Achieving "beyond CCA1" security for linearly homomorphic encryption, without SNARKs?*

Marina Checri, Pierre-Emmanuel Clet, Marc Renard and Renaud Sirdey

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France,
name.surname@cea.fr

Abstract. In the wake of Manulis and Nguyen's Eurocrypt'24 paper, new CCA security notions, vCCA and vCCA^D, and associated construction blueprints have been proposed to leverage either CPA or CPA^D secure FHE beyond the CCA1 security barrier. These two notions are the strongest CCA security notions so far achievable, respectively, by correct and approximate homomorphic schemes. However, the only known construction strategies intimately require advanced SNARK machinery, undermining their practicality. In this context, this paper is an attempt to achieve these advanced CCA security notions in the restricted case of linearly homomorphic encryption, without resorting to SNARKs. To do so, we investigate the relationship between the Linear-Only Homomorphism (LOH) assumption, an assumption that has been used for more than a decade at the core of several proof-of-knowledge constructions, and these two recent security notions (vCCA and vCCA^D). On the bright side, when working under the correctness assumption, we establish that the LOH property is sufficient to achieve vCCA security in both the private and public key settings. In the public key setting, we further show that a surprisingly simple and previously known Paillier-based construction also achieves this level of security, at only twice the cost of the baseline scheme. We then turn our attention to LWE-based schemes for which the Pandora box of decryption errors opens up. In the private key setting, we are able to achieve CPA^D and vCCA^D security but only in a fairly restrictive non-adaptive setting, in which vCCA^D collapses onto a weak relaxation of CCA1. Finally, we eventually achieve adaptive vCCA^D security provided that the number of ciphertexts given to the adversary is suitably restricted. While bridging the gap towards credible practicality requires further work, this is a first step towards obtaining linear homomorphic schemes achieving these recent CCA security notions by means only of relatively lightweight machinery.

Keywords: Homomorphic encryption · Linear-only homomorphism · CCA security.

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

1 Introduction

Since its inception more than ten years ago, Fully Homomorphic Encryption (FHE) has been the subject of much research towards more efficiency and better practicality. However, from a security perspective, FHE still raises several questions and challenges. In particular, all the FHE usable in practice, BFV [18,30], BGV [19], CKKS [25] and TFHE [26], achieve only CPA security and are also known 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. Recently, this topic is the subject of renewed attention at least from a theoretical viewpoint [47,21].

On the one hand, a number of correct linearly homomorphic schemes are known to achieve CCA1 security under standard or almost standard assumptions. This is the case, as recently established in [41], for a simple variant of the Paillier scheme where λ plaintext bits are forced to be zeroes or the Cramer-Shoup Lite scheme [28]. On the other hand, recent works investigating CCA1 or beyond-CCA1 security for FHE from a more theoretical perspective, do so by starting either from a CPA/correct [47] or CPA^D [21] secure scheme, and augment it with the machinery necessary for proving some form of plaintext awareness on fresh ciphertexts as well as correctness of evaluated ciphertexts derivation. Unfortunately, these generic construction blueprints are not easily amenable to efficient implementations in their full generality, essentially because they require powerful SNARK machinery.

A natural question is then the following: is it possible to achieve these new "beyond CCA1" security notions by resorting only to lightweight machinery? In particular, without using SNARKs? In this paper, we provide a first round of answers to this question in the case of *linearly* homomorphic encryption, using the Linear-Only Homomorphism (LOH) assumption [11] as a yardstick. We first establish a connection between the LOH assumption and the vCCA security notion recently introduced by Manulis and Nugyen [47] (CCA1$\prec$vCCA$\prec$CCA2), by showing that any CPA/correct linear HE satisfying this assumption also achieves vCCA security in both the private and public key settings. In the public key setting, we then show that a surprisingly simple (and previously known) construction based on the Paillier encryption scheme [51] is vCCA secure under the reasonable assumption that it has the LOH property. We then investigate LWE-based schemes which are notoriously prone to decryption errors. For such schemes, we have to craft a hierarchy of constructions in order to first achieve CPA^D security and, then, leverage on this to further achieve vCCA^D security (a strictly stronger generalization of vCCA security in the regime where the correctness assumption for the underlying homomorphic scheme is relaxed [21]) in the private key setting. We, however, only formally achieve this in a fairly restrictive *non-adaptive* setting (in the sense of [38]), in which vCCA^D collapses onto a weak relaxation of CCA1. Finally, we extend our line of LWE-based schemes to eventually achieve adaptive vCCA^D security under the conjectured

(adaptive) CPA^D security of our first construction and the restriction that the number of ciphertexts given to the adversary is suitably restricted.

Although generalizing our techniques to the FHE setting remains largely open, this work is a first step towards obtaining (linearly) homomorphic schemes that achieve the strongest CCA security notions so far known to be achievable by homomorphic schemes using relatively lightweight machinery.

1.1 Summary of contributions

The contributions of this paper are as follows:

- We first prove the general results that any (perfectly or statistically) correct CPA-secure linearly homomorphic scheme satisfying the Linear-Only Homomorphism assumption achieves vCCA security (hence also achieves CCA1 security) in both the private and public key settings.
- In the public key setting, we show that the (well-known) "two-ciphertexts" blueprint, which consists of sparsifying the ciphertext domain of a correct CPA-secure linearly homomorphic scheme by following a Knowledge-of-Exponent (KEA) template, achieves vCCA security under the assumption that it has the LOH property. We do so by explicitly considering the Paillier cryptosystem as the base scheme, but the approach can be expected to apply starting from any (statistically) correct CPA-secure linearly HE scheme.
- In the case of LWE-based linear HE, where we cannot operate under the correctness assumption, we focus on the *private key setting*:
  - Starting from the Regev scheme, we propose a multi-secret scheme with a ciphertext-dependent variance estimation procedure, which is then used to derive a safe-bound on the true ciphertext variance. This then allows proper smudging to occur within the decryption function of the scheme. We refer to this first baseline scheme as FS_1 and prove its CPA^D security *in the non-adaptive setting*, where the adversary specifies all its requests after seeing the public material of the scheme. In doing so, we positively answer an open question from [39] regarding the existence of schemes with dynamic error estimation achieving at least a weak form of CPA^D security. Furthermore, FS_1 is provably immune to the attacks in [39] on a "natural" class of schemes with dynamic error estimation.
  - We then build on that latter scheme to leverage the (well-known) approach consisting of applying the Knowledge-of-Exponent template to a multiple-secret variant of Regev (using independent key material for each slot) and show that it also is (non-adaptive) CPA^D secure. We refer to this scheme as FS_1' due to its connection to the previous one. Although we show that this scheme does not satisfy the LOH property (by exhibiting a concrete attack), we show that it achieves (non-adaptive) vCCA^D security under the credible assumption that it has a much weaker form of the LOH property. It should be emphasized that in the non-adaptive private-key setting, vCCA^D security collapses onto a very weak relaxation of CCA1 security, which we refer to as CCA0, in which the adversary specifies all its decryption requests before seeing any ciphertext.

However weak this relaxation is, we give concrete CCA0 attacks that are indeed thwarted by our scheme.

- We then study the adaptive setting under the conjectured (adaptive) CPA^D security of FS'_1 (a conjecture that is presently not disproved by any concrete attacks, including those in [39]). In that setting, we have to augment the latter scheme with a linearly homomorphic (keyed) hash function to credibly achieve a weak variant of LOH in which the adversary is allowed to see only a limited number of ciphertexts (with a number $n + K - O(1)$ of ciphertexts, where n is the LWE dimension and $K = O(\lambda)$ is the number of slots in a multi-secret LWE ciphertext). This further gives a generic pattern whereby any CPA^D secure scheme with *linear* homomorphic operators (should such schemes exist?) can be leveraged into a vCCA^D secure scheme (still under the above restriction on the number of ciphertexts). Because this contribution is thus more speculative, we provide its full details only in an appendix.
- As a bonus conribution, we also provide a new CPA^D attack path on the vanilla Regev scheme, its RLWE variant and the other mainstream FHE schemes which are based on them.

– As a last contribution, we discuss the practical limitations and concrete parameterization of our LWE-based schemes, although further research is needed to claim practicality. Still, in the correct regime, our results show that a vCCA secure linear HE scheme can be obtained at only twice the cost of a CPA secure one.

1.2 Paper organization

This paper is organized as follows. Section 2 covers the preliminaries. Then Section 3 focuses on the results and constructions we obtain under the correctness assumption of the underlying linear HE scheme, and Section 4 addresses our LWE-based constructions. Lastly, Sect. 5 concludes the paper by considerations towards practicality and perspectives.

2 Preliminaries

2.1 Basic notations

Given $l, u \in \mathbb{Z}^2$, 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 avoid possible ambiguities.

Given two discrete random variables X and Y we write $X \overset{i}{=} Y$ when the distribution of X and that of Y are such that $d(f_X, f_Y) \leq \mathrm{neg}(\lambda)$ where

$$d(f_X, f_Y) = \frac{1}{2} \sum_{k=-\infty}^{+\infty} |P(X = k) - P(Y = k)|$$

is the usual statistical distance. In this case, X and Y are said to be *statistically indistinguishable* or, for short, *indistinguishable* from one another. For simplicity sake, we write λ to denote both the computational security parameter *and* the statistical security parameter.

2.2 Basic definitions

We define an 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^λ, outputs an encryption key ek and a decryption key sk.
- Enc: on input $m \in \mathcal{P}$ and ek, outputs an encryption $c \in \mathcal{C}$ of m.
- Dec: on input $c \in \mathcal{C}$ and sk, outputs a[1] 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 $\mathsf{Enc}(m; r)$, with $m \in \mathcal{P}$ and $r \in \mathsf{COIN}$. In this latter case, the function $\mathsf{Enc} : \mathcal{P} \times \mathsf{COIN} \longrightarrow \mathcal{C}$ is deterministic. When ek is public, we say that $\mathcal{E}$ is a *public-key* encryption scheme. When for all $(\mathsf{ek}, \mathsf{sk}) \in \mathcal{K}$ and all $m \in \mathcal{P}$ we have that

$$\Pr_{r \in \mathsf{COIN}}(\mathsf{Dec}(\mathsf{Enc}(m; r)) \neq m) \leq \mathrm{neg}(\lambda), \tag{1}$$

we say that $\mathcal{E}$ is *statistically correct* or simply *correct*. When the above probability is zero, we talk of *perfect* correctness.

Given a function class $\mathcal{F}_H$, we define a homomorphic encryption (HE) scheme $\mathcal{E}_H$ as an encryption scheme augmented by a *deterministic*[2] polynomial-time algorithm Eval which, on input $f \in \mathcal{F}_H$ and $c_1, ..., c_L \in \mathcal{C}^L$, where L denotes the arity of function f, outputs a new *evaluated* ciphertext. When $\mathcal{E}_H$ satisfies condition (1) and when Eval is such that for all $(\mathsf{ek}, \mathsf{sk}) \in \mathcal{K}$, all $f \in \mathcal{F}_H$ and all $m_1, ..., m_L \in \mathcal{P}^L$

$$\Pr_{\vec{r} \in \mathsf{COIN}^L}(\mathsf{Dec}(\mathsf{Eval}(f, \mathsf{Enc}(m_1; r_1), ..., \mathsf{Enc}(m_L; r_L))) \neq f(m_1, ..., m_L)) \leq \mathrm{neg}(\lambda), \tag{2}$$

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. Consistently with [39], to avoid arbitrary schemes with unreliable Eval to be marketed as approximate HE schemes, we add an additional condition that, for some (small) $\varepsilon \geq 0$, the following holds

$$\Pr_{\vec{r} \in \mathsf{COIN}^L}(\|\mathsf{Dec}(\mathsf{Eval}(f, \mathsf{Enc}(m_1; r_1), ..., \mathsf{Enc}(m_L; r_L))) - f(m_1, ..., m_L)\|_\infty \leq \varepsilon) \geq \mu, \tag{3}$$

with[3] $\mu \geq \frac{3}{4}$. Lastly, a scheme such that $\varepsilon = 0$ and $\frac{3}{4} \leq \mu < 1 - \mathrm{neg}(\lambda)$ is said to be *somewhat correct*.

[1] Decryption may not be deterministic.

[2] As is the case for the mainstream FHE schemes such as BFV, BGV, TFHE and even CKKS.

[3] In practice μ is typically chosen above $1 - 2^{-40}$. In some contexts, e.g. [1], μ even has to be at least $1 - \mathrm{neg}(\lambda)$.

2.3 Security notions

CPA^D (and CPA$^D_0$) security. The CPA^D game has been introduced in the context of approximate FHE [38]. CPA^D security is a slight extension of CPA security (recalled in Sect. E.2) defined by the following Left-Or-Right multiple challenge security game.

Given a homomorphic encryption scheme $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$, an adversary $\mathcal{A}$ and value λ for the security parameter, the game is parameterized by a bit $\gamma \xleftarrow{\$} \{0,1\}$, unknown to $\mathcal{A}$, and an initially empty state S of message-message-ciphertext triplets:

- Key generation: run $(\mathsf{ek}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and, when the scheme is public-key, give ek to $\mathcal{A}$.
- Encryption request: When $\mathcal{A}$ queries $(\texttt{plaintext}, m)$, $m \in \mathcal{P}$ compute $c = \mathsf{Enc}(m)$, give c to $\mathcal{A}$ and update $S := [S; (m, m, c)]$.
- Challenge request: when $\mathcal{A}$ queries $(\texttt{test messages}, m_0, m_1)$, $m_0, m_1 \in \mathcal{P}^2$ $(m_0 \neq m_1)$ compute $c = \mathsf{Enc}(m_\gamma)$, give c to $\mathcal{A}$ and do $S := [S; (m_0, m_1, c)]$.
- Evaluation request: when $\mathcal{A}$ queries $(\texttt{eval}, f, l_0, \ldots, l_{L-1})$ $(l_i \in [\![0, |S|-1]\!], \forall i)$, compute
$$m'_0 = f(S[l_0].m_0, \ldots, S[l_{L-1}].m_0),$$
and
$$m'_1 = f(S[l_0].m_1, \ldots, S[l_{L-1}].m_1),$$
as well as
$$c' = \mathsf{Eval}(f, S[l_0].c, \ldots, S[l_{L-1}].c),$$
give c' to $\mathcal{A}$ and do
$$S := [S; (m'_0, m'_1, c')].$$
- Decryption request: when $\mathcal{A}$ queries $(\texttt{ciphertext}, l)$ $(l \in [\![0, |S|-1]\!])$ proceed as follows. If $S[l].m_0 \neq S[l].m_1$, return $\perp$ to $\mathcal{A}$. Otherwise return her $\mathsf{Dec}(S[l].c)$.
- Guessing stage (after polynomially many interleaved encryption, evaluation and decryption requests): when $\mathcal{A}$ outputs $(\texttt{guess}, \gamma')$, if $\gamma' = \gamma$ then $\mathcal{A}$ wins the game. Otherwise, $\mathcal{A}$ loses the game.

Remark that the decryption oracle accepts only ciphertexts from the game state which are necessarily *well-formed*, i.e. 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 ciphertexts from the game state (hence well-formed). As such, the above game does not capture any CCA aspects. Let us also emphasize that, in the above game, $\mathcal{A}$ controls the homomorphic calculations performed as f is included in the evaluation request parameters.

As defined in [38], CPA^D security admits a strictly weaker *non-adaptive* variant of interest to us in this paper. In this variant, the adversary specifies all its requests straight after seeing the public material of the scheme (if any). *We will refer to this variant as CPA$^D_0$*. Other variants have been defined and studied

in [21] (see Sect. E.1), where it is in particular shown that the single challenge variants of CPA^D security are strictly weaker than the multiple-challenge notion.

For completion, the definitions for CPA and CCA1 security are recalled in Sect. E.2. *Note that we define all the security games in this paper relatively to the above CPA^D game.*

vCCA, vCCA^D (and CCA0) security. As introduced in [47], vCCA security is a single challenge security notion. As such, the vCCA game has two decryption oracles. With the second step oracle assuming the existence of a PPT witness extractor $\mathsf{extract} : \mathcal{C} \times \mathcal{X} \longrightarrow \mathcal{F}_H \times \mathcal{C}^*$, where $\mathcal{X}$ denotes a set of auxiliary data[4]. Before the unique challenge encryption oracle request, the first step decryption oracle is then simply defined as follows:

- Decryption request (1st step): when $\mathcal{A}$ queries $(\texttt{ciphertext}, c)$, return $\mathsf{Dec}(c)$.

Then, after the generation of the unique challenge ciphertext c^*:

- Decryption request (2nd step): when $\mathcal{A}$ queries $(\texttt{ciphertext}, c)$ proceed as follows. Let $(f, c_0, ..., c_{L-1}) = \mathsf{extract}(c, \mathsf{aux})$. Then, return $\perp$ when $c^* \in \{c_0, ..., c_{L-1}\}$ and $\mathsf{Dec}(c)$ otherwise.

The vCCA game has no evaluation oracle as the adversary performs the homomorphic evaluations on its own in both the private and public key settings. In essence, 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). In [47], vCCA security is defined and studied under the correctness assumption of the underlying FHE scheme, and then further studied in [21] when that assumption is not satisfied. Then, [21] also defines the notion of vCCA^D security, which, in a nutshell, is a "CPA^D-style" multiple challenge variant of vCCA in which the decryption oracle also accepts byproducts of the challenge ciphertexts as long as the associated left and right cleartext evaluations coincide. More precisely, *in the private key setting*[5], the vCCA^D game decryption oracle is defined as:

- Decryption request: when $\mathcal{A}$ queries $(\texttt{ciphertext}, c)$ proceed as follows. Let $(f, c_0, ..., c_{L-1}) = \mathsf{extract}(c)$. If,

$$f(\mathrm{left}(c_0), ..., \mathrm{left}(c_{L-1})) \neq f(\mathrm{right}(c_0), ..., \mathrm{right}(c_{L-1})), \tag{4}$$

  then return $\perp$ to $\mathcal{A}$. Otherwise, return $\mathsf{Dec}(c)$.

[4] 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 forming the trace of the execution of the adversary, see [47,21] for more details.

[5] In the public key setting, the definition of vCCA^D security further has to assume the existence of a plaintext extractor. We refer the reader to [21] (Sect. 3.3.1) regarding this minor technical subtlety.

Where the vCCA^D challenger maintains an internal state S (similar to that of a CPA^D challenger) and where for any ciphertext $c \in \mathcal{C}$ we define[6]

$$\text{left}(c) = \begin{cases} S[i].m_0 & \text{if } \exists i : S[i].c = c, \\ \perp & \text{otherwise,} \end{cases} \tag{5}$$

as well as,

$$\text{right}(c) = \begin{cases} S[i].m_1 & \text{if } \exists i : S[i].c = c, \\ \perp & \text{otherwise,} \end{cases} \tag{6}$$

Finally, following [21], vCCA and vCCA^D security are related as follows. *Under the correctness assumption*, the two notions are equivalent. In the general regime where approximate or somewhat correct FHE are allowed, we have that

$$\text{vCCA} \prec \text{vCCA}^D_{\text{SC}} \prec \text{vCCA}^D,$$

where vCCA$^D_{\text{SC}}$ is the single-challenge variant of vCCA^D. Additionally, still in that latter regime, vCCA security remains equivalent to its multiple-challenge variant. In terms of which notion should be targeted, the bottom line is then to target (single-challenge) vCCA security when working under the correctness assumption (and it will be our focus in Sect. 3) and to target (multiple-challenge) vCCA^D security when working in the somewhat correct setting (in Sect. 4).

