

# Relations among new CCA security notions for approximate FHE<sup>\*</sup>

Sébastien Canard<sup>1</sup>, Caroline Fontaine<sup>2</sup>, Duong Hieu Phan<sup>1</sup>, David Pointcheval<sup>3</sup>, Marc Renard<sup>4</sup> and Renaud Sirdey<sup>4</sup>

<sup>1</sup> Télécom Paris, Institut Polytechnique de Paris, F-91120, Palaiseau, France,  
`sebastien.canard@telecom-paris.fr`; `hieu.phan@telecom-paris.fr`

<sup>2</sup> Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles,  
91190, Gif-sur-Yvette, France,  
`caroline.fontaine@cnrs.fr`

<sup>3</sup> DIENS, École normale supérieure, CNRS, Inria, PSL University, Paris, France,  
`david.pointcheval@ens.fr`

<sup>4</sup> Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France,  
`marc.renard@cea.fr`; `renaud.sirdey@cea.fr`

**Abstract.** In a recent Eurocrypt’24 paper, Manulis and Nguyen have proposed a new CCA security notion, vCCA, and associated construction blueprints to leverage both CPA-secure and correct FHE beyond the CCA1 security barrier. However, because their approach is only valid under the correctness assumption, it leaves a large part of the FHE spectrum uncovered as many FHE schemes used in practice turn out to be approximate and, as such, do not satisfy the correctness assumption. In this paper, we improve their work by defining and investigating a variant of their security notion which is suitable for a more general case where approximate FHE are included. As the passive security of approximate FHE schemes is more appropriately captured by CPA<sup>D</sup> rather than CPA security, we start from the former notion to define our vCCA<sup>D</sup> new security notion. Although, we show that vCCA and vCCA<sup>D</sup> are equivalent when the correctness assumption holds, we establish that vCCA<sup>D</sup> security is strictly stronger than vCCA security in the general case. In doing so, we interestingly establish several new separation results between variants of CPA<sup>D</sup> security of increasing strength. This allows us to clarify the relationship between vCCA security and CPA<sup>D</sup> security, and to reveal that the security notions landscape is much simpler for exact FHE than when approximate ones are included — in which case, for example, we establish that multiple challenges security notions are strictly stronger than single-challenge ones for both CPA<sup>D</sup> and vCCA<sup>D</sup> security. Lastly, we also give concrete construction blueprints, showing how to leverage some of the blueprints proposed by Manulis and Nguyen to achieve vCCA<sup>D</sup> security. As a result, vCCA<sup>D</sup> security is the strongest CCA security notion so far known to be achievable by both correct and approximate FHE schemes.

---

<sup>\*</sup> This work was supported by the France 2030 ANR Projects ANR-22-PECY-003 SecureCompute.

**Keywords:** FHE · CPA<sup>D</sup> · CCA security · SNARK · Verifiability.

## 1 Introduction

Since its inception more than ten years ago, Fully Homomorphic Encryption has been the subject of a lot of research toward more efficiency and better practicality. From a security perspective, however, FHE still raises a number of questions and challenges. In particular, all the FHE usable in practice, BFV [5,11], BGV [6], CKKS [9] and TFHE [10], achieve only CPA-security (and all of these schemes are trivially CCA1 insecure). Although it is well-known that malleability is contradictory with CCA2 security, building efficient FHE constructions achieving some degree of CCA security (e.g. CCA1) remains a very important open challenge.

From a theoretical perspective, a significant step has been recently achieved by Manulis and Nguyen in [19], with the introduction of the notion of vCCA security which is proven to be strictly stronger than CCA1 security while being achievable by FHE-based malleable schemes through several construction blueprints. In essence, these construction strategies consist in starting from a CPA secure and *correct* FHE and augmenting it with the machinery required for proving the well-formedness of *fresh* ciphertexts (i.e. ciphertexts which are direct outputs of the encryption function) as well as that of *evaluated* ciphertexts (i.e. ciphertexts derived from well-formed fresh ciphertexts by means of genuine homomorphic operations), with the decryption function of the augmented scheme returning  $\perp$  when the proof verification fails. The intuition behind such construction strategies is that the proof machinery downgrades attackers to CPA ones and that, as a result, some form of CCA security is achieved by the augmented scheme. Although several techniques can be used to ensure the well-formedness of fresh ciphertexts (such as signatures in the private key setting or Naor-Yung [20] in the public key setting), their approach intimately relies on Succinct Non-interactive Arguments of Knowledge (SNARKs) to enforce the well-formedness of evaluated ciphertexts. Under the assumption that the underlying SNARK is *simulation-sound extractable*, it then becomes possible to define a new (single challenge) security game along with a new security notion (vCCA), in the spirit of the CCA2 game: when the (second step) CCA2 game decryption oracle rejects the challenge ciphertext, the vCCA security game (second step) decryption oracle rejects all byproducts of the challenge ciphertext which it identifies by means of the SNARK extractor. One of their main contributions is to show, interestingly, that the resulting security notion is strictly stronger than CCA1 (no second step oracle) as well as a strict relaxation of CCA2 (because the CCA2 decryption oracle is more permissive than the vCCA decryption one). They also investigate the relationship between vCCA security and other CCA2 relaxations such as RCCA, CCA1.5, and others.

Another very important security notion for FHE is that of CPA<sup>D</sup> security [17] which formalizes the security of FHE against a slight and seemingly benign extension of CPA security where the adversary is granted access only to a

highly constrained decryption oracle which accepts only genuine ciphertexts or ciphertexts derived from genuine ciphertexts by means of genuine homomorphic operations. The initial intuition is that, by knowing the cleartext inputs of an FHE calculation, the adversary should be able to compute all the outputs of that decryption oracle by his or herself and that, as a consequence,  $\text{CPA}^D$  security is implied by or even equivalent to CPA security. However, the correctness assumption implicitly lies at the heart of this reasoning and Li and Micciancio [17] demonstrated that these intuitions are not true for approximate FHE schemes such as CKKS for which it turns out that the  $\text{CPA}^D$  decryption oracle outputs leak the LWE noises in the ciphertexts, resulting in the ability for the adversary to easily and practically recover the secret decryption key of the scheme. Although initially introduced for approximate FHE, recent works [7,8] have shown that the non-approximate FHE schemes that were previously considered immune to  $\text{CPA}^D$  attacks are, contrary to this folklore belief, all  $\text{CPA}^D$  insecure as soon as decryption errors can or can be made to occur with a non-negligible probability.

In their paper, Manulis and Nguyen [19] define and study vCCA security only under the correctness assumption and touch only very briefly on  $\text{CPA}^D$  security, essentially claiming informally that their vCCA scheme construction blueprints also apply to approximate FHE “with the caveat that approximate FHE schemes need to be  $\text{CPA}^D$ -secure”. In the present paper, we clarify the relationship between vCCA security and  $\text{CPA}^D$  security, and propose a new CCA security notion,  $\text{vCCA}^D$ , covering the spectrum of both correct *and* approximate FHE schemes. However, we show that both notions are equivalent when the correctness assumption holds, and we establish that  $\text{vCCA}^D$  security is strictly stronger than vCCA security in the general case where approximate FHE are allowed. In doing so, we interestingly establish several new separation results between variants of  $\text{CPA}^D$  security of increasing strength. This allows us to show that vCCA security does not imply  $\text{CPA}^D$  security but rather a much weaker single-challenge “CCA1 style” variant of it. We also reveal that the security notion landscape is much simpler for exact FHE than in the general case where, for example, we establish that multiple challenges security notions are strictly stronger than single-challenge ones for both  $\text{CPA}^D$  and  $\text{vCCA}^D$  security. Lastly, we also give concrete construction blueprints, showing how to leverage some of the blueprints proposed in [19] to achieve  $\text{vCCA}^D$  security.

### 1.1 Summary of security notions and contributions

In this work, we study the following (non standard) security notions:

- $\text{CPA}^D$ : the multiple challenges passive security notion introduced in [17] for approximate FHE.
- $\text{CPA}_2^D \equiv \text{CPA}_{\text{SC}}^D$ : restriction of  $\text{CPA}^D$  to the single challenge case.
- $\text{CPA}_1^D$ : restriction of  $\text{CPA}_2^D$  with the decryption oracle closing after the challenge request (similar in spirit to the CCA1/CCA2 definitions). Note that  $\text{CPA}_1^D$  is different from non-adaptive  $\text{CPA}^D$  as defined and studied in [17].

- $\text{vCCA}_{\text{SC}}$ : the single challenge CCA security notion introduced in [19] for correct FHE. Note that it is simply denoted  $\text{vCCA}$  in [19].
- $\text{vCCA}$ : the multiple challenge variant of  $\text{vCCA}$  (this variant was not considered in [19]).
- $\text{vCCA}^D$ : our main new multiple challenge CCA security notion for FHE in the general regime which includes approximate FHE.
- $\text{vCCA}_{\text{SC}}^D$ : restriction of  $\text{vCCA}^D$  to the single challenge case.

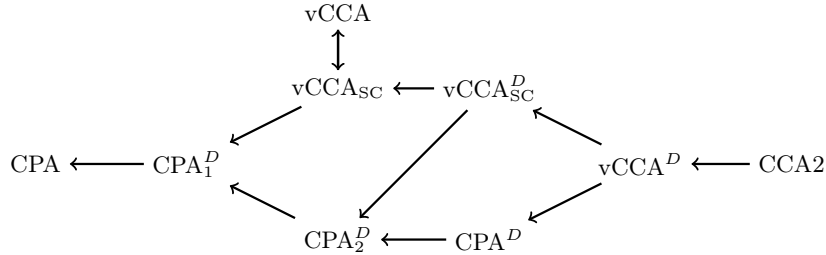
Note that, following standard conventions [3], the multiple challenge notions will sometimes be prefixed by LOR- to avoid any ambiguity with the corresponding single challenge notions (also known as FTG).

With this in mind, the contributions of this paper are as follows:

- When the correctness assumption holds for the underlying FHE, we show that:
  - $\text{CPA}_1^D, \text{CPA}_2^D \equiv \text{CPA}_{\text{SC}}^D$  and  $\text{CPA}^D$  security are (unsurprisingly) all equivalent to CPA security.
  - In that regime we also show that  $\text{vCCA}_{\text{SC}}, \text{vCCA}, \text{vCCA}_{\text{SC}}^D$  and  $\text{vCCA}^D$  security are also all equivalent.
- In the general case where the correctness assumption does not necessarily hold and approximate FHE are allowed, the picture we reveal is much more interesting:
  - For  $\text{CPA}^D$  security, we establish that  $\text{CPA}_1^D < \text{CPA}_2^D < \text{CPA}^D$  security which, as a bonus, settles the question of the relationship between single and multiple-challenge  $\text{CPA}^D$  security that was left open in [17].
  - We clarify the relationship between  $\text{vCCA}$  and  $\text{CPA}^D$  security by showing that  $\text{CPA}_1^D < \text{vCCA}_{\text{SC}}$  but that  $\text{vCCA}_{\text{SC}}$  security implies neither  $\text{CPA}_2^D$  nor  $\text{CPA}^D$  security (and vice-versa), contrary to what was informally claimed in [19].
  - For  $\text{vCCA}$  security, we further establish that  $\text{vCCA}_{\text{SC}} \equiv \text{vCCA}$  and that  $\text{vCCA}_{\text{SC}} < \text{vCCA}_{\text{SC}}^D$ , thus demonstrating that our new security notion is strictly stronger than  $\text{vCCA}$  security even in the single challenge case.
  - Lastly, for  $\text{vCCA}^D$  security, we prove that  $\text{CPA}^D < \text{vCCA}^D$  and further prove that  $\text{vCCA}_{\text{SC}}^D < \text{vCCA}^D$  which therefore implies that  $\text{vCCA}^D$  is the strongest of all these notions in the general FHE case and that it is the one that should be strived for.
- Lastly, we revisit the CPA-to- $\text{vCCA}$  FHE scheme construction blueprints proposed in [19] under the correctness assumption and turn them, when possible, into  $\text{CPA}^D$ -to- $\text{vCCA}^D$  blueprints. In particular, we are able to do so and prove the  $\text{vCCA}^D$  security of the (private key, designated verifier) Encrypt-then-MAC, (private key, public verifier) Encrypt-then-Sign, (public key, designated verifier) CCA2-Companion-Ciphertext blueprints while we also show that the (public key, public verifier) Naor-Young-based blueprint cannot be used when the correctness assumption does not hold.

