Optimally Sound Sigma Protocols Under DCRA

Helger Lipmaa

University of Tartu, Tartu, Estonia

Abstract. Given a well-chosen additively homomorphic cryptosystem and a Σ protocol with linear answer, Damgård, Fazio, and Nicolosi proposed a non-interactive designated-verifier zero knowledge argument in the registered public key model that is sound under non-standard complexity-leveraging assumptions. In 2015, Chaidos and Groth showed how to achieve the weaker yet reasonable culpable soundness notion under standard assumptions but only if the plaintext space order is prime. It makes use of Σ protocols that satisfy what we call the *optimal culpable soundness*. Unfortunately, most of the known additively homomorphic cryptosystems (like the Paillier Elgamal cryptosystem that is secure under the standard Decisional Composite Residuosity Assumption) have composite-order plaintext space. We construct optimally culpable sound Σ protocols and thus culpably sound non-interactive designated-verifier zero knowledge protocols for NP under standard assumptions given that the least prime divisor of the plaintext space order is large.

Keywords: Culpable soundness, designated verifier, homomorphic encryption, non-interactive zero knowledge, optimal soundness, registered public key model

1 Introduction

Non-interactive zero knowledge (NIZK, [5]) proof system enable the prover to convince the verifier in the truth of a statement without revealing any side information. Unfortunately, it is well known that NIZK proof systems are not secure in the standard model. Usually, this means that one uses the random oracle model [4] or the common reference string (CRS, [5]) model. In particular, Σ protocols [10] can be efficiently transformed into NIZK proof systems in the random oracle model by using the Fiat-Shamir heuristic [17]. However, the random oracle model (and this concrete transformation) is questionable, since there exist protocols secure in the random oracle model that are not instantiable with any function [7,19]. While newer transformations make less use of the random oracle (for example, by relying on non-programmable random oracles [27,9]), it is commonly felt that the random oracle model is at best a heuristic.

On the other hand, using the CRS model results often in less efficient protocols; moreover, also the CRS model is quite strong and requires significant amount of trust in the creator of the CRS. See [2] for some of the critique. It is desirable to construct NIZK proof systems based on a less demanding trust model.

Moreover, NIZK proof systems in the CRS model are not always perfect approximations of interactive zero knowledge proof systems [25,2,12].

First, interactive zero knowledge provides undeniability: since the verifier can simulate the proof, she cannot convince third parties that she received a ZK proof from the specific prover. Undeniability is important in many applications where it provides a certain amount of protection against third parties (for example, coercers, see [25] for more motivation).

To provide undeniability also in the case of NIZK, Jakobsson *et al.* [25] introduced the notion of *designated verifier proof systems*. A designated verifier NIZK (NIDVZK) proof system is of type "either the statement is true or I am the intended verifier (i.e., I know some witness w_V associated with the verifier)". Hence, the designated verifier is convinced that the claim is true, while for everybody else it could look like this proof came from the verifier instead of the prover and thus they will not be convinced in the veracity of the claim. While NIDVZK proofs are verifiable only by (the prover and) the designated verifier, one can argue that an NIDVZK proof system provides a good approximation of interactive zero knowledge proof systems since neither is transferable [25].

Second, one can rewind interactive zero knowledge proofs of knowledge to extract the prover's witness. This guarantees that an accepted prover also knows the witness. Such extraction is impossible, for example, in the case of some Groth-Sahai proof systems [24]. To "emulate" extractability, Groth *et al.* [23] introduced the notion of culpable soundness. In a nutshell, culpable soundness means that it should be difficult to break the soundness of a zero knowledge proof system while knowing a witness w_{guilt} that the input does not belong to the input language. Culpable soundness has been successfully used in applications like shuffling [22,16]; see [23] for other applications. Moreover, culpable soundness is also sometimes the most one can get since there exist no computationally (non-culpably) sound statistical NIZK argument systems for non-trivial languages under standard assumptions [1].

Closer to the current work, Damgård, Fazio, and Nicolosi [12] constructed a transformation (that we will call the *DFN transformation*) from an optimally sound [30][1] and specially honest-verifier zero knowledge Σ-protocol [10] with a linear answer to an NIDVZK argument system (i.e., a computationally sound NIDVZK proof system) under a complexity leveraging assumption. Recall that a Σ protocol for language $\mathcal{L}$ is *optimally sound* if the following holds: if the common input x is not in $\mathcal{L}$, then for every a there exists at most one *good* e for which there exists a z, such that (x, a, e, z) is an accepting view of the Σ protocol. Optimal soundness is a potentially weaker requirement than special soundness.

Importantly, the DFN transformation results in an NIDVZK argument system that is secure in the registered-public key (RPK, [2]) model that is considered to be significantly weaker than the CRS model. Moreover, the resulting NIDVZK argument systems are almost as efficient as the original Σ-protocols. While the DFN transformation can be only applied to optimally sound Σ-protocols with

[1] This property is also known under the name of *relaxed special soundness* [12]

linear answers, most of the known Σ-protocols in the discrete-logarithm based setting have those properties. In particular, [12] constructed an NIDVZK argument system in the RPK model for the NP-complete language Circuit-SAT.

As argued before, the designated verifier property of the DFN transformation is very useful in certain applications. Hence, the DFN transformation results in efficient argument systems, secure in a weaker trust model (the RPK model) that better approximate security properties of interactive zero knowledge proof systems than say the Groth-Sahai proof system. However, it also has weaknesses. In particular, the original DFN transform from [12] is only secure under non-standard complexity leveraging assumptions.

Ventre and Visconti [34] modified the DFN transformation to work under standard (non-leveraged) assumptions, but their NIDVZK argument system only achieves so called weak culpable soundness (called weak co-soundness in [34]).[2] As we argued before, culpable soundness approximates interactive zero knowledge. However, weak culpable soundness seems to be too restrictive, and results in undesirable overhead. We omit discussion due to space limits and refer to [8].

Recently, Chaidos and Groth [8] further modified the DFN transformation so that the resulting NIDVZK argument systems are culpably sound under standard assumptions. However, for this they assumed that the plaintext space of the underlying strongly additively homomorphic cryptosystem (see [8] for the definition of such cryptosystems), about which the Σ-protocols are, has a prime order p. Under this assumption, they showed that several known efficient Σ protocols have the optimal culpable soundness property.

However, the restriction that p is prime can be a problem in many applications, since only some cryptosystems with required properties (like the Okamoto-Uchiyama cryptosystem [31]) are known.[3] Moreover, in the Okamoto-Uchiyama cryptosystem, p must stay secret; this complicates the design of many common protocols where one needs to know the order of the plaintext space.

Our Contributions. We construct a DFN-transform under standard assumption for additively homomorphic cryptosystems where the plaintext space has a composite order N, such that it is solely required that the least prime factor of N is sufficiently large. While all our examples are about the DCRA-based Paillier Elgamal cryptosystem [14,6], it is clear that they can modified to work with other suitable cryptosystems. The main novelty of our work is proving that several *known* Σ protocols over composite order plaintext spaces are optimally culpably sound. We postpone the construction of culpably sound NIDVZK argument systems to the appendix.

[2] Briefly, weak culpable soundness means that it is difficult to cheat and at the same time know a witness assessing the fact that you are cheating, and also know that your cheating succeeds (i.e., know a witness that certifies that the verification equations hold). In the case of culpable soundness [23], the latter is not needed. See [34] for more details.

[3] The fact that one would like to have efficient Σ-protocols excludes known lattice-based cryptosystems with prime-order plaintext space.

More precisely, an optimally sound Σ protocol is *optimally culpable sound*[4] if the following property holds: a successful cheating prover $\mathcal{A}$ that knows that she cheats (e.g., by knowing the secret key of the public key cryptosystem) can efficiently recover the good e. That is, there exists an efficient extractor $\mathcal{S}.\mathsf{EX}$ that extracts good e (if it exists), given the common input, the first message of the Σ protocol (e.g., a tuple of ciphertexts) output by $\mathcal{A}$, and the guilt witness (e.g., the secret key of the underlying cryptosystem). We emphasize that the optimal culpable soundness is a stronger notion of security compared to the optimal soundness.

The main technical contribution of the current paper is the construction of an efficient $\mathcal{S}.\mathsf{EX}$ for several (known) Σ protocols about the plaintexts of the Paillier Elgamal cryptosystem. By using $\mathcal{S}.\mathsf{EX}$, we prove optimal culpable soundness of corresponding Σ protocols without relying on the Strong RSA or any other computational assumption. Importantly, the proofs of optimal culpable soundness are simpler than the special soundness proofs — that we also reproduce for the sake of completeness — for the same Σ protocols.

For the constructed extractors to be successful, it is only required that the least prime factor of N is large enough. This means that one can use essentially any known additively homomorphic public-key cryptosystem that has a large plaintext space. On the other hand, Chaidos and Groth [8] constructed $\mathcal{S}.\mathsf{EX}$ only in the case of prime-order plaintext space (with the Okamoto-Uchiyama cryptosystem being the sole mentioned candidate cryptosystem in [8]).

Before we give more details about the new Σ protocols, let us recall that the Paillier Elgamal cryptosystem has several other interesting properties:

1. First, it is double trapdoor [6]: it has two statistically independent trapdoors, the prime factorization sk_{fact} of an RSA modulus N, and an Elgamal-like secret key sk_{dl}. Decryption is possible, given either of the two trapdoors. Hence, given that N is securely generated, many different parties can operate with plaintexts and ciphertexts modulo the same N; this simplifies the design of threshold encryption schemes, [14].
2. Second, many of the standard Σ protocols, see [26], working on top of the Paillier Elgamal cryptosystem satisfy special soundness only under the Strong RSA assumption [3].

In the case of the Paillier Elgamal cryptosystem, $\mathcal{S}.\mathsf{EX}$ only needs to use the second trapdoor sk_{dl}. Hence, if a cheating prover manages to make the verifier to accept, the extractor who knows sk_{dl} can extract the good challenge, given that it exists. On top of it, the extractor may also extract a non-trivial factor of N, which means that he will break the factoring assumption. In practice, this fact is relevant in the case of threshold encryption, where such a factor can be recovered only when a majority of the key generating parties collaborate, while extraction is possible by every single party who knows the key sk_{dl}.

However, the extractor does not need factoring to be hard to be successful, i.e., extraction is unconditionally successful. Thus, while some Σ protocols about the plaintexts of the Paillier Elgamal cryptosystem are specially sound only

[4] Chaidos and Groth called it soundness with the unique identifiable challenge.

under the Strong RSA assumption, their optimal culpable soundness (and hence, also optimal soundness) is unconditional. Up to our knowledge, this separation has not been noticed before. We leave it as in interesting question whether such a phenomenon is widespread.