Lastly, like CPA^D security, vCCA^D security also admits a non-adaptive variant where the adversary specifies all its requests straight after seeing the public material of the scheme. We will sometimes refer to this variant as vCCA$^D_0$. Although this variant may not appear too restrictive in the public-key setting, where the adversary can generate ciphertexts on its own and also perform homomorphic evaluations (or any other treatments) over them before specifying its set of requests, it is much more restrictive in the private key setting. Indeed, in that case, the adversary has to specify all of its decryption requests before seeing any ciphertext and therefore can ask for the decryption only of a priori chosen ciphertexts. In the sequel, *we refer to this rather weak non-adaptive CCA security notion as CCA0*. In particular, we have CCA0 $\prec$ CCA1. For completeness (and sanity checking), we prove this separation in appendix Sect. E.3. With respect to CPA^D security, we have CPA$^D_0$ $\prec$ CCA0 (see also Sect. E.3).

2.4 Smudging

Smudging is a technique that consist in "hiding" a small noise by flooding it in a much larger noise such that the effect of the small noise becomes negligible. Smudging was first introduced in [7] in the context of threshold PKE and later, in the context of threshold FHE in [3,14] (essentially to make sure that a decryption oracle over well-formed ciphertexts can be simulated without actually decrypting

[6] With the convention that $f(m_0, ..., m_{L-1}) = \perp$ when $\exists i : m_i = \perp$ so that if the left and right evaluations both give $\perp$, condition (4) is not satisfied and $\mathsf{Dec}(c)$ is returned to $\mathcal{A}$.

in the relevant security reductions) and has been used, since then in several constructions proposals, e.g. [49]. Beyond threshold FHE, smudging has also been suggested as a countermeasure to CPA^D attacks against CKKS [40] (where it is referred to as noise flooding) or the other "exact" FHE [23]. Smudging comes into different flavors depending on whether the statistical distance or the Rényi divergence is considered [27,16,48] or whether worst-case/non-worst-case smudging should be performed [8,37].

In this work, for simplicity sake, we consider *worst-case* smudging based on the *statistical distance* (as such we do not claim that this simple approach is optimal and more advanced approaches, e.g. [52], may yield smaller smudging noise bounds or variances, eventually leading to smaller LWE parameters). More specifically, we rely only on simple "Smudging Lemmas" such as the following from [3] (Lemma 1 in that paper), which we reproduce below.

Lemma 1 (Smudging Lemma [3]). *Let B_0 and B_1 be two positive integers and let $e_0 \in [\![-B_0, B_0]\!]$ be a fixed integer. Let e_1 be chosen uniformly at random in $[\![-B_1, B_1]\!]$. Then, if $B_1 \geq 2^\lambda B_0$ the statistical distance between the distribution of e_1 and that of $e_0 + e_1$, $d(f_{e_1}, f_{e_1+e_0})$, is bounded by $neg(\lambda)$.*

The above Lemma is useful as it shows that a centered uniform noise with an appropriately large support can "smudge out" a constant value and, more generally, any random variable following a distribution with a bounded support. We can further extend it in order to "smudge out" a Gaussian noise, as we do just below.

Lemma 2. *Let ε be a centered Gaussian random variable with variance σ_0^2. Further let $B_0 = \sigma_0\sqrt{2(\lambda+1)\log 2}$, then $P(\varepsilon \notin [-B_0, B_0]) \leq 2^{-\lambda}$.*

Proof. Recall that the Chernoff bound for the (centered) Gaussian distribution[7] tells that $P(|\varepsilon| \geq B) \leq 2e^{-\frac{B^2}{2\sigma_0^2}}$. Then, consider B_0 such that $2e^{-\frac{B_0^2}{2\sigma_0^2}} = 2^{-\lambda}$ i.e. $B_0 = \sigma_0\sqrt{2(\lambda+1)\log 2}$. □

If we now choose

$$B_1 = 2^\lambda B_0 = 2^\lambda \sigma_0 \sqrt{2(\lambda+1)\log 2} \tag{7}$$

then Lemma 1 applies, directly leading the following Lemma.

Lemma 3. *Let ε be a centered Gaussian random variable with variance σ_0^2 and let $B_1 \geq 2^\lambda \sigma_0 \sqrt{2(\lambda+1)\log 2}$, then $d(f_v, f_{v+\varepsilon}) \leq 2^{-\lambda}$, where v is picked uniformly in $[\![-B_1, B_1]\!]$.*

[7] Remark that the Chernoff bound for the continuous Gaussian distribution also applies to the discrete Gaussian distribution for large enough LWE modulus q. Indeed, for a Gaussian deviates X, $P(|X| \geq a) = P(|\lceil X \rfloor| \geq \lceil a \rfloor)$ (ignoring the mod q as long as $q >> \sigma_0\sqrt{2\lambda}$, an assumption that will always be implicitly satisfied in this work).

Alternatively, a small (centered) Gaussian noise can also be smudged out by a Gaussian noise of much larger variance. This is implied by the following Lemma which proof is given in appendix Sect. J.1 for completeness (other distributions may further be used for smudging, e.g. Sec. J.2).

Lemma 4. *Let ε and X be centered Gaussian random variables with respective variance σ_0^2 and σ_1^2, with $\sigma_1^2 = \frac{2^{2\lambda}\sigma_0^2(\lambda+1)\log 2}{\pi}$, then $d(f_X, f_{X+\varepsilon}) \leq neg(\lambda)$.*

Please note that, for simplicity sake, *we stated the results in this section for the continuous rather than the discrete Gaussian distribution.* However, as this paper focuses only on linearly homomorphic encryption, it will be clear that all the noises occurring in the lattice-based schemes we consider are either (discrete) Gaussian deviates or linear combinations of independent (discrete) Gaussian deviates. Under these circumstances, bounds derived for continuous Gaussian deviates are also valid for discrete ones, following Theorem 9 in [13] (Theorem 4.13 in [12]) stating that the distribution obtained by linearly combining independent discrete Gaussian random variables is itself statistically indistinguishable from a discrete Gaussian distribution.

2.5 The Linear-Only Homomorphism (LOH) assumption

Informally, for an encryption scheme $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{ImVer}, \mathsf{Dec}, \mathsf{Eval})$, the Linear-Only Homomorphism (LOH) property states (as explained in [11]) 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 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 [11] to serve as the basis for several SNARK constructions in that paper and other subsequent works [15,35,50].

Formally, following [11], we have the following definition.

Definition 5 (LOH property, reproduced from [11]). *An encryption scheme $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{ImVer}, \mathsf{Dec}, \mathsf{Eval})$ (with $\mathcal{P} = \mathbb{Z}_t$[8]) satisfies the* Linear-only Homomorphism property *if for every PPT adversary $\mathcal{A}$, there is a PPT extractor $\mathsf{extract}$ such that for any auxilliary input $\mathsf{aux} \in \{0,1\}^{poly(\lambda)}$ and any plaintext generator $\mathcal{M}$,*

$$P\left(\begin{array}{l|l} \exists i \in [\![0, k-1]\!] \text{ s. t.} & (\mathsf{ek}, \mathsf{sk}) := \mathcal{E}_H.\mathsf{KeyGen}(1^\lambda) \\ \mathsf{ImVer}(c'_i) = \mathsf{True} & (a_0, ..., a_{m-1}) := \mathcal{M}([\mathsf{ek}]) \\ \text{and} & (c_0, ..., c_{m-1}) := (\mathcal{E}_H.\mathsf{Enc}(a_0), ..., \mathcal{E}_H.\mathsf{Enc}(a_{m-1})) \\ \mathcal{E}_H.\mathsf{Dec}(c'_i) \neq a'_i & (c'_0, ..., c'_{k-1}) := \mathcal{A}(c_0, ..., c_{m-1}, [\mathsf{ek}]; \mathsf{aux}) \\ & \Pi := \mathsf{extract}(c_0, ..., c_{m-1}, [\mathsf{ek}]; \mathsf{aux}) \\ & (a'_0, ..., a'_{k-1})^T := \Pi \cdot (a_0, ..., a_{m-1})^T + b \end{array}\right) \leq neg(\lambda). \tag{8}$$

[8] The definition still extends to the case where the plaintext domain is a ring [11].

where $\Pi \in \mathbb{Z}_t^{k\times m}$ and $b \in Z_t^k$, and with the convention that row i of Π 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. The notation $[\mathsf{ek}]$ indicates that the encryption key ek is optionally provided, depending on whether the setting is private or public key.

In summary, whenever $\mathcal{A}$ builds c'_i by doing "something equivalent to",

$$c'_i = \mathcal{E}_H.\mathsf{Add}(\mathcal{E}_H.\mathsf{Eval}(\mathtt{lincomb}_\pi, c_0, ..., c_{m-1}), \underbrace{\mathcal{E}_H.\mathsf{Eval}(\mathtt{lincomb}_{\pi'}, c''_0, ..., c''_{l-1})}_{\text{pub. key case only}}), \tag{9}$$

where $c''_0 = \mathcal{E}_H.\mathsf{Enc}(\mu_0; \mathsf{pk}), ..., c''_{l-1} = \mathcal{E}_H.\mathsf{Enc}(\mu_{l-1}; \mathsf{pk})$, 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,

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

as a shortcut for $\Pi_{i:c'_i=c}$ and $b_{i:c'_i=c}$ (i.e. π, respectively β, is the row of Π, respectively the component of β, associated to ciphertext c). 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 (a case that does not happen in the private key setting) and homomorphically add them to homomorphic evaluations of linear combinations over the c_i's. When operating in the private key setting, we will thus omit β and simply use the notation,

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

On top of the above definition, [11] 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 (e.g. [11,34]), 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 [29]. To illustrate this approach with the Paillier scheme, one may consider that the encryption of a message m consists of a pair of ciphertexts $(\mathsf{Enc}(m), \mathsf{Enc}(\alpha \cdot m))$ under the same key material, for a secret random value $\alpha \in \mathbb{Z}_n$ (with n the RSA modulus of the scheme), and with the 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). We will further explore the security properties of this construction in Sect. 3.2.

In the present paper (Sect. 4), we also consider LWE-based candidate schemes. However, when considering lattice-based constructions following the above "two-ciphertexts" blueprint, some additional precautions need to be taken. First, as already noted in [11], it is well-known that LWE or RLWE schemes can be turned

into Fully Homomorphic Schemes. So if one proceeds directly as above by encrypting a message m as a pair of independent ciphertexts $(\mathsf{Enc}(m), \mathsf{Enc}(\alpha \cdot m))$ under the same key, an adversary may eventually compute two ciphertexts $\mathsf{Enc}(f(m))$ and $\mathsf{Enc}(\alpha \cdot f(m))$, for some nonlinear function f, from an encryption of m and an encryption of 1 (giving the pair $(\mathsf{Enc}(1), \mathsf{Enc}(\alpha))$). In practice, this issue can be dealt with by having the two (or more, see bellow) ciphertexts be under *independent* keys, so that one cannot (homomorphically) obtain a consistent encryption of $\alpha \cdot f(m)$ from an encryption of m (say under $\mathsf{sk}^{(0)}$) and an encryption of α (say under $\mathsf{sk}^{(1)}$). Furthermore, all known FHE constructions require issuance of evaluation keys such as relinearization or bootstrapping keys. When no such keys are provided, these schemes are (to the best of the community's knowledge) stuck with linear-only homomorphic properties. The second pitfall, is that such schemes tend to use a plaintext modulus t of small size. Then, given an arbitrary first ciphertext in a "two-ciphertexts" pair, the adversary can randomly sample the second ciphertext and then succeed with probability $\frac{1}{t}$ to forge a valid ciphertext pair and thus violate the LOH property. This difficulty can easily be worked around either by choosing $t = O(2^\lambda)$ (but this option would result in very large ciphertext modulus) or to follow a "more-than-two-ciphertexts" blueprint using $1 + \lceil \lambda / \log_2 t \rceil$ ciphertexts rather than just 2 in order to ensure that an adversary "obliviously sampling" valid ciphertexts without knowing the corresponding plaintext succeeds only with $\mathrm{neg}(\lambda)$ probability. The last and not least pitfall is that LWE-based schemes are prone to decryption errors. Although the occurrence of such errors does not necessarily contradict the LOH assumption (for example if ImVer consistently return False when such an error occurs), credibly assuming the LOH property for LWE-based schemes lead to delicate issues to which a large part of this paper is devoted (Sect. 4). Note that prior works (e.g. [15,35,50]) essentially workaround this issue by assuming a weaker LOH property in which the adversary is restricted to evaluate linear combinations of bounded L_2 norm and choosing the LWE parameters such that (statistical) correctness is achieved under that bound constraint[9].

As a last remark, let us also emphasize that the LOH property requires $\mathcal{E}_H$'s decryption function to be *deterministic*, at least with overwhelming probability.

3 Results under the correctness assumption

In this section, we investigate the relationship between the LOH property and the vCCA security notion for *correct* linearly homomorphic encryption schemes. Because, for vCCA security, the single challenge notion is equivalent to the multiple challenge one [21], we focus our proofs only on the former notion.

The results in this section work under the natural assumption that

$$\mathcal{E}.\mathsf{Dec}(c) = \perp \Leftrightarrow \mathcal{E}.\mathsf{ImVer}(c) = \mathsf{False}, \tag{10}$$

[9] These works essentially use such LWE-based schemes as building blocks in the context of proof-of-knowledge constructions in which the L_2 bound constraint ends up satisfied "by construction".

and the assumption stating that for $(\mathsf{ek}, \mathsf{sk}) \in \mathcal{K}$, all $\pi \in \mathcal{P}^L$, all $m_1, ..., m_L \in \mathcal{P}^L$,

$$\Pr_{\vec{r} \in \mathsf{COIN}^L}(\mathsf{ImVer}(\mathsf{Eval}(\mathtt{lincomb}_\pi, \mathsf{Enc}(m_1; r_1), ..., \mathsf{Enc}(m_L; r_L))) = \mathsf{False}) \leq \mathrm{neg}(\lambda). \tag{11}$$

This latter assumption is natural under the correctness assumption (Eq. 2) and is consistent with the expected functionalities of a linear homomorphic scheme i.e. since Dec returns $\perp$ when $\mathsf{ImVer} = \mathsf{False}$, a scheme *not* satisfying this property is essentially *not* linearly homomorphic.

3.1 General black-box results

We first focus on the private key setting.

Proposition 6. *Let $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{ImVer}, \mathsf{Eval})$ be a private key correct CPA secure linearly homomorphic scheme that satisfies the LOH property. Then $\mathcal{E}_H$ is vCCA secure.*

Proof. We start by one step of game hoping.

First game hop. Let G_0 be the vCCA game against $\mathcal{E}_H$ and G_1 be the same game as G_0 where we modify the challenger such that, when handling a decryption request on ciphertext ct, the new challenger invokes the LOH extractor to verify that

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

rather than checking $\mathsf{ImVer}(\mathsf{ct}) = \mathsf{True}$. Indeed, the two games cannot be distinguished since the case where $(\mathsf{extract}(c) = \varnothing) \wedge (\mathsf{ImVer}(c) = \mathsf{True})$ contradicts the LOH property in conjunction with property (10); and the case where $(\mathsf{extract}(c) \neq \varnothing) \wedge (\mathsf{ImVer}(c) = \mathsf{False})$ contradicts property (11). Hence, $\mathsf{extract}(c) = \varnothing \Leftrightarrow \mathsf{ImVer}(c) = \mathsf{False}$.

Final reduction. To finalize the proof, we show that, from an adversary $\mathcal{A}$ against G_1 (or equivalently G_0), we can build an adversary $\mathcal{B}$ against the CPA security of $\mathcal{E}_H$ which uses $\mathcal{A}$ as a subroutine. For the reduction to work, we assume that $\mathcal{A}$ and $\mathcal{B}$ agree on a consistent numbering of the ciphertexts output by the encryption oracle. The reduction then starts by initializing an empty state $S := [\,]$ which will contain message-ciphertext pairs and then proceeds as follows.

- When receiving an encryption request over message $m \in \mathcal{P}$ from $\mathcal{A}$, it first transfers it as is to the CPA challenger to get ciphertext $\mathsf{ct} = \mathcal{E}_H.\mathsf{Enc}(m; \mathsf{ek})$ (for unknown ek since we are in the private key setting) which it sends back to $\mathcal{A}$ after updating its internal state as $S := [S; (m, \mathsf{ct})]$.
- When receiving the *single* challenge request over messages $m_0 \neq m_1 \in \mathcal{P}^2$ from $\mathcal{A}$, it transfers it as is to the CPA challenger to get ciphertext $\mathsf{ct}^* = \mathcal{E}_H.\mathsf{Enc}(m_\gamma; \mathsf{ek})$ (for unknown ek and γ), which it sends back to $\mathcal{A}$ after updating its internal state as $S := [S; (m_0, \mathsf{ct}^*)]$ (or, equivalently, $S := [S; (m_1, \mathsf{ct}^*)]$) and setting $i^* = |S| - 1$ (i^* is initially set to -1).

– When $\mathcal{A}$ issues a decryption request over ciphertext $\mathsf{ct} \in \mathcal{C}$, then $\mathcal{B}$ runs the LOH extractor to get $\pi = \mathsf{extract}(\mathsf{ct}, \mathsf{aux})$. When $\pi = \varnothing$, it returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ returns $\perp$ whenever $\pi_{i^*} \neq 0$ (with the convention that $\pi_{-1} = 0$) or,

$$\sum_{i:\pi_i \neq 0} \pi_i S[i].m, \tag{12}$$

when this is not the case (i.e. when $\pi_{i^*} = 0$).

The claim follows from the facts that, when $\mathsf{ImVer}(\mathsf{ct}) = \mathsf{True}$ (or, equivalently, when $\mathsf{extract}(\mathsf{ct}, \mathsf{aux}) \neq \varnothing$), the correctness of $\mathcal{E}_H$ implies that (12) cannot be distinguished from $\mathcal{E}_H.\mathsf{Dec}(\mathsf{ct}; \mathsf{sk})$ and that the reduction duly reply $\perp$ for all decryption requests on ciphertexts which are byproducts of the challenge ciphertexts (i.e. for which $c^* \in \{c_0, ..., c_{L-1}\}$ in the notations of the vCCA game 2nd step decryption oracle definition, p. 7, and $\pi_{i^*} \neq 0$), consistently with the vCCA game decryption oracle specification. □

As vCCA security is the strongest (so far known) CCA security notion achievable by FHE *under the correctness assumption*, this is a relatively strong implication. Remark also that we get CCA1 security as a corollary to Prop. 6 since vCCA security implies CCA1 security [47].

Proving a general result in the public key setting is a little bit more subtle.

Proposition 7. *Let $\mathcal{E}_H = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{ImVer}, \mathsf{Eval})$ be a* public key *correct CPA secure linearly homomorphic scheme that satisfies the LOH assumption. Then $\mathcal{E}_H$ is vCCA secure.*

Proof. We first start by showing how to turn a LOH extractor into a plaintext extractor in the public key setting.

Preliminary remarks. In the public key setting remark that there are two types of ciphertexts that the adversary may *not* generate on its own: a (possibly empty) set of ciphertexts $\mathsf{pk}_0, ..., \mathsf{pk}_N$ which are part of the public key as well as the unique challenge ciphertext ct^*. Assume that the vCCA challenger stores them in a state S containing message-ciphertext pairs i.e. such that

$$S = [(m_0, \mathsf{pk}_0), ..., (m_{N-1}, \mathsf{pk}_{N-1}), (m_0^*, \mathsf{ct}^*)] \tag{13}$$

where m_0^* is the first of the two messages that served for creating ct^* (S might equivalently contain (m_1^*, ct^*) in the last position, as it is expected to return $\perp$ for decryption requests over ciphertexts depending on ct^*). Further remark that the LOH extractor works only on ciphertexts generated by the adversary (aux is essentially the trace of execution of $\mathcal{A}$) and *not* over those generated by the challenger. Thus, following Def. 5 (p. 10), we have $m = N + 1$ and the set $(c_0, ..., c_{m-1})$ (in that definition) is reduced to $(\mathsf{pk}_0, ..., \mathsf{pk}_{N-1}, \mathsf{ct}^*)$.