Figure 1 summarizes the relationships between these notions in the general case. When proving relationships between security notions, we make a difference between the *correct case* or *correct regime*, where the FHE correctness

assumption is assumed to hold, and the *general case* or *general regime*, where approximate FHE are allowed. This terminology is used consistently in the paper.



**Fig. 1.** Summary of the security notions investigated in this paper and their relationships in the general regime where approximate FHE are allowed. Note that all single arrows are strict implications, and remind that  $CPA_2^D \equiv CPA_{sc}^D$  is the restriction of  $CPA^D$  to the single challenge case and that  $CPA_1^D$  is the restriction of  $CPA_2^D$  with the decryption oracle closing after the challenge request. Also remind that notions without a subscript in their names are multiple challenge ones and that, for consistency, we denote by  $vCCA_{sc}$  the single challenge security notion defined in [19].

### 1.2 Paper organization

This paper is organized as follows. After some preliminaries (Sect. 2), Sect. 3 introduces the definition of  $vCCA^D$  security and recalls the definitions of  $CPA^D$  and  $vCCA$  security. It also investigates the definitional connexions between these notions. Then, in Sect. 4, we investigate the relationship among the single challenge variants of these security notions in both the restricted setting where the correctness assumption holds and in the general case where approximate FHE are allowed. Then, in Sect. 5, we focus on unveiling the relationships between the single and multiple-challenge variants of these security notions as well as the relationships between the multiple-challenge variants between each other. Sect. 6 is devoted to prove the security of several generic scheme construction blueprints with respect to the stronger multiple challenge  $vCCA^D$  security notion. Sect. 7 then concludes the paper.

## 2 Preliminaries

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

- **KeyGen**: which, on input  $1^\lambda$ , outputs an encryption key  $ek$  and a decryption key  $dk$ .

- $\text{Enc}_{\text{ek}}$ : which, on inputs  $m \in \mathcal{P}$  and the encryption key  $\text{ek}$ , outputs an encryption  $c \in \mathcal{C}$  of  $m$ .
- $\text{Dec}_{\text{dk}}$ : which, on inputs  $c \in \mathcal{C}$  and the decryption key  $\text{dk}$ , outputs a decryption  $m \in \mathcal{P} \cup \{\perp\}$  of  $c$ .

Let  $\text{COIN}$  denote the randomness space of  $\mathcal{E}$ , we will sometimes externalize the randomness used in the encryption function by means of the notation  $\text{Enc}_{\text{ek}}(m; r)$ , with  $m \in \mathcal{P}$  and  $r \in \text{COIN}$  (in this case, the function  $\text{Enc}_{\text{ek}} : \mathcal{P} \times \text{COIN} \rightarrow \mathcal{C}$  is deterministic). When  $\text{ek}$  is public, we say that  $\mathcal{E}$  is a *public-key* encryption scheme. Conversely, when  $\text{ek}$  has to remain private and an adversary can create valid ciphertexts with at most  $\text{neg}(\lambda)$  probability using only public knowledge, we say that  $\mathcal{E}$  is a *private-key* encryption scheme. When for all  $(\text{ek}, \text{dk}) \in \mathcal{K}$  and all  $m \in \mathcal{P}$  we have that

$$\Pr_{r \in \text{COIN}} (\text{Dec}_{\text{dk}}(\text{Enc}_{\text{ek}}(m; r)) \neq m) \leq \text{neg}(\lambda), \quad (1)$$

we say that  $\mathcal{E}$  is *correct*. A ciphertext is *valid* if it is the output of the encryption function for some message  $m \in \mathcal{P}$ , that is, if there exists  $m \in \mathcal{P}$  and some randomness  $r \in \text{COIN}$  such that  $c = \text{Enc}_{\text{ek}}(m; r)$ . We further say that  $\mathcal{E}$  is *verifiable* if there exists a PPT algorithm  $\text{Verif}$ , taking a ciphertext as input, which tells whether or not this ciphertext is valid with a  $\text{neg}(\lambda)$  probability of error for an adversary with knowledge of only public data. For verifiable schemes, the decryption function with input ciphertext  $c$  outputs  $\perp$  when  $\text{Verif}(c) = \text{False}$ . Note that for all non-homomorphic schemes considered in this paper,  $\text{Dec}_{\text{dk}}$  will always be a *deterministic* polynomial-time algorithm. When there is no ambiguity, we omit the  $\text{ek}$  and  $\text{dk}$  subscripts to  $\text{Enc}$  and  $\text{Dec}$  to lighten the notation.

Given a function class  $\mathcal{F}_H$ , we define an homomorphic encryption (HE) scheme  $\mathcal{E}_H$  as an encryption scheme augmented by a *deterministic*<sup>1</sup> polynomial-time algorithm  $\text{Eval}$  which, on input  $f \in \mathcal{F}_H$  and  $c_1, \dots, c_K \in \mathcal{C}^K$  (where  $K$  denotes the arity of function  $f$ ), outputs a new *evaluated* ciphertext. When  $\mathcal{E}_H$  satisfies condition (1) and when  $\text{Eval}$  is such that for all  $(\text{ek}, \text{dk}) \in \mathcal{K}$ , all  $f \in \mathcal{F}_H$  and all  $m_1, \dots, m_K \in \mathcal{P}^K$

$$\Pr_{\tilde{r} \in \text{COIN}^K} (\text{Dec}(\text{Eval}(f, \text{Enc}(m_1; r_1), \dots, \text{Enc}(m_K; r_K))) \neq f(m_1, \dots, m_K)) \leq \text{neg}(\lambda),$$

we say that  $\mathcal{E}_H$  is a *correct* HE scheme. When this is not the case, we say that  $\mathcal{E}_H$  is an *approximate* HE scheme. To avoid arbitrary schemes with unreliable  $\text{Eval}$  to be marketed as approximate HE schemes, we may want to add an additional condition such that with high-enough ( $\lambda$ -independent) probability

$$|\text{Dec}(\text{Eval}(f, \text{Enc}(m_1), \dots, \text{Enc}(m_K))) - f(m_1, \dots, m_K)| \leq \varepsilon. \quad (2)$$

When  $\mathcal{E}_H$  achieves correctness only for  $\mathcal{F}_C \subset \mathcal{F}_H$  and  $\mathcal{K}(\mathcal{F}_C) \subset \mathcal{K}$ , it is said to be  $\mathcal{F}_C$ -*correct* (in the spirit of [1]).

<sup>1</sup> As is the case for the mainstream FHE schemes such as BFV, BGV, TFHE and even CKKS.

All the HE schemes we consider in this paper are public-key. Also note that for the homomorphic schemes considered in this paper,  $\text{Dec}_{\text{dk}}$  is by default a *deterministic* polynomial-time algorithm, unless explicitly stated otherwise (e.g. CKKS with noise flooding as defined in [18] has a probabilistic decryption algorithm). When  $\perp \in \mathcal{E}_H.\mathcal{P}$ , we will further always assume a *consistency* property which requires that  $\forall \vec{m}, \vec{r} \in \mathcal{P}^K \times \text{COIN}^K$ ,

$$\text{Dec}(\text{Eval}(f, \text{Enc}(m_1; r_1), \dots, \text{Enc}(m_K; r_K))) \neq \perp, \quad (3)$$

and

$$\text{Dec}(\text{Eval}(f, c_1, \dots, c_K)) = \perp, \quad (4)$$

whenever  $\exists i : \text{Dec}(c_i) = \perp$ .

As in [19], for signatures and MAC we use the standard definitions respectively  $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify})$  and  $M = (\text{KeyGen}, \text{Tag}, \text{Verify})$  and assume SUF-CMA security. In our case, EUF-CMA security will not be sufficient because we essentially use signatures as a building block for CCA2 encryption schemes in the private key setting. Lastly, we consider straightline-extractable SNARK,  $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ , over function class  $\mathcal{F}_E$  (slightly departing from [19] which required simulation-sound extractability). Indeed, in the security proofs of the constructions we study in Sect. 6, we are in the setup where only the adversary generates proofs and where our simulator only invokes the SNARK extractor when  $\Pi.\text{Verify}(\pi) = \text{True}$  on a proof  $\pi$ . Additionally, because we do not investigate circuit privacy, we do not need *zk*-SNARKs.

### 3 Defining $\text{vCCA}^D$ security

This section introduces our  $\text{vCCA}^D$  security notion which is an extension of both  $\text{vCCA}$  and  $\text{CPA}^D$  (hence the name). As they both are extensively referred to in this paper, we start by recalling the games associated with these two notions. We then define the  $\text{vCCA}^D$  game.

#### 3.1 The $\text{CPA}^D$ game

The  $\text{CPA}^D$  game has been introduced in the context of approximate FHE.  $\text{CPA}^D$  security is a slight extension of CPA security defined by the following Left-Or-Right multiple challenge security game.

Given an homomorphic encryption scheme

$$\mathcal{E}_H = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}),$$

an adversary  $\mathcal{A}$  and value  $\lambda$  for the security parameter, the game is parameterized by a bit  $b \xleftarrow{\$} \{0, 1\}$ , unknown to  $\mathcal{A}$ , and an initially empty state  $S$  of message-message-ciphertext triplets:

- Key generation: Run  $(\text{ek}, \text{dk}) \leftarrow \text{KeyGen}(1^\lambda)$ , and give  $\text{ek}$  to  $\mathcal{A}$  (when the scheme is public-key).

- Encryption request: When  $\mathcal{A}$  queries (`plaintext`,  $m$ ),  $m \in \mathcal{P}$  compute  $c = \text{Enc}(m)$ , give  $c$  to  $\mathcal{A}$  and do

$$S := [S; (m, m, c)].$$

- Challenge request: When  $\mathcal{A}$  queries (`test messages`,  $m_0, m_1$ ),  $m_0, m_1 \in \mathcal{P}^2$  ( $m_0 \neq m_1$ ) compute  $c = \text{Enc}(m_b)$ , give  $c$  to  $\mathcal{A}$  and do

$$S := [S; (m_0, m_1, c)].$$

- Evaluation request: When  $\mathcal{A}$  queries (`eval`,  $f, l_1, \dots, l_K$ ) ( $l_i \in \{1, \dots, |S|\}, \forall i$ ), compute

$$m'_0 = f(S[l_1].m_0, \dots, S[l_K].m_0),$$

and

$$m'_1 = f(S[l_1].m_1, \dots, S[l_K].m_1),$$

as well as

$$c' = \text{Eval}(f, S[l_1].c, \dots, f[l_K].c),$$

give  $c'$  to  $\mathcal{A}$  and do

$$S := [S; (m'_0, m'_1, c')].$$

- Decryption request: When  $\mathcal{A}$  queries (`ciphertext`,  $l$ ) ( $l \in \{1, \dots, |S|\}$ ) proceed as follows: if  $S[l].m_0 \neq S[l].m_1$  then return  $\perp$  to  $\mathcal{A}$ , otherwise return her  $\text{Dec}(S[l].c)$ .
- Guessing stage (after polynomially many interleaved encryption and decryption requests): When  $\mathcal{A}$  outputs (`guess`,  $b'$ ), the outcome of the game is determined as follows. If  $b' = b$  then  $\mathcal{A}$  wins the game. Otherwise,  $\mathcal{A}$  loses the game.

A number of points should be emphasized with respect to the above game. First, the decryption oracle accepts only ciphertexts from the game state which are necessarily well-formed (either produced by an encryption or challenge request, or derived by the evaluation oracle via an evaluation request i.e., derived by correctly applying homomorphic operators to well-formed ciphertexts). As such, the above game thus does not capture any CCA aspects. Second, when  $S[l].m_0 = S[l].m_1$  it is important that the decryption oracle returns  $\text{Dec}(S[l].c)$  and not  $S[l].m_0$  (or, equivalently in that case,  $S[l].m_1$ ). For exact FHE, this has no impact, as  $\text{Dec}(S[l].c) = S[l].m_0 = S[l].m_1$  in that case (and, as  $\mathcal{A}$  learns nothing it does not already know,  $\text{CPA}^D$  security coincides with CPA security for exact FHE). For approximate FHE, however, even when  $S[l].m_0 = S[l].m_1$ , we have (with overwhelming probability) that  $\text{Dec}(S[l].c) \neq S[l].m_0$  and  $\text{Dec}(S[l].c) \neq S[l].m_1$ . Thus for approximate FHEs, the decryption oracle grants  $\mathcal{A}$  access to information she cannot compute on her own, resulting or not in a guessing advantage depending on whether or not the cryptosystem at hand is  $\text{CPA}^D$  secure. Additionally, let us also emphasize that, in the above game,  $\mathcal{A}$  has control on the homomorphic calculations that are performed as  $f$  is included in the evaluation request. As a last remark, we acknowledge the fact that



explicitly adding encryption requests to the above game is redundant as these are simply challenge requests with  $m_0 = m_1$  (as was assumed in the original definition of [17]). However, since we are going back and forth between single and multiple challenges security notions in the sequel, we feel that this explicitation may avoid later confusions. Although the number of allowed challenge requests may vary from one security notion to another, the number of encryption requests an adversary can perform always remains “unlimited”.