The modified DFN-transform achieves culpable soundness in the sense that soundness is guaranteed against adversaries that return together with the accepting view also the secret key of the prover (but no other secret value). If we require the verifier to give to the authority a zero knowledge proof of knowledge of her secret key, we can construct an adversary that retrieves the secret key from the registration process, and thus achieves the standard (not culpable) notion of soundness.

2 Preliminaries

For a predicate P, let $[P(x)]$ be 1 iff $P(x)$ is true, and 0 otherwise. We denote uniform distribution on set S by $U(S)$, and let $a \leftarrow_r S$ to denote choosing a from $U(S)$. The statistical distance between two sets $S_1, S_2 \subseteq \Omega$ is $\mathrm{SD}(U(S_1), U(S_2)) = \frac{1}{2}\sum_{x\in\Omega} |\Pr[x \in S_2] - \Pr[x \in S_1]|$. We will implicitly use the following lemma.

Lemma 1. *Let S_1 and S_2 be two finite sets. If $S_1 \subseteq S_2$, we have $\mathrm{SD}(U(S_1), U(S_2)) = 1 - |S_1|/|S_2|$. In particular, if $|S_2| = (1 + 1/t) \cdot |S_1|$ for some positive integer t, then $\mathrm{SD}(U(S_1), U(S_2)) = 1/(t + 1)$.*

Proof. $\mathrm{SD}(U(S_1), U(S_2)) = \frac{1}{2}\left(|S_2 \setminus S_1|/|S_2| + |S_1| \cdot (1/|S_1| - 1/|S_2|)\right) = 1 - |S_1|/|S_2|$. □

For a positive integer N, let $\mathrm{lpf}(N)$ be its least prime factor. Let $\varphi(N)$ be the Euler totient function. Given that $\gcd(a, b) = \gamma$, the Extended Euclidean Algorithm returns integers α and β, such that $\alpha a + \beta b = \gamma$.

For any integer a and an odd prime p, the Legendre symbol $\left(\frac{a}{p}\right)$ is defined as $\left(\frac{a}{p}\right) = 0$, if $a \equiv 0 \pmod p$, $\left(\frac{a}{p}\right) = +1$, if $a \not\equiv 0 \pmod p$ and for some integer x, $a \equiv x^2 \pmod p$, and $\left(\frac{a}{p}\right) = -1$, if there is no such x. For any integer a and any positive odd integer N, the Jacobi symbol is defined as the product of the Legendre symbols corresponding to the prime factors of N: $\left(\frac{a}{N}\right) = \prod_{i=1}^{t}\left(\frac{a}{p_i}\right)^{\alpha_i}$, where $N = \prod_{i=1}^{t} p_i^{\alpha_i}$ for different primes p_i. Let $J_N = \{a \in \mathbb{Z}_N : \left(\frac{a}{N}\right) = 1\}$; clearly $J_N \trianglelefteq \mathbb{Z}_N^*$ (i.e., J_N is a subgroup of $\mathbb{Z}_N^*$). Let $Q_N \trianglelefteq J_N$ be the subgroup of quadratic residues in $\mathbb{Z}_N$. The Jacobi symbol can be computed in polynomial time, given only a and N.

2.1 Cryptographic Assumptions

Within this paper, κ is an exponential (e.g., $\kappa \approx 128$) security parameter. We denote $f(\kappa) \approx_\kappa f'(\kappa)$, if $|f(\kappa) - f'(\kappa)| = \kappa^{-\omega(1)}$. A function $f(\kappa)$ is *negligible*, if $f(\kappa) \approx_\kappa 0$. For any κ, we assume that factoring $\tau(\kappa)$-bit integers is intractable.

Strong RSA. We say that the *Strong RSA assumption* [3] holds, if given a product $N = pq$ of two randomly chosen $\tau(\kappa)/2$-bit safe primes $p = 2p' + 1$ and $q = 2q' + 1$, and $y \leftarrow_r \mathbb{Z}_N^*$, it is computationally difficult to output (x, e), such that $e > 1$ and $y \equiv x^e \mod N$.

DCR [32,11]. Let $N = pq$ be a product of two $\tau(\kappa)/2$-bit random safe primes $p = 2p' + 1$ and $q = 2q' + 1$. Let $N' = p'q'$. Let $s \geq 1$. Write $\mathbb{G} := \mathbb{Z}^*_{N^{s+1}} \cong G_{N^s} \oplus G_{N'} \oplus G_2 \oplus T$, where $\cong$ indicates group isomomorphism, $\oplus$ is the direct sum or Cartesian product, G_i are cyclic groups of order i, and T is the order-2 cyclic group generated by $-1 \mod N^{s+1}$. Let $\mathbb{X} := \mathbb{P} := J_{N^{s+1}} \cong G_{N^s} \oplus G_{N'} \oplus T$, $\mathbb{X}' := \mathbb{P}' := Q_{N^{s+1}} \cong G_{N^s} \oplus G_{N'}$, and $\mathbb{L} \cong G_{N'}$ be multiplicative groups.

Let g be a random generator of $\mathbb{L}$; g can be thought of as a random $2N^s$-th residue. It can be computed by choosing a random $\mu \leftarrow_r \mathbb{Z}_{N^{s+1}}$ and then setting $g \leftarrow \mu^{2N^s} \mod N^{s+1}$.

A witness $w \in \mathbb{W} := \mathbb{Z}$ for $x \in \mathbb{L}$ is such that $x \equiv g^w \pmod{N^{s+1}}$. Finally, let $g_\perp$ be an arbitrary generator of the cyclic group G_{N^s} (for example $g_\perp = 1 + N \in \mathbb{Z}_{N^{s+1}}$). We set $\Lambda = (N, s, g, g_\perp)$.

The Decisional Composite Residuosity (DCR, [32]) assumption says that it is difficult to distinguish random elements of $\mathbb{L}$ from random elements of $\mathbb{X}$.

We remark that we cannot sample uniform witnesses as $\mathbb{W} = \mathbb{Z}$ is infinite. From a mathematical standpoint, we could have set $\mathbb{W} = \mathbb{Z}_{N'}$, but we cannot do that here, as computing N' from Λ requires to factorize N. Instead, we sample witnesses uniformly from $\mathbb{W}^*_N := \mathbb{Z}_{\lfloor N/4 \rfloor}$. This is satistically close to uniform over $\mathbb{Z}_{N'}$ as: $\mathrm{SD}(U(\mathbb{Z}_{N'}), U(\mathbb{W}^*_N)) = 1 - p'q'/(pq/4) = (2p' + 2q' + 1)/(pq) < 2(p + q)/(pq) < 4/\operatorname{lpf}(N)$. From this distribution over $\mathbb{W}$, we can derive a statistically uniform distribution over $\mathbb{L}$.

2.2 Paillier Elgamal Cryptosystem

We use the following CPA-secure double-trapdoor cryptosystem $\Pi = (\mathsf{K}, \mathsf{VK}, \mathsf{E}, \mathsf{D})$ that is based on a projective hash proof system from [11]. We make it proof-friendly by using ideas from [14] and augment it with the VK procedure needed to get optimal culpable soundness. Following say [29], we call this cryptosystem *Paillier Elgamal*. See, e.g., [14,6] for variants of this cryptosystem.

Let $\Lambda = (N = pq, s, g, g_\perp)$ and $(p = 2p' + 1, q = 2q' + 1)$ be chosen as in Sect. 2.1, with $N' = p'q'$. Set $\mathsf{sk}_{fact} \leftarrow (p, q)$ and $\mathsf{sk}_{dl} \leftarrow_r \mathbb{W}^*_N$. Let $h \leftarrow g^{\mathsf{sk}_{dl}} \mod N^{s+1}$. Hence, $g, h \in \mathbb{P} = J_{N^{s+1}}$. The key generator $\Pi.\mathsf{K}(\Lambda)$ returns the public key $\mathsf{pk} := (\Lambda, h)$ and the secret key $\mathsf{sk} := (\mathsf{sk}_{fact}, \mathsf{sk}_{dl})$. The message space is equal to $\mathcal{M}_{\mathsf{pk}} := \mathbb{Z}_{N^s}$, the ciphertext space is equal to $\mathcal{C}_{\mathsf{pk}} := \mathbb{P}^2$, and the randomizer space is equal to $\mathcal{R}_{\mathsf{pk}} := \mathbb{W}^*_N \times \mathbb{Z}_2 \times \mathbb{Z}_2$.

Define $\mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk}) = 1$ iff sk_{dl} is the secret key, corresponding to the public key pk. In the case of the Paillier Elgamal, VK can be evaluated efficiently by checking whether $h \equiv g^{\mathsf{sk}_{dl}} \pmod{N^{s+1}}$.

Define

$$\mathsf{E}^s_{\mathsf{pk}}(m; r, t_0, t_1) := ((-1)^{t_0} g^r, (N+1)^m (-1)^{t_1} h^r) \mod N^{s+1} .$$

Here, t_0 and t_1 are only needed for the sake of constructing zero knowledge proofs, to obtain soundness also in the case when $g \notin Q_{N^{s+1}}$ or $h \notin Q_{N^{s+1}}$. By default, one just sets $t_0 = t_1 = 0$.

Given a ciphertext $\boldsymbol{C} = (C_1, C_2)$, the decryption algorithm $\mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{C})$ first checks that $C_1, C_2 \in \mathbb{P} = J_{N^{s+1}}$ and rejects otherwise. Second, it computes $(N+1)^{2m} = (C_2/C_1^{\mathsf{sk}_{dl}})^2 \mod N^{s+1}$, and then retrieves m from this by using the algorithm described in [13]. Π is IND-CPA secure under the DCR assumption, [11].

The Paillier Elgamal cryptosystem is additively homomorphic, since $\mathsf{E}^s_{\mathsf{pk}}(m_1; r_1, t_{01}, t_{11}) \cdot \mathsf{E}^s_{\mathsf{pk}}(m_2; r_2, t_{02}, t_{12}) = \mathsf{E}^s_{\mathsf{pk}}(m_1 + m_2; r_1 + r_2, t_{01} \oplus t_{11}, t_{02} \oplus t_{12})$. Moreover, it is blindable, since for $r' \leftarrow_r \mathbb{W}^*_N$, $t_{b0} \leftarrow_r \mathbb{Z}_2$ and $t_{b1} \leftarrow_r \mathbb{Z}_2$, $\mathsf{E}^s_{\mathsf{pk}}(m; r, t_0, t_1) \cdot \mathsf{E}^s_{\mathsf{pk}}(0; r'; t_{b0}, t_{b1}) = \mathsf{E}^s_{\mathsf{pk}}(m; r + r', t_0 + t_{b0} \mod 2, t_1 + t_{b1} \mod 2)$ is a (close to uniformly) random encryption of m.