Then, let us consider a ciphertext ct submitted as part of decryption request such that $\mathsf{extract}(\mathsf{ct}, \mathsf{aux}) \neq \varnothing$. Let $(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux})$, we have the two following properties:

– ct depends on ct^* (i.e. $c^* \in \{c_0, ..., c_{L-1}\}$ in the notations of the vCCA game 2nd step decryption oracle definition, p. 7) if, and only if, $\pi_N \neq 0$.

– If ct is independent of ct^* (i.e. $c^* \notin \{c_0, ..., c_{L-1}\}$ in the vCCA game 2nd step decryption oracle as well as $\pi_N = 0$) then, from the correctness of $\mathcal{E}_H$,

$$\mathcal{E}_H.\mathsf{Dec}(\mathsf{ct}) = \sum_{i=0}^{N-1} \pi_i S[i].m + \beta. \tag{14}$$

With these properties, we can now finalize the proof as follows.

First game hop. Identical to that of the proof of Prop. 6

Final reduction. To finalize the proof, we then show that, from an adversary $\mathcal{A}$ against G_0 (or equivalently G_1), we can build an adversary $\mathcal{B}$ against the CPA security of $\mathcal{E}_H$ which uses $\mathcal{A}$ as a subroutine. For the reduction to work, we assume that $\mathcal{A}$ and $\mathcal{B}$ agree on a consistent numbering of the ciphertexts generated by the reduction, which it stores in an internal state S containing message-ciphertext pairs following Eq. (13). Thus, after getting ek (which contains $\mathsf{pk}_0, ..., \mathsf{pk}_{N-1}$) and transferring it to $\mathcal{A}$, the reduction proceeds as follows[10]:

– When receiving the *single* challenge request over messages $m_0^* \neq m_1^* \in \mathcal{P}^2$ from $\mathcal{A}$, it transfers it as is to the CPA challenger to get ciphertext $\mathsf{ct}^* = \mathcal{E}_H.\mathsf{Enc}(m_\gamma^*; \mathsf{ek})$ (for unknown γ), which it sends back to $\mathcal{A}$ after updating its internal state as $S := [S; (m_0^*, \mathsf{ct}^*)]$ (or, equivalently, $S := [S; (m_1^*, \mathsf{ct}^*)]$). *Remark that following Eq. (13), ct^* is stored in the $N+1$-th position in S.*
– When $\mathcal{A}$ issues a decryption request over ciphertext $\mathsf{ct} \in \mathcal{C}$, it first checks that $\mathsf{extract}(\mathsf{ct}, \mathsf{aux}) \neq \varnothing$ and returns $\perp$ when this is not the case. Let $(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux})$, then $\mathcal{B}$ returns $\perp$ whenever $\pi_N \neq 0$ (case of a challenge-dependent ciphertext). Lastly, when this is not the case ($\pi_N = 0$), $\mathcal{B}$ runs the plaintext extractor given by Eq. (14), i.e. simply returns $\sum_{i=0}^{N-1} \pi_i S[i].m + \beta$.

□

3.2 A public-key vCCA secure construction based on Paillier

Let $\mathcal{E}_P = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add}, \mathsf{Mulc})$ denote the Paillier encryption scheme [51] (briefly recalled in Appendix F). We now consider the scheme $\mathcal{E}_P^{(2)}$, built from $\mathcal{E}_P$ as follows[11].

– $\mathcal{E}_P^{(2)}.\mathsf{KeyGen}$: run $\mathcal{E}_P.\mathsf{KeyGen}$ to get n, g and ω, then pick ξ uniformly at random in $\mathbb{Z}_n$. Generate ciphertext $\mathsf{ct}^\triangle = (\mathcal{E}_P.\mathsf{Enc}(1), \mathcal{E}_P.\mathsf{Enc}(\xi))$. The public key is set to $\mathsf{pk} = (n, \mathsf{ct}^\triangle)$, while all the other parameters remain private (including the generator g). As in the original scheme, the secret key is $\mathsf{sk} = \omega(n)$.
– $\mathcal{E}_P^{(2)}.\mathsf{Enc}$: given $m \in \mathbb{Z}_n$ and $\mathsf{pk} = (n, \mathsf{ct}^\triangle)$, pick r_0, r_1 uniformly in $\mathbb{Z}_{n^2}$. When $m = 0$, return

$$\mathsf{ct} = (c_0, c_1) = ([r_0^n]_{n^2}, [r_1^n]_{n^2}). \tag{15}$$

Otherwise, compute and return

$$\mathsf{ct} = (c_0, c_1) = ([(\mathsf{ct}^\triangle.c_0)^m r_0^n]_{n^2}, [(\mathsf{ct}^\triangle.c_1)^m r_1^n]_{n^2}). \tag{16}$$

[10] In the public-key setting, the reduction does not have to handle encryption requests.
[11] Following Sect. 2.5, we emphasize that this scheme is not new and has already been proposed and used in several works, e.g. [11,34]

- $\mathcal{E}_P^{(2)}.\mathsf{ImVer}$: given $\mathsf{ct} \in \mathbb{Z}_{n^2}^2$ and sk, let μ_0 denote $\mathcal{E}_P.\mathsf{Dec}(\mathsf{ct}.c_0;\mathsf{sk})$ (and respectively so for μ_1). Then return True if $[\xi\mu_0]_n = \mu_1$ and False otherwise.
- $\mathcal{E}_P^{(2)}.\mathsf{Dec}$: given $\mathsf{ct} \in \mathbb{Z}_{n^2}^2$ and sk, if $\mathcal{E}_P^{(2)}.\mathsf{ImVer}(\mathsf{ct};\mathsf{sk}) = \mathsf{True}$, then return $\mathcal{E}_P.\mathsf{Dec}(\mathsf{ct}.c_0;\mathsf{sk})$. Otherwise, return $\perp$.
- $\mathcal{E}_P^{(2)}.\mathsf{Add}$ and $\mathcal{E}_P^{(2)}.\mathsf{Mulc}$ are straightforwardly derived from those of $\mathcal{E}_P$.

We highlight that:

- $\mathsf{ct}^\triangle$ is a *privately generated* encryption of 1 under $\mathcal{E}_P^{(2)}$.
- Since g and ξ are *not* public, only encryptions of 0 under $\mathcal{E}_P^{(2)}$ can be generated without using $\mathsf{ct}^\triangle$ by picking r_0, r_1 uniformly in $\mathbb{Z}_{n^2}$ and following Eq. (15) above.
- $\mathcal{E}_P^{(2)}.\mathsf{Enc}$ (i.e. Eq. 16) can equivalently be written as

$$\mathsf{ct} = \mathcal{E}_P^{(2)}.\mathsf{Add}(\mathcal{E}_P^{(2)}.\mathsf{Mulc}_m(\mathsf{ct}^\triangle), \mathcal{E}_P^{(2)}.\mathsf{Enc}(0)). \tag{17}$$

Proposition 8. *$\mathcal{E}_P^{(2)}$ is CPA secure.*

Proof. The proof is done via a simple reduction to the CPA security of $\mathcal{E}_P$. The reduction picks ξ and, since the scheme is public-key, only has to generate a proper challenge ciphertext from the one it obtains from its CPA challenger against $\mathcal{E}_P$, which is denoted by $c^* = \mathcal{E}_P.\mathsf{Enc}(m_\gamma)$ (for unknwon bit γ). This is simply done by picking r uniformly in $\mathbb{Z}_n$ and returning, $\mathsf{ct}^* = (c^*, [c^{*\xi} r^n]_{n^2})$, i.e. $(c^*, [\mathcal{E}_P.\mathsf{Mulc}_\xi(c^*) r^n]_{n^2})$, to the CPA adversary against $\mathcal{E}_P^{(2)}$. □

Lemma 9. *Let $\mathsf{ct} = (c_0, c_1)$ denote a well-formed ciphertext under $\mathcal{E}_P^{(2)}$ encrypting a linear combination $\sum_i \alpha_i m_i$, then there exists r and r' such that*

$$c_0 = (\mathsf{ct}^\triangle.c_0)^{\sum_i \alpha_i m_i} r^n$$

and

$$c_1 = (\mathsf{ct}^\triangle.c_1)^{\sum_i \alpha_i m_i} r'^n$$

Proof. Left to the reader. □

Putting $\mathcal{E}_P^{(2)}$ into perspective with Def. 5, remark that given a well-formed ciphertext ct under $\mathcal{E}_P^{(2)}$ built by some adversary $\mathcal{A}$, the above Lemma (and the perfect correctness of $\mathcal{E}_P$) implies that π_0, with

$$(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux}) \tag{18}$$

gives a plaintext extractor (i.e. $\pi_0 = \mathcal{E}_P^{(2)}.\mathsf{Dec}(\mathsf{ct})$). Furthermore, since the only ciphertexts that an adversary can generate from scratch are encryptions of 0, $\beta = 0$ in Eq. (18).

We now prove that $\mathcal{E}_P^{(2)}$ achieves vCCA security.

Proposition 10. *$\mathcal{E}_P^{(2)}$ is vCCA secure under the assumption that it has the LOH property.*

Proof. This follows from Proposition 6. In the notations of the proof of that Prop., we have $N = 1$ and $\mathsf{pk}_0 = \mathsf{ct}^{\triangle}$. Lemma 9 (and Eq. 18) further gives a plaintext extractor which is consistent with Eq. (14) in that proof. Also note that $\mathcal{E}_P^{(2)}$ satisfies property (10). □

It is interesting that, under the assumption that it has the LOH property, such a simple construction eventually achieves vCCA security. As a byproduct, the previous proposition also implies that $\mathcal{E}_P^{(2)}$ achieves CCA1 security. Further note that the recent theoretical barriers revealed by Schäge [55] against the provable CCA1 security of vanilla ElGamal and Paillier[12] (as well as alike semi-homomorphic PKE schemes) do not apply here. Indeed, we establish CCA1 security for a "two-ciphertexts" variant of Paillier which is not covered by these results (essentially as this construction lacks the required property that the validity of ciphertexts can be publicly verified). The same remark applies to the modified Paillier scheme, where λ plaintext bits are forced to be zeroes, that is shown CCA1 secure in [41]. Consistently with these remarks, vanilla Paillier further does not have the LOH property as also argued in [11] ([10], p. 33).

4 LWE-based constructions

We now give up the comfort of working under the correctness assumption. In this "jungle", we have decided to straightaway focus on concrete candidate LWE-based constructions. Still, a natural question is whether black-box results, such as Prop. 6 and 7, can be obtained in the case of approximate schemes? I.e. may any CPA^D secure linearly homomorphic scheme with the LOH property be vCCA^D secure? We sketch such results in Sect. C. However, as we unveil in this section, building approximate or somewhat correct (LWE-based) schemes credibly satisfying even weak variants of the LOH property is particularly delicate. So it seems to us that this kind of black-box results have a limited relevance.

The Knowledge-of-Exponent pattern is also natural to apply to LWE-based schemes: start from the multi-secret variant of Regev, put the message in the first slot and multiples of that message in the subsequent ones, for a large-enough set of secret multipliers. However, in the LWE setting, we have to deal with CPA^D security[13]. There are two ways to do so, on the one hand, we can put additional restrictions on the adversary abilities, via the cryptosystem specification which a CPA^D adversary is bounded to follow, so as to achieve correctness [1]. However, in the CCA adversary regime, compliance with these constraints (e.g. a bound on the L_2 norm of the linear combinations that the adversary can evaluate over fresh ciphertexts) has to be enforced and this appears difficult to achieve without advanced proof-of-knowledge techniques. On the other hand, if we wish to avoid such additional constraints, we have to embed *within the cryptosystem* some

[12] Despite of the fact that CCA1 security proofs do exist either under non-standard assumptions [2] or in the idealized Generic as well as Algebraic Group Models [43,33].

[13] At the very least, a simple adaptation of the attack path in [23] allows to retrieve the L_∞ norm of the noise vector in such a ciphertext.

mechanism (usually some form of smudging) that eventually allows to handle CPA^D decryption requests in a reduction towards a CPA challenger. This is the path we follow in this section, by designing a line of schemes consistently following the Knowledge-of-Exponent pattern and achieving some degree of CPA^D security by means of smudging. We then investigate how far this approach takes us in terms of CCA security.

4.1 The basic Regev Scheme (FS_0)

We start from the usual Regev scheme. We consider the symmetric variant, which is parameterized by a security parameter λ, a dimension n, an integer q and a (discrete Gaussian) probability distribution χ_{σ_0} on $\mathbb{Z}_q$ with standard deviation σ_0. Plaintexts are elements of $\mathbb{Z}_t$ and ciphertexts are elements of $\mathbb{Z}_q^n \times \mathbb{Z}_q$. *Unless otherwise stated, we work under the assumption that t divides q.* The scheme $\mathsf{FS}_0(\lambda, n, q, t, \sigma_0)$ is then defined as follows:

- $\mathsf{FS}_0.\mathsf{KeyGen}$: pick a secret key $\mathsf{sk} \in \mathbb{Z}_q^n$ uniformly at random.
- $\mathsf{FS}_0.\mathsf{Enc}$: given a plaintext $m \in \mathbb{Z}_t$, pick $a \in \mathbb{Z}_q^n$ uniformly at random, pick e in $\mathbb{Z}_q$ according to χ_{σ_0}, and return (a, b) with $b = \langle a, \mathsf{sk} \rangle + \Delta m + e$ and $\Delta = q/t$.
- $\mathsf{FS}_0.\mathsf{Dec}$: given a ciphertext $c = (a, b)$, return $\left[\left\lceil \frac{1}{\Delta}(b - \langle a, \mathsf{sk} \rangle) \right\rfloor\right]_t$.

This scheme is trivially linearly homomorphic, offering homomorphic addition, mult-by-const ($\mathtt{mul}_\alpha$, $\alpha \in \mathbb{Z}_t$) and add-by-const ($\mathtt{add}_\alpha$) operators.

Regarding the security notions of interest to us in this paper, it is well-known that FS_0 is CPA secure under the LWE assumption [53]. However, FS_0 is CPA^D insecure [23,24] and also trivially CCA1 insecure. For example, the ill-formed ciphertext $(-\Delta \mathbf{1}_i, 0)$ decrypts to $[\mathsf{sk}_i]_t$, where $\mathbf{1}_i$ is the i-th standard basis vector, which is enough to recover the secret key with n decryption requests when its components are in $\mathbb{Z}_t$. When, as above, sk is picked uniformly in $\mathbb{Z}_q$ it is also easy for a CCA1 adversary to retrieve sk by means of this kind of decryption requests (we describe such an attack in Sect. G.1 for completeness).

4.2 Achieving (non-adaptive) CPA^D security (FS_1)

Preliminaries. The intuition behind our first construction is as follows: we append a clean noise sample of size K to each ciphertext and we use it as a (linearly-updatable) variance monitoring mechanism. To do so, we consider a multi-secret variant of FS_0 with $K+1$ slots, with the first slot carrying a message and the K others being vanilla LWE samples carrying only noise. Each slot is under key material *independent* of that of the others with *the same a vector* used for all slots. Upon decryption, we then use the noises that we recover in the K last slots to put a confidence interval with $1-2^{-\lambda}$ probability on the noise variance (we are able to do that with a Chernoff bound for the χ^2 distribution) and use this safe bound to generate an appropriate smudging noise at decryption. We do so by means of a result from [36] which tells the following about the lower tail of the χ^2 distribution.

Theorem 11 ([36], Theorem 2.). *Let X follows χ^2_K, then for $0 < c < K$,*

$$P(X < K - c) \leq e^{-\frac{c^2}{4K}}.$$

Lemma 12. *Let $K \geq 4\lambda \log 2 + 1$, given a set $E_0, ..., E_{K-1}$ of iid centered gaussian deviates with variance σ^2, then ,*

$$\sigma^2 \leq \frac{K\hat{\sigma}^2}{K - 2\sqrt{K\lambda \log 2}}$$

with overwhelming probability, where $\hat{\sigma}^2 = \frac{1}{K}\sum_{i=0}^{K-1} E_i^2$.

Proof. Under the statement assumption, $\frac{K\hat{\sigma}^2}{\sigma^2}$ follows χ^2_K. Theorem 11 then tells us that

$$P\left(\frac{K\hat{\sigma}^2}{\sigma^2} < K - c\right) = P\left(\sigma^2 > \frac{K\hat{\sigma}^2}{K - c}\right) \leq e^{-\frac{c^2}{4K}}.$$

Thus, letting $e^{-\frac{c^2}{4K}} = 2^{-\lambda}$ yields $c = 2\sqrt{K\lambda \log 2}$. It therefore follows (when $c < K$ so when $2\sqrt{K\lambda \log 2} < K$, i.e. $K \geq 4\lambda \log 2 + 1$) that

$$P\left(\sigma^2 > \frac{K\hat{\sigma}^2}{K - 2\sqrt{K\lambda \log 2}}\right) \leq 2^{-\lambda}.$$

□

Following this Lemma, we can thus use

$$\bar{\sigma}^2_K = \frac{K\hat{\sigma}^2}{K - 2\sqrt{K\lambda \log 2}} \tag{19}$$

as a safe upper bound for σ^2 given the sample $E_0, ..., E_{K-1}$ (for $K \geq 4\lambda \log 2 + 1$).

Scheme $\mathsf{FS_1}$. Let $K \geq 4\lambda \log 2 + 1$, the scheme $\mathsf{FS}_1(\lambda, n, q, t, \sigma_0, K)$ is then defined as follows:

- $\mathsf{FS}_1.\mathsf{KeyGen}$: for $k \in [\![0, K]\!]$, uniformly pick $\mathsf{sk}^{(k)} \in \mathbb{Z}_q^n$.
- $\mathsf{FS}_1.\mathsf{Enc}$: given a plaintext $m \in \mathbb{Z}_t$, uniformly pick a *single* $a \in \mathbb{Z}_q^n$ as well as vector $E \in \mathbb{Z}_q^{K+1}$ with each component drawn independently from χ_{σ_0}. Then, return $(a, B) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1}$ such that

  $$B_0 = \langle a, \mathsf{sk}^{(0)} \rangle + \Delta m + E_0.$$

  and, for $k \in [\![1, K]\!]$,

  $$B_k = \langle a, \mathsf{sk}^{(k)} \rangle + E_k.$$

  Note that we will sometimes refer to B_0 as the *payload slot* and to the B_k's ($k \in [\![1, K]\!]$) as the *noise slots*.

– $\mathsf{FS}_1.\mathsf{Dec}$: given ciphertext $c = (a, B) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1}$, first compute

$$\hat{\sigma}^2 = \frac{1}{K} \sum_{k=1}^{K} \left(B_k - \langle a, \mathsf{sk}^{(k)} \rangle \right)^2 .$$

Then (following Eq. 19 and Lemma 3), uniformly pick $v \in [\![-B, B]\!]$ with $B = 2^\lambda \bar{\sigma}_K \sqrt{2(\lambda+1)\log 2}$. If $|v| \geq \frac{\Delta}{2}$, return $\perp$. Else, return $\mathsf{FS}_0.\mathsf{Dec}(a, B_0; \mathsf{sk}^{(0)})$.

It is well-known that such multi-secret variants of the Regev scheme are CPA secure. The linearly homomorphic operators of FS_1 (homomorphic addition and mult-by-const) are trivially defined from those of FS_0. We emphasize that *we do not provide FS_1 with a direct add-by-const operator*. Although such an operator could be obtained by simply adding the said constant to B_0.