### 3.2 The vCCA game

We now consider the vCCA game as recently introduced in [19]. Contrary to the CPA<sup>D</sup> game presented in the previous section, that game is *single challenge* meaning that the adversary performs only one request to the challenge oracle with  $m_0 \neq m_1$ . As already stated in Sect. 1.1, we will now consistently refer to the original game and security notion in [19] as vCCA<sub>SC</sub> and reserve the vCCA naming for its multiple challenge generalisation which we introduce and study in Sect. 5. Also, in the vCCA<sub>SC</sub> security game, the cryptosystem is augmented with an extractor i.e., a PPT algorithm  $\text{Extract} : \mathcal{C} \rightarrow \mathcal{F}_E \cup \{\text{id}\} \times \mathcal{C}^*$  such that:

- For any ciphertext  $c \in \mathcal{C}$  which is obtained by invoking  $\text{Eval}(f, c_1, \dots, c_K)$ ,

$$\text{Extract}(c) = (f, c_1, \dots, c_K).$$

- Otherwise,  $\text{Extract}(c) = (\text{id}, c)$ .

In particular, the above definition implies that

$$\text{Extract}(\text{Enc}(m, r)) = (\text{id}, \text{Enc}(m, r)), \tag{5}$$

and that,

$$\text{Extract}(\text{Eval}(f, \text{Enc}(m_1, r_1), \dots, \text{Enc}(m_K, r_K))) = (f, \text{Enc}(m_1, r_1), \dots, \text{Enc}(m_K, r_K)). \tag{6}$$

Being single challenge, the vCCA<sub>SC</sub> game therefore has two decryption oracles<sup>2</sup>. Before the unique challenge encryption oracle request, the first step decryption oracle is simply as follows:

- Decryption request (first step): When  $\mathcal{A}$  queries  $(\text{ciphertext}, c)$  proceed as follows: return her  $\text{Dec}(c)$ .

Then, after challenge generation,

- Decryption request (second step): When  $\mathcal{A}$  queries  $(\text{ciphertext}, c)$  proceed as follows:

---

<sup>2</sup> Of course, the vCCA<sub>SC</sub> game has no evaluation oracle as the adversary performs the homomorphic evaluations on its own in both the private and public key setting.

1. Let
 
$$(f, c_1, \dots, c_K) = \text{Extract}(c),$$
 if
 
$$c^* \in \{c_1, \dots, c_K\} \tag{7}$$
 then return  $\perp$  to  $\mathcal{A}$ .
2. Otherwise return her  $\text{Dec}(c)$ .

where  $c^*$  is the challenge ciphertext. As such, the  $\text{vCCA}_{\text{SC}}$  game is exactly the single challenge CCA2 game, with the second step decryption oracle being augmented with case 1 above (which, in essence, filters out *all* byproducts of the challenge ciphertext). Let us emphasize that  $\text{vCCA}_{\text{SC}}$  security is defined and investigated in [19] only under the (strong) assumption that  $S$  is *correct*. However, let us also emphasize that *the correctness assumption plays no role in the above definition which remains meaningful in the general regime where approximate FHE are allowed*.

As a last comment, let us emphasize that, although its definition seems intrinsically single-hop (e.g., limited to one homomorphic evaluation over fresh ciphertexts),  $\text{vCCA}_{\text{SC}}$  security can, at least in principle, be extended to the multi-hop setting by allowing recursive calls to  $\text{Extract}$  [19, Remark 2, p. 28]. Still, as emphasized in [19] and later in Sect. 6, coming up with practically credible constructions for achieving  $\text{vCCA}$  security limited to the single-hop setting is already quite challenging yet sufficient to cover a wide range of FHE use-cases.

### 3.3 $\text{vCCA}^D$ security: definitions and first properties

Contrary to the original  $\text{vCCA}$  security game introduced and studied in [19] (which, again, we refer to as  $\text{vCCA}_{\text{SC}}$  in this paper), the  $\text{vCCA}^D$  game is a multiple challenge one. Due to subtleties that will soon be clear, *we first define it in the private key setting*, starting from the  $\text{CPA}^D$  game in Sect. 3.1 *without the evaluation oracle* and assuming, as in  $\text{vCCA}_{\text{SC}}$ , the existence of the same extractor.

*In the private key setting*, the  $\text{vCCA}^D$  game decryption oracle is defined as:

- Decryption request: When  $\mathcal{A}$  queries  $(\text{ciphertext}, c)$  proceed as follows:
  1. Let
 
$$(f, c_1, \dots, c_K) = \text{Extract}(c),$$
 if
 
$$f(\text{left}(c_1), \dots, \text{left}(c_K)) \neq f(\text{right}(c_1), \dots, \text{right}(c_K)) \tag{8}$$
 then return  $\perp$  to  $\mathcal{A}$ .
  2. Otherwise return her  $\text{Dec}(c)$ .

Where for any ciphertext  $c \in \mathcal{C}$  we define

$$\text{left}(c) = \begin{cases} S[i].m_0 & \text{if } \exists i : S[i].c = c, \\ \perp & \text{otherwise,} \end{cases} \tag{9}$$

as well as,

$$\text{right}(c) = \begin{cases} S[i].m_1 & \text{if } \exists i : S[i].c = c, \\ \perp & \text{otherwise,} \end{cases} \quad (10)$$

and with the convention that  $f(m_1, \dots, m_K) = \perp$  when  $\exists i : m_i = \perp$ . Note that if the left and right evaluations both give  $\perp$ , condition (8) is not satisfied and  $\text{Dec}(c)$  is returned to  $\mathcal{A}$ .

*Remark 1.* It is clear that any ciphertext accepted by the  $\text{vCCA}^D$  game decryption oracle is also accepted by the LOR-CCA2 game one but not vice-versa. For example, ciphertext  $\text{Eval}(\text{sum}, c^*, \text{Enc}(1))$  is accepted by the LOR-CCA2 decryption oracle (but rejected by the  $\text{vCCA}^D$  one) and trivially allows an adversary to win the CCA2 game. It follows that  $\text{vCCA}^D$  security is a strict relaxation of CCA2 security (which is well-known to exclude malleability<sup>3</sup>). †

*Remark 2.* In the private key setting, well-formed fresh ciphertexts (including challenge ones) can only be obtained by means of (encryption or challenge) oracle requests. Therefore, all such ciphertexts are registered in the game state  $S$ . It then follows that for any ciphertext of the form

$$c = \text{Eval}(f, \text{Enc}(m_1), \dots, \text{Enc}(m_K))$$

we have that  $\text{left}(c) \neq \perp$  and  $\text{right}(c) \neq \perp$ . However, when the correctness assumption does not hold, we may have that  $\text{Dec}(c) \neq f(\text{left}(c_1), \dots, \text{left}(c_K))$  as well as  $\text{Dec}(c) \neq f(\text{right}(c_1), \dots, \text{right}(c_K))$ . †

If we compare the  $\text{vCCA}_{\text{SC}}$  game in previous Sect. 3.2 and the single challenge variant,  $\text{vCCA}_{\text{SC}}^D$ , of the above game,  $\text{vCCA}_{\text{SC}}$ 's second step oracle filters out *all* byproducts of the challenge ciphertext whereas (single challenge)  $\text{vCCA}_{\text{SC}}^D$  filters out only those byproducts which allow to discriminate which of the two challenge plaintexts was encrypted.

*Remark 3.* From the definition of the two games, it is clear that all the ciphertexts accepted by the  $\text{vCCA}_{\text{SC}}$  decryption oracle are also accepted by the  $\text{vCCA}_{\text{SC}}^D$  one (but not vice-versa). There are indeed two types of ciphertexts which are rejected by the  $\text{vCCA}_{\text{SC}}$  decryption oracle but accepted by the  $\text{vCCA}_{\text{SC}}^D$  one:

1. Ciphertexts obtained through a legit call to  $\text{Eval}$  over well-formed fresh ciphertexts (with one of them being the challenge ciphertext) for which condition (8) does not hold, e.g.  $\text{Eval}(f, c^*, \text{Enc}(m_2), \dots, \text{Enc}(m_K))$  with

$$f(m_0^*, m_2, \dots, m_k) = f(m_1^*, m_2, \dots, m_k),$$

where  $m_0^*$  and  $m_1^*$  denote the two challenge plaintexts. For example, ciphertext  $\text{Eval}(\text{mul}, c^*, \text{Enc}(0))$  fall into this category.

<sup>3</sup> Additionally, it is well known that LOR-CCA2 security is equivalent to (single challenge) CCA2 (a.k.a., FTG-CCA2) in both the public key [4] and private key [3] settings. Also, note that CCA1 obviously does not really make sense in the multiple challenge setting.

2. Ciphertexts obtained through a legit call to `Eval` over arbitrary ciphertexts, with one of them being the challenge ciphertext and at least one of the others being ill-formed.

†

Throughout this paper and in particular in the separation results we establish, we (almost) always exploit the first of the above two gaps. As will be seen in Sect. 6, the second gap will be closed in the construction themselves by including the machinery necessary for the decryption *function* of the proposed schemes to reply  $\perp$  when given either an ill-formed ciphertext or an evaluated ciphertext over non well-formed ones.

**Defining vCCA<sup>D</sup> security in the public key case.** In the public key setting, the adversary can generate well-formed fresh ciphertexts independent of the challenge on his or her own. So only challenge dependent fresh ciphertexts are guaranteed to be registered in the game state. In order to perform the left and right cleartext evaluations we therefore need a mean to access the messages that were given as inputs to the encryption function for well-formed ciphertexts that the adversary generated by his or herself. Note that, when the correctness assumption does not hold (which is also the regime under which we are willing to operate in this paper), these inputs cannot be recovered by merely decrypting those ciphertexts within the vCCA<sup>D</sup> game decryption oracle.

Therefore, to define the vCCA<sup>D</sup> game in the public key setting we need an additional extractor, denoted `Extract'`, for recovering the encryption function inputs for fresh well-formed ciphertexts i.e.,

$$\text{Extract}'(c) = \begin{cases} (m; r) & \text{when } c := \text{Enc}(m, r), \\ \perp & \text{otherwise.} \end{cases}$$

Following this, *in the public key setting*, the vCCA<sup>D</sup> game decryption oracle is then defined similarly to the private key case but with the notable difference that the left and right functions are replaced by the left' and right' functions defined as

$$\text{left}'(c) = \begin{cases} S[i].m_0 & \text{if } \exists i : S[i].c = c, \\ \text{Extract}'(c).m & \text{otherwise,} \end{cases}$$

as well as,

$$\text{right}'(c) = \begin{cases} S[i].m_1 & \text{if } \exists i : S[i].c = c, \\ \text{Extract}'(c).m & \text{otherwise.} \end{cases}$$

Following this, we however emphasize that, as we shall later see in Sect. 6, among the two public key vCCA<sub>SC</sub> scheme construction blueprints considered in [19], only the one in which fresh ciphertexts are defined as the association of a FHE ciphertext encrypting  $m$  by means of randomness  $r$  and another ciphertext encrypting the concatenation of  $m$  and  $r$  under a CCA2-secure encryption scheme is applicable in the general case where the correctness assumption may

not hold. In this approach the well-formedness of fresh ciphertext is verified by first decrypting the companion CCA2 ciphertext to recover  $m$  and  $r$  and then checking that the associated FHE ciphertext is indeed equal to  $\mathcal{E}_H.\text{Enc}(m, r)$ . This, in essence, provides the additional extractor,  $\text{Extract}'$ , expected in the above definition.