This cryptosystem has two statistically independent trapdoors, $\mathsf{sk}_{fact} = (p, q)$ and sk_{dl}. To decrypt (C_1, C_2), it suffices to have either. However, in some applications N can be generated in a highly secure environment so that its factorization is not known to anybody. Alternatively, one can create a huge N randomly, so that with a high probability it is guaranteed that N has large factors, [33]. Many different parties can then have N as a part of their public key (*without* knowing the factorization), and generate their own trapdoor sk_{dl}. A natural application is threshold encryption, where the factorization of N is only known by a threshold of the parties, while each party has their own sk_{dl}; see [14].

2.3 Σ Protocols

Let $\mathcal{R} = \{(x, w)\}$ be a polynomial-time verifiable relation, and let $\mathcal{L}_{\mathcal{R}} = \{x : (\exists w)(x, w) \in \mathcal{R}\}$, where w has polynomial length.

A *Σ-protocol* [10] $\mathcal{S}$ is a three-message protocol between the prover $\mathcal{S}.\mathsf{P}$ and the verifier $\mathcal{S}.\mathsf{V}$, where the first and the third messages are send by the prover, and the second message is a uniformly random message $e \leftarrow_r \mathbb{Z}_{2^\kappa}$ chosen by the verifier. The prover $\mathcal{S}.\mathsf{P}$ and the verifier $\mathcal{S}.\mathsf{V}$ are two efficient algorithms that have a common input x. Additionally, the prover knows a secret witness w. At the end of the Σ protocol, the verifier either accepts ($x \in \mathcal{L}_{\mathcal{R}}$) or rejects ($x \notin \mathcal{L}_{\mathcal{R}}$). We will implicitly assume that the three messages of $\mathcal{S}$ belong to some sets whose memberships can be efficiently tested.

In addition, we require the Σ protocol to have a linear answer [12].

Definition 1. *A Σ* protocol with a linear answer *for an NP-relation $\mathcal{R}$ that consists of three messages and of the verifier's decision algorithm defined by a pair $(\mathcal{S}.\mathsf{P}, \mathcal{S}.\mathsf{V})$ of efficient algorithms as follows:*

1. *$(\boldsymbol{c}_a, \boldsymbol{z}_1, \boldsymbol{z}_2) \leftarrow \mathcal{S}.\mathsf{P}(x; w)$, where $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are two m-dimensional vectors for some m. Here, $\boldsymbol{c}_a$ is the first message sent by the prover to the verifier.*
2. *The second message is $e \leftarrow_r \mathbb{Z}_{2^\kappa}$, chosen by the verifier randomly, and sent to the prover.*
3. *The third message is $\boldsymbol{z} \leftarrow e\boldsymbol{z}_1 + \boldsymbol{z}_2$, sent by the prover to the verifier.*

4. *Finally, the verifier outputs* $\mathcal{S}.\mathsf{V}(x; \boldsymbol{c}_a, e, \boldsymbol{z}) \in \{0,1\}$*, that is, the verifier either accepts or rejects.*

Here, $(x, \boldsymbol{c}_a, e, \boldsymbol{z})$ is called the (real) *view* of the Σ protocol. Thus, the verifier either rejects or accepts the view. In the latter case, the view is said to be *accepting* (for $\mathcal{S}$).

A Σ protocol $\mathcal{S}$ with a linear answer for relation $\mathcal{R}$ is *perfectly complete*, if for every $(x, w) \in \mathcal{R}$ and every $(\boldsymbol{c}_a, \boldsymbol{z}_1, \boldsymbol{z}_2) \in \mathcal{S}.\mathsf{P}(x; w)$ and $e \in \{0,1\}^\kappa$, it holds that $\mathcal{S}.\mathsf{V}(x; \boldsymbol{c}_a, e, e\boldsymbol{z}_1 + \boldsymbol{z}_2) = 1$.

A Σ protocol $\mathcal{S}$ with a linear answer for relation $\mathcal{R}$ is *perfectly (resp., statistically) special honest-verifier zero knowledge* [10], if there exists an efficient simulator $\mathcal{S}.\mathsf{sim}$ that inputs x and $e \in \{0,1\}^\kappa$, and outputs $(\boldsymbol{c}_a, \boldsymbol{z})$, such that $(x, \boldsymbol{c}_a, e, \boldsymbol{z})$ is accepting, and moreover, if e is a uniform random element of $\{0,1\}^\kappa$, then $(x, \boldsymbol{c}_a, e, \boldsymbol{z})$ has the same (resp., is negligibly different from the) distribution as the real view of $\mathcal{S}$.

A Σ protocol $\mathcal{S}$ with a linear answer is *specially sound* [10] for $\mathcal{R}$ if, given two accepting views $(x, \boldsymbol{c}_a, e, \boldsymbol{z})$ and $(x, \boldsymbol{c}_a, e', \boldsymbol{z}')$ with the same $(x, \boldsymbol{c}_a)$ but with $e \neq e'$, one can efficiently recover a witness w, such that $(x, w) \in \mathcal{R}$. A Σ protocol is *computationally specially sound* for $\mathcal{R}$ if it is specially sound for $\mathcal{R}$ under a computational assumption.

Consider any input x (possibly $x \notin \mathcal{L}_\mathcal{R}$) and any $\boldsymbol{c}_a$. Then $e \in \{0,1\}^\kappa$ is a *good challenge* [12] for a Σ protocol $\mathcal{S}$, if there exists a $\boldsymbol{z}$ such that $(x, \boldsymbol{c}_a, e, \boldsymbol{z})$ is an accepting view for $\mathcal{S}$.

Definition 2 (Optimal Soundness). *A Σ protocol $\mathcal{S}$ is* optimally sound *[30] (also known as* relaxed specially sound *[12]) for $\mathcal{R}$, if for any $x \notin \mathcal{L}_\mathcal{R}$ and any purported first message $\boldsymbol{c}_a$, there exists* at most *one good $e \in \{0,1\}^\kappa$ for $\mathcal{S}$.*

We note that in some Σ protocols it will be important not to allow e to fall outside of $\{0,1\}^\kappa$. For example, it can be the case that if e is good, then also $e + p$ is good, where $p > 2^\kappa$ is a non-trivial factor of N. There will be at most one good $e < 2^\kappa$ under the assumption that $\mathrm{lpf}(N) > 2^\kappa$.

To make the definition of optimal soundness compatible with culpable soundness, Chaidos and Groth [8] modified it as follows. (In [8] , this property was called soundness with *uniquely identifiable challenge* using relation $\mathcal{R}^{guilt}$.) We note that differently from [8], we only require the extractor to return e, if it exists; as we will show, there are cases where such e is not available.

Definition 3 (Optimal culpable soundness). *For a relation $\mathcal{R}$, let $\mathcal{R}^{guilt} = \{(x, w)\}$ be a polynomial-time verifiable relation, where it is required that $x \notin \mathcal{L}_\mathcal{R}$ if $(x, w) \in \mathcal{R}^{guilt}$ for some w. A Σ protocol $\mathcal{S}$ has* optimal culpable soundness using relation $\mathcal{R}^{guilt}$ for $\mathcal{R}$*, if (i) it is optimally sound for $\mathcal{R}$, and (ii) there exists an efficient algorithm $\mathcal{S}.\mathsf{EX}$, such that if $(x, w_{guilt}) \in \mathcal{R}^{guilt}$ then $\mathcal{S}.\mathsf{EX}_{w_{guilt}}(x, \boldsymbol{c}_a)$ returns the unique good e where $\boldsymbol{c}_a$ is a first message returned by $\mathcal{S}.\mathsf{P}$.*

It is claimed in [12] that every specially sound Σ protocol is optimally sound. As we will show in Sect. 2.3, an even stronger claim holds: there exist cases where

1. $\mathcal{S}.\mathsf{P}(\mathsf{pk}, \boldsymbol{C}; (r \in \mathbb{Z}_{\lfloor N/4 \rfloor}, b_0 \in \mathbb{Z}_2, b_1 \in \mathbb{Z}_2))$ does the following:
   (a) Set $r_a \leftarrow_r \mathbb{Z}_{2^{2\kappa}\lfloor N/4 \rfloor}$, $t_0 \leftarrow_r \mathbb{Z}_2$, $t_1 \leftarrow_r \mathbb{Z}_2$,
   (b) Set $\boldsymbol{c}_a \leftarrow \mathsf{E}^s_{\mathsf{pk}}(0; r_a, t_0, t_1)$,
   (c) Return $(\boldsymbol{c}_a, \boldsymbol{z}_1 \leftarrow (r, b_0, b_1), \boldsymbol{z}_2 \leftarrow (r_a, t_0, t_1))$.
   The prover's first message is $\boldsymbol{c}_a$.
2. The verifier's second message is $e \leftarrow_r \mathbb{Z}_{2^\kappa}$.
3. The prover sets $r_b \leftarrow er + r_a$, $t_{b0} \leftarrow eb_0 + t_0 \mod 2$, $t_{b1} \leftarrow eb_1 + t_1 \mod 2$, and outputs $\boldsymbol{z} \leftarrow (r_b, t_{b0}, t_{b1})$ as the third message.
4. The verifier $\mathcal{S}.\mathsf{V}(\mathsf{pk}, \boldsymbol{C}; \boldsymbol{c}_a, e, \boldsymbol{z})$ checks that
   (a) $\boldsymbol{C}, \boldsymbol{c}_a \in \mathbb{P}^2 = J^2_{N^{s+1}}$,
   (b) $\boldsymbol{z} = (r_b, t_{b0}, t_{b1})$, where $r_b \in \mathbb{Z}_{(2^{2\kappa}+2^\kappa-1)\lfloor N/4 \rfloor - 2^\kappa + 1}$, $t_{b0} \in \mathbb{Z}_2$, $t_{b1} \in \mathbb{Z}_2$,
   (c) the following holds:

$$(\boldsymbol{C}^e \boldsymbol{c}_a \cdot \mathsf{E}^s_{\mathsf{pk}}(0; r_b, 0, 0)^{-1})^2 \equiv \mathbf{1} \pmod{N^{s+1}} \ . \tag{1}$$

Fig. 1. Σ protocol for ZERO

the Σ protocol is computationally specially sound (for example, one needs to rely on the Strong RSA assumption [3]) and unconditionally optimally culpably sound and thus also unconditionally optimally sound.