$\mathbf{CPA_0^D}$ security of $\mathbf{FS_1}$. To prove the CPA^D security of FS_1, we proceed via a reduction to the CPA security of FS_0. In a nutshell, the reduction operates the K noise slots, while the CPA challenger against FS_0 operates the first slot with a message payload. For processing decryption requests (over state indices in CPA^D), the reduction uses the noises it recovers from the noise slots to generate a smudging noise of large-enough variance (with overwhelming probability) to make the effect the noise in the first slot (which it does not know) negligible and, hence, provide outputs that are indinstinguishable from those of a true CPA^D decryption oracle. *However, the reduction is valid only in the* non-adaptive *setting (i.e. CPA_0^D, as defined towards the end of Sect. 2.3) where the adversary specifies all its queries in advance, straight after the key generation step* [38]. As we shall see in the proof of Proposition 15, this restriction is necessary to maintain the independence of the noises which are retrieved from the noise slots, an assumption which we require for proper smudging (essentially in order to be able to apply Lemma 12 to obtain a safe bound on the ciphertext noise variance).

Let us first consider the following Lemma which essentially states that FS_1's decryption function guarantees correct decryption when it does not return $\perp$.

Lemma 13. *Let $(a, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ be such that $b = \langle a, \mathsf{sk} \rangle + \Delta m + e$ where e is a Gaussian deviate of variance σ^2. Let v be drawn uniformly in $[\![-B, B]\!]$, independently of e, with $B \geq 2^\lambda \bar{\sigma} \sqrt{2(\lambda+1)\log 2}$ and $\bar{\sigma}^2 \geq \sigma^2$. Then, if $|v| < \frac{\Delta}{2}$, with overwhelming probability,*

$$\left\lceil \frac{1}{\Delta} (b - \langle a, \mathsf{sk} \rangle) \right\rfloor = m.$$

Proof. In the conditions of the statement, Lemma 3 applies and $e + v \overset{i}{=} v$. Hence, when $|v| < \frac{\Delta}{2}$, $|e| << \frac{\Delta}{2}$ (with overwhelming probability). The claim follows. □

Then the following Lemma captures the independence of the noises (in both the payload and noise slots) within an evaluated FS_1 ciphertext, as long as the coefficients of the linear combination are independent of the ciphertexts on which it has been applied to produce the said ciphertext.

Lemma 14. *Let $\pi \in \mathbb{Z}_t^L$, let $c^{(0)}, ..., c^{(L-1)}$ be any set of L fresh well-formed FS_1 ciphertexts. Further let*

$$c = (a, B) = \mathsf{FS}_1.\mathsf{Eval}(\texttt{lincomb}_\pi, c^{(0)}, ..., c^{(L-1)}),$$

then the $E_k = B_k - \langle a, \mathsf{sk}^{(k)} \rangle$'s ($k \in [\![0, K]\!]$) are independent (and Gaussian).

Proof. The claim follows trivially from the facts that fresh well-formed ciphertexts under FS_1 have this property, that FS_1 homomorphic operators proceed component-wise as well as that π is a priori chosen, *independently* of the $c^{(i)}$'s. □

Following this, we now establish the CPA_0^D security of FS_1, i.e. its CPA^D security in the non-adaptive setting.

Proposition 15. *Let $K \geq 4\lambda \log 2 + 1$, if there exists an adversary $\mathcal{A}$ against the CPA_0^D security of $\mathsf{FS}_1(\lambda, n, q, t, \sigma_0, K)$, then there exists an adversary $\mathcal{B}$ against the (LOR-)CPA[14] security of $\mathsf{FS}_0(\lambda, n, q, t, \sigma_0)$.*

Proof. Recall that the CPA^D game allows multiple challenge requests. So $\mathcal{B}$ starts by initializing an initially empty state $S := []$ which will contain message-message-ciphertext triplets and by uniformly picking $\mathsf{sk}^{(k)} \in \mathbb{Z}_q^n$, for $k \in [\![1, K]\!]$ (i.e. $\mathsf{sk}^{(0)}$ is in the CPA challenger against FS_0 and the other keys are in the reduction). Then, *since we are in the non-adaptive setting*, $\mathcal{A}$ sends all it requests to the reduction which processes them as follows:

- When processing an *encryption request* for message m, $\mathcal{B}$ first transfers it as is to the CPA challenger getting $(a, b = \langle a, \mathsf{sk}^{(0)} \rangle + \Delta m + e)$, for unknown $\mathsf{sk}^{(0)}$ and e. It then picks $E \in \mathbb{Z}_q^K$ following χ_{σ_0} and constructs (a, B) such that $B_0 = b$ and, for $k \in [\![1, K]\!]$,

$$B_k = \langle a, \mathsf{sk}^{(k)} \rangle + E_{k-1}.$$

  After adding it to its internal state by doing $S := [S; (m, m, (a, B))]$, $\mathcal{B}$ returns the ciphertext (a, B) to $\mathcal{A}$.
- When processing a *challenge request* for messages $m_0 \neq m_1$, $\mathcal{B}$ also first transfers it as is to the CPA challenger getting $(a, b = \langle a, \mathsf{sk}^{(0)} \rangle + \Delta m_\gamma + e)$, for unknown $\mathsf{sk}^{(0)}$, challenge bit γ and noise e. It then proceeds as for encryption requests above, however updating its internal state as $S := [S; (m_0, m_1, (a, B))]$.
- When processing an *evaluation request* (wlog broken-down in unitary $\texttt{sum}$ and $\texttt{mul}_\alpha$ homomorphic operations),

[14] Where (LOR-)CPA is the mutiple challenge variant of (IND-)CPA, or (FTG-)CPA in the terminology of [5,6] which show the equivalence between the two notions.

- $(\texttt{eval}, \texttt{sum}, i, j)$: $\mathcal{B}$ creates a new *evaluated* ciphertext

$$c = \mathsf{FS}_1.\mathsf{Eval}(\texttt{sum}, S[i].c, S[j].c),$$

and returns c to $\mathcal{A}$ after updating its internal state as

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

- $(\texttt{eval}, \texttt{mul}_\alpha, i)$: $\mathcal{B}$ similarly creates a new *evaluated* ciphertext,

$$c = \mathsf{FS}_1.\mathsf{Eval}(\texttt{mul}_\alpha, S[i].c),$$

and returns c to $\mathcal{A}$ after updating its internal state as

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

– Lastly, for processing a decryption request with state index i, $\mathcal{B}$ returns $\perp$ if $S[i].m_0 \neq S[i].m_1$. Otherwise, i.e. when $S[i].m_0 = S[i].m_1$, it first computes

$$\hat{\sigma}^2 = \frac{1}{K} \sum_{k=1}^{K} \left(B_k - \langle a, \mathsf{sk}^{(k)} \rangle \right)^2. \tag{20}$$

Then, it uniformly picks $v \in [\![-B, B]\!]$ with $B = 2^\lambda \bar{\sigma}_K \sqrt{2(\lambda+1)\log 2}$. Finally, it returns $\perp$ if $|v| \geq \frac{\Delta}{2}$, and $S[i].m_0$ (or, equivalently, $S[i].m_1$) otherwise.

The key point which makes the reduction works is that $\mathcal{B}$'s replies to $\mathcal{A}$'s decryption requests are indistinguishable from a true FS_1 decryption oracle. Let e be the (unknown to $\mathcal{B}$) noise in the first payload slot of $S[i].c$ and σ^2 denotes its variance. Since, we are in the non-adaptive setting, Lemma 14 applies (i.e. the $B_k - \langle a, \mathsf{sk}^{(k)} \rangle$'s in Eq. 20 are iid). Therefore we can use Lemma 12 (and Eq. 19) to claim that $\bar{\sigma}_K^2 \geq \sigma^2$ (with overwhelming probability). Then, since $\bar{\sigma}_K^2$ is independent from e, Lemma 13 applies. It thus follows that, in the reduction's processing of decryption requests, $\mathsf{FS}_0.\mathsf{Dec}(S[i].c.a, S[i].c.B_0) = S[i].m_0$ (or, equivalently, $S[i].m_1$) whenever $|v| < \frac{\Delta}{2}$ (with overwhelming probability).

□

Remark that, stricto sensu, the noises in the noise slots of an FS_1 ciphertext could even be given in cleartext form since, *in the non-adaptive private key setting* under which the above proof operates, the adversary eventually specifies all of her request before seeing any ciphertexts. However, in our next construction, built on FS_1, we will also use the noise slots to store additional private information. Hence, it will not be desirable to keep the content of the noise slots in cleartext.

Because it is using a ciphertext-dependent variance estimation procedure, FS_1 falls in a category of schemes, referred to as Dynamic Error Estimation-based schemes, introduced and studied in Sect. 5 of [39] in the context of CKKS. In a nutshell, that paper presents an approach (which the authors of that paper attribute to Y. Polyakov) leveraging on a special message encoding which fixes

many of the coordinates of CKKS message space to be 0 and to use these to obtain, at decryption, an estimation of the noise variance for the ciphertext and to use this estimation to set the variance of their noise flooding mechanism (which essentially is equivalent to the noise smudging mechanism initially introduced in [3] for threshold FHE). Then, still in the context of CKKS, the authors of [39] present CPA^D attacks on a "natural" class of such schemes and leave open the following problem: "While our results on "dynamic" error estimation are negative, we have not ruled out achieving some weaker security notion with these techniques (for natural schemes)." In essence, the previous proposition is a positive answer to this open problem, with the weaker security notion being non-adaptive CPA^D security as defined in [38]. Let us also emphasize that FS_1 departs slightly from the blueprint depicted in [39], in that we are using the dynamic variance estimation to obtain a safe-bound on the true variance which leads to a correct variance for the smudging noise (with overwhelming probability). Let us also emphasize that the attacks in [39] do not apply to FS_1 as they leverage on the noise/message dependencies which naturally arise in CKKS (and other schemes such as BGV and BFV) when performing homomorphic multiplications. In contrast, the noises in an FS_1 ciphertext are message-independent (and remain so under the linear homomorphic operators).

Finally, we provide concrete parameters for FS_1 in appendix Sect. D.

4.3 Achieving "non-adaptive" LOH (FS_1')

Let us emphasize that the schemes considered in this section are not new, to the exception of the dynamic estimation based smudging technique that we use in the decryption function of the FS_1' scheme below.

$\mathsf{FS}_0^{(K)}$ and FS_1'. We now consider the usual multi-secret variant of FS_0, where K messages are encrypted by ciphertexts under K instances of FS_0 with *independent* key material but *using the same a vector*. Then $\mathsf{FS}_0^{(K)}(\lambda, n, q, t, \sigma_0)$ is defined as follows:

- $\mathsf{FS}_0^{(K)}.\mathsf{KeyGen}$: for $k \in [\![0, K-1]\!]$, uniformly pick $\mathsf{sk}^{(k)} \in \mathbb{Z}_q^n$.
- $\mathsf{FS}_0^{(K)}.\mathsf{Enc}$: given plaintext $M \in \mathbb{Z}_t^K$, uniformly pick a *single* $a \in \mathbb{Z}_q^n$ as well as vector $E \in \mathbb{Z}_q^K$ with each component drawn independently from χ_{σ_0}. Then, return $(a, B) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^K$ such that for $k \in [\![0, K-1]\!]$

$$B_k = \langle a, \mathsf{sk}^{(k)} \rangle + \Delta M_k + E_k.$$

- $\mathsf{FS}_0^{(K)}.\mathsf{Dec}$: given ciphertext $(a, B) \in \mathbb{Z}_q^{n+K}$, let

$$\mu_k = \mathsf{FS}_0.\mathsf{Dec}(a, B_k; \mathsf{sk}^{(k)}),$$

  for $k \in [\![0, K-1]\!]$. Then return $(\mu_0, ..., \mu_{K-1})$.

We now define a variant of $\mathsf{FS}_0^{(K+1)}$, FS_1', with built-in additional verifications following the Knowledge-of-Exponent pattern as well as smudging in the decryption function.

Recall Lemma 12 and let

$$K \geq \max\left\{\left\lceil \frac{\lambda}{\log_2 t} \right\rceil, 4\lambda \log 2 + 1\right\}. \tag{21}$$

We then build FS_1' from $\mathsf{FS}_0^{(K+1)}$ as follows:

- $\mathsf{FS}_1'.\mathsf{KeyGen}$: run $\mathsf{FS}_0^{(K+1)}.\mathsf{KeyGen}$ and let $\xi_1, ..., \xi_K$ be *secret multipliers* uniformly picked in $\mathbb{Z}_t^*$.
- $\mathsf{FS}_1'.\mathsf{Enc}$: given a plaintext $m \in \mathbb{Z}_t$, return

$$(a, B) = \mathsf{FS}_0^{(K+1)}.\mathsf{Enc}(m, [\xi_1 m]_t, ..., [\xi_K m]_t).$$

- $\mathsf{FS}_1'.\mathsf{ImVer}$: given ciphertext $c = (a, B) \in \mathbb{Z}_q^{n+K+1}$, let

$$(\mu_0, \mu_1, ..., \mu_K) = \mathsf{FS}_0^{(K+1)}.\mathsf{Dec}(c).$$

  When[15] $a = \vec{0}$ or if

$$\exists k \in [\![1, K]\!], [\xi_k \mu_0]_t \neq \mu_k, \tag{22}$$

  then return False. Otherwise (i.e. when $a \neq \vec{0}$ and $\forall k \in [\![1, K]\!], [\xi_k \mu_0]_t = \mu_k$), let

$$\varepsilon_k = B_k - \langle a, \mathsf{sk}^{(k)} \rangle - \Delta \mu_k,$$

  and compute $\hat{\sigma}^2 = \frac{1}{K} \sum_{i=1}^{K} \varepsilon_i^2$ *(remark that $\hat{\sigma}^2$ is computed over the K last slots in order to preserve independence between $\hat{\sigma}^2$ and ε_0, when all the ε_k's, $k \in [\![0, K]\!]$, are independent)*. Then, pick $\upsilon \in \mathbb{Z}_q$ following a centered Gaussian distribution of variance (following Eq. 19 and Lemma 4)

$$\sigma_{\text{smg}}^2 = \frac{2^{2\lambda} \bar{\sigma}_K^2 (\lambda + 1) \log 2}{\pi}. \tag{23}$$

  Finally, return False when $|\upsilon| \geq \frac{\Delta}{2}$ and True otherwise.
- $\mathsf{FS}_1'.\mathsf{Dec}$: given ciphertext $c = (a, B) \in \mathbb{Z}_q^{K+1}$, if $\mathsf{FS}_1'.\mathsf{ImVer}(c) = \mathsf{False}$, then return $\perp$. Otherwise, let $(\mu_0, \mu_1, ..., \mu_K) = \mathsf{FS}_0^{(K+1)}.\mathsf{Dec}(c)$ and return μ_0.

We emphasize that FS_1' has no direct add-by-const operator, since the multipliers are private.

[15] This is to eliminate a corner case in a later security proof.

$\mathbf{CPA_0^D}$ security of FS_1'. To prove the CPA_0^D security of FS_1' we now relate it to the FS_1 scheme that we have studied in the previous section. Indeed, in FS_1' the K last slots serve the double purpose of enforcing ciphertext verification as well as of estimating the noise variance for smudging the first (payload) slot. Lemma 16 essentially shows that the two schemes are equivalent over well-formed ciphertexts. This will then allow us to prove the CPA_0^D security of FS_1' based on the CPA_0^D security of FS_1 which we have previously established.

The following lemma implies that FS_1''s decryption function admits an alternate version which "knows the noise".

Lemma 16. *Let $\mathsf{ct} = (a, B) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1}$ denote a well-formed FS_1' ciphertext, i.e. such that for $k \in [\![0, K]\!]$ ($\xi_0 = 1$),*

$$B_k = \langle a, \mathsf{sk}^{(k)} \rangle + \Delta \xi_k m + e_k, \tag{24}$$

and the e_k's are iid and Gaussian. Let us consider an alternative decryption function, $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$, which eventually knows the e_k's and always proceeds by smudging without checking cond. (22), then $\mathsf{FS}_1'.\mathsf{Dec}$ is equivalent to $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$.

Proof. Let $V[e_k] = \sigma_{\mathrm{ct}}^2$.

Case 1 (correct $\mathsf{FS}_0^{(K+1)}$ decryption). In this case, condition (22) is satisfied and, therefore, FS_1''s decryption function proceeds, following Eq. (23), with smudging based on $\bar{\sigma}_K^2 \geq \sigma_{\mathrm{ct}}^2$ (with overwhelming probability, from Lemma 12 and Eg. 19). Hence, FS_1''s decryption function proceeds as $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$.

Case 2 (incorrect $\mathsf{FS}_0^{(K+1)}$ decryption #1). We now consider the case where a decryption error occurs *only* in the first slot, i.e. $\mathsf{FS}_0.\mathsf{Dec}(a, B_0; \mathsf{sk}^{(0)}) \neq m$. On the one hand, condition (22) is not satisfied and $\mathsf{FS}_1'.\mathsf{Dec}(c) = \perp$. On the other hand, $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$ proceeds following Eq. (23), with smudging based on $\bar{\sigma}_K^2 \geq \sigma_{\mathrm{ct}}^2$ (with overwhelming probability, from Lemma 12 and Eq. 19). As a consequence, it picks a smudging noise ν such that $e_0 + \upsilon \overset{i}{=} \upsilon$ i.e. such that $|\upsilon| >> |e_0| \geq \frac{\Delta}{2}$ (with overwhelming probability). Hence, $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$ also returns $\perp$ (with overwhelming probability).

Case 3 (incorrect $\mathsf{FS}_0^{(K+1)}$ decryption #2). Lastly we consider the case where a decryption error occurs in at least one of the K last slots (and possibly in the first one also) i.e. $\exists k \in [\![1, K]\!] : \mathsf{FS}_0.\mathsf{Dec}(a, B_k; \mathsf{sk}^{(k)}) \neq \xi_k m$. On the one hand, condition (22) is again not satisfied and $\mathsf{FS}_1'.\mathsf{Dec}(c) = \perp$. However, $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$ (which eventually knows the e_k's) proceeds by smudging as in the previous cases. Since an error occurs in at least one of the K last slots, we have $\|(e_1 \ \dots \ e_K)\|_\infty \geq \frac{\Delta}{2}$, hence $\hat{\sigma}^2 \geq \frac{\Delta^2}{4K}$, and (Eq. 19),

$$\bar{\sigma}_K^2 = \frac{K\hat{\sigma}^2}{K - 2\sqrt{K\lambda \log 2}} \geq \frac{\Delta^2}{4(K - 2\sqrt{K\lambda \log 2})}.$$

Thus, the Gaussian smudging in Lemma 4, leads to smudge with a Gaussian noise having variance

$$\sigma_{\text{smg}}^2 = \frac{(\lambda+1)2^{2\lambda}\bar{\sigma}_K^2 \log 2}{\pi} \geq \frac{(\lambda+1)2^{2\lambda}\Delta^2 \log 2}{4\pi(K - 2\sqrt{K\lambda \log 2})}. \tag{25}$$

Consider the following trivial bound for the Gaussian: $P(|X| \leq a) \leq \frac{2a}{\sqrt{2\pi}\sigma}$. Then, plugging (25) in that bound yields,

$$\begin{aligned} P\left(|v| \leq \frac{\Delta}{2}\right) &\leq \frac{\Delta}{\sqrt{2\pi}}\sqrt{\frac{4\pi(K - 2\sqrt{K\lambda \log 2})}{(\lambda+1)2^{2\lambda}\Delta^2 \log 2}}, \\ &\leq \frac{\sqrt{K - 2\sqrt{K\lambda \log 2}}}{2^{\lambda-1}\sqrt{2(\lambda+1)\log 2}} \\ &\leq \text{neg}(\lambda), \end{aligned}$$

as long as $K \leq O(\text{poly}(\lambda))$. Therefore, $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$ also replies $\perp$ with overwhelming probability. □

Remark that Lemma 16 above tells us that FS_1' is equivalent to FS_1 (with the Gaussian smudging of Lemma 4 replacing the uniform smudging of Lemma 3), as $\widehat{\mathsf{FS}_1'.\mathsf{Dec}}$ is exactly $\mathsf{FS}_1.\mathsf{Dec}$. As the next proposition establishes, the (non-adaptive) CPA^D security of FS_1' then follows from that of FS_1 (Prop. 15).

