# On the CCA security properties (and more) of a new variant of Paillier-ElGamal* **

Duong Hieu Phan[1], Renaud Sirdey[2] and Jean Vacher[2]

[1] Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France,
hieu.phan@telecom-paris.fr,
[2] Université Paris-Saclay, CEA, List, Palaiseau, France,
renaud.sirdey@cea.fr, jean.vacher@cea.fr

**Abstract.** We solve the long-standing open question of designing a "truly" linearly homomorphic scheme – meaning it supports homomorphic additions on arbitrary plaintexts, with no restriction, in contrast to "somewhat" ones – that achieves CCA1 security under a standard assumption. We do so by introducing a new variant of Paillier-ElGamal, which we call Damgård-Paillier-ElGamal (DPEG) as its design follows a Knowledge-of-Exponent pattern. On top of being linearly homomorphic without any restriction, our scheme enjoys the following properties:

- It achieves CCA1 security solely under the DCR assumption. To the best of our knowledge, it is the first "truly" linearly homomorphic scheme proven CCA1-secure solely under this assumption (or any other standard one).
- It can be extended to support one level of multiplication while still preserving its CCA1 security under the same assumption. This extension is then the first concrete scheme supporting both homomorphic additions and multiplications (even limited to one level) that is proven CCA1-secure under DCR.
- It also achieves Manulis&Nguyen's stronger notion of vCCA security under an additional non-falsifiable linear-only homomorphism assumption that is commonly used in proof-of-knowledge constructs. DPEG is then the first scheme that is proven vCCA secure while being CCA1-secure under a standard assumption. This also carries over to the multiplicative extension.

Interestingly, DPEG achieves the above at only 1.5 times the cost of the baseline CPA-secure Paillier-ElGamal scheme.

To establish the CCA1 security of DPEG, we introduce a new abstract framework that allows us to prove CCA1 security of a large class of group-based PKE that also covers other *somewhat* linearly homomorphic schemes previously known to achieve CCA1 security under falsifiable assumptions such as Damgård-ElGamal, Cramer-Shoup-Lite and the recent variant of Paillier-ElGamal with plaintext zero padding of Libert. This framework may be of independent interest to more easily prove the CCA1 security of other schemes.

* This work was supported by the France 2030 ANR Projects ANR-22-PECY-003 SecureCompute.

** This paper has been published in the proceedings of ASIACRYPT'26 (this is the extended version).

Lastly, on the negative side, we take a first step in connecting vCCA security to an impossibility result of Gentry&Wichs and show that, under mild assumptions, the vCCA security of DPEG cannot be established from any falsifiable assumption by means of a black-box reduction.

## 1 Introduction

"Beyond CPA" security for homomorphic schemes has been the subject of intense research over recent years, with both theoretical advances (new security notions, impossibility results, or generic constructions) and more concrete constructions. Interestingly, there have been advances for both lattice-based and number-theoretic homomorphic schemes. However, going beyond CPA security often breaks algebraic (homomorphic) properties, typically by adding validity proofs in CCA1-secure public-key encryption (PKE) or even SNARKs in "beyond-CCA1"-secure fully homomorphic encryption (FHE). Even for linearly homomorphic encryption, which is already known to have significant practical applications, achieving security beyond CPA remains difficult, with no known techniques achieving CCA1 security under standard or falsifiable assumptions while *fully* preserving the homomorphic property.

Interestingly, several breakthroughs have recently occurred in the more "old-school" realm of number-theoretic linearly homomorphic encryption schemes, with several long-standing open questions finding either negative or positive answers. On the negative side, a recent impossibility result by S. Schäge [40] negatively answered the long-standing open question of the CCA1 security of ElGamal and Paillier under falsifiable assumptions [31] by showing that no black-box reduction can establish CCA1 security for a class of group-based linearly homomorphic schemes, termed SHPKE ("SH" stands for "semi-homomorphic" in Schäge's terminology, to distinguish it from FHE), under such an assumption. This impossibility result was later extended to cover the weaker notion of PCA1 security in [41]. On the positive side, Libert [27] recently achieved the tour de force of proving the CCA1 security of the Damgård-ElGamal scheme [18] solely under the standard DDH assumption, while previous proofs required a non-falsifiable Knowledge-of-Exponent assumption. Interestingly, [27] also provides proofs of CCA1 security, again under (almost) standard assumptions, for variants of BGN and Paillier-ElGamal with sparsified ciphertext domains. However, these schemes achieve this sparsification by forcing $\lambda$ plaintext bits to be zeroes and, as a result, are "only" *somewhat* linearly homomorphic: when a carry propagates into the zero padding during a homomorphic addition the resulting ciphertext decrypts to $\perp$ rather than the sum of the plaintexts of the two input ciphertexts. Beyond CCA1, [16] also investigates the connection between the non-falsifiable Linear-Only Homomorphism (LOH) assumption of Bitansky et al. [3,4] and the notion of vCCA security recently introduced by Manulis & Nguyen [30], and establishes the general result that any correct CPA-secure linearly homomorphic scheme with the LOH property achieves vCCA security. This yielded a first simple and practical construction, based on the Paillier scheme, that achieves

vCCA security at only twice the cost of achieving CPA security, albeit under a non-falsifiable assumption.

On the more theoretical side, it may be interesting to highlight that the large majority of works investigating (F)HE constructions achieving some degree of CCA security [7,13,29,30,10] require either the random oracle model or some non-falsifiable assumption due to their reliance on general $zk$-SNARKs [25] or other strong primitives such as iO. This is with the exception of the recent (Crypto'25) paper of Yang et al. [42] which achieves the tour de force of giving a construction that is IV-CCA secure (a stronger-than-CCA1 notion they define and study) in the standard model, based solely on the LWE assumption. On the downside, their construction is intrinsically *non-compact*. The compactness requirement which is naturally expected from FHE, is however achieved by some of the construction blueprints that are proposed in [30,10] (namely Encrypt-then-Sign in [30] and Encrypt-then-Prove in [10]). However, these generic constructions are not easily amenable to efficient implementations in their full generality, essentially because they intimately require powerful SNARK machinery.

### 1.1 Our Objective

In this paper, we follow the former line of work on designing simple, elegant linearly homomorphic CCA1-secure encryption schemes and complete it by proposing one that achieves *truly* linearly homomorphic functionality, without any validity conditions that would make the schemes only *somewhat* linearly homomorphic.

The intuition behind our new scheme is very similar to Bitansky et al.'s two-ciphertext Paillier heuristic that has already been considered in several works [3,2,4,23,16,21]: encrypt a message $m$ as a pair of Paillier ciphertexts $(\mathsf{Enc}(m), \mathsf{Enc}(\xi \cdot m))$ under the same key material, for a secret random value $\xi \in \mathbb{Z}_N$, and check this linear relation at decryption.

Inspired from this idea, we thus propose a two-message-slot variant of the Paillier-ElGamal scheme in which the two message slots are under independent secret keys. Because of this we can reuse the randomness across these two slots. We then sparsify the plaintext domain of the resulting scheme following Damgård's Knowledge-of-Exponent template, i.e. encrypt $m$ in the first slot and $\xi \cdot m$ in the second. This yields a compact linearly homomorphic scheme which we call Damgård-Paillier-ElGamal (DPEG) as its design is similar in spirit to the Damgård-ElGamal (DEG) scheme. However, contrary to DEG, DPEG does not induce any restriction in terms of linear homomorphism properties. Interestingly, because of this, DPEG can further be extended to support one level of multiplication through the Catalano-Fiore transform [15]. We then establish that the CCA1 security of DPEG holds solely under the DCR assumption. To obtain this result, we introduce an abstract framework allowing us to establish the CCA1 security of a relatively large class of group-based linearly homomorphic schemes, under falsifiable indistinguishability assumptions.

To define this framework, our starting point was the Semi-Homomorphic PKE (SHPKE) class of schemes introduced by S. Schäge to prove his first im-

possibility result [40]. In a nutshell, this class includes group-based, perfectly correct, linearly homomorphic encryption schemes for which the image of the encryption function defines a group $(H, \cdot) \subseteq \mathcal{C}$, with $\mathcal{C}$ the ciphertext domain of the scheme. Then, the key property is that $H$ is required to have an efficient public membership test or, equivalently, that valid ciphertexts under an SHPKE can be publicly efficiently recognized. On the other hand, B. Libert's paper [27] shows that several group-based schemes for which *valid ciphertexts cannot be efficiently recognized publicly*, and which therefore fall outside the SHPKE class, can be proven CCA1-secure under falsifiable assumptions, using relatively similar proof patterns. This naturally raises the question of defining an SHPKE-like class of schemes such that all schemes in this class can be proven CCA1-secure under falsifiable assumptions. We address it by defining a class of schemes obtained by *sparsifying* the ciphertext domain of any group-based linearly homomorphic scheme for which the image of the encryption function defines a group $(H, \cdot)$ that is embedded sparsely in a larger group $(H^*, \cdot) \subseteq \mathcal{C}$ such that random elements in $H$ are computationally indistinguishable from random elements in $H^*$, and thus membership in $H$ cannot be publicly verifiable. We call this class sGPKE for *sparsified Group-based (linearly homomorphic) PKE* and, under certain generic conditions, we show that all schemes in this class are CCA1-secure. Interestingly, the sGPKE class covers a number of previously known schemes (Damgård-ElGamal, Cramer-Shoup-Lite and Libert's variant of Paillier-ElGamal with plaintext zero-padding). We show that it also covers our new DPEG scheme and, thus, its CCA1 security follows.

Beyond CCA1, we show that our DPEG scheme also achieves the stronger notion of vCCA security under an additional previously known Linear-Only Homomorphism assumption that is used in several mainstream proof-of-knowledge constructions. While this assumption is non-falsifiable and quite strong, we finally rule out that DPEG's vCCA security can be proved by a black-box reduction to any falsifiable assumption without contradicting the no-go theorem of Gentry&Wichs [25] on the existence of SNARGs provably secure under falsifiable assumptions.

The next section summarizes our contributions in more details.

### 1.2 Summary of contributions

The contributions of this paper are as follows:

- *Our first main contribution* is to design a new variant of Paillier-ElGamal with two message slots, DPEG, that implements the Bitansky et al.'s two-ciphertext blueprint [3,4]: under DPEG, when the first message slot in a ciphertext contains a value $m$, the second message slot is expected to contain $\xi m$, for a secret integer $\xi$, and this linear relation is checked upon decryption. Interestingly DPEG is "truly" (i.e. "non-somewhat") linearly homomorphic. We show that DPEG is CCA1-secure under the sole DCR assumption. To the best of our knowledge, *DPEG is the first truly linearly homomorphic scheme that is proven CCA1-secure under a standard assumption* (closing an open

problem from [21] which also describes interesting use-cases for such kind of schemes). Furthermore, DPEG achieves CCA1 security at only 1.5 times the cost of the CPA-secure baseline Paillier-ElGamal scheme on which it is based.

- Complementarily, we show that DPEG achieves the stronger notion of vCCA security recently introduced by Manulis&Nguyen under the *non-falsifiable* assumption that it has the Linear-Only Homomorphism (LOH) property of Bitansky et al. [3,4]. As such, DPEG is the *first linearly homomorphic scheme that achieves vCCA security albeit under a non-falsifiable assumption while being CCA1-secure under a standard assumption*, while previous constructions [16,21] required the non-falsifiable assumption to close the gap from CPA to vCCA security rather than only from CCA1 to vCCA.
- Additionally, we also leverage on the Catalano-Fiore blueprint [15] to get one level of homomorphic multiplication from DPEG. In doing so and to the best of our knowledge, we obtain *the first concrete scheme supporting both homomorphic additions and multiplications (even limited to one level) that is proven CCA1-secure under a standard assumption*. Furthermore, this scheme inherits its vCCA security from that of DPEG.
- *As our second main contribution*, we define the class of Group-based linearly homomorphic PKE, GPKE. In essence, for this class of schemes, the image of the encryption function defines a group $(H, \cdot)$ embedded in a larger group $(H^*, \cdot)$ such that uniformly picked elements of $H$ and $H^*$ are computationally indistinguishable.
- We then define the class of sparsified GPKE, sGPKE. An sGPKE is built by sparsifying the ciphertext domain of an inner GPKE scheme such that the plaintext domain of the former is $2^\lambda$ times smaller than the latter. The decryption function then returns $\perp$ whenever the decryption function of the inner scheme returns an element that is not in the smaller plaintext domain.
- We then prove that *any* sGPKE achieves CCA1 security, hence under the assumption under which the computational indistinguishability between random elements of $H$ and $H^*$ holds.
- To illustrate the generality of the sGPKE framework, we show that DPEG as well as several other known schemes satisfy the definition. These include Paillier-ElGamal with plaintext zero-padding [27] (under DCR), Damgård-ElGamal [18,27] (under DDH) and Cramer-Shoup Lite (under DDH). All these schemes were already known to achieve CCA1 security under the same assumptions via different proofs, hence our framework paves the way for more easily proving the CCA1 security of new schemes. Interestingly, Paillier-ElGamal with plaintext zero-padding was only known to achieve CCA1 based solely under DCR by sacrificing 50% of the plaintext size. Our framework establishes that its CCA1 security holds solely under DCR without this restriction, closing an open question from [27]. We also emphasize that for reasons specific to each of them, all the above schemes are only *somewhat* linearly homomorphic.
- As a last bonus contribution, we start establishing a formal connection between vCCA security and Gentry&Wichs' no-go theorem on the existence of

SNARGs that can be proven secure under falsifiable assumptions [25] by establishing that DPEG cannot have a black-box proof of vCCA security based on any such assumption, without contradicting the latter. In the sense that it achieves CCA1 security under a falsifiable assumption and then vCCA security under an additional non-falsifiable one, DPEG thus attains some form of "optimality" regarding the falsifiable nature of the assumptions under which its security properties hold.

### 1.3 Paper organization

This paper is organized as follows. Sect. 2 covers the preliminaries. Sect. 3 introduces our new variant of Paillier-ElGamal, DPEG, establishes its first properties, investigates its vCCA security under the LOH assumption and further studies its Catalano-Fiore-style multiplicative variant and its security. Then, to establish the CCA1 security under DCR of DPEG, Sect. 4 gives the definition of the class of sGPKE schemes, proves our main result on their CCA1 security and also establishes that DPEG and several other known schemes fall into the class. Lastly, in Sect. 5, we investigate the connection between the vCCA security of DPEG and Gentry&Wichs' theorem.

## 2 Preliminaries

Given $l, u \in \mathbb{Z}$, we use $[\![l, u]\!]$ to denote the set $\{l, l+1, ..., u-1, u\}$. Reduction modulo $q$ is denoted as $[.]_q$. We use this notation explicitly only when it avoids possible ambiguities.

### 2.1 Basic definitions

We define a public-key encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ over key space $\mathcal{K}$, plaintext domain $\mathcal{P}$ and ciphertext domain $\mathcal{C}$ as a triplet of PPT algorithms:

- KeyGen: on input $1^\lambda$, outputs an encryption key pk and a decryption key sk. We then call $(\mathsf{pk}, \mathsf{sk})$ a *valid* key pair.
- Enc: on inputs pk and $m \in \mathcal{P}$, outputs an encryption $c \in \mathcal{C}$ of $m$.
- Dec: on inputs sk and $c \in \mathcal{C}$, outputs a decryption $m \in \mathcal{P} \cup \{\perp\}$ of $c$.

Let COIN denote the randomness space of $\mathcal{E}$. We sometimes externalize the randomness used in the encryption function by means of the notation $\mathsf{Enc}(\mathsf{pk}, m; r)$, with $m \in \mathcal{P}$ and $r$ is drawn from a preset distribution, $\chi$, over COIN. In this latter case, the function $\mathsf{Enc} : \mathcal{P} \times \mathsf{COIN} \longrightarrow \mathcal{C}$ is deterministic.

When, for all $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}$, all $m \in \mathcal{P}$ and all $r \in \mathsf{COIN}$, we have that,

$$\mathcal{E}.\mathsf{Dec}(\mathsf{sk}, \mathcal{E}.\mathsf{Enc}(\mathsf{pk}, m; r)) = m, \tag{1}$$

we say that $\mathcal{E}$ is *perfectly correct* or simply *correct*[1].

[1] In this paper we do not consider more relaxed notions such as statistical correctness where equality (1) holds with overwhelming probability over the encryption randomness or others [20].

Let $\mathcal{O}$ denote a set of operations and let $\mathcal{F}(\lambda)$ denote the set of circuits over $\mathcal{O}$ of polynomially bounded size in the security parameter $\lambda$. Given a function class $\mathcal{F}(\lambda)$, we define a homomorphic encryption (HE) scheme, $\mathcal{E}_H$, over $\mathcal{F}(\lambda)$ as an encryption scheme augmented by a PPT algorithm $\mathsf{Eval}$ which, on input $f \in \mathcal{F}(\lambda)$ and $c_0, ..., c_{L-1} \in \mathcal{C}^L$, where $L$ denotes the arity of function $f$, outputs a new *evaluated* ciphertext. When $\mathcal{E}_H$ satisfies condition (1) and when $\mathsf{Eval}$ is such that for all $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}$, all $f \in \mathcal{F}(\lambda)$, all $m_0, ..., m_{L-1} \in \mathcal{P}^L$ and all $r_0, ..., r_{L-1} \in \mathsf{COIN}^L$,

$$\mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c_e) = f(m_0, ..., m_{L-1}), \tag{2}$$

with $c_e = \mathcal{E}_H.\mathsf{Eval}(\mathsf{pk}, \mathcal{E}_H.\mathsf{Enc}(\mathsf{pk}, m_0; r_0), ..., \mathcal{E}_H.\mathsf{Enc}(\mathsf{pk}, m_{L-1}; r_{L-1}))$, we also say that $\mathcal{E}_H$ is a *perfectly correct* or *correct* HE scheme or that $\mathcal{E}$ is homomorphic over $\mathcal{F}(\lambda)$. When this property is satisfied only under additional conditions, we say that $\mathcal{E}$ is *somewhat* homomorphic over $\mathcal{F}(\lambda)$. We will sometimes also use the prefix *truly* homomorphic when we want to put an emphasis on the fact that a given scheme is *not* somewhat homomorphic.

Lastly, when $|c_e| \leq p(\lambda)$ for some fixed polynomial $p$ and, in particular, is independent of $|f|$ and $L$, we say that $\mathcal{E}_H$ is *compact*.

### 2.2 Security notions

Unless otherwise stated, we consider only (single-challenge) indistinguishability-based security notions and thus omit the IND prefix. In all the security games we consider, the adversary has to guess with non-negligible advantage a random bit $\gamma$, privately chosen by its challenger, given some oracles including at least a challenge oracle. Given two distinct messages $m_0$ and $m_1$, the challenge oracle returns an encryption $c^*$ of $m_\gamma$.

#### CPA, CCA1 and CCA2 security.

**Definition 1 (Generic indistinguishability experiment).** *Given a public-key encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, [...])$ and a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with state $S$, we define the following generic indistinguishability experiment, $\mathsf{Exp}_{\mathcal{E}}^{\mathcal{O}_1, \mathcal{O}_2}(\mathcal{A})$,*

$$\begin{aligned}
&(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{E}.\mathsf{KeyGen}(1^\lambda),\\
&(m_0, m_1, S) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(\mathsf{pk}),\\
&\gamma \xleftarrow{\$} \{0,1\},\\
&c^* \leftarrow \mathcal{E}.\mathsf{Enc}(\mathsf{pk}, m_\gamma),\\
&\gamma' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(\mathsf{pk}, S, c^*),\\
&\gamma \stackrel{?}{=} \gamma',
\end{aligned}$$

*where $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively denote first and second step decryption oracles and with the oracle $\mathcal{E}.\mathsf{Enc}(\mathsf{pk}, .)$ being provided only in the private key case.*

Then, for a PPT adversary, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the CPA and CCA1 experiments, respectively $\mathsf{Exp}_{\mathcal{E}}^{\mathrm{CPA}}(\mathcal{A})$ and $\mathsf{Exp}_{\mathcal{E}}^{\mathrm{CCA1}}(\mathcal{A})$, respectively correspond to $\mathsf{Exp}_{\mathcal{E}}^{\cdot,\cdot}(\mathcal{A})$ and $\mathsf{Exp}_{\mathcal{E}}^{\mathcal{E}.\mathsf{Dec}(\mathsf{sk},.),\cdot}(\mathcal{A})$. The CCA2 experiment is also defined as,

$$\mathsf{Exp}_{\mathcal{E}}^{\mathrm{CCA2}}(\mathcal{A}) \equiv \mathsf{Exp}_{\mathcal{E}}^{\mathcal{E}.\mathsf{Dec}(\mathsf{sk},.),\mathcal{O}_{\mathrm{CCA2}}(\mathsf{sk},c^*,.)}(\mathcal{A}),$$

with,

$$\mathcal{O}_{\mathrm{CCA2}}(\mathsf{sk}, c^*, c) = \begin{cases} \mathcal{E}.\mathsf{Dec}(\mathsf{sk}, c) & \text{if } c^* \neq c, \\ \perp & \text{otherwise.} \end{cases}$$

A scheme $\mathcal{E}$ is then said to be CPA-secure if for any PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{E}}^{\mathrm{CPA}}(\mathcal{A}) = \left| P\left(\mathsf{Exp}_{\mathcal{E}}^{\mathrm{CPA}}(\mathcal{A}) = \mathsf{True}\right) - 1/2 \right| \leq \mathrm{neg}(\lambda),$$

where $\lambda$ is the security parameter, and likewise for CCA1 and CCA2.

Note that some cryptographers consider that CCA1 security is a historical security notion from the time before the first CCA2 public encryption schemes were proposed. While this is a legitimate point of view, CCA1 may still be considered a meaningful CCA security target for homomorphic schemes in order to circumvent the CCA2 impossibility to which malleable schemes are subject [29,14].

**vCCA (and IV-CCA) security.** As introduced in [30], vCCA security is a "CCA2-style" security notion for FHE. As such, the vCCA game has two decryption oracles where the second step (post-challenge generation) oracle assumes the existence of a PPT witness extractor[2] defined as follows.

**Definition 2 ($\mathcal{H}$-extraction algorithm [30]).** *Given an FHE scheme $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with plaintext domain $\mathcal{P}$, ciphertext domain $\mathcal{C}$ and function domain $\mathcal{F}(\lambda)$, let $\mathcal{H} \subseteq \mathcal{F}(\lambda)$, an $\mathcal{H}$-extraction algorithm is a PPT algorithm $\mathsf{extract} : \mathcal{C} \times \Omega \longrightarrow \mathcal{H} \times \mathcal{C}^L \cup \{\perp\}$, where $\Omega$ is a set of auxiliary data and where, for simplicity's sake, all functions in $\mathcal{H}$ are assumed to have the same arity $L$. Additionally, a $\mathcal{H}$-extraction algorithm $\mathsf{extract}$ is sound if for any PPT adversary $\mathcal{A}$, any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{E}_H.\mathsf{KeyGen}(1^\lambda)$ and any $c \leftarrow \mathcal{A}(\mathsf{pk})$, $(f, c_0, ..., c_{L-1}) \leftarrow \mathsf{extract}(c, \mathsf{aux})$ is such that,*

- *$c_i$ is a fresh ciphertext, $\forall i \in [\![0, L-1]\!]$.*

[2] The spirit of the vCCA security notion is (at least) to model construction blueprints which embed proof material in their ciphertexts and rely on a SNARK to enforce correct homomorphic evaluations over some input ciphertexts. In this context, the above $\mathsf{extract}$ thus corresponds to the extractor of that underlying SNARK which allows to retrieve a witness from the proven statement as well as auxiliary data. The nature of the auxiliary data depends on whether extractability is black-box in some ROM [22] or white-box, as is usually the case for knowledge assumptions, see [30] for more details. The spirit of the original vCCA definition was arguably to favor the former option [32].

– *The distribution of* $\mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c)$ *is indistinguishable from that of*

$$f(\mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c_0), ..., \mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c_{L-1})).$$

*With the conventions that,* $f(\mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c_0), ..., \mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c_{L-1})) = \perp$, *if* $\exists i \in [\![0, L-1]\!] : \mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c_i) = \perp$ *and that* $\mathsf{extract}(c, \mathsf{aux}) = (\mathsf{id}, c)$ *for any fresh ciphertext* $c$.

Following this, the vCCA experiment is defined as follows.

**Definition 3 (vCCA experiment [30]).**

$$\mathsf{Exp}_{\mathcal{E}}^{vCCA}(\mathcal{A}) \equiv \mathsf{Exp}_{\mathcal{E}}^{\mathcal{E}.\mathsf{Dec}(\mathsf{sk}, .), \mathcal{O}_{vCCA}(\mathsf{sk}, c^*, .)}(\mathcal{A}),$$

*with, letting* $(f, c_0, ..., c_{L-1}) \leftarrow \mathsf{extract}(c, \mathsf{aux})$,

$$\mathcal{O}_{vCCA}(\mathsf{sk}, c^*, c) = \begin{cases} \mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c) & \text{if } c^* \notin \{c_0, ..., c_{L-1}\}, \\ \perp & \text{otherwise.} \end{cases}$$

**Definition 4 ((IND-)vCCA security [30]).** *A scheme* $\mathcal{E}_H$ *is (IND-)vCCA secure with respect to* $\mathcal{H} \subseteq \mathcal{F}(\lambda)$ *if there exists an* $\mathcal{H}$*-extraction algorithm* $\mathsf{extract}$ *such that for any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, *the advantage of* $\mathcal{A}$ *winning the vCCA experiment is negligible in the security parameter* $\lambda$.

In spirit, the vCCA game is exactly the single challenge CCA2 game, with the second step decryption oracle being augmented in order to filter out *all* byproducts of the challenge ciphertext (rather than just the challenge ciphertext). Most importantly, vCCA security is a milestone notion for FHE as it is the strongest CCA security notion so far known to be achievable by *correct* homomorphic schemes. In particular, it is stronger than several previously introduced notions such as funcCPA [1] and others.

To conclude this section, it is important to also mention the notion of IV-CCA (CCA1≺IV-CCA≺vCCA[3]) security recently introduced in Yang et al. [42]. However, the spirit of this notion is to model *non-compact* FHE. Indeed, in IV-CCA, the second step decryption oracle specification relies on a verification algorithm, which is part of the FHE scheme and which takes as input a ciphertext to be decrypted and *a set of input ciphertexts* that explains the former (via some legit homomorphic evaluation). At this price, they achieve the tour de force of giving a construction that is IV-CCA secure in the standard model, based solely on the LWE assumption. Regarding vCCA, we have used the definition from [30] as is, i.e. with an extractor explicitly included as part of the second-stage decryption oracle. This extractor naturally appears since vCCA security is, in spirit, intended to model FHE constructions that embed SNARKs. To us, it is a very interesting open question to design a security notion between IV-CCA and vCCA that does not involve any extractor in its definition while allowing for compact FHE. This is, however, beyond the scope of the present paper.

