Zero-Knowledge Arguments for Subverted RSA Groups

Dimitris Kolonelos$^{\star 1,2}$, Mary Maller^{3,4}, Mikhail Volkhov$^{\star 5}$

[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] PQShield, UK
[5] 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 present a NIZK range proof for general homomorphisms such 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 is reusable, can be maliciously generated and is linear in the number of proofs to be verified.

Update August 2023: Samuel Ranellucci reported a serious security issue with the sigma protocol in Fig. 2. Please see Section 1.2 for more details.

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 [62, 29] 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, 64], 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, 43, 36, 28]. 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 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[6] 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

[$\star$] Most of the work was done while the first and third authors were interns at Ethereum Foundation.
[6] By 'natural' we mean a protocol that works directly for the underlying language and does not involve NP-reductions.

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 [35, 58]. Currently the only known method to construct RSA-groups is via a trusted setup [49]. 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. [27]). 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 [41, 51, 53].

Further, the potential for N to be subverted is a delicacy which is rarely considered when using the additively homomorphic Paillier [57] 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. [49], 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 Appendix A . As a second example, in the Damgard-Jurik voting scheme [39], 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 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 use 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 1.4. We discuss how our range proof can be applied for non-injective homomorphisms in Appendix C.3.

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. [31] (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, [50]).

1.2 An issue with the relations of our protocols (Update August 2023)

Please be aware that a serious error was reported to us in August 2023. We have kept this work available for historical purposes but at present we cannot recommend its use in real world systems. We would like to thank Samuel Ranellucci not only for finding the mistake but also for communicating it to us in a professional manner.

The claim that our protocols can prove relations of the form $Y = \psi(w)$, $\psi : \mathcal{D} \to \mathbb{H}$ is incorrect. In fact, the extractors of our protocols can only extract w for the relaxed statement $Y = u \cdot \psi(w)$, where $u \in \mathbb{H}$ is a element of low ($\mathsf{poly}(\lambda)$) order. In (non-subverted) groups of unknown order such as Class Groups or the coset of an RSA group $\mathbb{Z}_N^*/\{1, -1\}$ elements of low order cannot be found (computationally in the former, information-theoretically in the latter). Therefore over these groups our protocols can prove the exact statement $Y = \psi(w)$ and this issue doesn't appear. However, in Subverted RSA groups, it cannot be excluded that low order elements exist, thus our protocols over Subverted RSA groups only prove statements of the relaxed form $(Y, \psi; u, w) : Y = u \cdot \psi(w)$. It is application-dependent whether this relaxed statement can be meaningful or not.

Technically, in the proof of 4.1, in case 4.1 (and 4.2) it is not true that $(\psi(s_1^{(j)})Y^{-1})^{c^{(j)}} = 1$ translates to $\psi(s_1^{(j)})Y^{-1} = 1$ but in fact it can traslate to $u \cdot \psi(s_1^{(j)})Y^{-1} = 1$ if $u^{c^{(j)}} = 1$. That happens with noticeable probability only if u is a low order element of order ℓ and ℓ is a factor of $c^{(j)}$ (the latter happens with probability $\approx 1/\ell$).

In future we plan to make a holistic update to the paper to remove any false claims.

1.3 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. Examples include the Fujisaki-Okamoto solution [43, 36, 28], the protocols of [19, 8] and the solution by Boneh et al. [13]. These are computationally-sound and thus would be insecure subverted RSA groups where the prover knows the group order. For specific relations, [39, 37] 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. [30, 31] presents a transformation that allows the protocol to have negligible soundness error when proving λ statements simultaneously. For a single proof it cannot be applied. Finally, Bangerter et al. [6] and Terelius et al. [64] 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 [56]. Their work improves upon xJsnark [52]. 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 prover overhead, however we do require a designated verifier.

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 [42, 26, 21, 36, 19, 12, 28]. Boudot [15] presents the first range proof for general range $[L, R]$ with slackness 1 (i.e. the message lies exactly in $m \in [0 \ldots R]$ as opposed to some extended range $m \in [0 \ldots \delta R]$). Further [15] uses a so-called four-squares integer decomposition property, a technique which is later used and improved in [54, 47, 66]. None of these works consider a subverted modulus.

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

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

Proofs of correct form of moduli. An orthogonal to the above line of work intends to prove that the group itself is not subverted [65, 44, 20, 10, 3, 45], 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 [24] or threshold ECDSA signatures [22]. All these solutions require parallel repetitions of the sigma protocol 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. Proving that an RSA group is the product of two safe primes is considerably more challenging than just proving biprimality [20].

1.4 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, 64].

Let $\psi : \mathcal{D} \to \mathbb{H}$ be a group homomorphism where $\mathbb{H}$ is an RSA-related group, such as exponentations $w \mapsto g^w$ 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, 64] 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

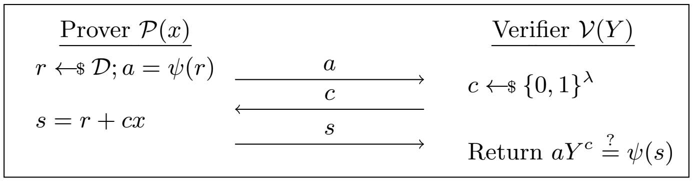
Prover $\mathcal{P}(x)$
Verifier $\mathcal{V}(Y)$
$r \leftarrow\!\!\$\ \mathcal{D}; a = \psi(r)$
a
$c \leftarrow\!\!\$\ \{0,1\}^\lambda$
c
$s = r + cx$
s
Return $aY^c \overset{?}{=} \psi(s)$

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.

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.

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 [35, 25]

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

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

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.

1.5 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 [43, 36, 28] 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 [17]. Solutions relying on class groups are outside the scope of this work.

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 [18, 27] 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 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 $[-\left\lfloor \frac{n}{2} \right\rfloor \ldots \left\lfloor \frac{n}{2} \right\rfloor]$. 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 [57, 39, 33, 16] in the RSA setting, the Castagnos-Laguillaumie cryptosystem over class groups [23] and schemes from lattices [46, 59]. Notably, no additively homomorphic public-key cryptosystems from groups of prime order exist.[10]

[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.

Paillier encryption scheme. We briefly recall the Paillier public key encryption scheme [57], and refer the reader to Appendix B for more details.

$\mathsf{KeyGen}(1^\lambda)$**:** 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 $\varSigma$-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 $\varSigma$-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 [29], 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 $\varSigma$-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 [33, 16]), are not known to be special, thus even the above extraction technique fails unless challenges are binary ($c_1 - c_2 = 1$).

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

[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 $N \neq 0 \mod \phi(N^2)$).

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 are below.

Definition 2.1 (DV Completeness). *An interactive protocol* $(\mathsf{KeyGen}, \mathcal{P}, \mathcal{V})$ *is statistically complete w.r.t. relation* $\mathcal{R}$ *if for all* $(\mathsf{vsk}, \mathsf{vpk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $(x, w) \in \mathcal{R}$*:*

$$\Pr[\langle \mathcal{P}(w), \mathcal{V}(\mathsf{vsk})\rangle(\mathsf{vpk}, x) = 1] = 1 - \mathsf{negl}(\lambda)$$

In the following notion we formalize soundness holding when $\mathcal{V}$ replies to at most Q verification queries (reminiscent of the bounded IND-CCA decryption oracle).

Definition 2.2 (DV Reusable Knowledge Soundness). *An interactive protocol* $(\mathsf{KeyGen}, \mathcal{P}, \mathcal{V})$ *is knowledge sound with soundness error* $\kappa(\lambda)$ *with* Q*-times reusable VPK w.r.t.* $\mathcal{R}$ *if the following holds.*

Let $\mathcal{A}$ *a malicious PPT prover, and* $(\mathsf{vsk}, \mathsf{vpk}, \tau) \leftarrow \mathsf{KeyGen}(1^\lambda, Q)$*. Let*

$$p(x) = \Pr[\langle \mathcal{A}^{\mathcal{O}^{\mathsf{Verify}}(\mathsf{vsk}, \cdot)}(1^\lambda), \mathcal{V}(\mathsf{vsk})\rangle(\mathsf{vpk}, x) = 1]$$

where $\mathcal{A}$ *has access to the non-rewindable verification oracle* $\mathcal{O}^{\mathsf{Verify}}(\mathsf{vsk}, \cdot, \cdot)$ *that it can query, on any* (x, π)*, at most* Q *times. Then for all* x *of size* λ*, if* $p(x) > \kappa(\lambda)$*, then there exists a PPT extractor* Ext *s.t.* $\mathsf{Ext}^{\mathcal{A}}(\mathsf{vpk}, \tau, x)$ *returns* w *satisfying* $(x, w) \in \mathcal{R}$*, and* Ext *terminates in expected number of steps* $\mathsf{poly}(\lambda)/(p(x) - \kappa(\lambda))$*.*

In practice, VPK is split into the verifier's encryption key, and a part resembling a reference string, where only this reference string has to be regenerated when the VRS is expired.

Definition 2.3 (DV Honest-Verifier Zero-Knowledge). *An interactive protocol* $(\mathsf{KeyGen}, \mathcal{P}, \mathcal{V})$ *is statistical honest-verifier zero-knowledge w.r.t.* $\mathcal{R}$ *if* $\forall (x, w) \in \mathcal{R}$ *there exists a PPT simulator* $\mathcal{S}$ *such that for* $(\mathsf{vsk}, \mathsf{vpk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *it holds that:*

$$\langle \mathcal{P}(w), \mathcal{V}(\mathsf{vsk})\rangle(\mathsf{vpk}, x) \overset{s}{\approx} \mathcal{S}(\mathsf{vsk}, \mathsf{vpk}, x)$$

We further consider the notion of zero-knowledge under maliciously generated verifier's public VPK. This notion captures the scenario where VPK is generated by an adversarial, untrusted verifier which still cannot break zero-knowledge. The definition was introduced by Quach et. al. [60] in the context of DV NIZKs, but we modify it for the interactive DV case. We note that although the adversary can subvert the setup, we still consider the verifier honest during the protocol execution (HVZK), having in mind Fiat-Shamir transforming this public-coin part of the interaction.

Definition 2.4 (DV HVZK Under Malicious VPK). *An interactive protocol* $(\mathsf{KeyGen}, \mathcal{P}, \mathcal{V})$ *is statistical honest-verifier zero-knowledge under maliciously-generated VPK w.r.t.* $\mathcal{R}$ *if* $\forall (x, w) \in \mathcal{R}$ *there exists a PPT simulator* $\mathcal{S}$ *such that for any PPT adversary* $\mathcal{A}$ *and* $(\mathsf{vsk}, \mathsf{vpk}) \leftarrow \mathcal{A}(1^\lambda)$ *it holds that:*

$$\langle \mathcal{P}(w), \mathcal{V}(\mathsf{vsk})\rangle(\mathsf{vpk}, x) \overset{s}{\approx} \mathcal{S}(\mathsf{vsk}, \mathsf{vpk}, x)$$

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 3.1, that describes how to extract a witness given M accepting transcripts such that the gcd of the challenges is 1. Our second lemma, Lemma 3.2, 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 3.1 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 Appendix D.1.

Lemma 3.1. *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 3.2 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 3.2 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 3.2 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 [1, Theorem 8] (that generalizes [34]) 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 3.2 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 3.2. *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 $(e^{(1)} \ldots e^{(M)})^T = \boldsymbol{B'}\boldsymbol{c}$. Then:

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

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

Proof. The proof goes as follows:

- The probability that $e^{(1)}, \ldots, e^{(M)}$ are coprime follows from the probability that no prime number q divides all $e^{(1)}, \ldots, e^{(\lambda)}$ at the same time:

$$\Pr\left[\gcd(e^{(1)}, e^{(2)}, \ldots, e^{(M)}) = 1\right] = \prod_{q,\ \text{prime}} \left(1 - \Pr\left[q \mid e^{(1)}, e^{(2)}, \ldots, e^{(M)}\right]\right)$$

where we abuse the notation with $\left(q \mid e^{(1)}, e^{(2)}, \ldots, e^{(M)}\right)$ to denote $(q \mid e^{(1)} \wedge q \mid e^{(2)} \wedge \ldots \wedge q \mid e^{(M)})$. The event on the right side of equation is equivalent to:

$$\begin{pmatrix} e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(M)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \pmod{q} \tag{1}$$

for every prime q.

Each $e \in \mathbb{Z}$ can be written as $e = \boldsymbol{b}\boldsymbol{c}^\top$, where $\boldsymbol{b} = (b_1\ b_2 \ldots b_\lambda) \in \{0,1\}^\lambda$, and $\boldsymbol{c} = (c_1\ c_2 \ldots c_\lambda) \in \left(\llbracket 2^\lambda \rrbracket\right)^\lambda$. Let the matrix of all possible e's, $\boldsymbol{E}$ resulting from $\boldsymbol{B}$ be:

$$\boldsymbol{E} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{\epsilon' 2^\lambda} \end{pmatrix} = \boldsymbol{B}\boldsymbol{c}^\top = \begin{pmatrix} b_{1,1} & b_{1,2} & \ldots & b_{1,\lambda} \\ b_{2,1} & b_{2,2} & \ldots & b_{2,\lambda} \\ \vdots & \vdots & \ddots & \vdots \\ b_{\epsilon' 2^\lambda,1} & b_{\epsilon' 2^\lambda,2} & \ldots & b_{\epsilon' 2^\lambda,\lambda} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_\lambda \end{pmatrix}$$

- To start with, we show that for any prime q, $\operatorname{rank}(\boldsymbol{B}) \geq 1 + \left\lceil \log_q(\epsilon' 2^{\lambda-1}) \right\rceil$ over $\mathbb{F}_q$. To show this, first recall that any n linearly independent vectors over $\mathbb{F}_q$ span at most q^n vectors: let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in (\mathbb{F}_q)^\lambda$ be n linearly independent vectors, then their span is $a_1\boldsymbol{x}_1 + a_2\boldsymbol{x}_2 + \ldots + a_n\boldsymbol{x}_n$ for $a_i \in \mathbb{F}_q$, so there at most q^n different coefficients $\{a_i\}_{i=1}^n$.

  In our case a more fine-grained analysis shows that n linearly independent vectors span at most $2q^{n-1}$ *binary* vectors in $\mathbb{F}_q$:

  $$a_1\boldsymbol{b}_1 + a_2\boldsymbol{b}_2 + \ldots + a_n\boldsymbol{b}_n = \begin{pmatrix} a_1 b_{1,1} + a_2 b_{2,1} + \ldots + a_n b_{n,1} \\ a_1 b_{1,2} + a_2 b_{2,2} + \ldots + a_n b_{n,2} \\ \vdots \\ a_1 b_{1,\lambda} + a_2 b_{2,\lambda} + \ldots + a_n b_{n,\lambda} \end{pmatrix}$$

  In our case only linear combinations that give a vector in $\{0,1\}^\lambda$ are valid. So for each $i \in [\lambda]$ we have that $\sum_{j=1}^n a_j b_{j,i}$ should be 0 or 1. Take a row i^* where not all b_{j,i^*} are 0 (there is such a row otherwise all $\boldsymbol{b}_j = 0$). This row restricts the a_j's to at most $2q^{n-1}$ valid: let the row having k number of 0's and $n-k$ number of 1's $(0 \leq k \leq n-1)$, wlog $b_{1,i^*} = 0, \ldots b_{k,i^*} = 0, b_{k+1,i^*} = 1, \ldots, b_{n,i^*} = 1$. For the 0-values any a_j is valid which gives us q^k combinations. For the 1-values we have the restriction that $a_{k+1} + \ldots + a_n \in \{0,1\}$ and from a simple combinatorial argument we can see that this gives $2q^{n-k-1}$ combinations. So overall at most $q^k \cdot 2q^{n-k-1} = 2q^{n-1}$ combinations are valid.

  Conversely, the minimal number of $\mathbb{F}_q$-vectors that can span a set of $\epsilon' 2^\lambda$ *binary* vectors is $d(q) := 1 + \left\lceil \log_q(\epsilon' 2^\lambda/2) \right\rceil = 1 + \left\lceil \log_q(\epsilon' 2^{\lambda-1}) \right\rceil$, meaning that there are at least $d(q)$ linearly independent rows in $\boldsymbol{B}$. Where unambiguous, we write d for simplicity.

