evaluate four cases, depending on whether we use the GGM optimisation or not, and whether we consider malicious VPK or a trusted one (for the ID-MPC case). In the latter case we do not consider VPK verification time.

For the baseline $\mathsf{Sigma}_{\mathsf{Prot}}$ and $\mathsf{SigmaRange}_{\mathsf{Prot}}$ we use standard Paillier. We evaluate $\mathsf{Sigma}_{\mathsf{Prot}}$ with naive $\lambda = 128$ reps, and also with varying $\log p_{\mathsf{max}} \in \{16, 19, 22, 26\}$. The range proof $\mathsf{SigmaRange}_{\mathsf{Prot}}$ cannot use the p_{max} optimisation. Note, importantly, that $\mathsf{SigmaRange}_{\mathsf{Prot}}$ has multiplicative range slack $2^{\lambda+1}$, while our $\mathsf{DVRange}_{\mathsf{Prot}}$ is tight; this means comparing them directly is not even possible for all applications.

Performance Overview. Below we will mostly consider the GGM optimised variants of our protocols that assumes trusted setup, since it gives us best performance, and fits ID-MPC case well. The main advantage of our $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$ is that they are single-shot, requiring no repetitions. It affects the two protocols non-proportionally, benefiting $\mathsf{DVRange}_{\mathsf{Prot}}$ more, since the baseline $\mathsf{SigmaRange}_{\mathsf{Prot}}$ cannot avoid λ repetitions. Our verification time is strictly less than the baseline: 1.5-2× for $\mathsf{DV}_{\mathsf{Prot}}$, and 10× for $\mathsf{DVRange}_{\mathsf{Prot}}$. Communication is more efficient too, since our proofs are strictly smaller. Even with our VPK being comparably heavy, its size together with $Q = 128$ proofs gives us 1.5-2× improvement for $\mathsf{DV}_{\mathsf{Prot}}$ and 6-9× improvement for $\mathsf{DVRange}_{\mathsf{Prot}}$. Our proving time is slightly higher for $\mathsf{DVRange}_{\mathsf{Prot}}$, and about 2× higher with $\mathsf{DV}_{\mathsf{Prot}}$.

Acknowledgements The first author received funding from projects from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under project PICOCRYPT (grant agreement No. 101001283), from the Spanish Government under project PRODIGY (TED2021-132464B-I00), and from the Madrid Regional Government under project BLOQUES (S2018/TCS-4339). The last two projects are co-funded by European Union EIE, and NextGenerationEU/PRTR funds. The last author was partially funded by Input Output (iohk.io) through their funding of the Edinburgh Blockchain Technology Lab.

References

[1] T. Attema and R. Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics". In: *CRYPTO 2020, Part III*. Ed. by D. Micciancio and T. Ristenpart. Vol. 12172. LNCS. Springer, Heidelberg, Aug. 2020, pp. 513–543. DOI: 10.1007/978-3-030-56877-1_18.

[2] T. Attema, R. Cramer, and L. Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *Annual International Cryptology Conference*. Springer. 2021, pp. 549–579.

[3] B. Auerbach and B. Poettering. "Hashing Solutions Instead of Generating Problems: On the Interactive Certification of RSA Moduli". In: *PKC 2018, Part II*. Ed. by M. Abdalla and R. Dahab. Vol. 10770. LNCS. Springer, Heidelberg, Mar. 2018, pp. 403–430. DOI: 10.1007/978-3-319-76581-5_14.

[4] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein. "BLAKE2: Simpler, Smaller, Fast as MD5". In: *ACNS 13*. Ed. by M. J. Jacobson Jr., M. E. Locasto, P. Mohassel, and R. Safavi-Naini. Vol. 7954. LNCS. Springer, Heidelberg, June 2013, pp. 119–135. DOI: 10.1007/978-3-642-38980-1_8.

[5] E. Bangerter. "Efficient zero knowledge proofs of knowledge for homomorphisms." PhD thesis. Citeseer, 2005.