[3] The notation ≺ means a strict separation among two notions.

### 2.3 The Linear-Only Homomorphism (LOH) assumption

Given an encryption scheme $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$, let $\mathsf{ImVer}$ denote a function that verifies if a ciphertext is in the image of the encryption function with the knowledge of $\mathsf{sk}$ (it is essentially the "verification" part of $\mathcal{E}_H.\mathsf{Dec}$ and $\mathcal{E}_H.\mathsf{ImVer}(\mathsf{sk}, c) = \mathsf{True} \Leftrightarrow \mathcal{E}_H.\mathsf{Dec}(\mathsf{sk}, c) \neq \perp$). Then, informally, the Linear-Only Homomorphism (LOH) property [3,4] states that given polynomially many ciphertexts $(c_0, ..., c_{m-1})$ under $\mathcal{E}_H$ it is infeasible for an adversary to create a new ciphertext $c'$, which is in the image of the encryption function (as verified by $\mathsf{ImVer}$) and cannot be expressed by (homomorphically) evaluating an affine combination of the ciphertexts in the previous list. The LOH property has been introduced in [3] to serve as the basis for several SNARK constructions in that paper and other subsequent works [6,24,33].

Formally, following [3,4], we have the definition below.

**Definition 5 (LOH property, reproduced from [3]).** *An encryption scheme $\mathcal{E}_H = (\mathit{KeyGen}, \mathit{Enc}, \mathit{ImVer}, \mathit{Dec}, \mathit{Eval})$ (with $\mathcal{P} \cong \mathbb{Z}_t$) satisfies the* Linear-only Homomorphism property *if for every PPT adversary $\mathcal{A}$, there is a PPT extractor $\mathit{extract}$ such that for any auxiliary input $\mathsf{aux} \in \{0,1\}^{poly(\lambda)}$ and any plaintext generator $\mathcal{M}$,*

$$P\left(\begin{array}{l|l} & (\mathit{pk}, \mathit{sk}) := \mathcal{E}_H.\mathit{KeyGen}(1^\lambda) \\ \exists i \in [\![0, k-1]\!] \ \textit{s. t.} & (a_0, ..., a_{m-1}) := \mathcal{M}([\mathit{pk}]) \\ \mathit{ImVer}(\mathit{sk}, c'_i) = \mathit{True} & (c_0, ..., c_{m-1}) := (\mathcal{E}_H.\mathit{Enc}(\mathit{pk}, a_0), ..., \mathcal{E}_H.\mathit{Enc}(\mathit{pk}, a_{m-1})) \\ \textit{and} & (c'_0, ..., c'_{k-1}) := \mathcal{A}(c_0, ..., c_{m-1}, \mathit{pk}; \mathit{aux}) \\ \mathcal{E}_H.\mathit{Dec}(\mathit{sk}, c'_i) \neq a'_i & (\Pi, b) := \mathit{extract}(c_0, ..., c_{m-1}, \mathit{pk}; \mathit{aux}) \\ & (a'_0, ..., a'_{k-1})^T := \Pi \cdot (a_0, ..., a_{m-1})^T + b^T \end{array}\right) \leq neg(\lambda). \tag{3}$$

*where $\Pi \in \mathbb{Z}_t^{k \times m}$ and $b \in \mathbb{Z}_t^k$, and with the convention that row $i$ of $\Pi$ is left empty (i.e. $(\Pi_i, b_i) = \varnothing$) and $a'_i = \perp$ when $c'_i$ was not generated by homomorphically evaluating an affine combination over the $c_j$'s.*

Remark that the $c'_i$ do not have to be explicitly included as inputs to the extractor since they are built by the adversary and therefore implicitly recoverable from $\mathsf{aux}$. In summary, whenever $\mathcal{A}$ builds $c'_i$ by doing "something equivalent to",

$$c'_i = \mathcal{E}_H.\mathsf{Eval}(\mathtt{add}, \mathcal{E}_H.\mathsf{Eval}(\mathtt{lincomb}_\pi, c_0, ..., c_{m-1}), \mathcal{E}_H.\mathsf{Eval}(\mathtt{lincomb}_{\pi'}, c''_0, ..., c''_{l-1})), \tag{4}$$

where $c''_0 = \mathcal{E}_H.\mathsf{Enc}(\mathsf{pk}, \mu_0), ..., c''_{l-1} = \mathcal{E}_H.\mathsf{Enc}(\mathsf{pk}, \mu_{l-1})$, then $\Pi_i = \pi$ and $b_i = \sum_{j=0}^{l-1} \pi'_j \mu_j$. Conversely, $(\Pi_i, b_i) = \varnothing$ when this is not the case.

In our security proofs, we will use the following more convenient one-ciphertext notation for the above extractor,

$$(\pi, \beta) = \mathsf{extract}(c, \mathsf{aux}), \tag{5}$$

as a shortcut for $\Pi_{i:c'_i=c}$ and $b_{i:c'_i=c}$ (i.e. $\pi$, respectively $\beta$, is the row of $\Pi$, respectively the component of $b$, associated to ciphertext $c$). While a more rigorous notation would be to use $(\pi, \beta) = \mathsf{extract}(c, \{c_0, ..., c_{m-1}\}, \mathsf{pk}; \mathsf{aux})$, we use Eq. (5)

in the sequel as the set of ciphertexts given to the LOH adversary will always be clear from the nearby context. Remark that affine rather than linear combinations are considered in the above definition to account for the fact that, in the public-key setting, the adversary can create (from scratch) additional fresh well-formed ciphertexts on its own and homomorphically add them to homomorphic evaluations of linear combinations over the $c_i$'s.

On top of the above definition, [3] further proposes several heuristic approaches to build schemes satisfying the LOH property, starting from a *correct* CPA-secure linear homomorphic scheme. An example, which has been considered in several works [3,2,4,23,16,21], is the "two-ciphertexts" blueprint which consists in sparsifying the ciphertext domain of a correct linearly homomorphic scheme (e.g. Paillier) by following a Knowledge-of-Exponent (KEA) template[4]. To illustrate this approach with the Paillier scheme, one may consider that the encryption of a message $m$ consists of a pair of ciphertexts $(\mathsf{Enc}(m), \mathsf{Enc}(\xi \cdot m))$ under the same key material, for a fixed secret random value $\xi \in \mathbb{Z}_n$ (with $n$ the RSA modulus of the scheme), and with the $\mathsf{ImVer}$ algorithm checking this linear relation by decrypting both ciphertexts. It is then assumed that this scheme satisfies Definition 5 as the underlying Paillier scheme only exhibits linear homomorphic properties (to the best of the research community's knowledge), although the baseline Paillier scheme does not have the LOH property as also argued in [3] ([2], p. 33). Interestingly, this construction has recently been proven to achieve vCCA security in [16,21].

### 2.4 Paillier-ElGamal

In this section, we specify the Paillier-ElGamal scheme that is used as a starting point in this work. This scheme originates in the Camenisch-Shoup scheme [11] (which is CCA2 under DCR and, as such, non-malleable) as it is essentially equivalent to a variant of the latter in which the last out of three slots is removed. We also stress that there are several CPA-secure schemes referred to as Paillier-ElGamal in the literature, e.g. [35,39,9,5], and they are not all equivalent. So, following [27], the baseline Paillier-ElGamal scheme (which may thus also be referred to as the Camenisch-Shoup Lite scheme) is defined as follows:

- $\mathsf{PEG.KeyGen}$: choose a RSA modulus $N = (2p'+1)(2q'+1)$, with $p'$ and $q'$ two distinct Sophie Germain primes of suitable size. Then, uniformly pick $g \in \mathbb{Z}^*_{N^2}$, $x \in [\![0, N(N-1)/4]\!]$ and let $h = [g^{4Nx}]_{N^2}$. Then, let $\mathsf{pk} = (N, g, h)$ and $\mathsf{sk} = x$. Remark that $(p', q')$ (that will later be denoted $\mathsf{sk}_0$) is not required to decrypt.
- $\mathsf{PEG.Enc}$: given $\mathsf{pk}$ and $m \in \mathbb{Z}_N$, pick $r \in [\![0, (N-1)/4]\!]$ and return $(c_0, c_1) \in \mathbb{Z}^2_{N^2}$ such that,

$$c_0 = [g^{2Nr}]_{N^2}, \text{ and } c_1 = [(1+N)^m h^r]_{N^2}.$$

[4] This is a slight abuse of terminology. Yet, we will use it in this paper as the "two-ciphertexts" blueprint is a distant sibling of Damgård's original heuristic [18] and similar in spirit.

– PEG.Dec: given sk and $(c_0, c_1) \in \mathbb{Z}_{N^2}^2$, let $\mu = [c_1 \cdot c_0^{-2x}]_{N^2}$. If $\mu$ is of the form $[(1+N)^m]_{N^2}$ for some $m \in \mathbb{Z}_N$, then return $m$ and, otherwise, return $\perp$.

Remark that computing $m$ in PEG.Dec may be done easily since $[(1+N)^m]_{N^2} = 1 + mN$. Hence, PEG.Dec can be equivalently written as returning $\perp$ when $[\mu - 1]_N \neq 0$ and $\frac{\mu-1}{N}$ otherwise.

This scheme is perfectly correct and linearly homomorphic offering operators:

– PEG.Add: given $(c_0, c_1)$ and $(c_0', c_1')$, both in $\mathbb{Z}_{N^2}^2$, return $([c_0 c_0']_{N^2}, [c_1 c_1']_{N^2})$.
– $\mathsf{PEG.MulByConst}_\alpha$: given $(c_0, c_1) \in \mathbb{Z}_{N^2}^2$ and $\alpha \in \mathbb{Z}_N$, return $([c_0^\alpha]_{N^2}, [c_1^\alpha]_{N^2})$.

It is well-known that it is CPA-secure under the standard DCR assumption.

### 2.5 Paillier-ElGamal with plaintext zero padding

In [27], the following variant of PEG is considered. Let $2^\lambda \leq B = 2^t < \lfloor 2^{-\lambda} N \rfloor$,

– $\mathsf{PEG}_z$.KeyGen: run $\mathsf{PEG.KeyGen}(1^\lambda)$ to get $(\mathsf{pk}, \mathsf{sk})$.
– $\mathsf{PEG}_z$.Enc: given pk and $m \in [\![0, B]\!]$, return $\mathsf{PEG.Enc}(\mathsf{pk}, m)$.
– $\mathsf{PEG}_z$.Dec: given sk and $\mathsf{ct} \in \mathbb{Z}_{N^2}^2$, if $\mathsf{PEG.Dec}(\mathsf{sk}, \mathsf{ct}) = \perp$, then return $\perp$. Otherwise, let $m = \mathsf{PEG.Dec}(\mathsf{sk}, \mathsf{ct})$, if $m > B$ then return $\perp$ and otherwise return $m$.

This scheme is only *somewhat* linearly homomorphic as it is no more correct:

$$\mathsf{PEG}_z.\mathsf{Dec}(\mathsf{sk}, \mathsf{PEG}_z.\mathsf{Add}(\mathsf{PEG}_z.\mathsf{Enc}(\mathsf{pk}, B), \mathsf{PEG}_z.\mathsf{Enc}(\mathsf{pk}, 1))) = \perp \neq B + 1.$$

Following [27, Theorem 3], it is now known that this scheme is CCA1-secure under the DCR assumption (see Def. 36 in Appendix D.1) and an additional Composite Non-Invertibility assumption (see Def. 37 in Appendix D.1) which, although less standard than the former, is falsifiable[5].

We also reproduce the following lemma from [27] that we use several times.

**Lemma 6 (Lemma 3 in [27]).** *Let $0 < \ell < L$ be integers. The statistical distance between the distributions $D_1 = \{[x]_\ell : x \xleftarrow{\$} \mathbb{Z}_L\}$ and $D_2 = \{x \xleftarrow{\$} \mathbb{Z}_\ell\}$ is bounded by* $\min(\ell, 2[L]_\ell)/L$.

## 3 A new CCA1 (and vCCA) secure variant of Paillier-ElGamal

The intuition behind our new scheme is very similar to Bitansky et al.' two-ciphertext Paillier heuristic that we have already discussed in Sect. 2.3.

Following this idea, we now define a two-message-slot variant of PEG, $\mathsf{PEG}_2$, in which the two message slots are under independent secret keys. Because of this, we can use the same randomness across these two slots. We then sparsify the plaintext domain of the resulting scheme following Damgård's Knowledge-of-Exponent template. This yields a compact scheme which we call Damgård-Paillier-ElGamal, denoted DPEG, as it is very similar in spirit to the Damgård-ElGamal scheme.

[5] Still, reliance on this latter assumption may be avoided under some conditions [27] on which we will come back later.

### 3.1 Paillier-ElGamal with two message slots ($\mathsf{PEG}_2$)

Following the definition of PEG (Sect. 2.4), $\mathsf{PEG}_2$ is simply defined as follows:

- $\mathsf{PEG}_2.\mathsf{KeyGen}$: choose a RSA modulus $N = (2p'+1)(2q'+1)$, with $p'$ and $q'$ two distinct Sophie Germain primes. Then, uniformly pick $g \in \mathbb{Z}_{N^2}^*$, $x, x' \in [\![0, N(N-1)/4]\!]$ and let $h = [g^{4Nx}]_{N^2}$ and $h' = [g^{4Nx'}]_{N^2}$. Then, $\mathsf{pk} = (N, g, h, h')$, $\mathsf{sk} = (x, x')$ and $\mathsf{sk}_0 = (p', q')$.
- $\mathsf{PEG}_2.\mathsf{Enc}$: given $\mathsf{pk}$ and $(m, m') \in \mathbb{Z}_N^2$, pick $r \in [\![0, (N-1)/4]\!]$ and return $(c_0, c_1, c_2) \in \mathbb{Z}_{N^2}^3$ such that,

$$(c_0, c_1, c_2) = ([g^{2Nr}]_{N^2}, [(1+N)^m h^r]_{N^2}, [(1+N)^{m'} h'^r]_{N^2}).$$

- $\mathsf{PEG}_2.\mathsf{Dec}$: given $\mathsf{sk}$ and $(c_0, c_1, c_2) \in \mathbb{Z}_{N^2}^3$, let $\mu = [c_1 \cdot c_0^{-2x}]_{N^2}$ and $\mu' = [c_2 \cdot c_0^{-2x'}]_{N^2}$. If $\mu$ is of the form $[(1+N)^m]_{N^2}$ for some $m \in \mathbb{Z}_N$ and $\mu'$ is of the form $[(1+N)^{m'}]_{N^2}$ for some $m' \in \mathbb{Z}_N$, then return $(m, m')$. Otherwise, return $\perp$.

As PEG, this scheme is perfectly correct and trivially linearly homomorphic offering operators,

- $\mathsf{PEG}_2.\mathsf{Add}$: given $(c_0, c_1, c_2)$ and $(c_0', c_1', c_2')$, both in $\mathbb{Z}_{N^2}^3$, return $([c_0 c_0']_{N^2}, [c_1 c_1']_{N^2}, [c_2 c_2']_{N^2})$.
- $\mathsf{PEG}_2.\mathsf{MulByConst}_\alpha$: given $\alpha \in \mathbb{Z}_N$ and $(c_0, c_1, c_2) \in \mathbb{Z}_{N^2}^3$, an encryption of $(m, m')$, return $([c_0^\alpha]_{N^2}, [c_1^\alpha]_{N^2}, [c_2^\alpha]_{N^2})$, an encryption of $(\alpha m, \alpha m')$.

The scheme is also CPA-secure under the DCR assumption as its CPA security is easy to establish from that of PEG. As a sanity check, we give a proof of this in appendix Sect. B.1 (Prop. 28).

### 3.2 Damgård-Paillier-ElGamal (DPEG)

We then apply the Knowledge-of-Exponent pattern within the two message slots of a $\mathsf{PEG}_2$ ciphertext. Like in Damgård-ElGamal, ciphertexts then consist of only 3 group elements. From the definition of $\mathsf{PEG}_2$ in the previous section, our new scheme DPEG is defined as follows:

- $\mathsf{DPEG}.\mathsf{KeyGen}$: run $\mathsf{PEG}_2.\mathsf{KeyGen}(1^\lambda)$ to get $\mathsf{PEG}_2.\mathsf{pk} = (N, g, h, h')$ and $(x, x')$. Further pick $\xi$ uniformly at random in $\mathbb{Z}_N$ and let,

$$\mathsf{ct}^\triangle = \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{PEG}_2.\mathsf{pk}, (1, \xi)).$$

  The public key is $\mathsf{pk} = (N, g, h, h', \mathsf{ct}^\triangle)$ while the secret key is $(x, x', \xi)$.
- $\mathsf{DPEG}.\mathsf{Enc}$: given $\mathsf{pk}$ and $m \in \mathbb{Z}_N$, return,

$$\mathsf{PEG}_2.\mathsf{Add}(\mathsf{PEG}_2.\mathsf{MulByConst}_m(\mathsf{ct}^\triangle), \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{PEG}_2.\mathsf{pk}, (0, 0))). \quad (6)$$

- $\mathsf{DPEG}.\mathsf{Dec}$: given $\mathsf{sk}$ and $\mathsf{ct} \in \mathbb{Z}_{N^2}^3$, if $\mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \perp$, then return $\perp$. Otherwise, let $(m, m') = \mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$, if $[\xi m]_N \neq m'$ then return $\perp$ and otherwise return $m$.

Remark that, since $\xi$ is private, the only well-formed ciphertexts that can be created without using $\mathsf{ct}^{\triangle}$ are encryptions of 0.

We now prove a first simple Lemma that establishes that for any $m \in \mathbb{Z}_N$ the distribution $\{\mathsf{DPEG.Enc}(\mathsf{DPEG.pk}, m; r) : r \xleftarrow{\$} [\![0, (N-1)/4]\!]\}$ cannot be distinguished from $\{\mathsf{PEG}_2.\mathsf{Enc}(\mathsf{PEG}_2.\mathsf{pk}, (m, \xi m); r) : r \xleftarrow{\$} [\![0, (N-1)/4]\!]\}$.

**Lemma 7.** *Let* $\mathsf{ct} = \mathsf{DPEG.Enc}(\mathsf{pk}, m)$*, then,*

$$\mathsf{ct} = ([g^{2Nr}]_{N^2}, [(1+N)^m h^r]_{N^2}, [(1+N)^{\xi m} h'^r]_{N^2})$$

*with* $r \xleftarrow{\$} [\![0, (N-1)/4]\!]$.

*Proof.* From Lemma 6, the distribution $\{z \xleftarrow{\$} [\![0, (N-1)/4]\!]\}$ is within $2^{-\lambda}$ statistical distance of the distribution $\{z \xleftarrow{\$} \mathbb{Z}_{p'q'}\}$, so that these two distributions can be used interchangeably. From Eq. (6), we have,

$$\begin{aligned}
\mathsf{ct} &= \mathsf{PEG}_2.\mathsf{Add}(\mathsf{PEG}_2.\mathsf{MulByConst}_m(\mathsf{ct}^{\triangle}), \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{PEG}_2.\mathsf{pk}, (0,0); r_0)) \\
&= ([g^{2Nr_{\triangle}m}]_{N^2}, [(1+N)^m h^{r_{\triangle}m}]_{N^2}, [(1+N)^{\xi m} h'^{r_{\triangle}m}]_{N^2}) \cdot ([g^{2Nr_0}]_{N^2}, [h_0^r]_{N^2}, h'^{r_0}]_{N^2}), \\
&= ([g^{2N[r_{\triangle}m+r_0]_{p'q'}}]_{N^2}, [(1+N)^m h^{[r_{\triangle}m+r_0]_{p'q'}}]_{N^2}, [(1+N)^{\xi m} h'^{[r_{\triangle}m+r_0]_{p'q'}}]_{N^2}), \\
&= ([g^{2Nr}]_{N^2}, [(1+N)^m h^r]_{N^2}, [(1+N)^{\xi m} h'^r]_{N^2}),
\end{aligned}$$

with, from $\mathsf{DPEG.KeyGen}$, $\mathsf{ct}^{\triangle} = \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{PEG}_2.\mathsf{pk}, (1, \xi); r_{\triangle})$. Since $r_0$ is uniformly picked in $\mathbb{Z}_{p'q'}$ at encryption, $r = [r_{\triangle}m + r_0]_{p'q'}$ is also uniform in $\mathbb{Z}_{p'q'}$. □

### 3.3 CCA1 security of DPEG under DCR

**Proposition 8.** *DPEG is CCA1-secure under the DCR assumption.*

We postpone the proof of this proposition to Sect. 4.3 (within the paper's body). In fact, we eventually prove the above proposition in two different ways:

- First, it will directly follow from two propositions that we establish in the next section (Sect. 4): Prop. 16 that establishes that DPEG belongs to the abstract class of sGPKE (under the DCR assumption) and Prop. 14 which shows that any sGPKE is CCA1-secure.
- Second, for the sake of it, we provide in Appendix B.2 another ad hoc proof that closely follows that of [27, Theorem 3] which establishes the CCA1 security of $\mathsf{PEG}_z$ under DCR and an additional (falsifiable) CNI assumption (Def. 37, in App. D.1 p. 51).

To the best of our knowledge, *DPEG is thus the first truly linearly homomorphic scheme that is proven CCA1-secure under a standard assumption.*

### 3.4 vCCA security of DPEG

Since DPEG instantiates the Bitansky et al. two-ciphertext heuristic [4] it is natural to assume that it has the LOH property that we recalled in Sect. 2.3 (p. 10). Under this assumption, there are several avenues for proving its vCCA security. First, [16, Proposition 2] provides the general result that any correct, CPA-secure public-key linearly homomorphic scheme that has the LOH property is vCCA secure, so the vCCA security of DPEG trivially follows from that result. For two-ciphertext Paillier, [21, Proposition 15] also gives a direct proof of vCCA security (by reduction to the CPA security of Paillier) that also works for proving the vCCA security of DPEG only at the cost of a few minor adaptations.

For self-containedness, we thus provide a proof that follows this latter strategy in appendix B.3. Remark that the fact that we have established the CCA1 security of DPEG (Prop. 8) does not help in proving vCCA security. Indeed, a reduction towards a CCA1 challenger against DPEG would still have to simulate a second step decryption oracle on its own since the challenger provides none.

**Proposition 9.** *DPEG is vCCA secure under the assumption that it has the LOH property and that it is CPA-secure.*

*Proof.* Deferred to appendix Sect. B.3 (p. 46). □

Although [16, Sect. 5.2] and [21, Sect. 6.3] already establish that Bitansky et al.'s two-ciphertext Paillier is vCCA secure under the same assumptions, it is a desirable feature that DPEG achieves CCA1 security under a falsifiable standard assumption and the stronger notion of vCCA security by adding a non-falsifiable one. Indeed, for the former scheme, the non-falsifiable LOH assumption bridges the much wider gap between CPA and vCCA security, a bolder leap of faith. It is also interesting to note that DPEG (as Bitansky et al.'s two-ciphertext Paillier) supports multi-hop (linear) homomorphic calculations while the known generic construction blueprints achieving vCCA security [30,10] are all single-hop.

### 3.5 A depth 1 multiplicative extension

In this section, we follow [21, Sect. 4.2] and apply the Catalano-Fiore blueprint [15] to get one level of homomorphic multiplication out of DPEG.

Indeed, in [15], Catalano and Fiore have proposed a clever and simple trick to get one level of multiplication from the Paillier scheme, or any other *truly* linearly homomorphic scheme. In a nutshell, considering Paillier, the trick consists in encrypting a message $m$ as a pair $(c_0, c_1) = ([m-b]_N, \mathsf{Pai.Enc}(b))$ for a uniformly chosen mask $b \in \mathbb{Z}_N$. Then, given two such ciphertexts $(c_0, c_1)$ and $(c'_0, c'_1)$, one may obtain an encryption of $mm'$ by building a triplet of the form

$$(\mathsf{Pai.Enc}(c_0 c'_0) c_1^{c'_0} {c'_1}^{c_0}, c_1, c'_1),$$

which first term is thus an encryption of $(m-b)(m'-b') + b(m'-b') + b'(m-b) = mm' - bb'$ under Pai. Although still linearly homomorphic to some extent, the

post-multiplication ciphertext structure is different from the pre-multiplication one and no procedure is known to switch from the latter form to the former, hence the limitation to one level of multiplication.

We can then apply this blueprint using DPEG as the underlying scheme, yielding scheme mDPEG which definition is deferred to Appendix C as the application of the above blueprint is direct and does not present any difficulty.

The CCA1 security of mDPEG follows by a simple reduction towards that of DPEG (Prop. 33, App. C, p. 50) and therefore also holds under DCR. To the best of our knowledge, *mDPEG is thus the first concrete scheme being both* truly *linearly homomorphic and supporting multiplications (even limited to one level) that is proven CCA1-secure under a standard assumption*. We emphasize that the BGN variant that is proven CCA1-secure under DDH in [27] uses the same plaintext zero padding trick as $\mathsf{PEG}_z$ and is therefore only somewhat homomorphic. Further remark that the Catalano-Fiore blueprint does not apply directly to a scheme, say E, encoding messages in the exponent of a group for which the discrete log is hard. Indeed, since the one-time mask $b$ is picked uniformly in the full plaintext domain of the scheme, retrieving it from $\mathsf{E.Enc}(b)$ almost surely fails with the decryption function returning $\perp$, even for fresh ciphertexts.

Likewise, the vCCA security of mDPEG also easily reduces to that of DPEG (Prop. 34, App. C, p. 50) and therefore holds under the same LOH assumption. Although [21, Sect. 6.3] already presents a closely related scheme that supports both homomorphic additions and (one level of) multiplications that is proven vCCA secure under the same assumptions (that scheme is essentially based on the Catalano-Fiore blueprint from the Bitansky et al. two-ciphertext Paillier scheme rather than DPEG), it is again a desirable feature that mDPEG achieves CCA1 security under a falsifiable standard assumption and then the stronger notion of vCCA security by adding a non-falsifiable one. Indeed, for the scheme in [21, Sect. 6.3], the non-falsifiable LOH assumption bridges the much wider gap between CPA and vCCA security.

## 4 On a new class of IND-CCA1-secure schemes

### 4.1 Sparsified Group-based PKE (sGPKE)

We now define the class of sGPKE schemes related to the main result in this section that establishes that any scheme in this class achieves CCA1 security under a falsifiable computational indistinguishability assumption. At the end of the section, we also show that DPEG belongs to this class, thus proving Prop. 8. Intuitively, any scheme in this class is obtained by *sparsifying* the ciphertext domain of another scheme that belongs to the class of GPKE. Before giving the formal definitions of both classes, we provide an intuitive overview (with pointers to the formal definitions such that both can be read side-by-side).