- Now, let $\boldsymbol{B'} = \left(\boldsymbol{b}^{(1)}\ \boldsymbol{b}^{(2)} \ldots\ \boldsymbol{b}^{(M)}\right)^T$ be a matrix consisting of uniformly random rows picked from $\boldsymbol{B}$ (the resulting $\boldsymbol{b}^{(i)}$ are pairwise different with overwhelming probablity). Again in the vector form:

  $$\boldsymbol{E'} = \begin{pmatrix} e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(M)} \end{pmatrix} = \boldsymbol{B'}\boldsymbol{c}^\top = \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \ldots & b_\lambda^{(1)} \\ b_1^{(2)} & b_2^{(2)} & \ldots & b_\lambda^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ b_1^{(M)} & b_2^{(M)} & \ldots & b_\lambda^{(M)} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_\lambda \end{pmatrix}$$

We claim that for any q and $M = O(\lambda \log \lambda)$, $\operatorname{rank}(\boldsymbol{B'}) = \operatorname{rank}(\boldsymbol{B}) \geq d(q)$ (over $\mathbb{F}_q$) with overwhelming probability.

The argument is by induction. Let the sampling (conceptually) proceed in $(d-1)$ steps, where at each step i we sample $\hat{k}(i)$ rows of $\boldsymbol{B}$, denote $\hat{\boldsymbol{b}}^{(i,1)}, \ldots, \hat{\boldsymbol{b}}^{(i,k(i))}$, until the overall sampled-until-now rows, denoted:

$$\hat{\boldsymbol{B}}'_i := \left(\hat{\boldsymbol{b}}^{(1,1)} \ \ldots \ \hat{\boldsymbol{b}}^{(1,k(1))} \ \ldots \ \hat{\boldsymbol{b}}^{(i,1)} \ \ldots \ \hat{\boldsymbol{b}}^{(i,k(i))}\right)$$

have rank i i.e. $\operatorname{rank}(\hat{\boldsymbol{B}}'_i) = i$. As a base case we have $\hat{k}(1) = 1$ and $\operatorname{rank}(\hat{\boldsymbol{B}}'_1) = 1$.

We claim that at each step $i \in [1, d-1]$ the number of samples needed (to reach $\operatorname{rank}(\hat{\boldsymbol{B}}'_i) = i$) is at most $\hat{k}(i) \leq \frac{\lambda}{(d-i)\log q}$ elements with overwhelming probability $1 - \mathsf{negl}(\lambda)$. In other words, the probability to sample $\hat{k}(i)$ row-vectors from $\boldsymbol{B}$ and *not* find a vector outside the span of the sampled in previous steps vectors $\hat{\boldsymbol{B}}'_{i-1} := \left(\hat{\boldsymbol{b}}^{(1,1)} \ldots \hat{\boldsymbol{b}}^{(1,k(1))}, \ldots, \hat{\boldsymbol{b}}^{(i-1,1)} \ldots \hat{\boldsymbol{b}}^{(i-1,k(i-1))}\right)$ is negligible. To see this, assume that $\operatorname{rank}(\hat{\boldsymbol{B}}'_{i-1}) \geq i-1$ (inductive hypothesis) and consider the probability for a *single* uniformly sampled $\hat{\boldsymbol{b}}^{(i,1)}$ to be in the span of $\hat{\boldsymbol{B}}'_{i-1}$:

$$\begin{aligned}
&\Pr\left[\hat{\boldsymbol{b}}^{(i,1)} \in \operatorname{Span}(\hat{\boldsymbol{B}}'_{i-1}) \mid \hat{\boldsymbol{b}}^{(i,1)} \leftarrow\!\$\ \boldsymbol{B}\right] \\
&\quad = \frac{|\operatorname{Span}(\hat{\boldsymbol{B}}'_{i-1}) \cap \boldsymbol{B}|}{|\boldsymbol{B}|} \\
&\quad \leq \frac{2q^{i-2}}{\epsilon' 2^{\lambda}} = \frac{q^{d-(d-i+2)}}{\epsilon' 2^{\lambda}} = \frac{1}{q^{d-i+2}} \frac{q^{1+\lceil \log_q(\epsilon' 2^{\lambda-1}) \rceil}}{\epsilon' 2^{\lambda}} \\
&\quad \leq \frac{1}{q^{d-i+2}} \frac{q^{\log_q(\epsilon' 2^{\lambda-1})+2}}{\epsilon' 2^{\lambda}} = \frac{1}{2q^{d-i}} \\
&\quad < \frac{1}{q^{d-i}}
\end{aligned}$$

The first ($\leq$) transition is crucial: in $\mathbb{F}_q$ the maximum number of binary vectors that is spanned by a n-dimensional subspace is at most $2q^{n-1}$, so any initial matrix $\boldsymbol{B}$ cannot contain overwhelmingly big subspaces of low rank.

This argument generalizes: if we sample k vectors instead of just one at each step, the probability of them all to be in the span of previous vectors is $\leq 1/q^{k(d-i)}$ for any choice of $\boldsymbol{B}$. So when k is chosen to be $k := \lambda/((d-i)\log q)$, this probability becomes $1/2^{\lambda} = \mathsf{negl}(\lambda)$. Therefore, if we query k values at step i we will get at least one new linearly independent vector. So $\hat{k}(i)$ is at most $\lambda/((d-i)\log q)$.

This means that in total for all steps, a uniformly random choice of

$$\hat{k}(1) + \ldots \hat{k}(d-1) \leq \frac{\lambda}{\log q} \sum_{i=1}^{d-1} \frac{1}{d-i} = O\left(\frac{\lambda \log d}{\log q}\right)$$

vectors, has rank d with overwhelming probability.

Hence, if we set $M = O\left(\frac{\lambda \log d}{\log q}\right)$ then $\operatorname{rank}(\boldsymbol{B'}) \geq d$ over $\mathbb{F}_q$ with overwhelming probability. For that we set $M = O(\lambda \log d) = O(\lambda \log \lambda)$ so that $\operatorname{rank}(\boldsymbol{B'}) \geq d$ over any $\mathbb{F}_q$, $q \geq 2$.

- Now we want to find an M such that $\mathrm{rank}(\boldsymbol{B'}) \geq d(q)$ over $\mathbb{F}_q$ for *all* $q = \mathsf{poly}(\lambda)$. By a simple union-bound argument we get that $M = O(\lambda \log \lambda) \cdot \mathsf{poly}(\lambda) = \mathsf{poly}(\lambda)$ suffices. Therefore we set $M = \mathsf{poly}(\lambda)$. On the other hand, for superpolynomial moduli $q > \mathsf{poly}(\lambda)$ we have $\mathrm{rank}(\boldsymbol{B'}) \geq 2$ over $\mathbb{F}_q$, since a single element generates at most $2q^{1-1} = 2$ elements. This is summarized in the following lemma:

Lemma 3.3. *Let $M = \mathsf{poly}(\lambda)$ then:*

$$\mathrm{rank}(\boldsymbol{B'}) \geq d^*(q) := \begin{cases} d(q), & \textit{if } q = \mathsf{poly}(\lambda) \\ 2, & \textit{otherwise} \end{cases}$$

- Now we are ready to evaluate the probability that Eq. (1) holds, for any arbitrary prime q. Wlog assume that the first $d^*(q) \times d^*(q)$ sub-matrix has non-zero determinant (since $\mathrm{rank}(\boldsymbol{B'}) \geq d^*(q)$). After Gaussian elimination we get the equivalent system of equations:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & \tilde{b}^{(1)}_{d+1} & \dots & \tilde{b}^{(1)}_{\lambda} \\ 0 & 1 & \dots & 0 & \tilde{b}^{(2)}_{d+1} & \dots & \tilde{b}^{(2)}_{\lambda} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & \tilde{b}^{(d)}_{d+1} & \dots & \tilde{b}^{(d)}_{\lambda} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_d \\ c_{d+1} \\ \vdots \\ c_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \pmod{q}$$

or

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_d \\ \vdots \end{pmatrix} = - \begin{pmatrix} \sum_{i=d+1}^{\lambda} \tilde{b}^{(1)}_i c_i \\ \sum_{i=d+1}^{\lambda} \tilde{b}^{(2)}_i c_i \\ \vdots \\ \sum_{i=d+1}^{\lambda} \tilde{b}^{(d)}_i c_i \\ \vdots \end{pmatrix} \pmod{q}$$

Since $c_1, \dots, c_\lambda$ are uniformly sampled and independently the above probability is $(1/q)^d$ (for any $\tilde{b}^{(j)}_i$). Therefore

$$\prod_{q,\text{ prime}} \left(1 - \Pr[q \mid e^{(1)}, \dots, e^{(M)}]\right) \geq \prod_{q,\text{ prime}} \left(1 - \frac{1}{q^{d^*(q)}}\right)$$

Now we discern two cases. For polynomial-size primes, $q = \mathsf{poly}(\lambda)$:

$$\begin{aligned}
\prod_{\substack{q=\mathsf{poly}(\lambda)\\ q \text{ prime}}} \left(1 - \frac{1}{q^{d^*(q)}}\right) &= \prod_{\substack{q=\mathsf{poly}(\lambda)\\ q \text{ prime}}} \left(1 - \frac{1}{q^{1+\left\lceil \log_q(\epsilon' 2^{\lambda-1})\right\rceil}}\right)\\
&\geq \prod_{\substack{q=\mathsf{poly}(\lambda)\\ q \text{ prime}}} \left(1 - \frac{1}{q\epsilon' 2^{\lambda-1}}\right)\\
&> \left(1 - \frac{1}{q_{\min}\epsilon' 2^{\lambda-1}}\right)^{\#q}\\
&\geq 1 - \frac{\#q}{2\epsilon' 2^{2^{\lambda-1}}} = 1 - \frac{\mathsf{poly}(\lambda)}{\epsilon' 2^{\lambda-2}}\\
&= 1 - \mathsf{negl}(\lambda)
\end{aligned}$$

For super-polynomial size prime, $q > \mathsf{poly}(\lambda)$ (hence $1/q = \mathsf{negl}(\lambda)$), we use a more fine-grained analysis that implicitly groups products of the same bit-size. Let $B_{\min}$ and $B_{\max}$ be the minimum and maximum bit-size respectively of all the $q > \mathsf{poly}(\lambda)$:

$$\begin{aligned}
\prod_{\substack{q>\mathsf{poly}(\lambda)\\ q \text{ prime}}} \left(1 - \frac{1}{q^{d^*(q)}}\right) &= \prod_{\substack{q>\mathsf{poly}(\lambda)\\ q \text{ prime}}} \left(1 - \frac{1}{q^2}\right)\\
&= \prod_{k=B_{\min}}^{B_{\max}-1} \prod_{\substack{q=2^k\\ q \text{ prime}}}^{2^{k+1}} \left(1 - \frac{1}{q^2}\right)\\
&> \prod_{k=B_{\min}}^{B_{\max}-1} \left(1 - \frac{1}{q_{\min}^2}\right)^{\#q}\\
&\geq \prod_{k=B_{\min}}^{B_{\max}-1} \left(1 - \frac{2^k}{(2^k)^2}\right)\\
&= \prod_{k=B_{\min}}^{B_{\max}-1} \left(1 - \frac{1}{2^k}\right)\\
&= \prod_{k=B_{\min}}^{B_{\max}-1} (1 - \mathsf{negl}(\lambda))\\
&= 1 - (B_{\max} - 1 - B_{\min})\mathsf{negl}(\lambda)\\
&= 1 - \mathsf{negl}(\lambda)
\end{aligned}$$

where in the above we used the fact that $e^{(i)} \leq \lambda 2^\lambda$ thus $B_{\max} - B_{\min} > \lambda + \log\lambda$ and that $1/q_{\min} = \mathsf{negl}(\lambda)$.
We conclude that $\prod_{q,\text{ prime}} \left(1 - \Pr[q \mid e^{(1)}, \ldots, e^{(M)}]\right) = 1 - \mathsf{negl}(\lambda)$

- Finally, to conclude our proof we need to show that:

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

Recall that

$$\Pr[\gcd(e^{(2)} - e^{(1)}, \ldots, e^{M} - e^{(1)}) = 1] =$$
$$= \prod_{q,\text{ prime}} \left(1 - \Pr\left[q \mid (e^{(2)} - e^{(1)}), \ldots, (e^{(M)} - e^{(1)})\right]\right)$$

So we need to evaluate the probability:

$$\begin{aligned}
&\Pr\left[q \mid (e^{(2)} - e^{(1)}), \ldots, (e^{(M)} - e^{(1)})\right] = \\
=&\Pr\left[(e^{(2)} - e^{(1)}) = 0 \wedge \ldots \wedge (e^{(M)} - e^{(1)}) \pmod q\right] = \\
=&\sum_{k=0}^{q-1} \Pr\left[e^{(2)} = k \wedge \ldots \wedge e^{(M)} = k \pmod q\right] \Pr\left[e^{(1)} = k \pmod q\right] \\
=&q \cdot \Pr\left[e^{(2)} = 0 \wedge \ldots \wedge e^{(M)} = 0 \pmod q\right] \frac{1}{q}
\end{aligned}$$

The last equality comes from the fact that $\Pr[e^{(i)} = x \pmod q] = \Pr[e^{(i)} = y \pmod q]$ for any $x, y \in [q-1]$ since $\boldsymbol{c}$ is uniformly sampled.

□

Remark 3.1. Our protocols use binary values for $\boldsymbol{B}$. The above lemma generalizes for any choice of domain $\mathcal{D} \subseteq \mathbb{Z}$ that is polynomially bounded, $|\mathcal{D}| = \mathsf{poly}(\lambda)$, for the elements of $\boldsymbol{B}$.

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 3.1 and Lemma 3.2). 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).

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

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

$\underline{\mathcal{V}.\mathsf{KeyGen}(1^{\lambda})\text{:}}$ 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 $\mathsf{vpk} = (\mathsf{pk},\mathsf{ct}_1,\ldots,\mathsf{ct}_{\lambda})$, $\mathsf{vsk} = \mathsf{sk}$.

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

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

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.

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 4.1 (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 $(\mathsf{vpk}, \mathsf{vsk}, \tau) \overset{\$}{\leftarrow} \mathsf{KeyGen}(1^\lambda)$, 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 3.1, 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:

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

and outputs $\mathcal{T}$. For Ext_1 we use a generic $(M, 2)$-special soundness extractor (see [14, Lemma 1] and for a more fine-grained analysis [2, Lemma 5]), 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 $\left(a, b^{(1)}, S^{(1)}, T^{(1)}, d^{(1,1)}, u_1^{(1,1)}, u_2^{(1,1)}\right)$ such that $T^{(1)}(S^{(1)})^{d^{(1,1)}} = (C^{(1)})^{u_1^{(1,1)}} \mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,1)}; 0)$, 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 $\left(a, b^{(1)}, S^{(1)}, T^{(1)}, d^{(1,2)}, u_1^{(1,2)}, u_2^{(1,2)}\right)$ such that $T^{(1)}(S^{(1)})^{d^{(1,2)}} = (C^{(1)})^{u_1^{(1,2)}} \mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,2)}; 0)$. So we have:

$$\begin{aligned} T^{(1)}(S^{(1)})^{d^{(1,1)}} &= (C^{(1)})^{u_1^{(1,1)}} \mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,1)}; 0) \\ T^{(1)}(S^{(1)})^{d^{(1,2)}} &= (C^{(1)})^{u_1^{(1,2)}} \mathsf{Enc}_{\mathsf{pk}}(u_2^{(1,2)}; 0) \end{aligned}$$

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 $s^{(1)} = s_2^{(1)} + c^{(1)}s_1^{(1)}$

[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.