[6] E. Bangerter, J. Camenisch, and S. Krenn. "Efficiency Limitations for S-Protocols for Group Homomorphisms". In: *TCC 2010*. Ed. by D. Micciancio. Vol. 5978. LNCS. Springer, Heidelberg, Feb. 2010, pp. 553–571. DOI: 10.1007/978-3-642-11799-2_33.

[7] E. Bangerter, J. Camenisch, and U. Maurer. "Efficient Proofs of Knowledge of Discrete Logarithms and Representations in Groups with Hidden Order". In: *PKC 2005*. Ed. by S. Vaudenay. Vol. 3386. LNCS. Springer, Heidelberg, Jan. 2005, pp. 154–171. DOI: 10.1007/978-3-540-30580-4_11.

[8] E. Bangerter, S. Krenn, A.-R. Sadeghi, T. Schneider, and J.-K. Tsay. "On the design and implementation of efficient zero-knowledge proofs of knowledge". In: *Software Performance Enhancements for Encryption and Decryption and Cryptographic Compilers–SPEED-CC* 9 (2009), pp. 12–13.

[9] N. Bari and B. Pfitzmann. "Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees". In: *EUROCRYPT'97*. Ed. by W. Fumy. Vol. 1233. LNCS. Springer, Heidelberg, May 1997, pp. 480–494. DOI: 10.1007/3-540-69053-0_33.

[10] F. Benhamouda, H. Ferradi, R. Géraud, and D. Naccache. "Non-interactive Provably Secure Attestations for Arbitrary RSA Prime Generation Algorithms". In: *ESORICS 2017, Part I*. Ed. by S. N. Foley, D. Gollmann, and E. Snekkenes.

Vol. 10492. LNCS. Springer, Heidelberg, Sept. 2017, pp. 206–223. DOI: 10.1007/978-3-319-66402-6_13.

[11] M. Blum, P. Feldman, and S. Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *20th ACM STOC*. ACM Press, May 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[12] F. Böhl, D. Hofheinz, T. Jager, J. Koch, J. H. Seo, and C. Striecks. "Practical Signatures from Standard Assumptions". In: *EUROCRYPT 2013*. Ed. by T. Johansson and P. Q. Nguyen. Vol. 7881. LNCS. Springer, Heidelberg, May 2013, pp. 461–485. DOI: 10.1007/978-3-642-38348-9_28.

[13] D. Boneh, B. Bünz, and B. Fisch. "Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains". In: *CRYPTO 2019, Part I*. Ed. by A. Boldyreva and D. Micciancio. Vol. 11692. LNCS. Springer, Heidelberg, Aug. 2019, pp. 561–586. DOI: 10.1007/978-3-030-26948-7_20.

[14] F. Boudot. "Efficient Proofs that a Committed Number Lies in an Interval". In: *EUROCRYPT 2000*. Ed. by B. Preneel. Vol. 1807. LNCS. Springer, Heidelberg, May 2000, pp. 431–444. DOI: 10.1007/3-540-45539-6_31.

[15] E. Bresson, D. Catalano, and D. Pointcheval. "A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications". In: *ASIACRYPT 2003*. Ed. by C.-S. Laih. Vol. 2894. LNCS. Springer, Heidelberg, 2003, pp. 37–54. DOI: 10.1007/978-3-540-40061-5_3.

[16] J. Buchmann and S. Hamdy. *A Survey on {IQ} Cryptography*. 2001. URL: http://tubiblio.ulb.tu-darmstadt.de/100933/.

[17] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018, pp. 315–334. DOI: 10.1109/SP.2018.00020.

[18] J. Camenisch, A. Kiayias, and M. Yung. "On the Portability of Generalized Schnorr Proofs". In: *EUROCRYPT 2009*. Ed. by A. Joux. Vol. 5479. LNCS. Springer, Heidelberg, Apr. 2009, pp. 425–442. DOI: 10.1007/978-3-642-01001-9_25.

[19] J. Camenisch and M. Michels. "Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes". In: *EUROCRYPT'99*. Ed. by J. Stern. Vol. 1592. LNCS. Springer, Heidelberg, May 1999, pp. 107–122. DOI: 10.1007/3-540-48910-X_8.