*GPKE (Def. 10).* The class of GPKE contains group-based linearly homomorphic schemes for which the plaintext domain defines a group $(G, +)$ and the image of the encryption function defines a group $(H, \cdot)$ that is embedded in a larger group

$(H^*, \cdot)$ such that random elements in $H$ are computationally indistinguishable from random elements in $H^*$ (property 3). From this class, we ask that the secret information that allows to distinguish between $\{z \xleftarrow{\$} H\}$ and $\{z \xleftarrow{\$} H^*\}$ is *not* required to decrypt (property 1) and further gives an efficient exact membership test for elements in $H$ (property 8). From this, we will be able to use a random element from either $H$ or $H^*$ to generate a challenge ciphertext (using the homomorphic properties) while giving the decryption key to a simulator playing the CCA1 game against some adversary, without the adversary noticing. Thanks to the exact membership test, we will be able to define an alternative decryption function that returns $\perp$ for any ciphertext not in $H$. As additional natural properties, we require that GPKE are linearly homomorphic over $H^*$ (property 5) and that evaluated ciphertexts are distributed like freshly encrypted ones (property 6). This latter property will allow us to build an alternative encryption function that uses the scheme homomorphic properties and that maps elements of $G$ onto elements of $H^*$ (rather than $H$), but cannot be distinguished from the original encryption function by an adversary. Lastly, given the public/private key pair of a GPKE, we ask that there exists many alternative secret keys that behave like the original one over $H$ (property 7) while yielding independent outputs outside of it. This last technical property will then allow us to conveniently switch among secret keys without an adversary noticing.

*sGPKE (Def. 11).* The class of sGPKE schemes is built from that of GPKE ones. Given a GPKE over groups $G$, $H$ and $H^*$, we obtain a sGPKE by considering a reduced plaintext domain $\mathcal{P}$ of size $2^{-\lambda}|G|$ such that there is an efficient (non-public) membership test for elements in $\mathcal{P}$. The decryption function of the sGPKE scheme then returns $\perp$ whenever the decryption function of the GPKE one returns an element that is not in the smaller plaintext domain $\mathcal{P}$. Thanks to property 8 (Def. 10) of GPKE, we are also able to define an alternative decryption function for the sGPKE that returns $\perp$ for any ciphertext not in $H$. Then, the main additional property (property 1 in Def. 11) that is required from a sGPKE is that, given the public key and a decryption oracle running this alternative decryption function, a PPT adversary cannot produce a ciphertext that does not belong to $H$ and such that the original decryption function does not return $\perp$. We will leverage on this property to prove that the decryption function conveniently does not leak any information to an adversary for ciphertexts not in $H$.

**Definition 10 (GPKE).** *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a Group-based linearly homomorphic Public-Key Encryption (GPKE) scheme if the following properties are fulfilled.*

1. *There exists functions* $\mathsf{KeyGen}_0$ *and* $\mathsf{KeyGen}_1$ *such that* $\mathsf{KeyGen}_0(1^\lambda)$ *outputs* $(\mathsf{pk}_0, \mathsf{sk}_0)$ *and* $\mathsf{KeyGen}_1(\mathsf{pk}_0)$ *outputs* $(\mathsf{pk}, \mathsf{sk})$*. Then,* $\mathsf{PKE.KeyGen}$ *is defined as* $\mathsf{PKE.KeyGen}(1^\lambda) = \mathsf{KeyGen}_1(\mathsf{KeyGen}_0(1^\lambda).\mathsf{pk}_0)$.
2. *For any security parameter* $\lambda$ *and all secret key/public key pairs* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$*, it is possible to define finite groups* $(G, +)$ *and* $(H, \cdot)$ *such that* $G$ *is the plaintext space, and* $H = \mathsf{PKE.Enc}(\mathsf{pk}, G; \mathsf{COIN})$ *is embedded in*

*the ciphertext space $\mathcal{C}$. Additionally, the uniform distributions over $G$ and $H$ must be efficiently sampleable up to negligible statistical distance in the security parameter.*

3. *Given $\mathsf{pk}_0$, $\mathsf{pk}$, $\mathsf{sk}$, with $(\mathsf{pk},\mathsf{sk}) = \mathsf{PKE.KeyGen}_1(\mathsf{pk}_0)$, there exists groups $(H^*,\cdot)$ and $(H',\cdot)$ such that $H^* = H \cdot H'$ with $H \cap H' = 1$, $\mathcal{C} \supseteq H^* \supset H$ and such that the distributions $\{z \xleftarrow{\$} H\}$ and $\{z \xleftarrow{\$} H^*\}$ are computationally indistinguishable without the knowledge of $sk_0$. Again, the uniform distributions over $H^*$ and $H'$ must be efficiently sampleable up to negligible statistical distance in the security parameter. Additionally, for all $c \notin H^*$, $\mathsf{PKE.Dec}(\mathsf{sk}, c) = \perp$.*
4. *The set $G$ grows exponentially in the security parameter $\lambda$.*
5. *$\mathsf{PKE}$ is perfectly correct and linearly homomorphic over $H^*$: $\forall m \in G$, it holds that $\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m)) = m$. Further, $\forall \mathsf{ct}, \mathsf{ct}' \in H^*$,*

$$\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct} \cdot \mathsf{ct}') = \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct}) + \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct}').$$

6. *For all $m, m' \in G$ and $r \in \mathsf{COIN}$, the distributions of $\mathsf{PKE.Enc}(\mathsf{pk}, m+m')$ and $\mathsf{PKE.Enc}(\mathsf{pk}, m) \cdot \mathsf{PKE.Enc}(\mathsf{pk}, m'; r)$ are within negligible statistical distance.*
7. *For all secret key/public key pairs $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, for $\mathfrak{h}' \xleftarrow{\$} H'$ and $\mu \xleftarrow{\$} G$, there is a PPT algorithm taking as input $(\mathsf{pk},\mathsf{sk},\mathfrak{h}',\mu)$ that outputs, with overwhelming probability, a pair $(\mathsf{pk}, \widehat{\mathsf{sk}})$ such that $\mathsf{PKE.Dec}(\mathsf{sk}, c) = \mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, c)$ for all $c \in H$, $\mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = \mu$ and the distribution of $\widehat{\mathsf{sk}}$ is indistinguishable from that of the second output of $\mathsf{KeyGen}$ conditioned on its first output being $\mathsf{pk}$.*
8. *Given $\mathsf{sk}$ and $\mathsf{sk}_0$, there exists an efficient exact membership test for elements in $H$. Equivalently, there exists an alternative decryption function $\widehat{\mathsf{PKE.Dec}}$ such that, for any $c \in \mathcal{C}$, $\widehat{\mathsf{PKE.Dec}}(\mathsf{sk}, \mathsf{sk}_0, c) = \perp$, if $c \notin H$, and $\widehat{\mathsf{PKE.Dec}}(\mathsf{sk}, \mathsf{sk}_0, c) = \mathsf{PKE.Dec}(\mathsf{sk}, c)$, otherwise ($c \in H$).*

**Definition 11 (sGPKE).** *Given a GPKE scheme $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, define $\mathsf{sPKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{ImVer})$, with plaintext domain $\mathsf{sPKE}.\mathcal{P} \subset \mathsf{PKE}.G$ with $|\mathsf{sPKE}.\mathcal{P}| \leq 2^{-\lambda}|\mathsf{PKE}.G|$ and ciphertext domain $\mathsf{sPKE}.\mathcal{C} = \mathsf{PKE}.\mathcal{C} = \mathcal{C}$, as follows,*

- *$\mathsf{sPKE.KeyGen}$: run $\mathsf{PKE.KeyGen}$ to get $(\mathsf{PKE.pk}, \mathsf{PKE.sk})$. Let $\mathsf{pk}'$ and $\mathsf{sk}'$ denote any additional key material[6], return $((\mathsf{PKE.pk}, \mathsf{pk}'), (\mathsf{PKE.sk}, \mathsf{sk}'))$ as $(\mathsf{sPKE.pk}, \mathsf{sPKE.sk})$ .*
- *$\mathsf{sPKE.Enc}$: given $\mathsf{sPKE.pk}$ and $m \in \mathsf{sPKE}.\mathcal{P}$, return $\mathsf{PKE.Enc}(\mathsf{PKE.pk}, m)$.*
- *$\mathsf{sPKE.ImVer}$: given $\mathsf{sPKE.sk}$ and $c \in \mathcal{C}$, let $\mu = \mathsf{PKE.Dec}(\mathsf{PKE.sk}, c)$, if $\mu \in \mathsf{sPKE}.\mathcal{P}$, then return $\mathsf{True}$ and otherwise return $\mathsf{False}$.*
- *$\mathsf{sPKE.Dec}$: given $c \in \mathcal{C}$ and $\mathsf{sPKE.sk}$, if $\mathsf{sPKE.ImVer}(\mathsf{sk}, c) = \mathsf{True}$ then return $\mathsf{PKE.Dec}(\mathsf{sk}, c)$, otherwise return $\perp$.*

[6] Here, $\mathsf{pk}'$, $\mathsf{sk}'$ refer to the additional key material that may be required for the $\mathsf{sGPKE}$ but not already included in the underlying $\mathsf{GPKE}$ key material. For instance, for $\mathsf{DPEG}$, $\mathsf{pk}'$ is $\mathsf{ct}^{\triangle}$ and $\mathsf{sk}'$ is $\xi$.

*Additionally, from property 8 in Def. 10, we also define,*

- $\mathsf{sPKE.}\widehat{\mathsf{Dec}}$*:* *given* $c \in \mathcal{C}$, $\mathsf{sPKE.sk}$ *and* $\mathsf{PKE.sk}_0$, *if* $\mathsf{sPKE.ImVer}(\mathsf{sk}, c) = \mathsf{True}$ *then return* $\mathsf{PKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{PKE.sk}_0, c)$, *otherwise return* $\perp$. *Hence, for any* $c \in \mathcal{C}$, $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{PKE.sk}_0, c) = \perp$, *if* $c \notin H$, *and* $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{PKE.sk}_0, c) = \mathsf{sPKE.Dec}(\mathsf{sk}, c)$ *otherwise* ($c \in H$).

*Then* sPKE *is a sparsified* GPKE *or* sGPKE *scheme when* ImVer *satisfies the following property,*

1. *For any PPT adversary* $\mathcal{A}$,

$$P\left(\begin{array}{l|l} \mathsf{sPKE.ImVer}(\mathsf{sk}, c) = \mathsf{True} & (\mathsf{pk}, \mathsf{sk}) := \mathsf{sPKE.KeyGen}(1^\lambda) \\ \wedge c \notin H & c \leftarrow \mathcal{A}^{\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, .)}(\mathsf{pk}) \end{array}\right) \leq neg(\lambda). \quad (7)$$

Remark that although GPKE's are correct and linearly homomorphic by definition, sGPKE's may not be correct and may only be somewhat linearly homomorphic. We will emphasize this when considering concrete examples of schemes.

We also highlight that several properties of our framework can be rephrased in the HPS framework. However, what a standard HPS does not capture and what is essential to a truly linearly homomorphic CCA1 scheme is homomorphism preservation: that the image $H$ of the encryption function is a subgroup of $H^*$ over which the decryption function is a group homomorphism, and that evaluated ciphertexts are distributed as fresh ones (Properties 5 and 6). This is exactly the distinction between a truly homomorphic sparsification, such as DPEG, whose valid domain $\{(m, \xi m)\}$ is closed under addition, and a merely somewhat homomorphic one, such as $\mathsf{PEG}_z$, whose valid domain is not. This is a distinction that the plain HPS view does not cover. However, although this is beyond the scope of the present paper, we think that our sGPKE framework may be seen as a subgroup-membership HPS specialized to a sparse subgroup and augmented with the homomorphism-preservation conditions that make truly-homomorphic CCA1 encryption possible.

### 4.2 A general result on sGPKE's CCA1 security

In this section, we establish the following result.

**Proposition 12 (informal).** *Any sGPKE is CCA1-secure (under a falsifiable computational indistinguishability assumption).*

*Proof outline.* To prove this result we first establish a convenient technical Lemma (Lemma 13) which establishes that a PPT adversary against an sGPKE cannot, given the public key and a decryption oracle, generate a ciphertext that does not belong to $H$ and that decrypts to anything other than $\perp$.

With this Lemma in our toolbox, the proof of the above result is not too hard. The idea of the proof relies on defining a new encryption function that needs knowledge of the secret key and uses the scheme's homomorphic properties. This new encryption function uniformly picks an element $c_r \in H$ as additional randomness. The proof then proceeds with the following steps:

- First, we show that the challenger can equivalently use the new encryption function to generate the challenge ciphertext.
- Second, we show that the challenger can be modified to pick the additional randomness $c_r$ uniformly in $H^*$ rather than in $H$. The adversary cannot notice this modification by reduction to the computational indistinguishability assumption in property 3 of Def. 10.
- The next step considers a further modified challenger that creates a new secret key $\widehat{\mathsf{sk}}$ using property 7 in Def. 10 that behaves like $\mathsf{sk}$ over $H$ but decrypts some $\mathfrak{h}' \xleftarrow{\$} H'$ to some $\mu \xleftarrow{\$} G$. From property 7, the challenger can then use $\widehat{\mathsf{sk}}$ instead of $\mathsf{sk}$ for the whole game without the adversary noticing.
- Finally, we change the challenger such that it replies all the decryption requests from the adversary using $\mathsf{sk}$ and only the challenge ciphertext is generated with $\widehat{\mathsf{sk}}$. Our technical Lemma 13 tells us that this change cannot be noticed by the adversary since it essentially tells us that the adversary cannot get a non-$\perp$ answer to any of its decryption requests over $H'$, while $\mathsf{sk}$ and $\widehat{\mathsf{sk}}$ yields the same answers over $H$. In this last step, the challenger picks $\mu \xleftarrow{\$} G$ and $\mathfrak{h}' \xleftarrow{\$} H'$ *after* the last decryption request from the adversary, such that the adversary's view is independent of both values.

With the last challenger in the sequence, the challenge ciphertext is a random element in $H^*$ as a product of $\mathsf{sPKE.Enc}(\mathsf{pk}, m_\gamma - \mu)$ by $\mathfrak{h}'$. Hence, from the adversary's view the challenge ciphertext is independent of the challenge bit, yielding a null advantage.

**Lemma 13.** *Given any sGPKE scheme* $\mathsf{sPKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, for any PPT adversary* $\mathcal{A}$*,*

$$P\left(\begin{array}{l}\mathsf{sPKE.Dec}(\mathsf{sk}, c) \neq \perp \\ \wedge c \notin H\end{array}\middle|\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) := \mathsf{sPKE.KeyGen}(1^\lambda) \\ c := \mathcal{A}^{\mathsf{sPKE.Dec}(\mathsf{sk},.)}(\mathsf{pk})\end{array}\right) \leq neg(\lambda), \tag{8}$$

*Proof.* Consistently with Eq. (8), we define $G_0$ as the game in which a challenger first runs $\mathsf{sPKE.KeyGen}(1^\lambda)$ to get $(\mathsf{pk}, \mathsf{sk})$ and gives $\mathsf{pk}$ to some PPT adversary $\mathcal{A}$. The challenger then answers polynomially many decryption requests from $\mathcal{A}$. The adversary then wins the game if, for its last decryption request, it submits a ciphertext $c \notin H$ and gets $\mathsf{sPKE.Dec}(\mathsf{sk}, c) \neq \perp$ in return.

First recall that, by property 3 in Def. 10, for any $c \notin H^*$, $\mathsf{PKE.Dec}(\mathsf{PKE.sk}, c) = \perp$ and, hence, $\mathsf{sPKE.Dec}(\mathsf{sPKE.sk}, c) = \perp$. So $\mathcal{A}$ cannot win $G_0$ by submitting a ciphertext $c \notin H^*$ for its last decryption request.

For notational convenience, $H_0$ is the same game as $G_0$.

*Hybrid argument.* We now consider the sequence of games $H_i$ ($i \in [\![1, Q]\!]$) where $Q$ is the number of decryption requests performed by the adversary. In game $H_i$, we proceed as follows,

- Up to the $i$-th decryption query, the challenger uses secret keys $\mathsf{sk}$ and $\mathsf{sk}_0$ with the modified decryption algorithm $\mathsf{sPKE.}\widehat{\mathsf{Dec}}$ (from Def. 11) to reply to the adversary's decryption requests, i.e. let $c_j$, $j \leq i$, be the $j$-th ciphertext submitted by the adversary, then $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, c_j)$ is returned.

– After the $i$-th decryption query, the challenger answers the remaining $Q - i$ decryption requests by means of the normal decryption function $\mathsf{sPKE.Dec}$, i.e. let $c_j$, $j > i$, be the $j$-th ciphertext submitted by the adversary, then $\mathsf{sPKE.Dec}(\mathsf{sk}, c_j)$ is returned.

*Indistinguishability between games $H_i$ and $H_{i-1}$.* Clearly, from the adversary's view, games $H_i$ (where the $i$-th decryption request is replied to by means of $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, .)$) and $H_{i-1}$ (where the $i$-th decryption request is replied to by means of $\mathsf{sPKE.Dec}(\mathsf{sk}, .)$) are indistinguishable unless the $i$-th decryption query involves a ciphertext that is rejected in the former but not the latter. Let, again, $c_i$ denote the ciphertext submitted by $\mathcal{A}$ at the $i$-th decryption query, we then consider the following cases:

– $c_i \in H$: in this case $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, c_i) = \mathsf{sPKE.Dec}(\mathsf{sk}, c_i)$, by property 8 in Def. 10, and both games proceed identically.
– $c_i \notin H$: in this case, $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, c_i) = \perp$ $(H_i)$. Also, $\mathsf{sPKE.ImVer}(\mathsf{sk}, c_i) = \mathsf{False}$ with overwhelming probability, by property 1 in Def. 11. So, both games proceed identically since $\mathsf{sPKE.Dec}(\mathsf{sk}, c) = \perp$ with overwhelming probability.

It thus follows from the standard hybrid argument that $H_0 \equiv G_0$ is indistinguishable from game $H_{Q-1}$, in which all but the last of the adversary's decryption requests are replied to by means of $\mathsf{sPKE.}\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, .)$ and the last one is replied to by means of $\mathsf{sPKE.Dec}(\mathsf{sk}, .)$.

*Game $G_1$.* Let $G_1 \equiv H_{Q-1}$. In this last game, when $\mathcal{A}$ submits its last decryption request over ciphertext $c \in H^* \setminus H$, property 1 in Def. 11 implies that $\mathsf{sPKE.ImVer}(\mathsf{sk}, c) = \mathsf{True}$ with negligible probability and, by definition of $\mathsf{sPKE.Dec}$, that $\mathsf{sPKE.Dec}(\mathsf{sk}, c) \neq \perp$ with the same probability. □

**Proposition 14.** *Any sGPKE, sPKE = (KeyGen, Enc, ImVer, Dec, $\widehat{\mathsf{Dec}}$), is CCA1-secure.*

*Proof.* We denote $W_i$ the event that an adversary $\mathcal{A}$ wins the game $G_i$, then

$$\mathsf{Adv}^{\mathrm{CCA1}}_{\mathsf{sPKE}}(\mathcal{A}) = |\Pr[W_0] - 1/2|.$$

We now define a new encryption function that needs knowledge of the secret key as it uses the decryption function of the GPKE, PKE, on which sPKE is built,

$$\mathsf{sPKE.Enc}'(\mathsf{sk}, \mathsf{pk}, m; (r, c_r)) = \mathsf{PKE.Enc}(\mathsf{pk}, m - \mathsf{PKE.Dec}(\mathsf{sk}, c_r); r) \cdot c_r, \quad (9)$$

with $c_r \in H^*$. For all $(\mathsf{pk}, \mathsf{sk})$, all $m \in \mathcal{P}$ and $c_r \in H$, the distributions $\mathsf{sPKE.Enc}(\mathsf{pk}, m)$ and $\mathsf{sPKE.Enc}'(\mathsf{sk}, \mathsf{pk}, m; c_r)$ are within negligible statistical distance. Indeed,

$$\begin{aligned}
\mathsf{sPKE.Enc}'(\mathsf{sk}, \mathsf{pk}, m; (r, c_r)) &= \mathsf{PKE.Enc}(\mathsf{pk}, m - \mathsf{PKE.Dec}(\mathsf{sk}, c_r); r) \\
&\quad \cdot \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{PKE.Dec}(\mathsf{sk}, c_r); r'), \\
&= \mathsf{PKE.Enc}(\mathsf{pk}, m - \mathsf{PKE.Dec}(\mathsf{sk}, c_r) + \mathsf{PKE.Dec}(\mathsf{sk}, c_r); r''), \\
&= \mathsf{sPKE.Enc}(\mathsf{pk}, m; r''), \qquad (10)
\end{aligned}$$

where, by property 6, the distribution of $\mathsf{sPKE.Enc}(\mathsf{pk}, m; r'')$ taken over $r''$ is negligibly close to that of $\mathsf{sPKE.Enc}(\mathsf{pk}, m)$.

*Game* $G_0$. This is the standard CCA1 game. $\mathcal{A}$ can ask an oracle for decryption of any ciphertext $c \in \mathcal{C}$. He then chooses $m_0, m_1$ and receives $c^* = \mathsf{sPKE.Enc}(\mathsf{pk}, m_\gamma; r)$, $\gamma \xleftarrow{\$} \{0,1\}$, in return from the challenger, outputs a guess $\gamma'$ for $\gamma$ and wins the game when $\gamma = \gamma'$.

*Game* $G_1$. In this game, the challenger returns,

$$c^* = \mathsf{sPKE.Enc}'(\mathsf{sk}, \mathsf{pk}, m_\gamma; (r, c_r)),$$

with $c_r$ chosen uniformly in $H$. In $G_1$, the challenge ciphertext distribution is, in $\mathcal{A}$'s view, negligibly close to that of one from $G_0$ by Eq. (10). It implies that $G_0$ and $G_1$ cannot be distinguished and that $|\Pr[W_0] - \Pr[W_1]| \leq \mathrm{neg}(\lambda)$.

*Game* $G_2$. This game is the same as $G_1$ except that $c_r$ is chosen uniformly in $H^*$ (in $G_2$) rather than $H$ (in $G_1$). The indistinguishability between $G_1$ and $G_2$ is given by property 3 in Def. 10.

Indeed, given an adversary $\mathcal{A}_0$ that can distinguish between $G_1$ and $G_2$, there exists an adversary $\mathcal{B}_0$ that breaks the indistinguishability assumption in property 3 using $\mathcal{A}_0$ as a subroutine, and which achieves the same advantage as $\mathcal{A}_0$. The reduction $\mathcal{B}_0$ then proceeds as follows. First, after it gets $\mathsf{pk}_0$ from its challenger which runs $\mathsf{PKE.KeyGen}_0(1^\lambda)$, $\mathcal{B}_0$ runs $\mathsf{sPKE.KeyGen}_1(\mathsf{pk}_0)$ to get $(\mathsf{pk}, \mathsf{sk})$. It then sends $\mathsf{pk}$ to $\mathcal{A}_0$ as well as $(\mathsf{pk}, \mathsf{sk})$ to its challenger. The reduction then simply answers $\mathcal{A}_0$'s decryption requests by running $\mathsf{sPKE.Dec}(\mathsf{sk}, .)$ on its own. Finally, when $\mathcal{A}_0$ issues its challenge request over $m_0, m_1 \in \mathsf{sPKE}.P$, $\mathcal{B}_0$ asks its challenger for a single sample

$$c_r \xleftarrow{\$} \begin{cases} H & \text{if } \beta = 0, \\ H^* & \text{if } \beta = 1, \end{cases}$$

where $\beta$ is the challenger's secret bit. Finally, it returns,

$$c^* = \mathsf{sPKE.Enc}'(\mathsf{sk}, \mathsf{pk}, m_\gamma; (r, c_r)),$$

for some random bit $\gamma$, to $\mathcal{A}_0$ and eventually replies as $\mathcal{A}_0$ does. Clearly, $\mathcal{B}_0$ perfectly simulates $G_1$ when $\beta = 0$ and $G_2$ when $\beta = 1$.

So we have $\mathsf{Adv}^{G_1/G_2}(\mathcal{A}_0) \leq \mathsf{Adv}(\mathcal{B}_0)$, and so $|\Pr[W_1] - \Pr[W_2]| \leq \mathsf{Adv}^{\cdot}(\mathcal{B}_0)$.

*Game* $G_3$. In this game, the challenger picks $\mu \xleftarrow{\$} G$ and $\mathfrak{h}' \xleftarrow{\$} H'$ and uses property 7 in Def. 10 to generate a new secret key $\widehat{\mathsf{sk}}$ which is used in lieu of $\mathsf{sk}$. Both games are indistinguishable since, also from that property, the distribution of $\widehat{\mathsf{sk}}$ is indistinguishable from that of the output of $\mathsf{KeyGen}$. Hence, $|\Pr[W_2] - \Pr[W_3]| \leq \mathrm{neg}(\lambda)$.

*Game* $G_4$. In $G_4$ all the adversary's decryption requests are replied to by means of $\mathsf{sPKE.Dec}(\mathsf{sk}, .)$ and only the challenge ciphertext is generated by means of $\widehat{\mathsf{sk}}$. Clearly, games $G_3$ and $G_4$ cannot be distinguished unless the adversary can issue a decryption request over a ciphertext $c \notin H$ that does not decrypt to $\perp$

under $\mathsf{sk}$ (respectively $\widehat{\mathsf{sk}}$) as $c$ would decrypt to $\perp$ with overwhelming probability under $\widehat{\mathsf{sk}}$ (respectively $\mathsf{sk}$). By Lemma 13, this occurs with negligible probability in the security parameter. As a result, in $G_4$, $\mu$ *and* $\mathfrak{h}'$ *can be uniformly picked in* $G$ *and* $H'$ *respectively after all the adversary's requests.*

Let $c_r = (\mathfrak{h}, \mathfrak{h}') \in H \times H'$. In this last game, the challenge ciphertext is thus,

$$\begin{aligned}
c^* &= \mathsf{sPKE.Enc}'(\widehat{\mathsf{sk}}, \mathsf{pk}, m_\gamma; (r, c_r)),\\
&= \mathsf{PKE.Enc}(\mathsf{pk}, m_\gamma - \mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, c_r); r) \cdot c_r,\\
&= \mathsf{PKE.Enc}(\mathsf{pk}, m_\gamma - \mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}) - \mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}'); r) \cdot c_r,\\
&= \mathsf{PKE.Enc}(\mathsf{pk}, m_\gamma - \mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}) - \mu; r) \cdot c_r,\\
&= \mathsf{PKE.Enc}(\mathsf{pk}, m_\gamma - \mathsf{PKE.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}) + \mathsf{PKE.Dec}(\mathsf{sk}, \mathfrak{h}) - \mu; r) \cdot \mathfrak{h}',\\
&= \mathsf{PKE.Enc}(\mathsf{pk}, m_\gamma - \mu; r) \cdot \mathfrak{h}'.
\end{aligned}$$

Since $\mu$ and $\mathfrak{h}'$ have, in this game, been chosen *independently* after all of $\mathcal{A}$'s queries, it implies that $m_\gamma - \mu$ is uniformly distributed in $G$ from $\mathcal{A}$'s view, in the statistical sense. Hence, $c^*$ is a random element in $H^*$ independent of $m_\gamma$ and $\Pr[W_4] = 1/2$.