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 3.1 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 $\mathsf{IND\text{-}CPA}$ that, using Ext, wins the $\mathsf{IND\text{-}CPA}$ game:

```
$\underline{\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}}(\mathsf{pk})}$
$c_1, z_1, \ldots, c_\lambda, z_\lambda \xleftarrow{\$} [\![2^\lambda]\!]$
$\mathsf{ct}_i \xleftarrow{\$} \mathcal{O}_{\mathsf{Enc}}(c_i, z_i) \quad \text{for } i \in [\lambda];$
$\mathsf{vpk} \leftarrow (\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\lambda)$
$\mathsf{coin} \xleftarrow{\$} [1, 2^\lambda];\ j \leftarrow 1$
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 4.1. 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:

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

[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]).

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 4.2. 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 4.3. 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 3.2). We thus argue that if Ext fails then $\mathcal{B}$ succeeds with overwhelming probability.

Indeed Lemma 3.2 shows that $\Pr[\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1] = 1 - \mathsf{negl}(\lambda)$.

To see why Lemma 3.2 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 (see Appendix F).

The main HVZK result is as follows :

Theorem 4.2 (Honest Verifier Zero Knowledge). $\mathsf{DV}_{\mathsf{Prot}}$ *is statistical honest-verifier zero-knowledge for the relation* $\mathcal{R}_{\mathsf{Hom}}$*.*

Proof. Deferred to Appendix D.2. □

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 $c_i \in \llbracket 2^\lambda \rrbracket$ 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.

The malicious-verifier alternative key generation procedure is presented in Fig. 3 below. We edit the setup algorithm such that the verifier must provide a range proof on the ciphertexts it generates for vpk.

We prove range of the VPK ciphertext efficiently with $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$, presented in Appendix E, together with its security proof. The protocol follows the transformation by Cramer et al. [30, 31] 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 $c_i \in \llbracket 2^\lambda \rrbracket$, the resulting messages in the ciphertexts ct_i of vpk are in the extended interval $\llbracket 2^{3\lambda+\log\lambda-1} \rrbracket$ (the slack is $2^{2\lambda+\log\lambda-1}$). Therefore the encrypted sum-challenge $\mathcal{P}$ replies to is in $\llbracket 2^{3\lambda+2\log\lambda-1} \rrbracket$. 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 $s = r_1 + cw \overset{s}{\approx} r_1$.

$\mathcal{V}.\mathsf{KeyGen}(1^\lambda)$**:** Generate a VPK together with well-formedness proofs:

- Sample $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\$} \mathsf{AHE.KeyGen}(1^\lambda)$ with $\mathsf{pk} = N_{\mathsf{pk}} > |\mathcal{D}|2^{4\lambda+2\log\lambda}$.
- $c_1, \ldots, c_\lambda \xleftarrow{\$} \llbracket 2^\lambda \rrbracket$; $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(c_i)$ for $i \in [1, \lambda]$
- Obtain proof π_1 by running [45, Protocol $\mathcal{P}_{\mathsf{Paillier-N}}$] to certify that $\gcd(N_{\mathsf{pk}}, \phi(N_{\mathsf{pk}})) = 1$
- Obtain range proof π_2 by running $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$ for $x = \{\mathsf{ct}_i\}, w = \{c_i\}_{i=1}^{\lambda}$, using range bound $R = 2^\lambda$
- Return $\mathsf{vpk} = (\mathsf{pk}, \{\mathsf{ct}_i\}_{i=1}^{\lambda}, \pi_1, \pi_2)$, $\mathsf{vsk} = \mathsf{sk}$

$\mathcal{P}(\mathsf{vpk} = (\mathsf{pk}, \{\mathsf{ct}_i\}_{i=1}^{\lambda}, \pi_1, \pi_2), \psi, Y, w)$**:**

- Verify π_1 w.r.t. $\mathsf{pk} = N_{\mathsf{pk}}$
- Verify π_2 w.r.t. $\{\mathsf{ct}_i\}$ and $R = 2^{\lambda-1}$
- Pick $r_1 \leftarrow\!\!\$ \; \llbracket 2^{4\lambda+2\log\lambda-1}|\mathcal{D}| \rrbracket$
- Proceed as the honest prover in $\mathsf{DV}_{\mathsf{Prot}}$

Fig. 3. Alternative key generation protocol for the $\mathsf{DV}_{\mathsf{Prot}}$ protocol.

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 , as we explain in Appendix C.3. For this we use [45, 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 4.3. *Protocol* $\mathsf{DV}_{\mathsf{Prot}}$*, augmented with* KeyGen *and* $\mathcal{P}$ *from Fig. 3, is statistical honest-verifier zero-knowledge under malicious VPK for the relation* $\mathcal{R}_{\mathsf{Hom}}$*.*

Proof. Deferred to Appendix D.3. □

4.4 Reusable VPK

In this section we present $\mathsf{DVReusable}_{\mathsf{Prot}}$(Fig. 4), 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.

$\mathcal{V}.\mathsf{KeyGen}(1^\lambda, Q)$**:**

- Sample a key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{AHE.KeyGen}(1^\lambda)$ with $|\mathcal{M}| > 2^{3\lambda + \log\lambda}|\mathcal{D}|$
- Sample challenges uniformly $c_1, \dots, c_\lambda \xleftarrow{\$} [\![2^\lambda]\!]$ and $c_{\lambda+1}, \dots, c_{\lambda+Q} \xleftarrow{\$} [\![\lambda 2^{2\lambda}]\!]$
- Encrypt them $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(c_i)$ for each $i \in [\lambda + Q]$
- Initialize $\mathcal{U} \leftarrow \emptyset$
- Return $\mathsf{vpk} = (\mathsf{pk}, \mathsf{ct}_1, \dots, \mathsf{ct}_{\lambda+Q})$, $\mathsf{vsk} = \mathsf{sk}$

$\mathcal{P}(\mathsf{vpk}, \psi, Y, w, \kappa) \leftrightarrow \mathcal{V}(\mathsf{vsk}, \mathsf{vpk}, \psi, Y, \kappa)$**:** The prover and the verifier interact as in Fig. 2, except:

- Both parties take as input the round index κ
- r_1 is sampled from $[\![\lambda 2^{3\lambda}|\mathcal{D}|]\!]$
- $\mathcal{P}$ replies w.r.t. $C = \mathsf{ct}_{\lambda+\kappa} \prod_{i=1}^{\lambda} \mathsf{ct}_i^{b_i}$
- Before running the final checks, $\mathcal{V}$ checks if $\kappa \overset{?}{\in} \mathcal{U}$, and if yes, updates $\mathcal{U} \leftarrow \mathcal{U} \cup \{\kappa\}$
- $\mathcal{V}$ runs checks as before, but w.r.t. $c = c_{\lambda+\kappa} + \sum c_i b_i$

Fig. 4. $\mathsf{DVReusable}_{\mathsf{Prot}}$: A variant of our $\mathsf{DV}_{\mathsf{Prot}}$ with Q-times reusable VPK. Here the prover and verifier agree in advance on the instance κ of the protocol being run, and $\mathcal{V}$ makes sure each κ is used once.

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 4.4. $\mathsf{DVReusable}_{\mathsf{Prot}}$ *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 $Q(\lambda)$.*

Proof. Deferred to Appendix D.4. □

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 4.1, 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 variant of Paillier with randomness in the exponent ($S^{(j)} = (N+1)^{s^{(j)}} h^r$, we will refer to this variant as "lite Cramer-Shoup", see [16]), then our extractor can obtain $s^{(j)}$ for free in the generic group model [63, 55] (GGM).

GGM for unknown order groups has been established [40, 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 4.1 except that the extractor now uses whitebox access to the prover instead of the rewinding argument to find a representation for S.

Theorem 4.5 (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.*

Proof. Deferred to Appendix D.5. □

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 3.1.

On top of that, we use the range proof from [28] for RSA groups, to achieve the final range protocol. In particular, we choose this protocol to prove range since it is the state-of-the-art for (exact) range proofs over (non-subverted) RSA groups. In combination with our extraction technique gives an (exact) range proof over subverted RSA groups.

The protocol from [28] works over an integer commitment [43, 36] $c = g^x h^r$ 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 [28] (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. 6,and Fig. 5 describes its key generation.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.

Security. The above protocol consists of two sub-protocols: our protocol of Section 4.1 and the range proof by Couteau et. al. [28] 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 5.1. *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*

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

[17] The ± 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.

$\underline{\mathcal{V}.\mathsf{KeyGen}(1^\lambda, R_{\mathsf{max}}, Q)}$**:**

- Sample an RSA modulus N_{cm} secure w.r.t. λ, and random group elements $h \leftarrow\!\!\$\ \mathbb{Z}^*_{N_{\mathsf{cm}}}, g = h^f$, where $f \leftarrow\!\!\$\ \phi(N_{\mathsf{cm}})$.
- Sample a key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{AHE.KeyGen}(1^\lambda)$, with $N_{\mathsf{pk}} := \mathsf{pk} > 2^{6\lambda+2\log\lambda+4} N_{\mathsf{cm}} R_{\mathsf{max}}$.
- Sample challenges uniformly $c_1, \ldots, c_\lambda \leftarrow\!\!\$\ [\![2^\lambda]\!]$ and $c_{\lambda+1}, \ldots, c_{\lambda+Q} \leftarrow\!\!\$\ [\![\lambda 2^{2\lambda}]\!]$ and encrypt them $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(c_i)$ for each $i \in [\lambda + Q]$.
- Run[18] [45, Protocol $\mathcal{P}_{\mathsf{Paillier-N}}$] to prove $\gcd(N_{\mathsf{pk}}, \phi(N_{\mathsf{pk}})) = 1$; obtain proof π_1.
- Run $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$ on $\{(x = \mathsf{ct}_i, w = c_i)\}_{i=1}^{\lambda}$, using range bound $R = 2^\lambda$; obtain proof π_2. Similarly for $\{c_i\}_{i=\lambda+1}^{\lambda+Q}$ and $R = \lambda 2^{2\lambda}$ in batches of size λ; obtain π_3.
- Run $\mathsf{Sigma}_{\mathsf{Prot}}$ for $\{(g, f) : g = h^f \mod N_{\mathsf{cm}}\}$ (a variant of Fig. 7); obtain proof π_4.
- Return $\mathsf{vpk} = ((\mathsf{pk}, \pi_1), (\{\mathsf{ct}_i\}_{i\in[\lambda+Q]}, \pi_2, \pi_3), (N_{\mathsf{cm}}, g, h, \pi_4))$, $\mathsf{vsk} = \mathsf{sk}$.

$\underline{\mathcal{P}(\mathsf{vpk}, \psi, \ldots)}$**:**

- Verify π_1 w.r.t. $\mathsf{vpk}.N_{\mathsf{pk}}$
- Verify π_2 and π_3 w.r.t. $\{\mathsf{vpk}.\mathsf{ct}_i\}_{i=1}^{\lambda+Q}$.
- Verify π_4 w.r.t. $(\mathsf{vpk}.N_{\mathsf{cm}}, \mathsf{vpk}.g, \mathsf{vpk}.h)$.
- Run the main $\mathcal{P}$ body from $\mathsf{DVRange}_{\mathsf{Prot}}$ (Fig. 6).

Fig. 5. The key generation procedure for the $\mathsf{DVRange}_{\mathsf{Prot}}$ range proof of a preimage of ψ. It closely follows the structure of $\mathsf{DV}_{\mathsf{Prot}}.\mathsf{KeyGen}$ in Fig. 3, but additionally introduces commitment scheme and π_3, π_4.

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

$\mathcal{P}_1$:

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. [61]).
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 $\alpha = \psi(r)$.

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

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

$\underline{\mathcal{V} \rightarrow \mathcal{P}}$: 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:
   - $U = \mathsf{Enc}_{\mathsf{pk}}(r;0) \cdot C^{R-x}$, $V = \mathsf{Enc}_{\mathsf{pk}}(\sigma;0) \cdot C^{-t}$.
   - $U_i = \mathsf{Enc}_{\mathsf{pk}}(r_i;0) \cdot C^{x_i}$, $V_i = \mathsf{Enc}_{\mathsf{pk}}(\sigma_i;0) \cdot C^{t_i}$, for $i \in [1,3]$.
   - $U_4 = \mathsf{Enc}_{\mathsf{pk}}(\tau;0) \cdot C^{\sum_{i=1}^{3} x_i t_i - 4(R-x)t}$.

$\underline{\mathcal{P} \rightarrow \mathcal{V}}$: 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 $\mathcal{R} = \big\{(S_i, C); (w_1, w_2) : S_i = \mathsf{Enc}_{\mathsf{pk}}(w_1; 0) \cdot C^{w_2}\big\}$ with $|\mathcal{C}| = 2^\lambda$. This can be done in two extra rounds starting with $\mathcal{P}_2$, as in Fig. 2.

Fig. 6. $\mathsf{DVRange}_{\mathsf{Prot}}$: The designated-verifier range proof of a preimage of ψ. The key generation phase is presented in Fig. 5.

	VPK Gen	VPK Verify	Prove	Verify	Proof size	VPK size
$\mathsf{DV}_{\mathsf{Prot}}$ M	9391	10923	79	59	4.86 KB	742 KB
$\mathsf{DV}_{\mathsf{Prot}}$ T	2377	-	67	50	4.54 KB	160 KB
$\mathsf{DV}_{\mathsf{Prot}}$ M GGM	9391	10923	45	38	2.32 KB	742 KB
$\mathsf{DV}_{\mathsf{Prot}}$ T GGM	2377	-	38	33	2.19 KB	160 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ M	13667	15911	777	467	30.39 KB	844 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ T	3657	-	623	373	28.16 KB	189 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ M GGM	13667	15911	192	125	11.05 KB	844 KB
$\mathsf{DVRange}_{\mathsf{Prot}}$ T GGM	3657	-	167	105	10.41 KB	189 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.

$\mathcal{R}_{\mathsf{HomRange}}$ *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.*

Proof. Deferred to Appendix D.6. □

$\mathsf{DVRange}_{\mathsf{Prot}}$ 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. 6 that proves knowledge of the plaintext inside S_i.

6 Evaluation and Performance

We implemented[19] and benchmarked our protocols, primarily focusing on evaluating $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$. Recall $\mathsf{DV}_{\mathsf{Prot}}$ proves knowledge of the ciphertext message and $\mathsf{DVRange}_{\mathsf{Prot}}$ shows it is in range. Our results are summarised in Table 1. As a baseline we also implemented several flavours of the basic Σ-protocol (Table 2). For simplicity here we only present non-interactive variants. Our benchmarks are run for commonly used parameters, on an accessible hardware, and without any complex or hardware optimisations.

Setup and Instantiation Details. 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. 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 low-level optimisations are used. Note it is likely that the use of parallelism and optimisations would substantially improve the proving and verification time.

For $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$ we use Paillier-ElGamal encryption as the target homomorphism. Paillier-ElGamal encryption is additively homomorphic in both message and randomness. For the additively homomorphic encryption scheme on the verifier's side we use the "lite Cramer-Shoup"

[19] The implementation is available publicly on Github: `https://github.com/volhovm/rsa-zkps-impl`. Note it is *not production-ready* and is only intended for approximate evaluation purposes.

	Prove	Pre-Verify	Verify	Proof size
$\mathsf{Sigma}_{\mathsf{Prot}}$, $\lambda = 128$ reps	1385	0	2445	134.00 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$, 8 reps	87	4	155	8.38 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$, 7 reps	76	36	137	7.33 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$, 6 reps	65	339	117	6.28 KB
$\mathsf{Sigma}_{\mathsf{Prot}}$, 5 reps	55	6573	98	5.23 KB
$\mathsf{SigmaRange}_{\mathsf{Prot}}$ (with slack)	1383	0	2458	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.

variant of Paillier where ciphertexts are computed as $(1 + mN)h^r \mod N^2$ (see Section 4.6). The performance of lite Cramer-Shoup is similar to the original Paillier but it is extractable in the GGM. For each of 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). For trusted VPK we do not consider VPK verification time.

For the baseline $\mathsf{Sigma}_{\mathsf{Prot}}$ and $\mathsf{SigmaRange}_{\mathsf{Prot}}$ we also use Paillier-Elgamal as our target homomorphism. We evaluate $\mathsf{Sigma}_{\mathsf{Prot}}$ with naive $\lambda = 128$ reps, and also with varying $\log p_{\mathsf{max}} \in \{16, 19, 22, 26\}$, which implies different number of repetitions $\lambda / \log p_{\mathsf{max}}$; when $p_{\mathsf{max}} > 1$, this means the verifier has to perform the small primes check for numbers up to p_{max}. This illustrates the trade-off between pre-verifying the malicious prover's modulus N once, which allows more efficient interaction afterwards. 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 possible for all applications.