**vCCA<sup>D</sup> security vs CPA<sup>D</sup> security.** As a warm-up, let us now prove a first separation result between CPA<sup>D</sup> and vCCA<sup>D</sup> security.

**Lemma 1.** *vCCA<sup>D</sup> security implies CPA<sup>D</sup> security.*

*Proof.* By definition of the two games, the result of any encryption or decryption request performed by a CPA<sup>D</sup> adversary is also accessible to a vCCA<sup>D</sup> adversary.  $\square$

**Proposition 1 (CPA<sup>D</sup>  $\not\Rightarrow$  vCCA<sup>D</sup>).** *If there exists a vCCA<sup>D</sup>-secure scheme  $S$ , then there exists a scheme  $S'$  which is CPA<sup>D</sup>-secure but not vCCA<sup>D</sup>-secure.*

*Proof.* Let us start from a vCCA<sup>D</sup>-secure scheme  $S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$ . We now consider the scheme  $S'$  which is exactly  $S$  except that the public material of  $S'$  further includes a special ciphertext  $c^\Delta = \text{Enc}(\text{sk})$ , where  $\text{sk}$  is the secret key of  $S$ . The CPA<sup>D</sup> security of  $S'$  then follows from the CPA<sup>D</sup> security of  $S$  (by Lemma 1) as well as the fact that the CPA<sup>D</sup> game decryption oracle takes state indices rather than ciphertexts as argument. As a consequence, a CPA<sup>D</sup> adversary against  $S'$  cannot add  $c^\Delta$  to the game state (with non-negligible probability) without already knowing its associated (non public) plaintext, as well as the randomness used for generating  $c^\Delta$ . Yet,  $S'$  is trivially vCCA<sup>D</sup>-insecure, as a vCCA<sup>D</sup> adversary can submit  $c^\Delta$  to the vCCA<sup>D</sup> game decryption oracle and get  $\text{sk}$  in return.  $\square$

This simple proof pattern will occur several times in this paper.

## 4 Relations among the single challenge notions

We consider in this section the single challenge variants of vCCA<sup>D</sup> and vCCA which we respectively denote vCCA<sup>D</sup><sub>SC</sub> and vCCA<sub>SC</sub>. Following Sect. 3.2, in these single challenge variants, we allow only one challenge request with  $m_0 \neq m_1$ . The challenge ciphertext and the associated messages are respectively denoted  $c^*$ ,  $m_0^*$  and  $m_1^*$ . We further consider in this section the single challenge variant of CPA<sup>D</sup> which we denote CPA<sup>D</sup><sub>SC</sub>. In the single challenge case, we can further meaningfully split CPA<sup>D</sup><sub>SC</sub> in CPA<sup>D</sup><sub>1</sub> and CPA<sup>D</sup><sub>2</sub>  $\equiv$  CPA<sup>D</sup><sub>SC</sub>. Analogously to the distinction between CCA1 and CCA2, in CPA<sup>D</sup><sub>1</sub>, the CPA<sup>D</sup><sub>SC</sub> decryption oracle closes after the challenge request is performed.

#### 4.1 Relations between single-challenge variants of CPA<sup>D</sup>

Because CPA<sup>D</sup> security collapses onto CPA security in the correct regime, then CPA<sub>1</sub><sup>D</sup> is equivalent to CPA<sub>2</sub><sup>D</sup> in that regime. In the general regime, however, we have the following separation result (considering that CPA<sub>2</sub><sup>D</sup> trivially implies CPA<sub>1</sub><sup>D</sup>).

**Proposition 2 (CPA<sub>1</sub><sup>D</sup>  $\not\Rightarrow$  CPA<sub>2</sub><sup>D</sup>).** *If there exists an FHE scheme  $S$  which is CPA<sub>2</sub><sup>D</sup>-secure, then there exists an FHE encryption scheme  $S'$  which is CPA<sub>1</sub><sup>D</sup>-secure but CPA<sub>2</sub><sup>D</sup>-insecure.*

*Proof.* Let us consider a CPA<sub>2</sub><sup>D</sup>-secure FHE scheme  $S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$ . Consider the approximate scheme  $S' = (\text{KeyGen}', \text{Enc}', \text{Dec}', \text{Eval}')$  built from  $S$  such that

$$\text{Enc}'(m) = \text{Enc}(m + g(m)),$$

where  $g$  is some function such that  $g(0) \neq 0$ , and  $\text{KeyGen}'$ ,  $\text{Dec}'$  and  $\text{Eval}'$  are similar to those of  $S$ . For example, we can choose  $g(m) = \max(1, \lceil m/B \rceil)$  for some constant  $B$ .

$S'$  is CPA<sub>1</sub><sup>D</sup>-secure. This follows from the CPA<sub>2</sub><sup>D</sup> security of  $S$  and the fact that the approximation noise  $g(m)$  is independent of  $S$  secret key material (which is the only information not available to a CPA<sub>1</sub><sup>D</sup> adversary).

$S'$  is not CPA<sub>2</sub><sup>D</sup>-secure. The adversary issues the *unique* encryption request with  $m_0 \neq m_1$  to get

$$m_0^*, m_1^*, \text{Enc}'(m_b^*),$$

say with  $m_0^* = 0$  and  $m_1^* = 1$ . He or she subsequently asks for an encryption of 0,  $c_0$ , and then asks for the computation of

$$c_{\text{mul}} = \text{Eval}'(\text{mul}, c_0, c^*)$$

which the CPA<sub>2</sub><sup>D</sup> decryption oracle accepts as it is associated to the triplet  $(0, 0, c_{\text{mul}})$  in the game state ( $c_{\text{mul}}$  is an encryption of 0 with respect to  $S'$ ). The adversary therefore gets  $(b + g(b))g(0)$  i.e.  $g(0)^2$  when  $b = 0$  or  $(1 + g(1))g(0)$  when  $b = 1$ . Thus, assuming  $g(0)^2 \neq (1 + g(1))g(0)$  (which is the case with the above example), the adversary can decide that  $b = 0$  when

$$\text{Dec}'(c_0)^2 = \text{Dec}'(c_{\text{mul}})$$

and  $b = 1$  otherwise, and win the CPA<sub>2</sub><sup>D</sup> game with certainty.  $\square$

Note that this result is different from Proposition 2 in [17] which establishes that there exists (approximate) FHE schemes which are non-adaptive CPA<sup>D</sup> secure while being adaptive CPA<sup>D</sup> insecure. Indeed, there is a slight difference between the notion of adaptability as understood in the multiple-challenge context of [17] (the adversary performs all its requests at once) and that which is usually assumed between single-challenge CCA1 and CCA2 (the adversary performs all its requests before the challenge is published).

## 4.2 Relations between $vCCA_{SC}$ and single-challenge variants of $CPA^D$

Let us emphasize that the issue of approximate schemes is only succinctly and informally discussed in [19]. Indeed, from a construction point of view, that paper claims to define blueprints for constructing  $vCCA$ -secure schemes from “state-of-the-art FHE such as TFHE or CKKS with the caveat that approximate FHE schemes need to be  $CPA^D$ -secure”<sup>4</sup>. It turns out that the results in this section clarify the relationship between  $vCCA_{SC}$  security and  $CPA^D$  security in the general regime where approximate FHE are allowed:  $vCCA_{SC}$  in fact requires much less than full-blown  $CPA^D$  security but rather its weaker “CCA1 style” variant,  $CPA_1^D$ . Implicitly, we consider here a generalization of  $vCCA_{SC}$  security beyond the correctness assumption. However, as argued in Sect. 3.2, in terms of security game definition, there is no dependency on the correctness assumption. Indeed, in [19], that assumption only steps in for proving the  $vCCA_{SC}$  security of the proposed constructions.

**Proposition 3.**  *$vCCA_{SC}$  security implies  $CPA_1^D$  security.*

*Proof.* By definition, a  $CPA_1^D$  adversary can only perform decryption requests which are independent of the challenge ciphertext. It thus follows that any request performed by a  $CPA_1^D$  adversary can also be performed by a  $vCCA_{SC}$  one.  $\square$

**Proposition 4** ( $vCCA_{SC} \not\Rightarrow CPA_2^D$ ). *If there exists a  $vCCA_{SC}$ -secure scheme  $S$ , then there exists a scheme  $S'$  which is  $vCCA_{SC}$ -secure and  $CPA_2^D$ -insecure.*

*Proof.* We proceed similarly to the proof of Proposition 2. Let us start from a  $vCCA_{SC}$ -secure scheme  $S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$  from which we build the scheme  $S'$  with the only modification that

$$\text{Enc}'(m) = \text{Enc}(m + g(m)).$$

with  $g$  as in the proof of Proposition 2.

*$S'$  is  $vCCA_{SC}$ -secure.* Let  $\mathcal{A}$  be a successful adversary against the  $vCCA_{SC}$  security of  $S'$ . It is then easy to build an adversary  $\mathcal{B}$  against the  $vCCA_{SC}$  security of  $S$ . Indeed,  $\mathcal{B}$  simulates  $\mathcal{A}$  encryption and challenge requests simply by adding  $g(m)$  to  $m$ . All other requests are transferred “as is” by  $\mathcal{B}$  to the  $vCCA_{SC}$  game against  $S$ .

*$S'$  is  $CPA_2^D$ -insecure.* Identical to proof of Proposition 2: the  $CPA_2^D$  decryption oracle accepts the  $c_{\text{mul}}$  ciphertext since it is duly registered in the game state within the triplet  $(0, 0, c_{\text{mul}})$  as an encryption of 0 with respect to  $S'$ .  $\square$

**Proposition 5** ( $CPA_2^D \not\Rightarrow vCCA_{SC}$ ). *If there exists a  $CPA_2^D$ -secure scheme  $S$ , then there exists a scheme  $S'$  which is  $CPA_2^D$ -secure and  $vCCA_{SC}$ -insecure.*

<sup>4</sup> Following the recent attacks in [7,8], the authors of [19] further updated their ePrint version to put additional emphasis on the FHE correctness assumption (see Remark 1 on p. 5 and Sect. 5.4).

*Proof.* The proof is essentially identical to that of Proposition 1, but starting from a  $\text{CPA}_2^D$ -secure scheme  $S$  from which we create a scheme  $S'$  in a similar way. On one hand, the  $\text{CPA}_2^D$ -security of  $S'$  follows from that of  $S$  and the fact that a  $\text{CPA}_2^D$  adversary against  $S'$  cannot add  $c^\Delta$  to the game state. On the other hand, the  $\text{vCCA}_{\text{SC}}$ -insecurity of  $S'$  follows from the fact that the  $\text{vCCA}_{\text{SC}}$  game decryption oracle accepts  $c^\Delta$  as it bears no relationship with the challenge ciphertext.  $\square$

### 4.3 Relations between $\text{vCCA}_{\text{SC}}$ and $\text{vCCA}_{\text{SC}}^D$ security

In this section, we establish the relationships between  $\text{vCCA}_{\text{SC}}$  security (recall that only  $\text{vCCA}_{\text{SC}}$  is studied in [19] and, as such, only denoted  $\text{vCCA}$ ) and  $\text{vCCA}_{\text{SC}}^D$  in both the correct regime and the general regime where approximate FHE are allowed. In a nutshell, we establish that, although the two notions are equivalent in the correct regime,  $\text{vCCA}_{\text{SC}}^D$  security is strictly stronger than  $\text{vCCA}_{\text{SC}}$  in the general case.

**Lemma 2.**  *$\text{vCCA}_{\text{SC}}^D$  security implies  $\text{vCCA}_{\text{SC}}$  security.*

*Proof.* By definition of the two games, all decryption requests accepted by the  $\text{vCCA}_{\text{SC}}$  decryption oracle are also accepted by the  $\text{vCCA}_{\text{SC}}^D$  one (recall also Remark 3 on p. 11). It thus follows that any request performed by a  $\text{vCCA}_{\text{SC}}$  adversary can also be performed by a  $\text{vCCA}_{\text{SC}}^D$  one.  $\square$

Note that the above implication holds in the general regime i.e., independently of the correctness assumption.

We then prove a first result showing that  $\text{vCCA}_{\text{SC}} \not\Rightarrow \text{vCCA}_{\text{SC}}^D$  in the correct regime, under some condition on the probability that an adversary may bypass plaintext awareness.