Hence, from the above sequence of game hops, $\mathsf{Adv}^{\mathrm{CCA1}}_{\mathsf{sPKE}}(\mathcal{A})$ is negligible. □

### 4.3 DPEG as a sGPKE (proof of Prop. 8)

To prove Prop. 8, i.e. establish the CCA1 security of DPEG, by means of the sGPKE framework we then proceed in two steps. First, we establish that $\mathsf{PEG}_2$ is a GPKE (under the DCR assumption) and then that DPEG is a sGPKE (under the same assumption). The CCA1 security of DPEG then follows from Prop. 14.

**Lemma 15.** *$\mathsf{PEG}_2$ is a GPKE under the DCR assumption.*

*Proof. Property 1.* Following the definition of $\mathsf{PEG}_2$,

- $\mathsf{KeyGen}_0$: on input $1^\lambda$, choose a RSA modulus $N = (2p'+1)(2q'+1)$, with $p'$ and $q'$ two distinct Sophie Germain primes of suitable size. Then $\mathsf{pk}_0 = N$ and $\mathsf{sk}_0 = (p', q')$.
- $\mathsf{KeyGen}_1$: on input $\mathsf{pk}_0 = N$, uniformly pick $g \in \mathbb{Z}^*_{N^2}$, $x, x' \in [\![0, N(N-1)/4]\!]$ and let $h = [g^{4Nx}]_{N^2}$ and $h' = [g^{4Nx'}]_{N^2}$. Then, $\mathsf{pk} = (N, g, h, h')$ and $\mathsf{sk} = (x, x')$.

*Property 2.* $G = \mathbb{Z}^2_N$ and[7],

$$H = \{(c_0, c_1, c_2) \in \mathbb{Z}^3_{N^2} : c_0 \in \langle g^{2N} \rangle, c_1 \in \langle 1+N \rangle c_0^{2x}, c_2 \in \langle 1+N \rangle c_0^{2x'}\}.$$

[7] Without loss of generality, in this proof and that of Prop. 16, we identify ciphertexts that differ by multiplication with an element of $T = \{(\zeta, 1, 1) : [\zeta^2]_{N^2} = 1\}$. Indeed, the output of $\mathsf{PEG}_2.\mathsf{Dec}$, $\mathsf{PEG}_2.\widehat{\mathsf{Dec}}$, $\mathsf{DPEG.Dec}$ and $\mathsf{DPEG}.\widehat{\mathsf{Dec}}$ over a ciphertext $\mathsf{ct}$ is invariant by such a multiplication and we can therefore restrict the analysis to the case where $\zeta = 1$.

For every key pair $(\mathsf{pk}, \mathsf{sk})$, let $\mathsf{ct}$ be a element of $H$, then,

$$\mathsf{ct} = (g^{2Nr}, (1+N)^m.g^{4Nrx}, (1+N)^{m'}.g^{4Nrx'}) = \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{pk}, (m, m'); r),$$

for some $r \in \mathbb{Z}_{p'q'} \subset [\![0, (N-1)/4]\!]$ and $(m, m') \in G$. So, $\mathsf{PEG}_2.\mathsf{Enc}(\mathsf{pk}, G; \mathsf{COIN}) = H$. Furthermore, the distribution of encryptions of uniformly random messages is negligibly close (in the security parameter) to the uniform distribution over $H$. This is due to the fact that the encryption randomness is sampled from $[\![0, (N-1)/4]\!]$ rather than $\mathbb{Z}_{p'q'}$. However, from Lemma 6, the distribution $\{z \xleftarrow{\$} [\![0, (N-1)/4]\!]\}$ is within $2^{-\lambda}$ statistical distance from $\{z \xleftarrow{\$} \mathbb{Z}_{p'q'}\}$ so the two distributions can be used interchangeably.

*Property 3.* Given $\mathsf{pk}_0 = N$, $\mathsf{PEG}_2.\mathsf{KeyGen}_1(\mathsf{pk}_0) = (\mathsf{pk}, \mathsf{sk}) = ((N, g, h, h'), x, x')$, let $H' = \left\langle ((1+N), (1+N)^{2x}, (1+N)^{2x'}) \right\rangle$. The indistinguishability between $H^* = H \cdot H'$ and $H$ follows by the DCR assumption, as it is equivalent to distinguish between, $\{[z^{2N}]_{N^2} : z \xleftarrow{\$} \mathbb{Z}_N\}$ and $\{[z^2]_{N^2} : z \xleftarrow{\$} \mathbb{Z}_{N^2}\}$, without knowledge of the factorization of $N$, i.e. $\mathsf{sk}_0 = (p', q')$. Also, the uniform distribution over $H'$ is efficiently sampleable by doing $((1+N)^u, (1+N)^{2xu}, (1+N)^{2x'u})$ for $u$ uniformly picked in $\mathbb{Z}_N$ and the uniform distribution over $H^*$ is thus efficiently sampleable up to negligible statistical distance by independently sampling $\mathfrak{h}$ in $H$ and $\mathfrak{h}'$ in $H'$ as above and returning $\mathfrak{h} \cdot \mathfrak{h}'$. Additionally, for any $c = (c_0, c_1, c_2) \notin H^*$, we have that either $c_1 c_0^{-2x} \notin \langle 1+N \rangle$ or $c_2 c_0^{-2x'} \notin \langle 1+N \rangle$ and by definition, $\mathsf{PEG}_2.\mathsf{Dec}(c) = \perp$.

*Property 4.* $G = \mathbb{Z}_N^2$ with $N >> 2^\lambda$.

*Property 5.* The perfect correctness of $\mathsf{PEG}_2$ follows from that of $\mathsf{PEG}$. Further, let $\mathsf{ct} = (c_0, c_1, c_2) \in H^*$ and $\mathsf{ct}' = (c_0', c_1', c_2') \in H^*$, with

$$\begin{aligned}
\mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) &= ((c_1 \cdot c_0^{-2x} - 1)/N, (c_2 \cdot c_0^{-2x'} - 1)/N) \\
&= (((1+N)^{\alpha_0} - 1)/N, ((1+N)^{\alpha_1} - 1)/N) \\
&= (((1+\alpha_0 N) - 1)/N, ((1+\alpha_1 N) - 1)/N) \\
&= (\alpha_0, \alpha_1),
\end{aligned}$$

for some $\alpha_0, \alpha_1 \in \mathbb{Z}_N$, and likewise for $\mathsf{ct}'$. Then $[(1+\alpha_0 N)(1+\alpha_0' N)]_{N^2} = [(1+(\alpha_0+\alpha_0')N)]_{N^2}$ (and similarly so for the second slot) implies property 5.

*Property 6.* Let $\mathsf{ct} = \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{pk}, (m_0, m_1); r)$ and $\mathsf{ct}' = \mathsf{PEG}_2.\mathsf{Enc}(\mathsf{pk}, (m_0', m_1'); r')$,

$$\begin{aligned}
\mathsf{ct} \cdot \mathsf{ct}' &= ([g^{2Nr}]_{N^2}, [(1+N)^{m_0} h^r]_{N^2}, [(1+N)^{m_1} h'^r]_{N^2}) \\
&\quad \cdot ([g^{2Nr'}]_{N^2}, [(1+N)^{m_0'} h^{r'}]_{N^2}, [(1+N)^{m_1'} h'^{r'}]_{N^2}), \\
&= ([g^{2N[r+r']_{p'q'}}]_{N^2}, [(1+N)^{(m_0+m_0')} h^{[r+r']_{p'q'}}]_{N^2}, [(1+N)^{(m_1+m_1')} h'^{[r+r']_{p'q'}}]_{N^2}), \\
&= ([g^{2Nr''}]_{N^2}, [(1+N)^{(m_0+m_0')} h^{r''}]_{N^2}, [(1+N)^{(m_1+m_1')} h'^{r''}]_{N^2}),
\end{aligned}$$

with either or both $r$ and $r'$ uniformly random in $\mathbb{Z}_{p'q'}$, $r'' = [r+r']_{p'q'}$ is also uniform in $\mathbb{Z}_{p'q'}$. Now, recall that the encryption randomness is picked uniformly in $[\![0, (N-1)/4]\!]$. Then Lemma 6 implies that the distributions, $\{[g^{2Nr}]_{N^2} : r \xleftarrow{\$}$

$\mathbb{Z}_{p'q'}\}$ and $\{[g^{2Nr}]_{N^2} : r \xleftarrow{\$} [\![0,(N-1)/4]\!]\}$, are within $2^{-\lambda}$ statistical distance since the subgroup of $2N$-th residues in $\mathbb{Z}^*_{N^2}$ is a cyclic group of order $p'q'$ and $[g^{2N}]_{N^2}$ is with overwhelming probability a generator for the latter [27].

*Property 7.* Given a key pair $(\mathsf{pk},\mathsf{sk}) = ((N,g,h,h'),x,x')$, $\mathsf{PEG}_2.\mathsf{Dec} : H^* \longrightarrow G$ takes $(c_0,c_1,c_2)$ and returns $(m,m') \in G$ such that $1+mN = [c_1 \cdot c_0^{-2x}]_{N^2}$ and $1+m'N = [c_2 \cdot c_0^{-2x'}]_{N^2}$. Now consider an uniformly picked $\mathfrak{h}' = ((1+N)^a,(1+N)^{2xa},(1+N)^{2x'a}) \in H'$, for some $a \in \mathbb{Z}_N$ (note that $a$ is invertible in $\mathbb{Z}_N$ with overwhelming probability), and $(\mu,\mu') \xleftarrow{\$} G$. Since $h$ and $c_1c_0^{-2x}$ only depend on $[x]_{p'q'}$ and since $h'$ and $c_2c_0^{-2x'}$ only depend on $[x']_{p'q'}$, we can compute another valid key pair $(\mathsf{pk},\widehat{\mathsf{sk}}) = ((N,g,h,h'),\hat{x},\hat{x}')$, such that $[\hat{x}]_{p'q'} = [x]_{p'q'}$ and $[\hat{x}']_{p'q'} = [x']_{p'q'}$. Then, we have $\mathsf{PEG}_2.\mathsf{Dec}(\widehat{\mathsf{sk}},\mathfrak{h}') = ([2(x-\hat{x})a]_N,[2(x'-\hat{x}')a]_N)$, and we can choose $\widehat{\mathsf{sk}}$ such that $[\hat{x}]_N = [x-\mu(2a)^{-1}]_N$ and $[\hat{x}']_N = [x'-\mu'(2a)^{-1}]_N$ to get $\mathsf{PEG}_2.\mathsf{Dec}(\widehat{\mathsf{sk}},\mathfrak{h}') = (\mu,\mu')$. By Lemma 6, the statistical distance between $\{z \xleftarrow{\$} \mathbb{Z}_{Np'q'}\}$ and $\{z \xleftarrow{\$} [\![0,N(N-1)/4]\!]\}$ (used for key generation) is negligible in $\lambda$. Since $(\mu,\mu') \xleftarrow{\$} G$, both $[\hat{x}]_N$ and $[\hat{x}']_N$ appear uniform in $\mathbb{Z}_N$. Also, $[\hat{x}]_{p'q'} = [x]_{p'q'}$ and $[\hat{x}']_{p'q'} = [x']_{p'q'}$ are uniform in $\mathbb{Z}_{p'q'}$ (from key generation). It follows that both $\hat{x}$ and $\hat{x}'$ are indistinguishable from uniform in $[\![0,N(N-1)/4]\!]$.

*Property 8.* For $c = (c_0,c_1,c_2) \in H$, $c_0$ is by definition of the form $[g^{2Nr}]_{N^2}$. For $g \in \mathbb{Z}_{N^2}$, $g^2$ is with overwhelming probability a generator of the squares of $\mathbb{Z}_{N^2}$, $(\langle g^{2N}\rangle,\cdot)$ is the only subgroup of $(\mathbb{Z}_{N^2},\cdot)$ of order $p'q'$. It follows that $c_0 \in (\langle g^{2N}\rangle,\cdot)$ if and only if $[c_0^{2p'q'}]_{N^2} = 1$. We can thus define an alternative decryption function that uses the factorization of $N$ given by $\mathsf{sk}_0 = (p',q')$,

- $\mathsf{PEG}_2.\widehat{\mathsf{Dec}}$: given $c = (c_0,c_1,c_2) \in \mathbb{Z}^3_{N^2}$, $\mathsf{sk}$ and $\mathsf{sk}_0 = (p',q')$, return $\perp$ when $[c_0^{2p'q'}]_{N^2} \neq 1$ and, otherwise, return $\mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk},c)$.

□

**Proposition 16.** *DPEG is a sGPKE under the DCR assumption.*

*Proof.* From its definition (Sect. 3.2) DPEG uses $\mathsf{PEG}_2$ as its inner scheme and, from Lemma 15 above, that scheme is a GPKE under the DCR assumption.

$\mathsf{DPEG}.\mathcal{P} = \{(m,\xi m) : m \in \mathbb{Z}_N\} \cong \mathbb{Z}_N$, hence $2^\lambda|\mathsf{DPEG}.\mathcal{P}| \leq |\mathsf{PEG}_2.G| = N^2$ since $N >> 2^\lambda$.

From Sect. 3.2 (p. 13), DPEG.Dec can equivalently be written as,

- DPEG.ImVer: given DPEG.sk and $\mathsf{ct} \in \mathbb{Z}^3_{N^2}$, if $\mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk},\mathsf{ct}) = \perp$, then return False. Otherwise, let $(\mu,\mu') = \mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk},\mathsf{ct})$, if $[\xi\mu]_N \neq \mu'$ then return False and otherwise return True.
- DPEG.Dec: given DPEG.sk and $\mathsf{ct} \in \mathbb{Z}^3_{N^2}$, if $\mathsf{DPEG.ImVer}(\mathsf{sk},\mathsf{ct}) = \mathsf{False}$, then return $\perp$. Otherwise, let $(\mu,\mu') = \mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk},\mathsf{ct})$, return $\mu$.

To establish the claim, it remains to prove that DPEG satisfies Eq. (7).

First recall that in $\mathsf{sk} = (x,x')$, $x$ and $x'$ are picked uniformly in $[\![0,N(N-1)/4]\!]$. Let us thus start with a first game hop between $G_0$, the experiment in

Eq. (7), and $G_1$ the same experiment with the key generation modified such that $x$ and $x'$ are picked uniformly in $\mathbb{Z}_{Np'q'}$ rather than $[\![0, N(N-1)/4]\!]$. The indistinguishability between these two experiments follows since the statistical distance between $\{[z]_{Np'q'} : z \xleftarrow{\$} [\![0, N(N-1)/4]\!]\}$ and $\{z \xleftarrow{\$} \mathbb{Z}_{Np'q'}\}$ is negligible in $\lambda$ (by Lemma 6). In $G_1$, $[x]_N$ is independent of $[x]_{p'q'}$ since $\gcd(N, p'q') = 1$, and likewise for $x'$. Now,

- In $\mathsf{pk} = (N, g, h, h')$, $h = [g^{4Nx}]$ is the only term that depends on $x$ and it only depends on $[x]_{p'q'}$. Similarly, $h' = [g^{4Nx'}]$ is the only term that depends on $x'$ and it only depends on $[x']_{p'q'}$.
- Additionally, for any ciphertext $\mathsf{ct} \in \mathbb{Z}^3_{N^2}$, $\mathsf{PEG}_2.\widehat{\mathsf{Dec}}(\mathsf{ct})$ also depends only on $[x]_{p'q'}$ and $[x']_{p'q'}$.

Hence, for any $(\mathsf{pk}, \mathsf{sk})$ and any $\mathcal{A}^{\mathsf{DPEG}.\widehat{\mathsf{Dec}}(\mathsf{sk},\mathsf{sk}_0,.)}(\mathsf{pk})$, $\mathcal{A}$'s view is independent from both $[x]_N$ and $[x']_N$[8]. Assume the adversary against $G_1$ submits a ciphertext in $H^*$[9], i.e. of the form,

$$\mathsf{ct} = ((1+N)^{a_0} g^{2Nr}, (1+N)^{a_1} h^r, (1+N)^{a_2} h'^r)$$

for some $a_0, a_1, a_2, r$. If $\mathsf{ct} \notin H$, then $a_0 \neq 0$ and $\mathsf{PEG}_2.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ returns $\mu = [a_1 - 2[x]_N a_0]_N$ and $\mu' = [a_2 - 2[x']_N a_0]_N$. We know that $[x]_N$ and $[x']_N$ are independent from the adversary's view. This implies that both $[a_1 - 2xa_0]_N$ and $[a_2 - 2x'a_0]_N$ are uniformly distributed in $\mathbb{Z}_N$ from $\mathcal{A}$'s view. Hence, $[\xi\mu]_N = \mu'$ only with probability $\mathrm{neg}(\lambda)$. Note that if $a_0$ is not invertible in $\mathbb{Z}_N$, say a multiple of $p$, then $[a_1 - 2xa_0]_N$ and $[a_2 - 2x'a_0]_N$ also respectively appear uniformly distributed in $[a_1 + p\mathbb{Z}_q]_N$ and $[a_2 + p\mathbb{Z}_q]_N$. Since $q >> 2^\lambda$ (and likewise for $p$), it is also the case that $[\xi\mu]_N = \mu'$ with probability $\mathrm{neg}(\lambda)$. □

Taken together, Prop. 16 and 14 therefore prove Prop. 8.

### 4.4 Other examples of known (somewhat homomorphic) sGPKE

In this section, we illustrate that our sGPKE framework allows to establish the CCA1 security of several known schemes. As all the schemes in this section are already known to achieve the same, we defer the details and proofs to App. A.

[8] We highlight that, in both Eq. (7) and $G_1$, the adversary is granted access to an oracle running $\mathsf{DPEG}.\widehat{\mathsf{Dec}}$ rather than $\mathsf{DPEG}.\mathsf{Dec}$. Then, in $G_1$, the output of $\mathsf{DPEG}.\widehat{\mathsf{Dec}}$ does not depend on $[x]_N$ or $[x']_N$ since: for a ciphertext not in $H$, $\mathsf{DPEG}.\widehat{\mathsf{Dec}}$ returns $\perp$, so its output does not depend on $x$ or $x'$ and, for a ciphertext in $H$, the output of $\mathsf{DPEG}.\widehat{\mathsf{Dec}}$ depends only on $[x]_{p'q'}$ and $[x']_{p'q'}$ via the terms $c_0^{-2x} = c_0^{-2[x]_{p'q'}}$ and $c_0^{-2x'} = c_0^{-2[x']_{p'q'}}$. Hence, before submitting its final ciphertext to the experiment in $G_1$, the adversary's view is independent of $[x]_N$ and $[x']_N$. Further note that the connection between Eq. (7) and an adversary with oracle access to $\mathsf{DPEG}.\mathsf{Dec}$ follows from Lemma 13 which applies generically to any sGPKE.

[9] $\mathsf{PEG}_2.\mathsf{Dec}$ returns $\perp$ for ciphertexts not in $H^*$.

**Paillier-ElGamal with plaintext zero-padding ($\mathsf{PEG}_z$, Sect. 2.5).** In Appendix A.1, Lemma 21 first establishes that PEG is a GPKE under DCR and Proposition 22 then that $\mathsf{PEG}_z$ belongs to the sGPKE class. Thus Proposition 14 implies that $\mathsf{PEG}_z$ is CCA1-secure solely under DCR. Interestingly, this result establishes this without reliance on the Composite Non-Invertibility assumption (Def. 37 in Appendix D.1) contrary to [27, Theorem 3] which requires this extra assumption in some parameter regime (see Remark 30, p. 46). This closes an open question from that paper: "An interesting open question is to prove the CCA1 security of the scheme in Sect. 5.1 under the sole DCR assumption and without sacrificing more than 50% of the message space" [27, Sect. 5.4].

**Cramer-Shoup Lite.** We recall the definition of the Cramer-Shoup Lite scheme [17] (denoted CSL) in Appendix A.2. To comply with Def. 11 we split the definition in two schemes: $\mathsf{CSL}_0$ and CSL. In that section, Lemma 24 thus first establishes that $\mathsf{CSL}_0$ is a GPKE under DDH and Proposition 25 that CSL belongs to the sGPKE class. Thus, Prop. 14 implies that CSL is CCA1-secure under DDH, a fact which was already established in [17] by a seminal dedicated proof.

We emphasize that CSL is truly multiplicatively homomorphic but that its additive variant, which consists in encrypting a message $m$ as $c_2 = g^m A^r$ rather than $mA^r$, then requires solving a discrete-log in the decryption function. So, its effective plaintext size is limited by the constraint that the decryption algorithm should be PPT. As a consequence, a bound $B$ has to be enforced over the messages. Hence, *this additive variant of the scheme is "only" somewhat linearly homomorphic* and does not satisfy correctness (recall Eq. 2). Interestingly, the Damgård-ElGamal scheme is equivalent to the three first slots of CSL. Since we now know that this latter scheme is CCA1-secure under DDH, due to the recent proof by Libert [27, Theorem 1], it means that the term $c_3$ can be dropped in the additive-variant of CSL.

**Damgård's additively homomorphic ElGamal.** We recall the definition of the Damgård-ElGamal scheme [27,18] (denoted DEG) in Appendix A.3. To comply with Def. 11 we split the definition in two schemes: $\mathsf{DEG}_0$ and DEG. In that section, Lemma 26 thus first establishes that $\mathsf{DEG}_0$ is a GPKE under DDH and Proposition 27 that DEG belongs to the sGPKE class. Thus Proposition 14 implies that DEG is CCA1-secure under DDH, a fact which also follows from Libert's recent proof [27, Theorem 1] (previous proofs of CCA1 security for that scheme required a non-falsifiable Knowledge-of-Exponent assumption).

Again, DEG is only somewhat linearly homomorphic because of the $B = 2^{O(\log(\lambda))}$ bound enforced on the message space to ensure that the decryption function is PPT. Hence, the scheme does not achieve correctness over evaluated ciphertexts (Eq. 2). A similar situation also occurs with the scheme in [28, Sect. 4.2] since its decryption function requires retrieving a plaintext $m$ from $\tilde{g}^m$, and may as such retrieve the sum of several messages only "as long as it remains in a sufficiently small range" (Sect. 4.1).

## 5 vCCA security of DPEG under falsifiable assumptions?

**Proposition 17.** *If DPEG is vCCA secure, then there exists a scheme, DPEG′, that has the LOH property.*

*Proof.* We build $\mathsf{DPEG}'$ from $\mathsf{DPEG}$ by modifying the following:

- $\mathsf{DPEG}'.\mathsf{KeyGen}$: run $\mathsf{DPEG.KeyGen}(1^\lambda)$ to get $(\mathsf{DPEG.pk}, \mathsf{DPEG.sk})$. Let $K = 4\lambda + 6\log_2|N|$, then $\mathsf{DPEG}'.\mathsf{sk} = (\mathsf{DPEG.pk}, \mathsf{DPEG.sk})$[10] and $\mathsf{DPEG}'.\mathsf{pk}$ is given by $(\mathsf{ct}^{(1)}, \mathsf{ct}_0^{(0)}, ..., \mathsf{ct}_{K-1}^{(0)})$, where,

$$\mathsf{ct}^{(1)} = \mathsf{DPEG.Enc}(\mathsf{DPEG.pk}, 1) \text{ and } \mathsf{ct}_i^{(0)} = \mathsf{DPEG.Enc}(\mathsf{DPEG.pk}, 0), \text{ for } i \in [\![0, K-1]\!]. \tag{11}$$

- $\mathsf{DPEG}'.\mathsf{Enc}$: given $m \in \mathcal{P}$ and $\mathsf{DPEG}'.\mathsf{pk}$, pick a random subset $E$ of $[\![0, K-1]\!]$ and return, $\mathsf{DPEG.MulByConst}_m(\mathsf{ct}^{(1)}) \oplus \bigoplus_{i \in E} \mathsf{ct}_i^{(0)}$.

Note that $\mathsf{DPEG}'$'s CPA security follows from property 3 (from Def. 10, i.e. the ciphertext pseudorandomness of $\mathsf{DPEG}$ from Lemma 15), our choice for $K$ and the LHL [38, Claim 5.3][11] which implies that ciphertexts under $\mathsf{DPEG}'$ are indistinguishable from uniform over $\mathbb{Z}_{N^2}^3$. Also remark that $\mathsf{DPEG}'$ vCCA security follows by a straightforward reduction to that of $\mathsf{DPEG}$ (Lemma 31 in App. B.4).

We now show that $\mathsf{DPEG}'$ has the LOH property (Def. 5). We prove this by using the vCCA extractor, which we denote $\mathsf{extract}_{\mathrm{vCCA}}$, to build $\mathsf{DPEG}'$'s LOH extractor, $\mathsf{extract}_{\mathrm{LOH}}$, as required by Eq. (3). Let $\mathcal{A}$ denote the adversary against the LOH property of $\mathsf{DPEG}'$ in Eq. (3),

- Since, in $\mathsf{DPEG}'$, $\mathsf{DPEG.pk}$ is not available to $\mathcal{A}$, the only well-formed ciphertexts it can use as inputs to homomorphic evaluations are the elements of $\mathsf{DPEG}'.\mathsf{pk}$. Therefore, any ciphertext produced by the adversary that does not decrypt to $\perp$ must be obtained by linearly combining these elements (Lemma 32, Sect. B.4).
- By definition, the vCCA extractor works *only* on ciphertexts generated by the adversary.

Recall Eq. (3)'s notations, let $\Omega = (c_0, ..., c_{m-1}) = (\mathsf{DPEG}'.\mathsf{Enc}(\mathsf{pk}, a_0), ..., \mathsf{DPEG}'.\mathsf{Enc}(\mathsf{pk}, a_{m-1}))$ denote the set of ciphertexts given to $\mathcal{A}$ and $\Omega' = (c'_0, ..., c'_{k-1})$ denote the set of ciphertexts ouput by the adversary. Let $c' \in \Omega'$, We distinguish two cases:

[10] This is a slight abuse of notation as we have to make the distinction between a scheme's public parameters and its public key which we have not yet needed in the paper. For $\mathsf{DPEG}$, the set of public parameters is the modulus $N$ and the public key is $(g, h, h', \mathsf{ct}^{\triangle})$. The key point is that the public parameters alone do not allow to build valid ciphertexts, although they may be needed for the homomorphic operators.

[11] Claim 5.3 in [38] establishes that, given a finite Abelian group $G$ and $K$ uniformly picked elements $g_0$, ..., $g_{K-1}$ in $G$, the probability that the statistical distance between the uniform distribution on $G$ and the distribution given by the sum of a random subset of $g_0$, ..., $g_{K-1}$ exceeds $\sqrt[4]{|G|/2^K}$ is bounded by $\sqrt[4]{|G|/2^K}$.