Performance Overview. Below we will mostly consider the GGM optimised variants of our protocols that assumes trusted setup because it gives us the best performance. However, the malicious GGM instantiation can also be competitive when amortization over VPK size and verification is taken into account. The evaluation indicates that our protocols are a strictly better choice for applications that can tolerate malicious VPK (e.g. ID-MPC such as RSA ceremonies), as they exhibit better verification time and communication size. The most noticeable performance improvements are verification time and communication space, however our proving time is also strictly less. However, the baseline protocols in Table 2 are publicly verifiable whereas our $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$ are designated verifier.

The main advantage of our $\mathsf{DV}_{\mathsf{Prot}}$ and $\mathsf{DVRange}_{\mathsf{Prot}}$ is that they are single-shot, requiring no repetitions. This benefits $\mathsf{DVRange}_{\mathsf{Prot}}$ more because the baseline $\mathsf{SigmaRange}_{\mathsf{Prot}}$ cannot avoid λ repetitions. Our verification time is strictly less than the baseline: 2-3.5× for $\mathsf{DV}_{\mathsf{Prot}}$, and 23× for $\mathsf{DVRange}_{\mathsf{Prot}}$. This is due to us evaluating the target homomorphism ψ less times for the final verification equation (exponents over N^2 are the most expensive operation).

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 $\mathsf{DVRange}_{\mathsf{Prot}}$. Smaller proofs amortize VPK size over time. The unavoidable bottleneck is the encrypted ciphertexts in the VPK, and their range proofs in the malicious VPK case.

Our proving time is about 1.5-1.7× smaller for $\mathsf{DV}_{\mathsf{Prot}}$, and about 8× smaller for $\mathsf{DVRange}_{\mathsf{Prot}}$. Proving time is one of the bottlenecks of our protocols, due to the necessity to evaluate homomorphic operations modulo verifier's N_{pk}^2. The N_{pk}^2 is bigger than the original language modulus due to the requirement that AHE's message space must fit all the homomorphic operations.

On Comparing DV and Public Coin. The comparisons between our DV protocols and the public-coin base-line protocols assume that two parties are interacting. In some scenarios, however, this is not the case: for example, whenever N parties all need to communicate pairwise about their moduli, this means creating 1 proof per user in the public coin setup, or N proofs per user in the DV model (since now every party must prove to every other party). In these cases our DV performance numbers for proof size, prover time, and verifier time should be multiplied by N.

Potential Improvements. Our target homomorphism was chosen to be Paillier-ElGamal.We could have instead targeted the "lite Cramer-Shoup" variant, which is more efficient and has smaller ciphertexts.

Finally, we note that it is possible to further save communication size by employing a well-known transformation where instead of the first round commitments one can send their hash. This would decrease the $\mathsf{DVRange}_{\mathsf{Prot}}$ proof size by 3.2 KB, which is about 30% in the GGM case. We did not implement this transformation.

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] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *EUROCRYPT 2016, Part II*. Ed. by M. Fischlin and J.-S. Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 327–357. DOI: 10.1007/978-3-662-49896-5_12.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] R. Cramer, R. Gennaro, and B. Schoenmakers. "A Secure and Optimally Efficient Multi-Authority Election Scheme". In: *EUROCRYPT'97*. Ed. by W. Fumy. Vol. 1233. LNCS. Springer, Heidelberg, May 1997, pp. 103–118. DOI: 10.1007/3-540-69053-0_9.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] I. Damgård and M. Jurik. "Client/Server Tradeoffs for Online Elections". In: *PKC 2002*. Ed. by D. Naccache and P. Paillier. Vol. 2274. LNCS. Springer, Heidelberg, Feb. 2002, pp. 125–140. DOI: 10.1007/3-540-45664-3_9.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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*. Cryptology ePrint Archive, Report 2011/494. https://eprint.iacr.org/2011/494. 2011.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] W. Quach, R. D. Rothblum, and D. Wichs. “Reusable Designated-Verifier NIZKs for all NP from CDH”. In: *EUROCRYPT 2019, Part II*. Ed. by Y. Ishai and V. Rijmen. Vol. 11477. LNCS. Springer, Heidelberg, May 2019, pp. 593–621. DOI: 10.1007/978-3-030-17656-3_21.

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

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

A Example Use Case: The HMRTN RSA Ceremony

To illustrate the complexity of the problem at hand, we comment on the protocol by Hazay et al. [49] that is constructing an RSA modulus jointly (the work was first presented in [48]). They provide a simulatable two-party protocol that constructs an RSA public key, and a related two-party threshold Paillier encryption scheme (that decrypts under the joint RSA key). They also generalise their result to the multiparty setting. Security in the multiparty setting is claimed to hold against active adversaries, tolerating $k-1$ out of k corrupted parties.

We claim that the multi-party construction in [49] requires the range proofs for Paillier ciphertexts to be subversion resistant. The suggestions cited in that work are unsuitable because they are not subversion resistant. Thus our zero-knowledge proofs are an essential extension to this setup ceremony. The issue arises where parties exchange Paillier ciphertexts created for their (potentially malicious) public keys.

To ellaborate, in [49] each party generates their own Paillier encryption key N_i which, if they are honest, has the form $N_i = p_i \times q_i$ for primes p_i and q_i. They prove in zero-knowledge that $\gcd(\phi(N_i), N_i) = 1$ in order to cover the scenario where N_i is maliciously chosen. However, they do not prove that N_i is actually a biprime. In Appendix C.1, Step 3, each party P_i sends a Paillier encryption $c_{\alpha_i,j}$ of α_i[20] to P_j under its own key N_i. Then, each party proves plaintext knowledge of α_i and that the value is bounded. Consider the scenario where α_i is not in fact bounded. Then in the next step, where P_j returns the ciphertext

$$\bar{c}_{i,j} = (c_{\alpha_i,j})^{\beta_j} \cdot \mathsf{Enc}_{N_i}(-s_{i,j}^{(j)} + Q \cdot r_{i,j})$$

then P_i could decrypt this value learn some partial information about β_j if, for example, we have that $\alpha_i > -s_{i,j}^{(j)} + Q \cdot r_{i,j}$. This goes against the claim that β_j is entirely hidden from the adversary and hence the proof for α_i being bounded is vital.

The suggested proofs that the α_i is bounded include decomposing the encryption into bits and proving their binary form using [32]; or using the range proofs of [15, 54, 38] whose solutions depend on a homomorphic integer commitment scheme such as [43, 36]. None of these solutions would suffice in this highly subverted setting where the only guarantee we have is that $\gcd(\phi(N_i), N_i) = 1$. Indeed, the solutions by [32] is given for ElGamal encryptions or for RSA encryptions that are generated under a modulus composed of safe primes. The solutions by [15, 54] require that N_i is a composite number whose factorisation is unknown. The [38] solution depends on the Strong RSA assumption which does not hold for subverted N_i. The [38] solution also points to [39] as providing range proofs for Paillier ciphertexts, but these all require that N_i has no small divisors.

B Background: Properties of the Paillier Cryptosystem with Subverted Moduli

In this section we revise the basic properties of Paillier cryptosystem [57] and the multiplicative group $\mathbb{Z}_{N^2}^*$. The properties listed here hold for any modulus N including if N is not biprime. Let $N = \prod p_i^{\alpha_i} \in \mathbb{N}$ be any natural number, where $\{p_i\}$ are all distinct prime numbers.

Lemma B.1. *The order of $\mathbb{Z}_{N^2}^*$ is equal to $\phi(N^2) = N\phi(N)$.*

[20] Here we diverge from the notation in [49] and use α_i in the place of their p_i to avoid overloading notation.

Proof. Given $\phi(N) = \prod p_i^{\alpha_i - 1}(p_i - 1)$, note that

$$\phi(N^2) = \prod p_i^{2\alpha_i - 1}(p_i - 1) = (\prod p_i^{\alpha_i} p_i^{\alpha_i - 1}(p_i - 1)) = N\phi(N)$$

□

Lemma B.2. *For any $N, i \in \mathbb{N}$, $(N+1)^i = 1 + iN \pmod{N^2}$.*

Proof. From the bionomial theorem, for any i we have that $(N+1)^i = \sum_{k=0}^{i} \binom{i}{k} N^k$. For any $i \geq 2$ see that $N^i \mod N^2 = 0$. Hence

$$(N+1)^i \mod N^2 = \binom{i}{0} N^0 + \binom{i}{1} N^1 \mod N^2 = 1 + iN \mod N^2.$$

□

Lemma B.3. *For any $N > 0$, the order of $(N+1)$ in $\mathbb{Z}_{N^2}^*$ is N i.e. N is the smallest integer such that $(N+1)^N = 1 \mod N^2$.*

Proof. By Lemma B.2 we have that $(1+N)^i \mod N^2 = 1 + iN \mod N^2$. For all $i < N$ we have that $1 < 1 + iN < N^2$. Thus N is the smallest integer such that $(1+N)^N \mod N^2 = 1$. □

Lemma B.4. *For all $G \in \mathbb{Z}_{N^2}^*$, the order of G is bounded by*

$$\operatorname{ord}(G) \leq \operatorname{lcm}(\{p_i^{2\alpha_i - 1}(p_i - 1)\}_i)$$

Proof. Given that $N^2 = \prod p_i^{2\alpha_i}$, we know that $\mathbb{Z}_{N^2}^* \cong \prod_i \mathbb{Z}_{p_i^{2\alpha_i}}^*$. Each group in the product has order $\phi(p_i^{2\alpha_i}) = p_i^{2\alpha_i - 1}(p_i - 1)$, and the maximum order of the element from the group product is equal to the LCM of product group orders. □

Definition B.1 (N-th residues). *A $z \in \mathbb{Z}_{N^2}^*$ is an N-th residue modulo N^2 if there is an $r \in \mathbb{Z}_{N^2}^*$ such that $z = r^N \pmod{N^2}$. The set of the N-th residues modulo N^2 is denoted by* $\operatorname{Res}(N)$.

Lemma B.5. $\operatorname{Res}(N)$ *forms a subgroup of $\mathbb{Z}_{N^2}^*$.*

Proof. We first show that $\operatorname{Res}(N)$ is a subgroup of $\mathbb{Z}_{N^2}^*$. Note that if $z \in \operatorname{Res}(N)$ then $z \in \mathbb{Z}_{N^2}^*$ and thus $\operatorname{Res}(N) \subset \mathbb{Z}_{N^2}^*$. We show that $\operatorname{Res}(N)$ is a group. There is an identity element because $1 = 1^N \mod N^2$. Associativity holds because $\mathbb{Z}_{N^2}^*$ is associative. We next show closure. If $a_1, a_2 \in \operatorname{Res}(N)$ then $a_1 = r_1^N \mod N^2$ and $a_2 = r_2 \mod N^2$ for some $r_1, r_2 \in \mathbb{Z}_{N^2}^*$. Hence $a_1 \cdot a_2 = (r_1 r_2)^N \mod N^2$ for $r_1 r_2 \in \mathbb{Z}_{N^2}^*$, implying that $a_1 \cdot a_2 \in \operatorname{Res}(N)$. Finally we see that if $a_1 \in \operatorname{Res}(N)$ then $a_1^{-1} = (r_1^{-1})^N \mod N^2$ and thus $a_1^{-1} \in \operatorname{Res}(N)$ and thus we have inverses. These are the four defining properties of a group and together give us that $\operatorname{Res}(N)$ is a subgroup of $\mathbb{Z}_{N^2}^*$. □

We define $\mathcal{B} = \{e \mid \operatorname{ord}(e) = \alpha N\}_{\alpha \in \mathbb{Z}}$. Now let the function $f_G : \mathbb{Z}_N \times \mathbb{Z}_N^* \to \mathbb{Z}_{N^2}^*$, $f_G(x, r) = G^x r^N \mod N^2$. f is a homomorphism between $\mathbb{Z}_N \times \mathbb{Z}_N^*$ and $\mathbb{Z}_{N^2}^*$ We use $N+1$ as a default base and say $f(x, r) = f_{N+1}(x, r)$.

Lemma B.6 (Injectivity of Paillier). *Let N be any composite number such that* $\gcd(\phi(N), N) = 1$, *$G \in \mathcal{B}$, and $x_1, x_2 \in \mathbb{Z}, r_1, r_2 \in \mathbb{Z}_{N^2}^*$ (note the extended domain). Then from $G^{x_1} r_1^N = G^{x_2} r_2^N$* $\mod N^2$ *it follows that $x_1 = x_2 \mod N$ (not just $\mod \operatorname{ord}(G)$), and $r_2 = r_1 G^{\frac{x_1 - x_2}{N}} \mod N^2$.*

Proof. Assume $G^{x_1}r_1^N = G^{x_2}r_2^N \mod N^2$, then $G^{x_2-x_1}(r_2/r_1)^N = 1$. After raising it to λ we get $G^{\lambda(x_2-x_1)} = 1 \mod N^2$ (Carmichael's theorem works for any N). From this we conclude that $\lambda(x_2 - x_1) = 0 \mod \alpha N$. Since $\gcd(\lambda, N) = 1$, necessarily $N \mid (x_2 - x_1)$ (since N doesn't divide the other term λ). Therefore $x_2 - x_1 = 0 \mod N$, say $x_2 - x_1 = kN$.

Returning back, we now see that $(r_2/r_1)^N = G^{-kN} \mod N^2$, then $r_2/r_1 = G^{-k}$. □

When $G = 1 + N$, the last condition means that $r_2 = r_1(1 + kN)$, so $r_1 - r_2 = 0 \mod N$.

Lemma B.7 (Bijectivity of Paillier). *Let N be any composite number such that $\gcd(\phi(N), N) = 1$, and $G \in \mathcal{B}$. Then $f_G : \mathbb{Z}_N \times \mathbb{Z}_N^* \to \mathbb{Z}_{N^2}^*$ is a bijection.*

Proof. Since the size of domain and range match, being equal to $N \times \phi(N)$, we only need to show injectivity. By Lemma B.6, if $f_G(x_1, r_1) = f_G(x_2, r_2)$ for $x_i \in \mathbb{Z}_N, r_i \in \mathbb{Z}_N^*$, we have $x_1 = x_2 \mod N$, so just $x_1 = x_2$. At the same time, $r_2 = r_1 G^0 \mod N^2$, so randomness is just equal too mod N, which concludes the injectivity proof. □

The following two lemmas points out a smaller condition of injectivity for $G = 1 + N$:

Lemma B.8. *If any composite N such that $\gcd(\phi(N), N) = 1$ it holds that $\{e \mid e^N = 1\} \cap \mathrm{Res}(N) = \{1\}$. In other words, $1 + iN \notin \mathrm{Res}(N)$ for $i \in [1, N - 1]$.*

Proof. The maximum order of any element in $\mathbb{Z}_{N^2}^*$ must divide $N\phi(N)$. Assuming that $e = \sqrt[n]{1}$, let $\mathrm{ord}(e) = k$. If there exists $g \in \mathbb{Z}_{N^2}^*$ such that $g^N = e$, then $g^{kN} = 1$. In this case $kN \mid \phi(N)N$, and since $\gcd(N, \phi(N)) = 1$, it means $k \mid \phi(N)$. But since any order of element e in subgroup of order N must have an order divisible by N, so it must be $k \mid N$. If $k \mid N$ and $k \mid \phi(N)$, it means $k = 1$, so 1 is the only N-th residue from the N-th roots of unity. □

Lemma B.9 (Injectivity from Weaker Assumptions). *Let $N \in \mathbb{Z}$ such that $\{e \mid e^N = 1\} \cap \mathrm{Res}(N) = \{1\}$, $G \in \mathcal{B}$, and $x_1, x_2 \in \mathbb{Z}, r_1, r_2 \in \mathbb{Z}_{N^2}^*$. Then $(N + 1)^{x_1} r_1^N = (N + 1)^{x_2} r_2^N \mod N^2$ implies $x_1 = x_2 \mod N$, and $r_2 = r_1 G^{\frac{x_1 - x_2}{N}} \mod N^2$.*

Proof. Consider the following:

$$\begin{aligned} (N+1)^{x_1} r_1^N &= (N+1)^{x_2} r_2^N && \pmod{N^2} \\ &\Rightarrow (N+1)^{x_1 - x_2} = (r_1^{-1} r_2)^N && \pmod{N^2} \\ &\Rightarrow (x_1 - x_2) = 0 && \pmod{N} \end{aligned}$$

The last implication is due to the fact that $(N + 1)^{x_1 - x_2}$ has a root of degree N $(r_1^{-1} r_2)$, therefore it must be that $x_1 - x_2 = 0 \mod N$ by the previous lemma. □

B.1 ElGamal-Paillier Cryptosystem

The ElGamal-Paillier [33, 16] is a variant of the original Paillier cryptosystem where the decrypter does not need to know the order of the group $\mathbb{Z}_N^*$. This makes it useful in threshold settings. It works as follows.

$\mathsf{KeyGen}(1^\lambda)$**:** samples two prime numbers p, q of the size λ and sets $N = p \cdot q$. Then samples $\mu \leftarrow\$ \mathbb{Z}_{N^2}^*$, $\tau \leftarrow\$ [1, \phi(N^2)/2]$ and sets $G = \mu^2$, $H = G^\tau$. Outputs $\mathsf{pk} = (N, G, H)$ and $\mathsf{sk} = \tau$.

$\mathsf{Enc}_{\mathsf{pk}}(m)$**:** samples $r \leftarrow\$ [0, N/2]$ and outputs $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2) = (G^r, (N + 1)^m H^r)$.

$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$**:** computes $c = \mathsf{ct}_2 \mathsf{ct}_1^{-\tau} \mod N^2$ and returns $m = \frac{c-1}{N}$.

Again the scheme is additively homomorphic and $\mathsf{IND\text{-}CPA}$-secure under DCRA.

C Efficient NIZKs for Key-Subverted Paillier Ciphertexts

In this section we present a zero-knowledge proof of knowledge for a plaintext inside a Paillier ciphertext (Appendix C.1). We then extend the protocol to encompass a range proof i.e. to show that the plaintext inside a Paillier ciphertext is within a given range (Appendix C.2). We also discuss how a batching method by Cramer et al. [30] can be applied to the range proofs such that we can prove many statements with considerably better efficiency than proving each statement individually.