3 New Optimally Culpably Sound Σ-Protocols

Let $\Pi = (\mathsf{K}, \mathsf{VK}, \mathsf{E}, \mathsf{D})$ be the double-trapdoor additively homomorphic cryptosystem from Sect. 2.2. We next describe two simple Σ protocols about the plaintext of a Π ciphertext that both satisfy optimal culpable soundness using a naturally defined relation $\mathcal{R}^{guilt}$ where the witness is just the secret key sk_{dl} of Π. Close variants of these Σ-protocols also work with the DCR-based cryptosystems from [13,14,6]; see, e.g., [26]. Basing the Σ protocols on Π (and not, say, on the cryptosystem from [13]) makes it easier to pinpoint some differences between the special soundness and the optimal culpable soundness.

3.1 Σ-Protocol for Zero

Consider the following Σ protocol, see Fig. 1, with linear answer for the relation

$$\mathcal{R}_{\text{ZERO}} = \{((\mathsf{pk}, \boldsymbol{C}), (r, b_0, b_1)) : \boldsymbol{C} = \mathsf{E}^s_{\mathsf{pk}}(0; r, b_0, b_1)\} \ .$$

That is, a honest verifier accepts iff $\boldsymbol{C}$ encrypts to 0.

Theorem 1. *Let Π be the Paillier Elgamal cryptosystem. The Σ protocol of Fig. 1 has linear answer, is perfectly complete, and statistically special HVZK. Assume* pk *is a valid public key. Then this Σ protocol is computationally specially sound for $\mathcal{R}$ under the Strong RSA assumption [3].*

Proof. First, clearly, $r_b \leq (2^{2\kappa} + 2^\kappa - 1)\lfloor N/4 \rfloor - 2^\kappa$.

Linear answer property: straightforward.

Perfect completeness: straightforward. If the prover is honest, we have $(\boldsymbol{C}^e \boldsymbol{c}_a \cdot \mathsf{E}^s_{\mathsf{pk}}(0; r_b, 0, 0)^{-1})^2 \equiv \mathsf{E}^s_{\mathsf{pk}}(0; er + r_a - (er + r_a), eb_0 + t_0 \mod 2, eb_1 + t_1 \mod 2))^2 \equiv \mathsf{E}^s_{\mathsf{pk}}(0; 0, 0, 0) = \mathbf{1} \pmod{N^{s+1}}$.

Statistical special HVZK: the simulator $\mathcal{S}.\mathsf{sim}(x, e)$ first sets $z \leftarrow \mathbb{Z}_{2^{2\kappa}\lfloor N/4 \rfloor}$, $t_0 \leftarrow_r \mathbb{Z}_2$, $t_1 \leftarrow_r \mathbb{Z}_2$, and then $\boldsymbol{c}_a \leftarrow \mathsf{E}^s_{\mathsf{pk}}(0; z, t_0, t_1)/\boldsymbol{C}^e$. Clearly, if $e \leftarrow_r \mathbb{Z}_{2^\kappa}$, then due to the choice of r_a, z is statistically close to z in the real protocol. Moreover, in both real and simulated protocols, $\boldsymbol{c}_a$ is defined by $((\mathsf{pk}, \boldsymbol{C}), e, z)$ and the verification equation.

Computational special soundness: From two accepting views $(\boldsymbol{c}_a, e, \boldsymbol{z} = (r_b, t_{b0}, t_{b1}))$ and $(\boldsymbol{c}_a, e', \boldsymbol{z}' = (r'_b, t'_{b0}, t'_{b1}))$ with $e \neq e'$ and Eq. (1), we get that

$$\boldsymbol{C}^{2(e-e')} \equiv \mathsf{E}^s_{\mathsf{pk}}(0; 2(r_b - r'_b), 0, 0) \equiv (g^{2(r_b - r'_b)}, h^{2(r_b - r'_b)}) \pmod{N^{s+1}} . \tag{2}$$

To recover from this the witness $r = (r_b - r'_b)/(e - e') \mod \varphi(N)$, we have to compute $(r_b - r'_b)/(e - e')$ modulo $\varphi(N)$, without knowing $\varphi(N)$. We show that one can either recover r, or break the Strong RSA assumption.

First, if $(e - e') \mid (r_b - r'_b)$ over $\mathbb{Z}$, then we set $r \leftarrow (r_b - r'_b)/(e - e')$, and we are done: $\boldsymbol{C}^2 = \mathsf{E}^s_{\mathsf{pk}}(0; 2r, 0, 0)$ and thus $\boldsymbol{C} = \mathsf{E}^s_{\mathsf{pk}}(0; r, b_0, b_1)$ for efficiently recoverable b_0 and b_1.

Second, assume $(e - e') \nmid (r_b - r'_b)$ over $\mathbb{Z}$. In this case, let $\gamma \leftarrow \gcd(2(e - e'), 2(r_b - r'_b))$, $y_e \leftarrow_r 2(e - e')/\gamma$, and $y_b \leftarrow 2(r_b - r'_b)/\gamma$. According to Eq. (2), $C_1^{2(e-e')} \equiv g^{2(r_b - r'_b)} \pmod{N^{s+1}}$, and thus also $(-1)^{t_0} C_1^{y_e} \equiv g^{y_b} \pmod{N^{s+1}}$ for efficiently computable $t_0 \in \mathbb{Z}_2$. Since $\gcd(y_b, y_e) = 1$, we can use the extended Euclidean algorithm to compute integers τ_b and τ_e, such that $\tau_b y_b + \tau_e y_e = 1$. Thus,

$$\begin{aligned} g =& g^{\tau_b y_b + \tau_e y_e} = g^{\tau_b y_b} g^{\tau_e y_e} \equiv (-1)^{\tau_b t_0} C_1^{\tau_b y_e} g^{\tau_e y_e} \\ =& (-1)^{\tau_b t_0} (C_1^{\tau_b} g^{\tau_e})^{y_e} \pmod{N^{s+1}} . \end{aligned}$$

Since $y_e > 1$, then this means that we have found a non-trivial root $(C_1^{\tau_b} g^{\tau_e} \mod N^{s+1}, y_e)$ of $(-1)^{\tau_b t_0} g$ modulo N^{s+1}, and thus also modulo N, and thus broken the Strong RSA assumption. □

Next, we will show that the same Σ-protocol from Fig. 1 has optimal culpable soundness using the relation

$$\mathcal{R}^{guilt}_{\text{ZERO}} = \left\{ \begin{array}{c} ((\mathsf{pk}, \boldsymbol{C}), \mathsf{sk}_{dl}) : \boldsymbol{C} \in \mathbb{P}^2 \wedge \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{C}) \neq 0 \wedge \\ \mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk}) = 1 \end{array} \right\} \tag{3}$$

without relying on any computational assumptions. Here, w_{guilt} is equal to sk_{dl}; hence, the extractor $\mathcal{S}.\mathsf{EX}$ gets sk_{dl} as the secret input.

Theorem 2. *Let Π be the Paillier Elgamal cryptosystem. Assume that $\mathrm{lpf}(N) > 2^\kappa$. Then the Σ protocol $\mathcal{S}$ from Fig. 1 has optimal culpable soundness using $\mathcal{R}^{guilt}_{\text{ZERO}}$.*

$\mathcal{S}.\mathsf{EX}^s_{\mathsf{sk}_{dl}}((\mathsf{pk}, \boldsymbol{C}), \boldsymbol{c}_a)$:

1. If $\boldsymbol{C} \notin \mathbb{P}^2$ or $\boldsymbol{c}_a \notin \mathbb{P}^2$: return "reject";
2. If $\mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk}) = 0$: return "reject";
3. Let $m \leftarrow \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{C})$; Let $m_a \leftarrow \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{c}_a)$;
4. If $m \equiv 0 \pmod{N^s}$: return "accept"; /* prover was honest */
5. Let $\gamma \leftarrow \gcd(m, N^s)$;
6. Let $\bar{m} \leftarrow m/\gamma$; Let $\bar{m}_a \leftarrow m_a/\gamma$; Let $\bar{N}_s \leftarrow N^s/\gamma$;
7. $e \leftarrow -\bar{m}_a/\bar{m} \mod \bar{N}_s$;
8. If $e < 2^\kappa$: return e;
9. else: return "no accepted challenges";

Fig. 2. Extractor from Thm. 2 for the Σ protocol from Fig. 1 for $\mathcal{R}^{guilt}_{\text{ZERO}}$

Proof. Consider the extractor in Fig. 2 that either returns "reject" (if $\boldsymbol{C}$ is not a valid ciphertext or $\mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk})$ does not hold; in such cases $\mathcal{S}.\mathsf{V}$ also rejects), "accept" (the prover was honest), or the good challenge (if it exists) together with a non-trivial factor of N.

We will now argue that this extractor functions as claimed. First, from the Eq. (1) of the Σ protocol in Fig. 1 it follows that

$$2(em + m_a) \equiv 0 \pmod{N^s} \ , \tag{4}$$

where m is the plaintext in $\boldsymbol{C}$ and m_a is the plaintext in $\boldsymbol{c}_a$. Since the verification accepts and N is odd, $em \equiv -m_a \pmod{N^s}$.

If $m \equiv 0 \pmod{N^s}$, then the prover is honest. Otherwise, setting $\gamma \leftarrow \gcd(m, N^s)$, we can retrieve an e that satisfies Eq. (4), given such an e exists. Really, if a good challenge e exists then $2(em + m_a) \equiv 0 \pmod{N^s}$, and thus $em + m_a \equiv 0 \pmod{N^s}$. Hence, $\bar{m}e + \bar{m}_a \equiv 0 \pmod{\bar{N}_s}$, and thus $e \equiv -\bar{m}_a/\bar{m} \pmod{\bar{N}_s}$. Since a good challenge is smaller than 2^κ, it is also smaller than $\bar{N}_s$, and thus computing e modulo $\bar{N}_s = N^s/\gamma$ does not throw away any information. Since $e\bar{m}\gamma + m_a \equiv 0 \pmod{N^s}$ and $\gamma \mid N^s$, we get $m_a \equiv 0 \pmod{\gamma}$ and thus $\gamma \mid m_a$. □

3.2 Σ Protocol for Boolean

Consider the following Σ protocol, see Fig. 3, with linear answer for the relation

$$\mathcal{R}_{\text{BOOLEAN}} = \{((\mathsf{pk}, \boldsymbol{C}), (m, r)) : \boldsymbol{C} = \mathsf{E}^s_{\mathsf{pk}}(m; r, b_0, b_1) \wedge m \in \{0, 1\}\} \ .$$

That is, a honest verifier accepts iff $\boldsymbol{C}$ encrypts to either 0 or 1. This Σ protocol is derived from the Σ protocol from [8] where it was stated for prime modulus only.