1. $\mathsf{extract}_{\mathrm{vCCA}}(c', \mathsf{aux})$ outputs[12] a vector $\rho \in \mathbb{N}^{m+K+1}$ such that,

$$c' = \bigoplus_{i=0}^{m-1} \bigoplus_{j=1}^{\rho_i} c_i \oplus \bigoplus_{j=1}^{\rho_m} \mathsf{ct}^{(1)} \oplus \bigoplus_{i=0}^{K-1} \bigoplus_{j=1}^{\rho_{i+m+1}} \mathsf{ct}_i^{(0)}.$$

   In this case, from the correctness of DPEG,

$$\mathsf{DPEG}'.\mathsf{Dec}(\mathsf{DPEG}'.\mathsf{sk}, c') = \mathsf{DPEG}.\mathsf{Dec}(\mathsf{DPEG}.\mathsf{sk}, c') = \sum_{i=0}^{m-1} \rho_i a_i + \rho_m.$$

   Hence, setting $\mathsf{extract}_{\mathrm{LOH}}(c', \mathsf{aux}) = (\pi, b)$ with $\pi \in \mathcal{P}^m : \pi_i = [\rho_i]_t, i \in [\![0, m-1]\!]$ and $b = [\rho_m]_t$ gives the expected LOH extractor output.
2. $\mathsf{extract}_{\mathrm{vCCA}}(c', \mathsf{aux}) = (\mathsf{id}, c')$:
   - If $c' = c_i$ for some $i \in [\![0, m-1]\!]$, then $((0 \ldots 1_i \ldots 0), 0)$ is returned as the output for the LOH extractor.
   - If $c' = \mathsf{ct}^{(1)}$, then $((0 \ldots 0), 1)$ is returned as the output for the LOH extractor.
   - If $c' = \mathsf{ct}_i^{(0)}$ for some $i \in [\![0, K-1]\!]$, then $((0 \ldots 0), 0)$ is returned as the output for the LOH extractor.
   - Otherwise, $\varnothing$ is returned as the output for the LOH extractor. Consistently with the above (and Lemma 32), $\mathsf{DPEG}'.\mathsf{Dec}(\mathsf{DPEG}'.\mathsf{sk}, c') = \mathsf{DPEG}.\mathsf{Dec}(\mathsf{DPEG}.\mathsf{sk}, c') = \perp$ in this case, so this also is the expected LOH extractor output.

□

To prove our next claim. We will need the following two Theorems.

**Theorem 18 ([3,4], Construction 6.1).** *If there exists a homomorphic scheme with the LOH property, then there exists a designated-verifier (preprocessing) SNARK for any language in $\mathcal{NP}$.*

**Theorem 19 ([25], Corollary 5.2).** *Under the assumption that there exists some sub-exponentially hard subset-membership problem in $\mathcal{NP}$, any candidate publicly verifiable or designated-verifier SNARK construction for any $\mathcal{NP}$-complete language cannot have a black-box proof of security based on a falsifiable assumption, unless the assumption is false.*

**Proposition 20.** *Under the assumptions of Theorem 19, DPEG cannot have a black-box proof of vCCA security based on a falsifiable assumption.*

*Proof.* Otherwise, DPEG′ would have the LOH property grounded in a falsifiable assumptions (Prop. 17), which would then allow to build a SNARK for any language in $\mathcal{NP}$ (Theorem 18) which security would also be based on the same assumption. Thus a contradiction to Theorem 19 would result. □

[12] This is a slight abuse of notation as we identify the circuit output by $\mathsf{extract}_{\mathrm{vCCA}}$ to the corresponding vector of coefficients of the linear combination it evaluates.

## 6 Conclusion and perspectives.

We believe that achieving "truly" linearly homomorphic encryption with CCA1 security is an important question, beautiful in theory and meaningful in practice. We have developed a framework toward this goal and leave as an open challenge the design of schemes with improved or optimal efficiency. While our results primarily concern linearly homomorphic encryption, they naturally raise questions about the feasibility of achieving CCA security for stronger forms of homomorphism, in particular fully homomorphic encryption (FHE), under standard falsifiable assumptions.

On the positive side, our construction shows that it is possible to obtain linearly homomorphic encryption supporting one level of multiplication together with CCA1 security from falsifiable assumptions. Although still far from FHE, this can be viewed as a promising step toward constructing CCA1-secure FHE, or at least somewhat homomorphic schemes, whose security relies on falsifiable assumptions. On the negative side, our impossibility result for the vCCA security of DPEG under falsifiable assumptions may extend to FHE schemes and suggests that FHE achieving stronger CCA1-type security may not be provable under purely falsifiable assumptions. To us, this is an important question to settle. A related important open question, also, remains to determine the optimal CCA security guarantees achievable from falsifiable assumptions for homomorphic schemes.

Additionally, we primarily focused on vCCA security as it is the strongest known-so-far CCA security notion achievable for FHE. In particular, vCCA security is strictly stronger than HCCA [37] and other notions [30, Fig. 1] such as CCA1.5 [19] while being strictly weaker than RCCA [12] which is not achievable by nontrivial homomorphic schemes. Furthermore, [30] generalizes targeted malleability [8] by defining TNM-vCCA and proves it equivalent to the IND notion. However, considering weaker notions gives an interesting perspective since we ruled out black-box proofs of DPEG's vCCA security from falsifiable assumptions and thus leave open the question of whether it may achieve another stronger-than-CCA1 notion under falsifiable ones.

## Acknowledgments

The authors would like to thank Pierre-Emmanuel Clet who read an early version of this work and made useful comments as well as Jérôme Nguyen for insightful discussions on vCCA security. We are also grateful to the anonymous reviewers who made several suggestions that led to improvements in the paper.

## References

1. Akavia, A., Gentry, C., Halevi, S., Vald, M.: Achievable CCA2 relaxation for homomorphic encryption. In: TCC. pp. 70 – 99 (2022)

2. Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., Paneth, O.: Succinct non-interactive arguments via linear interactive proofs. Tech. Rep. 718, IACR ePrint (2012)
3. Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., Paneth, O.: Succinct non-interactive arguments via linear interactive proofs. In: TCC. pp. 315–333 (2013)
4. Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., Paneth, O.: Succinct non-interactive arguments via linear interactive proofs. Journal of Cryptology **35**, 15–87 (2022)
5. Bondarchuk, A., Chakraborty, O., Couteau, G., Sirdey, R.: Downlink (T)FHE ciphertexts compression. In: SAC. pp. 647–678 (2025)
6. Boneh, D., Ishai, Y., Sahai, A., Wu, D.J.: Lattice-based snargs and their application to more efficient obfuscation. In: EUROCRYPT. p. 247–277 (2017)
7. Boneh, D., Segev, G., Waters, B.: Targeted malleability: homomorphic encryption for restricted computations. In: ITCS. pp. 350–366 (2012)
8. Boneh, D., Segev, G., Waters, B.: Targeted malleability: homomorphic encryption for restricted computations. In: ITCS. p. 350–366 (2012)
9. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: ASIACRYPT. pp. 37–54 (2003)
10. Brzuska, C., Canard, S., Fontaine, C., Phan, D.H., Pointcheval, D., Renard, M., Sirdey, R.: Relations among new CCA security notions for approximate FHE. IACR Communications in Cryptology **2**(1) (2025)
11. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: CRYPTO. pp. 126–144 (2003)
12. Canetti, R., Krawczyk, H., Nielsen, J.: Relaxing chosen-ciphertext security. In: CRYPTO. p. 565–582 (2003)
13. Canetti, R., Raghuraman, S., Richelson, S., Vaikuntanathan, V.: Chosen-ciphertext secure fully homomorphic encryption. In: PKC. pp. 213–240 (2017)
14. Canetti, R., Raghuraman, S., Richelson, S., Vaikuntanathan, V.: Chosen-ciphertext secure fully homomorphic encryption. In: PKC. pp. 213–240 (2017)
15. Catalano, D., Fiore, D.: Using Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data. In: ACM SIGSAC. pp. 1518–1529 (2015)
16. Checri, M., Clet, P.E., Renard, M., Sirdey, R.: Achieving vCCA security from the linear-only homomorphism assumption. Tech. Rep. ePrint 2025/894, IACR (2025)
17. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2003)
18. Damgard, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: CRYPTO. p. 445–456 (1992)
19. Das, A., Dutta, S., Adhikari, A.: Indistinguishability against chosen ciphertext verification attack revisited: The complete picture. In: ProvSec. p. 104–120 (2013)
20. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: EUROCRYPT. pp. 342–360 (2004)
21. Fontaine, C., Renard, M., Sirdey, R., Stan, O.: On new variants of funcCPA security and related CCA-secure constructions. In: PKC. pp. 32–64 (2026)
22. Ganesh, C., Kondi, Y., Orlandi, C., Pancholi, M., Takahashi, A., Tschudi, D.: Witness-succinct universally-composable SNARKs. In: EUROCRYPT. p. 315–346 (2023)
23. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: EUROCRYPT. pp. 625–645 (2013)

24. Gennaro, R., Minelli, M., Nitulescu, A., Orrù, M.: Lattice-based zk-SNARKs from square span programs. In: CCS. pp. 556–573 (2018)
25. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: STOC. pp. 99–108 (2011)
26. Hofheinz, D.: All-but-many lossy trapdoor functions. In: EUROCRYPT. p. 209–227 (2012)
27. Libert, B.: Leveraging small message spaces for CCA1 security in additively homomorphic and BGN-type encryption. In: EUROCRYPT. pp. 34–63 (2025)
28. Libert, B., Peters, T., Joye, M., Yung, M.: Linearly homomorphic structure-preserving signatures and their applications. In: CRYPTO. pp. 289–307 (2013)
29. Loftus, J., May, A., Smart, N.P., Vercauteren, F.: On CCA-secure somewhat homomorphic encryption. In: SAC. pp. 55–72 (2011)
30. Manulis, M., Nguyen, J.: Fully homomorphic encryption beyond IND-CCA1 security: Integrity through verifiability. In: EUROCRYPT. pp. 63–93 (2024)
31. Naor, M.: On cryptographic assumptions and challenges. In: CRYPTO. pp. 96–109 (2003)
32. Nguyen, J.: Personal communication (2026)
33. Nitulescu, A.: Lattice-based zero-knowledge SNARGs for arithmetic circuits. In: LATINCRYPT. pp. 217–236 (2019)
34. Nitulescu, A.: *zk*-snarks: A gentle introduction. Tech. rep., DI ENS (2020)
35. Orlandi, C., Scholl, P., Yakoubov, S.: The rise of paillier: Homomorphic secret sharing and public-key silent OT. In: EUROCRYPT. pp. 678–708 (2021)
36. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: EUROCRYPT. pp. 223–238 (1999)
37. Prabhakaran, M., Rosulek, M.: Homomorphic encryption with CCA security. In: ICALP. p. 667–678 (2008)
38. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM **56** (2009)
39. Roy, L., Singh, J.: Large message homomorphic secret sharing from DCR and applications. In: CRYPTO. pp. 687–717 (2021)
40. Schäge, S.: New limits of provable security and applications to ElGamal encryption. In: EUROCRYPT. p. 255–285 (2024)
41. Schäge, S., Vorstermans, M.: New limits for homomorphic encryption. In: ASIACRYPT. p. xxx (2025)
42. Yang, R., Yu, Z., Susilo, W.: Fully homomorphic encryption with chosen-ciphertext security from LWE. In: CRYPTO. p. 371–405 (2025)

## A Examples of known sGPKE

In this section, we illustrate that our sGPKE framework allows to establish the CCA1 security (under a falsifiable indistinguishability assumption) of several known schemes. All the schemes in this section are also already known to achieve the same, however with separate proofs.

### A.1 Paillier-ElGamal with plaintext zero-padding ($\mathsf{PEG}_z$, Sect. 2.5)

First, we consider the Paillier-ElGamal variant of Libert [27], that we defined in Sect. 2.5. The proofs in this section are very similar to those in Sect. 4.3, for DPEG.

**Lemma 21.** *PEG is a GPKE under the DCR assumption.*

*Proof. Property 1.* Following the definition of PEG (Sect. 2.4, p. 11),

- $\mathsf{KeyGen}_0$: on input $1^\lambda$, choose a RSA modulus $N = (2p'+1)(2q'+1)$, with $p'$ and $q'$ two distinct Sophie Germain primes of suitable size. Then $\mathsf{pk}_0 = N$ and $\mathsf{sk}_0 = (p', q')$.
- $\mathsf{KeyGen}_1$: on input $\mathsf{pk}_0 = N$, uniformly pick $g \in \mathbb{Z}^*_{N^2}$, $x \in [\![0, N(N-1)/4]\!]$ and let $h = [g^{4Nx}]_{N^2}$. Then, $\mathsf{pk} = (N, g, h)$ and $\mathsf{sk} = x$.

*Property 2.* $G = \mathbb{Z}_N$ and[13],

$$H = \{(c_0, c_1) \in \mathbb{Z}^2_{N^2} : c_0 \in \left\langle g^{2N} \right\rangle, c_1 \in \langle 1+N \rangle\, c_0^{2x}\}.$$

For every key pair $(\mathsf{pk}, \mathsf{sk})$, let $\mathsf{ct}$ be a element of $H$, then,

$$\mathsf{ct} = (g^{2Nr}, (1+N)^m . g^{4Nrx}) = \mathsf{PEG.Enc}(\mathsf{pk}, m; r),$$

for some $r \in \mathbb{Z}_{p'q'} \subset [\![0, (N-1)/4]\!]$ and $m \in G$. Hence, $\mathsf{PEG.Enc}(\mathsf{pk}, G; \mathsf{COIN}) = H$. Furthermore, the distribution of encryptions of uniformly random messages is negligibly close (in the security parameter) to the uniform distribution over $H$. This is due to the fact that the encryption randomness is sampled from $[\![0, (N-1)/4]\!]$ rather than $\mathbb{Z}_{p'q'}$. However, from Lemma 6, the distribution $\{z \xleftarrow{\$} [\![0, (N-1)/4]\!]\}$ is within $2^{-\lambda}$ statistical distance from $\{z \xleftarrow{\$} \mathbb{Z}_{p'q'}\}$ so the two distributions can be used interchangeably.

*Property 3.* Given $\mathsf{pk}_0 = N$, $\mathsf{PEG.KeyGen}_1(\mathsf{pk}_0) = (\mathsf{sk}, \mathsf{pk}) = (x, (N, g, h))$, let $H' = \left\langle ((1+N), (1+N)^{2x}) \right\rangle$. The indistinguishability between $H^* = H \cdot H'$ and $H$, follows by the DCR assumption, as it is equivalent to distinguish between,

$$\{[z^{2N}]_N : z \xleftarrow{\$} \mathbb{Z}_N\} \text{ and } \{[z^2]_{N^2} : z \xleftarrow{\$} \mathbb{Z}_{N^2}\},$$

without knowledge of the factorization of $N$, i.e. $\mathsf{sk}_0 = (p', q')$. Also, the uniform distribution over $H'$ is efficiently sampleable by doing $((1+N)^u, (1+N)^{2xu})$ for $u$ uniformly picked in $\mathbb{Z}_N$ and the uniform distribution over $H^*$ is thus efficiently sampleable up to negligible statistical distance by independently sampling $\mathfrak{h}$ in $H$ and $\mathfrak{h}'$ in $H'$ as above and returning $\mathfrak{h} \cdot \mathfrak{h}'$. Additionally, for any $c = (c_0, c_1) \notin H^*$, we have that $c_1 c_0^{-2x} \notin \langle 1+N \rangle$ and by definition, $\mathsf{PEG.Dec}(c) = \perp$.

*Property 4.* $G = \mathbb{Z}_N$ with $N >> 2^\lambda$.

*Property 5.* The perfect correctness of PEG has already been established. Further, let $\mathsf{ct} = (c_0, c_1) \in H^*$ and $\mathsf{ct}' = (c'_0, c'_1) \in H^*$, with,

$$\begin{aligned} \mathsf{PEG.Dec}(\mathsf{sk}, \mathsf{ct}) &= (c_1 \cdot c_0^{-2x} - 1)/N \\ &= ((1+N)^\alpha - 1)/N \\ &= ((1+\alpha N) - 1)/N \\ &= \alpha, \end{aligned}$$

[13] Again, throughout this proof and that of Prop. 22, we abstract away the 2-torsion in $c_0$. See footnote 7, p. 23.

for some $\alpha \in \mathbb{Z}_N$, and likewise for $\mathsf{ct}'$. Then $[(1+\alpha N)(1+\alpha' N)]_{N^2} = [(1+(\alpha+\alpha')N)]_{N^2}$ implies property 5.

*Property 6.* Let $\mathsf{ct} = \mathsf{PEG.Enc}(\mathsf{pk}, m; r)$ and $\mathsf{ct}' = \mathsf{PEG.Enc}(\mathsf{pk}, m'; r')$,

$$\begin{aligned}\mathsf{ct}\cdot\mathsf{ct}' &= ([g^{2Nr}]_{N^2}, [(1+n)^m h^r]_{N^2}) \cdot ([g^{2Nr'}]_{N^2}, [(1+N)^{m'} h^{r'}]_{N^2}),\\ &= ([g^{2N[r+r']_{p'q'}}]_{N^2}, [(1+N)^{(m+m')} h^{[r+r']_{p'q'}}]_{N^2}),\\ &= ([g^{2Nr''}]_{N^2}, [(1+N)^{(m+m')} h^{r''}]_{N^2}),\end{aligned}$$

with either or both $r$ and $r'$ uniformly random in $\mathbb{Z}_{p'q'}$, $r'' = [r+r']_{p'q'}$ is also uniform in $\mathbb{Z}_{p'q'}$. Now, recall that the encryption randomness is picked uniformly in $[\![0, (N-1)/4]\!]$. Then Lemma 6 implies that the distributions $\{[g^{2Nr}]_{N^2} : r \xleftarrow{\$} \mathbb{Z}_{p'q'}\}$ and $\{[g^{2Nr}]_{N^2} : r \xleftarrow{\$} [\![0, (N-1)/4]\!]\}$ are within $2^{-\lambda}$ statistical distance since the subgroup of $2N$-th residues in $\mathbb{Z}^*_{N^2}$ is a cyclic group of order $p'q'$ and $[g^{2N}]_{N^2}$ is with overwhelming probability a generator for the latter [27].

*Property 7.* Given a key pair $(\mathsf{pk}, \mathsf{sk}) = ((N, g, h), x)$, $\mathsf{PEG.Dec} : H^* \longrightarrow G$ takes $(c_0, c_1)$ and returns $m \in G$ such that $1 + mN = [c_1 c_0^{-2x}]_{N^2}$. Now consider an uniformly picked $\mathfrak{h}' = ((1+N)^a, (1+N)^{2xa}) \in H'$ for $a \in \mathbb{Z}_N$ (note that $a$ is invertible in $\mathbb{Z}_N$ with overwhelming probability) and $\mu \xleftarrow{\$} G$. Since $h$ and $c_1 c_0^{-2x}$ only depend on $[x]_{p'q'}$, we can compute another valid key pair $(\mathsf{pk}, \widehat{\mathsf{sk}}) = ((N, g, h), \hat{x})$, such that $[\hat{x}]_{p'q'} = [x]_{p'q'}$. Then, we have $\mathsf{PEG.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = [2(x-\hat{x})a]_N$, and we can choose $\widehat{\mathsf{sk}}$ such that $[\hat{x}]_N = [x - \mu(2a)^{-1}]_N$ to get $\mathsf{PEG.Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = \mu$. By Lemma 6, the statistical distance between $\{z \xleftarrow{\$} \mathbb{Z}_{Np'q'}\}$ and $\{z \xleftarrow{\$} [\![0, N(N-1)/4]\!]\}$ (used for key generation) is negligible in $\lambda$. Since, $\mu \xleftarrow{\$} G$, $[\hat{x}]_N$ appears uniform in $\mathbb{Z}_N$ and $[\hat{x}]_{p'q'} = [x]_{p'q'}$ is also uniform in $\mathbb{Z}_{p'q'}$ (from key generation). It follows that $\hat{x}$ is indistinguishable from uniform in $[\![0, N(N-1)/4]\!]$.

*Property 8.* For $c = (c_0, c_1) \in H$, $c_0$ is by definition of the form $[g^{2Nr}]_{N^2}$. For $g \in \mathbb{Z}_{N^2}$, $g^2$ is with overwhelming probability a generator of the squares of $\mathbb{Z}_{N^2}$, $(\langle g^{2N} \rangle, \cdot)$ is the only subgroup of $(\mathbb{Z}_{N^2}, \cdot)$ of order $p'q'$. It follows that $c_0 \in (\langle g^{2N} \rangle, \cdot)$ if and only if $[c_0^{2p'q'}]_{N^2} = 1$. We can thus define an alternative decryption function that uses the factorization of $N$ given by $\mathsf{sk}_0 = (p', q')$,

- $\widehat{\mathsf{PEG.Dec}}$: given $c = (c_0, c_1) \in \mathbb{Z}^2_{N^2}$, $\mathsf{sk}$ and $\mathsf{sk}_0 = (p', q')$, return $\perp$ when $[c_0^{2p'q'}]_{N^2} \neq 1$ and, otherwise, return $\mathsf{PEG.Dec}(\mathsf{sk}, c)$.

□

**Proposition 22.** *$\mathsf{PEG}_z$ is a sGPKE under the DCR assumption.*

*Proof.* From its definition (Sect. 2.5) $\mathsf{PEG}_z$ uses PEG as its GPKE and, from Lemma 21 above, that scheme is indeed a GPKE under the DCR assumption.

$\mathsf{PEG}_z.\mathcal{P} = [\![0, B]\!]$, with $2^\lambda \leq B = 2^t < \lfloor 2^{-\lambda} N \rfloor$, hence $2^\lambda |\mathsf{PEG}_z.\mathcal{P}| \leq |\mathsf{PEG}.G| = N$.

From Sect. 2.5 (p. 12), $\mathsf{PEG}_z.\mathsf{Dec}$ can equivalently be written as,

- $\mathsf{PEG}_z.\mathsf{ImVer}$: given $\mathsf{PEG}_z.\mathsf{sk}$ and $\mathsf{ct} \in \mathbb{Z}^2_{N^2}$, if $\mathsf{PEG.Dec}(\mathsf{sk}, \mathsf{ct}) = \perp$, then return False. Otherwise, let $\mu = \mathsf{PEG.Dec}(\mathsf{sk}, \mathsf{ct})$, if $\mu > B$ then return False and otherwise return True.

– $\mathsf{PEG}_z.\mathsf{Dec}$: given $\mathsf{PEG}_z.\mathsf{sk}$ and $\mathsf{ct} \in \mathbb{Z}_{N^2}^2$, if $\mathsf{PEG}_z.\mathsf{ImVer}(\mathsf{sk}, \mathsf{ct}) = \mathsf{False}$, then return $\perp$. Otherwise, return $\mathsf{PEG}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.

To establish the claim, there remains to prove that $\mathsf{PEG}_z$ satisfies Eq. (7).

First recall that $\mathsf{sk} = x$ is picked uniformly in $[\![0, N(N-1)/4]\!]$. Since, by Lemma 6, the statistical distance between $\{[x]_{Np'q'} : x \xleftarrow{\$} [\![0, N(N-1)/4]\!]\}$ and $\{x \xleftarrow{\$} \mathbb{Z}_{Np'q'}\}$ is negligible in $\lambda$, $x$ is indistinguishable from uniform over $\mathbb{Z}_{Np'q'}$. Furthermore, $[x]_N$ is independent from $[x]_{p'q'}$. Now,

– In $\mathsf{pk} = (N, g, h)$, $h = [g^{4Nx}]$ is the only term that depends on $x$ and it only depends on $[x]_{p'q'}$.
– Additionally, for any ciphertext $\mathsf{ct} \in \mathbb{Z}_{N^2}^2$, $\mathsf{PEG}.\widehat{\mathsf{Dec}}(\mathsf{ct})$ also depends only on $[x]_{p'q'}$.

Hence, for any $(\mathsf{pk}, \mathsf{sk})$ and any $\mathcal{A}^{\mathsf{PEG}_z.\widehat{\mathsf{Dec}}(\mathsf{sk},\mathsf{sk}_0,\cdot)}(\mathsf{pk})$, $\mathcal{A}$'s view is independent from $[x]_N$. Assume the adversary against (7) submits a ciphertext in $H^*$, i.e. of the form,

$$\mathsf{ct} = ((1+N)^{a_0} g^{2Nr}, (1+N)^{a_1} h^r),$$

for some $a_0, a_1, r$. If $\mathsf{ct} \notin H$, then $a_0 \neq 0$ and $\mathsf{PEG}.\mathsf{Dec}(\mathsf{ct})$ returns $m = [a_1 - 2xa_0]_N$. We know that $[x]_N$ is independent from the adversary's view. This implies that $[a_1 - 2xa_0]_N$ is uniformly distributed in $G$ from $\mathcal{A}$'s view, and is in $[\![0, B]\!]$ with probability $\mathrm{neg}(\lambda)$, as $B \leq N.2^{-\lambda}$. Note that if $a_0$ is not invertible in $\mathbb{Z}_N$, let say a multiple of $p$, then $[a_1 - 2xa_0]_N$ also appears uniformly distributed in $[a_1 + p\mathbb{Z}_q]_N$, and is still in $[\![0, B]\!]$ with probability $\frac{B/p}{N/p} = \frac{B}{N}$. □

The following result then directly follows from Prop. 14 and Prop. 22 above.

**Proposition 23.** *$PEG_z$ is CCA1-secure under the DCR assumption.*

### A.2 Cramer-Shoup Lite

We now turn our attention to The Cramer-Shoup Lite scheme, as defined in [17]. Following our sGPKE construction blueprint (Def. 11), we first define the following GPKE scheme:

– $\mathsf{CSL}_0.\mathsf{KeyGen}$: on input $1^\lambda$, choose a group $(\mathcal{G}, \cdot)$ of prime order $q >> 2^\lambda$ and $g$ a generator of $\mathcal{G}$, then pick $x_1, x_2, y_1, y_2, \alpha \xleftarrow{\$} [\![0, q-1]\!]$ and compute $h = g^\alpha$, $A = g^{x_1} h^{x_2}$ and $B = g^{y_1} h^{y_2}$. Then $\mathsf{pk} = (q, g, h, A, B)$ and $\mathsf{sk} = (x_1, x_2, y_1, y_2)$.
– $\mathsf{CSL}_0.\mathsf{Enc}$: given $\mathsf{pk}$ and $m, m' \in \mathcal{G}$, choose $r \xleftarrow{\$} [\![0, q-1]\!]$ and return $c = (g^r, h^r, mA^r, m'B^r)$.
– $\mathsf{CSL}_0.\mathsf{Dec}$: given $\mathsf{sk}$ and $c = (c_0, c_1, c_2, c_3)$, return $m = c_2/(c_0^{x_1} c_1^{x_2})$ and $m' = c_3/(c_0^{y_1} c_1^{y_2})$.