The proofs in this section are publicly verifiable and the number of parallel repetitions they require to achieve negligible soundness error depends on the security parameter. Later, in Section 4 we show how, in the designated verifier setting, we can avoid this need for parallel repetitions. However, for the verifier to prove that it's public key has indeed been generated honestly, we will use our publicly verifiable range proof from Appendix C.2.

C.1 The Basic Proof of Plaintext Knowledge

Dealing with an arbitrary modulus turns the construction of zero-knowledge proofs for corresponding ciphertexts into a very challenging task, even for the simplest relation of just proving knowledge of the plaintext encrypted. Formally, we wish to construct a protocol for the relation:

$$\mathcal{R}_{\mathsf{Pai}} = \left\{(N, \mathsf{ct}); (m, r) : \mathsf{ct} = (N+1)^m r^N \pmod{N^2}\right\}$$

for any $N \in \mathbb{Z}$, $\mathsf{ct} \in \mathbb{Z}^*_{N^2}$. Since N is any arbitrary integer, the usual three-round Schnorr protocol may result in $\gcd(N, c - c') \neq 1$, thus an the inverse is only guaranteed to exist if $c - c' = 1$. The folklore solution to achieving soundness when the $\gcd(N, c - c')$ may be different to 1 is to restrict the challenge space to $\{0, 1\}$ and soundness error $1/2$. To amplify soundness the protocol is iterated $O(\lambda)$ times, which can be done in parallel. This leads to an $O(\lambda)$ (multiplicative) overhead.

Smaller overhead is achievable when N does not have divisors smaller than p_{max}, using techniques inspired by [45]. In this case $\gcd(N, c - c') = 1$ whenever $c - c' \leq p_{\mathsf{max}}$ and the challenge space can instead be $[1, p_{\mathsf{max}}]$ (the soundness error is then $1/p_{\mathsf{max}}$). We thus consider the alternative relation

$$\mathcal{R}_{\mathsf{Pai},p_{\mathsf{max}}} = \left\{(N, \mathsf{ct}); (m, r) : \begin{array}{c} \mathsf{ct} = (N+1)^m r^N \pmod{N^2} \wedge \\ \mathsf{DivVer}(N, p_{\mathsf{max}}) = 1 \end{array}\right\}$$

where $\mathsf{DivVer} : \mathbb{Z} \times \mathbb{Z} \to \{0, 1\}$ is a function defined as follows:

$$\mathsf{DivVer}(N, p_{\mathsf{max}}) = \begin{cases} 0, & \text{if } \exists p \in [2, p_{\mathsf{max}}] \mid p \text{ divides } N \\ 1, & \text{otherwise} \end{cases}$$

Essentially, $\mathcal{R}_{\mathsf{Pai},p_{\mathsf{max}}}$ is $\mathcal{R}_{\mathsf{Pai}}$ where N is restricted to integers with prime factors greater than p_{max}. The restriction on N of having factors greater than p_{max} can be publicly verified by DivVer in $O(p_{\mathsf{max}})$ time by plain division tests. Provided that p_{max} is polynomial in λ we have that DivVer is efficient. The protocol, $\mathsf{Sigma}_{\mathsf{Prot}}$ is presented in Fig. 7. The soundness of this modified protocol is $1/p_{\mathsf{max}}$, and, therefore, knowledge of a Paillier plaintext can be proven using only $\lambda/\log(p_{\mathsf{max}}) = O\left(\frac{\lambda}{\log \lambda}\right)$ parallel iterations.

Theorem C.1. $\mathsf{Sigma}_{\mathsf{Prot}}$ *is a correct, perfect honest-verifier zero knowledge and knowledge-sound, with soundness error* $1/p_{\mathsf{max}}$*, protocol for the relation* $\mathcal{R}_{\mathsf{Pai},p_{\mathsf{max}}}$.

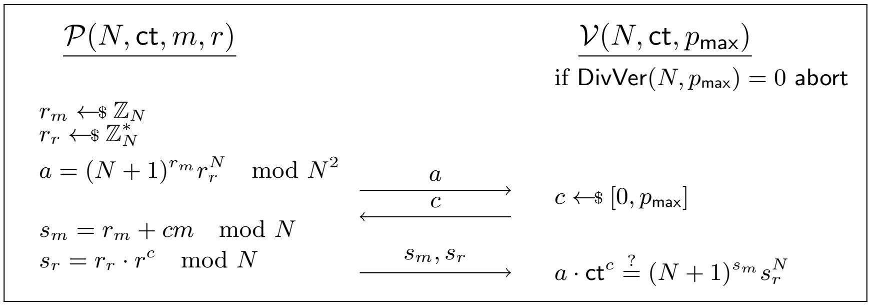
$\mathcal{P}(N, \mathsf{ct}, m, r)$
$\mathcal{V}(N, \mathsf{ct}, p_{\mathsf{max}})$
if $\mathsf{DivVer}(N, p_{\mathsf{max}}) = 0$ abort
$r_m \leftarrow\$ \mathbb{Z}_N$
$r_r \leftarrow\$ \mathbb{Z}_N^*$
$a = (N+1)^{r_m} r_r^N \mod N^2$
$\xrightarrow{a}$
$\xleftarrow{c}$
$c \leftarrow\$ [0, p_{\mathsf{max}}]$
$s_m = r_m + cm \mod N$
$s_r = r_r \cdot r^c \mod N$
$\xrightarrow{s_m, s_r}$
$a \cdot \mathsf{ct}^c \overset{?}{=} (N+1)^{s_m} s_r^N$

Fig. 7. $\mathsf{Sigma}_{\mathsf{Prot}}$: A zero-knowledge proof of knowledge of a plaintext for Paillier with soundness error $1/p_{\mathsf{max}}$. It can be similarly generalised to any hidden order co-domain homomorphism.

Proof. ***HV Zero-Knowledge.*** The simulator Sim on input $(N, \mathsf{ct}, p_{\mathsf{max}}, c)$, first checks the validity of the input. If $\mathsf{DivVer}(N, p_{\mathsf{max}}) = 0$ it aborts. If not it chooses at random $s_m^* \leftarrow\$ \mathbb{Z}_N, s_r^* \leftarrow\$ \mathbb{Z}_N^*$, sets $a^* = (N+1)^{s_m^*} s_r^{*N} \cdot \mathsf{ct}^{-c}$ and outputs a simulated transcript $(a^*, c, (s_m^*, s_r^*))$. Note that r_m, r_r and s_m^*, s_r^* are sampled uniformly at random in the simulated and real transcript resp. Also, $a^* = (N+1)^{s_m^* - m\cdot c}(s_r^* r^{-c})^N$ is distributed identically to a because $s_m^* - m \cdot c \overset{\mathrm{p}}{=} s_m$ and $s_r^* r^{-c} \overset{\mathrm{p}}{=} r_r$. Therefore, the distributions of $(a^*, c, (s_m^*, s_r^*))$ and $(a, c, (s_m, s_r))$ are identical.

Knowledge Soundness. For knowledge soundness consider an adversary $\mathcal{P}^*$ that plays the role of a malicious prover and has probability of success in convincing the verifier $\varepsilon > \frac{1}{p_{\mathsf{max}}} + \frac{1}{\mathsf{poly}(\lambda)}$. We construct an extractor Ext that has rewindable black-box access to $\mathcal{P}^*$ and can output a valid witness for $\mathcal{R}_{\mathsf{PaiRange}}$ corresponding to the statement (ct, R).

The extractor $\mathsf{Ext}(N, \mathsf{ct}, R)$ runs $\mathcal{P}^*$ (by choosing it's internal random coins and the challenges) until it gets a valid transcript (a, c, s_m, s_r). Then it continues running $\mathcal{P}^*$ until it gets a second valid transcript on the same first message and a different challenge (a, c', s'_m, s'_r). Assume wlog that $c > c'$. Then Ext checks whether $\gcd(c - c', N) = 1$ and aborts if not. If $\gcd(c - c', N) = 1$, then Ext computes the Bezout coefficients (γ, δ) such that $\gamma(c - c') + \delta N = 1$ and outputs[21]

$$m^* = \gamma(s_m - s'_m) \mod N \qquad r^* = \left(\frac{s_r}{s'_r}\right)^{\gamma} \mathsf{ct}^{\delta} \mod N$$

Note that inverting s'_r is always possible, since it is guaranteed (implicitly by $\mathcal{V}$) to be in $\mathbb{Z}_N^*$.

We now argue that Ext: does not abort, terminates with the correct output, and runs in polynomial time.

Observe that $0 < c - c' < p_{\mathsf{max}}$ and additionally from the relation description (that is checked before the protocol begins by DivVer) we know that N has no prime divisor smaller that p_{max}, which means that $\gcd(c - c', N) = 1$. Hence Ext does not abort.

Since both transcripts are valid it holds that $a \cdot \mathsf{ct}^c = (N+1)^{s_m} s_r^N$ and $a \cdot \mathsf{ct}^{c'} = (N+1)^{s'_m} (s'_r)^N$ and by dividing the two equations we get

$$\mathsf{ct}^{c - c'} = (N+1)^{s_m - s'_m} \left(\frac{s_r}{s'_r}\right)^N$$

[21] γ is merely an inverse of $(c - c') \mod N$.

Now see that

$$\mathsf{ct} = \mathsf{ct}^{\gamma(c-c')+\delta(N)} = \left[(N+1)^{s_m - s'_m}\left(\frac{s_r}{s'_r}\right)^N\right]^\gamma \mathsf{ct}^{\delta N} =$$
$$= (N+1)^{\gamma(s_m - s'_m)}\left[\left(\frac{s_r}{s'_r}\right)^\gamma \mathsf{ct}^\delta\right]^N = (N+1)^{m^*} r^{*N}$$

which justifies the output of the extractor.

Finally, the expected number of times the extractor runs the prover is $O(\frac{1}{\varepsilon - 1/p_{\mathsf{max}}})$ which is equal to $O(\mathsf{poly}(\lambda))$ [34]. After that, the process to output m^*, r^*, described above, takes $O(1)$-time. Overall, the extractor's expected running time is polynomial for a knowledge error of $\kappa = 1/p_{\mathsf{max}}$. □

C.2 Range Proof for Binary Challenges

The basic $\mathsf{Sigma_{Prot}}$ with binary challenges only needs a small modification to *additionally* prove that the message inside the ciphertext is in a certain range. Formally, the proof we construct is for:

$$\mathcal{R}^R_{\mathsf{PaiRange}} = \left\{(N, \mathsf{ct}); (m, r) : \begin{array}{c}\mathsf{ct} = (N+1)^m r^N \pmod{N^2} \wedge \\ m \in [\![R]\!]\end{array}\right\}$$

Range proofs such as [42, 26, 21, 15, 30, 27] are not applicable because they either assume trusted RSA setup, or a non-Paillier (discrete logarithm or Pedersen) relation. We present our range proof $\mathsf{SigmaRange_{Prot}}$ for Paillier in the subverted N setting. We cannot apply the p_{max} optimisation from the previous section in the range case due to the irregular nature of s_m distribution when the challenge is not binary.

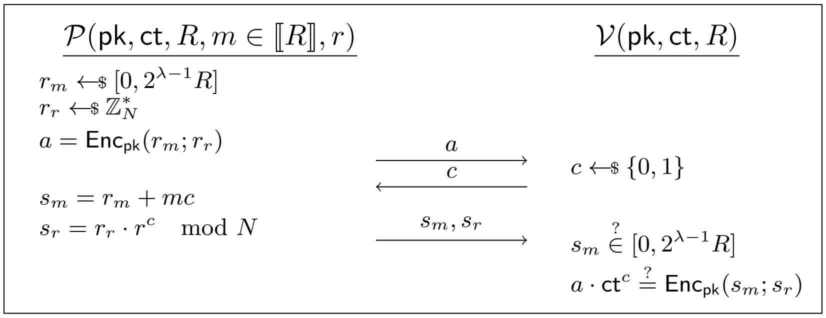
$\mathcal{P}(\mathsf{pk}, \mathsf{ct}, R, m \in [\![R]\!], r)$
$\mathcal{V}(\mathsf{pk}, \mathsf{ct}, R)$
$r_m \leftarrow\!\!\$\; [0, 2^{\lambda-1}R]$
$r_r \leftarrow\!\!\$\; \mathbb{Z}_N^*$
$a = \mathsf{Enc_{pk}}(r_m; r_r)$
a
c
$c \leftarrow\!\!\$\; \{0,1\}$
$s_m = r_m + mc$
$s_r = r_r \cdot r^c \mod N$
s_m, s_r
$s_m \overset{?}{\in} [0, 2^{\lambda-1}R]$
$a \cdot \mathsf{ct}^c \overset{?}{=} \mathsf{Enc_{pk}}(s_m; s_r)$

Fig. 8. $\mathsf{SigmaRange_{Prot}}$: The zero-knowledge proof for the range of a plaintext (proving $m \in [\![2^{\lambda+1}R]\!]$) encrypted with the Paillier cryptosystem. Can be generalized to other homomorphisms in the hidden order setting.

The following theorem states that when $m \in [\![R]\!]$, $\mathsf{SigmaRange_{Prot}}$ will prove $m \in [\![2^{\lambda+1}R]\!]$.

Theorem C.2. $\mathsf{SigmaRange_{Prot}}$*, when parameterized by* $R < N/2^{\lambda+1}$*, is a honest-verifier zero knowledge and knowledge-sound protocol, for* $\mathcal{R}^{[\![R\cdot 2^{\lambda+1}]\!]}_{\mathsf{PaiRange}}$*. It has soundness error* $1/2$*, and is statistically correct for messages* $m \in [\![R]\!]$*.*

Proof. This proof is similar to Theorem C.1 — except that in this protocol challenges are binary ($p_{\mathsf{max}} = 1$) and there is a range check.

Correctness. The correctness of the algebraic operations is straightforward, as before. For the correctness of the inequality check on $\mathcal{V}$'s side, note if $m \in [\![R]\!] = [-\frac{R}{2}, \frac{R}{2}]$, $r_m \in [0, 2^{\lambda-1}R]$ and $c \in [0,1]$ then

$$-\frac{R}{2} \leq s_m = r_m + cm \leq 2^\lambda \frac{R}{2} + \frac{R}{2}$$

However, note that $r_m \overset{\text{s}}{\approx} r_m + cm$ (since the distributions are only different at the interval $[-\frac{R}{2}, -1] \cup [2^\lambda \frac{R}{2} + 1, 2^\lambda \frac{R}{2} + \frac{R}{2}]$, which is negligibly small) Therefore $r_m + cm \notin [0, 2^{\lambda-1}R]$ occurs only with negligible probability and the check passes with overwhelming probability.

HV Zero-Knowledge. The simulator for the protocol works similarly to the one in the proof of Theorem C.1: the random response $s_m \in \mathcal{U}_{[0,2^{\lambda-1}R]}, s_r \in \mathcal{U}_{[0,N]}$ and challenge $c \in \{0,1\}$ are chosen, and then the commitment is computed as $a = \mathsf{Enc}_{\mathsf{pk}}(s_m, s_r)(\mathsf{ct}^c)^{-1}$. The transcript is then $(a, c, (s_m, s_r))$.

As shown previously, picking $s_m \in \mathcal{U}_{[0,2^{\lambda-1}R]}$ is statistically indistinguishable from picking it from the "honest" distribution as defined by $s_m = r_m + cm$. The challenge is distributed exactly the same in both worlds. As for a, the honest distribution is $G^{r_m} r_r^N$, and simulated distribution is $G^{s_m - c \cdot m}(s_r r^{-c})^N$ — since $r_m \overset{\text{p}}{=} s_m$ and $r_r \overset{\text{p}}{=} s_r$ (as distributions), and moreover $s_m - c \cdot m \overset{\text{s}}{\approx} s_m$, and $\forall r \neq 0.\ s_r \overset{\text{p}}{=} s_r r^{-1}$, we obtain that arguments to Enc are statistically indistinguishable, so $a \overset{\text{s}}{\approx} a^*$. Therefore, the two transcripts are also statistically indistinguishable.

Knowledge Soundness. Let Ext be an extractor, similar to the one in the proof of Theorem C.1. Ext gets two transcripts $(a, c, s_m, s_r), (a, c', s'_m, s'_r)$ and outputs:

$$m^* = s_m - s'_m \mod N \qquad r^* = s_r * (s'_r)^{-1}$$

By extraction $c' \neq c$, so wlog assume $c = 1, c' = 0$. Both transcripts are valid: $a \cdot \mathsf{ct}^1 = \mathsf{Enc}_{\mathsf{pk}}(s_m; s_r)$ and $a \cdot \mathsf{ct}^0 = \mathsf{Enc}_{\mathsf{pk}}(s'_m; s'_r)$. This implies

$$\mathsf{ct} = \mathsf{Enc}(s_m - s'_m \mod N; s_r * (s'_r)^{-1}) = \mathsf{Enc}_{\mathsf{pk}}(m^*; r^*)$$

where the modulo N reduction occurs because $(N+1)$ in $\mathsf{Enc}(m, r) := (N+1)^m r^N$ still has order N even when N is subverted. This attests to correctness of Ext as before.

Now we are left with proving that m^* is in the correct range. Because of the range check and validity of the transcripts, it holds that $0 \leq s_m, s'_m \leq 2^{\lambda-1}R$ which directly gives us:

$$-2^\lambda R < s_m - s'_m = m^* \leq 2^\lambda R \pmod N$$

This shows that extracted message is in the specified range $[\![2^{\lambda+1}R]\!]$. The expected extractor's running time is still $O(\frac{1}{\varepsilon - 1/2})$, as in Theorem C.1. □

;

Unlike in Appendix C.1, $\mathsf{SigmaRange}_{\mathsf{Prot}}$ cannot be trivially extended to support a bigger challenge space. Assuming that $\mathcal{C} > 2$, we have $m^* = \gamma(s_m - s'_m)$, where $\gamma = (c - c')^{-1} \mod N$, with $c - c' \in [-|\mathcal{C}|, |\mathcal{C}|]$. In this case, even though we know the bound on $s_m - s'_m$, deriving any non-trivial bound on m^* difficult. This is because γ, on average, is not restricted to any subinterval of $\mathbb{Z}_N$ (with binary challenges $\gamma = 1^{-1} = 1$). This is also why the optimization with p_{max} from Appendix C.1 cannot be applied.

C.3 Range Proof on Non-Injective Homomorphisms

One caveat with $\mathcal{R}_{\mathsf{PaiRange}}^{\llbracket R \rrbracket}$ occurs when N is subverted such that Paillier is non-injective. If N is not injective then we can extract some preimage which is in the range but there could still be another preimage outside of the range. Non-injective Paillier is degenerate thus it is not obvious how this style of subversion would lead to any concrete attacks. However an easy way to guarantee injectivity is to have $\mathcal{P}$ show that $\gcd(N, \phi(N)) = 1$. This proof [45] is efficient, and we use it in Section 4.3 when proving the range of ciphertexts in a KeyGen. Note that it still a challenge to prove the range even when $\gcd(N, \phi(N)) = 1$ is already proven; it is a relatively weak well-formedness property of N.

D Deferred Security Proofs

D.1 Proof of Lemma 3.1 (Generalized Extraction Lemma)

Proof. Let $\mathcal{T} = \left\{(a, c^{(i)}, s^{(i)})\right\}_{i=1}^{M}$ be a collection of M succesful transcripts with $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$. We will construct a polynomial-time extractor Ext that on input $\mathcal{T}$ outputs a valid witness with respect to $\mathcal{R}_{\mathsf{Hom}}$ and Y.

The extractor Ext first sets $(\gamma_2, \gamma_3, \ldots, \gamma_M) \in \mathbb{Z}$ to be the Bezout coefficients such that

$$\gamma_2(c^{(2)} - c^{(1)}) + \gamma_3(c^{(3)} - c^{(1)}) + \ldots + \gamma_M(c^{(M)} - c^{(1)}) = 1 \tag{2}$$

which exist and can be found in polynomial time because $\gcd(c^{(2)} - c^{(1)}, \ldots, c^{(M)} - c^{(1)}) = 1$. It then sets $\gamma_1 = -\gamma_2 - \gamma_3 - \ldots - \gamma_M$ and returns

$$w = \sum_{i=1}^{M} \gamma_i s^{(i)}$$

Note the size of γ_i's cannot be larger than the biggest $c^{(i)}$, which guarantees that w can also be computed in polynomial time.

To see that Ext succeeds observe that $aY^{c^{(i)}} = \psi(s^{(i)})$ for each $i \in \{1, \ldots, M\}$ implies that

$$a^{\gamma_1+\gamma_2+\ldots+\gamma_M} Y^{\gamma_1 c^{(1)}+\gamma_2 c^{(2)}+\ldots \gamma_M c^{(M)}} = \psi(s^{(1)})^{\gamma_1} \psi(s^{(2)})^{\gamma_2} \ldots \psi(s^{(M)})^{\gamma_M}$$

Now by design $\gamma_1 + \ldots + \gamma_M = 0$ and by rearranging (2) we see that

$$\gamma_2 c^{(2)} + \gamma_3 c^{(3)} + \ldots + \gamma_M c^{(M)} = 1 + (\gamma_2 + \ldots + \gamma_M) c^{(1)}$$

Hence we have

$$Y = a^0 Y^{-(\gamma_2+\ldots+\gamma_M)c^{(1)}+1+(\gamma_2+\ldots+\gamma_M)c^{(1)}} = \psi(\sum_{i=1}^{M} \gamma_i s^{(i)})$$

and w as constructed above is a valid witness. □

D.2 Proof of Theorem 4.2 ($\mathsf{DV}_{\mathsf{Prot}}$ HVZK)

Proof. Consider the following simulator for our protocol:

```
Sim(vpk, vsk, $\psi$, $Y$)

Sample $b \leftarrow\$ \{0,1\}^\lambda$. Extract challenges $c_1 \dots c_\lambda$ from vpk (decrypting it with vsk), set $c = \sum_{i=1}^{\lambda} c_i b_i$
and $C = \prod_{i=1}^{\lambda} \mathsf{ct}_i^{b_i}$. Sample $s \leftarrow\$ [\![\lambda 2^{2\lambda}|\mathcal{D}|]\!]$. Compute

    $a := \psi(s) Y^{-c}$

Encrypt $S = \mathsf{Enc}_{\mathsf{pk}}(s, r_s)$ as $\mathsf{Enc}_{\mathsf{pk}}(s; 0) \cdot \mathsf{Enc}_{\mathsf{pk}}(0; \sum_{i=1}^{\lambda} b_i r_i)$, where $r_i$ is the randomness used for
$\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(c_i; r_i)$. Sample $d \leftarrow\$ [\![2^\lambda]\!]$, $u_1, u_2 \leftarrow\$ \mathcal{M}$. Finally, set

    $T := C^{u_1} \cdot \mathsf{Enc}_{\mathsf{pk}}(u_2; 0) S^{-d}$

Return $\mathsf{trans}^* := (a, b, S, T, d, u_1, u_2)$
```

The simulated transcripts are valid because T and a are computed backwards *from* the verifier's equations, similarly to how it is done in Fig. 7.

We must now argue that simulated transcripts trans^* are indistinguishable from honest transcripts trans. First see that b, d are uniformly sampled in both transcripts and thus are indistinguishable. We must argue that a, S, T, u_1, u_2 are indistinguishable. Let us first start with a. In the real world it is $\psi(D_s)$, in the simulation it is $\psi(D_s) \cdot Y^{-D_{\mathcal{C}}}$. Because ψ is homomorphic, $\psi(D_s) \cdot Y^{-D_{\mathcal{C}}} = \psi(D_s - D_{\mathcal{C}} \cdot w)$. By Lemma F.1 we have that $D_s = [1, \lambda 2^{2\lambda}|\mathcal{D}|]$ is big enough to blind $D_{\mathcal{C}} \cdot w$. Indeed $D_c \in \lambda 2^\lambda$, since we sum λ plaintexts upper-bounded with 2^λ each, and therefore $w \cdot D_c(b) \in [0, |\mathcal{D}|\lambda 2^\lambda]$. This shows $a \overset{s}{\approx} a^*$.

Next, consider S. Here both the simulator's and user's randomness are the same ($r_s = \sum_{i=1}^{\lambda} b_i r_i$). Thus the only place where the distributions differ is with the plaintext s. . If $\psi()$ is injective then s is the unique plaintext satisfying the deterministic verification equation $aY^c = \psi(s)$. Thus we have that $S \overset{s}{\approx} S^*$.

The tuples $(u_1^*, u_2^*), (u_1, u_2)$ are both sampled uniformly from $\mathcal{M} \times \mathcal{M}$ correspondingly and thus are indistinguishable.

The component T, which is also deterministically defined by the $\mathcal{V}$'s equation $TS^d = C^{u_1} \cdot \mathsf{Enc}_{\mathsf{pk}}(u_2; 0)$ by all the previous elements, therefore $T \overset{s}{\approx} T^*$. This completes the proof. □

D.3 Proof of Theorem 4.3 (Malicious $\mathsf{DV}_{\mathsf{Prot}}$ HVZK)

Proof. The proof of zero-knowledge uses the same simulation strategy as in Theorem 4.2. The key differences are the following:

- As a very first step, the $\mathcal{P}$ and Sim checks the validity of proof for vpk, and abort if the proof does not verify. If $c_i \notin 2^{3\lambda + \log\lambda - 1}$ then $\mathcal{V}$ breaks the statistical soundness of the range proof.
- Sim chooses s from a bigger interval, equal to the new honest prover's r_1: $s \leftarrow\$ [\![2^{4\lambda + 2\log\lambda - 1}|\mathcal{D}|]\!]$. This value is equal as the bound of VPK range proof ($2^{3\lambda + \log\lambda - 1}$), multiplied by λ because we sum at most λ ciphertexts, multiplied by $|\mathcal{D}|$ (at this point this is the upper bound of cw), and finally multiplied by the 2^λ factor for proper statistical blinding.

The rest of the proof of the indistinguishability of simulated transcripts proceeds exactly the same as in Theorem 4.2. □

D.4 Proof of Theorem 4.4 (Reusable $\mathsf{DV}_{\mathsf{Prot}}$ Security)

Proof. To prove that knowledge soundness holds we only need to show that the adversary $\mathcal{B}$ of Theorem 4.1 can simulate successfully $Q = \mathsf{poly}(\lambda)$ verification oracle queries of $\mathcal{P}^*$. The rest of the proof remains identical.

We first change the knowledge soundness game to a game Game_1 such that the encryptions $\mathsf{ct}_{\lambda+1}, \ldots, \mathsf{ct}_{\lambda+Q}$ contain 0 but the extractor and the verification oracle behave identically to the case that the encryptions contain $c_{\lambda+1}, \ldots, c_{\lambda+Q}$. By the IND-CPA of the encryption scheme the probability that the extractor succeeds is negligibly close in both games.

The reduction $\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}}(\mathsf{pk})$ is sampling $c_1, \ldots, c_\lambda, z_1, \ldots, z_\lambda \xleftarrow{\$} [\![2^\lambda]\!]$ and similarly $e_\lambda, \ldots, e_{\lambda+Q} \xleftarrow{\$} [\![\lambda 2^{2\lambda}]\!]$ and is sending $\{c_i\}_{i\in[\lambda]}, \{z_i\}_{i\in[\lambda]}$ to the encryption oracle to get $\mathsf{ct}_1, \ldots, \mathsf{ct}_\lambda$. The rest of the ciphertexts are computed normally $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(0)$ for each $i \in [\lambda, \lambda+Q]$. We denote:

$$\mathbf{c} = (c_1, \ldots, c_\lambda), \quad \mathbf{z} = (z_1, \ldots, z_\lambda), \quad \mathbf{e} = (e_\lambda, \ldots, e_{\lambda+Q})$$

We need simulate the verifier in Q verification oracle queries of $\mathcal{P}^*$, without knowing vsk.

Assume a query $\mathcal{O}^{\mathsf{Verify}}(\mathsf{vsk}, (\psi, Y, \kappa), \pi)$, where $1 < \kappa < Q$ is unqueried and where $\pi = (a, b^{(1)}, S^{(1)}, T^{(1)}, d^{(1,1)}, u_1^{(1,1)}, u_2^{(1,1)}, u_3^{(1,1)})$. First $\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}}(\mathsf{pk})$ uses the extractor of the sub-protocol $(T^{(1)}, d^{1,1)}, u_1^{(1,1)}, u_2^{(1,1)}, u_3^{(1,1)})$ (for the input $S^{(1)} = C^w \cdot \mathsf{Enc}_{\mathsf{pk}}(r_1; r_2)$) in order to extract $s_1^{(1)}$ and $s_2^{(1)}$ such that $S^{(1)} = \mathsf{Enc}_{\mathsf{pk}}(s_2^{(1)} + x^{(1)} s_1^{(1)})$, where $x^{(1)}$ is either $x_c = e_\kappa + \mathbf{b^{(1)}} \cdot \mathbf{c}^\top$ or $x_z = e_\kappa + \mathbf{b^{(1)}} \cdot \mathbf{z}^\top$. In order to check if the proof verifies it needs to check whether $aY^{x^{(1)}} = \psi(s_2^{(1)} + x^{(1)} s_1^{(1)})$ holds. Recall that if $Y \neq \psi(s_1^{(1)})$ or $a \neq \psi(s_2^{(1)})$ then $\mathcal{B}$ can guess x^1 except with negligible probaility. Assume that $Y = \psi(s_1^{(1)})$ and $a = \psi(s_2^{(1)})$ then the only case where $aY^{x^{(1)}} \neq \psi(s_2^{(1)} + x^{(1)} s_1^{(1)})$ is if $s_2^{(1)} + x^{(1)} s_1^{(1)} > N$ over the integers, so that an overflow happens in $\mathbb{Z}_N$ during the encryption.

Here is how $\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}}(\mathsf{pk})$ is answering to the verification oracle queries:

1. If both $aY^{x_c} = \psi(s_2^{(1)} + x_c s_1^{(1)})$ and $aY^{x_z} = \psi(s_2^{(1)} + x_z s_1^{(1)})$ hold then output 1.
2. If neither $aY^{x_c} = \psi(s_2^{(1)} + x_c s_1^{(1)})$ nor $aY^{x_z} = \psi(s_2^{(1)} + x_z s_1^{(1)})$ hold then output 0.
3. If $aY^{x_c} \neq \psi(s_2^{(1)} + x_c s_1^{(1)})$ and $aY^{x_z} = \psi(s_2^{(1)} + x_z s_1^{(1)})$, choose a bit at random and output it.
4. If $aY^{x_c} = \psi(s_2^{(1)} + x_c s_1^{(1)})$ and $aY^{x_z} \neq \psi(s_2^{(1)} + x_z s_1^{(1)})$, choose a bit at random and output it.

The first two cases simulate the verification correctly, while the last two only with probability 1/2. We show that Cases 3 and 4 occur only with negligible probability.

Assume that Case 3 happens amd $\mathcal{B}$ does not terminate. This means that $s_2^{(1)} + x_c s_1^{(1)} > N$ but $s_2^{(1)} + x_z s_1^{(1)} < N$. Denote $T^{(1)} = \frac{N - s_2^{(1)}}{s_1^{(1)}}$ and see that

$$x_c > T^{(1)} \Rightarrow \mathbf{b}^{(1)} \mathbf{c}^\top + e_\kappa > T^{(1)}$$
$$x_z < T^{(1)} \Rightarrow \mathbf{b}^{(1)} \mathbf{z}^\top + e_\kappa < T^{(1)}$$

However, since by construction any $e_\kappa >> 2^\lambda \sum_{i=1}^{\lambda} c_i$ (and similarly for z_i), we get that $T^{(1)}$ is statistically close to e_κ. Where the adversary has no information about e_κ the above happens only with negligible probability. The same reasoning goes for Case 4.

Hence if $\mathcal{B}$ does not terminate then $\mathcal{P}^*$ only makes queries within Case 1 and 2 with overwhelming probability. But then $\mathcal{B}$'s simulation of the verification oracle is sound . By the same argument as Theorem 4.1 $\mathcal{B}$ will compute a correct IND-CPA response with overwhelming probability. □

D.5 Proof of Theorem 4.5 (GGM $\mathsf{DV}_{\mathsf{Prot}}$ Knowledge Soundness)

Proof. Suppose that $(\mathsf{vpk}, \mathsf{vsk}, \tau) \overset{\$}{\leftarrow} \mathsf{KeyGen}(1^\lambda)$, where $\tau = \{c_1, \dots, c_\lambda\}$ 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 white-box access to the prover $\mathcal{P}^*$, such that whenever $\mathcal{P}^*$ outputs verifying $(Y; (a, S))$ then $\mathsf{Ext}^{\mathcal{P}^*}(\tau, \mathsf{vpk}, \psi, Y)$ 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 3.1, 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 of its choice) until it obtains M accepting transcripts, for the same first message a. That is:

$$\mathcal{T} = \left\{ \left(a, b^{(j)}, S^{(j)} \right) \right\}_{j \in [M]}$$

and outputs $\mathcal{T}$. For Ext_1 we use the generic M-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 until it obtains $\left(a, b^{(1)}, S^{(1)}\right)$. 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 uses the generic representation of $S^{(1)}$ i.e. $(s_{1,g}, s_{1,1}, \dots, s_{1,\lambda})$ such that

$$S^{(1)} = (1+N)^{s_{1,g}} h^0 \prod_{i=1}^{\lambda} \mathsf{ct}^{s_{1,i}}$$

and sets

$$s^{(1)} = s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i} c_i$$

From here Ext_1 can verify $aY^{c^{(1)}} = \psi(s^{(1)})$ to confirm if the transcript is accepting or not. It continues in a similar manner until it obtains M accepting transcripts $\mathcal{T}$.

Now, the extractor Ext behaves as follows. It runs $\mathcal{T} \leftarrow \mathsf{Ext}_1^{\mathcal{P}^*}(\tau, \mathsf{vpk}, \psi, Y)$ and computes $c^{(j)} = \sum_{i=1}^{\lambda} c_i b_i^{(j)}$. If $\gcd(c^{(2)} - c^{(1)}, \dots, c^{(\lambda)} - c^{(1)}) \neq 1$ it aborts. Else it computes $s^{(j)} = \mathsf{Dec}_{\mathsf{sk}}(S^{(j)})$ for each $j \in [M]$ and runs $w \leftarrow \mathsf{Ext}_0(\psi, Y; (a, c^{(1)}, s^{(1)}), \dots, (a, c^{(M)}, s^{(M)}))$ 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 3.1 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 $\gcd(c^{(2)} - c^{(1)}, \dots, c^{(M)} - c^{(1)}) \neq 1$ with non-negligible probability. In order to show this, we design an adversary $\mathcal{B}$ against $\mathsf{IND\text{-}CPA}$ that, using Ext, wins the $\mathsf{IND\text{-}CPA}$ 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 $i \in [\lambda]$;
$\mathsf{vpk} \leftarrow (\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\lambda)$
$\mathsf{coin} \xleftarrow{\$} [1, 2^\lambda]$;
while $j < M$ :    $\mathsf{trans}_{j,1} \leftarrow \mathcal{P}^*(\mathsf{vpk}, \psi, Y; \mathsf{coin})$
    if $aY^{c^{(j)}} = \psi(s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i}c_i)$ and $aY^{z^{(j)}} \neq \psi(s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i}z_i)$ return 0
    if $aY^{c^{(j)}} \neq \psi(s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i}c_i)$ and $aY^{z^{(j)}} = \psi(s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i}z_i)$ return 1
    if $aY^{c^{(j)}} = \psi(s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i}c_i)$ and $aY^{z^{(j)}} = \psi(s_{1,g} + \sum_{i=1}^{\lambda} s_{1,i}z_i)$ $j \leftarrow j + 1$
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
```