Theorem 3. *The Σ protocol (Boolean Proof) of Fig. 3 has linear answer, and it is perfectly complete and statistically special HVZK. Assume that the Strong RSA assumption [3] holds, pk is a valid public key, and $\text{lpf}(N^s) > 2^\kappa$. Then this Σ protocol is computationally specially sound.*

1. $\mathcal{S}.\mathsf{P}(\mathsf{pk}, \boldsymbol{C}; m \in \mathbb{Z}_2, (r \in \mathbb{Z}_{\lfloor N/4 \rfloor}, b_0 \in \mathbb{Z}_2, b_1 \in \mathbb{Z}_2))$ does the following:
   (a) Let $m_a \leftarrow 2^{2\kappa+1} + U(\mathbb{Z}_{2^{2\kappa}})$, $r_a \leftarrow_r \mathbb{Z}_{2^{2\kappa}\lfloor N/4 \rfloor}$, $r_b \leftarrow_r \mathbb{Z}_{2^{3\kappa}\lfloor N/4 \rfloor}$;
   (b) Let $t_{a0}, t_{a1}, t_{b0}, t_{b1}, t_{c0}, t_{c1} \leftarrow_r \mathbb{Z}_2$;
   (c) Let $\boldsymbol{c}_a \leftarrow \mathsf{E}^s_{\mathsf{pk}}(m_a; r_a, t_{a0}, t_{a1})$, $\boldsymbol{c}_b \leftarrow \mathsf{E}^s_{\mathsf{pk}}(-mm_a; r_b, t_{b0}, t_{b1})$;
   (d) Return $((\boldsymbol{c}_a, \boldsymbol{c}_b), \boldsymbol{z}_1 = (m, r, r(m-1), b_0, b_1), \boldsymbol{z}_2 = (m_a, r_a, rm_a + r_b), t_{c0}, t_{c1})$.
   The prover's first message is $(\boldsymbol{c}_a, \boldsymbol{c}_b)$.
2. The verifier's second message is $e \leftarrow_r \mathbb{Z}_{2^\kappa}$,
3. The prover's third message is $\boldsymbol{z} = (z_m, z_a, z_b, t_{d0}, t_{d1})$, where $z_m \leftarrow em + m_a$, $z_a \leftarrow er + r_a$, $z_b \leftarrow er(m-1) + rm_a + r_b$, $t_{d0} \leftarrow eb_0 + t_{c0} \mod 2$, $t_{d1} \leftarrow eb_1 + t_{c1} \mod 2$.
4. The verifier checks that
   (a) $\boldsymbol{C}, \boldsymbol{c}_a, \boldsymbol{c}_b \in \mathbb{P}^2 = J^2_{N^{s+1}}$,
   (b) $z_m \in \mathbb{Z}_{3\cdot 2^{2\kappa}+2^\kappa-1}$, $z_a \in \mathbb{Z}_{(2^{2\kappa}+2^\kappa-1)\lfloor N/4 \rfloor - 2^\kappa + 1}$,
   (c) $z_b \in \mathbb{Z}_{(2^{3\kappa}+3\cdot 2^{2\kappa}-1)\cdot\lfloor N/4 \rfloor - 3\cdot 2^{2\kappa}+1}$, $t_{d0} \in \mathbb{Z}_2$, $t_{d1} \in \mathbb{Z}_2$,
   (d) the following holds:

$$\begin{aligned}(\boldsymbol{C}^e \boldsymbol{c}_a \cdot \mathsf{E}^s_{\mathsf{pk}}(z_m; z_a, 0, 0)^{-1})^2 &\equiv \mathbf{1} \pmod{N^{s+1}} \ , \\ (\boldsymbol{C}^{z_m - e} \boldsymbol{c}_b \cdot \mathsf{E}^s_{\mathsf{pk}}(0; z_b, 0, 0)^{-1})^2 &\equiv \mathbf{1} \pmod{N^{s+1}} \ . \end{aligned} \tag{5}$$

Fig. 3. Σ protocol for Boolean

Proof. Clearly, in the honest case, $z_b = r(z_m - e) + r_b$. The choice of m_a guarantees that $z_b \geq 0$. Now,

$$\begin{aligned} z_m =& em + m_a \leq (2^\kappa - 1) + (2^{2\kappa+1} + 2^{2\kappa} - 1) = 3 \cdot 2^{2\kappa} + 2^\kappa - 2 \ , \\ z_a =& er + r_a \leq (2^\kappa - 1)(\lfloor N/4 \rfloor - 1) + (2^{2\kappa} \lfloor N/4 \rfloor - 1) \\ =& (2^{2\kappa} + 2^\kappa - 1) \lfloor N/4 \rfloor - 2^\kappa \ , \end{aligned}$$

and (here we need that $m_a > e$)

$$\begin{aligned} z_b =& er(m-1) + rm_a + r_b \\ \leq& (2^\kappa - 1)(\lfloor N/4 \rfloor - 1) \cdot 0 + (\lfloor N/4 \rfloor - 1)(2^{2\kappa+1} + 2^{2\kappa} - 1) + (2^{3\kappa} \lfloor N/4 \rfloor - 1) \\ =& \left(2^{3\kappa} + 3 \cdot 2^{2\kappa} - 1\right) \cdot \lfloor N/4 \rfloor - 3 \cdot 2^{2\kappa} \ . \end{aligned}$$

Linear answer: straightforward. Completeness: let $t_{ei} = b_i(m_a + e(m-1)) + t_{bi}$ for $i \in \{0, 1\}$. Eq. (5) holds since

$$\begin{aligned} \boldsymbol{C}^{z_m - e} \boldsymbol{c}_b \equiv& \mathsf{E}^s_{\mathsf{pk}}((em + m_a - e)m - mm_a; r(z_m - e) + r_b, t_{e0}, t_{e1}) \\ \equiv& \mathsf{E}^s_{\mathsf{pk}}(e(m-1)m; z_b, t_{e0}, t_{e1}) \equiv \mathsf{E}^s_{\mathsf{pk}}(0; z_b, t_{e0}, t_{e1}) \ , \end{aligned}$$

if $m \in \{0, 1\}$. Thus, $\boldsymbol{C}^{2(z_m - e)} \boldsymbol{c}_b^2 \equiv \mathsf{E}^s_{\mathsf{pk}}(0; 2z_b, 0, 0)$ if $m \in \{0, 1\}$. Other verifications are straightforward.

Statistical special HVZK: Given $e \in \mathbb{Z}_{2^\kappa}$, the simulator generates $z_m \leftarrow_r 2^{2\kappa+1} + U(Z_{2^{2\kappa}})$, $z_a \leftarrow_r \mathbb{Z}_{2^{2\kappa}\lfloor N/4 \rfloor}$, $z_b \leftarrow_r \mathbb{Z}_{2^{3\kappa}\lfloor N/4 \rfloor}$, and $t_{a0}, t_{a1}, t_{b0}, t_{b1}, t_{d0}, t_{d1} \leftarrow_r \mathbb{Z}_2$. He sets $\boldsymbol{z} \leftarrow (z_m, z_a, z_b, t_{d0}, t_{d1})$,

$\boldsymbol{c}_a \leftarrow \mathsf{E}^s_{\mathsf{pk}}(z_m; z_a, t_{a0}, t_{a1})/\boldsymbol{C}^e \mod N^{s+1}$ and $\boldsymbol{c}_b \leftarrow \mathsf{E}^s_{\mathsf{pk}}(0; z_b, t_{b0}, t_{b1})/\boldsymbol{C}^{z_m - e} \mod N^{s+1}$, and returns $(\mathsf{pk}, \boldsymbol{C}; (\boldsymbol{c}_a, \boldsymbol{c}_b), e, \boldsymbol{z})$ as the view. Clearly, both in the real and simulated proof, $\boldsymbol{c}_a$ and $\boldsymbol{c}_b$ are fixed by $(\mathsf{pk}, \boldsymbol{C}; e, \boldsymbol{z})$ and the verification equations. Moreover, given that $e \leftarrow_r \mathbb{Z}_{2^\kappa}$, the simulated $z_m, z_a, z_b, t_{d0}, t_{d1}$ are statistically close to the values in the real proof.

Special Soundness: Assume that the verifier accepts two views $(\mathsf{pk}, \boldsymbol{C}; \boldsymbol{c}_a, \boldsymbol{c}_b, e, \boldsymbol{z})$ and $(\mathsf{pk}, \boldsymbol{C}; \boldsymbol{c}_a, \boldsymbol{c}_b, e', \boldsymbol{z}')$ for $e \neq e'$. From the first equality in Eq. (5) we get that

$$\boldsymbol{C}^{2(e-e')} \equiv \mathsf{E}_{\mathsf{pk}}(2(z_m - z'_m); 2(z_a - z'_a), 0, 0) \ . \tag{6}$$

Hence, $\boldsymbol{C}$ encrypts $m := (z_m - z'_m)/(e - e') \mod N^s$. (Here, we use the fact that $e, e' \in \mathbb{Z}_{2^\kappa} < \mathrm{lpf}(N^s)$, $e \neq e'$, and thus $e - e'$ is invertible.) To recover the randomizer used in encrypting $\boldsymbol{C}$, we use the same technique as in the proof of Thm. 1: we either obtain that $(e - e') \mid (z_a - z'_a)$ (in this case, we set $r \leftarrow (z_a - z'_a)/(e - e')$), or we break the Strong RSA assumption. Similarly, we obtain the randomizers b_0 and b_1 that were used when computing $\boldsymbol{C}$.

From the second equality in Eq. (5) holds, we get that

$$\boldsymbol{C}^{2(z_m - z'_m) - 2(e-e')} \equiv \mathsf{E}^s_{\mathsf{pk}}(0; 2(z_b - z'_b), 0, 0) \pmod{N^{s+1}} \ ,$$

and thus, when combining it with Eq. (6),

$$\begin{aligned} &\mathsf{E}^s_{\mathsf{pk}}(2(z_m - z'_m)m; 2(z_m - z'_m)r, 0, 0) \\ &\qquad \equiv \mathsf{E}^s_{\mathsf{pk}}(2(z_m - z'_m); 2(z_a - z'_a + z_b - z'_b), 0, 0) \pmod{N^{s+1}} \ , \end{aligned}$$

Since $z_m - z'_m \equiv (e - e')m \pmod{N^s}$, we get after decrypting that

$$2(e - e')m^2 \equiv 2(e - e')m \pmod{N^s} \ .$$

Since $\gcd(e - e', N^s) = 1$, $m \mod N^s \in \{0, 1\}$. □

Next, we show that this Σ protocol has optimal culpable soudness using the guilt relation

$$\mathcal{R}^{guilt}_{\textsc{Boolean}} = \left\{ \begin{array}{c} ((\mathsf{pk}, \boldsymbol{C}), \mathsf{sk}_{dl}) : \boldsymbol{C} \in \mathbb{P}^2 \wedge \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{C}) \notin \{0, 1\} \wedge \\ \mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk}) = 1 \end{array} \right\} \ . \tag{7}$$

Theorem 4. *Let Π be the Paillier Elgamal cryptosystem, and let $\mathrm{lpf}(N) > 2^\kappa$ (thus also $2 \nmid N$). Then the Σ protocol of Fig. 3 has optimal culpable soundness using $\mathcal{R}^{guilt}_{\textsc{Boolean}}$.*

Proof. We prove the optimal culpable soundness as in Thm. 2. The main new complication is that there can now be two strategies of cheating: it can be that either $\gcd(m, N^s) > 1$ or $\gcd(m - 1, N^s) > 1$, so the extractor has to test for both. We thus construct the following extractor, see Fig. 4.