Then, we define the associated sGPKE, CSL, as follows,

– $\mathsf{CSL}.\mathsf{KeyGen}$: on input $1^\lambda$, run $\mathsf{CSL}_0.\mathsf{KeyGen}(1^\lambda)$.

– CSL.Enc: given $\mathsf{pk}$ and $m \in \mathcal{G}$, run $\mathsf{CSL}_0.\mathsf{Enc}(\mathsf{pk}, m, m' = 1)$.
– CSL.Dec: given $\mathsf{sk}$ and $c = (c_0, c_1, c_2, c_3)$, run $\mathsf{CSL}_0.\mathsf{Dec}(\mathsf{sk}, c)$ to get $(m, m')$ and if $m' = 1$ return $m$, otherwise return $\perp$.

**Lemma 24.** *$CSL_0$ is a GPKE under the DDH assumption.*

*Proof. Property 1.* Following the definition of $\mathsf{CSL}_0$ above,

– $\mathsf{KeyGen}_0$: on input $1^\lambda$, choose a group $(\mathcal{G}, \cdot)$ of prime order $q >> 2^\lambda$ and $g$ a generator of $\mathcal{G}$, then pick $\alpha \xleftarrow{\$} [\![0, q-1]\!]$ and compute $h = g^\alpha$. Then $\mathsf{pk}_0 = (q, g, h)$ and $\mathsf{sk}_0 = \alpha$.
– $\mathsf{KeyGen}_1$: on input $\mathsf{pk}_0 = (q, g, h)$, pick $x_1, x_2, y_1, y_2 \xleftarrow{\$} [\![0, q-1]\!]$ and compute $A = g^{x_1} h^{x_2}$ and $B = g^{y_1} h^{y_2}$. Then $\mathsf{pk} = (q, g, h, A, B)$ and $\mathsf{sk} = (x_1, x_2, y_1, y_2)$.

*Property 2.* $G = \mathcal{G}^2$ and,

$$H = \{(c_0, c_1, c_2, c_3) \in \mathcal{G}^4 : c_1 = c_0^\alpha\}.$$

For every key pair $(\mathsf{pk}, \mathsf{sk})$, let $\mathsf{ct}$ be a element of $H$, then,

$$\mathsf{ct} = (g^r, g^{\alpha r}, mA^r, m'B^r) = \mathsf{CSL}_0.\mathsf{Enc}(\mathsf{pk}, m, m'; r),$$

for $r \in [\![0, q-1]\!]$ and $m, m' \in \mathcal{G}$. Hence, $\mathsf{CSL}_0.\mathsf{Enc}(\mathsf{pk}, G; \mathsf{COIN}) = H$. Furthermore, the uniform distributions over $G$ and $H$ are trivially efficiently sampleable.

*Property 3.* Given $\mathsf{pk}_0 = (q, g, h)$, $\mathsf{CSL.KeyGen}_1(\mathsf{pk}_0)$ outputs,

$$(\mathsf{pk}, \mathsf{sk}) = ((q, g, h, A, B), (x_1, x_2, y_1, y_2)).$$

Let $H' = \langle (1, g, g^{x_2}, g^{y_2}) \rangle$. The indistinguishability between $H^* = H \cdot H'$ and $H$, follows by the DDH assumption, as it is equivalent to distinguish between,

$$\{(g^\alpha, g^r, g^{\alpha r}) : r \xleftarrow{\$} [\![0, q-1]\!]\} \text{ and } \{(g^\alpha, g^r, g^{r'}) : r, r' \xleftarrow{\$} [\![0, q-1]\!]\},$$

without knowledge of $sk_0 = \alpha$. Again, the uniform distributions over $H^*$ and $H'$ can trivially be sampled efficiently. Remark that $\mathcal{C} = H^*$.

*Property 4.* $G = \mathcal{G}^2$ with $\mathcal{G}$ of order $q >> 2^\lambda$.

*Property 5.* Let $m \in \mathcal{G}$, we can trivially verify that $\mathsf{CSL}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{CSL}_0.\mathsf{Enc}(\mathsf{pk}, m)) = m$. Further, let $\mathsf{ct} = (c_0, c_1, c_2, c_3) \in H^*$ and $\mathsf{ct}' = (c_0', c_1', c_2', c_3') \in H^*$, with

$$\begin{aligned}
\mathsf{CSL}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) &= ((c_2/(c_0^{x_1} c_1^{x_2})), (c_3/(c_0^{y_1} c_1^{y_2}))) \\
&= ((\alpha_0 (c_0^{x_1} c_1^{x_2})/(c_0^{x_1} c_1^{x_2})), (\alpha_1 (c_0^{y_1} c_1^{y_2})/(c_0^{y_1} c_1^{y_2}))) \\
&= (\alpha_0, \alpha_1),
\end{aligned}$$

for some $\alpha_0, \alpha_1 \in \mathcal{G}$, and likewise for $\mathsf{ct}'$. Then $\alpha_0 (c_0^{x_1} c_1^{x_2}) \alpha_0' (c_0'^{x_1} c_1'^{x_2}) = \alpha_0 \alpha_0' ((c_0 c_0')^{x_1} (c_1 c_1')^{x_2})$ implies property 5.

*Property 6.* Let $\mathsf{ct} = \mathsf{CSL}_0.\mathsf{Enc}(\mathsf{pk}, m_0, m_1; r)$ and $\mathsf{ct}' = \mathsf{CSL}_0.\mathsf{Enc}(\mathsf{pk}, m_0', m_1'; r')$,

$$\begin{aligned}
\mathsf{ct} \cdot \mathsf{ct}' &= (g^r, h^r, A^r \cdot m_0, B^r \cdot m_1) \cdot (g^{r'}, h^{r'}, A^{r'} \cdot m_0', B^{r'} \cdot m_1'), \\
&= (g^{r+r'}, h^{r+r'}, A^{r+r'} \cdot m_0 m_0', B^{r+r'} \cdot m_1 m_1'), \\
&= (g^{r''}, h^{r''}, A^{r''} \cdot m_0 m_0', B^{r''} \cdot m_1 m_1'),
\end{aligned}$$

with either or both $r$ and $r'$ uniformly random in $[\![0, q-1]\!]$, $r'' = r + r'$ is also uniform in $[\![0, q-1]\!]$.

*Property 7.* Given a key pair $(\mathsf{pk}, \mathsf{sk}) = ((q, g, h, A, B), (x_1, x_2, y_1, y_2))$, $\mathsf{CSL}_0.\mathsf{Dec} : H^* \longrightarrow G$ takes $(c_0, c_1, c_2, c_3)$ and returns $(m, m') \in G$ such that $m = c_2/(c_0^{x_1} c_1^{x_2})$ and $m' = c_3/(c_0^{y_1} c_1^{y_2})$. Now consider an uniformly picked $\mathfrak{h}' = (1, g^a, g^{ax_2}, g^{ay_2}) \in H'$ for $a \in [\![0, q-1]\!]$ and $\mu = (g^b, g^c) \xleftarrow{\$} G$. Since $A$ and $B$ only depend on $x_1 + \alpha x_2$ and $y_1 + \alpha y_2$ respectively, we can compute another valid key pair $(\mathsf{pk}, \widehat{\mathsf{sk}}) = ((q, g, h, A, B), (\hat{x}_1, \hat{x}_2, \hat{y}_1, \hat{y}_2))$, such that $\hat{x}_1 + \alpha \hat{x}_2 = x_1 + \alpha x_2$ and $\hat{y}_1 + \alpha \hat{y}_2 = y_1 + \alpha y_2$. Then, we have $\mathsf{CSL}_0.\mathsf{Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = (g^{(x_2 - \hat{x}_2)a}, g^{(y_2 - \hat{y}_2)a})$, we can choose $\widehat{\mathsf{sk}}$ such that $\hat{x}_2 = x_2 - ba^{-1}$ and $\hat{y}_2 = y_2 - ca^{-1}$ to get $\mathsf{CSL}_0.\mathsf{Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = \mu$.

*Property 8.* For $c = (c_0, c_1, c_2, c_3) \in H$, by definition $c_1 = c_0^{\alpha}$, We can thus define an alternative decryption function that uses $\mathsf{sk}_0 = \alpha$,

- $\mathsf{CSL}_0.\widehat{\mathsf{Dec}}$: given $c = (c_0, c_1, c_2, c_3) \in \mathcal{G}^4$, $\mathsf{sk}$ and $\mathsf{sk}_0 = \alpha$, return $\perp$ when $c_0^{\alpha} \neq c_1$ and return $m = c_2/(c_0^{x_1 + \alpha x_2})$ and $m' = c_3/(c_0^{y_1 + \alpha y_2})$ otherwise.

It is obvious that $\mathsf{CSL}_0.\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, \mathsf{ct}) = \mathsf{CSL}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ for any $\mathsf{ct} \in H$. □

**Proposition 25.** *CSL is a sGPKE under the DDH assumption.*

*Proof.* From its definition CSL uses $\mathsf{CSL}_0$ as its GPKE and, from the Lemma above, that scheme is indeed a GPKE under the DDH assumption.

$\mathsf{CSL}.\mathcal{P} = \mathcal{G}$, hence $2^{\lambda} |\mathsf{CSL}.\mathcal{P}| \leq |\mathsf{CSL}_0.G| = \mathcal{G}^2$.

From its definition, CSL.Dec can equivalently be written as,

- CSL.ImVer: given CSL.sk and $\mathsf{ct} \in \mathcal{G}^4$, if $\mathsf{CSL}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = (m, m')$ with $m' \neq 1$, then return False. Otherwise, return True.
- CSL.Dec: given CSL.sk and $\mathsf{ct} \in \mathcal{G}^4$, if $\mathsf{CSL}.\mathsf{ImVer}(\mathsf{sk}, \mathsf{ct}) = \mathsf{False}$, then return $\perp$. Otherwise, return $m$ from $(m, m') = \mathsf{CSL}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.

To establish the claim, there remain to prove that CSL satisfies Eq. (7).

First recall that $\mathsf{sk} = (x_1, x_2, y_1, y_2)$ is picked uniformly in $[\![0, q-1]\!]^4$. Now,

- In $\mathsf{pk} = (q, g, h, A, B)$, $A = g^{x_1} h^{x_2}$ and $B = g^{y_1} h^{y_2}$ are the only term that depends on $\mathsf{sk}$ and it only depends on $x_1 + \alpha x_2$ and $y_1 + \alpha y_2$.
- Additionally, for any ciphertext $\mathsf{ct} \in \mathcal{G}^4$, $\mathsf{CSL}_0.\widehat{\mathsf{Dec}}(\mathsf{ct})$ also depends only on $x_1 + \alpha x_2$ and $y_1 + \alpha y_2$.

Hence, for any $(\mathsf{pk}, \mathsf{sk})$ and any $\mathcal{A}^{\mathsf{CSL}.\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, .)}(\mathsf{pk})$, $\mathcal{A}$'s view is independent from $x_2$ and $y_2$. Assume the adversary against (7) submits a ciphertext in $H^*$, i.e. of the form,

$$\mathsf{ct} = (g^r, h^{r'}, a_0 \cdot A^r, a_1 \cdot B^r),$$

for some $a_0, a_1, r$ and $r'$. If $\mathsf{ct} \notin H$, then $r \neq r'$ and $\mathsf{CSL}_0.\mathsf{Dec}(\mathsf{ct})$ returns $(m, m')$ with especially $m' = a_1 \cdot h^{y_2(r - r')}$. We know that $y_2$ is independent from the adversary's view. This implies that $a_1 \cdot h^{y_2(r-r')}$ is uniformly distributed in $\mathcal{G}$ from $\mathcal{A}$'s view, and is equal to 1 with probability $\mathrm{neg}(\lambda)$, as $2^{\lambda} << q$. □

The CCA1 security of CSL under DDH, which of course was already established in [17] by a seminal dedicated proof, thus also follows from Prop. 25 and 14.

### A.3 Damgård's additively homomorphic ElGamal

We now turn to the Damgård-ElGamal scheme, as described in [27,18]. Following our sGPKE blueprint (Def. 11), we first define the following GPKE scheme, $\mathsf{DEG}_0$:

- $\mathsf{DEG}_0.\mathsf{KeyGen}$: on input $1^\lambda$, choose a group $(\mathcal{G},\cdot)$ of prime order $q >> 2^\lambda$ and $g$ generator of $\mathcal{G}$, then pick $x_1, x_2, \alpha \xleftarrow{\$} [\![0, q-1]\!]$ and compute $h = g^\alpha$ and $A = g^{x_1}h^{x_2}$. Then $\mathsf{pk} = (q, g, h, A)$ and $\mathsf{sk} = (x_1, x_2)$.
- $\mathsf{DEG}_0.\mathsf{Enc}$: given $\mathsf{pk}$ and $m \in \mathcal{G}$, choose $r \xleftarrow{\$} [\![0, q-1]\!]$ and return $c = (c_0, c_1, c_2) = (g^r, h^r, g^m A^r)$.
- $\mathsf{DEG}_0.\mathsf{Dec}$: given $\mathsf{sk}$ and $c = (c_0, c_1, c_2)$, compute $M = c_2/(c_0^{x_1} c_1^{x_2})$. If there exists an integer $m \in [\![0, q-1]\!]$ such that $M = g^m$, return $m$. Otherwise return $\perp$.[14]

Then, we define the associated sGPKE, DEG, as follows,

- $\mathsf{DEG}.\mathsf{KeyGen}$: on input $1^\lambda$ and $t = O(\log(\lambda))$, run $\mathsf{DEG}_0.\mathsf{KeyGen}(1^\lambda)$ and return $\mathsf{pk} = (\mathsf{DEG}_0.\mathsf{pk}, B = 2^t)$ and $\mathsf{sk} = \mathsf{DEG}_0.\mathsf{sk}$. The plaintext domain is thus $[\![0, B]\!]$.
- $\mathsf{DEG}.\mathsf{Enc}$: given $\mathsf{pk}$ and $m \in [\![0, B]\!]$, run $\mathsf{DEG}_0.\mathsf{Enc}(\mathsf{pk}, m)$.
- $\mathsf{DEG}.\mathsf{Dec}$: given $\mathsf{sk}$ and $c = (c_0, c_1, c_2)$, compute $M = c_2/(c_0^{x_1} c_1^{x_2})$. If there exists an integer $m \in [\![0, B]\!]$ such that $M = g^m$, return $m$. Otherwise return $\perp$.

**Lemma 26.** *$\mathsf{DEG}_0$ is a GPKE under the DDH assumption.*

*Proof.* *Property 1.* Following the definition of $\mathsf{DEG}_0$ above,

- $\mathsf{KeyGen}_0$: Given a parameter $\lambda$, choose a group $(\mathcal{G},\cdot)$ of prime order $q >> 2^\lambda$ and $g$ generator of $\mathcal{G}$, then pick $\alpha \xleftarrow{\$} [\![0, q-1]\!]$ and compute $h = g^\alpha$. Then $\mathsf{pk}_0 = (q, g, h)$ and $\mathsf{sk}_0 = (\alpha)$.
- $\mathsf{KeyGen}_1$: on input $\mathsf{pk}_0 = (q, g, h)$, pick $x_1, x_2 \xleftarrow{\$} [\![0, q-1]\!]$ and compute $A = g^{x_1}h^{x_2}$. Then $\mathsf{pk} = (q, g, h, A)$ and $\mathsf{sk} = (x_1, x_2)$.

*Property 2.* $G = [\![0, q-1]\!]$ and

$$H = \{(c_0, c_1, c_2) \in \mathcal{G}^3 : c_1 = c_0^\alpha\}.$$

For every key pair $(\mathsf{pk}, \mathsf{sk})$, let $\mathsf{ct}$ be a element of $H$, then

$$\mathsf{ct} = (g^r, g^{r\alpha}, g^m A^r) = \mathsf{DEG}_0.\mathsf{Enc}(\mathsf{pk}, m; r)$$

for $r, m \in [\![0, q-1]\!]$. Hence, $\mathsf{DEG}_0.\mathsf{Enc}(\mathsf{pk}, G; \mathsf{COIN}) = H$. Furthermore, the uniform distributions over $G$ and $H$ are trivially efficiently sampleable.

*Property 3.* Given $\mathsf{pk}_0 = (q, g, h)$, $\mathsf{DEG}.\mathsf{KeyGen}_1(\mathsf{pk}_0)$ outputs,

$$(\mathsf{pk}, \mathsf{sk}) = ((q, g, h, A), (x_1, x_2)).$$

[14] Remark that $\mathsf{DEG}_0.\mathsf{Dec}$ is not a PPT algorithm, but this does not yield any issue with the proof of security of DEG as $\mathsf{DEG}.\mathsf{Dec}$ is eventually PPT.

Let $H' = \langle(1, g, g^{x_2})\rangle$. The indistinguishability between $H^* = H \cdot H'$ and $H$, follows by the DDH assumption, as it is equivalent to distinguish between,

$$\{(g^\alpha, g^r, g^{\alpha r}) : r \xleftarrow{\$} [\![0, q-1]\!]\} \text{ and } \{(g^\alpha, g^r, g^{r'}) : r, r' \xleftarrow{\$} [\![0, q-1]\!]\},$$

without knowledge of $sk_0 = \alpha$. Again, the uniform distributions over $H^*$ and $H'$ can trivially be sampled efficiently. Remark that $\mathcal{C} = H^*$.

*Property 4.* $G = [\![0, q-1]\!]$ with $q >> 2^\lambda$.

*Property 5.* Let $m \in \mathcal{G}$, we can trivially verify that $\mathsf{DEG}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{DEG}_0.\mathsf{Enc}(\mathsf{pk}, m)) = m$. Further, let $\mathsf{ct} = (c_0, c_1, c_2) \in H^*$ and $\mathsf{ct}' = (c_0', c_1', c_2') \in H^*$, with

$$\begin{aligned}\mathsf{DEG}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) &= \log_g(c_2/(c_0^{x_1} c_1^{x_2})) \\ &= \log_g(g^{\alpha_0}(c_0^{x_1} c_1^{x_2})/(c_0^{x_1} c_1^{x_2})) \\ &= \alpha_0,\end{aligned}$$

for some $\alpha_0 \in \mathcal{G}$, and likewise for $\mathsf{ct}'$. Then $\log_g(g^{\alpha_0+\alpha_0'}((c_0 c_0')^{x_1}(c_1 c_1')^{x_2})) = \log_g(g^{\alpha_0}(c_0^{x_1} c_1^{x_2}) g^{\alpha_0'}(c_0'^{x_1} c_1'^{x_2})) = \log_g(g^{\alpha_0}(c_0^{x_1} c_1^{x_2})) + \log_g(g^{\alpha_0'}(c_0'^{x_1} c_1'^{x_2}))$ implies property 5.

*Property 6.* Let $\mathsf{ct} = \mathsf{DEG}_0.\mathsf{Enc}(\mathsf{pk}, m; r)$ and $\mathsf{ct}' = \mathsf{DEG}_0.\mathsf{Enc}(\mathsf{pk}, m'; r')$,

$$\begin{aligned}\mathsf{ct} \cdot \mathsf{ct}' &= (g^r, h^r, g^m A^r) \cdot (g^{r'}, h^{r'}, g^{m'} A^{r'}), \\ &= (g^{r'+r}, h^{r'+r}, g^{m'+m} A^{r'+r}), \\ &= (g^{r''}, h^{r''}, g^{m+m'} A^{r''}),\end{aligned}$$

with either or both $r$ and $r'$ uniformly random in $[\![0, q-1]\!]$, $r'' = r + r'$ is also uniform in $[\![0, q-1]\!]$.

*Property 7.* Given a key pair $(\mathsf{pk}, \mathsf{sk}) = ((q, g, h, A), (x_1, x_2))$, $\mathsf{DEG}_0.\mathsf{Dec} : H^* \longrightarrow G$ takes $(c_0, c_1, c_2)$ and returns $m \in G$ such that $g^m = c_2/(c_0^{x_1} c_1^{x_2})$. Now consider an uniformly picked $\mathfrak{h}' = (1, g^a, g^{ax_2}) \in H'$ for $a \in [\![0, q-1]\!]$ and $\mu \xleftarrow{\$} G$. Since $A$ only depends on $x_1 + \alpha x_2$, we can compute another valid key pair $(\mathsf{pk}, \widehat{\mathsf{sk}}) = ((q, g, h, A), (\hat{x}_1, \hat{x}_2))$, such that $\hat{x}_1 + \alpha \hat{x}_2 = x_1 + \alpha x_2$. Then, we have $\mathsf{DEG}_0.\mathsf{Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = (x_2 - \hat{x}_2)a$, we can choose $\widehat{\mathsf{sk}}$ such that $\hat{x}_2 = x_2 - \mu a^{-1}$ to get $\mathsf{DEG}_0.\mathsf{Dec}(\widehat{\mathsf{sk}}, \mathfrak{h}') = \mu$.

*Property 8.* For $c = (c_0, c_1, c_2) \in H$, by definition $c_1 = c_0^\alpha$, We can thus define an alternative decryption function that uses $\mathsf{sk}_0 = \alpha$,

- $\mathsf{DEG}_0.\widehat{\mathsf{Dec}}$: given $c = (c_0, c_1, c_2) \in \mathcal{G}^3$, $\mathsf{sk}$ and $\mathsf{sk}_0 = \alpha$, return $\perp$ when $c_0^\alpha \neq c_1$ and return $m = c_2/c_0^{x_1 + \alpha x_2}$ otherwise.

It is obvious that $\mathsf{DEG}_0.\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, \mathsf{ct}) = \mathsf{DEG}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ for any $\mathsf{ct} \in H$. □

**Proposition 27.** *DEG is a sGPKE under the DDH assumption.*

*Proof.* From its definition DEG uses $\mathsf{DEG}_0$ as its GPKE and, from the Lemma above, that scheme is indeed a GPKE under the DDH assumption.

$\mathsf{DEG}.\mathcal{P} = [\![0, B]\!]$, hence $2^\lambda |\mathsf{DEG}.\mathcal{P}| \leq |\mathsf{DEG}_0.G| = \mathcal{G}$ as $q >> 2^\lambda$ and $t = O(\log(\lambda))$.

From its definition, DEG.Dec can equivalently be written as,

– $\mathsf{DEG.ImVer}$: given $\mathsf{DEG.sk}$ and $\mathsf{ct} \in \mathcal{G}^3$, if $\mathsf{DEG}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = m$ with $m \notin [\![0, B]\!]$, then return $\mathsf{False}$. Otherwise, return $\mathsf{True}$.
– $\mathsf{DEG.Dec}$: given $\mathsf{DEG.sk}$ and $\mathsf{ct} \in \mathcal{G}^3$, if $\mathsf{DEG.ImVer}(\mathsf{sk}, \mathsf{ct}) = \mathsf{False}$, then return $\perp$. Otherwise, return $\mathsf{DEG}_0.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.

To establish the claim, there remain to prove that $\mathsf{DEG}$ satisfies Eq. (7).

First recall that $\mathsf{sk} = (x_1, x_2)$ is picked uniformly in $[\![0, q-1]\!]^2$. Now,

– In $\mathsf{pk} = (q, g, h, A)$, $A = g^{x_1} h^{x_2}$ is the only term that depends on $\mathsf{sk}$ and it only depends on $x_1 + \alpha x_2$.
– Additionally, for any ciphertext $\mathsf{ct} \in \mathcal{G}^3$, $\mathsf{DEG}_0.\widehat{\mathsf{Dec}}(\mathsf{ct})$ also depends only on $x_1 + \alpha x_2$.

Hence, for any $(\mathsf{pk}, \mathsf{sk})$ and any $\mathcal{A}^{\mathsf{DEG}.\widehat{\mathsf{Dec}}(\mathsf{sk}, \mathsf{sk}_0, .)}(\mathsf{pk})$, $\mathcal{A}$'s view is independent from $x_2$. Assume the adversary against (7) submits a ciphertext in $H^*$, i.e. of the form,

$$\mathsf{ct} = (g^r, h^{r'}, g^{a_0} A^r),$$

for some $a_0, r$ and $r'$. If $\mathsf{ct} \notin H$, then $r \neq r'$ and $\mathsf{DEG}_0.\mathsf{Dec}(\mathsf{ct})$ returns $m$ with $m = a_0 + x_2(r - r')$. We know that $x_2$ is independent from the adversary's view. This implies that $a_0 + x_2(r - r')$ is uniformly distributed in $\mathcal{G}$ from $\mathcal{A}$'s view, and is in $[\![0, B]\!]$ with probability $\mathrm{neg}(\lambda)$. □

The CCA1 security of $\mathsf{DEG}$ under DDH, which was recently already established in [27] by a delicate proof, thus also follows from Prop. 27 and 14.

## B Additional proofs

### B.1 CPA security of $\mathsf{PEG}_2$

In this section, we prove the CPA security of $\mathsf{PEG}_2$ (Sect. 3.1). This result is easy to establish under the assumption that $\mathsf{PEG}$ is CPA-secure. Since the latter holds under the DCR assumption, the CPA security of $\mathsf{PEG}_2$ also holds under the same assumption.

**Proposition 28.** *$\mathsf{PEG}_2$ is CPA-secure under the assumption that $\mathsf{PEG}$ is.*

*Proof.* The proof proceeds by a quite conventional game hopping argument. Let CPA-$\gamma$ be the CPA game against $\mathsf{PEG}_2$ where the challenger picks the bit $\gamma$. Because, $\mathsf{PEG}_2$ has two message slots, i.e. its plaintext domain is $\mathbb{Z}_N^2$, the adversary submits $(m_0, m_0') \in \mathbb{Z}_N^2$ and $(m_1, m_1') \in \mathbb{Z}_N^2$ as a challenge request and gets $\mathsf{PEG}_2.\mathsf{Enc}(\mathsf{pk}, (m_\gamma, m_\gamma'))$ in return.

*First game hop.* Let $G_0$ be the CPA-0 game against $\mathsf{PEG}_2$ and let $G_1$ be the same as $G_0$ except that the challenge ciphertext is,

$$([g^{2Nr}]_{N^2}, [(1+N)^{m_0} h^r]_{N^2}, [(1+N)^{m_1'} h'^r]_{N^2}),$$

rather than (in $G_0$),

$$([g^{2Nr}]_{N^2}, [(1+N)^{m_0} h^r]_{N^2}, [(1+N)^{m_0'} h'^r]_{N^2}).$$

We now prove that the two games cannot be distinguished under the assumption that $\mathsf{PEG}$ is CPA-secure. Indeed, let $\mathcal{A}_0$ be an adversary that can distinguish between $G_0$ and $G_1$, then we can build as follows an adversary $\mathcal{B}_0$ against the CPA security of $\mathsf{PEG}$ that achieves at least $\mathcal{A}_0$'s advantage. First $\mathcal{B}_0$ gets $(N, g, h')$ from its CPA challenger against $\mathsf{PEG}$ (so the challenger operates slots 0 and 2 in the ciphertexts under $\mathsf{PEG}_2$), then it picks $x$ uniformly in $[\![0, N(N-1)/4]\!]$ (following $\mathsf{PEG}_2.\mathsf{KeyGen}$), computes $h = [g^{4Nx}]_{N^2}$ and forwards $(N, g, h, h')$ to $\mathcal{A}_0$. Because we are in the public key setting, the reduction only has to handle challenge requests. It does this as follows:

– When $\mathcal{A}_0$ issues a challenge request over $(m_0, m_0'), (m_1, m_1') \in \mathbb{Z}_N^2$, $\mathcal{B}_0$ first issues a challenge request over messages $m_0'$ and $m_1'$ towards its challenger to get $\mathsf{PEG}$ ciphertext,

$$(c_0^*, c_2^*) = ([g^{2Nr}]_{N^2}, [(1+N)^{m'_\gamma} h'^r]_{N^2}),$$

for unknown $\gamma$ and $r$. Then, the reduction computes,

$$c_1^* = (1+N)^{m_0} c_0^{*2x} = (1+N)^{m_0} g^{4Nrx} = (1+N)^{m_0} h^r,$$

and returns $(c_0^*, c_1^*, c_2^*)$ to $\mathcal{A}_0$.