[20] J. Camenisch and M. Michels. "Separability and Efficiency for Generic Group Signature Schemes". In: *CRYPTO'99*. Ed. by M. J. Wiener. Vol. 1666. LNCS. Springer, Heidelberg, Aug. 1999, pp. 413–430. DOI: 10.1007/3-540-48405-1_27.

[21] R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis, and U. Peled. "UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts". In: *ACM CCS 20*. Ed. by J. Ligatti, X. Ou, J. Katz, and G. Vigna. ACM Press, Nov. 2020, pp. 1769–1787. DOI: 10.1145/3372297.3423367.

[22] G. Castagnos and F. Laguillaumie. "Linearly Homomorphic Encryption from DDH". In: *CT-RSA 2015*. Ed. by K. Nyberg. Vol. 9048. LNCS. Springer, Heidelberg, Apr. 2015, pp. 487–505. DOI: 10.1007/978-3-319-16715-2_26.

[23] D. Catalano, D. Pointcheval, and T. Pornin. "IPAKE: Isomorphisms for Password-based Authenticated Key Exchange". In: *CRYPTO 2004*. Ed. by M. Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 477–493. DOI: 10.1007/978-3-540-28628-8_29.

[24] P. Chaidos and G. Couteau. "Efficient Designated-Verifier Non-interactive Zero-Knowledge Proofs of Knowledge". In: *EUROCRYPT 2018, Part III*. Ed. by J. B.

Nielsen and V. Rijmen. Vol. 10822. LNCS. Springer, Heidelberg, 2018, pp. 193–221. DOI: 10.1007/978-3-319-78372-7_7.

[25] A. H. Chan, Y. Frankel, and Y. Tsiounis. "Easy Come - Easy Go Divisible Cash". In: *EUROCRYPT'98*. Ed. by K. Nyberg. Vol. 1403. LNCS. Springer, Heidelberg, 1998, pp. 561–575. DOI: 10.1007/BFb0054154.

[26] G. Couteau, M. Klooß, H. Lin, and M. Reichle. "Efficient Range Proofs with Transparent Setup from Bounded Integer Commitments". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2021, pp. 247–277.

[27] G. Couteau, T. Peters, and D. Pointcheval. "Removing the Strong RSA Assumption from Arguments over the Integers". In: *EUROCRYPT 2017, Part II*. Ed. by J.-S. Coron and J. B. Nielsen. Vol. 10211. LNCS. Springer, Heidelberg, 2017, pp. 321–350. DOI: 10.1007/978-3-319-56614-6_11.

[28] R. Cramer. "Modular design of secure yet practical cryptographic protocols". In: *Ph. D. Thesis, CWI and University of Amsterdam* (1996).

[29] R. Cramer and I. Damgård. "On the Amortized Complexity of Zero-Knowledge Protocols". In: *CRYPTO 2009*. Ed. by S. Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 177–191. DOI: 10.1007/978-3-642-03356-8_11.

[30] R. Cramer, I. Damgård, and M. Keller. "On the Amortized Complexity of Zero-Knowledge Protocols". In: *Journal of Cryptology* 27.2 (Apr. 2014), pp. 284–316. DOI: 10.1007/s00145-013-9145-x.

[31] R. Cramer and V. Shoup. "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption". In: *EUROCRYPT 2002*. Ed. by L. R. Knudsen. Vol. 2332. LNCS. Springer, Heidelberg, 2002, pp. 45–64. DOI: 10.1007/3-540-46035-7_4.

[32] I. Damgård. "On Σ-protocols". In: *Lecture Notes, University of Aarhus, Department for Computer Science* (2002). Accessed: 16/02/2022, p. 84.

[33] I. Damgård, N. Fazio, and A. Nicolosi. "Non-interactive Zero-Knowledge from Homomorphic Encryption". In: *TCC 2006*. Ed. by S. Halevi and T. Rabin. Vol. 3876. LNCS. Springer, Heidelberg, Mar. 2006, pp. 41–59. DOI: 10.1007/11681878_3.

[34] I. Damgård and E. Fujisaki. "A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order". In: *ASIACRYPT 2002*. Ed. by Y. Zheng. Vol. 2501. LNCS. Springer, Heidelberg, Dec. 2002, pp. 125–142. DOI: 10.1007/3-540-36178-2_8.