**Proposition 6.** *Let  $S$  be a  $\text{vCCA}_{\text{SC}}$ -secure scheme and let  $\mu^\perp$  denotes the probability that  $\text{Dec}(u) = \perp$  for  $u \xleftarrow{\$} \mathcal{C}$ , then, under the correctness assumption, there exists an (active) adversary against the  $\text{vCCA}_{\text{SC}}^D$ -security of  $S$  which achieves advantage  $(1 - \mu^\perp)(1 - 1/|\mathcal{P}|)$ .*

*Proof.* Consider a  $\text{vCCA}_{\text{SC}}$ -secure scheme  $S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$  and assume  $m_0^* = 0$ ,  $m_1^* = 1$  and  $c^* = \text{Enc}(m_b^*)$ . We now consider the following active adversary  $\text{vCCA}_{\text{SC}}^D$  attack against  $S$ . First,  $\mathcal{A}$  picks a ciphertext  $c_{\text{rnd}}$  uniformly at random in  $\mathcal{C}$ . He or she then performs,

$$c_{\text{mul}} = \text{Eval}(\text{mul}, c^*, c_{\text{rnd}}).$$

where  $c_{\text{mul}}$  (as well as  $c_{\text{rnd}}$ ) decrypts to  $\perp$  with probability  $\mu^\perp$ <sup>5</sup>. Now, since it is a byproduct of the challenge ciphertext via an invocation of  $\text{Eval}$ , ciphertext  $c_{\text{mul}}$  is rejected by the  $\text{vCCA}_{\text{SC}}$  game decryption oracle. However,  $c_{\text{mul}}$  is accepted by the  $\text{vCCA}_{\text{SC}}^D$  game decryption oracle: since  $c_{\text{rnd}}$  cannot be part of the  $\text{vCCA}_{\text{SC}}^D$

<sup>5</sup> Following the consistency assumption (4) (Sect. 2).



game state (as  $\mathcal{A}$  does not know the associated plaintext), then neither can be  $c_{\text{mul}}$ . When  $\text{Dec}(c_{\text{mul}}) = 0$ , then  $\mathcal{A}$  decides that  $b = 0$  and, under the correctness assumption, wins the  $\text{vCCA}_{\text{SC}}^D$  game with probability  $1 - \frac{1}{|\mathcal{P}|}$ . Hence the claim.  $\square$

Note that thanks to condition (2), this proposition can also easily be generalized to the general regime which also include approximate FHE schemes. However this result is of limited interest as the above attack leads to a non-negligible advantage only if  $S$  is such that  $1 - \mu^\perp > \text{neg}(\lambda)$ . However, as already discussed in Sect. 3.2 (recall Remark 3, page 11 and its follow up discussion) all the  $\text{vCCA}_{\text{SC}}$ -secure constructions proposed in [19] and revisited in Sect. 6 include the machinery necessary for their decryption *function* to return  $\perp$  when given either an ill-formed ciphertext (which will then be the case of  $c_{\text{rnd}}$  with at least  $1 - \text{neg}(\lambda)$  probability) or an evaluated ciphertext over non well-formed ones (which is then the case of  $c_{\text{mul}}$ ). As a consequence, it is interesting to study the relationship between  $\text{vCCA}_{\text{SC}}$ -security and  $\text{vCCA}_{\text{SC}}^D$ -security when the adversary is limited to exploit only legit ciphertexts (well-formed fresh ciphertexts or ciphertexts derived from fresh well-formed ciphertexts via legit homomorphic evaluations), i.e. when the adversary remains *passive*.

**Proposition 7.** *Under both the FHE correctness and passive adversary assumptions,  $\text{vCCA}_{\text{SC}}$  security is equivalent to  $\text{vCCA}_{\text{SC}}^D$  security.*

*Proof.* Following Lemma 2 we have that  $\text{vCCA}_{\text{SC}}^D$  security implies  $\text{vCCA}_{\text{SC}}$  security. For the other direction, let us consider a scheme  $S$  which is  $\text{vCCA}_{\text{SC}}$  secure and  $\text{vCCA}_{\text{SC}}^D$  insecure. So let us consider a successful (*passive*)  $\text{vCCA}_{\text{SC}}^D$  attack against  $S$ . This means that there exist a set of ciphertexts of the form (assuming wlog that the challenge ciphertext appears only once as the first position argument)

$$c_i = \text{Eval}(g_i, c^*, c_1^{(i)}, \dots, c_{K-1}^{(i)})$$

where  $c_j^{(i)} = \text{Enc}(m_j^{(i)})$ ,  $j \in \{1, \dots, K-1\}$ , such that

$$g_i(m_0^*, m_1^{(i)}, \dots, m_{K-1}^{(i)}) = g_i(m_1^*, m_1^{(i)}, \dots, m_{K-1}^{(i)}) = r_i \quad (11)$$

and such that the  $\text{vCCA}_{\text{SC}}^D$  adversary correctly guesses the challenge bit with a non negligible advantage from the knowledge of the  $\text{Dec}(c_i)$ 's. It is easy to see that the  $c_i$ 's are filtered out by the  $\text{vCCA}_{\text{SC}}$  decryption oracle whereas the  $\text{vCCA}_{\text{SC}}^D$  one let their decryption pass through to the ( $\text{vCCA}_{\text{SC}}^D$ ) adversary. However, because the FHE scheme is correct, we have

$$\text{Dec}(c_i) = r_i$$

with probability at least  $1 - \text{neg}(\lambda)$ . Since, the ( $\text{vCCA}^D$ ) adversary can compute the  $r_i$ 's on his or her own by means of (11), he or she can extract new knowledge from the  $\text{Dec}(c_i)$ 's only with a negligible advantage. This contradicts the successful attack assumption and the claim follows.  $\square$

In the general case, however, it turns out that the two notions can be separated as established by the following proposition.

**Proposition 8** ( $\text{vCCA}_{\text{SC}} \not\Rightarrow \text{vCCA}_{\text{SC}}^D$ ). *In the general regime, if there exists a  $\text{vCCA}_{\text{SC}}^D$ -secure scheme  $S$ , then there exists a scheme  $S'$  which is  $\text{vCCA}_{\text{SC}}$ -secure but  $\text{vCCA}_{\text{SC}}^D$ -insecure, even against a passive adversary.*

*Proof.* We proceed similarly to the proof of Proposition 2 (and Proposition 4). Let us start from a  $\text{vCCA}_{\text{SC}}^D$ -secure scheme  $S = (\text{KeyGen}, \text{EncDec}, \text{Eval})$  from which we build the scheme  $S'$  with the only modification that

$$\text{Enc}'(m) = \text{Enc}(m + g(m)).$$

with function  $g$  as in the proof of Proposition 2.

$S'$  is  $\text{vCCA}_{\text{SC}}$ -secure. Since  $S$  is  $\text{vCCA}_{\text{SC}}^D$ -secure, it is also  $\text{vCCA}_{\text{SC}}$ -secure (from Lemma 2). Now, let  $\mathcal{A}$  be a successful adversary against the  $\text{vCCA}_{\text{SC}}$  security of  $S'$ . It is then easy to build an adversary  $\mathcal{B}$  against the  $\text{vCCA}_{\text{SC}}$  security of  $S$ . Indeed,  $\mathcal{B}$  simulates  $\mathcal{A}$  encryption and challenge requests simply by adding  $g(m)$  to  $m$ . All other request are transferred “as is” by  $\mathcal{B}$  to the  $\text{vCCA}_{\text{SC}}$  game against  $S$ .

$S'$  is  $\text{vCCA}_{\text{SC}}^D$ -insecure. Identical to proof of Proposition 2: the  $\text{vCCA}_{\text{SC}}^D$  decryption oracle accepts the  $c_{\text{mul}}$  ciphertext as, recall (9) and (10),  $\text{left}(c_{\text{mul}}) = \text{right}(c_{\text{mul}}) = 0$  (as  $c_{\text{mul}}$  is an encryption of 0 with respect to  $S'$ ).

Since this latter attack involves only legit ciphertexts, it can be performed by a passive adversary.  $\square$

Following Lemma 2 and Proposition 8 we can conclude that  $\text{vCCA}_{\text{SC}}^D$  security is strictly stronger than  $\text{vCCA}_{\text{SC}}$  security in the general regime.

## 5 Relations among the multiple challenge notions

### 5.1 Relations between $\text{CPA}_{\text{SC}}^D$ and $\text{CPA}^D$ security

In this section, we first focus on  $\text{CPA}^D$  and study the relationship between the single and multiple-challenge variants of this notion. This is an interesting question as, unless the FHE scheme is restricted to the evaluation of univariate functions, the usual hybrid argument e.g. in [3] (theorem 4) for showing the equivalence (up to an increase in advantage linear in the number of challenge ciphertexts) between FTG-CPA<sup>6</sup> (resp. FTG-CCA) and LOR-CPA (resp. LOR-CCA) does not work directly. This is so because an adversary confronted to a hybrid game (where the encryption oracle replies according to  $b = 0$  up to a random point after which it replies according to  $b = 1$ ) can detect the transition between the first and second phase since ciphertexts from the two phases may interact via evaluation requests. The relationship between single and multiple-challenge variants of  $\text{CPA}^D$  was also explicitly left as an open question in [17].

<sup>6</sup> Recall that Find-Then-Guess (FTG) is the terminology for single challenge security notions in the foundational papers [3,4].

Then, we also establish that, for  $\text{CPA}^D$ , the single and multiple challenge variants are (without surprise) equivalent in the correct regime and, more interestingly, that the two notions can be separated in the general regime (which is the non-trivial  $\text{CPA}^D$  setting).

We first recall the following well-known theorem from [3].

**Theorem 1.** *For any encryption scheme  $S = (\text{KeyGen}, \text{Enc}, \text{Dec})$ , LOR-CPA is equivalent to FTG-CPA<sup>7</sup>.*

For  $\text{CPA}^D$  security, in the correct FHE regime, we have the following equivalence. This equivalence is not surprising since  $\text{CPA}^D$  security collapses onto CPA security for correct FHE.

**Proposition 9.** *For any correct FHE scheme,  $S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$ ,  $\text{CPA}^D$  security is equivalent to  $\text{CPA}_{SC}^D$  security.*

*Proof.* For a correct FHE scheme, it is well-known that LOR- $\text{CPA}^D$  is equivalent to LOR-CPA [17]. This means that an adversary to the LOR- $\text{CPA}^D$  game has exactly the same advantage as an adversary to *the LOR-CPA<sup>D</sup> game without the decryption oracle*, which is the same as the LOR-CPA game plus the evaluation oracle (recall that the LOR-CPA<sup>D</sup> game decryption oracle, Sect. 3.1, takes indices from the game state rather than ciphertexts as input). Let’s call this the LOR-CPA<sup>E</sup> game (“CPA with an evaluation oracle”). It is easy to see that LOR-CPA<sup>E</sup> is exactly LOR-CPA. For the same reasons,  $\text{CPA}_{SC}^D$  is equivalent to FTG-CPA. The claim then follows from theorem 1 above.  $\square$

In the general regime where approximate FHE are allowed, things are more interesting as we can actually separate the two notions. We start by proving the separation in the special case of additive FHE scheme.

**Proposition 10.** *In the general regime, if there exists an additive HE scheme  $S$  which is  $\text{CPA}^D$ -secure, then there exists an additive HE scheme  $S'$  which is  $\text{CPA}_{SC}^D$ -secure and  $\text{CPA}^D$ -insecure.*

*Proof.* So let us start with a  $\text{CPA}^D$ -secure additive HE scheme

$$S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}).$$

Then, consider the scheme  $S' = (\text{KeyGen}', \text{Enc}', \text{Dec}', \text{Eval}')$  such that

$$\text{Enc}'(m) = \text{Enc}(m + g(m)),$$

where  $g$  is some non linear function such that  $g(a+b) \neq g(a) + g(b)$ <sup>8</sup>. Additionally  $\text{KeyGen}'$ ,  $\text{Dec}'$  and  $\text{Eval}'$  are the same as those of  $S$ .

<sup>7</sup> More precisely [3] established that the advantage of a LOR-CPA (resp. LOR-CCA) adversary is bounded by  $q_e \alpha_{sc}$ , where  $q_e$  is an upper bound the number of encryption queries with  $m_0 \neq m_1$  and  $\alpha_{sc}$  is the advantage of an FTG-CPA (resp. FTG-CCA) adversary.