Finally, when $\mathcal{A}_0$ outputs a guess $\hat{\gamma}$, $\mathcal{B}_0$ does the same. Clearly, $\mathcal{B}_0$ perfectly simulates game $G_0$ when $\gamma = 0$ and game $G_1$ when $\gamma = 1$. Therefore,

$$\mathsf{Adv}^{G_0/G_1}(\mathcal{A}_0) \leq \mathsf{Adv}^{\mathrm{CPA}}_{\mathsf{PEG}}(\mathcal{B}_0). \tag{12}$$

*Second game hop.* We now consider game $G_2$ which is the same as $G_1$ except that the challenge ciphertext is now,

$$([g^{2Nr}]_{N^2}, [(1+N)^{m_1} h^r]_{N^2}, [(1+N)^{m'_1} h'^r]_{N^2}),$$

rather than (in $G_1$),

$$([g^{2Nr}]_{N^2}, [(1+N)^{m_0} h^r]_{N^2}, [(1+N)^{m'_1} h'^r]_{N^2}).$$

We can also prove that the two games cannot be distinguished under the assumption that $\mathsf{PEG}$ is CPA-secure. Indeed, let $\mathcal{A}_1$ be an adversary that can distinguish between $G_1$ and $G_2$, then we can build an adversary $\mathcal{B}_1$ against the CPA security of $\mathsf{PEG}$ that achieves at least $\mathcal{A}_1$'s advantage, with a reduction similar to that in the first game hop. First $\mathcal{B}_1$ gets $(N, g, h)$ from its CPA challenger against $\mathsf{PEG}$ (so, this time, the challenger operates slots 0 and 1 in the ciphertexts under $\mathsf{PEG}_2$), then it picks $x'$ uniformly in $[\![0, N(N-1)/4]\!]$, computes $h' = [g^{4Nx'}]_{N^2}$ and forwards $(N, g, h, h')$ to $\mathcal{A}_1$. Then $\mathcal{B}_1$ proceeds as follows:

– When $\mathcal{A}_1$ issues a challenge request over $(m_0, m_0'), (m_1, m_1') \in \mathbb{Z}_N^2$, $\mathcal{B}_1$ first forward the request as is to its challenger to get ciphertext,

$$(c_0^*, c_1^*) = ([g^{2Nr}]_{N^2}, [(1+N)^{m_\gamma} h^r]_{N^2}).$$

for unknown $\gamma$ and $r$. Then, $\mathcal{B}_1$ computes,

$$c_2^* = (1+N)^{m_1'} c_0^{*2x'} = (1+N)^{m_1'} g^{4Nrx'} = (1+N)^{m_1'} h'^r,$$

and finally returns $(c_0^*, c_1^*, c_2^*)$ to $\mathcal{A}_1$.

Again, when $\mathcal{A}_1$ outputs a guess $\hat{\gamma}$, $\mathcal{B}_1$ does the same. Clearly, $\mathcal{B}_1$ perfectly simulates game $G_1$ when $\gamma = 0$ and game $G_2$ when $\gamma = 1$. Therefore,

$$\mathsf{Adv}^{G_0/G_1}(\mathcal{A}_1) \leq \mathsf{Adv}^{\mathrm{CPA}}_{\mathsf{PEG}}(\mathcal{B}_1). \tag{13}$$

Since $G_0$ is CPA-0 and $G_2$ is CPA-1, the claim follows by gathering Eqs. (12) and (13). □

### B.2 Alternative proof of Prop. 8

In this section, we provide an alternative direct proof of Prop. 8 by porting the proof of [27, Theorem 3] to DPEG. We thus closely follows the proof [27, Theorem 3] such that our adaptations can easily be spotted when reading both proofs. It turns out that our adaptations are relatively minor. Interestingly, for DPEG we can rely only on DCR and avoid the additional reliance on the CNI assumption in Libert's original proof for $\mathsf{PEG}_z$ (we further discuss this point in Remark 30, p. 46).

*Proof.* *Game $G_0$.* This is the CCA1 game against DPEG. Recall that in the CCA1 experiment the adversary may perform a single challenge request and decryption requests over arbitrary elements $\mathbb{Z}^3_{N^2}$ until the output of its challenge request is given to it. For the challenge request, the adversary choose $m_0, m_1 \in \mathbb{Z}_n$ and, following Lemma 7, gets ciphertext

$$\mathsf{ct}^* = (c_0^*, c_1^*, c_2^*) = ([g^{2Nr}]_{N^2}, [(1+N)^{m_\gamma} h^r]_{N^2}, [(1+N)^{\xi m_\gamma} h'^r]_{N^2}), \tag{14}$$

for unknown bit $\gamma$, randomness $r \in [\![0, (N-1)/4]\!]$ and $\xi$.

*Game $G_1$.* The first game hop is identical to that of the proof of [27, Theorem 3]: $G_1$ is the same as $G_0$ except that the challenge ciphertext generation is modified to be,

$$\mathsf{ct}^* = (c_0^*, c_1^*, c_2^*) = ([g^{2Nr}]_{N^2}, [(1+N)^{m_\gamma} c_0^{*2x}]_{N^2}, [(1+N)^{\xi m_\gamma} c_0^{*2x'}]_{N^2}), \tag{15}$$

rather than Eq. (14). This clearly does not change anything since $h^r = \left(g^{4Nx}\right)^r = \left(g^{2Nr}\right)^{2x} = c_0^{*2x}$, and similarly so for $h'^r$.

*Game $G_2$.* The second game hop is also identical to that of the proof of [27, Theorem 3]: $G_2$ is the same game as $G_1$ except that the challenge generation is further modified so that $c_0^*$ is now obtained by sampling a random $N$-th residue $z = [z_0^N]_{N^2}$, for some $z_0$ uniformly picked in $\mathbb{Z}_N^*$, and doing,

$$\mathsf{ct}^* = (c_0^*, c_1^*, c_2^*) = ([z^2]_{N^2}, [(1+N)^{m_\gamma} c_0^{*2x}]_{N^2}, [(1+N)^{\xi m_\gamma} c_0^{*2x'}]_{N^2}), \tag{16}$$

rather than Eq. (15). The two games are indistinguishable as the statistical distance between the distribution of $[z^2]_{N^2}$ in Eq. (16) and that of $[g^{2Nr}]_{N^2}$ in Eq. (15) is bounded by $2^{-\lambda}$. This follows from a simple argument that we do not reproduce here [27, Lemma 3].

*Game* $G_3$. The third game hop remains identical to that of the proof of [27, Theorem 3]: $G_3$ is identical to $G_2$ except that $c_0$ in Eq. (16) is no longer an $N$-th residue but generated by picking $z$ uniformly in $\mathbb{Z}^*_{N^2}$. The DCR assumption then implies that $G_2$ and $G_3$ cannot be distinguished. From the definition of the Paillier scheme (App. E) recall that $\{[(1+N)^m g^{2Nr}]_{N^2} : m \in \mathbb{Z}_N, r \in \mathbb{Z}_{4p'q'}\}$ (with $y = [g^2]_{N^2}$ in the notation of App. E) is isomorphic to $\mathbb{Z}^*_{N^2}$. Therefore, $\{[(1+N)^m g^{2Nr}]_{N^2} : m \in \mathbb{Z}_N, r \in \mathbb{Z}_{p'q'}\}$ is isomorphic to $\{z^2 : z \in \mathbb{Z}^*_{N^2}\}$ and $c_0^*$ may be interpreted as a Paillier encryption,

$$c_0^* = [(1+N)^\alpha g^{2N\beta}]_{N^2}, \tag{17}$$

for some $\alpha \in \mathbb{Z}_N$ and $\beta \in \mathbb{Z}_{p'q'}$. Then,

$$c_1^* = [(1+N)^{[m_\gamma + 2\alpha x]_N} g^{[4N\beta x]_{p'q'}}]_{N^2}, \tag{18}$$

and,

$$c_2^* = [(1+N)^{[\xi m_\gamma + 2\alpha x']_N} g^{[4N\beta x']_{p'q'}}]_{N^2}. \tag{19}$$

*Game* $G_4$. Here we depart from the proof of [27, Theorem 3] as, in our case, $G_4$ is identical to $G_3$. We however keep this step such that our game numbering remains consistent with that in the proof of [27, Theorem 3]. See the post proof Remark 30 (p. 46).

*Game* $G_5$. We are still progressing following the proof of [27, Theorem 3]. This time we modify the generation of $x$ which is sampled uniformly in $\mathbb{Z}_{Np'q'}$ (in $G_5$) rather than in $[\![0, N(N-1)/4]\!]$ (in $G_4$). Then, the distinguishing advantage between $G_4$ and $G_5$ is bounded by the statistical distance between $\{[x]_{Np'q'} : x \xleftarrow{\$} [\![0, N(N-1)/4]\!]\}$ and $\{x \xleftarrow{\$} \mathbb{Z}_{Np'q'}\}$ which is negligible in $\lambda$ [27, Lemma 3].

*Game* $G_{5'}$. We slightly deviate from the the proof of [27, Theorem 3] and apply the same modification to the generation of $x'$ which, likewise, is now sampled uniformly in $\mathbb{Z}_{Np'q'}$ (in $G_{5'}$) rather than in $[\![0, N(N-1)/4]\!]$ (in $G_5$). The exact same argument yields a distinguishing advantage between $G_5$ and $G_{5'}$ negligible in the security parameter $\lambda$.

We now adapt the hybrid argument in the proof of [27, Theorem 3]. First, for notational convenience, $G_{5,0}$ is the same game as $G_{5'}$.

*Game* $G_{5,i}$ *(*$i \in [\![1, Q]\!]$*).* Like in the original proof, we modify the key generation by letting $h = g^{4N\beta_x}$ and $h' = g^{4N\beta_{x'}}$ for both $\beta_x$ and $\beta_{x'}$ uniformly drawn in $\mathbb{Z}_{p'q'}$. We then define an alternative secret key $\mathsf{sk}' = (\beta_x, \beta_{x'}, \xi, p, q)$. To reply the $i$ first decryption queries, the following modified decryption algorithm is used:

– $\mathsf{DPEG.Dec}'$: on inputs $(c_0, c_1, c_2)$ and $\mathsf{sk}'$, return $\perp$ if $[c_0^{2p'q'}]_{N^2} \neq 1$. When this is not the case, compute $\mu = [c_1 \cdot c_0^{-2\beta_x}]_{N^2}$ and $\mu' = [c_2 \cdot c_0^{-2\beta_{x'}}]_{N^2}$. If there exists $m, m' \in \mathbb{Z}_N$ such that $\mu = [(1+N)^m]_{N^2}$, $\mu' = [(1+N)^{m'}]_{N^2}$ and $[\xi m]_N = m'$, return $m$. Otherwise return $\perp$.

After the $i$-th decryption query, the challenger picks $\alpha_x$ and $\alpha_{x'}$, both uniformly in $\mathbb{Z}_N$. It then chooses $x$ in $\mathbb{Z}_{Np'q'}$ such that $\alpha_x = [x]_N$ and $\beta_x = [x]_{p'q'}$, and similarly so for $x' \in \mathbb{Z}_{Np'q'}$ i.e. $\alpha_{x'} = [x']_N$ and $\beta_{x'} = [x']_{p'q'}$. Then it uses $\mathsf{sk} = (x, x', \xi)$ and the original decryption function, DPEG.Dec, to answer the remaining $Q - i$ decryption requests.

Lemma 29 implies that no adversary can distinguish between $G_{5,i}$ and $G_{5,i-1}$ so that, by the standard hybrid argument, $G_{5'} = G_{5,0}$ is indistinguishable from the last game in the sequence, $G_{5,Q}$, in which all the adversary's decryption requests are replied to by means of the above modified decryption algorithm.

*Game* $G_6$. This is the last game hop. Game $G_6$ is identical to $G_{5,Q}$ except that the challenge ciphertext is computed as,

$$\begin{aligned} c_0^* &= [(1+N)^{\alpha} g^{2N\beta}]_{N^2}, \\ c_1^* &= [(1+N)^{u} g^{4N\beta[x]_{p'q'}}]_{N^2}, \\ c_2^* &= [(1+N)^{u'} g^{4N\beta[x']_{p'q'}}]_{N^2}, \end{aligned}$$

with $\alpha$, $\beta$, $u$ and $u'$ uniformly picked in $\mathbb{Z}_N$, $\mathbb{Z}_{p'q'}$, $\mathbb{Z}_N$ and $\mathbb{Z}_N$, respectively. Now, remark that in $G_6$ (as in $G_{5,Q}$), all the adversary's decryption requests are replied to by means of the modified decryption algorithm, DPEG.Dec$'$, and, most importantly, by means of $\mathsf{sk}' = (\beta_x, \beta_{x'}, \xi, p, q) = ([x]_{p'q'}, [x']_{p'q'}, \xi, p, q)$. *Hence, the challenger's replies are independent of* $[x]_N$ *and* $[x']_N$ (which are further drawn *after* the last decryption request from the adversary). Now, recall that in $G_{5,Q}$, following Eqs. (17), (18) and (19), the challenge ciphertext is of the form,

$$\begin{aligned} c_0^* &= [(1+N)^{\alpha} g^{2N\beta}]_{N^2}, \\ c_1^* &= [(1+N)^{m_\gamma + 2\alpha[x]_N} g^{[4N\beta x]_{p'q'}}]_{N^2}, \\ c_2^* &= [(1+N)^{\xi m_\gamma + 2\alpha[x']_N} g^{[4N\beta x']_{p'q'}}]_{N^2}, \end{aligned}$$

with $\alpha$ and $\beta$ uniformly picked in $\mathbb{Z}_N$ and $\mathbb{Z}_{p'q'}$, respectively. Since $\gcd(\alpha, N) = 1$ (with overwhelming probability) and $\gcd(2, N) = 1$, the terms $[m_\gamma + 2\alpha[x]_N]_N$ and $[\xi m_\gamma + 2\alpha[x']_N]_N$ are, from $\mathcal{A}$'s view, both indistinguishable from uniform over $\mathbb{Z}_N$. The indistinguishability between $G_{5,Q}$ and $G_6$ thus follows.

The overall claim finally follows from the fact that, in $G_6$, the challenge ciphertext is now independent from $m_\gamma$ and, hence, from the challenge bit $\gamma$. □

The following lemma is a natural adaptation of [27, Lemma 5] to the case of DPEG. Most of the proof is identical to that of the aforementionned lemma and reproduced for self-containedness. Only the end of the argument is slightly modified.

**Lemma 29 (Adaptation of Lemma 5 in [27]).** *For each* $i \in [\![1, Q]\!]$, $G_{5,i}$ *is indinstinguishable from* $G_{5,i-1}$.

*Proof.* From the adversary's view, games $G_{5,i}$ (where the $i$-th decryption request is replied to by means of the *modified* decryption algorithm) and $G_{5,i-1}$

(where the $i$-th decryption request is replied to by means of the *original* decryption algorithm) are indistinguishable unless the $i$-th decryption query involves a ciphertext that is rejected in the former but not in the latter.

For any ciphertext $(c_0, c_1, c_2)$ such that $[c_0^{2p'q'}]_{N^2} = 1$, both algorithms yield the same output since,

$$[c_0^{-2\beta_x}]_{N^2} = [c_0^{-2[x]_{p'q'}}]_{N^2} \text{ and } [c_0^{-2\beta_{x'}}]_{N^2} = [c_0^{-2[x']_{p'q'}}]_{N^2}.$$

Then, on a ciphertext $(c_0, c_1, c_2)$ such that $[c_0^{2p'q'}]_{N^2} \neq 1$, $\mathsf{DPEG.Dec}'$ returns $\perp$. So the bottom line is to assess the probability that the original decryption function, $\mathsf{DPEG.Dec}$, does not return the same. For such a ciphertext, we have,

$$\begin{aligned} c_0 &= [\zeta_0(1+N)^{\alpha_0} g^{2N\beta_0}]_{N^2}, \\ c_1 &= [\zeta_1(1+N)^{\alpha_1} g^{2N\beta_1}]_{N^2}, \\ c_2 &= [\zeta_2(1+N)^{\alpha_2} g^{2N\beta_2}]_{N^2}, \end{aligned}$$

for some elements, $\zeta_0$, $\zeta_1$ and $\zeta_2$, of order at most 2 in $\mathbb{Z}_{N^2}^*$ and arbitrary values $\alpha_0 \in \mathbb{Z}_N \setminus \{0\}$, $\alpha_1 \in \mathbb{Z}_N$, $\alpha_2 \in \mathbb{Z}_N$ as well as $\beta_0$, $\beta_1$ and $\beta_2$ all in $\mathbb{Z}_{p'q'}$. Then, in $G_{5,i-1}$, $\mathsf{DPEG.Dec}$ computes,

$$\begin{aligned} \mu &= c_1 c_0^{-2x} \mod N^2, \\ &= c_1(\zeta_0(1+N)^{\alpha_0} g^{2N\beta_0})^{-2x} \mod N^2, \\ &= c_1(1+N)^{-2\alpha_0[x]_N} g^{-4N\beta_0[x]_{p'q'}} \mod N^2, \\ &= \zeta_1(1+N)^{\alpha_1 - 2\alpha_0[x]_N} g^{-2N(-\beta_1 + 2\beta_0[x]_{p'q'})} \mod N^2, \end{aligned}$$

and, similarly,

$$\begin{aligned} \mu' &= [c_2 c_0^{-2x'}]_{N^2}, \\ &= c_2(1+N)^{-2\alpha_0[x']_N} g^{-4N\beta_0[x']_{p'q'}} \mod N^2, \\ &= \zeta_2(1+N)^{\alpha_2 - 2\alpha_0[x']_N} g^{-2N(-\beta_2 + 2\beta_0[x']_{p'q'})} \mod N^2. \end{aligned}$$

It may then be assumed that $\zeta_1 = \zeta_2 = 1$, that $\beta_1 = 2\beta_0[x]_{p'q'}$ and that $\beta_2 = 2\beta_0[x']_{p'q'}$ since both decryption functions return $\perp$ unless $\mu$ *and* $\mu'$ are such that there exists $m$ and $m'$ in $\mathbb{Z}_N$ with $\mu = [(1+N)^m]_{N^2}$ and $\mu' = [(1+N)^{m'}]_{N^2}$. Additionally, $\alpha_x = [x]_N$ and $\alpha_{x'} = [x']_N$ are picked uniformly in $\mathbb{Z}_N$ *after* $\mathcal{A}$ submitted its $i$-th decryption query over $(c_0, c_1, c_2)$, thus after $\mathcal{A}$'s fixing of $\alpha_0$, $\alpha_1$ and $\alpha_2$. Since $[x]_N$ and $[x']_N$ are respectively independent of $[x]_{p'q'}$ and $[x']_{p'q'}$, $-2\alpha_0[x]_N$ and $-2\alpha_0[x']_N$ are independent from $\mathcal{A}$'s view. To finalize the proof, we have to handle the following cases:

1. $\gcd(\alpha_0, N) = 1$: in this case, $\gcd(2\alpha_0, N) = 1$. It then follows that $[\alpha_1 - 2\alpha_0[x]_N]_N$ and $[\alpha_2 - 2\alpha_0[x']_N]_N$ are both uniform in $\mathbb{Z}_N$ (and independent of each other). For this case, the claim thus follows from the fact that $[\xi(\alpha_1 - 2\alpha_0[x]_N)]_N \neq [\alpha_2 - 2\alpha_0[x']_N]_N$ with overwhelming probability, so $\mathsf{DPEG.Dec}$ also returns $\perp$ at the $i$-th decryption request.

2. $\gcd(\alpha_0, N) > 1$ and $[\xi\alpha_1]_p = [\alpha_2]_p$: we assume wlog that $\alpha_0 = kp$. Then $[2\alpha_0[x]_N]_N$ and $[2\alpha_0[x']_N]_N$ are uniform multiples of $p$ (and independent of each other). For this case, the claim follows from the fact that $[\xi 2\alpha_0[x]_N]_N \neq [2\alpha_0[x']_N]_N$ with overwhelming probability, so $\mathsf{DPEG.Dec}$ also returns $\perp$ at the $i$-th decryption request.
3. $\gcd(\alpha_0, N) > 1$ and $[\xi\alpha_1]_p \neq [\alpha_2]_p$: in this case, $[\xi(\alpha_1 - 2\alpha_0[x]_N)]_N$ and $[\alpha_2 - 2\alpha_0[x']_N]_N$ cannot be equal. So, again, $\mathsf{DPEG.Dec}$ also returns $\perp$ at the $i$-th decryption request.

□

*Remark 30.* In the original proof of [27, Theorem 3], $G_4$ introduces the following modification in the decryption oracle of the challenger. Indeed, $G_4$ is the same as $G_3$ except that decryption requests of the form $(c_0, c_1, c_2)$, where $c_0$ is a *Paillier* encryption of some $\alpha \in \mathbb{Z}_N$ such that $1 < \gcd(\alpha, N) < N$, are rejected. To do so, the challenger runs $\mathsf{Pai.Dec}(c_0)$ (thus, with the knowledge of the factorization of $N = pq$), checks the latter condition and returns $\perp$ whenever it is true. Then, $G_3$ and $G_4$ are indistinguishable by reduction to the Composite Non-Investibility (CNI) assumption [27, Lemma 4] (we recall the CNI assumption in Def. 37, Sect. D.1).

Although the distributions $\{\alpha \xleftarrow{\$} \mathbb{Z}_N\} \times \{\beta \xleftarrow{\$} \mathbb{Z}_{p'q'}\}$ and $\{\alpha \xleftarrow{\$} \mathbb{Z}_N^*\} \times \{\beta \xleftarrow{\$} \mathbb{Z}_{p'q'}\}$ are within a statistical distance negligible in the security parameter, reliance on the CNI assumption eventually occurs in the proof of [27, Lemma 5]. Indeed, due to the condition in $\mathsf{PEG}_z\mathsf{.Dec}$, the final argument in the proof of [27, Lemma 5] works under the restriction that $\gcd(\alpha_0, N) = 1$ over a $c_0$ that is generated by the adversary (which is not restricted to pick it at random).

In contrast, we explicitly deal with the case where $1 < \gcd(\alpha_0, N) < N$ in the proof of Lemma 29 above (cases 2 and 3 at the end of the proof): even when $\alpha_0$ is a multiple of (wlog) $p$, $[2\alpha_0[x]_N]_N$ and $[2\alpha_0[x']_N]_N$ appear (from $\mathcal{A}$'s view) uniformly distributed in $[p\mathbb{Z}_q]_N$ so that $[\xi 2\alpha_0[x]_N]_N = [2\alpha_0[x']_N]_N$ only with negligible probability. †

As a last remark, we think that the proof of [27, Lemma 5] can be modified likewise: when $\alpha_0$ is a multiple of (wlog) $p$, $[\alpha_1 - 2\alpha_0[x]_N]_N$ appears (from $\mathcal{A}$'s view) uniformly distributed in $[\alpha_1 + p\mathbb{Z}_q]_N$, and is still in $[\![0, B]\!]$ with probability $\frac{B/p}{N/p} = \frac{B}{N} \leq \mathrm{neg}(\lambda)$. This allows to remove game $G_4$ in the proof of [27, Theorem 3] and thus avoids reliance on the CNI assumption. In essence, we positively answers B. Libert's intuition: "we do not know if the Composite Non Invertibility assumption is strictly necessary or if it is an artifact of the proof" [27, p. 21].

### B.3 Proof of Prop. 9

*Proof.* The proof starts by one step of game hopping.

*First game hop.* Let $G_0$ be the vCCA game against $\mathsf{DPEG}$ and $G_1$ be the same game as $G_0$ where we modify the challenger such that, when handling a decryption request on ciphertext $c$, the new challenger invokes the LOH extractor

to verify that

$$\mathsf{extract}(c, \mathsf{aux}) \neq \varnothing,$$

rather than checking $\mathsf{DPEG.ImVer}(\mathsf{sk}, c) = \mathsf{True}$. Clearly, the two games cannot be distinguished, unless the adversary is able to create a ciphertext such that $\mathsf{DPEG.ImVer}(\mathsf{sk}, c) = \mathsf{True}$ and which is not an affine combination of the well-formed ciphertexts that it generated, in violation with the assumption that DPEG has the LOH property.

*Final reduction.* To finalize the proof, we then show that, from an adversary $\mathcal{A}$ against $G_1$, we can build an adversary $\mathcal{B}$ against the CPA security of DPEG which uses $\mathcal{A}$ as a subroutine and achieves at least $\mathcal{A}$'s advantage. For the reduction to work, we assume that $\mathcal{A}$ and $\mathcal{B}$ agree on a consistent numbering of the ciphertexts (remark then that $\mathsf{ct}^{\triangle}$ is the first generated well-formed ciphertext and is assumed to have number 0). The reduction then proceeds as follows,

- When receiving the *single* challenge request over messages $m_0 \neq m_1 \in \mathbb{Z}_N$ from $\mathcal{A}$, it transfers it as is to the CPA challenger to get ciphertext $\mathsf{ct}^* = \mathsf{DPEG.Enc}(\mathsf{pk}, m_\gamma)$ (for unknown bit $\gamma$), which it sends back to $\mathcal{A}$.
- When $\mathcal{A}$ issues a decryption request over ciphertext $\mathsf{ct} \in \mathbb{Z}^3_{N^2}$, $\mathcal{B}$ runs the LOH extractor to get,

  $$(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux}).$$

  Recall that since the only well-formed ciphertexts that $\mathcal{A}$ can generate from scratch (i.e. without using $\mathsf{ct}^{\triangle}$) are encryptions of 0, $\beta = 0$. Then, when $\pi = \varnothing$, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. When it is not the case, then $\mathcal{B}$ returns $\perp$ whenever $\pi_1 \neq 0$, or $\pi_0$ when this is not the case ($\pi_1 = 0$).