Proposition 17. *Let $K \geq 4\lambda \log 2 + 1$, FS_1' is CPA_0^D-secure.*

Proof. Let G_0 denote the CPA_0^D game against FS_1' and G_1 the CPA_0^D game against FS_1. The claim follows from the fact that the two games cannot be distinguished since, for both schemes, well-formed ciphertexts are indistinguishable from uniform over $\mathbb{Z}_q^{n+K+1}$ (from the LWE assumption) and from Lemma 16 which tells that over well-formed ciphertexts, $\mathsf{FS}_1'.\mathsf{Dec}$'s behavior in G_0 is indistinguishable from that of $\mathsf{FS}_1.\mathsf{Dec}$ in G_1. □

FS_1' and LOH. Several works since [11], e.g. [15,35,50], have consistently made the assumption that schemes like FS_1' satisfy a *weaker* variant of the LOH property with the constraint that the L^2-norm of the linear combinations that the adversary may apply is bounded[16] by $\frac{q\sqrt{\pi}}{2t\sigma_0\sqrt{(\lambda+1)\log 2}}$, a condition under which the scheme achieves (statistical) correctness.

Without such kind of restrictions, FS_1' does not satisfy the LOH property as the next proposition establishes.

[16] This follows from the Banaszczyk bound ([42, Lemma 2.2] and [4]) which states that given a vector x of iid discrete Gaussian deviates with variance σ_0^2, any $T \in \mathbb{R}^+$ and any $a \in \mathbb{R}^L$ we have that $P(|\langle x, a\rangle| \geq T\sigma_0\|a\|) \leq 2e^{-\pi T^2}$. Equating $2e^{-\pi T^2} = 2^{-\lambda}$ then yields $T = \sqrt{\frac{(\lambda+1)\log 2}{\pi}}$. A sufficient condition for correct decryption of a ciphertext of the form $c = \mathsf{Eval}(\texttt{lincomb}_a, c_0, ..., c_{L-1})$, where $c_0, ..., c_L$ denotes L fresh well-formed ciphertexts LWE ciphertexts, is thus that $T\sigma_0\|a\| < \frac{q}{2t}$, i.e. $q > 2tT\sigma_0\|a\|$.

Proposition 18. *FS'_1 does not have the LOH property.*

Proof. Recall the notations in Def. 5 (p. 10) and consider that the set of messages $(a_0, ..., a_{m-1})$ in that definition is $(1, 0, ..., 0)$. Hence c_0 is an encryption of 1, when $c_1, ..., c_{m-1}$ are all encryptions of 0. However, as soon as $m \geq n + K + 2$ is sufficiently large that $c_1, ..., c_{m-1}$ forms a generating set of vectors for $\mathbb{Z}_q^{n+K+1}$ with non-negligible probability, the adversary can find a linear combination of the $c_1, ..., c_{m-1}$ with coefficients in $\mathbb{Z}_q$ *thus depending on the c_j's* such that,

$$c'_0 = \sum_{j=1}^{m-1} \alpha_j(c_0, ..., c_{m-1})c_j = c_0,$$

Then, since c_0 is a fresh well-formed encryption of 1 under FS'_1, we have that $\mathsf{FS}'_1.\mathsf{ImVer}(c'_0) = \mathsf{FS}'_1.\mathsf{ImVer}(c_0) = \mathsf{True}$ as well as $\mathsf{FS}'_1.\mathsf{Dec}(c'_0) = \mathsf{FS}'_1.\mathsf{Dec}(c_0) = 1$ (with high probability). However, since c'_0 has been obtained by linearly combining encryptions of 0 we have $a'_0 = \Pi \cdot (a_0 \ ... \ a_{m-1})^T = 0 \neq \mathsf{FS}'_1.\mathsf{Dec}(c'_0)$, in violation of the LOH property. □

Interestingly, this attack strategy also leads a new CPA^D attack path on the vanilla Regev scheme, its RLWE variant and the other mainstream FHE schemes which are based on them. We descibe this such an attack in Sect. H.

Now, recall that we have done our CPA^D security proofs in the non-adaptive private key setting in which the adversary specifies all its requests under the drastic restriction that it has not yet seen any ciphertext. In this setting, the previous attack is not applicable and only a degenerate variant of the LOH property with $m = 0$ is relevant. We refer to this (rather weak) variant as LOH$_0$ and now assume that FS'_1 satisfies it.

Assumption 1 *Let $K \geq \left\lceil \frac{\lambda}{\log_2 t} \right\rceil$, FS'_1 satisfies the LOH$_0$ property.*

As discussed in Sect. 2.5, the LOH property requires a deterministic decryption function and FS'_1's is not. However, this does not disrupt LOH$_0$. Indeed, in the private key setting of FS'_1, remark that the only valid ciphertexts that a LOH$_0$ adversary could a priori create on its own would be "trivial" encryptions of 0 of the form $(\vec{0}, E)$, where E is a $K + 1$ dimensional vector such that $\|E\|_\infty < \frac{\Delta}{2}$ (as, indeed, such a ciphertext decrypts to $\vec{0}$ under $\mathsf{FS}_0^{(K+1)}$ and, as $\vec{0}$ satisfies condition 22, could decrypt to 0 under FS'_1 depending on smudging). However, this corner case is not an issue, as in the specification of our scheme, $\mathsf{FS}'_1.\mathsf{ImVer}$ returns False whenever $a = \vec{0}$. It follows that, for FS'_1, the LOH$_0$ property tells that an adversary can ex nihilo create on its own a ciphertext c such that $\mathsf{FS}'_1.\mathsf{ImVer}(c) = \mathsf{True}$ only with negligible probability.

We now show that FS'_1 achieves CCA0 security, as defined in Sect. 2.3.

Proposition 19. *FS'_1 is CCA0 secure under the assumption that it has the LOH$_0$ property.*

Proof. As just discussed, for FS_1', the LOH_0 property tells that an adversary can create on its own a ciphertext c such that $\mathsf{FS}_1'.\mathsf{ImVer}(c) = \mathsf{True}$ only with negligible probability. In the non-adaptive setting of CCA0 where the adversary specifies all its decryption requests before seeing any ciphertexts, the claim thus follows by a straightforward reduction to the CPA security of FS_1'. The reduction simply forwards encryption and challenge requests as is to its CPA challenger. Lastly, the reduction handles a decryption request over a ciphertext c by just returning $\perp$ unconditionally. □

So, eventually, the LOH_0 property earned us a little CCA security increment. At least sufficient to thwart (non-adaptive) CCA1 attacks such as the one we discussed at the end of Sect. 4.1 (see also Sect. G.1).

5 Concluding remarks

In this paper, our goal has been to investigate whether the LOH property and the associated Knowledge-of-Exponent design blueprint could help obtaining "beyond CCA1" secure linearly homomorphic schemes without relying on advanced SNARK machinery.

When working under the correctness assumption, the short answer is yes. Indeed, this paper has unveiled a fruitful connection between the LOH property and vCCA security, eventually yielding simple constructions achieving this strong CCA security notion at twice the cost of achieving only CPA security.

Relaxing the correctness assumption has, as is usually the case, revealed a more complicated picture. Although it can be expected that a CPA^D secure scheme with the LOH property achieves vCCA^D security (Sect. C), the second part of this paper shows that it is quite delicate to build schemes having these former properties by starting from a LWE-based scheme and extending it by naturally following the KEA design blueprint. Eventually, this strategy works only in a non-adaptive setting in which vCCA^D collapses onto a weak relaxation of CCA1. We are therefore left with the question asking how far can we go with our LWE-based line of schemes? If we conjecture that FS_1' is (adaptive) CPA^D secure, then it still does not satisfy the LOH assumption as the attack in the proof of Proposition 18 still works against it. However, that attack works under the assumption that $m \geq n + K + 1$. It turns out that the (adaptive) vCCA^D security of FS_1' can be established under the assumption that the adversary has access to only $m < n + K + 1$ ciphertexts, so under the conjecture that it is (adaptive) CPA^D secure and the assumption that it has the LOH property in the same conditions. To do so, we however have to associate a linearly homomorphic hash to FS_1' ciphertexts in order to deal with yet another corner case involving trivial encryptions of 0. This more speculative path is pursued in Sect. B.

Generalization to the public key setting however appears more problematic, even in the non-adaptive setting. Indeed, it is tempting to apply the well-known Regev's trick for turning FS_0 into a public key scheme [53,54]: define the public key as a large enough set of encryptions of 0 under FS_0 along with an additional

public encryption of 1 for injecting messages in ciphertexts via the mult-by-const operator[17]. Then, all well-formed ciphertexts would end up being linear combinations of the ciphertexts forming the public key, paving the way for a reduction able to feed all well-formed ciphertexts built by a vCCA^D adversary into the internal state of a (private key) CPA^D challenger by means of evaluation requests parameterized on the LOH extractor output. However, Claim 5.3 in [54][18] implies that we have to use $N \geq 4\lambda + (n + K + 1)\log_2 q$ ciphertexts to form a public-key. Unfortunately, that number of ciphertexts (which are observed, in the public-key setting, by a non-adaptive adversary before it specifies its requests) is much larger than the limit under which we can credibly claim adaptive vCCA^D security. Another trail, could be to adapt our schemes from the multi-secret to the RLWE setting [46] leading to a public key formed by a single encryption of 0. However, notwithstanding the other issues that may crop up when doing so, a reduction towards a CPA^D challenger would have to operate in the adversarially-chosen encryption randomness setup[19] (contrary to the above "Regev-style" approach where the LOH extractor would be able to retrieve the encryption randomness). This setup then would require the baseline scheme to achieve Strong CPA^D rather than only "standard" CPA^D security [9].

Acknowledgments

The authors would like to thank David Pointcheval for insightful discussions on secure encodings and the LOH property as well as for providing early feedback on some of the ideas in this paper. The authors would also like to thank Sven Schäge for interesting feedback.

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. Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption: characterizations, impossibility results, and applications. Designs, Codes and Cryptography **67**, 209–232 (2013)
3. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: EUROCRYPT. pp. 483–501 (2012)

[17] With the subtlety that an appropriate large noise would further have to be added to the encryptions of 0 forming the public key so as to smudge out the resulting message-noise dependency.

[18] Claim 5.3 in [54] establishes that, given a finite Abelian group G and any N elements g_0, ..., g_{N-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_{N-1} exceeds $\sqrt[4]{|G|/2^N}$ is bounded by $\sqrt[4]{|G|/2^N}$

[19] Similarly to the reductions in the proof of Prop. 17 and 18 in [21].

4. Banaszczyk, W.: Inequalites for convex bodies and polar reciprocal lattices in $\mathbb{R}^n$. Discrete & Computational Geometry **13**, 217–231 (1995)
5. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: IEEE SFCS. pp. 394–403 (1997)
6. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: CRYPTO. pp. 26–45 (1998)
7. Bendlin, R., Damgård, I.: Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In: TCC. pp. 201–218 (2010)
8. Bergamaschi, F., Costache, A., Dachman-Soled, D., Kippen, H., LaBuff, L., Tang, R.: Revisiting the security of approximate fhe with noise-flooding countermeasures. Tech. Rep. 2024/424, IACR ePrint (2024)
9. Bernard, O., Joye, M., Smart, N.P., Walter, M.: Drifting towards better error probabilities in fully homomorphic encryption schemes. Tech. Rep. 1718, IACR ePrint (2024)
10. Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., Paneth, O.: Succinct non-interactive arguments via linear interactive proofs. Tech. Rep. 718, IACR ePrint (2012)
11. Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., Paneth, O.: Succinct non-interactive arguments via linear interactive proofs. In: TCC. pp. 315–333 (2013)
12. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. Tech. Rep. 453, IACR ePrint (2010)
13. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In: PKC. pp. 1–16 (2011)
14. Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P.M.R., Sahai, A.: Threshold cryptosystems from threshold fully homomorphic encryption. In: CRYPTO. p. 565–596 (2018)
15. 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)
16. Boudgoust, K., Scholl, P.: Simple threshold (fully homomorphic) encryption from lwe with polynomial modulus. In: ASIACRYPT (2023)
17. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In: CRYPTO. pp. 505–524 (2011)
18. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: CRYPTO. pp. 868–886 (2012)
19. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. TOCT pp. 1–36 (2014)
20. Brown, W.C.: Matrices over commutative rings. Marcel Dekker, Inc. (1992)
21. 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)
22. Catalano, D., Fiore, D.: Using Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data. In: ACM SIGSAC. pp. 1518–1529 (2015)
23. Checri, M., Sirdey, R., Boudguiga, A., Bultel, J.P.: On the practical CPAD security of "exact" and threshold FHE schemes. In: CRYPTO. pp. 3–33 (2024)
24. Cheon, J.H., Choe, H., Passelègue, A., Stehlé, D., Suvanto, E.: Attacks against the IND-CPAD security of exact FHE schemes. In: CCS. pp. 2505 – 2519 (2024)
25. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: ASIACRYPT. pp. 409–437 (2017)
26. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: bootstrapping in less than 0.1 seconds. In: ASIACRYPT (2016)

27. Chowdhury, S., Sinha, S., Singh, A., Mishra, S., Chaudhary, C., Patranabis, S., Mukherjee, P., Chatterjee, A., Mukhopadhyay, D.: Efficient threshold fhe for privacy-preserving applications. Tech. Rep. 2022/165, IACR ePrint (2022)
28. 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)
29. Damgard, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: CRYPTO. p. 445–456 (1992)
30. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Tech. Rep. 2012/144, IACR ePrint (2012)
31. Fiore, D., Gennaro, R., , Pastro, V.: Efficiently verifiable computation on encrypted data. In: CCS. pp. 844–855 (2014)
32. Fiore, D., Gennaro, R., , Pastro, V.: Efficiently verifiable computation on encrypted data. Tech. Rep. ePrint 2014/202, IACR (2014)
33. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: CRYPTO. pp. 33–62 (2018)
34. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: EUROCRYPT. pp. 625–645 (2013)
35. Gennaro, R., Minelli, M., Nitulescu, A., Orrù, M.: Lattice-based zk-SNARKs from square span programs. In: CCS. pp. 556–573 (2018)
36. Ghosh, M.: Exponential tail bounds for chisquared random variables. Journal of Statistical Theory and Practice **15**, 35 (2021)
37. 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)
38. Li, B., Micciancio, D.: On the security of homomorphic encryption on approximate numbers. In: EUROCRYPT. pp. 648–677 (2021)
39. Li, B., Micciancio, D., Schultz, M., Sorrell, J.: Securing approximate homomorphic encryption using differential privacy. In: CRYPTO. pp. 560–589 (2022)
40. Li, B., Micciancio, D., Schultz-Wu, M., Sorrell, J.: Securing approximate homomorphic encryption using differential privacy. In: CRYPTO. p. 560–589 (2022)
41. Libert, B.: Leveraging small message spaces for CCA1 security in additively homomorphic and BGN-type encryption. In: EUROCRYPT (2025)
42. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: CT-RSA. pp. 319–339 (2011)
43. Lipmaa, H.: On the cca1-security of elgamal and damgård elgamal. In: Inscrypt. pp. 18–35 (2010)
44. Loftus, J., May, A., Smart, N.P., Vercauteren, F.: On CCA-secure somewhat homomorphic encryption. In: SAC. pp. 55–72 (2011)
45. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: Swifft: A modest proposal for fft hashing. In: FSE. pp. 54–72 (2008)
46. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: EUROCRYPT. p. 1–23 (2010)
47. Manulis, M., Nguyen, J.: Fully homomorphic encryption beyond IND-CCA1 security: Integrity through verifiability. In: EUROCRYPT. pp. 63–93 (2024)
48. Micciancio, D., Suhl, A.: Simulation-secure threshold pke from lwe with polynomial modulus. Tech. Rep. 2023/1728, IACR ePrint (2023)
49. Mouchet, C., Troncoso-Pastoriza, J., Bossuat, J.P., Hubaux, J.P.: Multiparty homomorphic encryption from Ring-Learning-with-Errors. In: PoPETS. pp. 291–311 (2021)

50. Nitulescu, A.: Lattice-based zero-knowledge SNARGs for arithmetic circuits. In: LATINCRYPT. pp. 217–236 (2019)
51. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: EUROCRYPT. pp. 223–238 (1999)
52. Passelègue, A., Stehlé, D.: Low communication threshold fully homomorphic encryption. In: ASIACRYPT. pp. 297–329 (2024)
53. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC. pp. 84–93 (2005)
54. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM **56** (2009)
55. Schäge, S.: New limits of provable security and applications to ElGamal encryption. In: EUROCRYPT. p. 255–285 (2024)

A Appendix sections organization

The appendix sections are organized as follows. First, Sect. B investigates whether and how vCCA^D security may be attained in the adaptive adversary setting, under the conjecture that FS_1' (and FS_1) achieves (adaptive) CPA^D security. Then, in the spirit of Sect. 3.1, Sect. C discusses the kind of black-box results that can be obtained in the regime where the correctness assumption is relaxed, and sketches how to obtain them. Subsequently, Sect. D further gives concrete guidelines for choosing the parameters of FS_1 (and FS_1'). For the sake of self-containedness, the other remaining appendix sections essentially contain either easy picks or additional background that is referred to from the main body or other appendix sections. Lastly, we have also included a table of contents at the end of this document to ease the reader's navigation.

B Achieving vCCA^D security, under the conjectured (adaptive) CPA^D security of FS_1' (and FS_1)

B.1 The ℓ-LOH property

Even if we are willing to conjecture that FS_1' is (adaptive) CPA^D secure, then it still does not satisfy the LOH assumption as the attack in the proof of Proposition 18 still works against it. However, that attack works under the assumption that $m \geq n + K + 1$. In this section, we thus investigate the restricted case where the LOH adversary only has access to $m < n + K + 1$ ciphertexts. Indeed, let $N = n + K + 1$ and assume that the adversary has access to only $0 \leq \ell < N$ randomly chosen ciphertexts. Then the probability for an arbitrary ciphertext to be in the span of these ℓ ciphertexts is upper bounded by[20] $\frac{q^\ell}{q^N} = q^{\ell - N}$. Equating $q^{\ell-N} = 2^{-\lambda}$ then yields $\ell = N - \lambda \frac{\log 2}{\log q}$. *We refer to the LOH property in this restricted setting as ℓ-LOH.* We thus now attempt to achieve (adaptive) ℓ-vCCA^D security in the private key setting (i.e. vCCA^D security where the adversary

[20] The bound is tight when $\mathbb{Z}_q$ is a field [20].

can cumulate no more than $\ell \leq N - \lambda\frac{\log 2}{\log q}$ encryption or challenge oracle calls). Further remark that since $\log_2 q$ generally is in $O(\lambda)$ (see also Sect. D), then we have $\ell \leq N - O(1)$.

Following this, there remain two issues to deal with.

The first is that FS'_1's decryption function is not deterministic (as discussed in Sect. 2.5, the LOH property requires deterministic decryption). This however can be dealt with by considering a variant of FS'_1 (or FS_1 in the rest of this Sect.) with *deterministic* decryption, e.g. by generating the smudging noise by means of a keyed PRF seeded on $H(c)$ for some hash function H). When this is so, our CPA_0^D security proofs (Prop. 15 and 17 as well as Lemma 16) remain valid "only" at the cost of modelling H as a Random Oracle.