[35] I. Damgård and M. Jurik. "A Length-Flexible Threshold Cryptosystem with Applications". In: *ACISP 03*. Ed. by R. Safavi-Naini and J. Seberry. Vol. 2727. LNCS. Springer, Heidelberg, July 2003, pp. 350–364. DOI: 10.1007/3-540-45067-X_30.

[36] I. Damgård and M. Jurik. "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System". In: *PKC 2001*. Ed. by K. Kim. Vol. 1992. LNCS. Springer, Heidelberg, Feb. 2001, pp. 119–136. DOI: 10.1007/3-540-44586-2_9.

[37] I. Damgård and M. Koprowski. "Generic Lower Bounds for Root Extraction and Signature Schemes in General Groups". In: *EUROCRYPT 2002*. Ed. by L. R. Knudsen. Vol. 2332. LNCS. Springer, Heidelberg, 2002, pp. 256–271. DOI: 10.1007/3-540-46035-7_17.

[38] S. Dobson, S. D. Galbraith, and B. Smith. *Trustless Groups of Unknown Order with Hyperelliptic Curves*. Cryptology ePrint Archive, Report 2020/196. https://eprint.iacr.org/2020/196. 2020.

[39] E. Fujisaki and T. Okamoto. "A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications". In: *EUROCRYPT'98*. Ed. by K. Nyberg. Vol. 1403. LNCS. Springer, Heidelberg, 1998, pp. 32–46. DOI: 10.1007/BFb0054115.

[40] E. Fujisaki and T. Okamoto. "Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations". In: *CRYPTO'97*. Ed. by B. S. Kaliski Jr. Vol. 1294. LNCS. Springer, Heidelberg, Aug. 1997, pp. 16–30. DOI: 10.1007/BFb0052225.

[41] R. Gennaro, D. Micciancio, and T. Rabin. "An Efficient Non-Interactive Statistical Zero-Knowledge Proof System for Quasi-Safe Prime Products". In: *ACM CCS 98*. Ed. by L. Gong and M. K. Reiter. ACM Press, Nov. 1998, pp. 67–72. DOI: 10.1145/288090.288108.

[42] S. Goldberg, L. Reyzin, O. Sagga, and F. Baldimtsi. "Efficient Noninteractive Certification of RSA Moduli and Beyond". In: *ASIACRYPT 2019, Part III*. Ed. by S. D. Galbraith and S. Moriai. Vol. 11923. LNCS. Springer, Heidelberg, Dec. 2019, pp. 700–727. DOI: 10.1007/978-3-030-34618-8_24.

[43] S. Goldwasser and D. Kharchenko. "Proof of Plaintext Knowledge for the Ajtai-Dwork Cryptosystem". In: *TCC 2005*. Ed. by J. Kilian. Vol. 3378. LNCS. Springer, Heidelberg, Feb. 2005, pp. 529–555. DOI: 10.1007/978-3-540-30576-7_29.

[44] J. Groth. "Non-interactive Zero-Knowledge Arguments for Voting". In: *ACNS 05*. Ed. by J. Ioannidis, A. Keromytis, and M. Yung. Vol. 3531. LNCS. Springer, Heidelberg, June 2005, pp. 467–482. DOI: 10.1007/11496137_32.

[45] C. Hazay, G. L. Mikkelsen, T. Rabin, T. Toft, and A. A. Nicolosi. "Efficient RSA Key Generation and Threshold Paillier in the Two-Party Setting". In: *Journal of Cryptology* 32.2 (Apr. 2019), pp. 265–323. DOI: 10.1007/s00145-017-9275-7.

[46] Y. Ishai, R. Ostrovsky, and V. Zikas. "Secure Multi-Party Computation with Identifiable Abort". In: *CRYPTO 2014, Part II*. Ed. by J. A. Garay and R. Gennaro. Vol. 8617. LNCS. Springer, Heidelberg, Aug. 2014, pp. 369–386. DOI: 10.1007/978-3-662-44381-1_21.