<sup>8</sup> More precisely, we need  $g$  such that there exists  $a, b, a'$  and  $b'$  such that  $a + b = a' + b'$  while  $g(a) + g(b) \neq g(a') + g(b')$ .

$S'$  is  $CPA_{SC}^D$ -secure. First of all, let us remark that the  $CPA_1^D$  security of  $S'$  follows from the  $CPA^D$  security of  $S$  and the fact that the approximation noise  $g(m)$  is independent of  $S$  secret key material (which is the only information not available to a  $CPA_1^D$  adversary). So to break the  $CPA_{SC}^D$  security of  $S'$ , an attacker has to exploit the challenge ciphertext. However, after the adversary issues his or her unique challenge request with  $m_0 \neq m_1$  to get  $m_0^*$ ,  $m_1^*$  and  $c^* = \text{Enc}'(m_b^*)$ , he or she can only

- asks for a decryption of  $c^*$ , which is rejected by the  $CPA_{SC}^D$  decryption oracle since  $m_0^* \neq m_1^*$ ,
- asks for a decryption of  $c_{\text{sum}} = \text{Eval}'(\text{sum}, c^*, c^{(1)}, \dots, c^{(K)})$  where<sup>9</sup> all the  $c^{(i)}$ 's are such that  $m_0^{(i)} = m_1^{(i)}$ . The decryption of  $c_{\text{sum}}$  is also blocked by the  $CPA_{SC}^D$  decryption oracle since

$$m_0^* + \sum_i m_0^{(i)} \neq m_1^* + \sum_i m_1^{(i)}.$$

So the adversary can learn nothing on  $b$  and the  $CPA_{SC}^D$ -security of  $S'$  follows.

$S'$  is not  $CPA^D$ -secure. Now let the adversary issue two challenge requests with  $m_0 \neq m_1$  getting,

$$m_0^*, m_1^*, c^* = \text{Enc}'(m_b^*),$$

as well as

$$m_0^\dagger, m_1^\dagger, c^\dagger = \text{Enc}'(m_b^\dagger),$$

such that  $Z = m_0^* + m_0^\dagger = m_1^* + m_1^\dagger$  and  $g(m_0^*) + g(m_0^\dagger) \neq g(m_1^*) + g(m_1^\dagger)$ . The adversary then computes

$$c_{\text{sum}} = \text{Eval}'(\text{sum}, c^*, c^\dagger)$$

which the  $CPA^D$  decryption oracle accepts since the left and right evaluations both give  $Z$  i.e.,  $c_{\text{sum}}$  is an encryption of  $Z$  with respect to  $S'$ . However, with respect to  $S$ ,  $c_{\text{sum}}$  is an encryption of  $Z + g(m_b^*) + g(m_b^\dagger)$ , hence the adversary gets

$$\text{Dec}'(c_{\text{sum}}) = \text{Dec}(c_{\text{sum}}) = Z + g(m_b^*) + g(m_b^\dagger)$$

as a result of a decryption request on  $c_{\text{sum}}$ . Since  $g(m_b^*) + g(m_b^\dagger) \neq g(m_b^* + m_b^\dagger)$  and  $g(m_0^*) + g(m_0^\dagger) \neq g(m_1^*) + g(m_1^\dagger)$ , the adversary recovers  $b$  with certainty, leading the claim.  $\square$

As an example of concrete values for the attack in the above proof we can consider  $g(x) = \lfloor x/B \rfloor^2$  with e.g.  $B = 10000$ . The adversary then chooses  $m_0^* = m_0^\dagger = 10000$  as well as  $m_1^* = 0$  and  $m_1^\dagger = 20000$ . With these parameters,  $Z = 20000$  and  $\text{Dec}'(c_{\text{sum}}) = 20002$  when  $b = 0$  or  $20004$  when  $b = 1$ .

We now further establish the separation result without the restriction to additive HE schemes.

<sup>9</sup> Without loss of generality we assume that  $c^*$  appears only once and in the first place in the  $\text{Eval}'(\text{sum}, \dots)$  arguments.

**Corollary 1** ( $\text{CPA}_{\text{SC}}^D \not\Rightarrow \text{CPA}^D$ ). *In the general regime, if there exists an FHE scheme  $S$  which is  $\text{CPA}^D$ -secure, then there exists an FHE scheme  $S'$  which is  $\text{CPA}_{\text{SC}}^D$ -secure and  $\text{CPA}^D$ -insecure.*

*Proof.* So let us start with a  $\text{CPA}^D$ -secure FHE scheme

$$S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Add}, \text{Mul}).$$

Then, consider the scheme  $S' = (\text{KeyGen}', \text{Enc}', \text{Dec}', \text{Add}', \text{Mul}')$  such that

$$\text{Enc}'(m) = (\text{Enc}(m), \text{Enc}(g(m))) = (c_0, c_1) = c,$$

(with  $g$  being the same as in the proof of the previous Proposition 10) and

$$\text{Dec}'(c) = \text{Dec}(c_0) + \text{Dec}(c_1).$$

Now we consider the following homomorphic addition and multiplication operators:

$$\text{Add}'(c, c') = (\text{Add}(c_0, c'_0), \text{Add}(c_1, c'_1))$$

and

$$\text{Mul}'(c, c') = (\text{Mul}(c_0, c'_0), \text{Enc}(0)).$$

Essentially, in  $S'$ , the approximation noise is encrypted separately to the message (to make both easily separable) and the multiplication operator resets that noise. So as soon as the adversary performs a multiplication, he or she closes the information channel that function  $g$  opens. Although  $S'$  is fully homomorphic, we therefore end up in the same conditions as in the proof of Proposition 10. Indeed, in any successful  $\text{CPA}_{\text{SC}}^D$  or  $\text{CPA}^D$  attack involving both homomorphic additions and multiplications, the multiplications are redundant.  $\square$

Since  $\text{CPA}^D$  security trivially implies  $\text{CPA}_{\text{SC}}^D$  security, the following result is a direct consequence of Corollary 1.

**Proposition 11.** *In the general regime,  $\text{CPA}^D$  is strictly stronger than  $\text{CPA}_{\text{SC}}^D$ .*

Note that the above proposition (partly) settles the question of the relationship between single and multiple-challenge  $\text{CPA}^D$  security that was left open in [17] (p. 14): “It remains an interesting open question to find out the relationship between  $(q; \ell)$ -IND- $\text{CPA}^D$  and  $(q; \ell + 1)$ -IND- $\text{CPA}^D$  securities (and same for SIM- $\text{CPA}^D$ ).” (in their notations,  $\ell$  is the number of queries to the encryption oracle with  $m_0 \neq m_1$  and  $q$  the number of decryption queries). Indeed, we have shown above that, even for  $q = 1$ , there exists homomorphic schemes which are  $(q; 1)$ - $\text{CPA}^D$ -secure while being  $(q, 2)$ - $\text{CPA}^D$ -insecure (in the notations of [17]). Of course, to completely settle the above question we also need to prove separation or equivalence of  $(q, \ell)$ - $\text{CPA}^D$ -security and  $(q, \ell + 1)$ - $\text{CPA}^D$ -security with  $\ell > 2$ . We leave this remaining question as an open problem, although we conjecture that  $(q, \ell)$ - $\text{CPA}^D$  security is equivalent to  $(q, 2)$ - $\text{CPA}^D$  security for all  $\ell \geq 2$ .

## 5.2 Relations between $vCCA_{SC}$ and $vCCA$ security

Recall that [19] defines and studies only  $vCCA_{SC}$  security (also recall that in that paper it is simply referred to as  $vCCA$ ). The question of the relationship between  $vCCA_{SC}$  security and LOR- $vCCA$  security (or simply  $vCCA$  security with our present conventions) therefore deserves to be settled. So let LOR- $vCCA$  denote the multiple challenges variant of  $vCCA$  in which the decryption oracle condition (7) is replaced by

$$C^* \cap \{c_1, \dots, c_K\} \neq \emptyset, \quad (12)$$

where  $C^*$  is the set of challenge ciphertexts. We then have the proposition below.

**Proposition 12.**  *$vCCA$  security is equivalent to  $vCCA_{SC}$  security.*

*Proof.* The standard hybrid argument, e.g. in the proof of [3, Theorem 4] (which corresponds to Theorem 1 for both CPA and CCA), holds without modification, since an adversary confronted to a hybrid game (where the encryption oracle replies according to  $b = 0$ , up to a random point after which it replies according to  $b = 1$ ) *cannot* detect the transition between the first and second phases. Indeed, although challenge ciphertexts from both phases may interact via homomorphic evaluations, condition (12) above guarantees that such interactions lead to ciphertexts rejected by the above LOR- $vCCA$  decryption oracle.  $\square$

Note that in the above proof, we make no assumption about the correctness of the FHE scheme, so the above equivalence holds in the general regime.

As a consequence of Proposition 8, this establishes that the single challenge variant of  $vCCA^D$  security,  $vCCA_{SC}^D$ , is already strictly stronger than the multiple challenge variant of  $vCCA$ .

## 5.3 Relations between $vCCA_{SC}^D$ and $vCCA^D$ security

Lastly, we now unveil the relationship between the single and multiple challenge variant of  $vCCA^D$  security. Similarly to the  $CPA^D$  case, there is a distinction between the correct and general regime.

**Proposition 13.** *In the correct regime,  $vCCA^D$  is equivalent to  $vCCA_{SC}^D$ .*

*Proof.* The claim follows directly from proposition 7 (and its straightforward generalization to the multiple challenges variants of  $vCCA$  and  $vCCA^D$ ) as well as proposition 12 above.  $\square$

We now turn to the general regime and, as in Sect. 5.1, first consider the case of linearly homomorphic schemes.

**Proposition 14.** *In the general regime, if there exists an additive HE scheme  $S$  which is  $vCCA^D$ -secure, then there exists an additive HE scheme  $S'$  which is  $vCCA_{SC}^D$ -secure and  $vCCA^D$ -insecure.*

*Proof.* We proceed similarly to the proof of Proposition 10. So let us start with a  $vCCA^D$ -secure *additive* HE scheme  $S = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$  from which we build the scheme  $S'$  with the only modification that

$$\text{Enc}'(m) = \text{Enc}(m + g(m)),$$

where  $g$  is as in the proof of Proposition 10.

$S'$  is  $vCCA_{SC}^D$ -secure. Since  $vCCA^D$  security trivially implies  $vCCA_{SC}^D$  security,  $S$  is  $vCCA_{SC}^D$ -secure. Let  $\mathcal{A}$  be a successful adversary against the  $vCCA_{SC}^D$  security of  $S'$ . It is then easy to build an adversary  $\mathcal{B}$  against the  $vCCA_{SC}^D$  security of  $S$ . Indeed,  $\mathcal{B}$  simulates  $\mathcal{A}$  encryption and challenge requests simply by adding  $g(m)$  to  $m$ . All other request are transferred “as is” by  $\mathcal{B}$  to the  $vCCA_{SC}^D$  game against  $S$ .

$S'$  is  $vCCA^D$ -insecure. Identical to proof of Proposition 10: the  $vCCA^D$  decryption oracle accepts the  $c_{\text{sum}}$  ciphertext as, recall (9) and (10),  $\text{left}(c_{\text{sum}}) = \text{right}(c_{\text{sum}}) = Z$  (as an encryption of  $Z$  with respect to  $S'$ ).  $\square$

**Corollary 2** ( $vCCA_{SC}^D \not\Rightarrow vCCA^D$ ). *In the general regime, if there exists an FHE scheme  $S$  which is  $vCCA^D$ -secure, then there exists an FHE scheme  $S'$  which is  $vCCA_{SC}^D$ -secure and  $vCCA^D$ -insecure.*

*Proof.* Identical to that of Corollary 1.  $\square$

Since  $vCCA^D$  security trivially implies  $vCCA_{SC}^D$  security, the following result is a direct consequence of Corollary 2.

**Proposition 15.** *In the general regime,  $vCCA^D$  is strictly stronger than  $vCCA_{SC}^D$ .*

It follows that  $vCCA^D$  security is the strongest CCA security notion so far known to be achievable by FHE in the general regime.