The last issue is as follows. Let us consider a trivial encryption of 0 under $\mathsf{FS}_0^{(K+1)}$ of the form $(\vec{0}_n, E)$ where E is a $K+1$ dimensional vector such that $0 < \|E\|_\infty < \frac{\Delta}{2}$. Recall that $\mathsf{FS}'_1.\mathsf{ImVer}((\vec{0}_n, E)) = \mathsf{False}$ (as $\mathsf{FS}'_1.\mathsf{ImVer}((a, B))$ returns False whenever $a = 0$). Because of this, a ciphertext of the form

$$c'_0 = \mathsf{FS}'_1.\mathsf{Eval}(\texttt{lincomb}_\pi, c_0, ..., c_{\ell-1}) + (\vec{0}_n, E), \tag{26}$$

where $c_0, ..., c_{\ell-1}$ are the fresh well-formed ciphertexts available to the ℓ-LOH adversary, should yield $\mathsf{extract}(c'_0) = \varnothing$ and $a_0 = \perp$ (in Def. 5's notations, p. 10). However, when $\|E\|_\infty << \frac{\Delta}{2}$, $\mathsf{FS}'_1.\mathsf{ImVer}(c'_0) = \mathsf{True}$ and $\mathsf{FS}'_1.\mathsf{Dec}(c'_0) \neq \perp$, with high probability. We finally deal with this last issue by associating a tag to FS'_1 ciphertexts obtained by means of a collision-resistant (keyed) hash function, thus preventing a ℓ-LOH adversary to build ciphertexts of the form (26) on its own (as, as just argued, it can get a valid hash for vector $(\vec{0}_n, E)$ only with negligible probability when $\ell \leq N - \lambda\frac{\log 2}{\log q}$). Interestingly, this also makes our next scheme immune to the folklore (adaptive) CCA1 attack in Sect. G.2 (under the same restriction).

Lastly, we emphasize that the conjecture that FS'_1 is (adaptive) CPA^D secure (at least when less than $n + K + 1$ ciphertexts are available to the adversary) is not so far disproved by any attacks, indeed, to the best of our knowledge:

- The attacks in [39] do not apply to FS'_1 as they leverage on the noise/message dependencies which naturally arise in CKKS, and other schemes such as BGV and BFV, when performing homomorphic multiplications. In contrast, the noises in a FS'_1 ciphertext are message independent.
- Among the CPA^D attacks in [24], only the *non-adaptive* attack in Sect. 4 of that paper would be applicable to FS'_1. However, since it is non-adaptive, the CPA_0^D security of FS'_1 implies that it is not subject to it. The others attacks in [24] either need multiplications or bootstrappings.
- The adaptive LWE noise recovery dichotomic attack in [23] manipulates only ciphertexts of the form $\alpha \cdot c_0$ (with c_0 a well-formed fresh encryption of 0) and is heuristically thwarted by FS'_1's smudging (under the assumption of the independence of the E_k's, in the notations of Lemma 14, which is not stricto sensu true since the α coefficient choice during the dichotomy depends on outputs of the CPA^D decryption oracle which depend on the E_k's. However

we conjecture that this dependence is benign when the adversary is restricted to evaluate single-coefficient linear combinations as in the latter attack).

– Lastly, FS_1' is *provably* immune against the attack we present in Sect. H as soon as the adversary is restricted to see less than $N - \lambda\frac{\log 2}{\log q}$ ciphertexts (an assumption that we explicitly make in the rest of this Sect.). In the regime beyond this bound, it is however unclear how to obtain concrete attacks against FS_1' by following an attack path like that of Sect. H thanks to the smudging in FS_1''s decryption function (although, as above, the independence assumption on the E_k's, in the notations of Lemma 14, is jeopardized).

B.2 Linearly homomorphic hash functions

The last tool we need is thus a *collision-resistant* keyed linearly homomorphic one-way hash function

$$\hat{H} = (\mathsf{KeyGen}, \mathsf{Digest}, \mathsf{Add}, \mathsf{Mulc}),$$

with $\hat{H}.\mathsf{Digest} : \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1} \longrightarrow \mathcal{H}$, $\hat{H}.\mathsf{Add} : \mathcal{H}^2 \longrightarrow \mathcal{H}$ and $\hat{H}.\mathsf{Mulc} : \mathcal{H} \times \mathbb{Z}_q \longrightarrow \mathcal{H}$ such that, with the convention that $\oplus \equiv \hat{H}.\mathsf{Add}$ and $\odot \equiv \hat{H}.\mathsf{Mulc}$, the following property holds,

$$\hat{H}.\mathsf{Digest}\left(\sum_{l=0}^{L-1} \pi_l c_l\right) = \bigoplus_{l=0}^{L-1} \pi_l \odot \hat{H}.\mathsf{Digest}(c_l).$$

Although the construction can be instantiated from any linearly homomorphic hash function (with domain $\mathbb{Z}_q^N$), there are only a very limited number of candidate constructions in the state-of-the-art. As a concrete example, we can use the Fiore-Gennaro-Pastro hash function introduced in [31] ([32], Sect. 4.2) in order to build several VC schemes on top of the BV scheme [17]. We give the hash function details in Sect. I for completion.

B.3 Scheme FS_2'

Following the discussion just above, we now consider an additional scheme, FS_2', which we build from a variant of FS_1' with deterministic decryption (Sect. B.1), by further onboarding hash function $\hat{H}$ in the following Encrypt-then-Hash fashion:

– $\mathsf{FS}_2'.\mathsf{KeyGen}$: run $\mathsf{FS}_1'.\mathsf{KeyGen}$ as well as $\hat{H}.\mathsf{KeyGen}$.
– $\mathsf{FS}_2'.\mathsf{Enc}$: given $m \in \mathbb{Z}_t$, return

$$\mathsf{ct} = (c, \tau) = (\mathsf{FS}_1'.\mathsf{Enc}(m), \hat{H}.\mathsf{Digest}(c)).$$

– $\mathsf{FS}_2'.\mathsf{ImVer}$: given $\mathsf{ct} \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1} \times \mathcal{H}$, return False if either $\mathsf{FS}_1'.\mathsf{ImVer}(\mathsf{ct}.c) = \mathsf{False}$ or

$$\hat{H}.\mathsf{Digest}(\mathsf{ct}.c) \neq \mathsf{ct}.\tau, \tag{27}$$

and True otherwise.

– $\mathsf{FS}_2'.\mathsf{Dec}$: given $\mathsf{ct} \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1} \times \mathcal{H}$, if $\mathsf{FS}_2'.\mathsf{ImVer}(\mathsf{ct}) = \mathsf{False}$, then $\perp$ is returned. Otherwise return $\mathsf{FS}_1'.\mathsf{Dec}(\mathsf{ct}.c)$.

Consistently with Sect. B.1, we will now work under the following assumption.

Assumption 2 *Let $\ell \leq n + K + 1 - \lambda \frac{\log 2}{\log q}$, FS_2' has the ℓ-LOH property.*

B.4 ℓ-vCCA^D security of FS_2'

For the proof in this section, we assume a slightly modified mult-by-const operator and associated extractor. Indeed, we assume that the mult-by-const operator, $\mathtt{mul}_\alpha$, accepts $\alpha \in \mathbb{Z}_q$ rather than $\mathbb{Z}_t$ with the effect of producing an encryption of $[[\alpha]_t m]_t$ (under our running assumption that t divides q, Sect. 4.1). This operator naturally occurs whenever usual mult-by-const operators are chained during an homomorphic evaluation. Consistently, given a FS_2' ciphertext ct, we also assume that (π, β), as returned by $\mathsf{extract}(\mathsf{ct}, \mathsf{aux})$, is in $Z_q^m \times Z_q$ rather than $Z_t^m \times Z_t$ (in the notations of Def. 5, p. 10) and that the following property holds,

$$(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux}) \Leftrightarrow \mathsf{ct}.c = c_e, \tag{28}$$

with $\pi \in \mathbb{Z}_q^{|S_\mathcal{F}|}$, $\beta = 0$ (in the private key setting) and,

$$c_e = \mathsf{FS}_1'.\mathsf{Eval}(\mathtt{lincomb}_\pi, S_\mathcal{F}[0].c, ..., S_\mathcal{F}[|S_\mathcal{F}| - 1].c),$$

where $S_\mathcal{F}$ denotes a common list of well-formed ciphertexts on which an adversary and its challenger agree. This modification is a mild assumption as the usual extractor output is naturally a reduction modulo t of the modified one's output (again, when t divides q). This slight modification is important as it will allow us to rebuild the ciphertexts given by the adversary when we need to do so in the proof of the next proposition.

Proposition 20. *FS_2' is (adaptive) ℓ-vCCA^D secure, under the assumptions that it satisfies the ℓ-LOH property and that FS_1 is (adaptive) CPA^D secure.*

Proof. The proof works under the natural assumption that $\mathcal{A}$ and the challenger share a common numbering for the ciphertexts output by the encryption oracle (i.e., *fresh well-formed ciphertexts*).

We then start by one step of game hoping.

First game hop. Let G_0 be the ℓ-vCCA^D game against FS_2' and G_1 be the same game as G_0 where we modify the challenger as follows. First (consistently with the above common numbering assumption), the challenger $\mathcal{C}_1$ stores the fresh ciphertexts it generates in an internal state $S_\mathcal{F}$ containing message-message-ciphertext triplets (and $\mathcal{A}$ is assumed to maintain a similar array). Then, when handling a decryption request on ciphertext ct, it first invokes the LOH extractor to verify that

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

rather than checking conditions (22) and (27) in $\mathsf{FS}_2'.\mathsf{ImVer}(\mathsf{ct})$, and returns $\perp$ when this is not the case. Otherwise, $\mathcal{C}_1$ proceeds as follows. Let $(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux})$[21],

$$\mu_0 = \sum_{i=0}^{|S_{\mathcal{F}}|-1} \pi_i S_{\mathcal{F}}[i].m_0 \text{ and } \mu_1 = \sum_{i=0}^{|S_{\mathcal{F}}|-1} \pi_i S_{\mathcal{F}}[i].m_1. \tag{29}$$

Then, *whenever* $\mu_0 = \mu_1$ (otherwise it returns $\perp$), *rather than returning* $\mathsf{FS}_1'.\mathsf{Dec}((\mathsf{ct}.c.a, \mathsf{ct}.c.B))$, $\mathcal{C}_1$ returns $\mathsf{FS}_1.\mathsf{Dec}((\mathsf{ct}.c.a, B'))$ with

$$B_0' = B_0 \tag{30}$$

and, for $k \in [\![1, K]\!]$,

$$B_k' = B_k - \Delta\xi_k \mu_0. \tag{31}$$

We now argue that G_0 and G_1 cannot be distinguished. Indeed, in the case where $\mathsf{extract}(\mathsf{ct}) = \varnothing$, the indistinguishability of the two games follows from the ℓ-LOH property as, in this case, we have that $\mathsf{FS}_2'.\mathsf{ImVer}(\mathsf{ct}) = \mathsf{False}$ with overwhelming probability. When, $\mathsf{extract}(\mathsf{ct}) \neq \varnothing$, then (by Eq. 28) $\mathsf{ct}.c$ is a well-formed FS_1' ciphertext, i.e. is such that for $k \in [\![0, K]\!]$ ($\xi_0 = 1$),

$$\mathsf{ct}.c.B_k = \langle \mathsf{ct}.c.a, \mathsf{sk}^{(k)} \rangle + \Delta\xi_k \mu + e_k,$$

and the e_k's are iid and Gaussian, which may decrypt to $\perp$ depending on the e_k's (remark that μ may not be $\mathbb{Z}_t$). Now (recall the properties of the extractor discussed at the begining of this Sect.), when $\mu_0 = \mu_1$ in Eq. (29), we exactly have that $\mu = \mu_0$. Hence, Eq. (31) turns $(\mathsf{ct}.c.a, \mathsf{ct}.c.B)$ into a well-formed ciphertext $(\mathsf{ct}.c.a, B')$ under FS_1 with

$$B_0' = \langle \mathsf{ct}.c.a, \mathsf{sk}^{(0)} \rangle + \Delta\mu + e_0,$$

and, for $k \in [\![1, K]\!]$,

$$B_k' = \langle \mathsf{ct}.c.a, \mathsf{sk}^{(k)} \rangle + e_k.$$

The indistinguishability of the two games, in the case where $\mathsf{extract}(\mathsf{ct}) \neq \varnothing$, thus follows from the indistinguishability between FS_1's and FS_1''s decryption functions which we have established in Lemma 16.

Second game hop. We now consider game G_2 where we further modify the challenger $\mathcal{C}_1$ such that (recall $\mathsf{FS}_1'.\mathsf{Enc}$ definition on p. 24) the new challenger $\mathcal{C}_2$ replies to a challenge request on $m_0 \neq m_1 \in \mathbb{Z}_t^2$ with ciphertext

$$\mathsf{ct}' = (c', \tau') = (\mathsf{FS}_0^{(K+1)}.\mathsf{Enc}(m_\gamma, \xi_1 m_0, ..., \xi_K m_0), \hat{H}.\mathsf{Digest}(c')),$$

rather than, in G_1,

$$\mathsf{ct}^* = (c^*, \tau) = (\mathsf{FS}_0^{(K+1)}.\mathsf{Enc}(m_\gamma, \xi_1 m_\gamma, ..., \xi_K m_\gamma), \hat{H}.\mathsf{Digest}(c^*)).$$

These two games cannot be distinguished for the following reasons:

[21] Recall, following Def. 5 (p. 10), that the LOH assumption enforces linear (rather than affine) combinations in the private key setting. Hence, in the context of this proof, when the LOH extractor does not return $\varnothing$, we get $(\pi, \beta) = \mathsf{extract}(\mathsf{ct}, \mathsf{aux})$ with $\beta = 0$.

- Both c' and c^* are indistinguishable from uniform (by the LWE assumption).
- When handling a decryption request over a ciphertext ct such that $\varnothing \neq \mathsf{extract}(\mathsf{ct}, \mathsf{aux}) = (\pi, \beta)$ (again, with $\beta = 0$ in the private key setting), and $\mu_0 = \mu_1$ in Eq. (29), in which case we also have $\mu_0 = \mu_\gamma$, both challengers reply with $\mathsf{FS}_1.\mathsf{Dec}((\mathsf{ct}.c.a, B'))$ (with B' obtained, in both cases, following Eqs. 30 and 31 above). When this is not the case, i.e. when either $\mathsf{extract}(\mathsf{ct}, \mathsf{aux}) = \varnothing$ or $\mu_0 \neq \mu_1$ (in Eq. 29), then both challengers consistently reply $\perp$.

Final reduction. We next prove that an adversary $\mathcal{B}$ against the CPA^D security of FS_1 (consistently with $\mathcal{C}_2$ specification) can be build using an adversary $\mathcal{A}$ against G_2 as a subroutine. The reduction thus maintains an initially empty state $S_\mathcal{F}$ containing message-message-ciphertext (under FS'_2) triplets (and $\mathcal{A}$'s is assumed to maintain a similar state). The reduction also maintains a conversion table between ciphertext indices in $S_\mathcal{F}$ and S, where S is the internal state of the CPA^D challenger against FS_1, $\mathfrak{I} : [\![0, |S_\mathcal{F}| - 1]\!] \longrightarrow [\![0, |S| - 1]\!]$ such that $S_\mathcal{F}[l].c = S[\mathfrak{I}(l)].c$ (it does so by means of a counter ctr initially set to 0). *Note that to ease understanding, we also show the internal state of the CPA^D challenger in the following.* The reduction subsequently picks the multipliers $\xi_1, ..., \xi_K$ uniformly in $\mathbb{Z}_t^*$ and runs $\hat{H}.\mathsf{KeyGen}$. It then handles $\mathcal{A}$'s requests as follows:

- When receiving an encryption request over message $m \in \mathbb{Z}_t$, $\mathcal{B}$ transfers it as is to the CPA^D challenger to get ciphertext $c = (a, B)$. It then computes ciphertext $c' = (a, B')$, with $B'_0 = B_0$ and $B'_k = B_k + \Delta\xi_k m$ (for $k \in [\![1, K]\!]$) and returns $\mathsf{ct} = (c', \hat{H}.\mathsf{Digest}(c'))$ to $\mathcal{A}$ after updating its internal state as $S_\mathcal{F} := [S_\mathcal{F}; (m, m, \mathsf{ct})]$ and its conversion table as $\mathfrak{I}(|S_\mathcal{F}| - 1) := \mathsf{ctr}$ (then doing $\mathsf{ctr} := \mathsf{ctr}{+}1$). *This has the side effect of updating the CPA^D challenger's internal state as $S := (S; (m, m, c))$ (with the invariant that $\mathsf{ctr} = |S|$).*
- When receiving a challenge request over messages $m_0 \neq m_1 \in \mathbb{Z}_t^2$, then $\mathcal{B}$ also transfers it as is to the CPA^D challenger to get ciphertext $c^* = (a^*, B^*)$. It then computes ciphertext $c'^* = (a^*, B'^*)$, with $B'^*_0 = B^*_0$ and $B'^*_k = B^*_k + \Delta\xi_k m_0$ (for $k \in [\![1, K]\!]$), yielding $c'^* = \mathsf{FS}_0^{(K+1)}.\mathsf{Enc}(m_\gamma, \xi_1 m_0, ..., \xi_K m_0)$ consistently with $\mathcal{C}_2$'s specification, and returns $\mathsf{ct} = (c'^*, \hat{H}.\mathsf{Digest}(c'^*))$ to $\mathcal{A}$ after updating its internal state and conversion table as in the above case. *As in the previous case, this has the side effect of updating the CPA^D challenger's internal state as $S := (S; (m_0, m_1, c^*))$.*
- When receiving a decryption request over ciphertext $\mathsf{ct} \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1} \times \mathcal{H}$, $\mathcal{B}$ proceeds as follows:
  - *Fresh ciphertext.* If $\exists l : S_\mathcal{F}[l].c = \mathsf{ct}$, it issues a decryption request with index $\mathfrak{I}(l)$ towards its CPA^D challenger and return the result to $\mathcal{A}$. Remark that the adversary expects the decryption of FS'_1 ciphertext $\mathsf{ct}.c = S_\mathcal{F}[l].c$ while the reduction returns in fact the decryption of FS_1 ciphertext $S[\mathfrak{I}(l)].c$ however with the guarantee that $\mathsf{FS}_1.\mathsf{Dec}(S[\mathfrak{I}(l)].c)$ is indistinguishable $\mathsf{FS}'_1.\mathsf{Dec}(S_\mathcal{F}[l].c)$ (by Lemma 16).
  - *Evaluated ciphertext.* Otherwise, it invokes $\mathsf{extract}(\mathsf{ct})$ getting either $\varnothing$, in which case $\perp$ is returned to $\mathcal{A}$ (following $\mathcal{C}_1$'s specification in the

first game hop), or a vector $\pi \in \mathbb{Z}_q^{|S_{\mathcal{F}}|}$. Let $l_0, ..., l_{L-1}$ denote the indices such that $\pi_{l_j} \neq 0$, then $\mathcal{B}$ issues an evaluation request over indices $\mathfrak{I}(l_0), ..., \mathfrak{I}(l_{L-1})$ towards its CPA^D challenger to get ciphertext

$$c_e = \mathsf{FS}_1.\mathsf{Eval}(\mathtt{lincomb}_{\pi_{l_0},...,\pi_{l_{L-1}}}, S[\mathfrak{I}(l_0)].c, ..., S[\mathfrak{I}(l_{L-1})].c).$$

Remark that, when $\mu_0 = \mu_1$ in Eq. (29), we have $\mathsf{ct}.c.a = c_e.a$, $\mathsf{ct}.c.B_0 = c_e.B_0$ as well as $\mathsf{ct}.c.B_k = c_e.B_k + \Delta\xi_k\mu_0$ ($k \in [\![1, K]\!]$). *This has the side effect of augmenting the internal state of the CPA^D challenger with the triplet*

$$\left(\sum_{i=0}^{L-1} \pi_{l_i} S[\mathfrak{I}(l_i)].m_0, \sum_{i=0}^{L-1} \pi_{l_i} S[\mathfrak{I}(l_i)].m_1, c_e\right). \tag{32}$$

Additionally $\mathcal{B}$ increments ctr to keep $\mathfrak{I}$ consistent with the internal state of its challenger (i.e. maintain the invariant that $\mathsf{ctr} = |S|$). Finally, $\mathcal{B}$ issues a decryption request with index $\mathsf{ctr} - 1$ towards its CPA^D challenger to get a decryption of c_e (which may be $\perp$) which it sends back to $\mathcal{A}$. Remark that because of the above relation between $\mathsf{ct}.c$ and c_e we have that $\mathsf{FS}_1'.\mathsf{Dec}(\mathsf{ct}.c)$ is indistinguishable from $\mathsf{FS}_1.\mathsf{Dec}(c_e)$ (by Lemma 16). Recall that, when handling a decryption request on state index l, the CPA^D challenger's decryption oracle verifies that $S[l].m_0 = S[l].m_1$ and returns the decryption of $S[l].c$ only when this is the case (and $\perp$ otherwise) so the reduction does not even have to check this.