$\mathcal{S}.\mathsf{EX}_{\mathsf{sk}_{dl}}(\boldsymbol{C}, \boldsymbol{c}_a, \boldsymbol{c}_b)$:

1. If $\boldsymbol{C} \notin \mathbb{P}^2$ or $\boldsymbol{c}_a \notin \mathbb{P}^2$ or $\boldsymbol{c}_b \notin \mathbb{P}^2$: return "reject";
2. If $\mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk}) = 0$: return "reject";
3. Let $m \leftarrow \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{C})$;
4. Let $m_a \leftarrow \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{c}_a)$, $m_b \leftarrow \mathsf{D}^s_{\mathsf{sk}_{dl}}(\boldsymbol{c}_b)$;
5. Let $m^* \leftarrow (m-1)m \mod N^s$;
6. If $m^* \equiv 0 \pmod{N^s}$: return "accept";
7. else if $m^* \in \mathbb{Z}^*_{N^s}$: let $e \leftarrow -(mm_a + m_b)/m^* \mod N^s$;
8. else if $\gcd(m, N^s) > 1$:
   (a) Let $\gamma \leftarrow \gcd(m, N^s)$;
   (b) Let $\bar{m} \leftarrow m/\gamma$; $\bar{m}_b \leftarrow m_b/\gamma$, $\bar{m}^* \leftarrow m^*/\gamma$; $\bar{N}_s \leftarrow N^s/\gamma$;
   (c) Let $e \leftarrow_r -(m_a\bar{m} + \bar{m}_b)/\bar{m}^* \mod \bar{N}_s$;
9. else: /* $\gcd(m-1, N^s) > 1$ */
   (a) Let $\gamma \leftarrow \gcd(m-1, N^s)$;
   (b) Let $\bar{m}_1 \leftarrow (m-1)/\gamma$, $\bar{m}_{ab} \leftarrow (m_a + m_b)/\gamma$, $\bar{m}^* \leftarrow m^*/\gamma$, $\bar{N}_s \leftarrow N^s/\gamma$;
   (c) Let $e \leftarrow_r -(m_a\bar{m}_1 + \bar{m}_{ab})/\bar{m}^* \mod \bar{N}_s$;
10. If $e < 2^\kappa$: return e;
11. else: return "no accepted challenges";

Fig. 4. Extractor in Thm. 4 for $\mathcal{R}^{guilt}_{\textsc{Boolean}}$

Let $m^* := (m-1)m \mod N^s$. From the verification equalities in Eq. (5) we get that $z_m \equiv em + m_a \pmod{N^s}$ and $(z_m - e)m + m_b \equiv 0 \pmod{N^s}$, thus $(em + m_a - e)m + m_b \equiv 0 \pmod{N^s}$, and thus

$$em^* \equiv -(m_a m + m_b) \pmod{N^s} \ . \tag{8}$$

Clearly, the constructed extractor works correctly. If $m^* \equiv 0 \pmod{N^s}$ or $m^* \equiv 1 \pmod{N^s}$, then the prover was honest. Otherwise, if $m^* \in \mathbb{Z}^*_{N^s}$, then one can recover e from Eq. (8) efficiently. Otherwise, if $\gcd(m^*, N^s) > 1$, we have either $\gcd(m, N^s) > 1$ or $\gcd(m-1, N^s) > 1$. Those two possibilities are mutually exclusive, since $\gcd(m, m-1) = 1$.

In the case $\gamma = \gcd(m, N^s) > 1$, we can divide the left hand side and right hand side of Eq. (8) by γ, and obtain $e \mod (N^s/\gamma)$ as in Fig. 4, line 8c. This is possible since in this case, from Eq. (8) we get that $e(m-1)\bar{m}\gamma \equiv -(m_a\bar{m}\gamma + m_b) \pmod{N^s}$ and hence $m_b \equiv 0 \pmod{\gamma}$ and $\gamma \mid m_b$. Since $e < 2^\kappa < \mathrm{lpf}(N)$, we have obtained e.

In the case $\gamma = \gcd(m-1, N^s) > 1$, we can divide the left hand side and right hand side of Eq. (8) by γ, and obtain $e \mod (N^s/\gamma)$ as in Fig. 4, line 9c.. This is possible since in this case, we can rewrite Eq. (8) as $e(m-1)m \equiv -(m_a(m-1) + m_a + m_b) \pmod{N^s}$. Thus, we get that $e\bar{m}_1\gamma m \equiv -(m_a\bar{m}_1\gamma + m_a + m_b) \pmod{N^s}$ and hence $m_a + m_b \equiv 0 \pmod{\gamma}$ and $\gamma \mid (m_a + m_b)$. Since $e < 2^\kappa < \mathrm{lpf}(N)$, we have obtained e.

This finishes the proof. □

3.3 Σ Protocol for Circuit-SAT

To construct a Σ protocol for the NP-complete language Circuit-SAT, it suffices to construct a Σ protocol for Boolean [8]. Really, each circuit can be represented only by using NAND gates, and a NAND $b = c$ iff $a + b + 2c - 2 \in \{0, 1\}$ [23].

One hence just has to prove that (i) each input and wire value is Boolean, and (ii) each gate is correctly evaluated. According to [15], each test in step ii can be reformulated as a Boolean test. Hence, it is sufficient to run $m + n$ Σ protocols for Boolean in parallel, where m is the summatory number of the inputs and the wires, and n is the number of gates. See [8] for more information.

3.4 General Idea

In both covered cases (Zero and Boolean), we constructed Σ protocols that were specially sound and HVZK, and then applied the following idea to obtain optimal culpable soundness. We expect the same idea to work also in general.

Let $\mathcal{L} \subset \mathcal{C}^{\mathsf{n}}_{\mathsf{pk}}$ be a language about the ciphertexts of Π that naturally defines a language $\mathcal{L}_M \subset \mathcal{M}^{\mathsf{n}}_{\mathsf{pk}}$ about the plaintexts. For example, in the case $\mathcal{L} = \text{Zero}$, $\mathcal{L}_M = \{0\}$. Let $\mathcal{R} = \{(x, w) : x \in \mathcal{L}\}$ and, for some n,

$$\mathcal{R}^{guilt} = \left\{ \begin{array}{l} (x = (\mathsf{pk}, \boldsymbol{C}, \mathsf{sk}_{dl}) : \boldsymbol{C} \in \mathcal{C}^{\mathsf{n}}_{\mathsf{pk}} \wedge (\boldsymbol{C}_i)_{i=1}^{\mathsf{n}} \notin \mathcal{L}_{\mathcal{R}} \wedge \\ \mathsf{VK}(\mathsf{sk}_{dl}, \mathsf{pk}) = 1 \end{array} \right\} . \tag{9}$$

The general idea is to construct a Σ-protocol with the following property. If the prover is cheating, then for each first message $\boldsymbol{c}_a$ there is at most one good e. Moreover, this e can be computed as $e = e_1/e_2$, where either e_2 is invertible modulo N^s or e_2/γ is invertible modulo N^s/γ, where γ is the greatest common divisor of N^s and some function $f(m)$ of $m \notin \mathcal{L}_M$ such that $f(m) \neq 0$.

Acknowledgments. We would like to thank Jens Groth, Ivan Visconti and anonymous reviewers for insightful comments. The authors were supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 653497 (project PANORAMIX), and by institutional research funding IUT2-1 of the Estonian Ministry of Education and Research.

References

1. Abe, M., Fehr, S.: Perfect NIZK with Adaptive Soundness. In: TCC 2007. LNCS, vol. 4392, pp. 118–136
2. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally Composable Protocols with Relaxed Set-Up Assumptions. In: FOCS 2004, pp. 186–195
3. Barić, N., Pfitzmann, B.: Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In: EUROCRYPT 1997. LNCS, vol. 1233, pp. 480–494
4. Bellare, M., Rogaway, P.: Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. In: ACM CCS 1993, pp. 62–73

5. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and Its Applications. In: STOC 1988, pp. 103–112
6. Bresson, E., Catalano, D., Pointcheval, D.: A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications. In: ASIACRYPT 2003. LNCS, vol. 2894, pp. 37–54
7. Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Methodology, Revisited. In: STOC 1998, pp. 209–218
8. Chaidos, P., Groth, J.: Making Sigma-Protocols Non-interactive Without Random Oracles. In: PKC 2015. LNCS, vol. 9020, pp. 650–670
9. Ciampi, M., Persiano, G., Siniscalchi, L., Visconti, I.: A Transform for NIZK Almost as Efficient and General as the Fiat-Shamir Transform Without Programmable Random Oracles. In: TCC 2016-A (2). LNCS, vol. 9563, pp. 83–111
10. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: CRYPTO 1994. LNCS, vol. 839, pp. 174–187
11. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64
12. Damgård, I., Fazio, N., Nicolosi, A.: Non-interactive Zero-Knowledge from Homomorphic Encryption. In: TCC 2006. LNCS, vol. 3876, pp. 41–59
13. Damgård, I., Jurik, M.: A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In: PKC 2001. LNCS, vol. 1992, pp. 119–136
14. Damgård, I., Jurik, M.: A Length-Flexible Threshold Cryptosystem with Applications. In: ACISP 2003. LNCS, vol. 2727, pp. 350–364
15. Danezis, G., Fournet, C., Groth, J., Kohlweiss, M.: Square Span Programs with Applications to Succinct NIZK Arguments. In: ASIACRYPT 2014 (1). LNCS, vol. 8873, pp. 532–550
16. Fauzi, P., Lipmaa, H.: Efficient Culpably Sound NIZK Shuffle Argument without Random Oracles. In: CT-RSA 2016. LNCS, vol. 9610, pp. 200–216
17. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: CRYPTO 1986. LNCS, vol. 263, pp. 186–194
18. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic Span Programs and NIZKs without PCPs. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645
19. Goldwasser, S., Kalai, Y.T.: On the (In)security of the Fiat-Shamir Paradigm. In: FOCS 2003, pp. 102–113
20. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459
21. Groth, J.: Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In: ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340
22. Groth, J., Lu, S.: A Non-interactive Shuffle with Pairing Based Verifiability. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 51–67
23. Groth, J., Ostrovsky, R., Sahai, A.: New Techniques for Noninteractive Zero-Knowledge. Journal of the ACM **59**(3) (2012)
24. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432
25. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated Verifier Proofs and Their Applications. In: EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154
26. Jurik, M.J.: Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols. PhD thesis, University of Aarhus, Denmark (2003)