The argument that $\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}}$ succeeds against IND-CPA is identical to the proof in Theorem 4.1. □

D.6 Proof of Theorem 5.1 ($\mathsf{DVRange}_{\mathsf{Prot}}$ Security)

Proof. Correctness. It holds provided that the message space of AHE is larger than the maximum of $u, v, \{u_i\}_{i\in[1,4]}, \{v_i\}_{i\in[1,3]}$, so that the encryption is correct. Since the maximum is u_4, encrypting the value statistically close to the distribution of $\tau \leftarrow\!\!\$\ [\![2^{6\lambda+2\log\lambda+4}\frac{N_{\mathsf{cm}}}{2}R]\!]$, and since the $|\mathcal{M}|$ is chosen to match exactly this value, the encryption scheme is correct for $u, v, \{u_i\}_{i\in[1,4]}, \{v_i\}_{i\in[1,3]}$. The correctness of the algebraic operations can be confirmed by inspection.

Honest-Verifier Zero-Knowledge. First we need to show that the integer commitment scheme preserves hiding under maliciously generated N_{cm}, as long as $g = h^f$ for any integer f. In fact, for a uniformly random $r \leftarrow\!\!\$\ [\![2^\lambda \frac{N}{2}]\!]$ we get that $g^m h^r = g^{mf+r}$. The order of g is at most $\frac{\phi(N)}{2} < \frac{N}{2}$. Thus we see that r is at least 2^λ larger than $mf \mod \operatorname{ord}(g)$, which gives us that $mf + r \mod \operatorname{ord}(g)$ is statistically close to uniform. So any two commitments $g^{m_0}h^{r_0}$ and $g^{m_1}h^{r_1}$ are statistically close.

The simulator Sim is verifying the proofs $\pi_1, \pi_2, \pi_3, \pi_4$. With overwhelming probability this implies that the vpk is correctly formed. Then HVZK comes in a similar way to Theorem 4.3 and under the fact that the integer commitment is statistically hiding.

Knowledge-Soundness. Consider an adversary $\mathcal{P}^*$ that has probability of success in convincing the verifier ϵ, for a non-negligible ϵ. We construct an extractor Ext that has rewindable black-box access to $\mathcal{P}^*$ and can output a valid witness for $\mathcal{R}_{\mathsf{HomRange}}$ corresponding to the statement $(\mathsf{vpk}, \psi, Y, R)$.

Ext uses Ext' the extractor of Theorem 4.1, as a black-box. Ext' can simulate up to Q verification oracle queries of $\mathcal{P}^*$ correctly with overwhelming probability. Then Ext', running in expected polynomial time, outputs integers $u^*, v^*, \{u_i^*\}_{i\in[1,4]}, \{v_i^*\}_{i\in[1,3]} \in \mathbb{Z}$ such that

$$\begin{aligned}
Y^{-1}\psi(R) &= \psi(u^*) \\
\mathsf{cm}^{-1}g^R &= g^{u^*}h^{v^*} \\
\mathsf{cm}_i &= g^{u_i^*}h^{v_i^*}, \quad i \in [1,3] \\
\prod_{i\in[1,3]} \mathsf{cm}_i^{u_i^*} &= h^{u_4^*}g \cdot \mathsf{cm}^{4u^*}
\end{aligned}$$

assuming IND-CPA security of AHE holds. Combining the last three we get

$$g^{\sum_{i\in[1,3]}(u_i^*)^2-4(u^*-R)u^*-1}h^{\sum_{i\in[1,3]}-v_i^*u_i^*-4u^*v^*-u_4^*}=1$$

which under the factoring assumption over $\mathbb{Z}_N$ gives us that $\sum_{i\in[1,3]}(u_i^*)^2-4(u^*-R)u^*-1=0$ or $4(u^*-R)u^*+1=\sum_{i\in[1,3]}(u_i^*)^2$ over the integers. Under the three-square theorem we get that $u^*\in[0,R]$.

Combining the last with $Y=\psi(R-u^*)$, the extractor outputs $u=R-u^*$ which is in the range $u^*\in[0,R]$. □

E Amortizing the $\mathsf{DV}_{\mathsf{Prot}}$ VPK Generation

In this section we argue that the amortization result from [30] applies to our smaller range proof ($\mathsf{SigmaRange}_{\mathsf{Prot}}$) from Appendix C. In this section we refer to the newer version of this paper [31].

Our protocol, however similar to [31, Prototocol 1] (including modifications from [31, Sec. 6]), has a number of differences which makes it impossible to *directly* apply the results of that work to our case. Namely:

1. Section 6 of [31] covers the case of groups of unknown order, but only considers the one-argument exponentiation homomorphism $x\to g^x$, while we work with the two-argument Paillier. This is a crucial difference, since in the Paillier case the message space is public (equal to $\mathbb{Z}_N$), while in the DL case it is hidden (as $|g|$ is), so computations are done over $\mathbb{Z}$.
2. Lemmas 7 to 9 of their work explain how to deal with the case of commitment values (i.e. r_r, r_m) sampled from a big uniform subspace over integers. For the range proof in DL case they suggest using $[0, NRn]$ for n being number of iterations. In the case of Paillier it does not make sense to sample r_m bigger than the message space $\mathbb{Z}_N$, so our r_m is only $\lambda+\log(Rn)$ bits big (where R is a message range). This affects the HVZK proof.
3. The previous difference is also related the size of the range proof, which is in our case is much smaller.
4. In our case, unlike in [31], the prover knows the factorization of N, so we need to pay close attention to the extraction strategy (which, luckily, is the same as in [31]).

We use the same notation as [31]. First, all vector products, e.g. $\boldsymbol{a}\cdot\boldsymbol{b}$ are component-wise, while matrix-vector multiplication $E\boldsymbol{m}$ is as usual. Second, when E is an $m\times n$ matrix, and $\boldsymbol{x}$ is a vector of size n, $\boldsymbol{x}^E$ is defined as $(\boldsymbol{x}^E)_i=\prod_{j=1}^{n}\boldsymbol{x}_j^{E_{i,j}}$.

Let n be the number of repetitions we want the range protocol to be scaled up to (think $n=\lambda$ *at least* since our base protocol has binary challenges). Let $w:\mathcal{C}\to\mathbb{Z}^{m\times n}$, where $m=2n+1$, be a function mapping challenges from $\mathcal{C}=[0,2^n-1]$ to $m\times n$ matrices as follows:

$$w(c)=\begin{pmatrix} c_1 & & 0 \\ \vdots & \ddots & \\ c_n & & c_1 \\ & \ddots & \vdots \\ 0 & & c_n \end{pmatrix}$$

where c_i is a i-th bit of c.

$$\begin{array}{lcl}
\underline{\mathcal{P}(\mathsf{pk}, R, \boldsymbol{m} \in [\![R]\!]^n, \boldsymbol{r})} & & \underline{\mathcal{V}(\mathsf{pk}, \mathbf{ct}, R)} \\
\boldsymbol{r}_m \leftarrow\!\!\$\; [0, 2^l]^m & & \\
\boldsymbol{r}_r \leftarrow\!\!\$\; (\mathbb{Z}_N^*)^m & & \\
\boldsymbol{a} = \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{r}_m; \boldsymbol{r}_r) & \xrightarrow{\quad\boldsymbol{a}\quad} & \\
& \xleftarrow{\quad c \quad} & c \leftarrow\!\!\$\; [0, 2^n - 1] \\
\boldsymbol{s}_m = \boldsymbol{r}_m + E\boldsymbol{m} & & \\
\boldsymbol{s}_r = \boldsymbol{r}^E \cdot \boldsymbol{r}_r \mod N & \xrightarrow{\quad\boldsymbol{s}_m, \boldsymbol{s}_r\quad} & \forall i.\, (\boldsymbol{s}_m)_i \overset{?}{\in} [0, 2^l] \\
& & \boldsymbol{a} \cdot \mathbf{ct}^E \overset{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{s}_m; \boldsymbol{s}_r)
\end{array}$$

Fig. 9. $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$: Amortized variant of $\mathsf{SigmaRange}_{\mathsf{Prot}}$ which proves knowledge of Paillier preimage and its range for n instances, where $E = w(c)$, and $l = \lambda + \log n + \log R - 1$.

The presented $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$ is a variant of [31][Protocol 1 and Lemma 10], adapted to the Paillier case, which is proving the preimage knowledge and range for n instances simultaneously, with error probability $1/2^n$, and slack $2^{\lambda-1+n}n$.

The following result is a (close) adaptation of [31, Lemmas 10 and 11] to our setup.

Theorem E.1. $\mathsf{SigmaRangeA}_{\mathsf{Prot}}$ *is a complete, statistical honest-verifier zero-knowledge, and knowledge-sound PoK for* $\boldsymbol{m} \in [\![R]\!]^n$, *with soundness error* 2^{-n} *for the language*

$$\left\{ \mathbf{ct} \in (\mathbb{Z}_{N^2}^*)^n \mid \exists \boldsymbol{m}, \boldsymbol{r}.\ \begin{array}{c} \mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{m}, \boldsymbol{r}) \ \wedge \\ \boldsymbol{m} \in [\![2^{\lambda-1+n} nR]\!]^n \end{array} \right\}$$

Proof. ***Correctness.*** The correctness of this protocol is easy to verify. First, observe that the latter verifier's equality check passes:

$$\begin{aligned}
\boldsymbol{a}_i &= G^{(\boldsymbol{r}_m)_i} (\boldsymbol{r}_r)_i^N \\
(\boldsymbol{s}_m)_i &= (\boldsymbol{r}_m)_i + \sum_{j=1}^n E_{i,j} \boldsymbol{m}_j \qquad (\boldsymbol{s}_r)_i = (\boldsymbol{r}_r)_i \cdot \left(\prod_{j=1}^n \boldsymbol{r}_j^{E_{i,j}} \right) \\
(\boldsymbol{a} \cdot \mathbf{ct}^E)_i = \boldsymbol{a}_i \left(\prod_{j=1}^n \mathbf{ct}_j^{E_{i,j}} \right) &= G^{(\boldsymbol{r}_m)_i} (\boldsymbol{r}_r)_i^N \left(G^{\sum_{j=1}^n \boldsymbol{m}_j E_{i,j}} \left(\prod_{j=1}^n \boldsymbol{r}_j^{E_{i,j}} \right)^N \right) \\
&= G^{(\boldsymbol{r}_m)_i + \sum_{j=1}^n \boldsymbol{m}_j E_{i,j}} \left((\boldsymbol{r}_r)_i \prod_{j=1}^n \boldsymbol{r}_j^{E_{i,j}} \right)^N \\
&= G^{(\boldsymbol{s}_m)_i} ((\boldsymbol{s}_r)_i)^N \\
&= (\mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{s}_m; \boldsymbol{s}_r))_i
\end{aligned}$$

Therefore, indeed $\boldsymbol{a} \cdot \mathbf{ct}^E = \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{s}_m; \boldsymbol{s}_r)$ holds.

As for correctness of the inequality range check on verifier's side, note what happens to $\boldsymbol{s}_m = \boldsymbol{r}_m + E\boldsymbol{m}$. Since E is a binary matrix, entries in $E\boldsymbol{m}$ are in $[\![Rn]\!]$, since $(E\boldsymbol{m})_i = \langle E_i, \boldsymbol{m} \rangle$, where $\sum_j E_{i,j} \leq n$. Adding this to $\boldsymbol{r}_m$, we see that $\boldsymbol{s}_m \in [-Rn/2, 2^{\lambda-1}Rn + Rn/2]$. However, $Rn/2$ is negligibly smaller than $2^{\lambda-1}Rn$, so $\Pr[\boldsymbol{s}_m \notin [0, 2^{\lambda-1}Rn]] = \mathsf{negl}(\lambda)$ for honest provers.

Soundness. Let $(\boldsymbol{a}, c, \boldsymbol{s}_m, \boldsymbol{s}_r)$ and $(\boldsymbol{a}, \hat{c}, \hat{\boldsymbol{s}}_m, \hat{\boldsymbol{s}}_r)$ be two valid transcripts for the same commitment $\boldsymbol{a}$:

$$\mathsf{Enc}(\boldsymbol{s}_m, \boldsymbol{s}_r) = \mathbf{ct}^E \cdot \boldsymbol{a}$$
$$\mathsf{Enc}(\hat{\boldsymbol{s}}_m, \hat{\boldsymbol{s}}_r) = \mathbf{ct}^{\hat{E}} \cdot \boldsymbol{a}$$

where E and $\hat{E}$ are the two matrices corresponding to $e, \hat{e}$.