The claim then follows from the (adaptive) CPA^{D}security of FS_1 which we conjecture in this section. □

As a last remark, it may be worth noting that applying the Encrypt-then-Hash blueprint that yielded FS_2' from FS_1' (Sect. B.3) *directly to FS_1* may be sufficient to obtain a scheme that also credibly satisfies the ℓ-LOH property, i.e. under the not too far-fetched assumption that the (linearly homomorphic) hash verification is sufficient to get this property. Although this depart from the more conventional Knowledge-of-Exponent blueprint, the resulting scheme (say FS_2) would be simpler and would also admit a less cumbersome proof of ℓ-vCCA^D security, in particular with a more direct final reduction towards a CPA^D challenger against FS_1 which is left to the reader.

C General results for approximate schemes?

As discussed in Sect. 4, a natural question is whether black-box results, such as Prop. 6 and 7, can be obtained in the case of approximate schemes? E.g. if any CPA^D secure linearly homomorphic scheme with the LOH property is vCCA^D secure? A natural proof strategy for obtaining such a result (say, first, in the private key setting) is to perform a reduction towards a CPA^D challenger. For such a reduction to work, it then has to handle decryption requests over well-formed evaluated ciphextexts from the vCCA^D adversary by means of its

CPA^D challenger's one. To do so, the reduction then has to populate the internal state of its CPA^D challenger with the exact same ciphertext provided by the vCCA^D adversary, by means of evaluation requests parameterized by the LOH extractor output and the set of fresh well-formed ciphertexts output by the encryption oracle. Then, in the notations of Def. 5 (p. 10) and following Eq. (9), the reduction works by relying on the property that[22],

$$(\pi, \beta) = \mathsf{extract}(c', \mathsf{aux}) \Leftrightarrow c' = \mathcal{E}_H.\mathsf{Eval}(\mathtt{lincomb}_\pi, c_1, ..., c_m),$$

(recall that $\beta = 0$ in the private key setting). Still, as illustrated in Sect. 4 and B, building approximate or somewhat correct (LWE-based) schemes satisfying even weak variants of the LOH property is particularly delicate. So it seems to us that the kind of black-box results sketched above has a limited practical relevance.

We also emphasize that, in the public-key setting, in order for a reduction to properly rely on a CPA^D challenger's decryption oracle to handle decryption requests from a vCCA^D adversary, it further has to populate the challenger's internal state with the well-formed fresh ciphertexts generated on its own by the adversary (which, in that case, controls the encryption randomness). To do so, the reduction hence necessarily has to operate in the adversarially-chosen encryption randomness setup which is accounted for by the notion of Strong CPA^D security introduced in [9][23]. We think the only way whereby this reliance on Strong CPA^D could be avoided, would be by proceeding via a reduction which does not have to rely on a challenger with a decryption oracle (e.g a reduction to the CPA rather than "CPA^D" security of the homomorphic scheme). This however appears difficult to achieve without introducing non-black box assumptions on $\mathcal{E}_H$ as the reduction would then have to handle the adversary's decryption requests without relying on any decryption oracle. .

D Choosing the ciphertext modulus for FS_1

Recall FS_1 definition in Sect. 4.2 as well as Eq. (7), we consider a FS_1 ciphertext with noise variance σ^2_{ct}. For such a ciphertext v is uniformly picked in $[\![-B, B]\!]$ with $B = 2^\lambda \bar{\sigma}_K \sqrt{2(\lambda+1)\log 2}$ and $\bar{\sigma}_K$ with $\bar{\sigma}^2_K = \frac{K\hat{\sigma}^2}{K - 2\sqrt{K\lambda \log 2}}$, following Eq. (19). Since, $E[\hat{\sigma}^2] = \sigma^2_{\mathsf{ct}}$, then, *on average*,

$$B = 2^\lambda \sigma_{\mathsf{ct}} \sqrt{\frac{2K(\lambda+1)\log 2}{K - 2\sqrt{K\lambda \log 2}}}. \tag{33}$$

Now consider the following upper tail bound for the χ^2 distribution [36].

[22] Although, stricto sensu, Eq. (9) only gives us the right to left implication (see the discussion at the beginning of Sect. B.4 leading to Eq. 28).

[23] Similarly to the reductions in the proof of Prop. 17 and 18 in [21].

Theorem 21 ([36], Theorem 1.). *Let X follows χ^2_K, then for $c > 0$,*

$$P(X > K + c) \leq e^{-\frac{c^2}{4(K+c)}}.$$

Since $\frac{K\hat{\sigma^2}}{\sigma^2_{\mathsf{ct}}}$ follows χ^2_K, we have that,

$$P\left(\hat{\sigma}^2 > \frac{\sigma^2_{\mathsf{ct}}(K+c)}{K}\right) \leq e^{-\frac{c^2}{4(K+c)}}$$

Assuming we target the probability of erroneous decryption to be less than a preset target value of $2^{-\epsilon}$. Then, letting

$$e^{-\frac{c^2}{4(K+c)}} = 2^{-\epsilon}$$

yields

$$c^* = 2(\epsilon \log 2 + \sqrt{\epsilon \log 2(\epsilon \log 2 + K)}).$$

We may then use $\frac{\sigma^2_{\mathsf{ct}}(K+c^*)}{K}$ as a bound for $\hat{\sigma}^2$, plug it in Eq. (33) (instead of σ^2_{ct}) and thus, since v is uniformly distributed in $[-B, B]$, choose q such that,

$$\begin{aligned}\frac{q}{2t} &> 2^{\lambda}\sqrt{\frac{\sigma^2_{\mathsf{ct}}(K+c^*)}{K}}\sqrt{\frac{2K(\lambda+1)\log 2}{K - 2\sqrt{K\lambda \log 2}}} \\ &> 2^{\lambda}\sigma_{\mathsf{ct}}\sqrt{\frac{2(K+c^*)(\lambda+1)\log 2}{K - 2\sqrt{K\lambda \log 2}}}\end{aligned}$$

Also recall from Sect. 4.2 that $K \geq 4\lambda \log 2 + 1$.

If we apply a linear combination $\alpha_0, ..., \alpha_L$ over fresh FS_1 ciphertexts with *independent* noises (note that since FS_1 is CPA^D secure, even only non-adaptively, evaluating such a linear combination over ciphertexts with non-independent noise decreases reliability but causes no security issue), then the variance of the resulting noise is

$$\sigma^2_{\mathsf{ct}} = \sigma^2_0 \sum_{j=0}^{L-1} \alpha_j^2.$$

We will refer to $\|\alpha\|^2 = \sum_{j=0}^{L-1} \alpha_j^2$ as an L_2-*budget*. For a given such budget, following the above calculations, one may perform either (considering both extremes) $\|\alpha\|^2$ additions of ciphertexts with *independent* noises or only $\|\alpha\|$ additions of a given ciphertext with itself, and then achieve $2^{-\epsilon}$ probability of getting $\perp$ upon decryption. Table 1 provides some examples of parameters for FS_1. For example, with a plaintext modulus of 2^{32}, a ciphertext modulus q on around 170 bits allows to sum up to 1000 ciphertexts with independent noises or up to around 30 times the given ciphtertext with itself, in both cases, with a probability on average less than 2^{-40} of getting $\perp$ at decryption of the result.

t	$\log_2 q$	n	$\hat{\lambda}$	t	$\log_2 q$	n	$\hat{\lambda}$
2	146	8192	171	2	150	8192	165
256	153	8192	160	256	157	8192	154
2^{16}	161	8192	149	2^{16}	165	8192	144
2^{32}	177	8192	130	2^{32}	181	8192	126
2^{64}	209	10240	141	2^{64}	213	10240	137
2^{64}	209	16384	266	2^{64}	213	16384	259

Table 1. Example of parameters for FS_1, in function of the plaintext modulus t. With an L_2-*budget* of 1000 (left half) and 10^6 (right half) as well as $\sigma_0 = 3.19$, $\epsilon = 40$ (i.e. a probability of decrypting to $\perp$ below $2^{-\epsilon} = 2^{-40}$). The target security level is $\lambda = 128$ ($K = 355$), estimated security levels ($\hat{\lambda}$) have been computed by means of the lattice-estimator. Note that increasing K, e.g. to 1000, leads slightly smaller modulus, e.g. 5 bits less.

E Additional security notions

E.1 Additional variants of CPA^D security

In addition to the multiple-challenge notion initially defined in [38] (Sect. 2.3), the following weaker restrictions of it have been defined and studied in [21]:

- CPA_2^D: restriction of CPA^D to the single challenge case where the adversary is allowed only one request of the form $(\texttt{test messages}, m_0, m_1)$ with $m_0 \neq m_1$.
- CPA_1^D: restriction of CPA_2^D with the decryption oracle closing after the unique challenge request (similar in spirit to the CCA1/CCA2 definitions, hence the choice for the names).
- KR^D: a challenge-less variant of CPA^D in which the adversary wins the game when it retrieves the secret decryption key.

With respect to these latter notions, [21] has established the following separation results in the general regime where approximate or somewhat correct FHE schemes are considered:

$$\mathrm{CPA} \prec \mathrm{CPA}_1^D \prec \mathrm{CPA}_2^D \prec \mathrm{CPA}^D. \tag{34}$$

Note that CPA_1^D is different from non-adaptive CPA^D (CPA_0^D in this paper) as defined and studied in [38] (recall also Sect. 2.3). Indeed, there is a (not so slight) difference between the notion of adaptability as understood in the multiple-challenge context of [38] (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 decryption requests before the *unique* challenge ciphertext is published).

E.2 CPA and CCA1 security.

Recall that by convention CPA and CCA1 security notion are usually single challenge. We define them relatively to the CPA^D game of Sect. 2.3.

In the *CPA game*, the adversary only has access to encryption requests and can perform a unique challenge request. Note that the encryption oracle is necessary only in the private key case as, in the public key case, the adversary can generate ciphertexts on its own. In the CPA game, there is also no need for an evaluation oracle since the adversary can always perform homomorphic evaluation on its own and their is no need to fill a game state (recall that in the CPA^D game, the purpose of the encryption and evaluation oracles is to fill the game state with well-formed ciphertexts for handling subsequent decryption requests on state indices).

In the *CCA1 game*, the adversary has access to encryption requests and can also perform a single challenge request. Before this unique challenge request, the adversary is additionally granted access to a first step decryption oracle which simply proceeds as follows:

- Decryption request (before the unique challenge request). When $\mathcal{A}$ queries $(\texttt{ciphertext}, c)$: return her $\mathsf{Dec}(c)$.

Then after the single challenge request, the decryption oracle systematically replies $\perp$. Note that the CCA1 game has no evaluation oracle as the adversary performs the homomorphic evaluations on its own in both the private and public key setting and there is no need to fill a game state (since the decryption oracle accepts arbitrary ciphertexts rather than indices pointing to well-formed ciphertexts stored in a game state).

In the general regime where approximate or somewhat correct FHE schemes are allowed, we also have the following separation [21]:

$$\mathrm{CPA} \prec \mathrm{CPA}_1^D \prec \mathrm{CCA1}. \tag{35}$$

E.3 Separation results for CCA0

In this section, for sanity checking, we establish a few easy separation results to position the CCA0 security notion (Sect. 2.3) with respect to other ones (we do so in the private key setting and implicitly assuming single challenge). We do not claim that we are the first to consider this rather weak (yet intuitive) relaxation of CCA security.

Proposition 22. *If there exist a correct private key scheme S which is CCA1 secure, then there exists a scheme S' which is CCA0 secure and CCA1 insecure.*

Proof. The proof works under the mild assumption that $|\mathcal{P}| \geq O(2^\lambda)$. Let $S = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a CCA1 secure private key correct scheme. We then consider $S' = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}')$ such that:

$$\mathsf{Dec}'(c) = \begin{cases} \mathsf{sk} & \text{if } \mathsf{Dec}(c) = 0 \\ \mathsf{Dec}(c) & \text{otherwise} \end{cases}$$

S' is CCA0 secure. CCA1 security implies CCA0 security, hence S is CCA0 secure. Then S''s CCA0 security follows from a trivial reduction to that of S with

the reduction transferring all decryption requests consisting of a priori chosen ciphertext from the CCA0 adversary as is to the CCA0 challenger against S. Finally, under the assumption that $|\mathcal{P}| \geq O(2^\lambda)$, $\mathcal{A}$ has a negligible probability of submitting an encryption of 0.

S' is CCA1 insecure. The CCA1 adversary simply asks an encryption of 0 to get $c = \mathsf{Enc}(0)$ and then ask the decryption of c to get $\mathsf{sk} = \mathsf{Dec}'(c)$. Then, once the challenge ciphertext is obtained, $\mathcal{A}$ simply decrypts it on its own. □

We also separate CCA0 with the notion of non-adaptive CPA^D security from [38] which we refer to as CPA_0^D (Sect. 2.3).

Proposition 23. *If there exist a correct private key scheme S which is CCA0 secure, then there exists a scheme S' which is CPA_0^D secure and CCA0 insecure.*

Proof. Since CCA0, in the private key setting, is equivalent to vCCA_0^D (Sect. 2.3), the proof is similar to that of Prop. 1 in [21] ($\mathrm{CPA}^D < \mathrm{vCCA}^D$). □

F The Paillier cryptosystem

This section briefly presents Paillier's original cryptosystem [51], denoted $\mathcal{E}_P$, which security is grounded in the Composite Residuosity Class Problem hardness assumption. The scheme is partially homomorphic allowing additions, or multiplications by a constant, but does not support multiplications between two ciphertexts. Note that a relatively simple modification of this scheme, described in [22], allows to perform one level of multiplications.

Let n be an RSA modulus. The plaintext space is $\mathbb{Z}_n$ and the ciphertext space is $\mathbb{Z}_{n^2}^\times$. Let $\mathcal{S}_n$ be the set $\mathcal{S}_n = \{u \in \mathbb{Z}_{n^2}^\times \mid u \equiv 1 \pmod{n}\}$, which is a multiplicative subgroup of $\mathbb{Z}_{n^2}^\times$. For all $u \in \mathcal{S}_n$ we define the function $L : \mathcal{S}_n \to \mathbb{Z}_n$, such that $L(u) = \frac{u-1}{n}$.

- $\mathcal{E}_P.\mathsf{KeyGen}$: sample a RSA modulus $n = pq$ such that p and q are distinct large prime numbers and such that $\gcd(pq, (p-1)(q-1)) = 1$. Let $\varphi(n) = (p-1)(q-1)$ and $\omega := \omega(n) = \mathrm{lcm}(p-1, q-1)$. Choose uniformly at random an integer $g \in \mathbb{Z}_{n^2}^\times$, such that $L(g^\omega \pmod{n^2}) \wedge n = 1$. Set the public key $\mathsf{pk} = (n, g)$ and the secret key $\mathsf{sk} = \omega(n)$.
- $\mathcal{E}_P.\mathsf{Enc}$: given $m \in \mathbb{Z}_n$ and pk, sample uniformly at random $r \xleftarrow{\$} (\mathbb{Z}_n)^\times$ and return $c = g^m r^n \pmod{n^2}$.
- $\mathcal{E}_P.\mathsf{Dec}$: given $c \in \mathbb{Z}_{n^2}$ and sk, return $\dfrac{L(c^{\mathsf{sk}} \pmod{n^2})}{L(g^{\mathsf{sk}} \pmod{n^2})} \pmod{n}$.
- $\mathcal{E}_P.\mathsf{Add}$: given $c, c' \in \mathbb{Z}_{n^2}^2$, compute and return $[c \cdot c']_{n^2}$.
- $\mathcal{E}_P.\mathsf{Mulc}$: given $\alpha \in \mathbb{Z}_n$ and $c \in \mathbb{Z}_{n^2}$, compute and return $[c^\alpha]_{n^2}$.

As a notable property with respect to the present work, the Paillier scheme achieves perfect correctness.

G Folklore CCA1 attacks against Regev

In this section, we briefly present two lines of folklore CCA1 attacks against the Regev scheme (FS_0).

G.1 A non-adaptive ill-formed ciphertext-based attack

As discussed in Sect. 4.1, let us remark that the the ill-formed ciphertext $(-\Delta\mathbf{1}_i, 0)$ decrypts to $[\mathsf{sk}_i]_t$, where $\mathbf{1}_i$ is the i-th standard basis vector. As a slight simplification, let us assume that $\Delta = \frac{q}{t} = \frac{2^k}{2^l} = 2^{k-l}$ and asks for the decryption of ill-formed ciphertext $c_l = \left(-\frac{\Delta}{t}\mathbf{1}_i, 0\right) = (-2^{k-2l}\mathbf{1}_i, 0)$ which gives $\left[\left\lceil \frac{\mathsf{sk}_i}{t} \right\rfloor\right]_t$. Since $\frac{\mathsf{sk}_i}{t} = \left\lfloor \frac{\mathsf{sk}_i}{t} \right\rfloor + \frac{[\mathsf{sk}_i]_t}{t}$, two cases can then occur,

- When $\frac{[\mathsf{sk}_i]_t}{t} < \frac{1}{2}$ (recall that we know $[\mathsf{sk}_i]_t$ from above), $\mathsf{FS}_0.\mathsf{Dec}(c_l) = \left[\left\lceil \frac{\mathsf{sk}_i}{t} \right\rfloor\right]_t = \left[\left\lfloor \frac{\mathsf{sk}_i}{t} \right\rfloor\right]_t$ and the decryption of c_l straightaway gives us bits $l, l+1, ..., 2l-1$ of sk_i.
- When $\frac{[\mathsf{sk}_i]_t}{t} \geq \frac{1}{2}$, $\mathsf{FS}_0.\mathsf{Dec}(c_l) = \left[\left\lceil \frac{\mathsf{sk}_i}{t} \right\rfloor\right]_t = \left[\left\lfloor \frac{\mathsf{sk}_i}{t} \right\rfloor + 1\right]_t$. So either $\mathsf{FS}_0.\mathsf{Dec}(c_l) = 0$ and thus $\left[\left\lfloor \frac{\mathsf{sk}_i}{t} \right\rfloor\right]_t = t-1$ or, $\left[\left\lfloor \frac{\mathsf{sk}_i}{t} \right\rfloor\right]_t = \mathsf{FS}_0.\mathsf{Dec}(c_l) - 1$ otherwise. So $\mathsf{FS}_0.\mathsf{Dec}(c_l) - 1 \mod t$ gives away bits $l, l+1, ..., 2l-1$ of sk_i.