Further remark that, by definition, the LOH extractor works only on ciphertexts generated by the adversary and *not* over those generated by the challenger. Thus, following Def. 5, we have $m = 2$ and the set $(c_0, ..., c_{m-1})$ (in that definition) is reduced to $(\mathsf{ct}^{\triangle}, \mathsf{ct}^*)$ where $\mathsf{ct}^*$ is the challenge ciphertext. Hence, for a ciphertext $\mathsf{ct}$ independent of $\mathsf{ct}^*$ (i.e. such that $\pi_1 = 0$), $\mathsf{DPEG.Dec}(\mathsf{sk}, \mathsf{ct}) = \pi_0$. This follows from the perfect correctness of DPEG and the fact that, by definition of DPEG's encryption function, Lemma 7 as well as Lemma 15 ($\mathsf{PEG}_2$ satisfies property 6 in Def. 10), any well-formed ciphertext $\mathsf{ct} = (c_0, c_1, c_2)$ which encrypts a linear combination $\sum_i \alpha_i m_i$ is such that there exists $r$ such that

$$c_0 = (\mathsf{ct}^{\triangle}.c_0)^{\sum_i \alpha_i m_i} g^{2Nr},\ c_1 = (\mathsf{ct}^{\triangle}.c_1)^{\sum_i \alpha_i m_i} h^r,\ \text{and}\ c_2 = (\mathsf{ct}^{\triangle}.c_2)^{\xi \sum_i \alpha_i m_i} h'^r,$$

yielding $\pi_0 = \sum_i \alpha_i m_i$.

Lastly, since $\mathsf{ct}^*$-dependent ciphertexts are such that $\pi_1 \neq 0$, the above reduction duly replies $\perp$ for all decryption requests on ciphertexts which are byproducts of the challenge ciphertext, consistently with the vCCA game decryption oracle specification. □

### B.4 Additional technical lemmas for Sect. 5

Recall the notations in Sect. 5.

**Lemma 31.** *Let DPEG′ denote the scheme in the proof of Prop. 17. Then DPEG′ is vCCA secure under the assumption that DPEG is.*

*Proof.* Let $G_0$ denote the vCCA game against $\mathsf{DPEG}'$. We first consider game $G_1$ which is the same as $G_0$ except that we modify the challenger such that the challenge ciphertext is generated (in $G_1$) as $\mathsf{DPEG.Enc}(\mathsf{DPEG.pk}, m_\gamma)$ rather than (in $G_0$) as $\mathsf{DPEG}'.\mathsf{Enc}(\mathsf{DPEG}'.\mathsf{pk}, m_\gamma)$. From the adversary's view, the two games are computationally indistinguishable since ciphertexts under both DPEG and $\mathsf{DPEG}'$ are indistinguishable from uniform over $\mathbb{Z}^3_{N^2}$.

To complete the proof, we show that given an adversary $\mathcal{A}$ against $G_1$, we can build an adversary $\mathcal{B}$ against the vCCA security of DPEG that uses $\mathcal{A}$ as subroutine and achieves the same advantage. The reduction $\mathcal{B}$ is simple: after getting DPEG.pk from its challenger, it uses it to generate $\mathsf{DPEG}'.\mathsf{pk}$ following Eq. (11) and forwards $\mathsf{DPEG}'.\mathsf{pk}$ (but *not* DPEG.pk) to $\mathcal{A}$. Then, $\mathcal{B}$ forwards all of $\mathcal{A}$'s requests, which include one challenge request and polynomially many decryption requests, as is to its challenger. □

**Lemma 32.** *If DPEG is vCCA secure, we have that for any $m_0, ..., m_{K-1} \in \mathbb{Z}_N$ and any PPT adversary $\mathcal{A}$,*

$$P\left(\begin{array}{l|l} \mathit{DPEG.Dec}(\mathsf{sk}, c_e) \neq \perp \wedge & (\mathsf{pk}, \mathsf{sk}) := \mathit{DPEG.KeyGen}(1^\lambda) \\ \{c'_0, ..., c'_{L-1}\} \nsubseteq \{c_0, ..., c_{K-1}\} & (c_0, ..., c_{K-1}) = (\mathit{DPEG.Enc}(\mathsf{pk}, m_0), ..., \mathit{DPEG.Enc}(\mathsf{pk}, m_{K-1})) \\ & c_e := \mathcal{A}(c_0, ..., c_{K-1}, \mathit{aux}) \\ & (\rho, c'_0, ..., c'_{L-1}) := \mathit{extract}_{vCCA}(c_e, \mathit{aux}) \end{array}\right) \leq \mathit{neg}(\lambda), \tag{20}$$

*with $\rho \in \mathbb{Z}^L$.*

*Proof.* Consistently with Eq. (20), let us define $G_0$ as the game, parameterized by polynomially many messages $m_0, ..., m_{K-1} \in \mathbb{Z}_N$, in which a challenger first runs $\mathsf{DPEG.KeyGen}(1^\lambda)$ to get $(\mathsf{pk}, \mathsf{sk})$, with neither of them being communicated to a PPT adversary $\mathcal{A}$. The challenger then gives,

$$(c_0, ..., c_{K-1}) = (\mathsf{DPEG.Enc}(\mathsf{pk}, m_0), ..., \mathsf{DPEG.Enc}(\mathsf{pk}, m_{K-1})) \tag{21}$$

to $\mathcal{A}$. The adversary wins $G_0$ if it generates a ciphertext $c$ such that $\mathsf{DPEG.Dec}(\mathsf{sk}, c) \neq \perp$ which it did not obtain by linearly combining ciphertexts $c_0, ..., c_{K-1}$.

*Game $G_1$.* We now consider game $G_1$, also parameterized by polynomially many messages $m_0, ..., m_{K-1} \in \mathbb{Z}_N$, in which the challenger first runs $\mathsf{DPEG.KeyGen}(1^\lambda)$ to get $(\mathsf{pk}, \mathsf{sk})$, with neither of them being communicated to $\mathcal{A}$, but then communicates $K$ uniformly picked elements from $\mathbb{Z}^3_{N^2}$ instead of (21).

*Indistinguishability between $G_0$ and $G_1$.* We now show that, given an adversary $\mathcal{A}_0$ than can distinguish between $G_0$ and $G_1$, we can build an adversary $\mathcal{B}_0$ against the ciphertext pseudorandomness of DPEG that uses $\mathcal{A}_0$ as a subroutine and achieves the same advantage as $\mathcal{A}_0$. The challenger against the pseudorandomness of DPEG initially runs $\mathsf{DPEG.KeyGen}(1^\lambda)$ to get $(\mathsf{pk}, \mathsf{sk})$, communicates pk to $\mathcal{B}_0$ and picks its challenge bit $\gamma \xleftarrow{\$} \{0, 1\}$. The challenger also provides an encryption oracle that replies with genuine ciphertexts when $\gamma = 0$ and with uniformly picked elements in $\mathbb{Z}^3_{N^2}$, $u_0, ..., u_{L-1}$, otherwise. The reduction $\mathcal{B}_0$ then

only performs $K$ encryption requests towards its challenger and forwards the resulting challenger outputs directly to $\mathcal{A}_0$. It is clear that $\mathcal{B}_0$ exactly simulates $G_0$ when $\gamma = 0$ in its challenger, and $G_1$ otherwise.

In this last game, $\mathcal{A}$'s view is independent from $\mathsf{DPEG.pk}$. Hence, the distribution of $\mathcal{A}$'s output is independent of the structure of the well-formed ciphertext under $\mathsf{DPEG}$, i.e. $\mathcal{C}_E = \{c \in \mathbb{Z}_{N^2}^3 : c \in H \wedge \mathsf{DPEG.ImVer}(\mathsf{sk}, c) = \mathsf{True}\}$, which depends on $\mathsf{pk}$. Since $|\mathcal{C}_E| \leq 2^{-\lambda}|\mathcal{C}|$, with $\mathcal{C} = \mathbb{Z}_{N^2}^3$, even given polynomially many uniform samples from $\mathcal{C}$, a PPT adversary can output an element of $\mathcal{C}_E$ only with negligible probability[15]. Following this, and also Lemma 13 (and Prop. 16), we thus have,

$$P(\mathsf{DPEG.Dec}(\mathsf{sk}, c) \neq \perp | (\mathsf{pk}, \mathsf{sk}) := \mathsf{DPEG.KeyGen}(1^\lambda), c := \mathcal{A}(u_0, ..., u_{L-1})) \leq \mathrm{neg}(\lambda).$$

□

## C The mDPEG depth-1 multiplicative scheme

Following Sect. 3.5, we apply the Catalano-Fiore blueprint from $\mathsf{DPEG}$ as the underlying scheme, yielding scheme $\mathsf{mDPEG}$ which is defined as follows:

- $\mathsf{mDPEG.KeyGen}$: run $\mathsf{DPEG.KeyGen}(1^\lambda)$ to get $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{mDPEG.Enc}$: given $m \in \mathbb{Z}_N$ and $\mathsf{pk}$, pick uniformly $b \in \mathbb{Z}_N$ and return[16],

$$(b, \mathsf{DPEG.Enc}([m - b]_N)) = (c_0, c_1) \in \mathbb{Z}_N \times \mathbb{Z}_{N^2}^3. \tag{22}$$

- $\mathsf{mDPEG.Add}$: given two (first form) ciphertexts $(c_0, c_1) \in \mathbb{Z}_N \times \mathbb{Z}_{N^2}^3$ and $(c_0', c_1') \in \mathbb{Z}_N \times \mathbb{Z}_{N^2}^3$, return (first form) ciphertext,

$$(c_0'', c_1'') = ([c_0 + c_0']_N, \mathsf{DPEG.Add}(c_1, c_1')).$$

- $\mathsf{mDPEG.Mul}$: given two (first form) ciphertexts $(c_0, c_1) \in \mathbb{Z}_N \times \mathbb{Z}_{N^2}^3$ and $(c_0', c_1') \in \mathbb{Z}_N \times \mathbb{Z}_{N^2}^3$, return (second form) ciphertext,

$$(c_0'', c_1'', c_2'') = (\mathsf{DPEG.Add}(\alpha, \alpha', \alpha''), c_1, c_1') \in \mathbb{Z}_{N^2}^9,$$

  where $\alpha = \mathsf{DPEG.Enc}([c_0 c_0']_N)$, $\alpha' = \mathsf{DPEG.MulByConst}_{c_0'}(c_1)$ and $\alpha'' = \mathsf{DPEG.MulByConst}_{c_0}(c_1')$.
- $\mathsf{mDPEG.Dec}_1$ (first form): given $(c_0, c_1) \in \mathbb{Z}_N \times \mathbb{Z}_{N^2}^3$, if $\mathsf{DPEG.Dec}(\mathsf{sk}, c_1) = \perp$ then return $\perp$. Otherwise return $[c_0 + \mathsf{DPEG.Dec}(\mathsf{sk}, c_1)]_N$.

[15] We also highlight that the only trivially valid ciphertext (i.e. that do not decrypt to $\perp$ under a non-negligible proportion of key pairs) under $\mathsf{DPEG}$ is $(1, 1, 1)$, the neutral element of $(\mathbb{Z}_{N^2}^3, \cdot)$, that decrypts to 0 under all key pairs.

[16] Remark that, still following [21, Sect. 4.2], we slightly deviate from the original Catalano-Fiore blueprint which rather encrypts by doing $([m-b]_N, \mathsf{DPEG.Enc}(b)) = (c_0, c_1)$. These two variants are equivalent, but Eq. (22) is more convenient for handling challenge requests in reductions [21].

– $\mathsf{mDPEG.Dec}_2$ (second form): given $(c_0, c_1, c_2) \in \mathbb{Z}^9_{N^2}$, if $\exists i \in [\![0, 2]\!] : \mathsf{DPEG.Dec}(\mathsf{sk}, c_i) = \perp$ then return $\perp$. Otherwise return,

$$[\mathsf{DPEG.Dec}(\mathsf{sk}, c_0) + \mathsf{DPEG.Dec}(\mathsf{sk}, c_1)\mathsf{DPEG.Dec}(\mathsf{sk}, c_2)]_N.$$

It is then easy to check that, for all $m, m' \in \mathbb{Z}_N$,

$$\mathsf{mDPEG.Dec}_2(\mathsf{sk}, \mathsf{mDPEG.Mul}(\mathsf{mDPEG.Enc}(\mathsf{pk}, m), \mathsf{mDPEG.Enc}(\mathsf{pk}, m'))) = [mm']_N.$$

The CCA1 security of mDPEG then follows from that of DPEG from the next proposition which is a slight adaptation of [21, Prop. 4].

**Proposition 33.** *mDPEG is CCA1-secure under the DCR assumption.*

*Proof.* We proceed by showing that if there exists a CCA1 adversary $\mathcal{A}$ against mDPEG, then there exists a CCA1 adversary $\mathcal{B}$ against DPEG that uses $\mathcal{A}$ as a subroutine and achieves at least $\mathcal{A}$'s advantage. This is done via a simple reduction, $\mathcal{B}$, proceeds as follows:

– When $\mathcal{A}$ issues a challenge request with messages $m_0, m_1 \in \mathbb{Z}_N$, then $\mathcal{B}$ uniformly picks $b \in \mathbb{Z}_n$ and issue a challenge request with $[m_0 - b]_N, [m_1 - b]_N$ towards the CCA1 challenger against DPEG to get $c_1 = \mathsf{DPEG.Enc}(m_\gamma - b)$ (for unknown bit $\gamma$). It then returns $(b, c_1)$ to $\mathcal{A}$.
– When $\mathcal{A}$ issues a decryption request with (first form) ciphertext $(c_0, c_1) \in \mathbb{Z}_N \times \mathbb{Z}^3_{N^2}$, $\mathcal{B}$ first issues a decryption request over $c_1$ towards its challenger to get either $\perp$, in which case it also returns $\perp$, or $\mu \neq \perp$, in which case it returns $[c_0 + \mu]_n$ to $\mathcal{A}$.
– When $\mathcal{A}$ issues a decryption request with (second form) ciphertext $(c_0, c_1, c_2) \in \mathbb{Z}^9_{n^2}$, $\mathcal{B}$ issues decryption requests over $c_0$, $c_1$ and $c_2$ towards its challenger. If one or more $\perp$ reply is received, $\mathcal{B}$ also returns $\perp$. Otherwise, let $\mu_0$, $\mu_1$ and $\mu_2$ denote the three replies from the challenger, $\mathcal{B}$ then returns $[\mu_0 + \mu_1\mu_2]_N$ to $\mathcal{A}$.

The claim then follows from Prop. 8. □

**Proposition 34.** *mDPEG is vCCA secure under the DCR and LOH assumption.*

*Proof.* The reduction in the proof of Prop. 33 towards a vCCA challenger (rather than a CCA1 challenger) against DPEG establishes that, if there exists a vCCA adversary $\mathcal{A}$ against mDPEG, then there exists a vCCA adversary $\mathcal{B}$ against DPEG that uses $\mathcal{A}$ as a subroutine and achieves at least $\mathcal{A}$'s advantage. The claim then follows from Prop. 9. □

# D Additional preliminaries

## D.1 Computational assumptions

**Definition 35 (DDH).** *Given a cyclic group of prime order $p$, the Decision Diffie-Hellman assumption states that the distributions $\{(g, g^a, g^b, g^{ab}) : a, b \xleftarrow{\$} \mathbb{Z}_p\}$ and $\{(g, g^a, g^b, g^c) : a, b, c \xleftarrow{\$} \mathbb{Z}_p\}$ are computationnally indistinguishable.*

**Definition 36 (DCR, [27,36]).** *Let $N = pq$, the Decision Composite Residuosity assumption states that the distributions $\{[z^N]_{N^2} : z \xleftarrow{\$} \mathbb{Z}_N^*\}$ and $\{z \xleftarrow{\$} \mathbb{Z}_{N^2}^*\}$ are computationnally indistinguishable.*

Compared to the above two, the Composite Non-Invertibility (CNI) assumption is less standard although it is falsifiable and has been around for several years [26, Assumption 4.4]. The name CNI has been coined by Libert [27, Def. 1]. Informally, this assumption states that given a RSA modulus $N = pq$, an adversary cannot create an encryption of a multiple of either $p$ or $q$ under the Paillier scheme.

**Definition 37 (CNI, [27,26]).** *Let $\mathsf{Pai} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ denote the Paillier scheme (Sect. E), the Composite Non-Invertibility assumption states that for any PPT adversary $\mathcal{A}$ we have,*

$$P\left(\begin{array}{l|l} \mu = \mathsf{Pai.Dec}(c) \wedge & ((N,y),\omega) := \mathsf{Pai.KeyGen}(1^\lambda) \\ 1 < \gcd(\mu, N) < N & c := \mathcal{A}(N) \end{array}\right) \leq neg(\lambda).$$

### D.2 S. Schäge's SHPKE scheme class

We recall the definition of the class of SHPKE recently introduced by Schäge [40] and over which no black-box reduction can establish CCA1 security under a falsifiable assumption. This impossibility result has been later extended to cover the weaker notion of PCA1 security in [41].

**Definition 38 (Semi-Homomorphic public-key encryption (SHPKE) reproduced from [40,41]).** *A public-key encryption scheme $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is semi-homomorphic if the following properties are fulfilled.*

1. *For any security parameter $\lambda$ and all secret key/public key pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, it is possible to define finite, cyclic groups $(G, +)$ and $(H, \cdot)$ such that $G$ is the plaintext space, and $H = \mathsf{PKE.Enc}(\mathsf{pk}, G; \mathsf{COIN})$ that is embedded in the ciphertext space $\mathcal{C} \supseteq H$. Also, $G$ and $H$ have an efficient membership test.*
2. *The set $G$ grows exponentially in the security parameter $\lambda$.*
3. *There exists an efficient algorithm that draws integer $t$ uniformly at random from $t \leftarrow [|G|]$ with only statistically small error probability $\delta$.*
4. *Let $m, m' \in G$ and take any scalar $t \in [|G|]$. Then it holds that*

$$\begin{aligned} \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m') \cdot \prod_{i=1}^{t} \mathsf{PKE.Enc}(\mathsf{pk}, m)) &= m' + \sum_{i=1}^{t} m \\ &= m' + tm. \end{aligned}$$

5. *If $m \in G$ and $r \in D$ are taken uniformly random, then $\mathsf{PKE.Enc}(\mathsf{pk}, m; r)$ is distributed like a uniformly sampled ciphertext $c \in H$.*

It is important to point out that a key property of the SHPKE class is that $H$ is required to have an efficient public membership test or, equivalently, that valid ciphertexts under a SHPKE can be publicly efficiently recognized.

### D.3 Preliminaries on SNARGs and SNARKs

We follow the definitions in [25]. A SNARG system $\Pi$ for a boolean relation $\mathcal{R} : X \times W \longrightarrow \{\mathsf{True}, \mathsf{False}\}$ of a $\mathcal{NP}$ language,

$$L = \{x \in X : \exists w \in W, \mathcal{R}(x, w) = \mathsf{True}\},$$

is a triplet of PPT algorithms $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ defined as,

- $\mathsf{Setup}$: on input $1^\lambda$, outputs $(\mathsf{crs}, [\mathsf{sk}])$, where $\mathsf{crs}$ is a public common reference string and $\mathsf{sk}$ is an optional secret verification key.
- $\mathsf{Prove}$: given $\mathsf{crs}$, a statement $x \in X$ and a witness $w \in W$ such that $\mathcal{R}(x, w) = \mathsf{True}$, produces a proof $\pi$ for $x$.
- $\mathsf{Verif}$: given $\mathsf{crs}$ and optionally $\mathsf{sk}$ as well as a statement $x$ and a proof $\pi$, decides whether $\pi$ is a valid proof for $x$.

When $\mathsf{Verif}$ need some secret input $\mathsf{sk}$, the SNARG system is *designated verifier*. When no secret information is required to verify proofs, it is *publicly verifiable*. Then $\Pi$ is a SNARG for $L$ if the following properties holds:

1. *Completeness*: for all $(x, w) \in X \times W : \mathcal{R}(x, w) = \mathsf{True}$,

$$P\left(\Pi.\mathsf{Verif}(\mathsf{crs}, [\mathsf{sk}], x, \pi) = \mathsf{False} \;\middle|\; \begin{array}{l}(\mathsf{crs}, [\mathsf{sk}]) := \Pi.\mathsf{Setup}(1^\lambda) \\ \pi := \Pi.\mathsf{Prove}(\mathsf{crs}, x, w)\end{array}\right) \leq \mathrm{neg}(\lambda).$$

2. *Soundness*: for all PPT adversary $\mathcal{A}$,

$$P\left(\begin{array}{l}\Pi.\mathsf{Verif}(\mathsf{crs}, [\mathsf{sk}], x, \pi) = \mathsf{True} \\ \wedge x \notin L\end{array} \;\middle|\; \begin{array}{l}(\mathsf{crs}, [\mathsf{sk}]) := \Pi.\mathsf{Setup}(1^\lambda) \\ (x, \pi) := \mathcal{A}(\mathsf{crs})\end{array}\right) \leq \mathrm{neg}(\lambda).$$

3. *Succinctness*: the length of a proof is $|\pi| = \mathsf{poly}(\lambda)(|x| + |w|)^{o(1)}$.

The SNARG $\Pi$ is further a SNARK for $L$ when properties 1 and 3 hold and property 2 is replaced by the stronger property that:

- *Knowledge Soundness*: for all PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{extract}$ such that,

$$P\left(\begin{array}{l}\Pi.\mathsf{Verif}(\mathsf{crs}, [\mathsf{sk}], x, \pi) = \mathsf{True} \\ \wedge \mathcal{R}(x, w) = \mathsf{False}\end{array} \;\middle|\; \begin{array}{l}([\mathsf{crs}], \mathsf{sk}) := \Pi.\mathsf{Setup}(1^\lambda) \\ (x, \pi) := \mathcal{A}(\mathsf{crs}, \mathsf{aux}) \\ w := \mathsf{extract}(x, \pi, \mathsf{aux})\end{array}\right) \leq \mathrm{neg}(\lambda).$$

Note that this definition does not cover $zk$-SNARKs as we do not need any zero-knowledge property in this paper. Also, the above definition does not cover the notion of simulation-extractability (see e.g. [34, Sect. 5] for details) that is also not needed in this work.

## E The Paillier scheme

In this section, we remind the Paillier scheme. We focus of the folklore "new school" variant which departs slightly from the original scheme presented in [36].

- Pai.KeyGen: choose a RSA modulus $N = (2p' + 1)(2q' + 1)$, with $p'$ and $q'$ two distinct Sophie Germain primes of suitable size. Then, uniformly pick $y \in \mathbb{Z}_{N^2}$[17]. Then $\mathsf{pk} = (N, y)$ forms the public key. Further let $\omega = \mathrm{lcm}(p-1, q-1)$ which is the private decryption key, $\mathsf{sk}$.
- Pai.Enc: given $m \in \mathbb{Z}_N$ and $(N, y)$, picks $r$ uniformly in $\mathbb{Z}_N$[18] and return $[(1+N)^m y^{rN}]_{N^2}$. Alternatively, one may equivalently return $[(1+mN)y^{rN}]_{N^2}$.
- Pai.Dec: given $c \in \mathbb{Z}_{N^2}$ and $\omega$, let,

$$L(x) = \frac{x-1}{N}, \tag{23}$$

and return

$$\frac{L([c^\omega]_{N^2})}{\omega} \mod N.$$

It should be emphasized that the division in Eq. (23) is an integer division when all other divisions should be understood as multiplications by an inverse relatively to the appropriate modulus (i.e. either $N$ or $N^2$).

[17] Stricto sensu, $y$ (resp. $r$) should be picked in $\mathbb{Z}^*_{N^2}$ (resp. $\mathbb{Z}^*_N$), i.e. $\{x \in \mathbb{Z}_{N^2} : \gcd(x, N^2) = 1\}$ (resp. $\{x \in \mathbb{Z}_N : \gcd(x, N) = 1\}$). However, the probability of picking a $y$ (resp. $r$) uniformly in $\mathbb{Z}_{N^2}$ (resp. $\mathbb{Z}_N$) such that $\gcd(y, N^2) \neq 1$ (resp. $\gcd(y, N) \neq 1$) is negligible in the security parameter $\lambda$.

[18] See footnote 17.

# Table of Contents


On the CCA security properties (and more) of a new variant of Paillier-ElGamal . . . 1
*Duong Hieu Phan[1], Renaud Sirdey[2] and Jean Vacher[2]*
1 Introduction . . . 2
1.1 Our Objective . . . 3
1.2 Summary of contributions . . . 4
1.3 Paper organization . . . 6
2 Preliminaries . . . 6
2.1 Basic definitions . . . 6
2.2 Security notions . . . 7
CPA, CCA1 and CCA2 security. . . . 7
vCCA (and IV-CCA) security. . . . 8
2.3 The Linear-Only Homomorphism (LOH) assumption . . . 10
2.4 Paillier-ElGamal . . . 11
2.5 Paillier-ElGamal with plaintext zero padding . . . 12
3 A new CCA1 (and vCCA) secure variant of Paillier-ElGamal . . . 12
3.1 Paillier-ElGamal with two message slots ($\mathsf{PEG}_2$) . . . 13
3.2 Damgård-Paillier-ElGamal (DPEG) . . . 13
3.3 CCA1 security of DPEG under DCR . . . 14
3.4 vCCA security of DPEG . . . 15
3.5 A depth 1 multiplicative extension . . . 15
4 On a new class of IND-CCA1-secure schemes . . . 16
4.1 Sparsified Group-based PKE (sGPKE) . . . 16
4.2 A general result on sGPKE's CCA1 security . . . 19
4.3 DPEG as a sGPKE (proof of Prop. 8) . . . 23
4.4 Other examples of known (somewhat homomorphic) sGPKE . . . 26
Paillier-ElGamal with plaintext zero-padding ($\mathsf{PEG}_z$, Sect. 2.5). . 27
Cramer-Shoup Lite. . . . 27
Damgård's additively homomorphic ElGamal. . . . 27
5 vCCA security of DPEG under falsifiable assumptions? . . . 28
6 Conclusion and perspectives. . . . 30
A Examples of known sGPKE . . . 32
A.1 Paillier-ElGamal with plaintext zero-padding ($\mathsf{PEG}_z$, Sect. 2.5) . . 32
A.2 Cramer-Shoup Lite . . . 35
A.3 Damgård's additively homomorphic ElGamal . . . 38
B Additional proofs . . . 40
B.1 CPA security of $\mathsf{PEG}_2$ . . . 40
B.2 Alternative proof of Prop. 8 . . . 42
B.3 Proof of Prop. 9 . . . 46
B.4 Additional technical lemmas for Sect. 5 . . . 47
C The mDPEG depth-1 multiplicative scheme . . . 49

D Additional preliminaries . . . . . 50
D.1 Computational assumptions . . . . . 50
D.2 S. Schäge's SHPKE scheme class . . . . . 51
D.3 Preliminaries on SNARGs and SNARKs . . . . . 52
E The Paillier scheme . . . . . 53