The inverse of the last equation is equal to:

$$\mathsf{Enc}(-\hat{\boldsymbol{s}}_m, \hat{\boldsymbol{s}}_r^{-1}) = \mathbf{ct}^{-\hat{E}} \cdot \boldsymbol{a}^{-1}$$

Where the negation in the first argument of Enc is as usual modulo N, and $\hat{\boldsymbol{s}}_r^{-1}$ always exists because $\hat{\boldsymbol{s}}_r \in \mathbb{Z}_{N^2}^*$ is checked by $\mathcal{V}$. Now when we multiply this by the first equation we get:

$$\mathsf{Enc}(\boldsymbol{s}_m, \boldsymbol{s}_r) \cdot \mathsf{Enc}(-\hat{\boldsymbol{s}}_m, \hat{\boldsymbol{s}}_r^{-1}) = \mathbf{ct}^{E-\hat{E}}$$

By the homomorphic property of Enc we transform the left side:

$$\mathsf{Enc}(\boldsymbol{s}_m - \hat{\boldsymbol{s}}_m, \boldsymbol{s}_r / \hat{\boldsymbol{s}}_r) = \mathbf{ct}^{E-\hat{E}} \tag{3}$$

Call $\boldsymbol{\delta_s} := \boldsymbol{s}_m - \hat{\boldsymbol{s}}_m$ and $\varDelta_E := E - \hat{E}$. Let j_0 be the first index such that $c_{j_0} \neq \hat{c}_{j_0}$, let $d_i := c_i - \hat{c}_i$, and w.l.o.g assume $d_{j_0} = 1$. Then we know by construction that $E - \hat{E}$ has a diagonal that starts at row j_0; it consists of 1, and above this diagonal are only zeroes. Because $\varDelta_E$ has such a diagonal, it has rank n and therefore is left-invertible over $\mathbb{Z}$ (and thus over $\mathbb{Z}_N$). Let M be the left inverse of $\varDelta_E$ ($M \cdot \varDelta_E = \mathbf{1}^n$), where M is $n \times m$ and consists of $\mathbb{Z}$ entries. More concretely, here is how these matrices look like:

$$M = \begin{array}{c} \begin{array}{ccccccccc} 1 & & j_0 & & & & j_0+n & & m \end{array} \\ \begin{pmatrix} 0 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & -a & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & (a^2-b) & -a & 1 & \dots & 0 & \dots & 0 \\ & & \vdots & \ddots & \ddots & \ddots & & & \\ 0 & \dots & & & & -a & 1 & \dots & 0 \end{pmatrix} \end{array} \quad \varDelta_E = \begin{pmatrix} \vdots & & 0 \\ 1 & & \\ d_{j_0+1} & \ddots & \\ \vdots & \ddots & 1 \\ d_n & & d_{j_0+1} \\ & \ddots & \vdots \\ 0 & & d_n \end{pmatrix}$$

where $(a, b, \dots) := (d_{j_0+1}, d_{j_0+2}, \dots)$ for readability. The logic of how M is constructed is that every next M_{i,j_0} is equal to $-\langle M_{i-1,j}, (d_{j_0+1}, d_{j_0+2}, \dots)\rangle$, which *importantly* is always computable over the integers and does not require taking inverses, even if N is subverted by $\mathcal{P}$.

The Eq. (3) we had before can be rewritten as:

$$(1+N)^{\boldsymbol{\delta_s}} (\boldsymbol{s}_r / \hat{\boldsymbol{s}}_r)^N = \mathbf{ct}^{\varDelta_E} \tag{4}$$

Taking a look at the exponents of $(1+N)$, we obtain $(\boldsymbol{\delta_s})_i = \sum \boldsymbol{m}_j (\varDelta_E)_{i,j} \mod N$, which is essentially $\boldsymbol{\delta_s} = \varDelta_E \cdot \boldsymbol{m} \mod N$. So to find $\boldsymbol{m}$ we must be a solution of this system over $\mathbb{Z}_N$. A similar logic holds for $\boldsymbol{r}$.

When we raise both sides of Eq. (4) to M on the left[22] we obtain:

$$(1+N)^{M(\boldsymbol{s}_m - \hat{\boldsymbol{s}}_m)}((\boldsymbol{s}_r/\hat{\boldsymbol{s}}_r)^M)^N = \boldsymbol{\mathsf{ct}}^{M(E-\hat{E})} = \boldsymbol{\mathsf{ct}}$$

which implies that $\forall i \in [1,n]$, $(\boldsymbol{m}_i, \boldsymbol{r}_i)$ is a valid message-randomness pair for $\boldsymbol{\mathsf{ct}}_i$, where $\boldsymbol{m}, \boldsymbol{r}$ are defined as follows:

$$\boldsymbol{m}_i = \sum_{j=1}^{m} M_{i,j}((\boldsymbol{s}_m)_i - (\hat{\boldsymbol{s}}_m)_i) \pmod{N}$$

$$\boldsymbol{r}_i = \prod_{j=1}^{m} \left(\frac{(\boldsymbol{s}_r)_i}{(\hat{\boldsymbol{s}}_r)_i}\right)^{M_{i,j}} \pmod{N^2}$$

This is what Ext returns, and we have just elaborated why these values are valid.

Regarding soundness of the *range check* — that is, that the protocol indeed proves $\boldsymbol{m} \in [\![2^{l+n}]\!]^n$, the analysis is similar to the one presented in [31][Lemma 11], except for our range values being different.

So we have $\boldsymbol{\delta_s} = \boldsymbol{s}_m - \hat{\boldsymbol{s}}_m = (E - \hat{E})m$, and we know by checking $\boldsymbol{s}_m$ range that $\boldsymbol{\delta_s} \in [\![2^{\lambda + \log R + \log n}]\!]^m = [\![2^{l+1}]\!]^m$. where $l = \lambda + \log R + \log n - 1$ as before.

Looking at Eq. (4), we can derive that $(\boldsymbol{\delta_s})_i = \sum_{j=1}^n \boldsymbol{m}_j \cdot (\mathit{\Delta}_E)_{i,j}$ for all $i \in [m]$. Given that the first $j_0 - 1$ rows of $\mathit{\Delta}_E$ are empty, and j_0th row is $(1\ 0 \ldots 0)$, we conclude that $(\boldsymbol{\delta_s})_{j_0} = \boldsymbol{m}_1$, where $\boldsymbol{m}_i$ is a message for $\boldsymbol{\mathsf{ct}}_i$.

By induction on $i \in [1,n]$ we will argue that $\boldsymbol{m}_i \in [\![2^{l+i}]\!]$. We have already confirmed the base case: $\boldsymbol{m}_1 = (\boldsymbol{\delta_s})_{j_0} \in [\![2^{l+1}]\!]$. First, with the change of indices:

$$\begin{aligned}(\boldsymbol{\delta_s})_{j_0-1+i} &= \sum_{j=1}^{n} \boldsymbol{m}_j \cdot (\mathit{\Delta}_E)_{j_0-1+i,j} \\ &= \sum_{j=1}^{i-1} \boldsymbol{m}_j \cdot (\mathit{\Delta}_E)_{j_0-1+i,j} + \boldsymbol{m}_i \cdot \underbrace{(\mathit{\Delta}_E)_{j_0-1+i,i}}_{=1,\text{ on diagonal}}\end{aligned}$$

Then for $i \in [2,n]$:

$$\begin{aligned}|\boldsymbol{m}_i| &= |(\boldsymbol{\delta_s})_{j_0-1+i} - \sum_{j=1}^{i-1} \boldsymbol{m}_j \cdot (\mathit{\Delta}_E)_{j_0-1+i,j}| \\ &\leq |(\boldsymbol{\delta_s})_{j_0-1+i}| + \sum_{j=1}^{i-1} |\boldsymbol{m}_j| \\ &\leq 2^l + \sum_{j=1}^{i-1} 2^{l+j-1} = 2^l \cdot (1 + \sum_{j=0}^{i-2} 2^j) = 2^{l+i-1}\end{aligned}$$

where the first $\leq$ is by triangle inequality, and second one by inductive hypothesis. This concludes the proof that $\forall i \in [1,n].\, \boldsymbol{m}_i \in [\![2^{l+i}]\!] \subset [\![2^{l+n}]\!]$, in other words $\boldsymbol{m} \in [\![2^{l+n}]\!]^n = [\![2^{\lambda+n-1} nR]\!]$.

[22] This transformation may appear somewhat nontrivial. For $\boldsymbol{x}$ of size n and two matrices A and B of sizes $n \times m$ and $m \times n$ correspondingly, it maps $\boldsymbol{x}^B$ to $\boldsymbol{x}^{AB}$, transforming every row i from $\prod_{j=1}^n \boldsymbol{x}_j^{B_{i,j}}$ into $\prod_{j=1}^n \boldsymbol{x}_j^{\sum_{k=1}^m A_{i,k}B_{k,j}}$. With a bit of linear algebra, we can express this transformation directly: given $\boldsymbol{a} = \boldsymbol{x}^B$ it returns a vector where ith component is $\prod_{k=1}^m \boldsymbol{a}_k^{A_{i,k}}$, which is simply $\boldsymbol{a}^A$.

Honest-Verifier Zero-Knowledge. The simulation strategy is as follows: $\mathcal{S}$ samples $c^* \hookleftarrow\$ \; [0, 2^\lambda]$, $\boldsymbol{s}_m^* \hookleftarrow\$ \; [0, 2^l]^m$, $\boldsymbol{s}_r^* \hookleftarrow\$ \; (\mathbb{Z}_N^*)^m$, and then computes $\boldsymbol{a}^* = \mathsf{Enc}(\boldsymbol{s}_m^*, \boldsymbol{s}_r^*) \cdot (\mathbf{ct}^{w(c)})^{-1}$. We must argue that the transcript $\mathsf{trans}^* = (\boldsymbol{a}^*, c^*, (\boldsymbol{s}_m^*, \boldsymbol{s}_r^*))$ produced by $\mathcal{S}$ is statistically indistinguishable from an honest one. Obviously, we sample $c = c^*$ identically in both worlds, so we need to prove statistical indistinguishability only for the commitment and the responses.

Starting with the responses, $\boldsymbol{s}_m^* = \mathcal{U}^m_{[0,2^l]}, \boldsymbol{s}_r^* = \mathcal{U}^m_{\mathbb{Z}_N^*}$ are uniform, while $\boldsymbol{s}_m = \boldsymbol{r}_m + E\boldsymbol{m} \mod N$ and $\boldsymbol{s}_r = \boldsymbol{r}^E \cdot \boldsymbol{r}_r \mod N$. Compare these two:

$$(\boldsymbol{s}_m^*)_i \in \mathcal{U}_{[0,2^\lambda Rn]} \qquad (\boldsymbol{s}_m)_i = \mathcal{U}_{[0,2^\lambda Rn]} + \sum (E_{i,j} \cdot \mathcal{U}_{[-R,R]} \mod N)$$

As we argued in completeness, $\sum E_{i,j} \leq n$ (since E is binary), and $Rn \ll 2^\lambda Rn$, so $\forall E$ it is easy to show $(\boldsymbol{s}_m)_i \overset{\text{s}}{\approx} \mathcal{U}_{[0,l]} = (\boldsymbol{s}_m^*)_i$. And also we have that $\forall E$,

$$(\boldsymbol{s}_r)_i = \underbrace{\left(\prod_{j=1}^{m} \boldsymbol{r}_j^{E_{i,j}} \right)}_{e \in \mathbb{Z}_N^*} \cdot (\boldsymbol{r}_r)_i = e \cdot \mathcal{U}_{\mathbb{Z}_N^*} \overset{\text{p}}{=} \mathcal{U}_{\mathbb{Z}_N^*} = (\boldsymbol{s}_r^*)_i$$

This is because $e = \left(\prod_{j=1}^{m} \boldsymbol{r}_j^{E_{i,j}} \right)$ is an element of $\mathbb{Z}_N^*$ with overwhelming probability ($\mathcal{P}, \mathcal{V}$ are honest), and thus $x \mapsto e \cdot x$ is injective.

Now as for the commitment, $\boldsymbol{a} = (1+N)^{\boldsymbol{r}_m} (\boldsymbol{r}_r)^N$ in the real world, but $\boldsymbol{a}^* = (1+N)^{\boldsymbol{s}_m - E\boldsymbol{m}} (\boldsymbol{s}_r / (\boldsymbol{r}^E))^N$ in the simulation. Again, viewing this component-wise, by the same argument as in the previous paragraph (except now we *subtract* m), for any E:

$$(\boldsymbol{s}_m)_i - \sum_{j=1}^{n} E_{i,j} \boldsymbol{m}_j \overset{\text{s}}{\approx} \mathcal{U}_{[0,2^l]} = \boldsymbol{r}_m$$

And similarly

$$\boldsymbol{s}_r / \underbrace{(\boldsymbol{r}^E)}_{e \in \mathbb{Z}_N^*} = \mathcal{U}_{\mathbb{Z}_N^*} \cdot e \overset{\text{p}}{=} \mathcal{U}_{\mathbb{Z}_N^*} = \boldsymbol{r}_r$$

Therefore, since the arguments of $\mathsf{Enc}(\cdot, \cdot)$ for $\boldsymbol{a}, \boldsymbol{a}^*$ are statistically indistinguishable, $\boldsymbol{a}^* \overset{\text{s}}{\approx} \boldsymbol{a}$. So $\mathsf{trans}^* \overset{\text{s}}{\approx} \mathsf{trans}$. □

F Statistical Blinding Lemma

In this section we recall and prove a statistical blinding lemma, Lemma F.1, that is used to prove the zero-knowledge of our protocols. This lemma shows that if an integer is chosen from a large enough domain then it statistically blinds a smaller integer.

Lemma F.1. *Let X be a random variable on $[0, R] \subset \mathbb{Z}$, and let $\mathcal{U} := \mathcal{U}_{[0,R2^\lambda]}$, then $\mathcal{U} \overset{\text{s}}{\approx} \mathcal{U} + X$.*

Proof. What we must show is

$$\varDelta(\lambda) = \frac{1}{2} \sum_{v \in \mathbb{Z}} \Big| \Pr[\mathcal{U} = v] - \Pr[\mathcal{U} + X = v] \Big| = \mathsf{negl}(\lambda)$$

Immediately, note that $\mathcal{U}+X$ is defined on $[0, R2^\lambda + R]$, therefore we should only consider v in this range.

The first probability $\Pr[\mathcal{U} = v] = 1/R2^\lambda$ for all $v \in [0, R2^\lambda]$, and is zero otherwise.

The second probability is more interesting. For any v,

$$\Pr[\mathcal{U}+X=v] = \sum_{x\in[0,R]} \Pr[X=x]\Pr[\mathcal{U}=v-x]$$

When $v \in [R, R2^\lambda]$, for any $x \in [0, R]$ we have $v - x \in [0, R2^\lambda]$, so

$$\Pr[\mathcal{U}+X=v] = \frac{1}{R2^\lambda}\sum_{x\in[0,R]} \Pr[X=x] = \frac{1}{R2^\lambda}$$

When $v \in [0, R-1]$, x only ranges in $[0, v]$, because otherwise $\mathcal{U} < 0$ and the related probability is 0:

$$\Pr[\mathcal{U}+X=v] = \sum_{x\in[0,v]} \Pr[X=x]\Pr[\mathcal{U}=v-x] = \frac{1}{R2^\lambda}F_X(v)$$

where $F_X = \Pr[X \le x]$ is a CDF of X. A similar statement holds for $v \in [R2^\lambda + 1, R2^\lambda + R]$:

$$\begin{aligned}\Pr[\mathcal{U}+X=v] &= \sum_{x\in[v-R2^\lambda,R]} \Pr[X=x]\Pr[\mathcal{U}=v-x] \\ &= \frac{1}{R2^\lambda}\Pr[X \ge v-R2^\lambda] = \frac{1}{R2^\lambda}(1-F_X(v-R2^\lambda-1))\end{aligned}$$

Combining this all together, we see that for $v \in [R, R2^\lambda]$ the value of $\mathit{\Delta}(\lambda) = 0$, so we compute it over the other two intervals:

$$\begin{aligned}\mathit{\Delta}(\lambda) &= \frac{1}{2}\left(\sum_{v\in[0,R-1]}\left|\frac{1}{R2^\lambda}-\frac{1}{R2^\lambda}F_X(v)\right| + \sum_{v=R2^\lambda+1}^{R2^\lambda+R}\left|\frac{1}{R2^\lambda}(1-F_X(v-R2^\lambda-1))\right|\right) \\ &= \frac{1}{R2^{\lambda+1}}\left(\sum_{v\in[0,R-1]}|1-F_X(v)| + \sum_{v=R2^\lambda+1}^{R2^\lambda+R}\left|1-F_X(v-R2^\lambda-1)\right|\right) \\ &= \frac{1}{R2^{\lambda+1}}\left(\sum_{v\in[0,R-1]}|1-F_X(v)| + \sum_{v\in[0,R-1]}|1-F_X(v)|\right) \\ &= \frac{1}{R2^\lambda}\left(\sum_{v\in[0,R-1]}|1-F_X(v)|\right) \le \frac{1}{2^\lambda} = \mathsf{negl}(\lambda)\end{aligned}$$

which concludes the proof. □