[47] P. Kirchner and P.-A. Fouque. *Getting Rid of Linear Algebra in Number Theory Problems*. Cryptology ePrint Archive, Report 2020/1619. https://ia.cr/2020/1619. 2020.

[48] A. Kosba, C. Papamanthou, and E. Shi. "xJsnark: A framework for efficient verifiable computation". In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 944–961.

[49] S. Kunz-Jacques, G. Martinet, G. Poupard, and J. Stern. "Cryptanalysis of an Efficient Proof of Knowledge of Discrete Logarithm". In: *PKC 2006*. Ed. by M. Yung, Y. Dodis, A. Kiayias, and T. Malkin. Vol. 3958. LNCS. Springer, Heidelberg, Apr. 2006, pp. 27–43. DOI: 10.1007/11745853_3.

[50] J. Lee. *The security of Groups of Unknown Order based on Jacobians of Hyperelliptic Curves*. Cryptology ePrint Archive, Report 2020/289. https://eprint.iacr.org/2020/289. 2020.

[51] H. Lipmaa. "On Diophantine Complexity and Statistical Zero-Knowledge Arguments". In: *ASIACRYPT 2003*. Ed. by C.-S. Laih. Vol. 2894. LNCS. Springer, Heidelberg, 2003, pp. 398–415. DOI: 10.1007/978-3-540-40061-5_26.

[52] U. M. Maurer. "Abstract Models of Computation in Cryptography (Invited Paper)". In: *10th IMA International Conference on Cryptography and Coding*. Ed. by N. P. Smart. Vol. 3796. LNCS. Springer, Heidelberg, Dec. 2005, pp. 1–12.

[53] A. Ozdemir, R. Wahby, B. Whitehat, and D. Boneh. "Scaling verifiable computation using efficient set accumulators". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, pp. 2075–2092.

[54] P. Paillier. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". In: *EUROCRYPT'99*. Ed. by J. Stern. Vol. 1592. LNCS. Springer, Heidelberg, May 1999, pp. 223–238. DOI: 10.1007/3-540-48910-X_16.

[55] R. Pass, a. shelat, and V. Vaikuntanathan. "Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One". In: *CRYPTO 2006*. Ed. by C. Dwork. Vol. 4117. LNCS. Springer, Heidelberg, Aug. 2006, pp. 271–289. DOI: 10.1007/11818175_16.

[56] C. Peikert and B. Waters. "Lossy trapdoor functions and their applications". In: *40th ACM STOC*. Ed. by R. E. Ladner and C. Dwork. ACM Press, May 2008, pp. 187–196. DOI: 10.1145/1374376.1374406.

[57] M. O. Rabin and J. O. Shallit. "Randomized algorithms in number theory". In: *Communications on Pure and Applied Mathematics* 39.S1 (1986), S239–S256.

[58] C.-P. Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *CRYPTO'89*. Ed. by G. Brassard. Vol. 435. LNCS. Springer, Heidelberg, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22.

[59] V. Shoup. "Lower Bounds for Discrete Logarithms and Related Problems". In: *EUROCRYPT'97*. Ed. by W. Fumy. Vol. 1233. LNCS. Springer, Heidelberg, May 1997, pp. 256–266. DOI: 10.1007/3-540-69053-0_18.

[60] B. Terelius and D. Wikström. "Efficiency Limitations of S-Protocols for Group Homomorphisms Revisited". In: *SCN 12*. Ed. by I. Visconti and R. D. Prisco. Vol. 7485. LNCS. Springer, Heidelberg, Sept. 2012, pp. 461–476. DOI: 10.1007/978-3-642-32928-9_26.

[61] J. van de Graaf and R. Peralta. "A Simple and Secure Way to Show the Validity of Your Public Key". In: *CRYPTO'87*. Ed. by C. Pomerance. Vol. 293. LNCS. Springer, Heidelberg, Aug. 1988, pp. 128–134. DOI: 10.1007/3-540-48184-2_9.

[62] T. H. Yuen, Q. Huang, Y. Mu, W. Susilo, D. S. Wong, and G. Yang. "Efficient non-interactive range proof". In: *International Computing and Combinatorics Conference*. Springer. 2009, pp. 138–147.