A CCA1 adversary may then further proceeds similarly by exploiting ill-formed ciphertext $c_{2l} = \left(-\frac{\Delta}{t^2}\mathbf{1}_i, 0\right) = (-2^{k-3l}\mathbf{1}_i, 0)$ and his/her knowledge of $[\mathsf{sk}_i]_{t^2}$ (from above) to retrieve bits $2l, 2l+1, ..., 3l-1$ of sk_i, and so on and so forth. Let us emphasize that this attack is non-adaptive.

G.2 An adaptative noise recovery attack

Another line of folklore attacks follows a dichotomic search pattern (which has been used numerous times, e.g. [44,23]) where the adversary starts from a well-formed encryption of 0 $c_0 = (a, b)$ with $b = \langle a, \mathsf{sk}\rangle + e)$ to find the critical value α^* such that $\mathsf{Dec}(a, b + \alpha^*) \neq 0$ (i.e. such that $e + \alpha^* = \frac{\Delta}{2}$. The adversary may then conclude that the noise in c_0 is equal to $\frac{\Delta}{2} - \alpha^*$ thereby getting one linear equation in sk. By repeating this (adaptive) process over n encryptions of 0, the adversary can eventually retrieve sk by means of linear algebra techniques.

Remark that this attack does not use the legit add-by-const operator wihch only allows to add multiples of Δ to the b-term of an LWE pair.

H Yet another CPA^D attack path on "exact" FHE

Interestingly, in the wake of [23,24], the attack in Proposition 18 gives us yet another attack path on the vanilla Regev scheme, its RLWE variant and the other mainstream FHE schemes which are based on them.

For simplicity sake, we state the attack against vanilla RLWE encryption with a binary secret key $\mathsf{sk} \in \{0,1\}^n$. In such a condition, remark that the ill-formed ciphertext $(\Delta \cdot 1, 0)$ decrypts to sk (similarly to the attack in Sect. G.1). Let $c^{\triangle}$ denotes this ciphertext.

The CPA^D attack then goes as follows:

- The adversary, say $\mathcal{A}$, requests encryptions of 0, until it gets a generating set of vectors for $\mathbb{Z}_q^{2n}$ [24]. Let $c_0, ..., c_{N-1}$ denotes these ciphertexts.
- Using elementary linear algebra, the adversary then finds a linear combination with coefficients in $\mathbb{Z}_q$ such that,

$$c^{\triangle} = \sum_{i=0}^{N-1} \pi_i c_i,$$

  with $\pi \in \mathbb{Z}_q^N$.
- For $i \in [\![0, N-1]\!]$, the adversary then builds ciphertext $c_i^{\triangle}$ with the following legit homomorphic operations, which it performs by means of CPA^D game evaluation oracle calls. Assuming $t > 2$, let $k = \lceil \log_{t-1} \pi_i \rceil$ and let the $\pi_{i,j}$'s denote the k digits in the base $(t-1)$ decomposition of π_i, i.e.

$$\pi_i = \sum_{j=0}^{k-1} \pi_{i,j} (t-1)^j.$$

  Then $\mathcal{A}$ performs,

$$c_{i,j}^{\triangle} = \pi_{i,j} \otimes \underbrace{(t-1) \otimes ... \otimes (t-1)}_{j \text{ times}} \otimes c_i,$$

  where $\alpha \otimes c$ denotes $\mathsf{Eval}(\mathrm{mul}_\alpha, c)$, $\alpha \in \mathbb{Z}_t$, and,

$$c_i^{\triangle} = \mathsf{Eval}(\mathrm{sum}, c_{i,0}^{\triangle}, ..., c_{i,k-1}^{\triangle}).$$

- Following this, by an evaluation request of the form $\mathsf{Eval}(\mathrm{sum}, c_0^{\triangle}, ..., c_{N-1}^{\triangle})$, the adversary eventually puts $c^{\triangle}$ in the internal state of its CPA^D challenger.
- Finally, a single decryption requests on state index $|S| - 1$, where S is the internal state of the CPA^D challenger, gives sk to $\mathcal{A}$.

Remark that the above attack also shows that, unless the number of ciphertexts it has access to is suitably restricted, a CPA^D adversary against the vanilla RLWE scheme can get *any* ciphertext into the internal state of a CPA^D challenger and, as such, has as much power as a CCA adversary (in the special case of these LWE-based schemes).

I The $\widehat{\mathsf{FGP}}$ linearly homomorphic hash function [31]

For illustrative purpose, we briefly describe the Fiore-Gennaro-Pastro hash function introduced in [31] ([32], Sect. 4.2) in order to build several VC schemes on top of the BV scheme [17]. Although FS_2' can be instantiated from any linearly

[24] When $q \geq O(2^\lambda)$ is prime, $2n$ encryptions of 0 are enough (with overwhelming probability).

homomorphic hash function (with domain $\mathbb{Z}_q^N$), this is an example of a concrete candidate.

The hash function consists in interpreting a FS_1 ciphertext as a polynomial in $\mathbb{Z}_q[X][Y]$ with degree $\max(K, n-1)$ in X and degree 1 in Y, i.e. a ciphertext $\mathsf{ct} = (a, B) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1}$ is associated to polynomial

$$p_{\mathsf{ct}}(X, Y) = \sum_{i=0}^{n-1} a_i X^i + Y \sum_{i=0}^{K} B_i X^i.$$

Given α and β, two *secret* values uniformly picked in $\mathbb{Z}_q^2$, the hash function is then simply defined as

$$\mathsf{FGP}_{\alpha,\beta}.\mathsf{Digest}(\mathsf{ct}) = p_{\mathsf{ct}}(\alpha, \beta) = \sum_{i=0}^{n-1} a_i \alpha^i + \sum_{i=0}^{K} B_i \beta \alpha^i.$$

This function is trivially linearly homomorphic (and *not* affine homomorphic). It is further shown in [32] (Theorem 2) that it is universal one-way for $q > 2^\lambda$, although not collision resistant. To achieve collision resistance, [31,32] then extends the hash function to work in the exponent of a multiplicative group $\mathbb{G}$ for which the discrete logarithm is hard, e.g. with a prime modulus $M = q + 1$, with $q = tp$, where p is a suitably large prime (and t is the plaintext modulus[25]),

$$\widehat{\mathsf{FGP}}_{\alpha,\beta}.\mathsf{Digest}(\mathsf{ct}) = g^{p_{\mathsf{ct}}(\alpha,\beta)} \mod M.$$

where g in a generator of $\mathbb{G}$. Note that [31,32] uses bilinear groups rather than cyclic groups in order for the homommorphic property to hold for degree-2 functions which we do not need as the present work explicitly focuses on linear-only homomorphic schemes. In summary, the specification of $\widehat{\mathsf{FGP}}$ is as follows:

- $\widehat{\mathsf{FGP}}.\mathsf{KeyGen}$: uniformly pick $\alpha, \beta \in \mathbb{Z}_q^2$, choose a prime M as above and a generator g of $\mathbb{Z}/M\mathbb{Z}$.
- $\widehat{\mathsf{FGP}}.\mathsf{Digest}$: given $\mathsf{ct} = (a, B) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{K+1}$, return

$$g^{\sum_{i=0}^{n-1} a_i \alpha^i + \sum_{i=0}^{K} B_i \beta \alpha^i} \mod M.$$

- $\widehat{\mathsf{FGP}}.\mathsf{Add}$: given $h, h' \in (\mathbb{Z}/p\mathbb{Z})^2$, return $hh' \mod M$.
- $\widehat{\mathsf{FGP}}.\mathsf{Mulc}$: given $h \in \mathbb{Z}/p\mathbb{Z}$ and $\alpha \in \mathbb{Z}_q$, return $h^\alpha \mod M$.

Still, in terms of practical implications, the fact that the LWE ciphertext modulus connects to the discrete log modulus requires using unusually large LWE moduli (e.g. compared to Sect. D). As already investigated in [31], there are protocols in which function FGP, rather than $\widehat{\mathsf{FGP}}$, can be used (essentially when only one FHE calculation is performed). In such cases, the aforementioned difficulty disappear.

[25] It is desirable that t divides q for a number of reasons (including but not limited to the discussion in Sect. B.4).

To the best of our knowledge, there are only a very limited number of candidate constructions in the state-of-the-art. Other approaches may consider using the Boneh-Freeman lattice-based linearly homomorphic signature scheme [13] (modified to authenticate vectors with coefficients over extension fields $\mathbb{F}_{2^k}$, as explained in that paper) or the SWIFFT hash function [45] (although a variant working over non-binary inputs remains to be defined and studied). Both approaches would lead to constructions consistently only based on lattices.

J More smudging lemmas

J.1 Gaussian smudging

Lemma 24. *Let X denote a centered Gaussian random variable with variance σ^2 and $\varepsilon > 0$, then the statistical distance between the distribution of X and that of $X + \varepsilon$ is bounded by $\frac{\varepsilon}{\sqrt{2\pi}\sigma}$.*

Proof. Let $f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ and $f_{X+\varepsilon}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\varepsilon)^2}{2\sigma^2}}$. We have,

$$\begin{aligned} f_X(x) - f_{X+\varepsilon}(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} - \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\varepsilon)^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi}\sigma} \left(e^{-\frac{x^2}{2\sigma^2}} - e^{-\frac{x^2-2x\varepsilon+\varepsilon^2}{2\sigma^2}} \right) \\ &= \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \left(1 - e^{\frac{2x\varepsilon-\varepsilon^2}{2\sigma^2}} \right). \end{aligned}$$

Then $f_X(x) - f_{X+\varepsilon}(x) \geq 0$ when $e^{\frac{2x\varepsilon-\varepsilon^2}{2\sigma^2}} \leq 1$ i.e., for $\frac{2x\varepsilon-\varepsilon^2}{2\sigma^2} \leq 0$, so

$$x \leq \frac{\varepsilon}{2} = \varepsilon_0.$$

It thus follows that,

$$\begin{aligned} d(f_X, f_{X+\varepsilon}) &= \frac{1}{2} \int_{-\infty}^{+\infty} |f_X(x) - f_{X+\varepsilon}(x)| dx \\ &= \frac{1}{2} \int_{-\infty}^{\varepsilon_0} f_X(x) - f_{X+\varepsilon}(x) dx + \frac{1}{2} \int_{\varepsilon_0}^{+\infty} f_{X+\varepsilon}(x) - f_X(x) dx \\ &= F_X(\varepsilon_0) - F_{X+\varepsilon}(\varepsilon_0) \\ &= F_X(0) + \int_0^{\varepsilon_0} f_X(x) dx - \left(F_{X+\varepsilon}(\varepsilon) - \int_{\varepsilon_0}^{\varepsilon} f_{X+\varepsilon}(x) dx \right) \\ &= \frac{1}{2} + \int_0^{\varepsilon_0} f_X(x) dx - \frac{1}{2} + \int_{\varepsilon_0}^{\varepsilon} f_{X+\varepsilon}(x) dx \\ &= 2 \int_0^{\varepsilon_0} f_X(x) dx \\ &\leq 2 f_X(0) \varepsilon_0. \end{aligned}$$

Since $2 f_X(0) \varepsilon_0 = \frac{2}{\sqrt{2\pi}\sigma} \frac{\varepsilon}{2} = \frac{\varepsilon}{\sqrt{2\pi}\sigma}$, the claim follows.

Lemma 25. *Let $\varepsilon \in [-B, B]$ be some fixed value and X denote a centered Gaussian random variable with variance $\sigma^2 = \frac{2^{2\lambda}B^2}{2\pi}$ then $d(f_X, f_{X+\varepsilon}) \leq neg(\lambda)$.*

Proof. Following Lemma 24, choosing σ^2 such that

$$2^{-\lambda} = \frac{|\varepsilon|}{\sqrt{2\pi}\sigma},$$

i.e. $\sigma = \frac{2^{\lambda}B}{\sqrt{2\pi}}$, leads to

$$d(f_X, f_{X+\varepsilon}) \leq d(f_X, f_{X+B}) \leq 2^{-\lambda}.$$

The above Lemma is useful as it shows that a Gaussian noise with an appropriately large variance can "smudge out" a constant value and, more generally, any random variable following a distribution with a bounded support. We can further extend it in order to "smudge out" a Gaussian noise, as we do just below.

Lemma 26 (Same as Lemma 4). *Let ε and X be centered Gaussian random variables with respective variance σ_0^2 and σ_1^2, with $\sigma_1^2 = \frac{(\lambda+1)2^{2\lambda}\sigma_0^2 \log 2}{\pi}$, then $d(f_X, f_{X+\varepsilon}) \leq neg(\lambda)$.*

Proof. Recall that the Chernoff bound for the Gaussian distribution tells that,

$$P(|\varepsilon| \geq B) \leq 2e^{-\frac{B^2}{2\sigma_0^2}}.$$

Let us consider B_0 such that $2e^{-\frac{B_0^2}{2\sigma_0^2}} = 2^{-\lambda}$ i.e.,

$$B_0 = \sigma_0\sqrt{2(\lambda+1)\log 2}.$$

Then $\varepsilon \in [-B_0, B_0]$ with probability $1 - \mathrm{neg}(\lambda)$. The claim then follows from Lemma 25. □

Since $\mathrm{neg}(\lambda) = O(2^{-\lambda})$, we eventually get the "Smudging lemma for Gaussians" (e.g. notably used in [49]).

Lemma 27 (Smudging Lemma for Gaussians (SLG)). *Let ε and X be centered Gaussian random variables with respective variances σ_0^2 and $\sigma_1^2 = 2^{2\lambda}\sigma_0$, then $d(f_X, f_{X+\varepsilon}) \leq neg(\lambda)$.*

J.2 Triangle smudging

Recall that the triangle distribution over $[\![-B, B]\!]$ is such that

$$P(X = k) = \frac{B + 1 - |k|}{(B+1)^2}.$$

It is well-known that the triangle distribution is the distribution followed by the difference of two discrete uniform random variables over $[\![0, B]\!]$.

Lemma 28. *Let X be a random variable following the triangle distribution over $[\![-B, B]\!]$ and let $e \in \mathbb{N}^*$, then $d(f_X, f_{X+e}) \leq \frac{e+2}{B}$.*

Proof.

$$
\begin{aligned}
d(f_X, f_{X+e}) &= \frac{1}{2}\sum_{k=-B}^{B+e} |P(X=k) - P(X+e=k)| \\
&= \frac{1}{2}\left(\sum_{k=-B}^{\lfloor e/2 \rfloor} P(X=k) - P(X+e=k) + \sum_{k=\lfloor e/2 \rfloor+1}^{B+e} P(X+e=k) - P(X=k)\right) \\
&= \frac{1}{2}\left(P(X \leq \lfloor e/2 \rfloor) - P(X+e \leq \lfloor e/2 \rfloor) + P(X+e \geq \lfloor e/2 \rfloor + 1) - P(X \geq \lfloor e/2 \rfloor + 1)\right) \\
&= P(X \leq \lfloor e/2 \rfloor) - P(X+e \leq \lfloor e/2 \rfloor) \\
&= P(X \leq -1) + P(0 \leq X \leq \lfloor e/2 \rfloor) - \underbrace{P(X+e \leq e-1)}_{P(X \leq -1)} + P(\lfloor e/2 \rfloor + 1 \leq X+e \leq e-1) \\
&= P(0 \leq X \leq \lfloor e/2 \rfloor) + P(\lfloor e/2 \rfloor + 1 \leq X+e \leq e-1) \\
&\leq P(0 \leq X \leq \lfloor e/2 \rfloor) + P(\lceil e/2 \rceil \leq X+e \leq e) \\
&\leq P(0 \leq X \leq \lfloor e/2 \rfloor) + P(-\lfloor e/2 \rfloor \leq X \leq 0) \\
&= 2P(0 \leq X \leq \lfloor e/2 \rfloor) \\
&\leq 2P(X=0)(\lfloor e/2 \rfloor + 1) \\
&= \frac{2(\lfloor e/2 \rfloor + 1)}{B+1} \\
&\leq \frac{e+2}{B+1} \\
&\leq \frac{e+2}{B}.
\end{aligned}
$$

□

Lemma 29 (Triangle Smudging Lemma). *Let B_0 and B_1 be two positive integers and let $e_0 \in [\![-B_0, B_0]\!]$ be a fixed integer. Let e_1 be chosen following the triangle distribution in $[\![-B_1, B_1]\!]$, then if $B_1 \geq 2^\lambda(B_0+2)$, the statistical distance between the distribution of e_1 and that of $e_0 + e_1$, $d(f_{e_1}, f_{e_1+e_0})$, is bounded by $neg(\lambda)$.*

Proof. Following Lemma 28, we have that

$$d(f_{e_1}, f_{e_1+e_0}) \leq \frac{B_0+2}{B_1}.$$

Letting $2^{-\lambda} = \frac{B_0+2}{B_1}$ leads the claim. □

Table of Contents

Achieving "beyond CCA1" security for linearly homomorphic encryption, without SNARKs? . . . 1
Marina Checri, Pierre-Emmanuel Clet, Marc Renard and Renaud Sirdey
1 Introduction . . . 2
1.1 Summary of contributions . . . 3
1.2 Paper organization . . . 4
2 Preliminaries . . . 4
2.1 Basic notations . . . 4
2.2 Basic definitions . . . 5
2.3 Security notions . . . 6
CPA^D (and CPA_0^D) security. . . . 6
vCCA, vCCA^D (and CCA0) security. . . . 7
2.4 Smudging . . . 8
2.5 The Linear-Only Homomorphism (LOH) assumption . . . 10
3 Results under the correctness assumption . . . 12
3.1 General black-box results . . . 13
3.2 A public-key vCCA secure construction based on Paillier . . . 15
4 LWE-based constructions . . . 17
4.1 The basic Regev Scheme (FS_0) . . . 18
4.2 Achieving (non-adaptive) CPA^D security (FS_1) . . . 18
Preliminaries. . . . 18
Scheme FS_1. . . . 19
CPA_0^D security of FS_1. . . . 20
4.3 Achieving "non-adaptive" LOH (FS_1') . . . 23
$\mathsf{FS}_0^{(K)}$ and FS_1'. . . . 23
CPA_0^D security of FS_1'. . . . 25
FS_1' and LOH. . . . 26
5 Concluding remarks . . . 28
A Appendix sections organization . . . 32
B Achieving vCCA^D security, under the conjectured (adaptive) CPA^D security of FS_1' (and FS_1) . . . 32
B.1 The ℓ-LOH property . . . 32
B.2 Linearly homomorphic hash functions . . . 34
B.3 Scheme FS_2' . . . 34
B.4 ℓ-vCCA^D security of FS_2' . . . 35
C General results for approximate schemes? . . . 38
D Choosing the ciphertext modulus for FS_1 . . . 39
E Additional security notions . . . 41
E.1 Additional variants of CPA^D security . . . 41
E.2 CPA and CCA1 security. . . . 41

E.3 Separation results for CCA0 42
F The Paillier cryptosystem 43
G Folklore CCA1 attacks against Regev 44
G.1 A non-adaptive ill-formed ciphertext-based attack 44
G.2 An adaptative noise recovery attack 44
H Yet another CPA^D attack path on "exact" FHE 44
I The $\widehat{\mathsf{FGP}}$ linearly homomorphic hash function [31] 45
J More smudging lemmas 47
J.1 Gaussian smudging 47
J.2 Triangle smudging 48