## 6 Construction blueprints

In this section, we revisit the four construction blueprints proposed in [19] to leverage CPA-secure and correct FHE schemes into  $vCCA_{SC}$ -secure schemes, and study both their applicability in the general regime where approximate FHE are allowed as well as their  $vCCA^D$  security. As such, we emphasize that the constructions themselves are not new.

### 6.1 Private key constructions

We first consider the Encrypt-then-Sign construction blueprint proposed in [19]. The construction is built over a public-key FHE scheme  $\mathcal{E}_H$ , a public-key signature scheme  $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify})$ , as well as a designated-verifier SNARK  $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ , to obtain an encryption scheme  $\mathcal{E}_H^*$  defined as follows.

- $\mathcal{E}_H^*.\text{KeyGen}$ : run  $\mathcal{E}_H.\text{KeyGen}$ ,  $\Sigma.\text{KeyGen}$ ,  $\Pi.\text{Setup}$ , let  $\text{ek} = (\mathcal{E}_H.\text{pk}, \text{Sign}.\text{sk}, \Pi.\sigma)$  as well as  $\text{dk} = (\mathcal{E}_H.\text{sk}, \text{Sign}.\text{pk}, \Pi.\tau)$ .

- $\mathcal{E}_H^*$ .Enc: given  $m \in \mathcal{P}$  generate ciphertext  $(c, \pi) = (\mathcal{E}_H.\text{Enc}(m), \Sigma.\text{Sign}(c))$ .
- $\mathcal{E}_H^*$ .Eval: given ciphertexts  $(c_1, \pi_1), \dots, (c_K, \pi_K)$  such that  $\Sigma.\text{Verify}(c_i, \pi_i), \forall i$ , generate ciphertext  $(c_e, \pi_e)$  such that

$$c_e = \mathcal{E}_H.\text{Eval}(f, c_1, \dots, c_K),$$

and

$$\pi_e = \Pi.\text{Prove}(\Sigma.\text{Verify}(c_i, \pi_i), \forall i \wedge c_e = \mathcal{E}_H.\text{Eval}(f, c_1, \dots, c_K)).$$

- $\mathcal{E}_H^*$ .Dec: given ciphertext  $(c, \pi)$  return  $\mathcal{E}_H.\text{Dec}(c)$  when  $\Sigma.\text{Verify}(c, \pi) = \text{True}$  or  $\Pi.\text{Verify}(c, \pi) = \text{True}$  and  $\perp$  otherwise.

Intuitively, the essence of this construction is to rely on a trusted encryption oracle that generates only well-formed ciphertexts with respect to  $\mathcal{E}_H$  and signs them such that they are recognizable. As such, this construction is not public key, due to the presence of  $\Sigma.\text{sk}$  in  $\text{ek}$ . To improve the practicality of this construction (at the expense of input privacy), it is also possible to modify it such that the statements for which the  $\Pi$  outputs a proof during Eval does not have to include the verification of the input ciphertexts' signatures. In that case, both the input *and* output ciphertexts must be available to the decryption algorithm which is then responsible for verifying the signatures of the former. When this is the case, Eval and Dec are therefore modified as follows:

- $\mathcal{E}_H^*$ .Eval: given ciphertexts  $(c_1, \pi_1), \dots, (c_K, \pi_K)$  compute
 
$$(c_e = \mathcal{E}_H.\text{Eval}(f, c_1, \dots, c_K), \pi_e = \Pi.\text{Prove}(c' = \mathcal{E}_H.\text{Eval}(f, c_1, \dots, c_K)))$$
 and return ciphertext  $((c_e, \pi_e), (c_1, \pi_1), \dots, (c_K, \pi_K))$ .
- $\mathcal{E}_H^*$ .Dec (fresh ciphertext): given  $(c, \pi)$ , return  $\mathcal{E}_H.\text{Dec}(c)$  if  $\Sigma.\text{Verify}(c, \pi) = \text{True}$  and  $\perp$  otherwise.
- $\mathcal{E}_H^*$ .Dec (evaluated ciphertext): given  $((c_e, \pi_e), (c_1, \pi_1), \dots, (c_K, \pi_K))$  return  $\mathcal{E}_H.\text{Dec}(c_e)$  when  $\Pi.\text{Verify}(c_e, \pi_e) = \text{True}$  and  $\Sigma.\text{Verify}(c_i, \pi_i) = \text{True}, \forall i$  and  $\perp$  otherwise.

In this modified construction, the signature scheme can be replaced by a MAC  $M = (\text{KeyGen}, \text{Tag}, \text{Verify})$ , leading to the Encrypt-then-MAC blueprint of [19].

The above Encrypt-then-Sign construction was proved in [19] to lead a  $\text{vCCA}_{\text{SC}}$ -secure scheme from a CPA-secure *correct* FHE scheme, a SUF-CMA-secure signature scheme and a simulation-sound extractable SNARK (which implies the existence of an extractor as defined in Sect. 3.2). Their proof technique consists in showing that a successful  $\text{vCCA}_{\text{SC}}$  attack over scheme  $\mathcal{E}_H^*$  implies a successful CCA2 attack against the *private key* (non homomorphic) encryption scheme obtained by associating  $\mathcal{E}_H$  and  $\Sigma$ . It turns out that the security of this construction goes beyond this setting as we now prove that the above Encrypt-then-Sign blueprint in fact offers  $\text{vCCA}^D$  security beyond the correct FHE regime, as long as we instantiate it from a  $\text{CPA}^D$ -secure rather than CPA-secure/correct FHE.

To do so, we now prove that an adversary breaking  $\text{vCCA}^D$  security on the above construction also breaks the  $\text{CPA}^D$  security of the underlying FHE scheme, thus leading to a contradiction with the the  $\text{CPA}^D$  security of the latter.



**Proposition 16.** *Let  $\mathcal{A}$  be an adversary against the  $vCCA^D$  security of  $\mathcal{E}_H^*$ , then, under the assumption that  $\Sigma$  is  $SUF-CMA$  secure and  $\Pi$  is straightline-extractable, there exists an adversary  $\mathcal{B}$  against the  $CPA^D$  security of  $\mathcal{E}_H$  which uses  $\mathcal{A}$  as a subroutine.*

*Proof.* Because the scheme is not public key, all encryptions are performed through  $vCCA^D$  game encryption requests. Also,  $\mathcal{B}$  mimics the  $CPA^D$  game state and initially starts with an empty state  $S^{\mathcal{B}} = []$ . Then, given a ciphertext  $c$ , we denote by  $\text{idx}(c)$ , its index in the game state  $S^{\mathcal{B}}$  (which is the same as the index at which the ciphertext is stored in the  $CPA^D$  game state  $S$ ). Then  $\mathcal{A}$  can issue the following requests, which  $\mathcal{B}$  emulates as follows:

- When  $\mathcal{A}$  issues a  $vCCA^D$  game encryption request for plaintext  $m$ , then  $\mathcal{B}$  issues a  $CPA^D$  game encryption request to get ciphertext  $c$ . He or she then does  $S^{\mathcal{B}} := [S^{\mathcal{B}}; (m, m, c)]$ , generates  $\pi = \Sigma.\text{Sign}(c)$  and return  $(c, \pi)$  to  $\mathcal{A}$ .
- When  $\mathcal{A}$  issues a  $vCCA^D$  game challenge request for plaintexts  $m_0, m_1$  ( $m_0 \neq m_1$ ), then  $\mathcal{B}$  issues a  $CPA^D$  game challenge request to get ciphertext  $c$ . He or she then does  $S^{\mathcal{B}} := [S^{\mathcal{B}}; (m_0, m_1, c)]$ , generates  $\pi = \Sigma.\text{Sign}(c)$  and return  $(c, \pi)$  to  $\mathcal{A}$ .
- When  $\mathcal{A}$  issues a  $vCCA^D$  game decryption request for ciphertext  $(c, \pi)$ , proceeds as follows:
  - If  $\Sigma.\text{Verify}(c, \pi) = \text{True}$  (i.e., when  $(c, \pi)$  is a fresh well-formed ciphertext) he or she issues a  $CPA^D$  game decryption request on  $\text{idx}(c)$  and return the result to  $\mathcal{A}$ .
  - If  $\Pi.\text{Verify}(c, \pi) = \text{True}$ ,  $\mathcal{B}$  invokes  $\Pi$ 's Extract procedure over  $(c; \pi)$  to get  $f; (c_1; \pi_1), \dots, (c_K; \pi_K)$ . In this case, if  $\Sigma.\text{Verify}(c_i, \pi_i) = \text{False}$  for some  $i$  he or she returns  $\perp$  to  $\mathcal{A}$ . Otherwise (when  $(c, \pi)$  is a well-formed evaluated ciphertext), he or she then does a  $CPA^D$  game evaluation request with parameters  $f; \text{idx}(c_1), \dots, \text{idx}(c_K)$  and get ciphertext  $c' = c$  (recall that Eval is deterministic) in return (with also the effect of adding  $c' = c$  along with the associated left and right cleartext evaluations in the  $CPA^D$  game state), he or she also performs the left and right cleartext evaluations for his or herself to get

$$m'_0 = f(S^{\mathcal{B}}[\text{idx}(c_1)].m_0, \dots, S^{\mathcal{B}}[\text{idx}(c_K)].m_0)$$

and

$$m'_1 = f(S^{\mathcal{B}}[\text{idx}(c_1)].m_1, \dots, S^{\mathcal{B}}[\text{idx}(c_K)].m_1)$$

and do  $S^{\mathcal{B}} := [S^{\mathcal{B}}; (m'_0, m'_1, c')]$ . Finally,  $\mathcal{B}$  issues a  $CPA^D$  game decryption request with  $\text{idx}(c') = |\mathcal{S}^{\mathcal{B}}|$  and returns the result to  $\mathcal{A}$ .

- Lastly, when  $\Sigma.\text{Verify}(c, \pi) = \text{False}$ ,  $\mathcal{B}$  returns  $\perp$  to  $\mathcal{A}$ .

Thus,  $\mathcal{B}$  can duly simulate all the  $vCCA^D$  game requests issued by  $\mathcal{A}$ . Because the simulator  $\mathcal{B}$  does not provide an evaluation oracle and, hence, does not generate any proofs ( $\mathcal{B}$  only invokes the SNARK extractor when  $\Pi.\text{Verify}(\pi) = \text{True}$  on a proof  $\pi$  he or she has not generated),  $\mathcal{A}$  has only access to proofs he or she has generated by his or herself and, as such, has no access to any simulator

oracle against  $\Pi$ . It follows that the straightline-extractability of  $\Pi$  is sufficient for the  $\text{vCCA}^D$  security of the construction to hold without any additional non-malleability property for the SNARK.  $\square$

## 6.2 Public-key constructions

We now consider the public key, designated-verifier<sup>10</sup> construction blueprint proposed in [19] which we refer to as the *CCA2-Companion-Ciphertext* approach in this paper. The construction is built over a public-key FHE scheme  $\mathcal{E}_H$ , a public-key (CCA2-secure) scheme  $\mathcal{E} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ , and a designated-verifier SNARK  $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ , to obtain encryption scheme  $\mathcal{E}_H^*$ :

- $\mathcal{E}_H^*.\text{KeyGen}$ : run  $\mathcal{E}_H.\text{KeyGen}$ ,  $\mathcal{E}.\text{KeyGen}$ ,  $\Pi.\text{Setup}$ , let  $\text{ek} = (\mathcal{E}_H.\text{pk}, \mathcal{E}.\text{pk}, \Pi.\sigma)$  as well as  $\text{dk} = (\mathcal{E}_H.\text{sk}, \mathcal{E}.\text{sk}, \Pi.\tau)$ .
- $\mathcal{E}_H^*.\text{Enc}$ : given  $m \in \mathcal{P}$  generate ciphertext  $(c, c') = (\mathcal{E}_H.\text{Enc}(m; r), \mathcal{E}.\text{Enc}(m|r))$  (where  $|$  denotes the concatenation operator).
- $\mathcal{E}_H^*.\text{Eval}$ : given ciphertexts  $(c_1, c'_1), \dots, (c_K, c'_K)$ , compute

$$(c_e = \mathcal{E}_H.\text{Eval}(f, c_1, \dots, c_K), \pi_e = \Pi.\text{Prove}(c_e = \mathcal{E}_H.\text{Eval}(f, c_1, \dots, c_K))).$$

and return ciphertext  $((c_e, \pi_e), (c_1, c'_1), \dots, (c_K, c'_K))$ .