27. Lindell, Y.: An Efficient Transform from Sigma Protocols to NIZK with a CRS and Non-programmable Random Oracle. In: TCC 2015 (1). LNCS, vol. 9014, pp. 93–109
28. Lipmaa, H.: Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In: TCC 2012. LNCS, vol. 7194, pp. 169–189
29. Malkin, T., Teranishi, I., Yung, M.: Efficient Circuit-Size Independent Public Key Encryption with KDM Security. In: EUROCRYPT 2011. LNCS, vol. 6632, pp. 507–526
30. Micciancio, D., Petrank, E.: Simulatable Commitments and Efficient Concurrent Zero-Knowledge. In: EUROCRYPT 2003. LNCS, vol. 2656, pp. 140–159
31. Okamoto, T., Uchiyama, S.: A New Public-Key Cryptosystem as Secure as Factoring. In: EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318
32. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238
33. Sander, T.: Efficient Accumulators without Trapdoor. In: ICICS 1999. LNCS, vol. 1726, pp. 252–262. ISBN 3-540-66682-6.
34. Ventre, C., Visconti, I.: Co-sound Zero-Knowledge with Public Keys. In: AFRICACRYPT 2009. LNCS, vol. 5580, pp. 287–304

A Preliminaries: DFN

A.1 RPK Model

In the registered public key (RPK, [2]) model, we assume that everybody has an access to a key registration functionality F_{kr}. A party (say, Alice) generates her public and secret key pair, and then sends both (together with used random coins) to F_{kr}, who verifies that the keys were created correctly (this means that to register her public key, Alice must know the corresponding private key), and then stores the public key together with Alice's identity in a repository.

Later, Bob (for this, it is not necessary for Bob to register his public key) can query F_{kr} and then retrieve the public key of Alice together with a corresponding certificate. On the other hand, in security proofs, we may give an adversary control over F_{kr}, enabling access not only to the public but also to the secret key of Alice. While every party can use a different F_{kr}, all parties need to trust F_{kr} of other parties in the following sense. F_{kr} guarantees that

(i) the public keys of uncorrupted parties are safe (the corresponding secret key is chosen randomly, and kept secret from the adversary), and
(ii) the public keys of corrupted parties are well-formed (the functionality has seen the corresponding secret key).

Hence, Alice must trust her F_{kr} to do key registration correctly, and Bob must trust that Alice's F_{kr} has verified that Alice knows the corresponding secret key.

As noted in [2,12], one can make this model more realistic by letting Alice to send her public key to F_{kr} and then give an interactive zero knowledge proof that she knows the corresponding private key. In the security proof, we can then construct an adversary who rewinds Alice to extract her private key.

A.2 NIDVZK Argument Systems

In a *non-interactive designated verifier zero knowledge (NIDVZK, [8]) argument system in the RPK model*, the verifier has a public key $\mathcal{Z}.\mathsf{pk}$ and a corresponding secret key $\mathcal{Z}.\mathsf{sk}$ specific to this argument system, that she has set up by using a trusted functionality F_{kr}. An NIDVZK argument system $\mathcal{Z}$ consists of the following three efficient algorithms:

$\mathcal{Z}.\mathsf{G}(1^\kappa)$**:** generates, registers (by using F_{kr}), and then returns a key pair $(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk})$.

$\mathcal{Z}.\mathsf{P}(\mathcal{Z}.\mathsf{pk}, x, w)$**:** given a public key $\mathcal{Z}.\mathsf{pk}$ obtained from F_{kr}, an input x and a witness w, returns a proof π.

$\mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk}, x, \pi)$**:** given a secret key, an input x, and a proof π, returns either 1 (accept) or 0 (reject).

Next, $\mathcal{Z} = (\mathcal{Z}.\mathsf{G}, \mathcal{Z}.\mathsf{P}, \mathcal{Z}.\mathsf{V})$ is an *NIDVZK argument system*[5] *for* $\mathcal{R}$ *with culpable soundness for* $\mathcal{R}^{guilt}$, if it is perfectly complete, culpably sound [23] for $\mathcal{R}^{guilt}$, and statistically (or computationally) composable zero knowledge, given that the parties have access to the certified public key of the verifier. More precise definitions follow.

Let $\ell_x(\kappa)$ be a polynomial, such that (common) inputs of length $\ell_x(\kappa)$ correspond to security parameter κ. Then let $\mathcal{R}_\kappa = \{(x, w) : \mathit{bitlength}(x) = \ell_x(\kappa)\}$ and $\mathcal{L}_{\mathcal{R},\kappa} = \{x : (\exists w)(x, w) \in \mathcal{R}_\kappa\}$, where again w has polynomial length.

$\mathcal{Z}$ is *perfectly complete*, if for all $\kappa \in \mathbb{N}$, all $(x, w) \in \mathcal{R}_\kappa$, and all $(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}) \in \mathcal{Z}.\mathsf{G}(1^\kappa)$, $\mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk}, x, \mathcal{Z}.\mathsf{P}(\mathcal{Z}.\mathsf{pk}, x, w)) = 1$.

In our constructions we will get zero-knowledge even if the adversary knows the secret verification key. This strong type of zero-knowledge is called composable zero-knowledge in [20] due to it making composition of zero-knowledge arguments easier. More precisely, it is required that even an adversary who knows the secret key (or trapdoor, in the CRS model) cannot distinguish between the real and the simulated argument, [20].

Definition 4. *$\mathcal{Z}$ is* computationally composable zero-knowledge *if there exists an efficient simulator $\mathcal{Z}.\mathsf{sim}$, such that for all probabilistic polynomial-time stateful adversaries $\mathcal{A}$,*

$$\Pr\left[\begin{array}{l}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}) \leftarrow \mathcal{Z}.\mathsf{G}(1^\kappa),\\ (x, w) \leftarrow \mathcal{A}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}),\\ \pi \leftarrow \mathcal{Z}.\mathsf{P}(\mathcal{Z}.\mathsf{pk}, x, w) :\\ (x, w) \in \mathcal{R} \wedge \mathcal{A}(\pi) = 1\end{array}\right] \approx_\kappa \Pr\left[\begin{array}{l}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}) \leftarrow \mathcal{Z}.\mathsf{G}(1^\kappa),\\ (x, w) \leftarrow \mathcal{A}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}),\\ \pi \leftarrow \mathcal{Z}.\mathsf{sim}(\mathcal{Z}.\mathsf{sk}, x) :\\ (x, w) \in \mathcal{R} \wedge \mathcal{A}(\pi) = 1\end{array}\right].$$

$\mathcal{Z}$ is statistically composable zero-knowledge *if this holds for all (not necessarily efficient) adversaries $\mathcal{A}$. A statistically composable zero-knowledge argument system is perfectly composable, if $\approx_\kappa$ can be replaced with $=$ (i.e., the above two probabilities are in fact equal).*

[5] We recall that an argument system is a proof system where soundness only holds against efficient adversaries.

In the case of culpable soundness [23], we only consider false statements from some language $\mathcal{L}_{guilt} \subseteq \overline{\mathcal{L}}$ characterized by a relation $\mathcal{R}^{guilt}$. We require a successfully cheating prover to output, together with an input x and a successful argument π, also a guilt witness w_{guilt} such that $(x, w_{guilt}) \in \mathcal{R}^{guilt}$. That is, we require a successful cheater to be aware of the fact that she cheated.

Formally, $\mathcal{Z}$ is *(non-adaptively) culpably sound for* $\mathcal{R}^{guilt}$, if for all probabilistic polynomial-time adversaries $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}) \leftarrow \mathcal{Z}.\mathsf{G}(1^\kappa), (x, \pi, w_{guilt}) \leftarrow \mathcal{A}(\mathcal{Z}.\mathsf{pk}) : \\ (x, w_{guilt}) \in \mathcal{R}^{guilt} \wedge \mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk}, x, \pi) = 1\end{array}\right] \approx_\kappa 0 \ .$$

Note that culpable soundness is implicitly computational (defined only w.r.t. to an efficient adversary), thus a culpably sound proof system is always an argument system.

In our applications, w_{guilt} will be the secret key of the cryptosystem, about which the NIDVZK arguments are about. For example, in an NIDVZK argument that the plaintext is 0 (or Boolean), w_{guilt} is equal to the secret key that enables to decrypt the ciphertext. Such culpable soundness is fine in many applications, as we will discuss at the end of the current subsection.

Finally, for some $\varrho = \varrho(\kappa)$, $\mathcal{Z}$ is *ϱ-adaptively culpably sound for* $\mathcal{R}^{guilt}$, if for all probabilistic polynomial-time adversaries $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}) \leftarrow \mathcal{Z}.\mathsf{G}(1^\kappa), (x, \pi, w_{guilt}) \leftarrow \mathcal{A}^{\mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk}, \cdot, \cdot)}(\mathcal{Z}.\mathsf{pk}) : \\ (x, w_{guilt}) \in \mathcal{R}^{guilt} \wedge \mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk}, x, \pi) = 1\end{array}\right] \approx_\kappa 0 \ .$$

Here, the adversary is allowed to make up to ϱ queries to the oracle $\mathcal{Z}.\mathsf{V}$.

As shown in [12], one can handle cases where the adversary has an access to a logarithmic number of queries, simulating their answers by guessing their answers; this still guarantees that her success probability is inverse polynomial.

On Culpable Soundness. We will prove culpable soundness [23] of argument systems about the plaintexts of a cryptosystem by showing that if an adversary outputs an accepting argument and the secret key sk, then she has broken an underlying assumption. This version of culpable soundness is acceptable since in protocols that we are interested in, there always exists a party (namely, the verifier) who knows sk. Hence, the cheating adversary together with the verifier can break the (non-culpable) soundness of the argument system.

Thus, such culpable soundness is very natural the RPK model, especially if we assume that the verifier has provided an interactive zero knowledge proof of knowledge of sk while registering it with the authority. Then, in the soundness proof, we can just construct an adversary who first retrieves sk from the latter zero knowledge proof, and then uses the culpable soundness adversary whom we already have.