- $\mathcal{E}_H^*.\text{Dec}$  (fresh ciphertext): given  $(c, c')$ , return  $\mathcal{E}_H.\text{Dec}(c)$  if  $\text{Verif}((c, c') = \text{True}$  and  $\perp$  otherwise<sup>11</sup>.
- $\mathcal{E}_H^*.\text{Dec}$  (evaluated ciphertext): given ciphertext  $((c_e, \pi_e), (c_1, c'_1), \dots, (c_K, c'_K))$  return  $\mathcal{E}_H.\text{Dec}(c_e)$  when  $\text{Verif}(c_i, c'_i) = \text{True}, \forall i$  and  $\Pi.\text{Verify}(c_e, \pi_e)$ . Return  $\perp$  otherwise.

Where  $\text{Verif}((c, c'))$  runs  $(m', r') = \mathcal{E}.\text{Dec}(c')$  and returns  $\text{True}$  iff

$$c = \mathcal{E}_H.\text{Enc}(m', r').$$

Intuitively, the essence of this construction is to define fresh ciphertexts as the association of an FHE ciphertext encrypting  $m$  by means of randomness  $r$  and another ciphertext encrypting the concatenation of  $m$  and  $r$  under a CCA2-secure encryption scheme. This allows to verify the well-formedness of these fresh ciphertexts by first decrypting the companion CCA2 ciphertext to

<sup>10</sup> In [19] terminology this just tells whether or not the well-formedness of fresh ciphertexts is verifiable publicly or privately. This is independent of whether the SNARK is publicly verifiable or designated-verifier.

<sup>11</sup> Here, we slightly depart from the construction of [19] in the following sense. When decrypting a fresh ciphertext  $(c, c')$ , they indeed proceed by calling  $\mathcal{E}.\text{Dec}(c')$  to get  $m$  and  $r$  and return  $m$  when  $\text{Verif}((c, c')) = \text{True}$  (i.e., they never decrypt the FHE ciphertext). We, on the contrary, return  $\mathcal{E}_H.\text{Dec}(c)$  when  $\text{Verif}((c, c')) = \text{True}$ . Although both options are equivalent under the correctness assumption of  $\mathcal{E}_H$ , this is not the case in the general regime. However, when  $\mathcal{E}_H$  is  $\text{CPA}^D$ -secure (as required for the construction in the general regime), this difference has no security implications.

recover  $m$  and  $r$ <sup>12</sup> and then checking that the associated FHE ciphertext is indeed equal to  $\mathcal{E}_H.\text{Enc}(m, r)$  (note that the verification may succeed when  $\mathcal{E}_H.\text{Dec}(\mathcal{E}_H.\text{Enc}(m, r)) \neq m$  which is what we want when the correctness assumption does not hold for  $\mathcal{E}_H$ ). The approach, however, has the drawback that it cannot achieve any form of input privacy as this verification requires the knowledge of the CCA2 scheme decryption key and, as a consequence, can be performed only in the decryption function of the overall scheme, requiring the availability of the input ciphertexts.

The above CCA2-Companion-Ciphertext blueprint was proved in [19] to lead a  $\text{vCCA}_{\text{SC}}$ -secure scheme from a CPA-secure *correct* FHE scheme, a CCA2-secure scheme and a simulation-sound extractable SNARK (which implies the existence of an extractor as defined in Sect. 3.2). Their proof technique consists in showing that a successful  $\text{vCCA}_{\text{SC}}$  attack over scheme  $\mathcal{E}_H^*$  implies a successful CCA2 attack on the companion CCA2 scheme. As in the previous section, it turns out that the security of this construction goes beyond this setting as we now prove that it also achieves  $\text{vCCA}^D$  security in the general regime, as long as we instantiate it from a  $\text{CPA}^D$ -secure rather than CPA-secure/correct FHE.

**Proposition 17.** *Let  $\mathcal{A}$  be an adversary against the  $\text{vCCA}^D$  security of  $\mathcal{E}_H^*$ , then, under the assumption that  $\mathcal{E}$  is CCA2-secure and  $\Pi$  is straightline-extractable, there exists an adversary  $\mathcal{B}$  against the  $\text{CPA}^D$  security of  $\mathcal{E}_H$  which uses  $\mathcal{A}$  as a subroutine.*

*Proof.* The proof is quite similar to that of Proposition 16 except that we slightly modify the  $\text{CPA}^D$  encryption oracle (but *not* the challenge oracle) such that it further takes randomness  $r$  as a parameter (this is benign for public-key FHE schemes, and all FHE schemes in this paper are public key). When  $\mathcal{B}$  processes a decryption request (assuming evaluated ciphertexts), the input ciphertexts which are challenge ones are already in his or her internal state (ditto for the  $\text{CPA}^D$  defender). For the input ciphertext which are challenge-independent, the message and randomness are recovered by  $\mathcal{B}$  via the decryption of the CCA2 companion ciphertexts. Then  $\mathcal{B}$  can issue the proper  $\text{CPA}^D$  encryption requests (with the additional randomness parameter) to populate the  $\text{CPA}^D$  defender state (and his or her mirrored one at the same time). Then  $\mathcal{B}$  can issue the  $\text{CPA}^D$  evaluation request with the appropriate game state indices to obtain  $c' = c$  (due to the deterministic evaluation assumption) and add it to the  $\text{CPA}^D$  defender game state. Finally,  $\mathcal{B}$  issues the  $\text{CPA}^D$  decryption request with  $\text{idx}(c)$ .  $\square$

Lastly, let us emphasize that the Naor-Yung-based [20] construct of [19], which, in order to encrypt a plaintext, associates two FHE ciphertexts of this plaintext under different keys, and bind them by a proof that these two ciphertexts are encrypting the same plaintext, requires correctness. Indeed, in [20] (definition 3.4), the scheme used in the construction must verify  $\forall m \in \mathcal{P}, \forall r \in$

<sup>12</sup> As such, in the CCA2-Companion-Ciphertext construct, we exactly get the additional extractor  $\text{Extract}'$  needed in the  $\text{vCCA}^D$  game definition in the public key case (Sect. 3.3).

$\{0;1\}^{p(n)}$ ,  $\text{Dec}(\text{Enc}(m, r)) = m$ . As such it is not applicable in the general case where approximate FHE schemes are allowed.

## 7 Conclusion and future work

Following the work of Manulis & Nguyen [19] as well as the improvements on that work we presented in this paper, designing practical FHE-style malleable schemes enforcing CCA security properties beyond the CCA1 barrier seems within reach, at least for specific applications. Indeed, recent advances in SNARK for ring arithmetic, such as [12], give us the necessary toolbox for attempting concrete instantiations of the construction blueprints discussed in Sect. 6. Furthermore, in many usual applications of FHE, the set of algorithms that needs to run in the encrypted domain is very limited (for example, a FHE aggregation server involved in a Federated Training protocol for a machine learning model may only have to run a simple average [13] or a majority voting algorithm [14,15]). This gives us hope to be able to design practical  $v\text{CCA}^D$ -secure schemes with simplified SNARK or Verifiable Computing techniques tailored to these sets of algorithms. Lastly, it will also be interesting to investigate which building blocks are friendly towards each others e.g., finding “SNARK friendly” signature schemes for concrete instantiation of the Encrypt-then-Sign blueprint.

Also, following a recent burst of new  $\text{CPA}^D$  attacks on both noise-flooded CKKS and “exact” schemes such as BFV, BGV or TFHE [16,7,8] new FHE security paradigms are being proposed. As an example, Alexandru et al. [1] have proposed a new *weaker* variant of  $\text{CPA}^D$  security, termed application-aware security. In essence, this new definition acknowledges that for non-exact FHE schemes,  $\text{CPA}^D$  security should be defined relatively to a function class  $\mathcal{F}_C$  and a ciphertext noise estimation strategy, rather than absolutely. With respect to that new security notion, the cryptosystem parameters should then be set relatively to these, and the homomorphic evaluations should be limited to the functions or circuits in the class. However, one of the main drawbacks of the application-aware approach is that the burden of enforcing the above constraints lies, so far, solely on the library user’s shoulders (see also [2] and, in particular, its new Sect. 2.6.1). As a starting point, an interesting line of research would then be to connect the application-aware paradigm with both  $v\text{CCA}$  and  $v\text{CCA}^D$  security notions by defining new weaker variants of these notions, e.g.  $\mathcal{F}_C$ - $v\text{CCA}$  and  $\mathcal{F}_C$ - $v\text{CCA}^D$  security, for leveraging somewhat correct (and  $\text{CPA}$ ) or  $\text{CPA}^D$  schemes, i.e. schemes achieving correctness or  $\text{CPA}^D$  security only over  $\mathcal{F}_C$ , to CCA security levels. For example, the  $v\text{CCA}_{\text{SC}}$  and  $v\text{CCA}^D$  decryption oracles may further check that  $f \notin \mathcal{F}_C$  in conditions (7) and (8), respectively (which is precisely what is suggested for the  $\text{CPA}^D$  game evaluation oracle in [1]). Our intuitions are that the picture depicted in this paper will be relatively similar for these restricted security notions but we leave this for further work. However, a difficult point will be to capture the dependency of the application aware approach upon noise estimation strategies in meaningful CCA security notions,

with the hope of achieving both beyond CCA1 security and relieving the library users of the burden of enforcing by hand the constraints of that paradigm.

## References

1. Alexandru, A., Badawi, A.A., Micciancio, D., Polyakov, Y.: Application-aware approximate homomorphic encryption: Configuring FHE for practical use. Tech. Rep. 203, IACR ePrint (2024)
2. Badawi, A.A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Quah, I., Polyakov, Y., Saraswathy, R.V., Rohloff, K., Saylor, J., Suponitsky, D., Triplett, M., Vaikuntanathan, V., Zucca, V.: Openfhe: Open-source fully homomorphic encryption library. In: WAHC. pp. 53–63 (2022)
3. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: SFCS. pp. 394–403 (1997)
4. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: CRYPTO. pp. 26–45 (1998)
5. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical gapsvp. In: CRYPTO. pp. 868–886 (2012)
6. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM TOCT pp. 1–36 (2014)
7. Checri, M., Sirdey, R., Boudguiga, A., Bultel, J.P.: On the practical cpad security of “exact” and threshold FHE schemes. In: CRYPTO (2024)
8. Cheon, J.H., Choe, H., Passelègue, A., Stehlé, D., Suvanto, E.: Attacks against the IND-CPAD security of exact FHE schemes. Tech. Rep. 127, IACR ePrint (2024)
9. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: ASIACRYPT, pp. 409–437 (2017)
10. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: ASIACRYPT (2016)
11. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR ePrint (2012)
12. Ganesh, C., Nitulescu, A., Soria-Vazquez, E.: Rinocchio: Snarks for ring arithmetic. J. Cryptol. p. 41 (2023)
13. Grivet-Sébert, A., Checri, M., Stan, O., Sirdey, R., Gouy-Pailler, C.: Combining homomorphic encryption and differential privacy in federated learning. In: IEEE PST. pp. 1–7 (2023)
14. Grivet-Sébert, A., Pinot, R., Zuber, M., Gouy-Pailler, C., Sirdey, R.: SPEED: secure, private, and efficient deep learning. Machine Learning pp. 675–694 (2021)
15. Grivet-Sébert, A., Zuber, M., Stan, O., Sirdey, R., Gouy-Pailler, C.: A probabilistic design for practical homomorphic majority voting with intrinsic differential privacy. In: WAHC. pp. 47–58 (2023)
16. Guo, Q., Nabokov, D., Suvanto, E., Johansson, T.: Key recovery attacks on approximate homomorphic encryption with nonworst-case noise flooding countermeasures. In: Usenix Security (2024)
17. Li, B., Micciancio, D.: On the security of homomorphic encryption on approximate numbers. In: EUROCRYPT. pp. 648–677 (2021)
18. Li, B., Micciancio, D., Schultz, M., Sorrell, J.: Securing approximate homomorphic encryption using differential privacy. In: CRYPTO. pp. 560–589 (2022)

19. Manulis, M., Nguyen, J.: Fully homomorphic encryption beyond IND-CCA1 security: Integrity through verifiability. In: EUROCRYPT (2024)
20. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: ACM STOC. pp. 427–437 (1990)