A.3 DFN Transform for the Paillier Elgamal Cryptosystem

Consider the DFN [12] transformation, given the Paillier Elgamal cryptosystem $\Pi = (\Pi.\mathsf{K}, \mathsf{VK}, \mathsf{E}, \mathsf{D})$ where the plaintext space is $\mathbb{Z}_{N^s}$ for some reasonably large

$\mathcal{Z}.\mathsf{G}(1^\kappa)$
$(\mathsf{sk}_e, \mathsf{pk}_e) \leftarrow \Pi.\mathsf{K}(1^\kappa)$
$r_e \leftarrow_r U(\mathbb{W}_N^*)$
$e \leftarrow_r \mathbb{Z}_{2^\kappa}$
$\boldsymbol{c}_e \leftarrow_r \mathsf{E}^s_{\mathsf{pk}_e}(e; r_e)$
$\mathcal{Z}.\mathsf{pk} \leftarrow (\mathsf{pk}_e, \boldsymbol{c}_e)$
$\mathcal{Z}.\mathsf{sk} \leftarrow (\mathsf{sk}_e, e)$
Return $(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk})$

$\mathcal{Z}.\mathsf{P}(\mathcal{Z}.\mathsf{pk}; \boldsymbol{C}; \boldsymbol{m}, \boldsymbol{r}, \boldsymbol{b}_0, \boldsymbol{b}_1)$
// $\boldsymbol{C}_i = \mathsf{E}^s_{\mathsf{pk}}(m_i; r_i, b_{0i}, b_{1i})$
$(\boldsymbol{c}_a, \boldsymbol{z}_1, \boldsymbol{z}_2) \leftarrow$
$\mathcal{S}.\mathsf{P}(\mathsf{pk}, \boldsymbol{C}; \boldsymbol{m}, \boldsymbol{r}, \boldsymbol{b}_0, \boldsymbol{b}_1)$
For $i = 1$ to n:
$\quad r_i \leftarrow \mathbb{W}_N^*$
$\quad c_{zi} \leftarrow \boldsymbol{c}_e^{z_{1i}} \cdot \mathsf{E}^s_{\mathsf{pk}_e}(z_{2i}; r_i, b_{0i}, b_{1i})$
Return $\pi \leftarrow (\boldsymbol{c}_a, \boldsymbol{c}_z)$

$\mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk}; \boldsymbol{C}, \pi)$
Parse $\pi = (\boldsymbol{c}_a, \boldsymbol{c}_z)$
For $i = 1$ to n:
$\quad z_i \leftarrow \mathsf{D}^s_{\mathsf{sk}_e}(c_{zi})$
Return $\mathcal{S}.\mathsf{V}(\boldsymbol{C}; \boldsymbol{c}_a, e, \boldsymbol{z})$

Fig. 5. The DFN transform for the Paillier Elgamal cryptosystem. Here we assume $s = \max_i \lceil \log_N(z_{2i} + 1) \rceil$ is fixed by the description of $\mathcal{S}.\mathsf{P}$ and thus known to the verifier

s. W.l.o.g., we assume that the same cryptosystem is used to encrypt the challenge e and the witness plaintexts and the same value of s, but by using the different secret and public keys where one secret key sk_e is known by the verifier and another secret key sk is (possibly) known by the prover. For the sake of efficiency, one could use different cryptosystems or at least different values of s but we will avoid the general case not to clutter the notation.

This transformation assumes that the original Σ-protocol $\mathcal{S}$ is has a linear answer and optimal culpable soundness using some relation $\mathcal{R}^{guilt}$, see Sect. 2.3. More precisely, we assume that $\mathcal{R}^{guilt}$ is as defined by Eq. (9).

The description of the DFN transform is given in Fig. 5. The following theorem and its proof follows [12,8] in its structure. The part of using the extractor to achieve culpable soundness is from [8] while the idea of letting the constructed adversary $\mathcal{A}_\pi$ answer randomly to oracle queries goes back to [12,8]. The latter means that we only get $O(\log \kappa)$-adaptive soundness.

Theorem 5. *Assume that $\mathcal{S}$ is a complete and computationally (resp., statistically) special HVZK Σ protocol with linear answer for $\mathcal{R}$ that is optimally culpably sound for $\mathcal{R}^{guilt}$. Let $\Pi = (\mathsf{K}, \mathsf{VK}, \mathsf{E}, \mathsf{D})$ be the Paillier Elgamal cryptosystem. Then the NIDVZK argument system for $\mathcal{R}$ of Fig. 5 is ϱ-adaptively computationally culpably sound for $\mathcal{R}^{guilt}$ of Eq. (9) for $\varrho = O(\log \kappa)$, and computationally (resp., statistically) composable zero knowledge for $\mathcal{R}$.*

Proof. Adaptive culpable Soundness. We show that if a cheating prover $\mathcal{A}_{zk}$ returns a good challenge e' for the NIDVZK argument system with some probability $\varepsilon = \delta$, then we can break the message recovery security of Π with probability $\varepsilon_\pi = 1/(\varrho 2^\varrho)\delta$.

For this, we note that $\mathcal{A}_{zk}$ gets information about e from two sources, from $\boldsymbol{c}_e$ and from the response of the verifier to different queries. We now construct an adversary $\mathcal{A}_\pi$ that, given access to $\mathcal{A}_{zk}$, breaks the message recovery security of Π (where the public key $\mathcal{Z}.\mathsf{pk}$ includes $\boldsymbol{c}_e$). It uses the extractor $\mathcal{S}.\mathsf{EX}$, who — given that the prover is dishonest and such a challenge exists — returns the good challenge e'.

First, the challenger uses $\mathcal{Z}.\mathsf{G}(1^\kappa)$ to generate a secret key $\mathcal{Z}.\mathsf{sk} = (\mathsf{sk}_e, e)$ and a public key $\mathcal{Z}.\mathsf{pk} = (\mathsf{pk}_e, \boldsymbol{c}_e)$, and sends $\mathcal{Z}.\mathsf{pk}$ to $\mathcal{A}_\pi$. $\mathcal{A}_\pi$ then runs $\mathcal{A}_{zk}^{\mathcal{Z}.\mathsf{V}(\mathcal{Z}.\mathsf{sk};\cdot,\cdot)}(\mathcal{Z}.\mathsf{pk})$. Assume $\mathcal{A}_{zk}$ replies with a tuple (x_i, π_i, w_i). Since $\mathcal{A}_{zk}$ is successful, $\mathcal{A}_\pi$ emulates the verifier by replying with a random bit b. Once $\mathcal{A}_{zk}$ stops (say after $\varrho = \Theta(\log \kappa)$ steps), $\mathcal{A}_\pi$ chooses uniformly one tuple $(x_{i_0}, \pi_{i_0}, w_{i_0})$, and then runs the extractor with the input (x_{i_0}, w_{i_0}), and obtains either "accept", or a candidate challenge e'. Then, $\mathcal{A}_\pi$ outputs what the extractor outputs.

With probability $2^{-\varrho} = 2^{-\Theta(\log \kappa)} = \kappa^{-\Theta(1)}$, all bits that $\mathcal{A}_\pi$ chose are equal to the bits that the verifier would have sent. Since $\mathcal{A}_{zk}$ is successful, then with a non-negligible probability, one of the input/argument tuples, say $(x_{i_1}, \pi_{i_1}, w_{i_1})$, is such that $(x_{i_1}, w_{i_1}) \in \mathcal{R}^{guilt}$ but the verifier accepts. With probability $1/\varrho = \Theta(1/\log \kappa)$, $i_0 = i_1$. Thus, with probability $\varepsilon_\pi = \frac{\delta}{\varrho 2^\varrho} = \kappa^{-\Theta(1)}$, $\mathcal{A}_\pi$ has given to the extractor an input $(x_{i_0}, w_{i_0}) \in \mathcal{R}^{guilt}$ such that there exists π_{i_0} such that the verifier accepts $(x_{i_0}, \pi_{i_0}, w_{i_0})$. With such inputs, since the verifier accepts, there exists a good challenge e', and the extractor outputs it. In this case, $\mathcal{A}_\pi$ has returned a good e'.

Finally, if the verifier accepts then due to the optimal culpable soundness, the value e' returned by the extractor must be equal to the value e that has been encrypted by $\boldsymbol{c}_e$. Since the only information that $\mathcal{A}_\pi$ has about e is given in $\boldsymbol{c}_e$ (since $\mathcal{A}_\pi$'s random answers do not reveal anything), this means that $\mathcal{A}_\pi$ has returned the plaintext of $\boldsymbol{c}_e$ with non-negligible probability, and thus break the message recovery security of Π.

Composable Zero Knowledge. Assume that $(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk}) \leftarrow \mathcal{Z}.\mathsf{G}(1^\kappa)$, and $(x, w) \leftarrow \mathcal{A}(\mathcal{Z}.\mathsf{sk}, \mathcal{Z}.\mathsf{pk})$. The simulator $\mathcal{Z}.\mathsf{sim}(\mathcal{Z}.\mathsf{sk}, x)$ can obtain e from $\boldsymbol{c}_e$ by decrypting it. Given e, he runs $\mathcal{S}.\mathsf{sim}(x, e)$ to obtain an accepting view $(\boldsymbol{c}_a, e, \boldsymbol{z})$. He then computes $\boldsymbol{c}_z \leftarrow \mathsf{E}_{\mathsf{pk}_e}(\boldsymbol{z})$ and returns $\pi \leftarrow (\boldsymbol{c}_a, \boldsymbol{c}_z)$.

We now show that the transcript comes from a distribution that is indistinguishable from that of the real view. Cnsider the following hybrid simulator $\mathcal{Z}.\mathsf{sim}^w$ that gets the witness w as part of the input. $\mathcal{Z}.\mathsf{sim}^w$ does the following:

1. Create $(\boldsymbol{c}_a, \boldsymbol{z}_1, \boldsymbol{z}_2) \leftarrow \mathcal{S}.\mathsf{P}(x, w)$ and the Σ protocol transcript $(\boldsymbol{c}_a, e, \boldsymbol{z})$, $\boldsymbol{z} \leftarrow e\boldsymbol{z_1} + \boldsymbol{z_2}$, by following the Σ-protocol.
2. Encrypt $\boldsymbol{z}$ component-wise to get $\boldsymbol{c}_z$.
3. Return $\pi \leftarrow (\boldsymbol{c}_a, \boldsymbol{c}_z)$

Since the encryption scheme is blindable, such a hybrid argument is perfectly indistinguishable from the real argument. Since the Σ-protocol is specially HVZK, hybrid arguments and simulated arguments are computationally indistinguishable. If the Σ-protocol is statistically specially HVZK, then hybrid arguments and simulated arguments (and thus also real arguments and simulated arguments) are statistically indistinguishable.

□